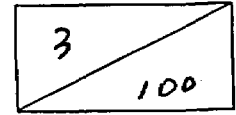


최종보고서



# 자료처리 및 3차원 지형분석 S/W 개발

Development of Image Processing and  
3-D Terrain Analysis S/W

연구기관  
한국전자통신연구원

과학기술부

# 제 출 문

과학기술부 장관 귀하

본 보고서를 “자료처리 및 3차원 지형분석 소프트웨어 개발”의 보고서로 제출합니다.

1998. 11. 30.

주관연구기관명	: 한국전자통신연구원
주관연구책임자	: 안 종현
연 구 원	: 신 대혁
	: 권 태영
	: 김 경옥
	: 정 인숙
	: 강 태호
	: 김 경숙
	: 박 종현
	: 윤 기준
	: 김 주완
	: 최 정단
	: 성 민규
	: 김 용일(서울대학교)
	: 김 태정(KAIST)
위탁연구기관명(I)	: 중앙대학교
위탁연구 책임자	: 윤 경현
연 구 원	: 류 승택
	: 하 정호
	: 김 성철
	: 한 주연



	: 최 윤석
	: 정 은종
	: 심 영수
	: 송 지호
위탁연구기관명(II)	: KAIST
위탁연구 책임자	: 이 흥규
연 구 원	: 백 윤주
	: 김 형신
	: 김 길운
	: 오 황석
	: 강 성환
	: 이 귀영
참 여 기 업	: 한국항공
	: 유니세크
	: 공간기술정보
	: 현대전자산업
	: 쌍용정보통신
	: 레이시시스템
	: 거림시스템
	: 한조엔지니어링
	: 캐드랜드
연 구 원	: 이 강원(한국항공)
	: 윤 종성(한국항공)
	: 김 충평(한국항공)
	: 나 방현(현대전자산업)
	: 최 효진(쌍용정보통신)
	: 이 종수(유니세크)
	: 원 태연(유니세크)
	: 정 규장(유니세크)
	: 박 창택(공간기술정보)
	: 한 덕상(레이시시스템)
	: 박 태수(레이시시스템)
	: 조 남희(레이시시스템)
	: 류 중희(캐드랜드)
	: 김 세일(거림시스템)

# 요 약 문

## I. 제 목

자료처리 및 3차원 지형분석 소프트웨어 개발

## II. 연구개발의 목적 및 필요성

위성에 의하여 관측된 자료는 최종 사용자가 이용할 수 있는 정보로 변환되기까지는 여러 단계의 복잡한 과정을 거쳐야 한다. 일반 사용자에게 제공되는 위성자료는 보통 위성 자세에 대한 보정, 지구 회전에 대한 보정 및 센서 보정이 완료된 이른바 bulk 전처리 과정을 거친 것으로 사용자에게 따라 다른 위성영상 처리 프로그램을 사용하여 목적에 맞게 분석한 정보를 이용하고 있다.

그러나 최근 위성자료의 다양화와 대용량화 추세에 따라 위성정보 처리를 위한 자동화 기술이 요구되며, 또한 수년 이내에 상용화되어 GIS분야에서 크게 활용될 것으로 기대되는 고분해능 위성영상(high resolution sensor image) 처리 및 초다대역 분광분해능 위성자료(hyperspectral sensor image)로부터 정보추출을 위한 위성자료의 응용 분석 기술 및 S/W의 개발이 요구된다. 이를 위해서 기본적인 자료처리 기능을 구비한 위성정보 처리 엔진의 개발이 요구되며, 향후 이를 바탕으로 부가적인 기능을 갖는 차세대 자료 처리 모듈을 추가시킬 수 있도록 자체적인 기술력의 확보가 필요하다.

현재 산업 및 사회구조의 정보화에 따라 국내의 위성정보에 대한 수요가 날로

증가하고 있는 추세이며, 또한 고성능 컴퓨터의 출현과 보급의 확대, 초고속 정보통신망의 구축, GIS 시장의 활성화와 같은 정보인프라 구축에 대한 수요의 증대는 여러 분야에서의 위성영상의 활용범위를 급속히 증대시키고 있다. 따라서, 앞으로 다가올 위성영상 활용 시장의 급성장에 대비하여, 국내의 관련 기반기술을 확고히 하고, 현재의 외국산 소프트웨어에 대응할 수 있는 소프트웨어를 개발함으로써 조만간 형성될 위성영상 관련 시장에 대비하여야 한다. 또한 차세대 센서에 의하여 획득되는 위성영상의 상품화에 대비한 관련기술 및 제품의 개발에 박차를 가하고 있는 선진국에 대하여 우리나라도 적극 대응하여, 필요 기술의 개발이 적시에 이루어지지 않는다면, 국내의 기술과 선진국의 기술 격차는 더욱 심화될 것이며, 궁극적으로는 외국 기술에 종속되어 외국 제품을 구입하여 사용해야 하는 형편이 될 것이다.

이와 같은 상황에 대처하기 위하여 본 연구는 위성영상과 수치표고모델을 자료 입력원으로 하여 GIS에서 필요로 하는 광역 지표면에 대한 정보를 추출하고, 3차원적 지형분석이 가능한 소프트웨어의 개발을 주목적으로 한다.

### III. 연구개발의 내용 및 범위

본 연구에서는 위성영상과 수치표고정보를 자료 입력원으로 하여 GIS 응용 분야에서 필요로 하는 광역 지표면에 대한 정보의 추출과 3차원적인 지형분석이 가능한 소프트웨어를 개발하는 것이다. 본 과제 of 개발 범위 및 내용은 다음과 같다.

1. 대용량 위성영상의 효율적 처리를 위한 자료처리 엔진 개발
2. 위성 영상의 파일 입출력 및 화면 출력 모듈 개발
4. 위성 영상 처리 모듈 개발
5. 위성 영상을 활용한 지표면 정보 추출 모듈 개발
6. 수치 지형 모델(Digital Elevation Model)을 이용한 3차원 지형 모델링 모듈 개발
7. Computer Graphic 기술을 이용한 3차원 지형 렌더링 모듈 개발
8. GIS 활용을 위한 지형 특성 분석 모듈 개발
9. 지형 시뮬레이션을 위한 3차원 fly 모듈 개발

## IV. 연구개발결과의 활용계획

본 연구에서 개발된 영상처리 및 지형 분석 모듈은 일반적으로 사용하는 기능들을 우선적으로 구현한 것으로 연구의 성과물을 활용하기 위하여 보다 다양한 기능의 추가와 함께 민간기업에 의한 상품화가 이루어져야 한다. 이를 위하여 민간 기업으로의 기술이전을 통하여 상품화를 추진하고, 상품화된 S/W의 성능 향상 및 유지보수 등 관련된 서비스가 지원하기 위해서 자체적으로 개발인력을 확보하고 있고, 기존의 벡터 기반의 GIS툴을 가지고 있는 기업과 연계되어 하나의 패키지로서 공급 가능한 기업을 우선적으로 선정하여야 한다.

아직까지 국내 원격탐사 관련 S/W 시장의 규모는 그다지 크지 않으나, 1999년부터 상용화될 것으로 기대되는 고분해능 위성영상과 다목적 실용위성의 개발 완료로 상당한 신규 수요가 생길 것으로 예측되며, 이들 수요자는 대부분 단순 원격탐사 자료의 처리보다는 GIS와 연계된 분석을 원하는 부류로 판단되므로, 철저한 시장 조사 및 현재 상용화되어 있는 많은 외국산 S/W와의 차별화와 함께 취약한 분야에 대한 틈새시장을 개척하여 점차 상품의 인지도를 높이며, 중국 등 신규 시장의 진출을 꾀하여야 한다.

한편, 정부에서는 민간 기업이 자체적인 기술력을 확보할 때까지 최신 기술의 도입 및 S/W 성능향상이 가능하도록 각종 정책을 통하여 지원해야 하며, 다목적 실용위성을 통하여 얻어지는 위성영상의 교육적 활용을 위한 교육용 S/W의 제작도 가능하다.

여 백

# S U M M A R Y

## I. Title

The title of this research is "Development of Image Processing and 3-D Terrain Analysis S/W"

## II. Objectives and the Importance

The objectives of this research is to develop a satellite image processing and three dimensional terrain analysis software for GIS(Geographic Information System) applications.

As for GIS software R&D trend and market of the 21th century, the following things have been pointed out : three-dimensional geo-processing technologies, Internet based application system development, distributed processing technologies for large volume of spatial information, real-time geo-processing technologies. On the other hands, several commercial high resolution satellite projects have been started since 1995 and expected that remote sensing data will be used more active in GIS fields as effective spatial information acquisition tools. Scheduled for launch is a next generation of imaging satellites that offers spatial, spectral and temporal resolutions never before available to the public. The term 'GeoSpatial Imagery' best describes the powerful and accurate information that will be contained in this unique, digital imagery.

From those expectation, several major R/S software company like ERDAS, PCI was started developing Geo-Imaging software which has more tight connection with GIS spatial analysis functions. Moreover, new GIS market trends for 3D-GIS such as ERDAS Imagine Virtual GIS, Arc/View 3D Analyst

requires various raster-based terrain analysis functions linked with vector-based spatial analysis functions.

Therefore, the products of this research - GeoPixel for remote sensing data processing and GeoPixel-Terrain for terrain analysis - will have important meaning in domestic remote sensing and 3D-GIS fields.

### III. Contents and Result

This research was performed as a part of NGIS projects during three years since Dec. 1995. The main target of the first year of proposed three year research was the basic design of data processing system and analysis of image processing algorithm. The target of the second year is to implementation of designed system using Java for remote sensing data processing and C++ for terrain analysis. In this period, prototype satellite data processing system based on multi-thread and object pipeline concepts was developed and tested in various CPU capacity. The third year's target is to implemented GeoPixel 1.0 which is first software written Java language for remote sensing data processing, and GeoPixel-Terrain software was developed using C++ also.

From this view points, the contents and the result of this research can be divided into two major sections : GeoPixel for remote sensing data processing and GeoPixel-Terrain for terrain analysis.

#### A. Remote Sensing Data Processing S/W - GeoPixel

- development of core module for efficient treatment of huge volume of remote sensing data
- development of data I/O module and screen display module
- development of image processing and data analysis module such as image enhancement, principal component analysis, classification for land use
- development of extraction on available surface information such as

temperature, radiance, reflectance, albedo

#### B. Terrain Analysis S/W - GeoPixel-Terrain

- development of three dimensional terrain modelling module using DEM
- development of three dimensional terrain rendering module using computer graphic techniques.
- development of terrain analysis module such as slope, watershed, visibility, shortest path for GIS application
- development of three dimensional flight simulation module

## IV. Application Plans of Research Results

The significant results obtained from this three-year research projects are thought to be regarded as leading technologies in the domestic remote sensing fields, at both practical basis and research basis.

To improve and practical use of this results, it must be develop commercial S/W via the transfer of technical know-how to private company which has man power for RS-related software developments.



# 여 백

# C O N T E N T S

Chapter 1. Introduction .....	19
Section 1. Research objective .....	19
Section 2. Research necessity .....	19
Section 3. Research scope .....	21
Chapter 2. Technical situation of domestic and overseas .....	23
Section 1. Domestic .....	23
Section 2. Overseas .....	24
Chapter 3. Contents and Results .....	27
Section 1. System Design .....	27
Section 2. Developed modules of GeoPixel for remote sensing data processing .....	28
Section 3. Developed modules of GeoPixel-Terrian for terrain analysis .....	34
Chapter 4. Java for satellite image processing .....	35
Section 1. Processing speed and stability .....	36
Section 2. Parallel processing .....	40
Chapter 5. GeoPixel: Satellite image processing S/W .....	47
Section 1. Data structure .....	47
Section 2. Core program structure .....	56
Section 3. Image processing modules .....	60
Section 4. User interface .....	78
Section 5. Land-cover classification example .....	86

Chapter 6. GeoPixel-Terrain S/W for terrain analysis .....	93
Section 1. 3-Dimensional Terrain Modelling .....	93
Section 2. 3-Dimensional Visualization .....	102
Section 3. Terrain Analysis .....	104
Section 4. Flight Simulation .....	117
Section 5. Example of terrain analysis .....	122
Chapter 7. Conclusion .....	127
Chapter 8. References .....	129
Appendix 1. Benchmark with commercial S/W .....	133
Appendix 2. Research results .....	137
Appendix 3. Research objectives accomplishment .....	139
Appendix 4. GeoPixel API document .....	141
Appendix 5. GeoPixel source codes for geometric correction .....	669
Appendix 6. GeoPixel-Terrain API document .....	693

## List of Figures

Figure 4-1 Basic structure of MTDIP Model .....	41
Figure 4-2 Data exchange and synchronization between operators through object pipeline .....	42
Figure 4-3 Program structure for Sobel filtering based on MTDIP model .....	43
Figure 4-4 Comparison of speedup between MTDIP-based and sequential Sobel filtering. Speedup is the ratio of the computing speed of n processors to that of 1 processor. AQUANTA HS/6 of UNISYS, 128MB RAM, 512KB cache, Window NT 4.0 Server OS, Symantec Java! ByteCode Compiler ver.210.065 are used. i8192(s) means the speedup of sequential processing on I8912 image .....	44
Figure 5-1 Systematic structure for image display on screen .....	61
Figure 5-2 Image display on scroll, main, zoom window .....	62
Figure 5-3 Interface of Lookup Table for image enhancements .....	62
Figure 5-4 Interface of Palette for changing pixel colors .....	63
Figure 5-5 Interface of Area of Interest(AOI) for define reference area .....	64
Figure 5-6 Interface for vector drawing on raster image .....	64
Figure 5-7 Interface for setting of cursor type, rectangle color and grid. ....	65
Figure 5-8 Example of edge detection using sobel operator .....	69
Figure 5-9 Example of fast forward fourier transform and inverse filtering ....	69
Figure 5-10 Example of Histogram .....	71
Figure 5-11 Example of Scattergram .....	71
Figure 5-12 Example of Land Cover Classification using MLC method .....	72
Figure 5-13 Example of extraction of surface temperature .....	73
Figure 5-14 Example of extraction vegetation index(NDVI) .....	74
Figure 5-15 Example of principal component analysis .....	74
Figure 5-16 Example of tasseledcap transform .....	75
Figure 5-17 Example of binary arithmetic operation(overlay) .....	76

Figure 5-18 Example of extraction partial image .....	77
Figure 5-19 Example of resizing with arbitrary scale factor .....	77
Figure 5-20 Example of flip/translation/rotation of image .....	78
Figure 5-21 General structure of user interface for keyword search .....	80
Figure 5-22 Access of processing module via keyword search .....	81
Figure 5-23 Example of command description file and frame .....	83
Figure 5-24 Structure of user interface for programmable menu .....	84
Figure 5-25 Example of menu description .....	85
Figure 5-26 Example of menu description file and generated frame .....	86
Figure 6-1 Terrain modelling using rectangular grid .....	94
Figure 6-2 Graphical properties of terrain specific points .....	95
Figure 6-3 Extracted terrain specific points from DEM .....	96
Figure 6-4 Voronoi Diagram and Delaunay Triangulation .....	97
Figure 6-5 Correction of Triangle .....	98
Figure 6-6 General process of radial sweep triangulation .....	99
Figure 6-7 Comparison of Delaunay and Radial sweep triangulation .....	101
Figure 6-8 Result of delaunay triangulation .....	101
Figure 6-9 View point transformation and projection .....	102
Figure 6-10 Terrain representation in different view point .....	103
Figure 6-11 Terrain representation with pseudo shading .....	104
Figure 6-12 Calculation of volume of terrain cutting using unit-cell area .....	105
Figure 6-13 volume analysis of terrain cutting .....	107
Figure 6-14 Relationship between Intervisibility and Blocking Object .....	110
Figure 6-15 Two-block test for line visibility .....	111
Figure 6-16 Shortest path analysis over terrain .....	114
Figure 6-17 Definition of height plane .....	115
Figure 6-18 Exceptional case in plane intersection .....	115
Figure 6-19 Contour extraction from DEM .....	116
Figure 6-20 Contour extraction from TIN .....	116

Figure 6-21	Generated Flight path using spline function with control points .....	118
Figure 6-22	Flight simulation using wireframe .....	119
Figure 6-23	Flight simulation using pseudo shading .....	119
Figure 6-24	Flight simulation using texture mapping of satellite data .....	120
Figure 6-25	Flight simulation using TIN wireframe .....	121
Figure 6-26	Flight simulation using TIN pseudo shading .....	121

# 여 백

## List of Tables

Table 4-1 Computation time for array operation .....	37
Table 4-2 Computation time for array construction and destruction .....	38
Table 4-3 Images for measurement of parallel processing speed .....	43
Table 6-1 Geometrical properties of terrain specific points .....	95



여 백

# 목 차

제 1 장 서론 .....	19
제 1 절 연구개발의 목적 .....	19
제 2 절 연구개발의 필요성 .....	19
제 3 절 연구개발의 범위 .....	21
제 2 장 국내외 기술개발 현황 .....	23
제 1 절 국내 기술 현황 .....	23
제 2 절 해외 기술 현황 .....	24
제 3 장 연구개발 수행 내용 및 결과 .....	27
제 1 절 위성영상 처리 및 지형분석 S/W 설계 .....	27
제 2 절 개발된 위성 영상 처리 S/W GeoPixel 모듈 .....	28
제 3 절 개발된 지형분석 S/W GeoPixel-Terrain 모듈 .....	34
제 4 장 위성영상 처리에서 자바 언어의 이용 가능성 .....	35
제 1 절 처리 속도와 안정성 .....	36
제 2 절 병렬처리 .....	40
제 5 장 GeoPixel: 위성영상 통합 처리 소프트웨어 .....	47
제 1 절 자료구조 .....	47
제 2 절 핵심동작 원리 .....	56
제 3 절 영상처리 모듈 .....	60
제 4 절 사용자 인터페이스 .....	78
제 5 절 개발된 위성영상 처리 기능을 이용한 토지 이용 분류 .....	86
제 6 장 3차원 지형 분석 S/W GeoPixel-Terrain 개발 .....	93
제 1 절 3차원 지형 모델링 .....	93

제 2 절 3차원 가시화 .....	102
제 3 절 지형 분석 .....	104
제 4 절 비행 시뮬레이션 .....	117
제 5 절 개발된 지형 분석 기능을 이용한 지형 분석 .....	122
제 7 장 결론 .....	127
제 8 장 참고문헌 .....	129
부록 1. 상용 S/W와의 기능 비교 .....	133
부록 2. 연구 발표 성과 .....	137
부록 3. 연구 개발 목표 달성도 및 대외 기여도 .....	139
부록 4. GeoPixel API 설명서 .....	141
부록 5. 기하 보정을 위한 GeoPixel 소스 코드 .....	669
부록 6. GeoPixel-Terrain API 설명서 .....	693

# 제 1 장 서 론

## 제 1 절 연구개발의 목적

본 연구는 위성영상과 수치표고모델을 입력원으로 하여 GIS에서 필요로 하는 광역 지표면에 대한 정보를 추출하고, 3차원적 지형분석이 가능한 소프트웨어의 개발을 주목적으로 한다.

## 제 2 절 연구개발의 필요성

### 1. 기술적 측면

위성에 의하여 관측된 자료는 최종 사용자가 이용할 수 있는 정보로 변환되기까지는 여러 단계의 복잡한 과정을 거쳐야 한다. 일반 사용자에게 제공되는 위성자료는 보통 위성 자세에 대한 보정, 지구 회전에 대한 보정 및 센서 보정이 완료된 이른바 bulk 전처리 과정을 거친 것으로 사용자에게 따라 다른 위성영상 처리 프로그램을 사용하여 목적에 맞게 분석한 정보를 이용하고 있다.

그러나 최근 위성자료의 다양화와 대용량화 추세에 따라 위성정보 이용 방법에 자료처리의 자동화 기술이 요구되며, 또한 수년 이내에 상용화되어 GIS분야에서 크게 활용될 것으로 기대되는 고분해능 위성영상(high resolution sensor image) 처리 및 초다대역 분광분해능 위성자료(hyperspectral sensor image)로부터 정보추출을 위한 위성영상으로부터 다양한 정보를 추출하는 위성자료의 응용 분석 기술 및 S/W의 개발이 요구된다.

이와 같은 차세대 위성정보 처리 S/W의 개발을 위하여는 기본적인 자료처리 기능을 구비한 위성정보 처리 엔진의 개발이 요구되며, 향후 이를 바탕으로 부가적인 기능을 갖는 모듈을 추가시킬 수 있도록 자체적인 기술력의 확보가 필요하다.

## 2. 경제 산업적 측면

다목적 실용위성의 개발 및 고해상도 위성영상의 상용화는 우리나라나 세계 관련시장의 수요증가와 함께 활용분야의 다변화가 예상된다. 현재 산업 및 사회구조의 정보화에 따라 국내의 위성정보에 대한 수요가 날로 증가하고 있는 추세이며, 또한 고성능 컴퓨터의 출현과 보급의 확대, 초고속 정보 통신망의 구축, GIS 시장의 활성화와 같은 정보인프라 구축에 대한 수요의 증대는 여러 분야에서 위성영상의 활용범위를 급속히 증대시키고 있다. 국토 정보의 전산화와 국가 GIS의 구축에 따라 위성정보의 실시간적인 이용이 필요하며, 또한 국토 자원의 효율적인 개발 및 관리 유지, 개발 계획의 수립, 자연 환경의 변화 감시, 재해 감시 등 최신의 위성자료를 활용하여 효율적으로 이용되어야 한다.

따라서, 앞으로 다가올 위성영상 활용 시장의 급성장에 대비하여, 국내의 관련 기반 기술을 확고히 하고, 현재의 외국산 소프트웨어에 대응할 수 있는 소프트웨어를 개발함으로써 조만간 형성될 위성영상 관련 시장에 대비하여야 한다. 또한 차세대 센서에 의하여 획득되는 위성영상의 상품화에 대비한 관련기술 및 제품의 개발에 박차를 가하고 있는 선진국에 대하여 우리나라도 적극 대응하여, 필요 기술의 개발이 적시에 이루어지지 않는다면, 국내의 기술과 선진국의 기술 격차는 더욱 심화될 것이며, 궁극적으로는 외국 기술에 종속되어 외국 제품을 구입하여 사용해야 하는 형편이 될 것이다.

## 3. 사회 문화적 측면

위성영상의 활용은 벡터형 GIS와 중첩한 분석이나 의사결정에서 사용자에게 시각적인 정보와 비경험자라 할지라도 직관적인 해석이 가능케 한다. 이는 일반 사용자들도 쉽게 멀티미디어 지도를 제작할 수 있다는 것에서 아주 중요한 의미를 갖는다. 또한 위성정보의 고분해능화와 초다대역화는 그 활용범위를 매우 광범위하게 넓히게 된다. 지자체의 입장에서는 관할 구역의 시설물 관리에 고분해능 영상이 유용한 도구로 사용될 수 있으며, 정밀한 토지 이용 분류, 산불, 홍수 피해 등 재해 조사에서는 초다대역 위성영상이 유용하게 이용된다. 이외에도 관광사진 등 지역 안내도 등에서도 위

성영상의 활용은 벡터 정보에서 생략되어 있는 많은 정보를 제공할 수 있으며, 실제 지형 및 경관을 위성영상과 수치표고모델을 이용하여 재현해 내는 등, 각종 GIS와 관련된 분야에서 위성영상의 역할은 매우 커질 것으로 예상된다.

### 제 3 절 연구개발의 범위

본 연구에서의 개발 범위는 다음과 같다.

1. 대용량 위성영상의 효율적 처리를 위한 자료처리 엔진 개발
2. 위성 영상의 파일 입출력 및 화면 출력 모듈 개발
3. 위성 영상 전처리 모듈 개발
4. 위성 영상 처리 모듈 개발
5. 위성 영상을 활용한 지표면 정보 추출 모듈 개발
6. 수치 지형 모델(Digital Elevation Model)을 이용한 3차원 지형 모델링 모듈 개발
7. Computer Graphic 기술을 이용한 3차원 지형 렌더링 모듈 개발
8. GIS 활용을 위한 지형 특성 분석 모듈 개발
9. 지형 시뮬레이션을 위한 3차원 fly 모듈 개발

여 백

## 제 2 장 국내외 기술 현황

### 제 1 절 국내 기술 현황

1992년 KAIST 인공위성연구센터와 영국의 서리대학과의 합작에 의한 과학실험용 소위성인 우리별 1호를 궤도에 올린 이후 항공우주연구소를 중심으로 본격적 지구관측위성인 다목적 실용위성 아리랑 1호를 개발하고 있다. 아리랑 1호의 주임무는 한반도의 DEM을 제작하는 것으로 6.6m의 공간해상도를 갖는 1개의 팬크로마틱 밴드와 6개의 밴드를 갖는 해양관측센서로 구성되어 있어, 인공위성 원격탐사에 대한 관심이 높아지고 있다. 특히 아리랑 1호의 등장으로 위성관측 자료의 실시간 수신이 가능하게 되었으나, 이를 뒷받침할 수 있는 응용기술 및 S/W의 개발은 아직 미미한 수준에 머무르고 있어, 현재 국내에서는 상업용으로 혹은 GIS 분야에 있어 실무적으로 사용할 만한 원격탐사 자료처리 S/W를 찾아보기는 어려운 실정이다.

비록, 국내 여러 대학과 연구소에서 위성영상을 처리하기 위한 연구들이 수행되어 컴퓨터 소프트웨어 기술연구소(구 시스템공학연구소)에서는 80년대말과 90년대 초 KMIPS, ERIMS라는 DOS, Window 3.1기반의 S/W를 개발한 바 있으나 OS의 변화에 능동적으로 대처하지 못하고, 고가의 특정 하드웨어에 대한 의존성이 너무 커 상용화까지는 이르지 못하였으며, 1995년 자원연구소에서도 Windows 3.1용의 IMAPRO를 개발하였으나 지속적인 유지 보수가 이루어지지 않아 상업화에는 실패한 바 있다.

비교적 GIS 보다 일찍 시작되었던 국내의 원격탐사 분야에서의 기술개발로 기본적인 위성영상처리 및 관련 기술들은 상당히 많이 확보되었지만, 원천적인 대용량 영상 자료의 효율적인 입출력 및 메모리 관리 기능, GIS와 연계되어 특정 분석을 수행하는 기능적인 면에서는 아직 부족한 점이 많다. 이는 앞으로 위성영상 시장에서 주류로 될 것으로 예상되는 고공간분해능 센서 및 초다대역분광분해능 센서에 의하여 획득되는 위성자료를 이용한 지표면 정보의 추출 및 GIS data layer의 구축에 있어 핵심적인 기술로 국내 기술 개발이 시급한 형편이다..

또한 수치표고모델을 이용한 3차원 지형분석은 일반 사용자가 3차원 정보를 제작하



거나 입수하기가 용이하지 않아 일부 토목관련 분야에서 부분적으로 이용되어 왔으며, 특히 3차원 렌더링과 시뮬레이션에 대하여는 컴퓨터 그래픽과 GIS분석기능이 결합된 것으로 세계적으로도 1997년부터 단순 기능을 갖는 제품이 시장에 등장하기 시작한 개발 초기 단계에 있다 할 수 있으므로 본 과제를 통하여, 3차원적인 지형의 분석과 시각화를 위한 기반 기술이 확보로 시장 성숙단계에서 조기에 진입할 수 있게 되었다 할 수 있다.

## 제 2 절 국외 기술 현황

GIS 기술을 주도하는 선진국에서는 공간 영상 정보를 얻기 위한 방법으로 인공위성 및 항공기를 이용한 원격탐사 기술과 GIS를 결합하는 시도가 많이 행하여지고 있다. 현재 원격탐사 기술의 발전 추세를 정리하여 보면 다음과 같다.

1. 공간 분해능의 고해상도 센서(high resolution sensor) : 1m 정도의 공간 분해능을 갖는 위성영상의 제공. 항공사진 대체 수단으로서 기대됨. 수치사진 측량 및 컴퓨터 비전 관련 기술이 많이 요구되며, 특히 AM/FM 분야에서의 활용이 기대됨.
2. 초 다대역 분광센서(hyperspectral sensor) : 150 - 300개 정도의 분광대역에서의 지표면 정보를 관측함. 정밀한 토지 피복의 분류가 가능함.
3. 마이크로 레이다 센서(microwave radar sensor) : 기상 조건과 관계없이 전천후로 자료를 획득할 수 있음. 일반 광학 센서로 획득할 수 없는 정보의 획득이 가능.

이와 같은 다양한 센서에 의하여 획득되는 위성자료를 처리하기 위하여 현재 상용화되어 있는 많은 S/W들은 기본적인 위성자료 처리 모듈에 각 응용 분야별로 특화된 기능을 갖는 모듈을 옵션으로 제공하고 있다(예 ERDAS사의 Imagine OrthoMax/OrthoBase/StereoBase, Imagine Expert Classifier, OrthoRadar/ StereoSAR DEM, IFSAR DEM, PCI사의 EASI/PACE OrthoEngine, RadarSoft). 특히 1m의 고해

상도 위성영상 자료의 상용화가 1999년 후반부터는 보급될 것이 확실해 짐에 따라 미국의 GIS회사들은 이를 대비한 시스템의 개발에 많은 연구를 수행하고 있으며, 이들과 해상도 위성영상으로부터 보다 용이하게 정밀한 3차원 정보의 추출이 가능해 짐에 따라, 2차원적인 공간 분석 기능을 주로 제공하였던 많은 GIS S/W, 위성영상분석 S/W에서도 3차원적인 공간분석 기능이 강화된 S/W가 1997년부터 시장에 출시되기 시작하였다(예;ESRI사의 3D Analyst, ERDAS사의 Virtual GIS).

여 백

## 제 3 장 연구개발 수행 내용 및 결과

### 제 1 절 위성영상 처리 및 지형분석 S/W 설계

본 연구의 영상 처리 시스템, GeoPixel 및 GeoPixel-Terrain은 다음과 같은 설계 개념을 토대로 구현되었다.

1. 100% Pure Java : GeoPixel은 세계 최초의 100% Pure Java을 이용한 위성 영상 처리 소프트웨어이다. 따라서 Java 프로그래밍 언어가 가지는 멀티플랫폼, 다중 쓰레드, 객체지향과 같은 특성을 고스란히 이어받고 있다. 과거 Java 프로그램에서 문제가 되었던 처리 속도 문제는 JIT(Just In Time) 컴파일러를 사용하여 극복하였다. 한편 GeoPixel-Terrain은 컴퓨터 그래픽에서 많이 활용되는 OpenGL 라이브러리를 이용한 관계로 Visual C++ 5.0 으로 개발되었다. 그러나 현재 개발이 진행 중인 Java 3D API가 공식 발표되면 이식이 가능하도록 구현되어 있다.
2. 객체지향적 설계 : GeoPixel의 모든 프로그램은 객체지향적 설계 개념에 의해서 디자인되었다. 시스템의 하부 구조는 잘 정의된 클래스의 집합으로 정의되어 있으며 모든 명령어는 Supervisor 클래스에서, 모든 실행모듈은 Operator 클래스에서 상속받고 있다. 따라서 상위 클래스의 수정을 통해 시스템의 기능 확장이 용이하고 버그 수정으로 인한 부 효과를 최소화시킬 수 있다.
3. 대용량 영상 처리 : GeoPixel은 대용량 위성 영상을 처리하기 위한 소프트웨어이다. 따라서 대용량 자료에 최적한 시스템 구조를 가지고 있다. 예를 들어 영상의 화면 출력시 세 개의 화면을 동시에 사용하여 가상메모리에서 발생할 수 있는 Swapping을 피하고 있으며 영상 처리 모듈은 대부분 라인 단위로 영상을 읽어들이 처리하도록 구현되었다. 또한 대용량 영상의 처리 속도를 향상시

키기 위해서 BIL(Band Interleaved by Line) 구조를 기존 영상 파일 구조로 사용하였다. 작은 용량의 신속한 처리를 위해서 메모리에 저장하는 영상을 별도로 지원하고 있으며 모든 명령어는 파일에 저장된 영상뿐만 아니라 메모리에 저장된 영상도 처리할 수 있도록 구현되었다.

4. 강력한 GUI : GeoPixel은 AWT뿐만 아니라 Third Vendor의 JavaBeans 컴포넌트를 사용하여 동적인 사용자 인터페이스를 제공한다. 모든 명령어는 자신의 고유한 다이얼로그를 통해 사용자로부터 입력 변수를 받는다. 또한 처리 대상 영상, 대상 지역, 대상 분광대 등의 선택도 모두 GUI를 통해 지원하고 있다. 나아가 사용자의 선택이 바로 처리 결과에 반영될 수 있는 실시간 영상 처리를 지원하고 있다.
5. 확장성의 극대화 : GeoPixel은 향후 시스템 기능을 사용자가 손쉽게 확장할 수 있도록 설계되었다. 우선 핵심어 탐색을 위한 명령어 기술 파일을 텍스트 형태로 제공하여 사용자가 자신만의 처리 모듈이나 핵심어를 손쉽게 추가할 수 있다. 메뉴 또한 메뉴 기술 파일을 제공하여 사용자가 자신이 원하는 구조의 메뉴를 자유롭게 설계할 수 있다. 이를 위해 모든 처리 모듈은 동적 연결(dynamic linking)을 통해 메모리에 옮겨진 뒤 실행된다.

## 제 2 절 개발된 위성영상 처리 S/W GeoPixel 모듈

### ● 기본 영상 처리

#### ◆ 영상 목록 관리

- 영상 삽입, 삭제, 정보 출력
- 영상 헤더 구조 정의 및 해석(parsing) 프로그램
- 영상 헤더 편집 기능
- 트리 구조 및 영상 폴더 개념의 영상 목록 관리 다이얼로그

- ◆ 파일 입출력 지원
  - BSQ, BIP, 일반 Graphic Format 입출력
  - LANDSAT TM, SPOT HRV, RADARSAT, IRS-1C CD-ROM 영상 직접 입력
  - 복수의 영상을 단일 영상으로의 자유로운 변환
- ◆ 파일 영상 및 메모리 영상 지원
  - 공통의 parent class 정의에 의한 인터페이스 통일
  - 파일 영상과 메모리 영상의 상호 변환
  - 영상 처리 프로그램의 최적화
- ◆ 기하학적 영상 처리
  - Flip, Rotate, Translate, Subset, Resize 연산
  - 빈 영상 작성
  - 임의의 영상 혼합 편집
- ◆ 화소형(Pixel Type) 변환
  - Byte, Unsigned Short, Signed Short, Integer, Float 화소형간 변환
- 화면 출력
  - ◆ 영상의 화면 출력
    - 흑백, RGB, 위색 칼라 영상 출력
    - Scroll, Main, Zoom 세 개의 화면을 이용한 대용량 영상 출력
    - Zoom-In, Zoom-Out, Panning
    - Cursor 종류 및 Viewport 색상 선택
    - Grid 중첩, 면적, 둘레, 길이 측정
  - ◆ 히스토그램 출력
    - 원 영상 및 향상된 영상 히스토그램 동시 출력
    - 히스토그램 분석 기능
    - Interactive 영상 향상
    - On-Line 영상 향상시 실시간 출력

◆ 색상표 출력

- 영상 색상 및 벡터 색상표
- 색상표 파일 구조 정의 및 해석 프로그램
- 슬라이드바, 캔버스, 텍스트 입력을 통한 색상 선택
- Gray, Ramped, Smooth, Random 기본 색상표.
- 색상표 파일 입출력

◆ 벡터 출력

- 벡터 파일 구조 정의 및 해석 프로그램
- DXF 파일을 내부 벡터포맷으로의 변환 프로그램
- 총 64개의 벡터 레이어(layer) 처리
- 벡터 파일 입출력

◆ AOI(Area Of Interest) 출력

- AOI 파일 구조 정의 및 해석 프로그램
- 총 64개의 AOI 레이어
- AOI 편집 기능
- 레이어별 색상 선택
- AOI 파일 입출력

◆ 스캐토그램 출력

◆ 3차원 클래스 통계 출력

◆ 밴드 분광 특성 분석

- 한 점에서의 분광대 특성 출력
- 선분에서의 분광 특성 단면 출력
- 영상 화소값의 수치 출력

● 영상 분석

◆ 단항(Unary) 연산

- 삼각함수, 로그, 지수, 정수화 함수
- 산술, 논리, 비교, 비트 연산 함수

◆ 이항(Binary) 연산

- 산술, 논리, 비교 함수

◆ 주성분 분석

◆ 색도 분석

◆ 색 좌표 변환

◆ 조직 특성 분석

- Entropy, Energy, Correlation

◆ 형상 특성 분석

- Erosion, Dilation, Thinning

◆ 물리량 변환

- NDVI 식생지수

- Tasseled Cap 식생지수

- 온도

- Radiance 값 변환

- Reflectance 값 변환

● 영상 향상

◆ On-Line 사용자 정의 영상 향상

◆ 선형 영상 향상

- level slicing

- 최대-최소 선형 영상 향상

- 사용자 정의 선형 영상 향상

- 퍼센트 선형 영상 향상

◆ 비선형 영상 향상

- 히스토그램 평탄화(equalization) 영상 향상

- Square Root, Cubic Root, Uniform 영상 향상

- Gaussian, Rayleigh, Exponential, Logarithm 영상 향상

◆ 히스토그램 일치화에 의한 영상 정규화

● 영상 필터링



- ◆ 통계치를 이용한 필터링
  - 최소값, 최대값, 범위, 역범위 평균, 중간값, 최빈수 필터링
  - Outliner, Rank, MNN, MTM, Contrast, Variance 필터링
  - Difference, Homogeneity 필터링
- ◆ 미리 정의된 커널을 이용한 필터링
  - 저주파, 고주파, Embossing, Sobel, Prewitt 필터링
  - Roberts, Laplacian, Kirsh, Frei-Chen 필터링
- ◆ 사용자 정의 커널을 이용한 필터링
- ◆ 주파수 영역 필터링
  - FFT 변환 및 역변환
  - 사용자 인터랙티브 고주파, 저주파, 밴드 패스 필터링
- 영상 보정
  - ◆ 기하 보정
    - 인터랙티브 지상 기준점(ground control point) 수집
    - 1,2,3차 다항식에 의한 기하 보정
    - 다항식을 이용한 화면 좌표 역계산에 의한 GCP 예측
    - 최근린법(Nearest Neighborhood), Bilinear, Cubic, Spline 리샘플링
    - 영상 대 영상 및 영상 대 지도 기하 보정
    - 벡터 중첩을 통한 기하 보정 검증
- 영상 분류
  - ◆ 무감독 분류법
    - K-Means 분류법
    - ISODATA 분류법
  - ◆ 감독 분류법
    - 훈련지역 선정을 위한 사용자 인터페이스
    - 훈련지역 통계치 산출
    - 최소거리분류법(minimum distance classification)

- Parallelepiped 분류법
- 최우도 분류법(maximum likelihood classification)
- Lookup 표 참조법
- ◆ 분류 정확도 검증
  - 분류오차표(error matrix) 작성
- ◆ 분류 클래스 통계의 3차원 뷰
- 응용
  - ◆ 다중 분광 밴드의 최적조합 선정
  - ◆ 임의의 시기에서의 태양 정보 계산
  - ◆ 변화 탐지
    - 밴드간 연산에 의한 변화 탐지
    - 후 분류 비교에 의한 변화 탐지
    - Change Vector 계산에 의한 변화 탐지
  - ◆ DXF 파일에서 Vertex 추출
    - Grid 자료 생성
  - ◆ VRML 파일 생성
- 사용자 작업환경 인터페이스
  - ◆ 핵심어 탐색 엔진을 통한 모듈 접근
    - 명령어 기술(command description) 파일 구조 정의 및 해석 프로그램
    - 해쉬 맵 구조를 이용한 핵심어 고속 탐색 프로그램
    - 객체 직렬화(serialization)를 이용한 핵심어 연결 구조 저장 프로그램
    - 명령어 history 기능
  - ◆ 사용자 정의 메뉴 인터페이스
    - 메뉴 기술(menu description) 파일 구조 정의 및 해석 프로그램
    - 메뉴 이벤트의 동적 연결 프로그램
    - 객체 직렬화를 이용한 사용자 메뉴 저장 프로그램

- ◆ 동적 연결(dynamic linking)을 통한 사용자 인터페이스와 모듈 연결 기술
  - 프로그램 실행 중 핵심어 및 메뉴의 이벤트 핸들러와 모듈의 연결 프로그램
  - 다중 쓰레드 환경하에서 모듈 작업 종료시 사용 자원(resource) 회수(recollecting) 프로그램

● 기타

- ◆ 다중 쓰레드를 이용한 사용자 인터럽트
- ◆ 이벤트를 통한 다이얼로그 간 통신

### 제 3 절 개발된 지형분석 S/W GeoPixel-Terrain 모듈

● 지형 모델링

- ◆ 지형 특징점 추출, 불규칙 삼각망 구성, 등고선 추출, 자료 내삽

● 지형 분석

- ◆ 경사각, 방위각, 음영도 작성
- ◆ 절토/성토량 계산, 가시권 분석, 집수역 경계 분석, 저수량 분석, 최단거리 분석

● 지형 렌더링

- ◆ HFRT에 의한 지형 렌더링
- ◆ Wireframe, Grid, TIN에서의 지형 렌더링
- ◆ 인공 구조물 설치
- ◆ 경관 시뮬레이션
  - 경로 선정
  - 3차원 실시간 경관 시뮬레이션

## 제 4 장 위성영상 처리에서 자바 언어의 이용 가능성

초창기 자바 언어는 네트워크를 손쉽게 지원하는 언어로서 그 명성을 얻었으므로 버전 1.2 이전의 JDK(Java Development Kit)는 영상을 네트워크를 통해 전달받는다는 전제하에 점진적인 영상 입력을 처리할 수 있는 API(Application Programming Interface)를 갖고 있었다. 따라서 대표적인 영상 관련 클래스인 `java.awt.Image`, `java.awt.ImageProducer`, `java.awt.ImageConsumer`의 인터페이스는 X-Window나 Windows API와 달리 특이한 구조를 가지고 있어 영상 처리 소프트웨어 개발에 어려움이 따랐다(Chan, 1997). 또한 초기 JDK는 단순한 형태의 영상 처리를 인터프리터 방식으로만 실행하였으므로 실시간 영상 처리에는 부적합한 프로그램 언어로서 자바를 인식하게 만들었다. 따라서 이제까지 자바 언어를 영상 처리에 활용한 분야는 네트워크를 통한 영상 정보 서비스에서 단순한 처리 효과를 얻는데 그치고 있었던 것이 사실이다.

그러나 자바 언어가 네트워크 서비스이외의 분야에서도 널리 사용되면서 보편적인 프로그램 언어로서의 사용자 요구가 커지자, 이에 부응하기 위해 최근 자바 표준 라이브러리에 대한 대대적인 개선 작업이 시도되고 있다. 그 결과 처리 속도, 사용자 인터페이스의 다양성과 유연성, 안정성 모두에서 괄목할만한 성과를 거두고 있다[2]. 또한 네트워크 서비스에서 시각 정보가 차지하는 비중이 날로 커짐에 따라 다양한 영상 정보 처리를 지원하기 위한 API들이 새로 개발되고 있다(Sun Microsystems, 1998b, 1998c, 1998d).

이 같은 시점에서 본 연구는 위성 영상 처리 분야에서 자바 언어가 가질 수 있는 가능성에 대하여 새롭게 조명하고자 한다. 이를 위해 영상 처리 소프트웨어의 시장성을 결정하는 중요한 조건인 처리속도와 안정성, 병렬 처리라는 관점에서 자바 언어를 현재 널리 사용하고 있는 C++ 언어와 비교, 설명할 것이다.

## 제 1 절 처리 속도와 안정성

영상 처리에 있어 자바의 실행 속도 향상에 극적인 변화를 가져온 것은 JDK ver.1.1.6부터 Sun Microsystems사가 Symantec사에서 라이선스를 받아 제공하는 JIT(Just-In-Time) 컴파일러이다. JIT 컴파일러는 중간 코드 형태인 바이트 코드를 메서드 실행 전에 기계어 코드로 바꾸므로 반복해서 실행하는 코드를 빠르게 처리할 수 있다(Carmine, 1998).

본 장에서는 JDK의 가장 최신판인 버전 1.2RC2가 제공하는 JIT의 배열 처리 속도와 최적화된 C++의 실행 코드가 가지는 속도를 비교하여 자바의 영상 처리 소프트웨어 응용 가능성을 점검하고자 한다. 최적화된 실행속도를 비교하기 위해 C++ 코드는 Visual C++ ver.5.0에서 디버깅 정보 없이 maximize speed로 최적화하였다. 자바 언어와 비슷한 기능을 지원하기 위해 RTTI(Run-Time Type Information), multithread, exception-handling, inline function을 모두 활성화하였다. 자바 코드는JDK ver.1.2 RC2에서 디버그 정보 없이 바이트코드로 컴파일한 뒤 JIT를 사용하여 실행하였다.

### 1. 배열 연산 속도

영상 처리 속도에 가장 직접적인 영향을 미치는 요인은 배열에 대한 연산 속도이다. 대용량의 영상을 처리하는 소프트웨어가 만족시켜야 할 가장 중요한 사용자 요구가 빠른 처리 속도임을 감안할 때 배열 연산 속도는 소프트웨어의 시장성 여부를 결정짓는다고 해도 과언이 아닐 것이다. 따라서 본 연구는 우선 자바 언어와 C++를 사용한 프로그램의 배열 연산 속도를 먼저 비교하였다.

영상 처리에서 자주 사용하는 연산의 속도를 비교하기 위해 Code-1, Code-2와 같은 자바 프로그램의 실행 속도를 측정하였다. 이 코드들은 삼차원 배열의 접근 시간, 곱연산 시간, 인덱스 루핑 시간을 합친 실행 소요 시간을 가진다. 속도 비교에 사용된 C++ 코드는 똑같은 루프 구조를 가지며 Code-1에서는 32bit 정수 배열을, Code-2에서는 unsigned char 배열을 사용하였다.

Code-1: 정수형 배열의 연산 속도

```
for (int i=0; i<100; I++)  
    for(int j=0; j<100; j++)  
        for(int k=0; k<100; k++)  
            a[i][j][k] *= a[i][j][k]; //int
```

Code -2: 바이트형(unsigned 8bit) 배열의 연산 속도

```
for (int i=0; i<100; I++)  
    for(int j=0; j<100; j++)  
        for(int k=0; k<100; k++) {  
            v = a[i][j][k] & 0xff;  
            a[i][j][k] = (byte) v * v;//byte  
        }  
    }
```

실행 소요 시간을 측정한 결과는 Table 4-1과 같다.

Table 4-1 Computation time for array operation

	JDK 1.2 RC2	Visual C++ ver. 5.0
Code-1의 실행 소요 시간	6.26 sec	6.38 sec
Code-2의 실행 소요 시간	2.47 sec	3.26 sec

이 결과는 JIT를 사용한 자바 프로그램의 배열 연산 속도가 오히려 C++보다 더 빠르다는 사실을 보여준다. 이는 자바 언어의 배열 연산에 대한 일반적인 생각과 배치되는 결과이다. 자바 언어는 배열에 대한 잘못된 인덱싱이 초래할 시스템 오류를 방지하기 위해 배열 접근시 반드시 배열 경계를 벗어나는지 여부를 점검한다. 따라서 경계조건 점검에 따른 부담으로 인해 배열 연산 속도가 느려질 것이라는 것이 일반적인 견해이다(Mark, 1998).

그러나 JDK 1.2RC2에서 제공하는 JIT의 경우 루프 안에서 이루어진 배열 접근을 최적화하여 오히려 Visual C++ ver. 5.0에서 작성한 C++프로그램보다 더 빠른 연산 속도를 보이고 있다. 따라서 배열 연산 속도가 전체의 성능을 크게 좌우하는 영상 처

리에서는 자바 언어가 C++ 프로그램 못지않은 성능을 가질 것으로 기대된다. 또한 자바 언어는 배열 경계 점검을 통해 C++에서 배열 접근시 종종 발생하는 시스템 오류를 사전에 방지할 수 있으므로 소프트웨어 생산성 입장에서도 우위를 차지한다.

## 2. 배열 생성과 파괴 속도

영상 처리에서 배열의 연산 속도와 더불어 성능에 영향을 미칠 수 있는 요인으로는 배열의 생성과 파괴 속도이다. 이를 비교하기 위해 Code-3와 Code-4 프로그램의 실행 소요 시간을 측정하였다. 컴파일과 실행 조건은 앞에서 기술한 것과 같으며 단지 자바 코드의 실행시 최대 힙 크기를 64MByte로 지정하였다.

Code-3: Java 프로그램  

```
for (int i=0; i<10000; i++)
    a = new byte[1000000];
```

Code-4: C++ 프로그램  

```
for(int i=0; i<10000; i++) {
    a = (unsigned char*)calloc(1, 1000000);
    free(a); }
```

실행 소요 시간을 측정한 결과는 Table 4-2와 같다.

Table 4-2 Computation time for array construction and destruction

JDK 1.2 RC2	Visual C++ ver. 5.0
98.31 sec	1.70 sec

이 결과에서 알 수 있듯이 자바 언어는 객체의 생성과 파괴에서 치명적인 약점을 가지고 있다. C++ 언어는 객체의 파괴를 사용자에게 일임하는데 반해 자바 언어는 쓰레기 수거기(*garbage collector*)의 비동기적인 개입에 의존하고 있다. 따라서 별도의 쓰레드로 실행되는 쓰레기 수거기가 수거해야 할 객체를 탐색하는 데 많은 시간을 소모한다(Carmine, 1998).

대용량 영상을 처리하는 프로그램에서 배열의 생성과 파괴는 배열 연산만큼 자주 이루어지는 작업이 아니므로 위와 같은 성능 저하는 대부분의 영상 처리 프로그램에

서 큰 부담으로 작용하지 않을 것이다. 그러나 많은 배열을 생성했다가 버리는 구조로 설계된 영상 처리 프로그램에서는 성능을 떨어뜨리는 주요 원인이 될 수 있다. 따라서 자바 언어를 사용한 영상 처리 프로그램은 객체의 쓰레기화를 최대한 억제하는 구조로 개발되어야 하며 이를 위해서는 사용한 배열을 재사용하는 object recycling 기작을 고안해야 할 것이다.

한편 쓰레기 수거기는 소프트웨어의 치명적인 버그일 수 있는 메모리 누수를 효과적으로 막을 수 있는 기작이므로 소프트웨어의 안정성에 기여하는 바가 크다 하겠다. 객체의 생성과 파괴 속도를 높이기 위해 Sun Microsystems사는 HotSpot 프로젝트를 통해 각각의 쓰레드에게 스택의 일종인 힙 캐쉬를 사용하는 방법을 고안하고 있다[8]. 이 방법을 JIT가 지원한다면 객체의 생성과 파괴가 빈번한 프로그램에서는 상당한 속도 증가가 기대된다.

### 3. 배열 처리와 관련된 다른 특징들

이외에도 C++에서는 포인터 조작을 이용한 배열 접근이 자유로운 반면에 자바에서는 불가능하다. 포인터에 대한 잘못된 조작으로 시스템 오류는 물론 종종 대단히 찾기 어려운 버그를 만들어서 안정성과 생산성에 치명적인 문제를 일으키는 것을 사전에 피하기 위해서 자바 언어는 배열 접근을 엄격히 규제하는 것이다. 그러나 경우에 따라서는 이 같은 엄격함이 생성된 배열의 일부분을 가져오기 위해 그 내용을 다시 복제해야 하는 부담을 안겨 준다. 이를 해결하기 위해서 배열을 다루는 API는 대부분 오프셋과 길이를 넘겨 받아 배열의 일부분을 처리할 수 있도록 설계하지만 C++의 다양한 포인터 조작을 대신하기 위해서는 신중한 인터페이스 설계가 필요한 것이 사실이다.

한편 Code-2 프로그램은 자료형과 관련된 자바 언어의 미묘한 문제를 보여주고 있다. 많은 영상들이 unsigned 8bit형으로 저장되는 반면 자바 언어는 signed 8bit형만 지원한다(Gosling, 1996). 그 결과 unsigned 8bit형인 화소값을 연산하기 위해서는 추가적인 비트 연산을 실시해야 한다. 이는 unsigned 16bit형의 화소인 경우에도 마찬가지로 적용된다. 이 때문에 프로그램 코드는 복잡한 모습을 가지지만 비트 연산 자체가



부하가 많지 않은 연산이고 C++에서도 대부분 연산 전에 정수형 이하의 모든 형을 정수형으로 변환하므로 Table 4-1과 같이 처리 속도에는 별다른 영향을 미치지 않는다.

배열 처리와 관련하여 버전 1.2부터 추가된 기능 중에 중요한 것은 JNI(Java Native Method)의 변화이다. JNI는 자바 프로그램과 C 프로그램을 연결하는 인터페이스로서, 버전 1.2이전에는 할당된 메모리를 특정 주소에 고정시키는 pinning을 지원하지 않는 가상 머신에서 대용량의 배열을 C 프로그램과 교류할 때 똑같은 배열을 새로 복제해야 하는 불편함이 따랐다. 이 같은 불필요한 복제는 C 코드를 일부 사용하면서 대용량의 배열을 처리하는 자바 프로그램의 성능을 떨어뜨리는 요인이었다. 그러나 버전 1.2에서는 pinning을 제공하지 않은 가상 머신에서도 별도의 메서드를 통해 대용량의 배열을 주고 받을 수 있게 배려하여 JNI를 사용한 자바와 C++의 효과적인 배열 교환을 지원하고 있다(Sun Microsystems, 1998e).

## 제 2 절 병렬 처리

여기서는 대용량의 영상을 병렬처리로 신속하게 처리하는 분야에서 자바가 기여할 수 있는 가능성을 살펴보고자 한다. 다중 프로세서 시스템이 보편화되면서 다중 쓰레드를 이용한 병렬 처리 소프트웨어에 대한 요구가 커지고 있으나 C, C++가 같은 언어에서는 다중 쓰레드를 다루는 명령어가 표준화되어 있지 않아 시스템마다 별도의 라이브러리를 사용하여 다른 코드로 구현해야 하는 문제점을 갖고 있다. 이에 반해 자바 언어는 언어 자체를 다중 쓰레드 개념 위에 정의하고 있으며 쓰레드 객체에 대한 제어어를 통해 쓰레드를 제어할 수 있도록 디자인되어 있어, 보다 쉽게 병렬 처리 소프트웨어를 구현할 수 있는 장점을 가지고 있다(Lea, 1997).

## 1. MTDIP 모형

본 연구는 자바 언어의 다중 스레드 기능을 살려 위성 영상 처리를 위한 표준 모형인 MTDIP(MultiThreaded and Data-driven Image Processing)를 개발하였다[12]. MTDIP 모형의 기본 구조는 Fig. 4-1과 같다.

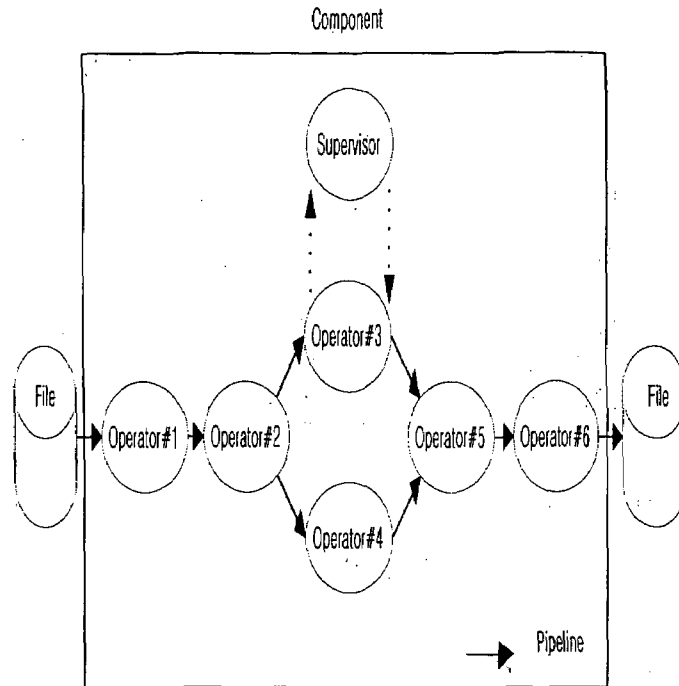


Figure 4-1 Basic structure of MTDIP Model

하나의 영상 처리 작업은 하나의 감독자(supervisor)와 여러 개의 작업자(operator)의 협동에 의해 이루어지며 감독자와 작업자들은 각각 동시에 실행되는 별도의 스레드로 만들어진다. 작업자들간의 동기화와 자료 교환은 Fig. 4-2와 같은 구조의 객체 파이프라인(object pipeline)을 통해 이루어진다. 대부분의 영상 처리에서 대상 영상은 다수의 패킷으로 분해되어 객체 파이프라인을 따라 이동하는데 이 때 객체 파이프라인은 레퍼런스 복사를 사용하여 패킷을 다시 복제하는 부담을 피하고 있다.

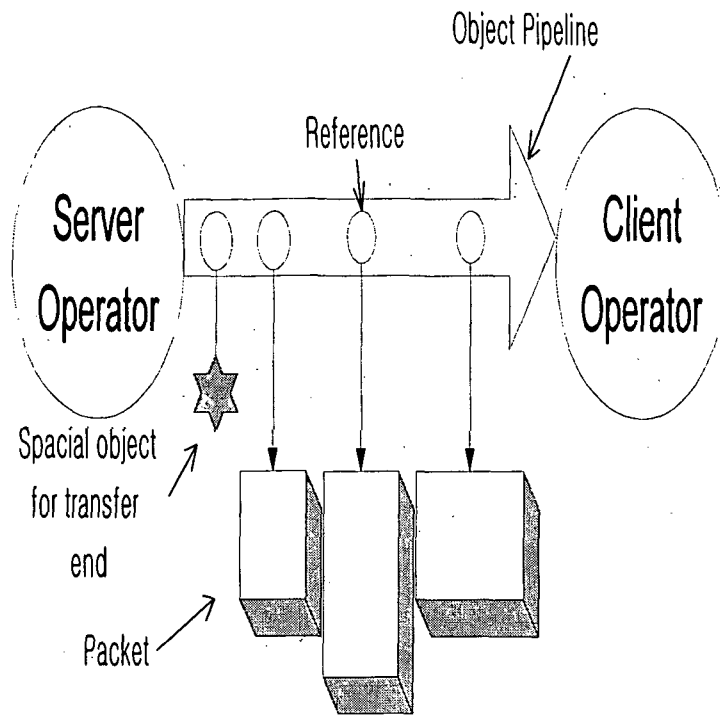


Figure 4-2 Data exchange and synchronization between operators through object pipeline

Sobel 필터를 구현한 MTDIP 모형 기반의 프로그램은 Fig. 4-3과 같다. 위성 영상을 분광대별도 나눈 후 서로 다른 이동 경로를 따라 처리한 뒤 Collector에서 합친 후 파일에 저장한다.

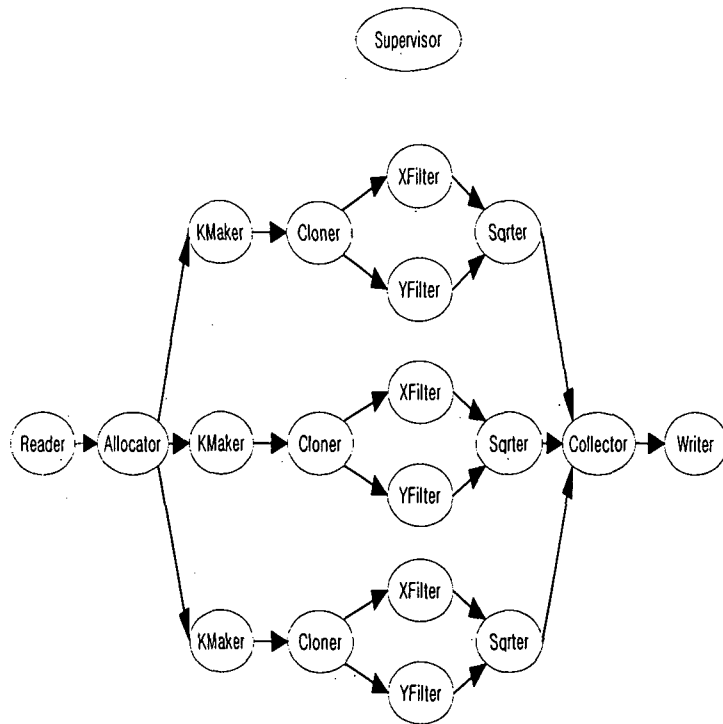


Figure 4-3 Program structure for Sobel filtering based on MTDIP model

이 프로그램에 대한 처리 속도를 Table 4-3과 같은 영상에 대하여 측정한 결과는 Fig 4-4과 같다.

Table 4-3 Images for measurement of parallel processing speed

Image ID	i512	i724	i1024	i2048	i4096	i8192
Format	BIL	BIL	BIL	BIL	BIL	BIL
Pixel Type	Byte	Byte	Byte	Byte	Byte	Byte
Samples	512	724	1024	2048	4096	8192
Lines	512	724	1024	2048	4096	8192
Bands	3	3	3	3	3	3

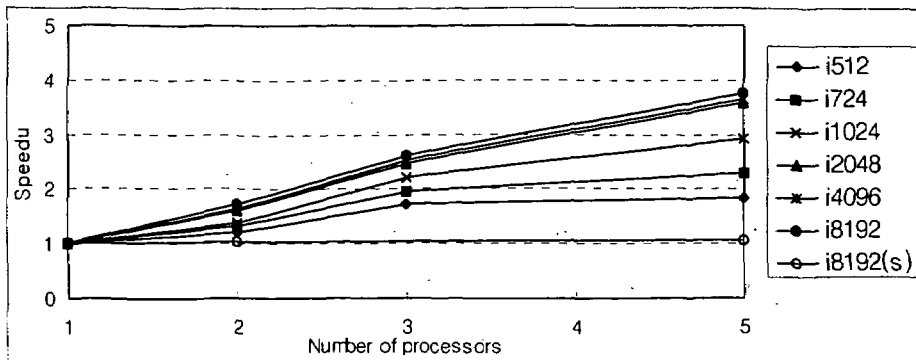


Figure 4-4 Comparison of speedup between MTDIP-based and sequential Sobel filtering.

Speedup is the ratio of the computing speed of  $n$  processors to that of 1 processor. AQUANTA HS/6 of UNISYS, 128MB RAM, 512KB cache, Window NT 4.0 Server OS, Symantec Java! ByteCode Compiler ver.210.065 are used. i8192(s) means the speedup of sequential processing on i8912 image.

이 결과에서 주목해야 할 점은 단일 스레드를 사용한 프로그램의 처리 속도는 프로세서 수에 따라 별다른 변화를 보이고 있지 않는 반면 MTDIP 모형을 사용한 프로그램의 속도는 프로세서의 수에 비례해서 증가한다는 사실이다. 특히 영상의 크기가 클수록 높은 speedup을 보이고 있는 데 이는 패킷의 크기가 커짐에 따라 작업자간의 동기화와 자료 교환에 소모하는 시간이 작업자가 패킷을 가공하는데 소모하는 시간에 비해 줄어들기 때문으로 사료된다. 이와 같은 구조의 위성 영상 처리 소프트웨어는 지상 관제소와 같이 특정 위성의 관측 영상을 같은 방법으로 반복해서 실시간 처리할 경우에 매우 유용할 것이다.

## 2. 향상된 다중 스레드 기능

자바 언어는 JDK ver.1.2 이후부터 다중 스레드를 사용한 위성 영상 처리에 중요한 의미를 가지는 몇 가지 기능을 수정 또는 보완하였다. 이전 버전에서는 Windows 시스템만 OS 고유의 스레드를 자바에서 지원하였고 Solaris 시스템은 green thread라 하여 일종의 시분할법을 이용한 에뮬레이션만 지원하였다. 그러나 1.2 버전부터는 Solaris 시스템에서도 JIT와 더불어 시스템 고유의 스레드를 자바 프로그램이 사용할

수 있어 대형의 UNIX 시스템에서 위성 영상을 고속 처리할 수 있는 길을 마련해 놓았다(Sun Microsystems, 1998f).

또한 JDK ver.1.2는 쓰레드의 동작을 보다 안정적으로 관리할 수 있도록 개선되었다. 과거 버전에서는 interrupt 기능이 동작하지 않았으므로 stop(), suspend(), resume()과 같은 메서드의 사용을 허용하였다. 그러나 이 메서드들은 때에 따라서 쓰레드의 상태를 불안정하게 만들 수 있으므로 자바로 만들어진 다중 쓰레드 프로그램이 제대로 작동하지 않는 문제를 발생하곤 하였다. 1.2버전에서는 이 모든 메서드를 없애고 interrupt 기능을 새롭게 지원하여 쓰레드를 보다 안정된 상태에서 관리할 수 있도록 개선하였다(Sun Microsystems, 1997).

다중 쓰레드 프로그램에서 실행 속도에 영향을 미치는 주요 요인 중에 하나는 쓰레드 모니터링과 동기화된 메서드 접근 속도인데 JDK ver. 1.2에서는 이를 개선하여 성능 향상을 꾀하였다.

그리고 버전 1.2부터는 쓰레기 수거기의 동작을 보다 섬세하게 제어하기 위해서 Reference Object라는 개념을 새롭게 도입했다[15]. 이 중 특히 Soft Reference는 다중 쓰레드를 이용한 위성 영상 처리의 메모리 관리에 크게 기여할 수 있을 것이다. MTDIP 모형에서 대상 영상은 여러 개의 똑같은 구조를 가진 패킷으로 분리되어 객체 파이프라인을 따라 이동하는데, 파일 출력을 마친 패킷을 그대로 버린다면 대량의 패킷들이 버려져서 쓰레기 수거기의 잦은 개입으로 소프트웨어 성능이 떨어질 수 밖에 없다.

만약 사용을 마친 패킷을 Soft Reference를 사용해 보관한다면 쓰레기 수거기는 여분의 메모리가 필요할 때 Soft Reference로 연결된 객체를 탐색하여 가장 오래된 객체부터 수거해 나갈 것이다. 따라서 가장 빈번하게 사용하는 패킷을 재사용할 수 있는 확률이 높아져 처리 속도를 높일 수 있다.

여 백

## 제 5 장 GeoPixel: 위성영상 통합 처리 소프트웨어

본 장에서는 개발한 위성 영상 통합 처리 소프트웨어인 GeoPixel에 대해서 기술하고자 한다. GeoPixel은 현재 1.0 버전을 가지고 있으며 세계 최초로 100% 순수 자바로 구현된 위성 영상 처리 소프트웨어이다. 또한 객체 지향적 설계를 토대로 개발되어 향후 시스템 기능 확장이 용이하며 수백 MBytes 이상의 영상도 원활하게 처리할 수 있는 구조를 가지고 있다. 아울러 그래픽 기반의 동적인 사용자 인터페이스와 함께 핵심어 탐색 엔진, 프로그래머블 메뉴와 같은 편리한 기능도 제공하고 있다.

### 제 1 절 자료 구조

GeoPixel은 다양한 위성 영상 처리를 하나의 통합된 환경에서 지원하기 위해서 영상뿐만 아니라 벡터, 색상표, 배열과 같은 다양한 자료의 구조를 정의하고 이를 다룰 수 있는 클래스를 가지고 있다.

#### 1. 문자형 파일의 기본 구조

GeoPixel의 문자 자료는 대부분 다음과 같은 형태의 키와 이에 해당하는 값의 조합으로 구성된다. 키의 대소문자 구별은 없으며 등호 양옆의 빈칸은 사용자가 임의로 줄 수 있다.

Key1 = Value1

Key2 = Value2

키값은 단어(word), 문자열(string), 숫자(numeric), 문자열 배열(string array), 숫자 배열(numeric array) 등의 속성을 가진다. 각각의 속성에 따라 다음과 같은 형태로 기록된다.



단어

```
Key = WORD
```

문자열

```
Key = "This is an example"
```

숫자

```
Key = 1.234
```

문자열 배열

```
Key = {"TM Band 1", "TM Band 2", "TM Band 3"}
```

숫자 배열

```
Key = {12.3 , 13.2 , 16E2}
```

코멘트는 행의 첫 열에서 #로 시작한다.

## 2. 속성 파싱 클래스

위와 같은 구조의 문자 자료를 파싱하는 클래스는 AttributeParser이다. 이 클래스의 메서드 구조와 해당하는 키값 속성은 다음과 같다.

```
public String nextKey();//다음 키를 읽는다.  
public String valueOfWord();//문자형 키값을 읽는다.  
public double valueOfNumeric();//숫자형 키값을 읽는다.  
public String valueOfString();//문자열형 키값을 읽는다.  
public Vector valueOfStringArray(Vector v);//문자열 배열형 키값을 읽는다.  
public Vector valueOfNumericArray(Vector v);//숫자 배열형 키값을 읽는다.
```

### 3. 영상(Image)

하나의 영상은 영상 헤더(header)와 영상 바디(body)의 두 개의 자료로 구성된다. 영상 헤더 파일은 chd 확장자를 가진다. 영상 헤더의 자료 구조에 대한 예는 다음과 같다.

```
#GeoPixel Cube Header
```

Title = "This is title"	제목
Samples = 1000	열수
Lines = 1000	행수
Bands = 7	분광대수
Pixel Type = Byte	화소형
Xmin = 0.0	지도좌표의 X 최소값
Xmax = 1.0	지도좌표의 X 최대값
Ymin = 0.0	지도좌표의 Y 최소값
Ymax = 1.0	지도좌표의 Y 최대값
Zmin = {20 , 10 , 9999 , 9999 , 9999 , 9999 , 9999}	최소 화소값
Zmax = {50 , 100 , -9999 , -9999 , -9999 , -9999 , -9999}	최대 화소값
Bands Name = { "Band 1" , "" , "" , "" , "" , "" , "" }	분광대명

이 때 Zmin, Zmax가 동시에 9999와 -9999 값을 동시에 가지면 화소값의 최대 최소값이 계산되지 않은 경우다. 화소형은 unsigned 8bits(Byte), signed 16bits(Short), unsigned 16bits(USHort), signed 32bits(Integer), 32bits float(Float)를 지원한다. 영상 헤더에 대한 관리는 CubeHeader 클래스가 담당한다(API에 대한 정보는 부록 참조).

영상 바디는 대용량의 위성 영상을 효율적으로 다루기 위해 BIL(Band Interleaved by Line) 양식으로 저장되며 cub 확장자를 가진다. 영상에 대한 관리는 Target 클래스가 담당한다.

Target 클래스는 두 가지 종류의 클래스로 세분된다. Target.File은 파일에 저장된

영상을 다룰 때 사용하는 클래스로서 Pixel 변수는 java.io.File형을 가진다. 반면에 Target.Memory 클래스는 메모리에 저장된 영상을 다룰 때 사용하는 클래스로서 Pixel 변수는 java.lang.Object형으로 byte[], short[], int[], float[] 네 가지 형 중 하나의 배열을 가진다. 배열의 첫 인덱스는 분광대를, 두 번째 인덱스는 한 분광대의 영상을 일차원 배열로 가지고 있다. 따라서 i번째 분광대 영상의 j번째 행과 k번째 열의 바이트형 화소값은 다음과 같다.

$$((byte'[][])\text{Pixel}).i][j*\text{samples}+k]$$

Target.Memory는 위성 영상의 일부분을 메모리에 올려놓고 신속하게 처리 결과를 확인하는데 사용할 수 있다. 따라서 사용자는 파일에 저장된 전체 영상에 대한 처리를 수행하기 전에 자신의 작업이 생성할 결과를 대략적으로 살펴볼 수 있다. GeoPixel의 모든 모듈은 다섯 가지 화소형과 두 가지 영상형을 모두 지원할 수 있도록 개발되었다.

#### 4. 벡터(Vector)

GeoPixel의 벡터는 신속한 벡터 처리를 위해 이진형을 가진다. 벡터 자료의 구조는 다음과 같다.

```

title : UTF String
x of minimum limit : double
y of minimum limit : double
x of maximum limit : double
y of maximum limit : double
number of features : int
type of 1st feature : int
id of 1st feature : int

```

min x of 1st feature : double  
min y of 1st feature : double  
max x of 1st feature : double  
max y of 1st feature : double  
number of vertex : int  
x of 1st vertex : double  
y of 1st vertex : double  
x of 2nd vertex : double  
y of 2nd vertex : double  
...  
type of 2nd feature : int  
id of 2nd feature : int  
min x of 2nd feature : double  
min y of 2nd feature : double  
max x of 2nd feature : double  
max y of 2nd feature : double  
number of vertexes : int  
x of 1st vertex : double  
y of 1st vertex : double  
x of 2nd vertex : double  
y of 2nd vertex : double  
...

각각의 feature는 고유의 id를 가지고 있는데 DXF 파일에서 벡터 자료를 읽어들이  
경우 고도나 color 혹은 layer order 등이 이 id에 할당된다. 벡터 자료에 대한 관리는  
GeoVector 클래스가 맡고 있다. 이 클래스에 대한 자세한 사양은 부록을 참조하기 바  
란다.

## 5. 행렬(Matrix)

행렬 자료는 이차원 숫자 행렬을 나타내며 각종 계산 결과와 커널을 저장한다. 이 자료는 `mtx` 확장자를 가지며 이에 대한 관리는 `Matrix` 클래스가 맡고 있다. 이 클래스에 대한 자세한 사양은 부록을 참조하기 바란다. 행렬 자료 구조에 대한 예는 다음과 같다.

```
#GeoPixel Matrix
```

```
Title = ""           제목
Columns = 4          열수
Rows = 4             행수
Matrix = {84.0 , 38.0 , 165790.0 , 440560.0 ,   행렬값
411.0 , 1070.0 , 169356.0 , 409919.0 ,
2086.0 , 1073.0 , 216317.0 , 400983.0 ,
967.0 , 1654.0 , 181927.0 , 390511.0
}
```

## 6. 색상표(Palette)

색상표 자료는 화소값에 색상을 대응시키는 역할을 담당한다. 이 자료는 `pal` 확장자를 가지고 이에 대한 관리는 `Palette` 클래스가 맡고 있다. 이 클래스에 대한 자세한 사양은 부록을 참조하기 바란다. 색상표 자료 구조에 대한 예는 다음과 같다.

```
#GeoPixel Palette
```

```
Title="부드럽게 변하는 색상표"   제목
Columns = 3                      red, green, blue 값이므로 열수는 항상 3임
Rows = 256                       모두 256개의 색상을 정의하므로 행수는 항상 256임
Matrix = {                        색상값(R,G,B)
```

```

0,0,0,                화소값 0에 해당하는 색상
0,0,255,
0,8,255,
0,16,255,
....
255,160,255,
255,255,255          화소값 255에 해당하는 색상
}

```

## 7. 통계(Statistics)

통계 자료는 영상에 대한 통계 결과를 일관되게 다루기 위해 고안된 것으로 sta 확장자를 가진다. 이에 대한 관리는 Statistics 클래스가 맡고 있다. 이 클래스에 대한 자세한 사양은 부록을 참조하기 바란다. 통계 자료 구조에 대한 예는 다음과 같다.

```

#GeoPixel Statistics
Title = "Statistic of TM9305.cub"
No of Bands = 5
No of Classes = 5
No of Pixels = {570 , 180 , 157 , 2489 , 331}
Min = {
72.0 , 32.0 , 27.0 , 76.0 , 67.0 ,
...
134.0 , 68.0 , 91.0 , 83.0 , 153.0
}
Range = {
10.0 , 8.0 , 12.0 , 81.0 , 39.0 ,
...

```

}

Mean = {

76.94035087719298 , 34.719298245614034 , 30.66140350877193 , 119.31228070175439 ,  
82.57894736842105 ,

...

}

Var = {

2.740301631270995 , 2.8194521391199032 , 2.1502646968298222 , 211.75511234225814 ,  
94.96657433056316 ,

...

}

Covar = {

2.740301631270995 , 1.356940597107041 , 1.3078731917512414 , 9.294065866421079 ,  
7.495937211449018 ,

...

}

Correl = {

1.0000000000000002 , 0.48817921864704394 , 0.5387911963613757 , 0.38582432873296985  
, 0.46466650650343627 ,

...

}

Sum X = {

43856.0 , 19790.0 , 17477.0 , 68008.0 , 47070.0 ,

...

}

Sum XY = {

3375858.0 , 1523423.0 , 1345432.0 , 5237857.0 , 3625855.0 ,

...

}

## 8. AOI(Area of Interest)

AOI 자료는 특정 영역을 영상 좌표로 저장할 때 사용하는 것으로 AOI 클래스가 관리한다. AOI 클래스는 다시 AOI.Bit 클래스와 AOI.Graph 클래스로 나뉘는데 AOI.Bit 클래스는 영역을 비트 배열로 저장하고 AOI.Graph 클래스는 영역을 정수형 지도 좌표로 저장한다. AOI.Bit는 저장 용량을 줄이기 위해 비트 배열을 RLC(Run Length Code) 방식에 의해 저장한다. AOI.Graph 클래스는 정수형 영상 좌표를 이용해서 비트 배열의 자료를 생성해주는 메서드를 제공하여 AOI.Bit 클래스와의 호환을 지원한다. 이 클래스에 대한 자세한 내역은 부록을 참조하기 바란다.

AOI.Bit형 자료는 다음과 같은 사양을 따른다.

```
AOI type : int, always 2
number of samples : int
number of lines : int
1st code value : short
2nd code value : short
/....
// last code value : short
```

AOI.Graph형 자료는 다음과 같은 사양을 따른다.

```
AOI type : int, always 1
Title : UTF String
number of polygon : int
number of vertex : int
```



x of 1st vertex : int  
y of 1st vertex : int  
x of 2nd vertex : int  
y of 2nd vertex : int  
...  
number of vertex : int  
x of 1st vertex : int  
y of 1st vertex : int  
x of 2nd vertex : int  
y of 2nd vertex : int

## 제 2 절 핵심 동작 원리

본 절에서 GeoPixel이 하나의 영상 처리 작업을 수행하는 핵심 기작을 자세히 설명하고자 한다.

GeoPixel ver.1.0에서 모든 실행 모듈은 동적 연결(dynamic link)에 의해 메모리로 옮겨진 후 실행된다. 사용자가 원하는 작업을 메뉴나 명령어를 통해 선택한 후에야 비로소 해당 모듈을 메모리로 읽는다. 따라서 프로그램 초기 실행시 초기 동작에만 필요한 클래스만 메모리로 옮기므로 초기 동작에 소요되는 시간이 짧은 장점을 가지고 있다. 또한 프로그램 컴파일 때 제공하지 않았던 클래스도 단순한 텍스트 편집만으로 시스템에 통합할 수 있다.

실행 모듈의 동적 연결을 위한 프로그램 코드는 다음과 같다.

```
public void commandSelected(CommandEvent event) {  
    Object obj = event.getSource();  
    try {
```

```

        if(event.getCommand().getID().equalsIgnoreCase("QUIT")) {
            quit();
        }else {
            Class c = Class.forName(event.getCommand().getID());
            Supervisor spv = (Supervisor)c.newInstance();
            spv.setParentFrame((Frame)obj);
            spv.setDescription(event.getCommand().getDescription());
            spv.execute();
        }
    }catch(Exception e) {
        e.printStackTrace();
    }
}
}

```

선택한 메뉴나 명령어와 연결된 실행 모듈의 이름은 CommandEvent의 getID() 메서드를 통해 확인할 수 있다. 각각의 실행 모듈은 반드시 Supervisor 클래스를 상속받고 처리 작업은 execute()의 호출에서 시작하도록 설계되어 있다. 그러므로 일단 실행 모듈의 이름이 확인되면 Class.forName() 메서드에 의해 Class형으로 파일에서 메모리로 읽어 들인 다음 Supervisor형으로 캐스팅한 후 execute()를 호출한다.

앞의 프로그램에서 확인할 수 있듯이 하나의 영상 처리 작업을 전체적으로 관리하는 클래스는 Supervisor에서 상속받은 감독자다. 이 감독자는 Operator형을 상속받은 여러 개의 작업자를 사용하여 자신의 처리 작업을 완수한다.

감독자가 맡은 임무 중 가장 중요한 일은 필요한 작업자의 생성과 파괴이다. 하나의 영상 처리 작업을 수행할 경우 대부분 세분화된 1개 이상의 일련의 작업을 통해 그 결과를 얻을 수 있다. 이와 같이 세분화되어 잘 정의될 수 있는(well-defined) 처리 작업을 담당하는 것이 작업자이다. 감독자는 각 단계에서 필요한 작업자를 생성하고 실행하며 그 결과를 다음에 생성한 작업자에게 넘겨주는 역할을 수행한다. 또한 이미 작업을 마친 작업자는 메모리 확보를 위해 레퍼런스 연결을 끊는다.

감독자가 수행하는 또 다른 임무는 사용자 인터페이스를 통해 영상 처리에 필요한 입력 매개변수 값을 사용자로부터 받아 이를 각 작업자에게 넘겨주는 것이다. 이를 위해 감독자는 작업자를 생성하기 전에 필요한 사용자 인터페이스를 화면에 표출하여 사용자의 입력을 기다린다. 사용자가 입력 영상이나 출력 영상과 같은 필요한 매개변수 값을 지정하고 확인 버튼을 누르면 감독자는 이를 DialogActionEvent를 통해 넘겨받는다. 넘겨받은 매개변수 값을 근거로 필요한 작업자를 생성하고 작업자의 매개변수 값을 지정한다.

감독자의 또 다른 중요한 임무는 작업 진행 상황을 알려주는 ProgressStream형 객체를 생성하고 화면에 표출한 후 이를 생성한 작업자에게 지정해주는 것이다. ProgressStream은 작업 진행을 알려주는 클래스로서 주로 DefaultProgressStream과 GeoProgressDialog형으로 주로 사용한다. DefaultProgressStream은 진행상황을 System.out에 숫자로써 나타내므로 디버거 모드에서 많이 사용된다. GeoProgressDialog는 ProgressBar를 가진 다이얼로그로서 사용자는 작업 진행 중에 다이얼로그를 닫아 중단할 수 있다.

일련의 작업자들은 작업 도중 하나의 ProgressBar를 갱신한다. 각각의 작업자는 자신이 작업을 마칠 때까지 ProgressBar를 갱신하는 횟수를 미리 정해놓고 있는데 대부분 입력 영상의 라인 수를 사용한다. 따라서 일련의 작업자들이 보고하는 전체 횟수를 ProgressStream의 최대 갱신 횟수로 지정해야 모든 영상 처리 작업을 끝마쳤을 때 100% 진척도를 표시할 수 있다. 감독자는 이를 위해 대상 영상에 대한 작업자의 갱신 횟수를 미리 살펴본 후 총 횟수를 ProgressStream의 최대 갱신 횟수로 지정한다.

자바 언어는 별도의 하드웨어 인터럽터 기능을 제공하지 않으므로 영상 처리 작업 도중 사용자가 GeoProgressDialog형 다이얼로그를 닫을 때 처리 작업을 멈추게 만들기 위해서는 쓰레드에 대한 조심스러운 설계가 필요하다. 감독자를 별도의 쓰레드로 구현하지 않은 경우 작업자를 실행하는 쓰레드는 사용자 인터페이스의 확인 버튼이 발생한 이벤트를 처리한 AWT Event-Queue Thread이다. 이 쓰레드는 이벤트 큐에 저장된 이벤트를 꺼내어 해당 핸들러에게 넘겨주는 역할을 담당하고 있다. 따라서 작업자가 자신의 작업을 수행하고 있는 동안 이 쓰레드는 처리 작업에 참가하고 있기 때문에 고유의 임무인 이벤트 처리를 할 수 없다. 그런데 사용자가 작업 도중 중단하

고 싶어 GeoProgressDialog형 다이얼로그를 닫을 때 이를 처리해야 하는 스레드가 바로 AWT Event-Queue Thread라는 것이 문제다. 사용자가 아무리 닫기 버튼을 눌러도 이 스레드가 작업을 모두 마칠 때까지는 닫기 버튼이 발생한 이벤트는 처리되지 않는 것이다.

이를 해결하기 위해서 GeoPixel은 감독자를 별도의 스레드로 구성하였다. 이 때문에 AWT Event-Queue Thread는 사용자 인터페이스의 확인 버튼이 발생한 이벤트만 처리하고 별도의 스레드를 생성하고 이를 실행한 후 다시 다른 이벤트 처리를 위해 대기한다. 새로 생성된 스레드는 GeoProgressDialog형 다이얼로그를 화면 표출하고 입력 받은 매개변수 값을 토대로 작업자를 생성하고 이를 실행한다. 따라서 사용자가 다이얼로그를 닫을 경우 이는 대기 중인 AWT Event-Queue Thread에 의해 즉시 처리되어 감독자에게 작업 중단을 지시할 수 있다. 별도의 스레드 구현을 위해 모든 감독자는 Supervisor 클래스를 상속받고 Supervisor 클래스는 다시 Runnable 인터페이스를 구현하였다.

감독자에게 영상 처리에 필요한 매개변수 값을 사용자 인터페이스를 통하지 않고도 전달할 수 있다. 한 가지 방법은 입출력 영상에 대한 정보만 constructor를 통해 전달하는 것인데 이는 사용자가 아닌 통합 소프트웨어 환경이 감독자를 호출할 때 주로 이용한다. 감독자는 통합 환경이 제공한 정보만 사용자 인터페이스에 지정한 후 화면 표출하고 나머지 매개변수 값은 사용자가 지정한다. 또 다른 방법은 감독자가 필요로 하는 모든 정보를 constructor를 통해 전달받는 것이다. 이 방법은 주로 처리 작업에 해당하는 명령어를 사용자가 직접 키입력해서 실행할 경우에 사용된다. 사용자가 키입력한 정보를 파싱(parsing)하여 매개변수 값을 추출하고 이를 해당 감독자의 constructor에 지정하는 것이다. 이 때 감독자는 사용자 인터페이스를 화면 표출하지 않고 바로 작업자를 생성, 실행한다.

작업자들이 맡은 임무는 세분화되고 잘 정의된 영상 처리 작업을 수행하는 것이다. 이를 위해서 작업자들은 영상의 화소형과 영상의 종류, 즉 파일에 저장된 영상인가 아니면 메모리에 저장된 영상인가에 따라 분기된 메서드를 사용하여 작업을 수행한다. 아울러 작업자는 사용자의 작업 중단 요구를 처리하기 위해서 작업 도중 중단 요구를 표시하는 플래그를 항상 점검하도록 설계되었다.

위와 같이 감독자와 작업자는 다양한 활동을 서로 공유하고 있으므로 GeoPixel의 개발 라이브러리를 공통의 임무를 각각 Supervisor와 Operator 클래스에 정의하고 감독자와 작업자들이 이 클래스를 상속받아 재사용하도록 설계하였다. 개발자는 감독자와 작업자의 대부분 기능은 단지 클래스 상속만으로 구현할 수 있으므로 새로운 감독자와 작업자를 손쉽게 개발할 수 있다. 아울러 감독자와 작업자가 필요한 새로운 공통의 기능도 단지 Supervisor와 Operator 클래스의 수정만으로 추가할 수 있으므로 프로그램 기능 향상을 쉽게 꾀할 수 있다.

### 제 3 절 영상 처리 모듈

본 절에서는 GeoPixel에서 지원하는 주요 영상 처리 관련 모듈에 대해서 간단히 소개하고자 한다.

#### 1. 영상 화면 표출

GeoPixel ver.1.0은 현재 바이트형 화소의 흑백, RGB 칼라, 위색 화면 표출을 지원한다. 화면 표출은 다음 네 가지 레이어를 통해 이루어진다. 가장 밑에 놓인 Image Layer는 영상의 화면 표출을 담당한다. 그 위에 놓인 Vector Layer는 벡터 자료를 영상 위에 중첩시킬 때 벡터의 화면 표출을 담당한다. AOI Layer는 사용자가 선택한 영역을 표시하며 Object Layer는 GCP, Grid 같은 다양한 형태의 사상을 영상 위에 중첩시킨다(Fig. 5-1).

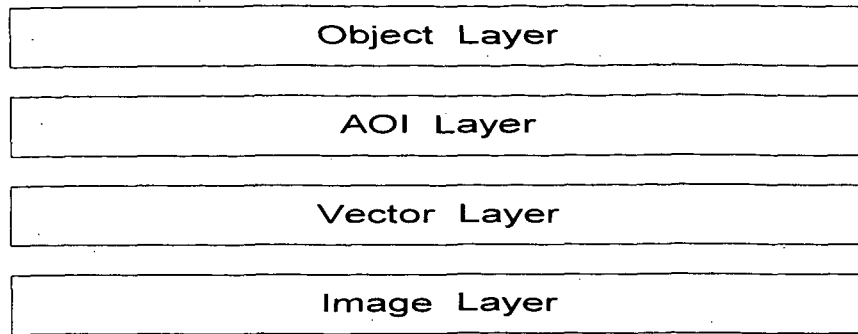


Figure 5-1. Systematic structure for image display on screen

하나의 영상은 세 개의 화면 영역으로 분할되어 표출되는데 Scroll View는 영상 전체에 대해 resample한 영상을 사용자에게 보여주고 Main View는 Scroll View에서 사용자가 선택한 영역을 1:1 비율로 보여주며 Zoom View는 Main View에서 선택한 영역을 확대하여 보여준다(Fig. 5-2). 따라서 화면 표출시 메모리 사용량은 전체 영상의 일부분인 resample한 영상과 1:1 비율 영상뿐이므로 대용량 위성 영상인 경우에도 실시간으로 사용자에게 원하는 영역을 화면 표출할 수 있다.

Main View는 Lookup Table, Palette, AOI, Vector, Option 등 네 가지 팝업 메뉴가 지원한다. Lookup Table 메뉴는 각 채널의 원 영상과 향상된 영상의 히스토그램을 보여주고 다양한 영상 향상을 사용자 선택할 수 있도록 지원한다(Fig. 5-3). Palette 메뉴는 영상의 화소값이나 벡터의 ID에 대응하는 칼라를 설정한다(Fig. 5-4). AOI 메뉴는 64개의 채널을 통해 사용자가 다각형을 편집하여 원하는 영역을 표시하도록 지원한다(Fig. 5-5). Vector 메뉴는 영상 위에 벡터 자료를 중첩해서 표출할 때 사용하는 메뉴로서 벡터의 ID나 채널에 따라 칼라를 달리할 수 있고 폴리곤의 채움 기능도 선택할 수 있다(Fig. 5-6). 또한 화면에 나타나는 사각형의 색상이나 커서의 형태를 정의할 수 있으며, 영상에 주어진 거리에 따라 격자망을 표현할 수 있다(Fig. 5-7).



Figure 5.2 Image display on scroll, main, zoom window

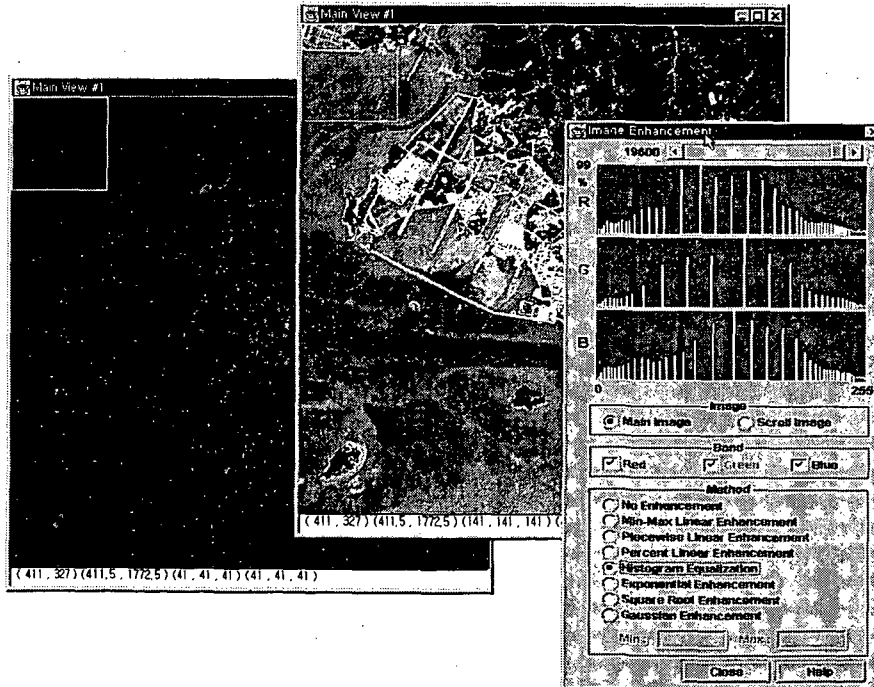


Figure 5-3. Interface of Lookup Table for image enhancements





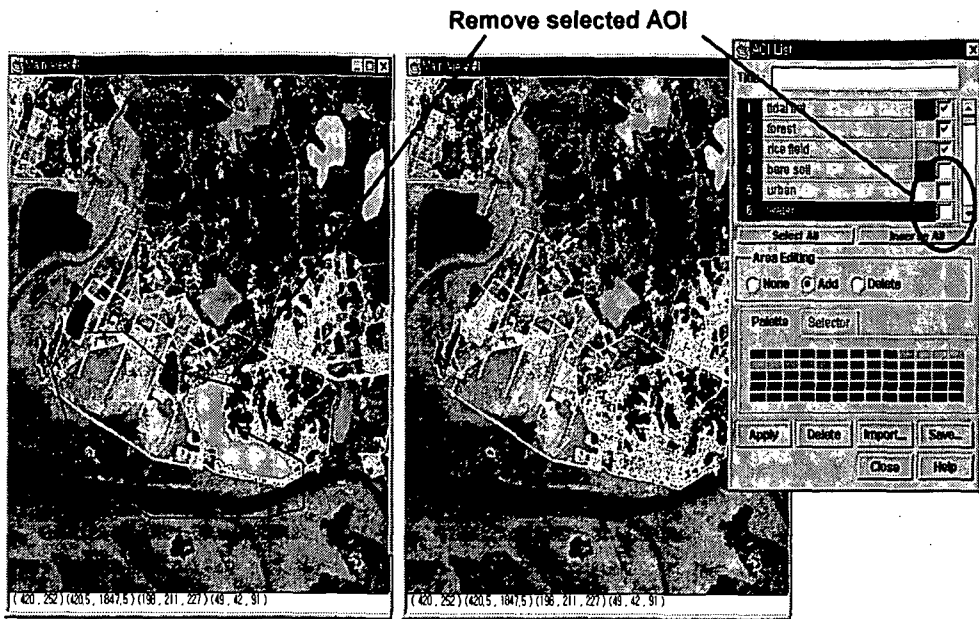


Figure 5-5. Interface of Area of Interest(AOI) for define refernce area

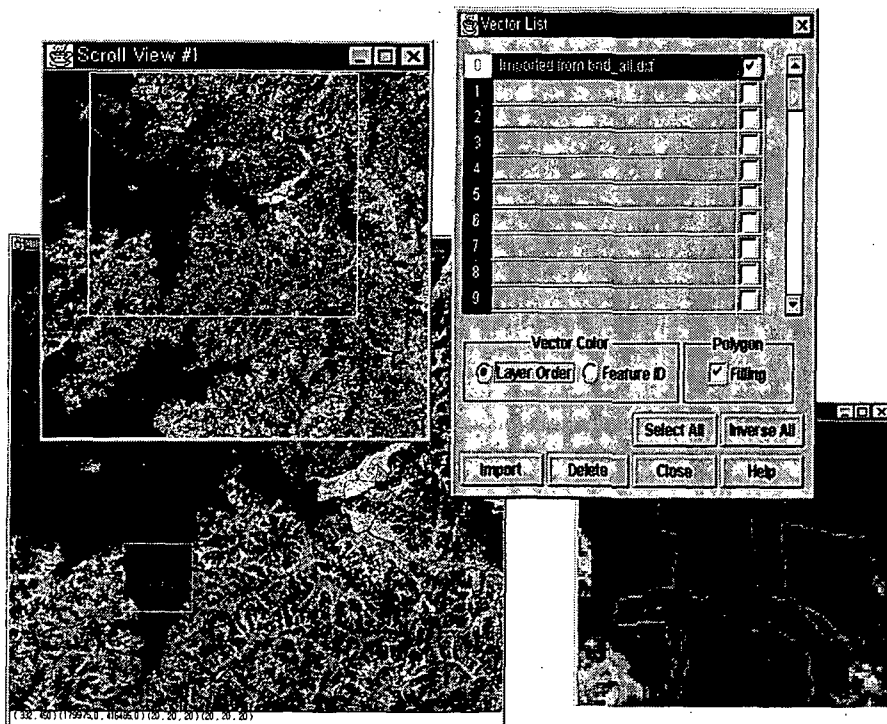


Figure 5-6. Interface for vector drawing on raster image

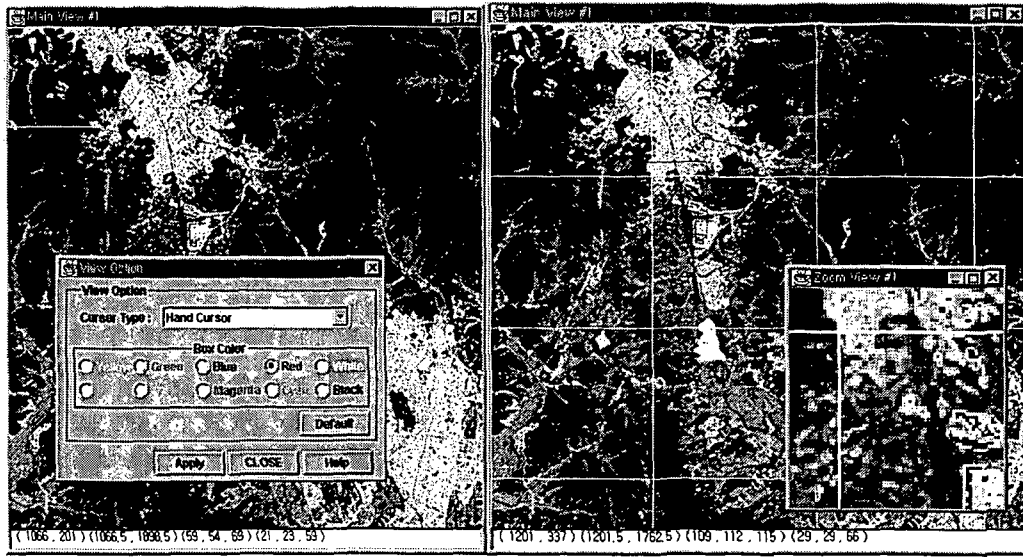


Figure 5-7 Interface for setting of cursor type, rectangle color and grid

영상 화면 표시와 직접적으로 관련있는 클래스와 그 기능은 다음과 같다.

CubeViewer	화면 표시를 관리하는 감독자
CubeViewOp	화면 표시를 수행하는 작업자
GeoScrollViewFrame	Scroll View Frame
GeoMainViewFrame	Main View Frame
GeoZoomViewFrame	Zoom View Frame
LookupReplacer	Lookup Table을 이용한 영상 향상

## 2. 영상 향상

GeoPixel은 영상 향상(image enhancement)을 두 가지 모드에서 지원한다. On-line 모드에서는 영상을 화면 출력한 상태에서 Lookup Popup Menu를 선택하면 lookup table를 변환하여 화면 영상을 향상시킬 수 있다. 이 때 지원되는 화소형은 바이트형에 한한다. On-line 영상 향상(Fig. 5-3)과 직접적으로 관련된 클래스와 그 기능은 다음과 같다.

```

public class LookupReplacer {
    public void setLookupTable(byte[] lookup);//lookup table을 지정한다.
    public byte[] getLookupTable();//지정된 look table을 가져온다.
    public byte[] replace(byte[] a, byte[] b);
    //a 배열을 지정된 lookup table을 사용하여 b 배열로 변환한다.
    public static byte[] interactivelinearEnhance(byte[] a, byte[] lookup, int[] lut);
    //사용자가 지정한 lookup table을 사용하여 변환한다.
    public static byte[] linearEnhance(byte[] a, int min, int max, byte[] lookup);
    //선형 영상 향상
    public static byte[] minmaxLinearEnhance(byte[] a, byte[] lookup);
    //최대-최소 선형 영상 향상
    public static byte[] percentLinearEnhance(byte[] a, float low, float high, byte[]
lookup);//퍼센트 선형 영상 향상
    public static byte[] histogramEqualize(byte[] a, byte[] lookup);
    //히스토그램 균등화 영상 향상
    public static byte[] histogramGaussian(byte[] a, byte[] lookup);
    //Gaussian 영상 향상
    public static byte[] histogramLog(byte[] a, int min, int max, byte[] lookup);
    //Log 영상 향상
    public static byte[] histogramSquareRoot(byte[] a, int min, int max, byte[]
lookup);
    //Square root 영상 향상
    public static byte[] histogramSquare(byte[] a, int min, int max, byte[] lookup);
    //Square 영상 향상
    public static byte[] histogramExponential(byte[] a, int min, int max, byte[]
lookup);
    //Exponential 영상 향상

```

Off-line 모드에서는 임의의 화소형을 가진 영상 전체에 대해서 영상 향상을 실행한다. 히스토그램이 반드시 필요한 영상 향상에서 화소형이 바이트형이 아닌 경우 최대,최소값을 512단계로 나누어 히스토그램을 만든 후 이를 다시 256단계로 영상 향상을 실행하였다.

대용량 영상을 메모리에 모두 올려놓고 향상 작업을 할 경우 swaping에 의한 심각한 속도 저하가 발생할 수 있으므로 GeoPixel에서는 영상의 한 라인만 읽은 후 향상 작업을 수행하고 이를 출력 파일에 저장한 후 다시 다음 라인을 읽는 방법을 사용하였다. 따라서 영상 향상 작업시 GeoPixel이 사용하는 시스템 메모리는 한 라인의 두 세배에 그친다.

관련된 클래스와 그 기능은 다음과 같다.

MinMaxLinearEnhancer	선형 영상 향상을 위한 감독자
PercentLinearEnhancer	히스토그램 균등화와 같은 히스토그램을 반드시 사용하는 영상 향상을 위한 감독자
InteractiveEnhancer	사용자가 지정한 형태의 변환함수를 사용한 영상 향상을 작업자
MinMaxLinearEnhanceOp	선형 영상 향상을 위한 작업자
HistogramModifyEnhanceOp	히스토그램 균등화와 같은 히스토그램을 반드시 사용하는 영상 향상을 위한 작업자
InteractiveEnhanceOp	사용자가 지정한 형태의 변환함수를 사용한 영상 향상을 작업자

### 3. 영상 필터링

GeoPixel은 공간 영역 필터링(spatial domain filtering)과 주파수 영역 필터링

(frequency domain filtering)을 지원한다. 대용량 영상을 메모리에 모두 올려놓고 공간 영역 필터링 작업을 할 경우 swapping에 의한 심각한 속도 저하가 발생할 수 있으므로 GeoPixel에서는 커널의 행수에 해당하는 라인만 영상에서 읽은 후 필터링 작업을 수행하고 이를 출력 파일에 저장한 후 다시 다음 라인을 읽는 방법을 사용하였다. 따라서 공간 영역 영상 필터링 작업시 GeoPixel이 사용하는 시스템 메모리는 커널의 행수 만큼의 라인의 두 세배에 그친다.

공간 영역 필터링과 직접적으로 관련된 클래스와 그 기능은 다음과 같다.

SpatialPDKernelFilter	이미 정의된 커널을 이용한 필터링을 위한 감독자
SpatialUDKernelFilter	사용자가 정의한 커널을 이용한 필터링을 위한 감독자
SpatialKernelFilterOp	커널을 사용한 필터링을 위한 작업자
GeoKernel	커널에 대한 정보를 갖는 클래스

주파수 영역 필터링은 순방향 FFT 변환된 영상을 원형의 AOI와 함께 역방향 FFT 변환하여 작업을 수행한다. 주파수 영역 필터링과 직접적으로 관련된 클래스와 그 기능은 다음과 같다.

ForwardFFTTTransformer	순방향 FFT 변환을 위한 감독자
BackwardFFTTTransformer	역방향 FFT 변환을 위한 감독자
FFTOp	FFT 변환을 위한 작업자
FFTPowerSpectrumOp	Power Spectrum 생성을 위한 작업자

대상 영상에 대하여 dpt지 추출을 위한 필터링과 주파수 영역에서의 필터링 결과는 Fig. 5-8과 Fig.5-9에 나타난 것과 같다.

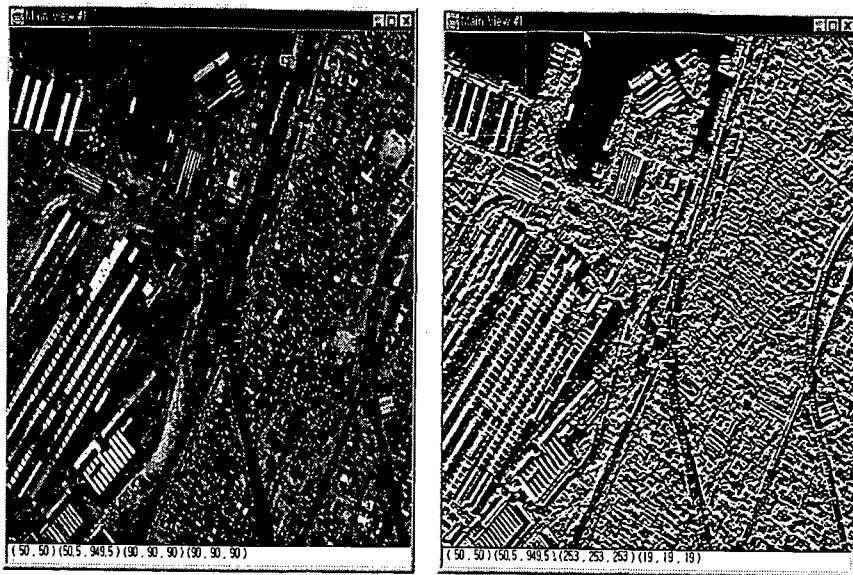


Figure 5-8 Example of edge detection using sobel operator

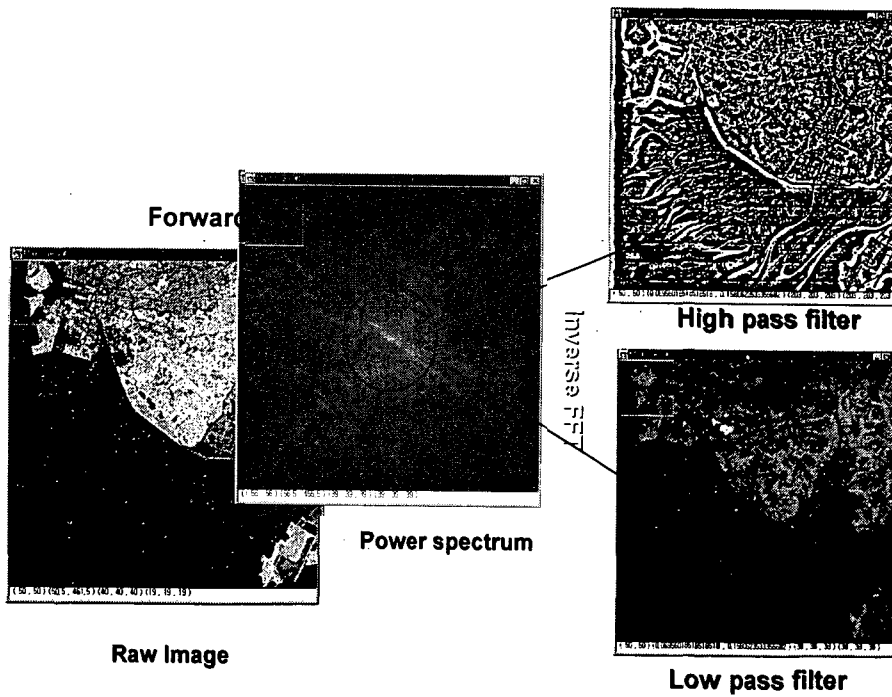


Figure 5-9 Example of fast forward fourier transform and inverse filtering

#### 4. 영상 분류

GeoPixel은 K-Means, ISODATA 등의 무감독자 분류와 MLC 감독자 분류를 지원한다.

대용량 영상을 메모리에 모두 올려놓고 분류를 수행할 경우 심각한 속도 저하가 발생할 수 있으므로 GeoPixel에서는 커널의 행수에 해당하는 라인만 영상에서 읽은 후 통계 추출과 분류 작업을 수행하고 이를 출력 파일에 저장한 후 다시 다음 라인을 읽는 방법을 사용하였다. 따라서 공간 영역 영상 필터링 작업시 GeoPixel이 사용하는 시스템 메모리는 한 라인의 두 세배에 그친다.

무감독 분류와 직접적으로 관련된 클래스와 그 기능은 다음과 같다.

KMeansClassifier	K-Means 무감독 분류를 위한 감독자
KMeansClassifyOp	K-Means 무감독 분류를 위한 작업자
ISODATAClassifier	ISODATA 무감독 분류를 위한 감독자
ISODATAClassifyOp	ISODATA 무감독 분류를 위한 작업자

MLC 분류는 사용자가 훈련지역을 AOI로 지정하면 GeoPixel은 AOI에서 지정한 영역의 영상에 대한 통계 자료를 계산한 후 이 통계 자료를 토대로 분류를 수행한다. 이때 대용량 영상을 효율적으로 다루기 위해서 통계자료와 분류는 각 라인별로 메모리에 읽어 들여 수행한다.

이를 위하여 영상의 기초적 특성을 분석하기 위한 것으로 히스토그램, 스캐터그램 분석이 있다(Fig. 5-10, Fig. 5-11).

HistogramMaker/Viewer	영상 단일 분광대의 농도분포를 작성을 위한 감독자
HistogramOp	단일 분광대의 농도 분포 작성을 위한 작업자
ScattergramMaker/Viewer	2개 분광대의 농도분포를 작성을 위한 감독자
ScattergramOp	2개 분광대의 농도 분포 작성을 위한 작업자

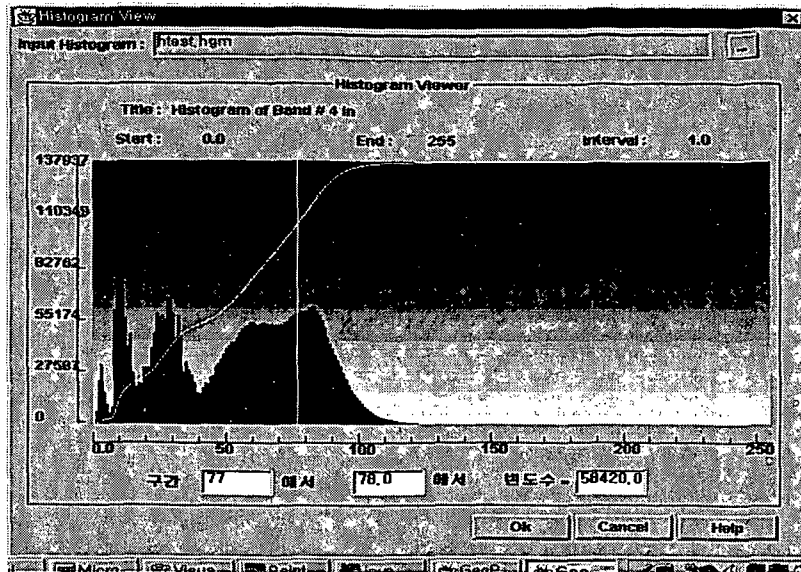


Figure 5-10 Example of Histogram

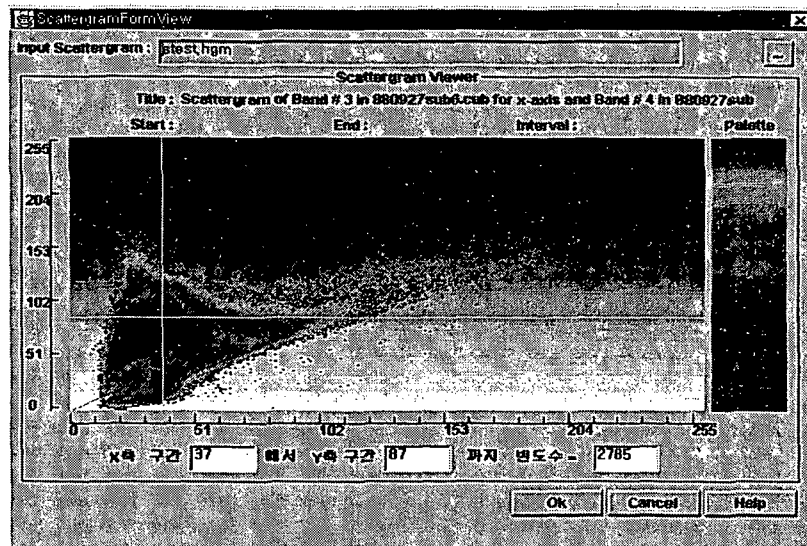


Figure 5-11 Example of Scattergram

MLC 분류와 직접적으로 관련된 클래스와 그 기능은 다음과 같다(Fig. 5-12)

- |                          |                           |
|--------------------------|---------------------------|
| CubeStatisticsCalclater  | 영상 통계 산출을 위한 감독자          |
| AOIStatisticsCalculateOp | AOI를 사용한 영상 통계 산출을 위한 작업자 |



MLClassifier  
MLClassifyOp

MLC를 위한 감독자  
MLC를 위한 작업자

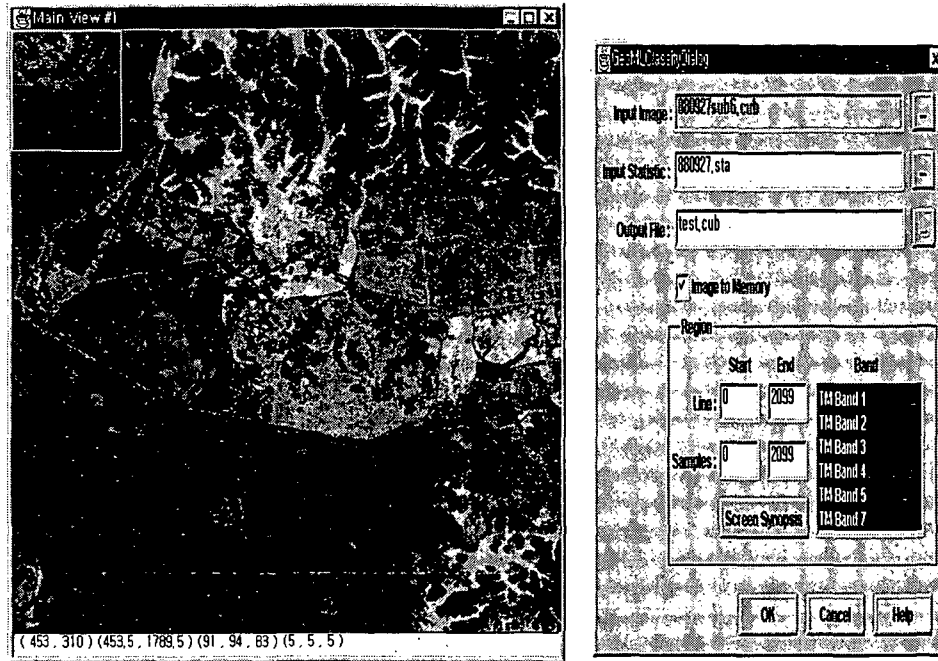


Figure 5-12 Example of Land Cover Classification using MLC method

## 5. 지표면 특징 분석

영상으로부터 지표면의 정보를 추출하기 위한 정보 분석으로는 주성분 분석, TasseledCap분석, 지표면 온도 계산, 식생 지수 산출 등이 있다. 이러한 분석은 위성 센서별로 그 적용이 달라지기 때문에 대화창에서의 선택항목을 주의하여 선택하여야 한다. 지표면 특성 분석과 관련된 클래스와 그 기능은 다음과 같다(Fig 5-13 ~ Fig. 5-16) .

PCATransformer  
PCATransformOp

영상의 주성분 분석을 위한 감독자  
영상의 주성분 분석을 위한 작업자

TasseledCapTransformer

영상의 TasseledCap변환을 위한 감독자

TasseledCapTransformOp	영상의 TasseledCap변환을 위한 작업자
VegetationIndexer	식생 지수 계산을 위한 감독자
VegetationIndexOp	식생 지수 계산을 위한 작업자
TM6ToTemperatureConverter	LANDSAT TM band 6로부터 지표면 온도 산출을 위한 감독자
TM6ToTemperatureOp	LANDSAT TM band 6로부터 지표면 온도 산출을 위한 작업자

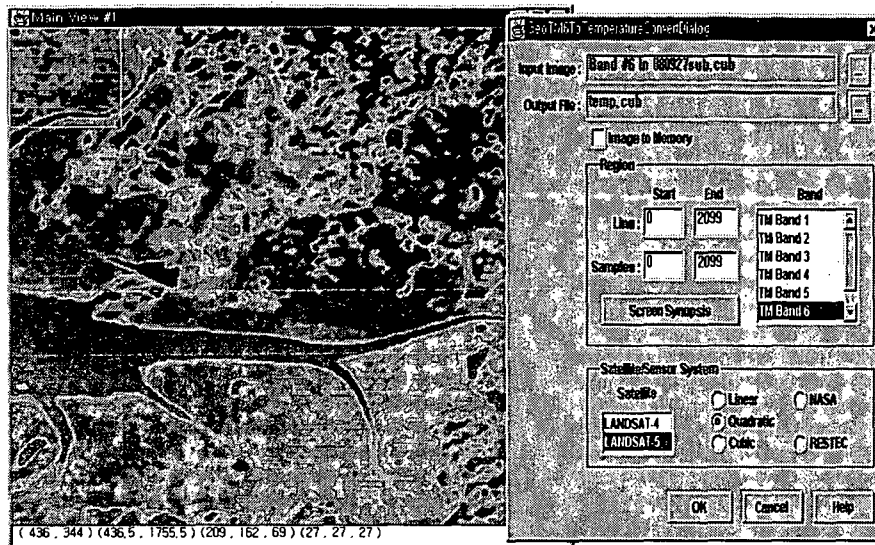


Figure 5-13 Example of extraction of surface temperature

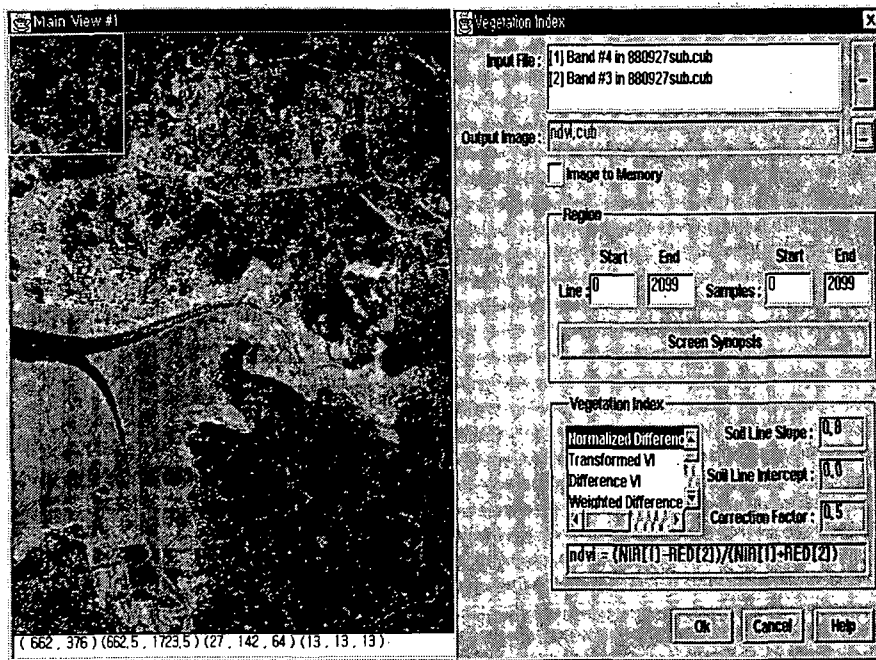


Figure 5-14 Example of extraction vegetation index(NDVI)

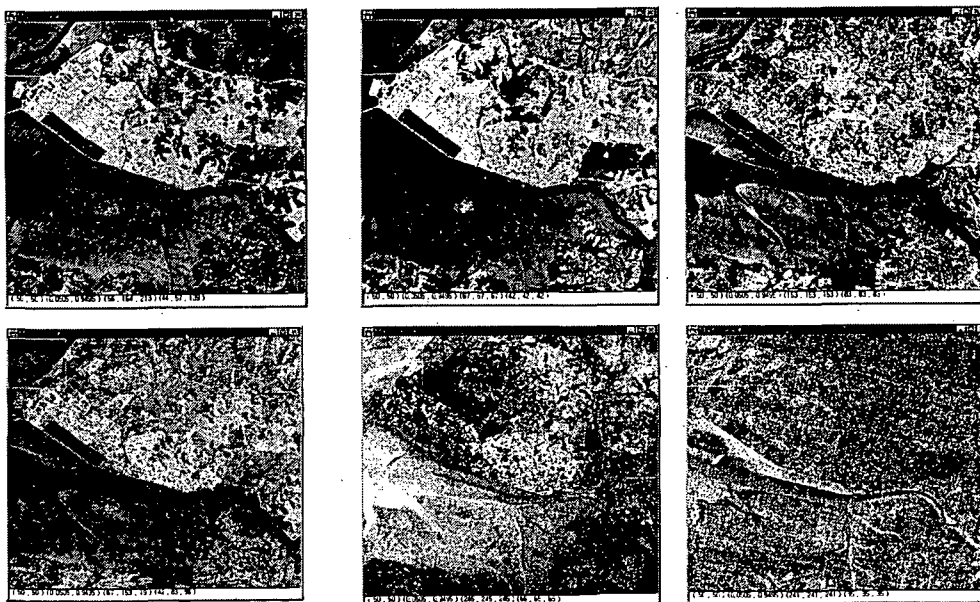


Figure 5-15 Example of principal component analysis

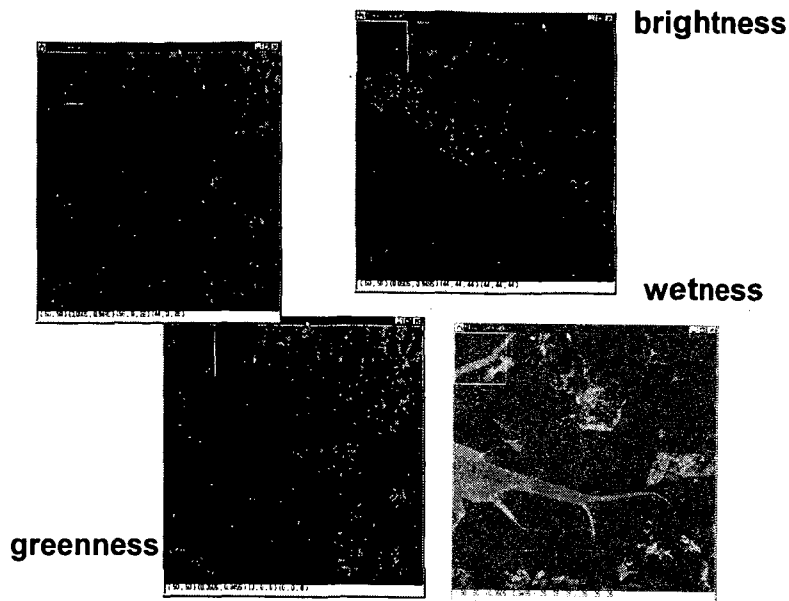


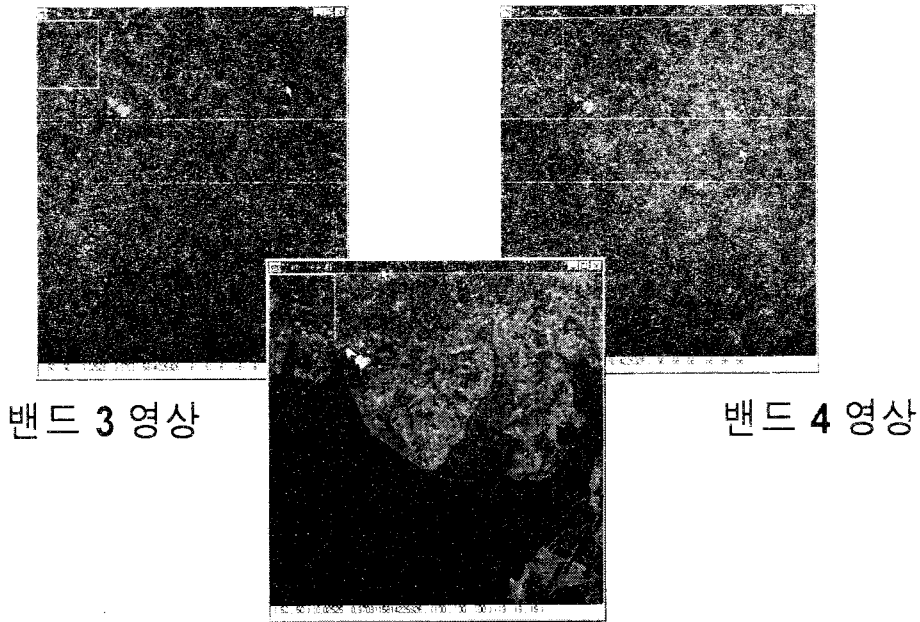
Figure 5-16 Example of tasseledcap transform

## 6. 기타 영상 처리

일반적인 영상 처리를 위한 기능으로는 부분 영상의 추출, 확대 및 축소, 영상의 회전, 단일 분광대 또는 복수 분광대에 대한 산술 연산등의 기능을 제공한다(Fig. 5-17 5-18, 5-19, 5-20).

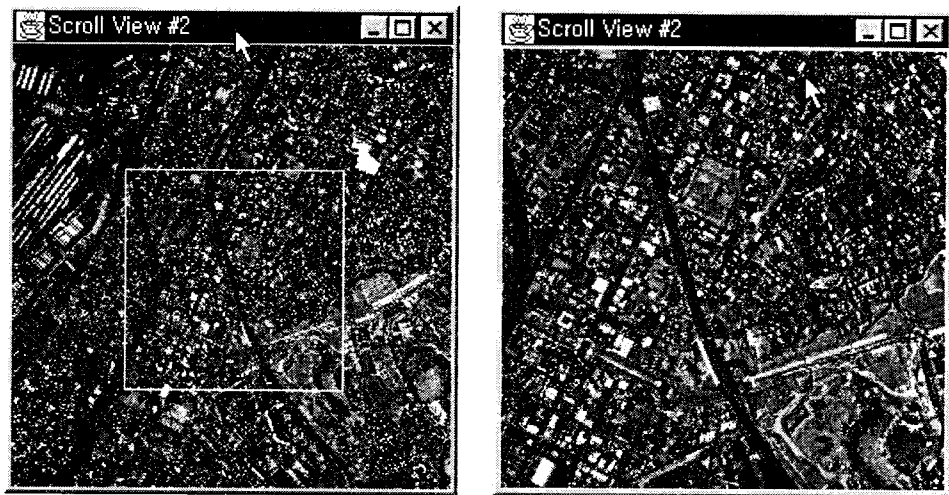
UnaryArithmeticFunctioner	단일 분광대에 대한 산술연산을 위한 감독자
UnaryArithmeticFunctionOp	단일 분광대에 대한 산술연산을 위한 작업자
UnaryBitwiseFunctioner	단일 분광대에 대한 비트연산을 위한 감독자
UnaryBitwiseFunctionOp	단일 분광대에 대한 비트연산을 위한 작업자
UnaryLogicalFunctioner	단일 분광대에 대한 논리연산을 위한 감독자
UnaryLogicalFunctionOp	단일 분광대에 대한 논리연산을 위한 작업자
UnaryMathFunctioner	단일 분광대에 대한 수학연산을 위한 감독자
UnaryMathFunctionOp	단일 분광대에 대한 수학연산을 위한 작업자

BinaryArithmeticFunctioner	두 분광대에 대한 산술연산을 위한 감독자
BinaryArithmeticFunctionOp	두 분광대에 대한 산술연산을 위한 작업자
BinaryLogicalFunctioner	두 분광대에 대한 논리연산을 위한 감독자
BinaryLogicalFunctionOp	두 분광대에 대한 산술연산을 위한 작업자
CubeGeometryFliper	영상에 대한 기하학적 변형을 위한 감독자
CubeGeometryFlipOp	영상에 대한 기하학적 변형을 위한 작업자
CubeGeometryResizer	영상의 크기 변환을 위한 감독자
CubeGeometryResizeOp	영상의 크기 변환을 위한 작업자
CubeGeometryRotater	영상의 회전을 위한 감독자
CubeGeometryRotateOp	영상의 회전을 위한 작업자



**밴드 3과 4 overlay연산(k = 0.3)**

Figure 5-17 Example of binary arithmetic operation(overlay)



원 영상

부분 영상

Figure 5-18 Example of extraction partial image

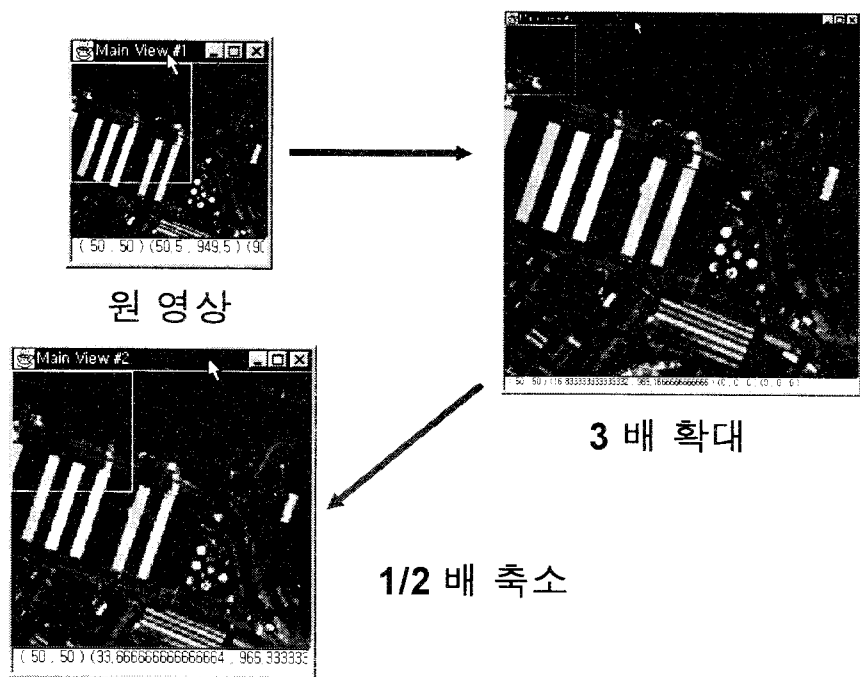


Figure 5-19 Example of resizing with arbitrary scale factor

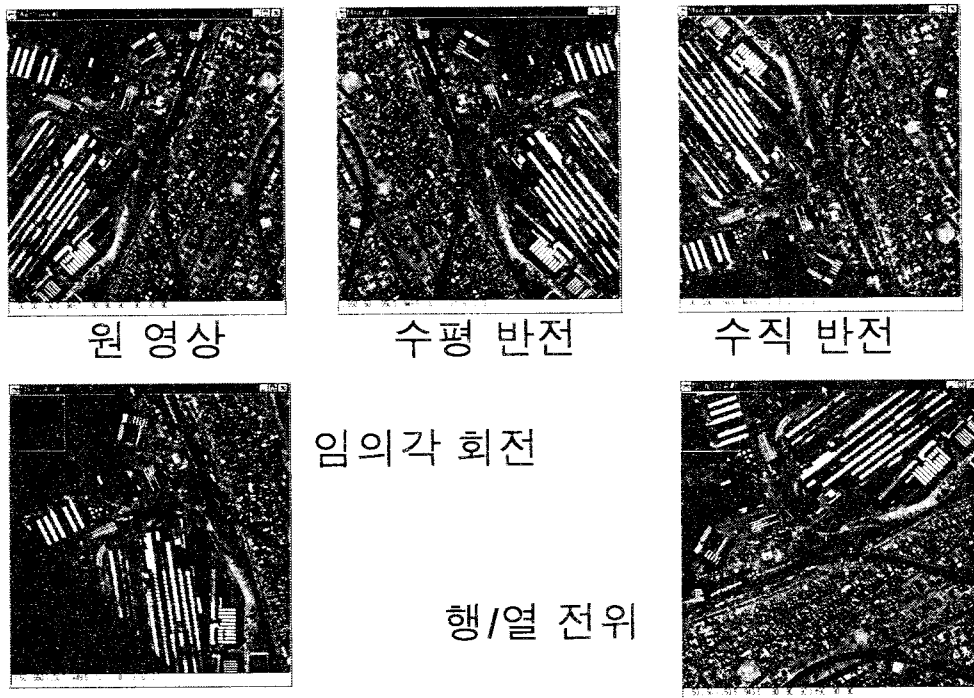


Figure 5-20 Example of flip/translation/rotation of image

## 제 4 절 사용자 인터페이스

### 1. 핵심어 탐색 엔진

#### 가. 연구 배경

위성 영상 처리에서 사용하는 알고리즘은 매우 다양하므로 통합 소프트웨어는 많은 처리 모듈과 각 모듈의 여러 가지 선택 사항의 조합으로 구성된다. 따라서 사용자가 자신이 원하는 모듈에 쉽고 빠르게 접근할 수 있는 보다 전문화된 인터페이스가 요구된다.

현재 사용되고 있는 모듈 접근 방법은 크게 두 가지로 분류될 수 있다. 텍스트 모드에서 키보드 입력을 통한 모듈 선택은 전산기의 초창기 모델 때부터 사용된 접근 방법으로서 해당 명령어나 약자를 기억해야 하므로 초보자가 원하는 모듈에 쉽게 접

근할 수 없다. 한편 GUI(Graphic User Interface)로 구현된 메뉴를 통한 모듈 선택은 최근 사용자 인터페이스에 대한 연구 성과에 힘입어 위성 영상 처리 분야에도 널리 도입되고 있다. 그러나 제공되는 모듈의 수가 많아지고 서로 간의 관계 설정이 까다로워짐에 따라 지나치게 복잡한 메뉴 계층 구조를 가질 수 있다.

특히 위성 영상 처리 모듈은 기능, 대상 영상의 종류, 응용 분야 등에 따라 여러 가지 다른 그룹으로 재분류될 수 있는데 위에서 언급한 두 가지 접근 방법을 사용하여 재편성된 그룹을 통한 단계적인 접근을 지원하기는 어렵다.

이를 극복하기 위해서 본 발표에서는 사용자가 지정한 다양한 핵심어를 이용하여 원하는 모듈을 선택하는 새로운 형태의 사용자 인터페이스를 위성영상 처리 통합 소프트웨어에 이용한 예를 기술하고자 한다.

#### 나. 핵심어 탐색을 이용한 모듈 접근 과정

본 연구에서 핵심어를 이용하여 사용자의 기호에 따라 모듈 접근 경로를 변경할 수 있도록 개발된 사용자 인터페이스의 기본 원리는 Fig. 5-21과 같다. 사용자가 문자를 입력할 때마다 입력한 문자열을 단어별로 나눈 후 각 단어와 연결된 모듈들의 교집합을 사용자에게 제시한다. 만약 교집합이 공집합일 경우에는 다음 핵심어를 입력하기 위한 중간 과정으로 보고 사용자에게 제시된 명령어의 리스트를 갱신하지 않는다. 이 방법에 따라 하나의 모듈은 다양한 기준에 따라 재분류될 수 있으므로 사용자는 익숙한 분류 기준에 따라 점진적으로 원하는 모듈에 접근할 수 있다.

사용자가 문자를 입력할 때마다 핵심어와 관련된 모듈 집합의 탐색 작업이 수행되어야 하므로 핵심어와 모듈을 연결하는 자료 구조의 적절한 선택이 인터페이스의 성능을 결정한다.



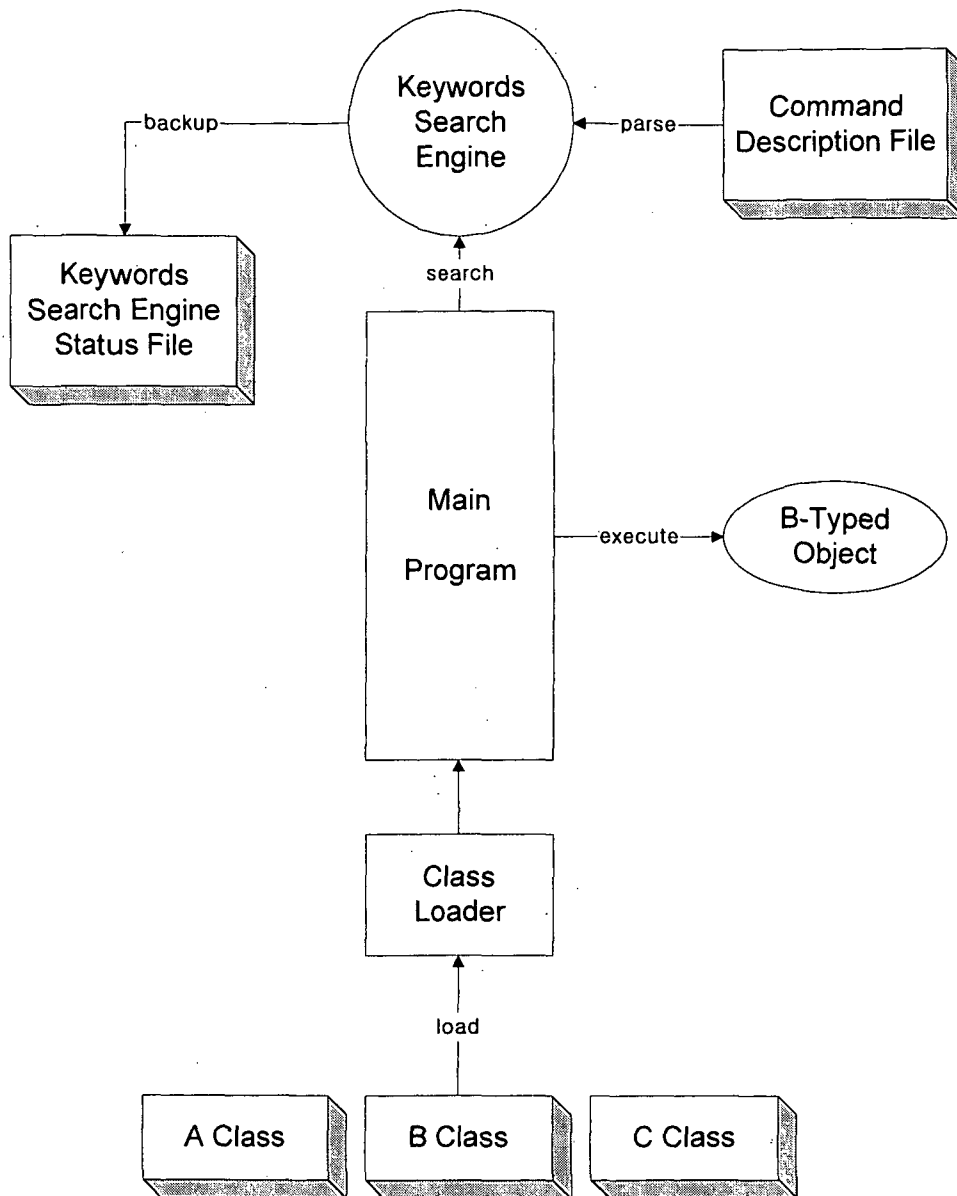


Figure 5-21 General structure of user interface for keyword search

본 연구에서는 해쉬 코드(hash code)를 이용하여 하나의 키(key)와 연결된 복수의 원소(element)를 신속히 찾아주는 해쉬 집합(hash set)을 사용하였다. 메모리의 효율적인 사용을 위하여 각 단계에서 해쉬 집합은 교집합을 구한 후 메모리에서 제거된다.

다. 사용자 인터페이스 구현

본 연구에서 구현한 핵심어 탐색 엔진을 이용한 사용자 인터페이스의 전체 구조는 Fig. 5-22와 같다.

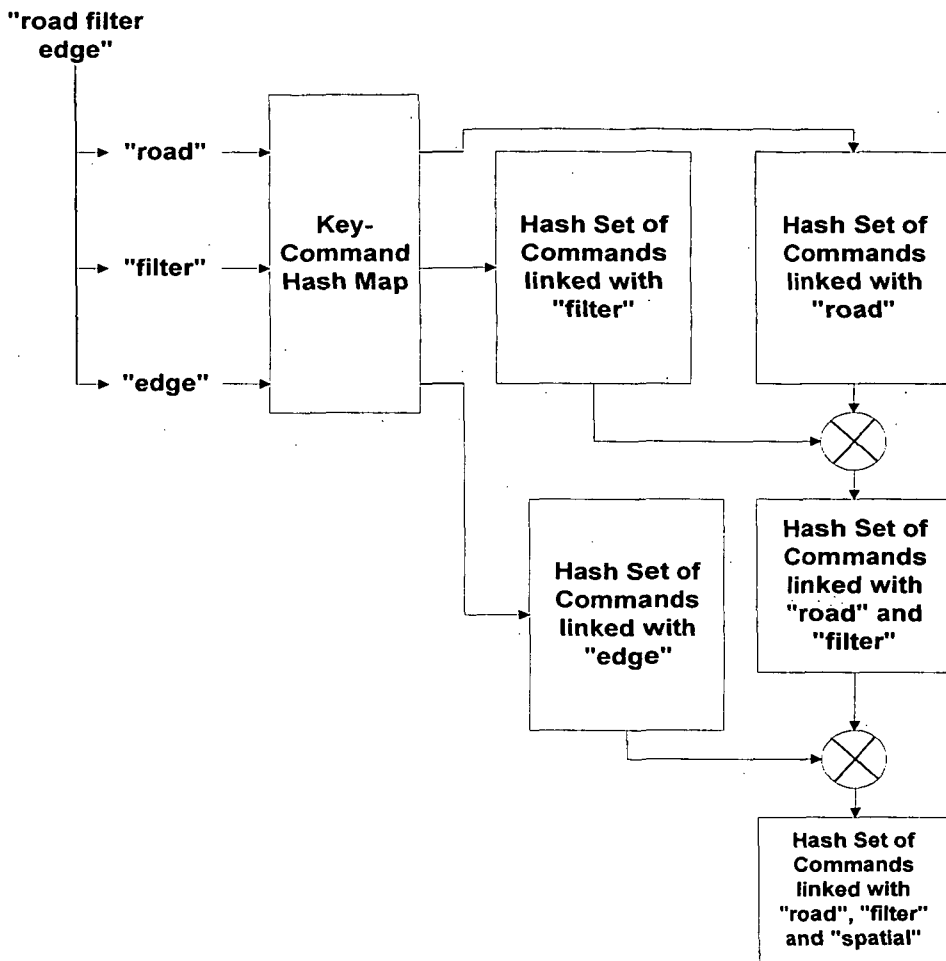


Figure 5-22 Access of processing module via keyword search

먼저 핵심어 탐색 엔진은 텍스트 형식의 명령어 기술 파일(command description file)을 해석하여 핵심어와 모듈간의 연결 관계를 정의한다. 이 연결 관계는 핵심어 탐색 엔진의 상태 저장 파일(keyword search engine status file)에 바이너리 형태로 저장되어 소프트웨어 재가동시에도 연결관계를 그대로 보존한다. 주 프로그램은 사용자

가 입력한 문자열을 핵심어 탐색 엔진에 문의하여 핵심어들과 연결된 모듈들을 추출하고 이에 대한 설명을 사용자에게 보여준다. 사용자가 그 중 한 모듈을 선택하면 해당 모듈을 메모리로 불러들여 실행한다.

본 연구에서 사용된 명령어 기술 파일의 예는 다음과 같다.

```
#GeoPixel Command Descriptions
```

```
ID=MeanFilter
```

```
DESCRIPTION="Mean Filter with 3x3 Kernel"
```

```
KEYS={SPATIAL,MEAN,FILTER,LOW}
```

```
ID=SobelFilter
```

```
DESCRIPTION="Sobel Filter with 3x3 Kernel"
```

```
KEYS={SPATIAL, SOB, SOBEL, FILTER, EDGE, HIGH, ROAD}
```

하나의 명령어는 처리 모듈의 인식자인 ID와 모듈의 기능을 간단하게 서술한 DESCRIPTION, 해당 모듈과 연결될 핵심어의 배열인 KEYS로 구성된다. 명령어 기술 파일은 텍스트 형태로 사용자에게 제공되므로 사용자는 이를 편집하여 기존의 핵심어와 모듈간의 연결 관계를 바꾸거나 새로운 핵심어 및 모듈을 추가할 수 있다.

사용자가 소프트웨어의 실행을 종료한 후에도 핵심어와 모듈간의 연결 상태를 따로 저장하여서 다음 사용시에도 같은 연결 상태를 가진 인터페이스에서 작업할 수 있어야 한다. 명령어 기술 파일은 텍스트 형태로 저장되므로 이를 다시 해석하는데 많은 시간이 필요하다. 연결 관계에 대한 빠른 해석을 지원하기 위해서 엔진이 갖고 있는 해쉬 집합의 키와 원소의 관계를 그대로 저장하고 있는 이진 파일 형식의 핵심어 탐색 엔진의 상태 저장 파일이 만들었다. 본 연구에서는 이 파일을 객체의 직렬화(serialization)을 이용하여 구현하였다.

텍스트 편집만으로 새로운 모듈을 소프트웨어에 통합시키기 위해서는 동적 연결(dynamic linking)을 통해 해당 모듈을 메모리에 불러 들여야 한다. 본 연구에서 각 처리 모듈은 Supervisor라는 공통의 클래스에서 상속받아 구현하였다. 주 프로그램은 해

당 클래스를 메모리로 읽어 들인 후 객체의 다형성(polymorphism)을 이용하여 Supervisor형의 객체를 생성하고 공통의 메서드를 호출함으로써 처리 모듈을 실행한다 (Fig. 5-23).

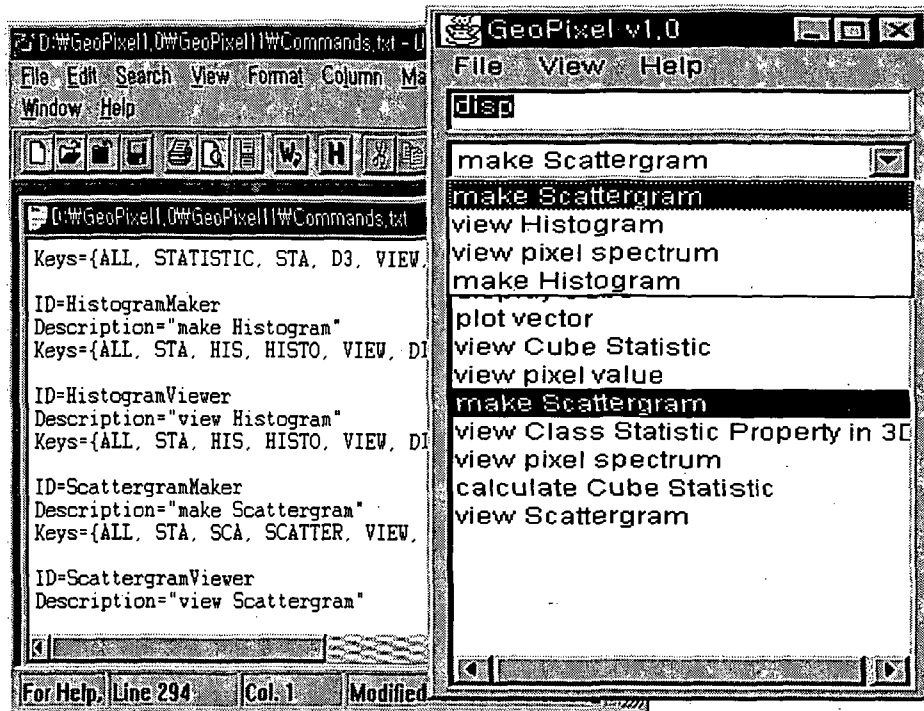


Figure 5-23 Example of command description file and frame

## 2. 프로그래머블 메뉴

### 가. 연구의 필요성

GUI(Graphic User Interface)를 통한 메뉴 접근 방법은 현재 가장 널리 사용되고 있는 인터페이스이다. 그러나 위성 영상 처리 통합 소프트웨어와 같이 다양한 종류의 알고리즘을 지원하는 프로그램에서는 제공되는 모듈의 수가 많아지고 서로 간의 관계 설정이 까다로워짐에 따라 지나치게 복잡한 메뉴 계층 구조를 가질 우려가 있다.

또한 사용자의 업무에 특화된 메뉴 구조를 사용자가 직접 꾸밀 수 있고 새로 개발된 모듈을 텍스트 편집을 통해 손쉽게 추가할 수 있어야 한다. 이를 지원하기 위해서



본 연구에서 사용된 메뉴 기술 파일의 예는 Figure 5-25와 같다. 하나의 메뉴는 메뉴의 인식자인 ID와 상위 메뉴를 지정하는 PARENT, 메뉴의 레이블을 정의하는 LABEL, 단축키를 지정하는 KEY, 연결될 모듈명을 가리키는 CLASS 등의 키로 구성된다. 메뉴 기술 파일은 텍스트 형태로 사용자에게 제공되므로 사용자는 이를 편집하여 메뉴 계층의 구조이나 모듈간의 연결 관계를 바꾸거나 새로운 메뉴 및 모듈을 쉽게 추가할 수 있다.

```
#GeoPixel Menu Descriptions

ID=MENU1
PARENT=ROOT

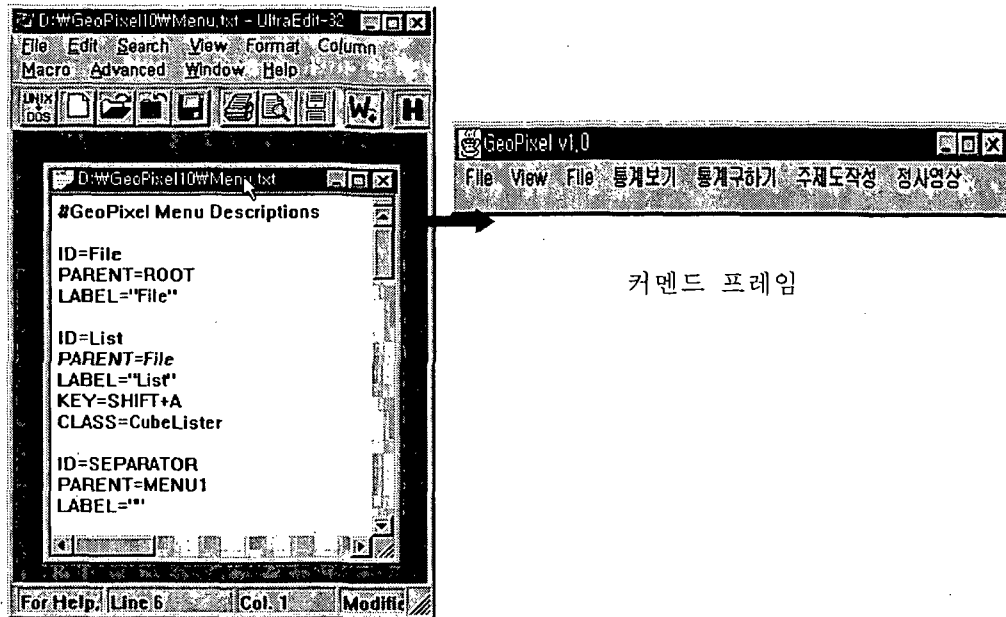
ID=MENU11
PARENT=MENU1
LABEL="Filter"

ID=MENU111
PARENT=MENU11
LABEL="User Defined"
KEY=SHIFT+U
CLASS=SpatialUDKernelFilter
```

Figure 5-25 Example of menu description

사용자가 소프트웨어의 실행을 종료한 후에도 메뉴와 모듈간의 연결 상태를 메뉴 전체를 이진 파일 형식으로 저장한 메뉴 상태 저장 파일이 만들었다. 본 연구에서는 이 파일을 객체의 직렬화(serialization)을 이용하여 구현하였다.

메뉴와 모듈의 동적 연결은 앞서 설명한 핵심어 탐색 엔진과 동일한 방법으로 Supervisor 클래스로 캐스팅하여 실행한다(Fig. 5-26).



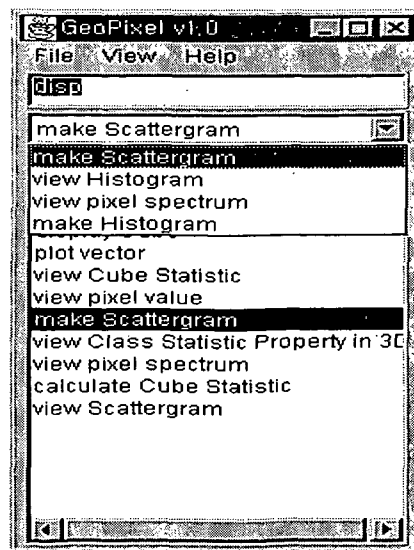
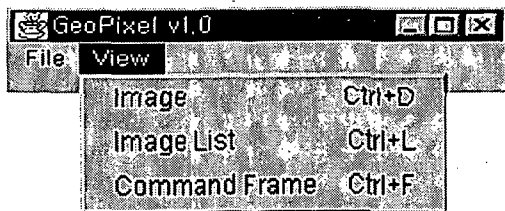
커맨드 프레임

Figure 5-26. Example of menu description file and generated frame

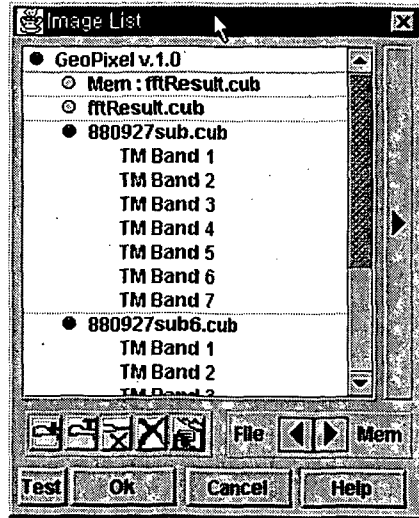
## 제 5 절 개발된 위성영상 처리 기능을 이용한 토지 이용 분류

위성 영상을 이용한 토지 이용 분류에 대한 GeoPixel의 응용 모듈 처리 과정을 요약하면 다음과 같다.

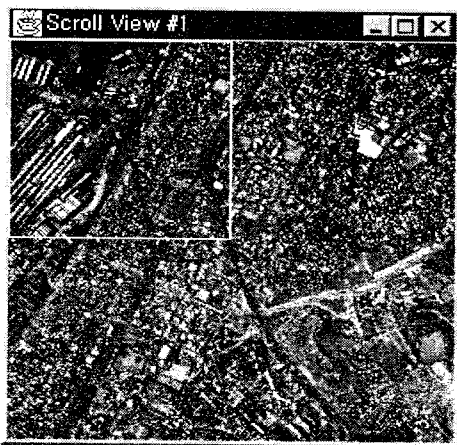
1. 시스템을 실행한다. 커맨드 프레임과 사용자가 정의한 메뉴 프레임이 화면에 출력되어 자료 읽기 모듈을 실행한다.



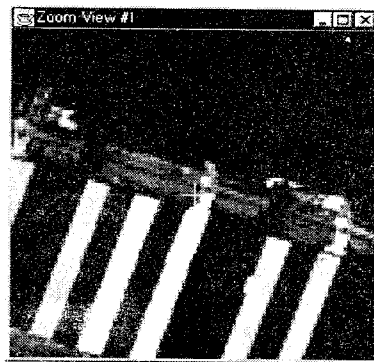
2. CD-ROM의 위성영상을 입력한다. 사용자가 구입한 위성 영상 CD-ROM을 바로 GeoPixel이 사용가능한 구조로 하드디스크에 읽어들이는 것이다. 이 때 구입한 영상의 헤더에 들어있는 정보를 해석하여 사용자에게 보여준다. 입력된 영상은 파일 관리자에 등록된다.



3. RGB 모드로 영상을 화면에 출력한다. CD-ROM에서 읽어들이는 파일을 RGB모드로 화면에 출력하여 제대로 변환과정이 수행되었는지 확인한다. 이 때 RGB의 서로 다른 조합을 통해 색상을 달리할 수 있다. 또한 시각적 분석을 돕기 위해 영상을 향상시킨다.



Scroll Window



Zoom Window

화면의 출력은 전체 영상을 축소하여 나타내는 Scroll window, 처리하고자 하는 지역을 1 : 1로 나타내는 Main window, Main window의 일부 영역을 확대 축소하는 Zoom window로 구성되어 있다. 사용자 대화형 영상처리나 영상 분석 기능은 주 원



도우상에서 pop up메뉴로 지원된다.



Main Window

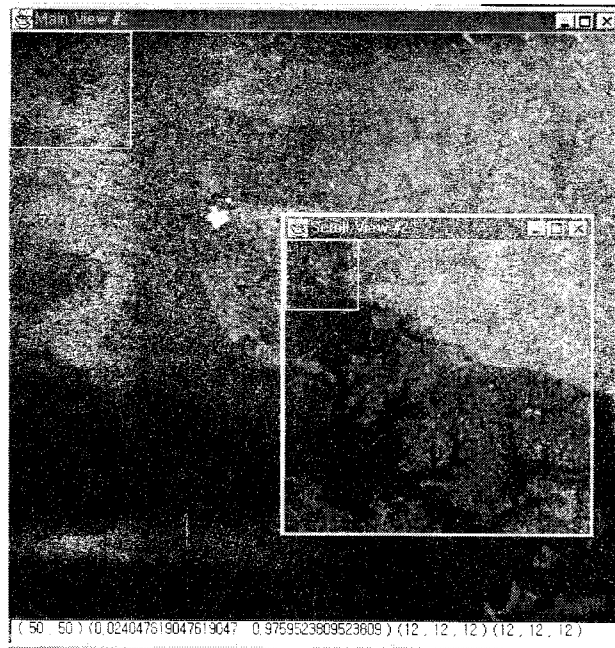
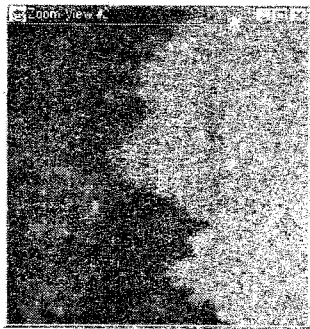
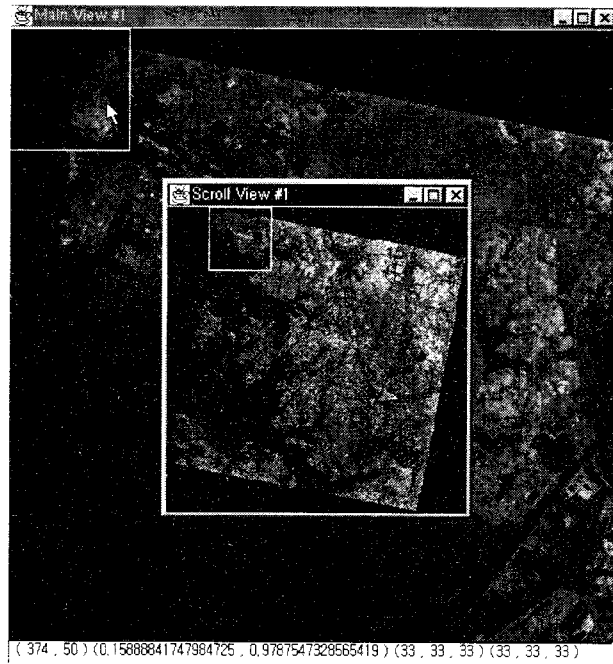
4. 시각적 분석을 돕기 위해 영상을 향상시킨다. 선형, 비선형 영상 향상을 수행하여 지상기준점을 가장 뚜렷이 볼 수 있는 영상을 찾아낸다. 이 때 사용자의 입력은 실시간으로 반영되어 영상을 향상시킨다.
5. 지상 기준점을 수집한다. 영상을 지도좌표에 맞추기 위한 기하보정을 수행하기 위해 영상 좌표를 입력받을 화면을 선택한다. 그 다음 Zoom 화면에서 지상기준점을 클릭한 후 해당 지도좌표를 입력한다. 선택한 지상기준점은 순서대로 화면에 십자 형태로 표시된다. 이 때 RMS 오차, 지도좌표에서 화면좌표로의 역변환 등을 실행할 수 있으므로, 찾고자 하는 지상 기준점에 일차적으로 접근할 수 있도록 지원한다. 최종적으로 수집한 지상기준점을 별도의 행렬 파일에 저장한다.

대상 영상



기준점

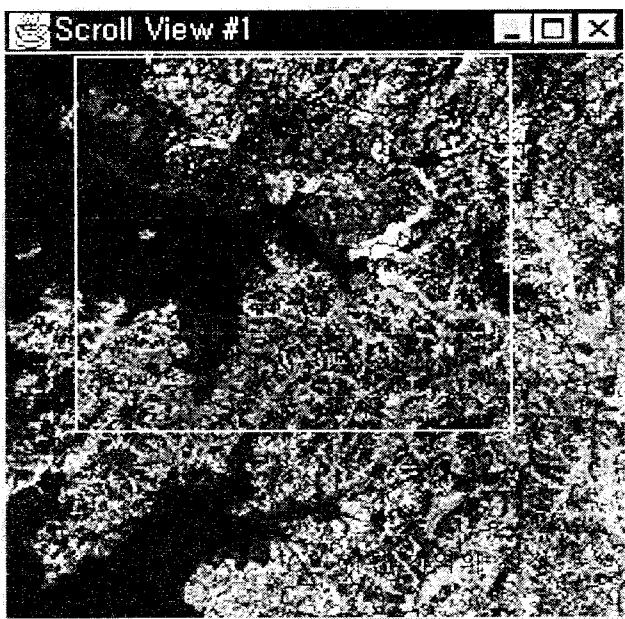
기준 영상



6. 기하 보정을 실행한다. 앞에서 저장한 행렬 파일을 불러 들여 기하 보정을 수행한다. 이 때 다항식의 차수, 자료 재배열 방법 등을 선택할 수 있다.

7. DXF 파일을 벡터 파일로 변환한다. 기하 보정 결과를 육안으로 확인하기 위해서 디지털라이저로 입력한 DXF 파일을 GeoPixel의 벡터 파일(GeoVector Format)로 변환한다.

8. 기하 보정된 영상 위에 벡터 파일을 중첩한다. 변환된 벡터 파일을 영상위에 중첩, 표출하여 기하보정 상태를 육안으로 확인한다. 이 때 벡터의 색상을 임의로 바꿀 수 있다.



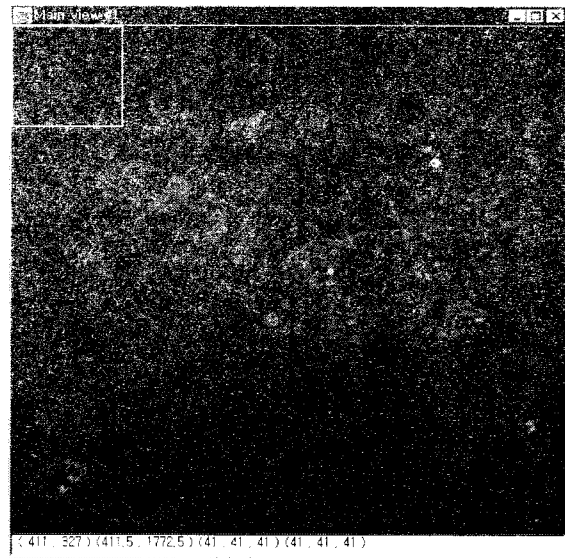
기하 보정된 영상과  
벡터의 중첩 표현

9. 출력 영상을 RGB 모드에서 향상시킨다. 영상 분류를 위한 훈련지역을 선정하기 위해 우선 기하보정된 위성영상을 향상시킨다.

10. 훈련지역을 선정한다. AOI 편집을 통해 훈련지역을 선정한다. 선정된 AOI는 클래스별로 서로 다른 파일에 저장한다.



11. 훈련지역에 대한 통계치를 구한다. 선정된 AOI 파일을 이용하여 선택한 지역에 대한 통계치를 구한다. 특히 최우도 분류법을 위해서는 공분산-분산 행렬의 계산이 중요하다.
12. 최우도 분류법을 이용하여 영상을 분류한다. 앞에서 구한 통계 파일을 이용하여 영상을 분류한다. 이 때 출력 영상의 화소값은 AOI 파일을 읽어들이는 순서와 같다.
13. 출력 영상을 B/W모드로 화면 출력한다. 출력 영상의 분류 상태를 살피기 위해서 흑백 모드로 영상을 읽어 들인다. 색상표를 편집하여 클래스별로 색상을 부여한다. 색상표를 편집하여 각 클래스에 적절한 색상을 부여한다.



여 백

## 제 6 장 3차원 지형 분석 S/W GeoPixel-Terrain 개발

경사도, 절토량 등 지형에 대한 분석을 수행하기 위하여는 대상 지역의 표고에 대한 정보가 필요하다. 이러한 지형에 대한 3차원적인 정보는 불규칙하게 분포하고 있는 관측점이나 스캐닝 또는 디지털라이징 과정에 의하여 얻어진 동고선을 이용하여 내삽하거나, 지도로부터 주어진 셀의 크기에 맞추어 해석된 지형의 표고를 사각 그리드의 형태로 저장하고 있는 DEM(Digital Elevation Model)을 많이 사용한다. 여기에서는 이러한 DEM을 입력원으로 하여 3차원 공간에서 지형분석 및 가시화를 위한 GeoPixel-Terrain S/W를 C++언어와 OpenGL을 이용하여 개발하였으며, 이는 전처리 과정을 수행하는 Modeller, 화면에 결과를 출력하는 Renderer, 지형 분석 기능을 수행하는 Analyser, 응용 모듈로서 경관 시뮬레이션 등 미리 주어진 경로를 따라 동화상으로 지형을 표현하는 Flight Simulator로 구성되어 있다.

### 제 1 절 3차원 지형 모델링

3차원 공간에서 DEM을 이용한 지형 분석을 위하여 모델링의 구성은 아래의 그림과 같이 원 자료를 그대로 활용하거나, DEM이 갖는 특징을 유지하면서 자료량을 줄일 수 있는 삼각망 구성법을 활용하여 이루어진다.

#### 1. 사각그리드에 의한 지형 모델링

사각그리드에 의한 방법은 표면 모델링의 초기에 사용되던 방법으로 사각그리드 형태로 수집된 데이터가 있을 때 많이 사용된다. 이 방법은 모든 자료가 동일한 크기의 격자 즉 단일 밴드의 래스터 영상과 같은 형태로 저장되어 있으므로 구현이 간단하고 평지 모델링에 적합하다. 사각그리드에 의한 지형 모델링 방법은 각각의 인접 셀들의 높이 값들을 서로 연결하여 실제 지형과 유사한 모형을 구성한다. 대개의 모델링은 각 인접한 4 점들을 연결하여 면을 구성하는 단순 폴리곤을 기초로한 모델링

(polygon-based modeling)을 수행하는 것이 대부분이다. Fig. 6-1은 수치고도데이터를 사각그리드에 의해 모델링한 지형의 모습이다.

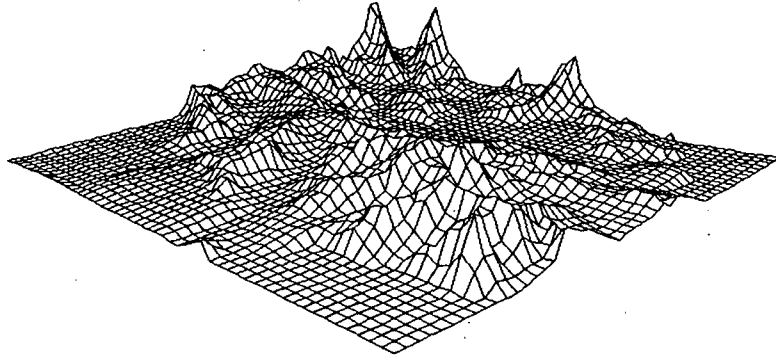


Figure 6-1 Terrain modelling using rectangular grid

## 2. 불규칙 삼각망(TIN)에 의한 지형 모델링

TIN에 의한 지형 모델링은 불규칙 관측점을 서로 삼각형으로 연결하거나 또는 DEM으로부터 지형적 특징을 갖는 의미점을 추출한 후 추출된 의미점들을 삼각형화 과정으로 이루어진다.

### 가. 지형 특징점 추출

실제 지형은 어떠한 수식이나 디지털 입력기로도 만들어낼 수 없는 매우 복잡한 모습을 하고 있다. 따라서 이러한 복잡한 모습을 표현하고 있는 방대한 수치고도데이터로부터 의미점 추출을 통해 데이터 양을 감소시킴으로써 적은 렌더링 시간으로 실시간적인 시각화를 가능하게 하고, 불필요한 데이터 처리를 줄일 수 있도록 한다. 의미점이란 DEM 데이터 중 지형적으로 의미있는 위치에 대한 점을 말하며 산 꼭대기(peak), 움덩이(pit), 고갯길(pass), 능선(ridge-line), 계곡(ravine-line), 급경사(break), 비탈(slope), 평지(flat)의 8가지로 분류한다(Peucker and Douglas, 1975). 의미점 추출은 지형에서 획득된 DEM 데이터의 각 점과 이웃점들과의 관계로 정의된다. 즉, 한 점을 중심으로 그 점을 둘러싸고 있는 8방향의 각 이웃점들에 대해 시계 방향이나 반시

계 방향으로 돌며 고도차(이웃점에서 기준점을 뺀 값)를 계산하여 의미점을 추출할 수 있다. 기준점의 고도를 중심으로 각 이웃점 고도와의 계산 결과는 양의 값, 음의 값 또는 0을 가지게 된다. 본 연구에서 사용된 각 의미점의 특성은 Table 6-1과 같다. 이와 같은 고도차를 연결하면 Fig. 6-2와 같은 그래프를 생성할 수 있다. 따라서 이와 같은 각 지형의 특징을 이용하여 사각그리드의 DEM으로부터 특징점들을 추출할 수 있다(Fig. 6-3).

Table 6-1 Geometrical properties of terrain specific points.

의미점	특 성
산꼭대기(peak)	주위의 점보다 상대적으로 높음
웅덩이(pit)	주위의 점보다 상대적으로 낮음
고갯길(pass)	주위의 점들이 한쪽방향으로 차례로 높고 낮음을 반복.
능선(ridge-line)	주위의 점 중에서 극히 일부분만이 높고 나머지는 모두 낮음
계곡(ravine-line)	능선과 반대의 성질로 극히 일부분만이 낮고 나머지는 높음
급경사(break)	주위의 점 대부분이 현재의 점보다 높음
비탈(slope)	주위의 점 중 반은 높고 나머지는 낮음
평지(flat)	주위의 점과 거의 차이가 없음

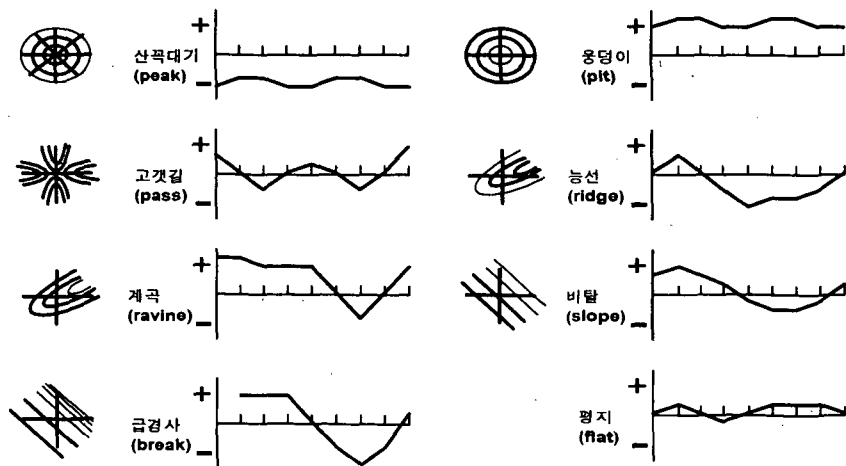


Figure 6-2 Graphical properties of terrain specific points



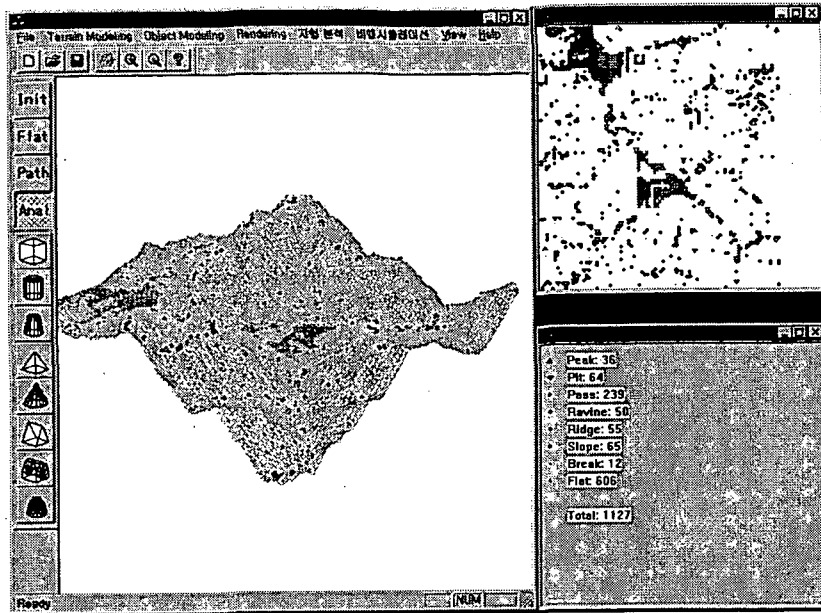


Figure 6-3 Extracted terrain specific points from DEM

#### 나. TIN의 생성

전술한 DEM 데이터에서 불규칙하게 추출된 의미 데이터는 보르노이 다이어그램에 의한 삼각형화 방법과 래디얼 스위프에 의한 방법으로 TIN을 구성할 수 있다.

● 보르노이 다이어그램(Voronoi Diagram)은 1908년 러시아의 수학자 Voronoi에 의해 연구된 이후, Dirichlet 영역, Thiessen 다각형, Wigner-Seitz 셀 등으로 명명되어 왔으며 이를 이용하여 공간의 분할을 목적으로 보다 효율적으로 수행하기 위한 연구가 계속되고 있다. 보르노이 다이어그램은 면적을 기초로 두고 특정점의 영향 영역에 대한 기하학적 관계를 설정하기 위한 것이다. 이는 각 점들을 연결하는 직선의 중심을 서로 연결하여 구성된 영역의 중심점 P가 포함된 영역에 속한 점들은 다른 어떤 점들보다 점 P에 가까움을 이용한 것으로 이러한 조건을 만족하도록 설정된 각각의 영역을 보르노이 다각형이라 하며 반드시 하나의 대표점을 가진다. 또한, 두 영역의 경계선 상의 점들은 인접한 두 영역의

대표점들과 거리가 동일하다. 보르노이 다이어그램에 의한 삼각형화는 인접한 보르노이 다각형의 대표점들을 직선으로 연결한 TIN 즉, 보르노이 다각형의 듀얼 그래프(dual graph)에 의해 이루어진다. 지형 모델링 분야에서 이러한 보르노이 다이어그램에 의한 삼각형화는 보르노이 다이어그램의 특성에 따라 최적의 삼각형화 방법으로 각광 받고 있으며, 이에 대한 구현 방법도 다양하게 제시되었다. 여기서 각 3개나 그이상의 이분선이 만나는 점을 보르노이 점(Voronoi Point)라 하고 보르노이 다각형의 두 개의 보르노이 점사이의 이분선을 보르노이 선분(Voronoi edge)라고한다. 이러한 보르노이 선분을 공유하는 점들을 연결한 보르노이 다이어그램의 듀얼 그래프를 델러니 삼각형(Delaunay Triangulation)이라 한다(Fig. 6-4).

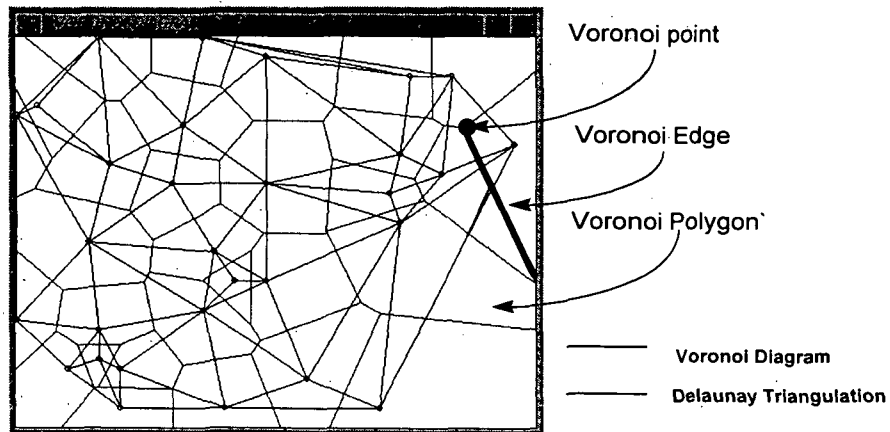


Figure 6-4 Voronoi Diagram and Delaunay Triangulation

● **라디얼 스위프에 의한 TIN의 생성**은 추출된 의미데이터로부터 라디얼 스위프 알고리즘에 의한 TIN생성은 다음과 같이 2단계로 이루어진다.

- ① 초기 삼각망 구성 : 의미점들 중에서 중심점(Centroid)을 찾아내어 그점과 가까이 있는 점들을 방향, 거리와 경사도 순으로 정렬한다. 중심점에서 정렬된 점으로 방사(Radiating)를 시작하여 방사된 현재 점의 방향이 이전 점의 방향

과 같지 않으면 중심점, 현재 점과 이전 점으로 삼각형을 구성한다. 같은 방향을 가질 경우, 현재 점을 스택에 저장 후 다른 방향을 가진 점이 발견되면 스택에 저장한 점을 꺼내 삼각형을 구성한다. 이와 같은 방법으로 모든 점들에 대해 삼각형을 만들어간다. 이러한 과정(Initial Radial Sweep)을 수행한 후 만들어진 삼각망은 굴곡(Concavity)을 가지고 있어 굴곡을 가진 부분을 채워야만 한다. 이 과정은 외곽 선을 갖고 있는 링크드 리스트에 의해 수행된다. 링크드 리스트에서 차례로 세 점이 안쪽으로 각을 이루고 있는지를 검사한다. 안쪽으로 각을 이루고 있을 경우 세 점으로 삼각형을 구성하고 두번째 점은 링크드 리스트에서 제거한다. 안쪽으로 각을 이루고 있는 것이 없을 때까지 이러한 과정을 반복한다.

- ② 삼각형의 보정 : 초기 삼각망 단계에서 구성된 TIN의 모양은 예각이 포함된 삼각형의 형태를 갖는다. 이러한 삼각형 모양의 보정은 이웃하는 두 개의 삼각형으로 이루어진 사각형의 두 대각선의 길이를 비교함으로써 행해진다. Fig. 6-5와 같이 삼각형  $\triangle ABC$ 와 삼각형  $\triangle BCD$ 로 구성된 사각형은 두 개의 대각선  $\overline{BC}$  (공유 선분)와  $\overline{AD}$  (고유선분)을 갖는다. 이 두 대각선의 길이를 비교하여 공유선분  $\overline{BC}$ 의 길이가 고유선분  $\overline{AD}$ 의 길이보다 길다면 삼각형의 모양을 보정한다. 즉, 삼각형  $\triangle ABC$ 와 삼각형  $\triangle BCD$ 를 삼각형  $\triangle ABD$ 와 삼각형  $\triangle ACD$ 로 구성하여 준다. 삼각형 모양의 보정이 일어나지 않을 때까지 위의 과정을 반복한다. 이 전과정을 그림으로 나타내면 Fig. 6-6과 같다.

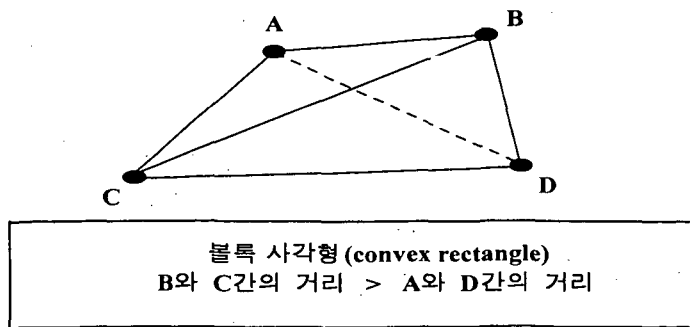
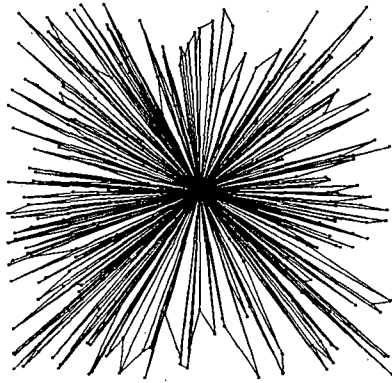
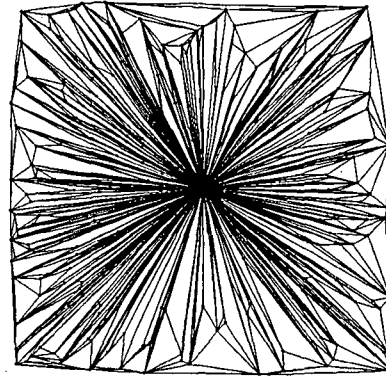


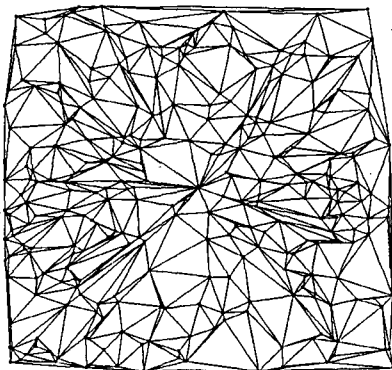
Figure 6-5 Correction of Triangle



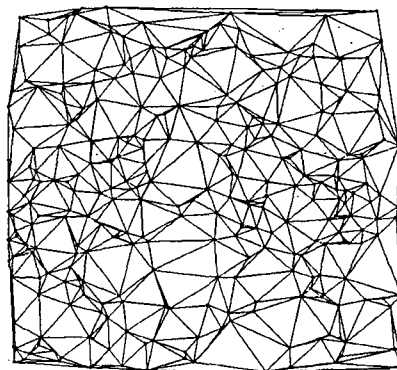
(a) 초기 래디얼 스위프  
(Initial Radial Sweep)



(b) 굴곡(Concavity)을 채운 후



(c) 첫 번째 삼각형 모양의  
보정 단계 후



(d) 모든 삼각형 모양의  
보정을 마친 후

Figure 6-6 General process of radial sweep triangulation

다. 델러니 삼각형화와 래디얼 스위프 알고리즘에 의한 삼각형화의 비교

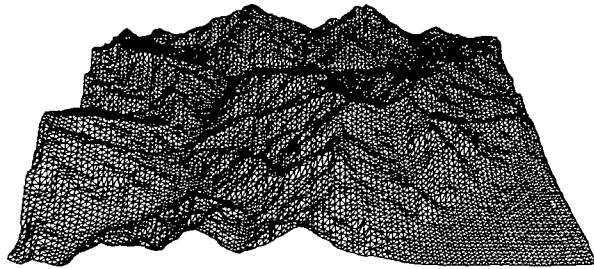
본 연구에서 불규칙 삼각망을 형성하는 알고리즘으로 보로노이 다이어그램에 의한 방법과 래디얼 스위프 알고리즘에 의한 방법을 사용하였다. 이 두 방법에 의해 불규칙 삼각망을 형성할 때 만들어진 불규칙 삼각망의 형태가 다르게 나타난다. 그 차이점에

대해 알아보면 다음과 같다. 보로노이 다이어그램에 의해 생성된 불규칙 삼각망은 보르노이 다이어그램의 특성에 따라 최적의 삼각형화 방법으로 삼각형화된다. 최적의 삼각형에 대한 정의는 다음과 같다.

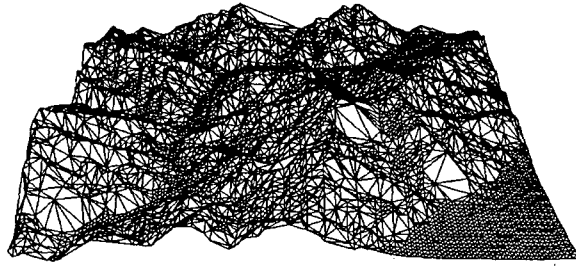
$\triangle ABC$ 에서  $\left| \frac{\pi}{3} - \angle A \right| + \left| \frac{\pi}{3} - \angle B \right| + \left| \frac{\pi}{3} - \angle C \right|$ 의 값이 0에 근접할수록 최적의 삼각형이다.

반면, 래디얼 스위프 알고리즘에 의해 생성된 불규칙 삼각망은 두 개의 삼각형에 의해 구성된 사각형의 두 대각선중 길이가 짧은 대각선을 선택하여 삼각형화된다. 이러한 차이점으로 인하여 생성된 불규칙 삼각망의 모습이 다르게 나타난다. 실질적인 지형오차분석을 위해서는 높이 값을 포함한 3차원 좌표 값을 이용하여 불규칙 삼각망을 생성하여야한다.

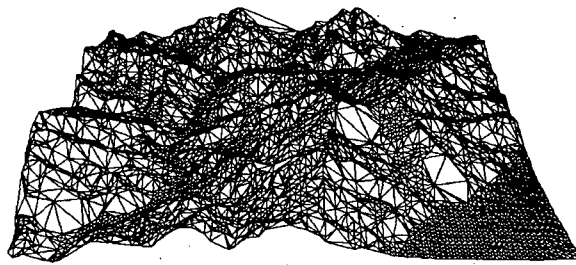
Fig. 6-7은 사각그리드에 의해 모델링된 지형과 두가지 방법(Voronoi Diagram에 의한 방법, Radial Sweep Algorithm에 의한 방법)에 의해 만들어진 불규칙 삼각망에 의해 표현된 지형의 선구조 형상 모습이고 Fig. 6-8은 실행화면이다.



(a) 사각그리드



(b) Delaunay Triangulation



(c) 래디얼 스윙프 알고리즘

Figure 6-7 Comparison of Delaunay and Radial sweep triangulation

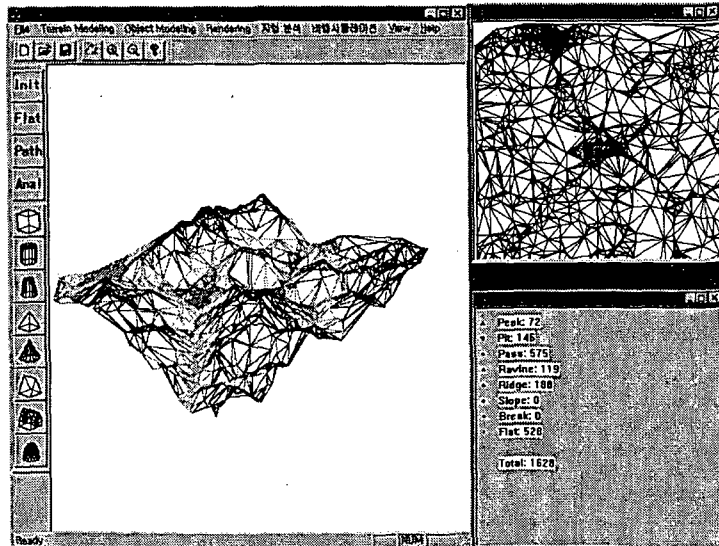


Figure 6-8 Result of delaunay triangulation

## 제 2 절 3차원 가시화

### 1. 3차원 Viewing

관찰자의 이동에 따른 시점의 변환은 Fig. 6-9와 같이 지형이 실세계 좌표 상에 존재한다고 할 때 카메라를 세가지 방향 즉, X축 Y축 그리고 Z축으로 이동시키면서 카메라의 망막에 나타나는 상을 화면 상에 표현하는 것이다. 3차원으로 표현된 지형을 관측자의 위치변화에 따른 모든 각도에서 관찰한 지형을 평행투사(parallel projection) 기법을 사용 화면에 출력하도록 하였다

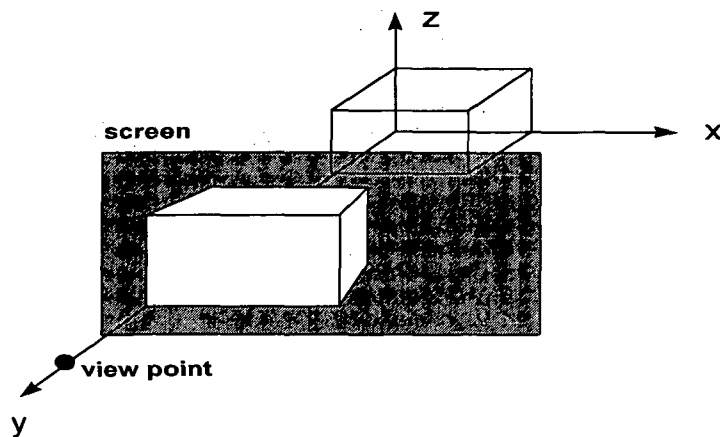


Figure 6-9 View point transformation and projection

### 2. 투시 투사에 의한 Viewing

원근 투영은 관찰자를 중심으로 모이는 투영광선을 발생시켜 매핑시키는 방법을 사용한다. 물체와 관찰자와의 거리가 먼 물체일수록 작게 보이는 왜곡현상이 발생하는데 이러한 왜곡 현상은 실제 사람이 눈이 느끼는 원근감을 표현할 수 있으므로 시뮬레이션 등의 응용 프로그램에서 주로 쓰이는 투영 방법이다. 원근 투영에서는 실제 데이터와 투영되는 영상이 거리에 따라 크기가 변하므로 다음과 같은 식을 사용하여 시각평

면에 투영되는 영상의 좌표를 계산하게 된다.  $z_{prp}$ 는 원근 투영의 중심점의  $z$  좌표 값이며,  $z_{vp}$ 는 시각 평면의  $z$  좌표 값이다. 실제 데이터  $(x, y, z)$ 값을 시각평면에 투영하여 평면내의 좌표  $(x_p, y_p)$ 를 계산한다. Fig. 6-10, 6-11은 개발된 모듈의 실행 결과를 관찰자의 시각 방향에 따라 다르게 나타낸 것이다.

$$x_p = x \left( \frac{z_{prp} - z_{vp}}{z - z_{prp}} \right) = x \left( \frac{d_p}{z - z_{prp}} \right)$$

$$y_p = y \left( \frac{z_{prp} - z_{vp}}{z - z_{prp}} \right) = y \left( \frac{d_p}{z - z_{prp}} \right)$$

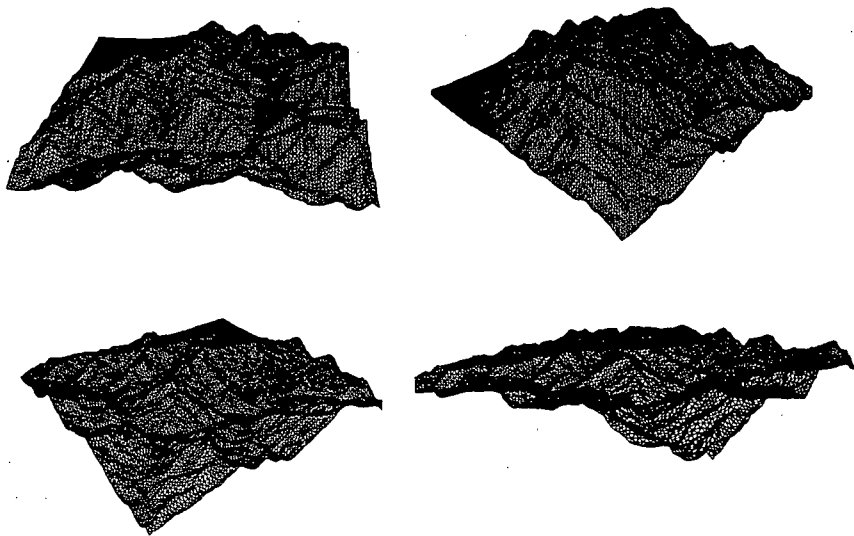


Figure 6-10 Terrain representation in different view point



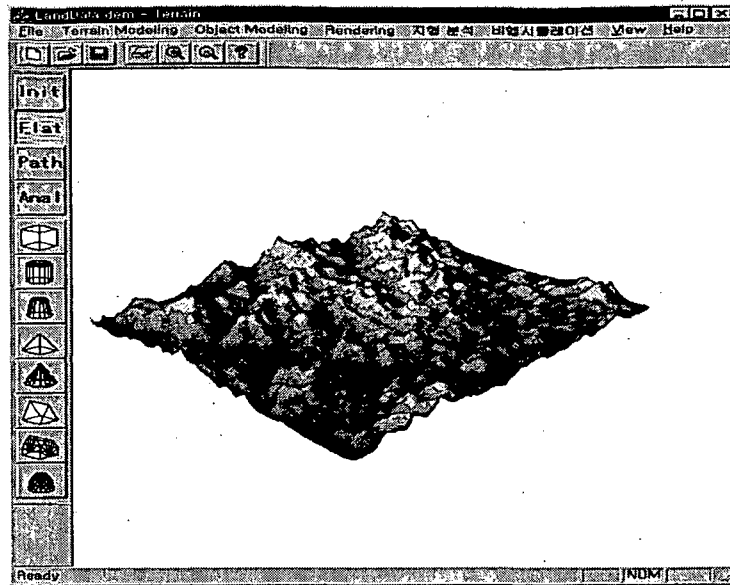


Figure 6-11 Terrain representation with pseudo shading

### 제 3 절 지형분석

#### 1. 경사각 및 방향 계산

경사각은 지형에서 한 점을 선택했을 경우 그 점을 중심으로 주위를 조사하여 고도차를 이용해 주위 지형과의 각도를 구하는 것이다. 그리고 방향은 주위점 중에서 가장 경사가 급한, 즉 각도가 가장 큰 점의 방향을 나타내는 것이다. 예를 들어, 한 방향의 경사각을 구하는 방법은, 한 점을 선택했을 때 경사각은 선택된 점(a)을 중심으로 이웃한 점(b)과 a에서 z좌표를 점 b의 z좌표와 같이한 점(c)을 연결한 삼각형에서 값을 구한다. 이렇게 구성된 삼각형에서 a,b를 연결한 벡터를 p라 하고 b와 c를 연결한 벡터를 q라 했을 때 p와 q에서 다음과 같은 식을 얻을수 있다.

$$p \cdot q = |p||q| \cos \theta$$

이 식을  $\cos \theta$ 에 관하여 정리하면,  $\cos \theta = \frac{p \cdot q}{|p||q|}$  여기서 각도를 구하려면  $\cos \theta$ 의 값을 x라 했을 때  $\arccos(x)$ 를 하면  $\theta$ 를 알아낼 수 있다. 이렇게 경사도를 8방향을 처음부터 차례로 조사해 가면서 가장 큰 값을 저장하고 방향은 45도 각도로 증가하므로 그때의 각도를 보고 방향을 설정해준다.

## 2. 절토량, 성토량, 저수량 계산

### 가. 절토량

절토량이란 기존의 지형 정보에서 건물을 세우거나 또는 다른 이유로 인해 없어진 지형의 양을 말한다. 본 보고서에서 절토량을 계산하기 위해서 사용할 알고리즘은 최대한 근사치를 얻기위해서 기본 3차원 단위(정육면체를 사용함)를 설정해 두고 그 단위를 이용하여 사용된 량을 계산하는 방법과 적분에 의한 방법을 사용하였다. 기본 3차원 단위를 사용하는 것은,  $10 \times 10 \times 10$  정육면체를 기본 단위로 설정해 두고 절토된 위치의 높이를 DEM 데이터로부터 얻어서 기본 단위가 들어갈 수 있는 양을 계산한다 (Fig. 6-12).

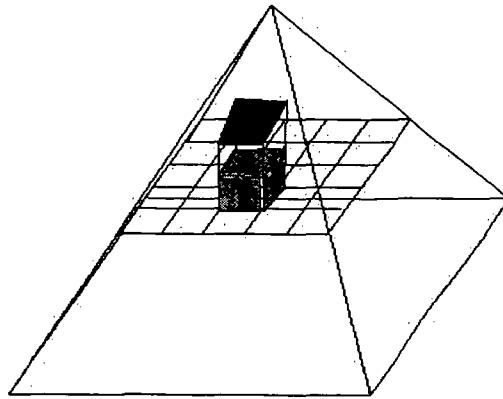


Figure 6-12 Calculation of volume of terrain cutting using unit-cell area

예를 들어, 그림에서 표시된 부분의 높이는 50이고 절삭되기전 원래 지형의 높이는

105라고 하고 기본 단위를  $10 \times 10 \times 10$ 을 사용한다고 하면 표시된 부분의 절토량은  $10 \times 10 \times 10 \times (105 - 50) = 55,000$ 가 된다. 이런식으로 절삭된 모든 부분을 모으게 되면 전체 절토량이 계산된다.

다음은 적분을 이용한 방법인데, 사용자가 선택한 높이를  $a$ 라 했을 때, 점 A, B 그리고  $z_1, z_2$ 대신에  $z$ 값을  $a$ 로 둔 점 A', B'로 하나의 평면(P)이 생긴다. 그 평면의 면적을 구한뒤에 점 D와 점 E에 관해서도 똑같은 방법을 적용하면 평면(Q)의 면적을 구할수 있다. 두 면적의 평균(L)을 P와 Q사이의 면적으로 생각하여 A와 D사이의 거리를 곱해주면 한 입방에 대한 면적이 나오게 된다. 이런 방법으로 선택된 영역 내의 점들에게 적용시켜 구해진 면적들을 더하면 선택된 곳의 절토량이 나오게 된다. 적분을 적용시킬 때 지형의 경우가 4가지가 나오게 되는데 그 지형에 따라 계산하는 방식이 각각 틀리게 된다.

먼저 4가지 지형에 대하여 살펴 보면, 선택된 높이에 따라서 처음점의 고도가 높이보다 높고, 끝점의 고도가 높이보다 낮은 경우, 처음점의 고도가 높이보다 낮고, 끝점의 고도가 높이보다 높은 경우, 그리고 2점의 고도가 모두 높이보다 높은 경우, 2점의 고도가 모두 높이보다 낮은 경우를 생각할 수 있다. 이 경우에 2점의 고도가 모두 높이보다 낮은 경우는 절토할 면적이 없으므로 제외 시키고, 1번째와 2번째 경우는 두 점을 잇는 선분과 높이의 선분과 만나는 점을 구하여 그 점을 적분에 적용시킨다. 3가지 경우에 처음점을  $A(x_0, y_0, z_0)$ , 끝점을  $B(x_1, y_1, z_1)$ , 그리고 교차하는 점을  $A'(x_2, y_2, z_2)$ 라고 한다면 두점을 지나는 직선의 방정식은 다음과 같다.

$$y = \frac{z_1 - z_0}{x_1 - x_0} x + z_0$$

위 식을 적분하면,

$$\int_{x_0}^{x_1} \frac{z_1 - z_0}{2(x_1 - x_0)} x^2 + z_0 x + d$$

위 식에서 1번째 경우는 구간은  $x_0$ 에서  $x_2$ 까지 2번째 경우는  $x_2$ 에서  $x_1$ 까지 3번째 경우는  $x_0$   $x_1$ 까지로 주면 각각의 경우에 대한 면적을 구할수 있다(Fig. 6-13).

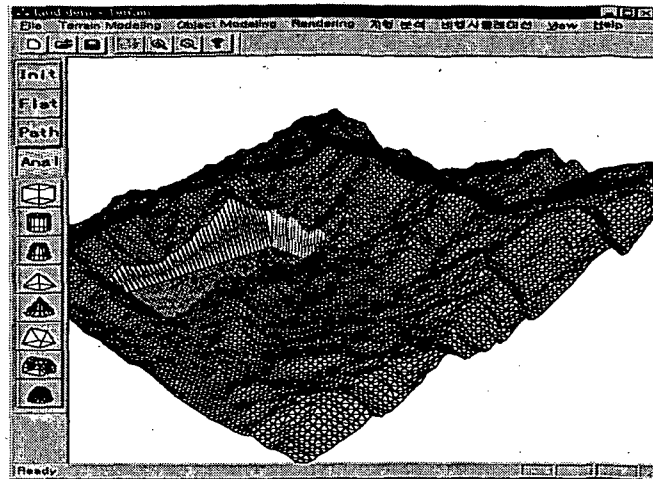


Figure 6-13 volume analysis of terrain cutting

#### 나. 성토량

성토량은 지형에서 빈공간을 채우는 것을 말한다. 즉, 성토량을 구하는 것도 절토량과 마찬가지로 voxel을 이용한 방법과 적분을 이용한 방법으로 계산을 할수 있다. 사용자가 높이를 입력했을 경우 그 높이 까지의 면적에 선택된 부분의 실질적인 지형의 면적을 빼주면 성토량이 나오게 된다. 성토량을 계산할때도 절토량 계산과 같은 경우가 나오게 된다. 이러한 경우에 절토량과는 다른 부분의 면적을 구하여야 한다. 예를 들어 아래와 같은 경우 절토량은 A지역의 면적을 구하나 성토량은 B지역의 면적을 구하여 모두 더해주는 연산을 한다. 사용하는 식은 절토량의 식과 같으나 구간만을 다르게 준다.

#### 다. 저수량

저수량은 지형에서 어떤 지역을 막았을 경우 그 곳에 모일수 있는 물의 양을 계산하는 것이다. 저수량도 voxel이나 적분에 의해 계산할수 있다. 저수량은 성토량과 같은 개념을 사용하여 계산을 하게 된다. 아래의 그림과 같은 지형이 선택되었다고 하면 높이까지 빈 공간을 물로 채워 넣는 계산을 해주면 그 지형의 저수량이 나오게 된다. 여

기서 물론 선택된 지형의 높이를 고려하여 방수가 되는 것을 생각해야 하지만 지금 단계에서는 그것을 계산에 넣지는 않았다. 그리고 댐을 설치 했을 경우에 댐의 높이보다는 저수량이 적게 계산이 된다.

### 3. 집수역 분석

집수역 분석은 물의 흐름을 조사하여 수원을 알아내거나, 그 포인트에서 물의 양을 알아내는 방법이다. 물의 양을 알아내는 방법에는 2가지 알고리즘이 사용될수 있다. 첫 번째는 한 포인트에서 물이 경사가 가장 큰 방향, 즉 고도차가 가장 심한 방향으로 물이 흐른다고 가정을 하여 계산을 하는 방법이고, 두 번째는 각 방향의 고도차의 비율을 구해서 그 비율에 따라서 방향마다 물의 양을 일정량 분배하는 방법이다. 여기에서는 2번째 방법을 이용하여 워터쉐이드를 구현하였다. 각 점에서의 물의 양을 구하는 알고리즘을 요약하면 다음과 같다.

1. 중심점을 설정하고 8방향의 고도를 조사
2. 중심점의 고도와 비교하여 고도가 높은 점들은 제거
3. 2번에서 남은 점들의 고도를 더하며, 각점의 고도를 나눔
4. 3번식에서 나온 값을 중심점의 물의양과 곱함.
5. 전 포인트에 대해서 1번부터 반복수행.

이러한 집수역 분석은 임의의 지역에서의 물의 양을 알 수 있기 때문에 저수량을 구할 때 같이 응용하면 좋은 결과를 얻을 수 있을 것이다. 예를 들어 댐의 설치와 같은 것은 물의 흐름이 빈번한 곳에 설치하는 것이 좋은 선택이 되기 때문에 여기서 구한 물의 양을 보고 그 지역에 댐 설치를 하였을 때 어떤 결과를 얻을 수 있는지를 미리 예측할 수 있다. 이러한 집수역의 알고리즘은 만약 중심점이 웅덩이가 된다면 평지일경우, 더 이상 물이 흐르지 않게 되는 결과를 초래하게 되므로 분석 이전에 DEM에 대한 전처리 과정을 통하여 Depressionless DEM으로 만들어 주어야 한다. 그리고 집수역을 구할 때 지형의 특성에 따른 물의 흡수도나, 증발량과 같은 환경요건을 계산하면 보다 정확한 결과를 얻을 수 있다.

#### 4. 가시권 분석

가시권 분석은 시뮬레이션의 속도 향상 또는 임의의 지역에서 관찰할 수 있는 최대의 가시권을 분석하는 것이다. 그리고 가시권 분석 방법은 관찰자의 위치와 지형모델의 종류에 따라 구분할 수 있다. 관찰차 위치에 따라서는 지형 위에 있는 경우와 지형 밖에 위치하는 경우로 나눌 수 있고, 지형 모델의 종류에 따라서는 고도 모델(DEM)과 불규칙 삼각망(TIN)에 따른 가시권 분석방법으로 나누어 진다.

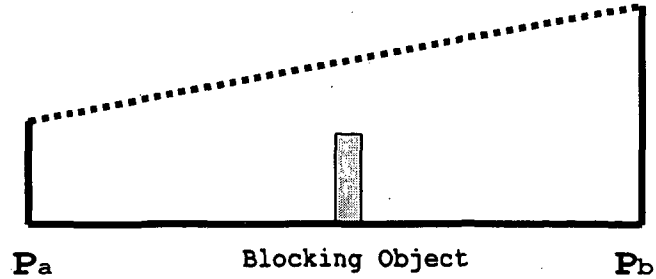
먼저 시뮬레이션의 속도 향상 방법은 지형의 영상을 표현할 때 보이지 않는 부분을 제거하고 보이는 부분만을 표현하거나, 가까운 물체에 대해서는 상세도를 높여주고 멀리 보이는 물체에 대해서는 간략히 표현하여 물체에 대한 데이터의 양을 줄임으로써 이루어진다. 최대의 가시권을 얻는 방법은 지형에서 최대로 보이는 위치를 파악하는 것이다. 예를 들어 산불 감시소와 같은 것을 지형에 설치하고자 할 때, 최대의 가시영역을 가지는 곳에 설치하여, 산불 감시소의 숫자를 줄이면서 모든 곳을 효율적으로 관리할 수 있는 방법을 제시한다. 가시권 분석의 응용분야는 네비게이션(Navigation), 지형 탐사(terrain exploration), 군사 감시(Military surveillance), 산불 감시소, 레디오 전 송국, 전망탑 설치와 같은 분야에 응용된다.

가시권 분석을 사용하는 방법을 크게 나누어 보면 전체의 지형을 보기 위해 몇 개의 viewpoint가 필요한지를 구하는 방법과 정해진 viewpoint의 개수로 최대의 가시권을 구하는 방법으로 나눌 수 있다. 가시권 분석의 종류를 보면 점 가시도(Point Visibility), 선 가시도(Line Visibility), 지역 가시도(Region Visibility)로 나눌 수 있다

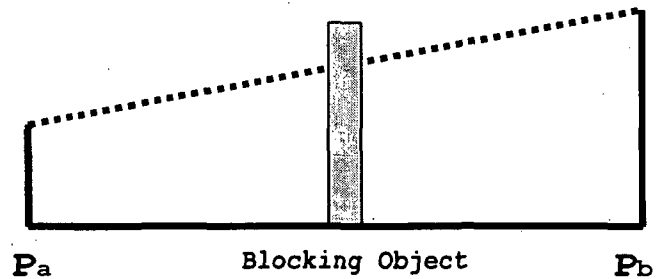
##### 가. 점 가시도 분석

점의 가시도 분석은 두 개의 점을 연결하는 선분이 두 개의 점 이외에 다른 물체와의 교점이 있는지 없는지를 조사하여 교점이 없는 경우 상호 가시(inter-visible)하다고 할 수 있고 교점이 있는 경우 상호 가시(inter-visible)하지 않다고 한다. 가시도 함수를 보면 Fig. 6-14a)와 같이  $P_a$ 와  $P_b$ 가 서로 보이는 경우  $V_{ab}=1$ 이라고 하고 Fig. 6-14b)와 같이  $P_a$ 와  $P_b$ 가 다른 물체(Blocking Object)에 의해 가리는 경우  $V_{ab}=0$ 이

라고 정의한다.



(a)  $P_a$  is visible from  $P_b$



(b)  $P_a$  is invisible from  $P_b$

Figure 6-14 Relationship between Intervisibility and Blocking Object

#### 나. 선 가시도 분석

선분에 대한 가시도 분석 방법은 two-block 테스트에 의해 수행된다. 선분에 대한 가시권 분석 방법은 다음과 같다. 먼저 각 선분의 끝점과 시점을 연결한다. 분석할 선분이 two-block되어 있는지를 조사한다. two-block되었다는 의미는 분석할 선분이 다른 선분에 의해 막혀 있다는 것을 말한다. two-block이 되어 있으면 two-block된 선분에 two-block 테스트를 한다. two-block 테스트의 종류는 partially two blocked와 totally two blocked 테스트가 있다. 각 각에 대해 알아보면 다음과 같다. Fig. 6-15의 왼쪽 부분과 같이 선분  $\overline{E}_j$  전체가 선분  $\overline{E}_i$ 를 가리는 경우를 totally two blocked 되었

다고 하고 그림 2-5의 오른쪽 부분과 같이 선분  $\overline{E_i}$ 의 일부분이 선분  $\overline{E_j}$ 를 가리는 경우를 partially two blocked 되었다고 한다. 이러한 방법으로 선분의 보이는 부분과 보이지 않는 부분을 분석하여 가시권 분석을 한다.

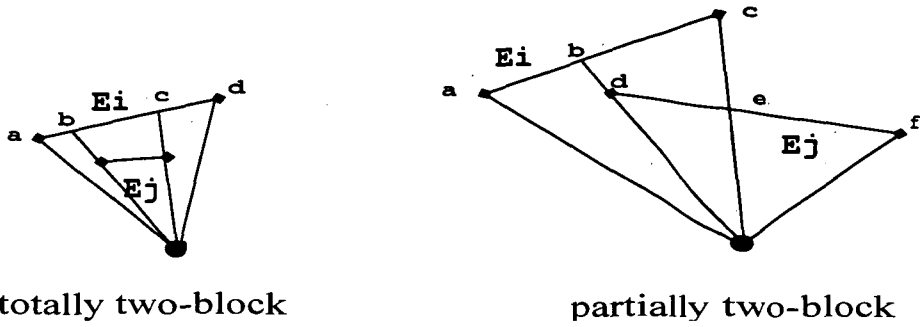


Figure 6-15 Two-block test for line visibility

다. 최대 가시권

앞에서 설명한 알고리즘은 한 viewpoint로부터 가시의 범위가 되는 영역의 point를 찾아 내는 알고리즘이다. 위에서 얻어진 point로 가시의 면적을 구하고, 사용자가 정해 준 viewpoint의 수로 최대의 가시권을 얻는 알고리즘이 필요하게 된다. 이와 같이 viewpoint의 개수로 최대의 영역을 구하는 최적화 된 알고리즘은 아직 개발되지 못하였으며, 여기에선, s 일반적인 3가지 algorithm을 사용하여 구현하였다.

● Greed Add algorithm : 이 알고리즘은 미리 정해진 viewpoint로부터 최소의 viewpoint를 사용하여 최대의 영역을 구하는 알고리즘 중에 하나이다. 이 알고리즘은 3가지 알고리즘 중에 속도는 가장 빠르지만 구하는 영역의 정확성에서는 가장 떨어진다. 시스템의 사양이 좋지 않고 빠른 계산을 요구할 때 이 알고리즘을 사용한다. Greed Add algorithm은 아래와 같이 이루어진다.

1. solution set을 empty로 만든다.
2. Visible Triangles의 가장 큰 영역을 가지는 viewpoint를 찾는다.
3. 첫 번째 viewpoint의 영역에 속하지 않는 viewpoint중에서 가장 큰 영역을 가지는 viewpoint를 찾는다.



4. 사용자가 입력한 viewpoint의 개수가 되거나 전 영역이 cover될 때 까지 반복 수행한다

● Greedy add with swap algorithm : 이 알고리즘은 Greedy add algorithm의 단점을 보완하기 위해서 나온 알고리즘이다. Greedy add algorithm은 한번 solution set에 포함되면 다시는 수정을 할 수 없다는 단점을 가지고 있다. 위와 같은 문제점 때문에 영역을 구할 때 정확성이 떨어지게 되는데, Greedy add with swap algorithm의 모든 과정은 Greedy add algorithm과 동일하지만 한 solution을 찾고 난 후에 해가 아닌 viewpoint의 집합으로부터 최적의 해가 존재하는 지를 검사하여, 새로운 해가 있게 된다면 그 viewpoint를 해로 선택하는 방법을 취한다. 이 알고리즘은 3가지 알고리즘 중에서 가장 정확한 영역을 구할 수 있지만 계산량의 증가로 인한 시간상의 문제가 크게 된다는 단점을 가지게 된다.

● Stingy drop algorithm : 앞에서 설명한 2가지 알고리즘과 상반되는 개념의 알고리즘이다. 위의 2가지 알고리즘은 모두 solution set을 empty로 설정했지만, Stingy drop algorithm은 solution set을 미리 정해진 모든 viewpoint로 설정하고 최적의 viewpoint를 찾아가는 방법이다.

1. solution set을 모든 viewpoint로 초기화한다.
2. viewpoint중에서 가시영역이 최소인 것을 drop 한다.
3. invisible point를 drop 한다.
4. 위의 과정을 사용자가 원하는 viewpoint가 되거나 전 영역이 cover될때까지 반복한다.

이상과 같은 3가지 알고리즘을 사용하여 사용자가 원하는 viewpoint개수로 최대의 영역을 구하거나, 최대의 영역을 얻을 수 있는 viewpoint의 개수를 구하는 작업을 수행한다. 위와 같은 알고리즘들이 optimize 될 수 없는 이유는 처음에 viewpoint를 미리 정해줘야 하는데, 처음에 설정되는 viewpoint들 중에서 최대의 영역을 가지는 것이 존재하지 않을 수도 있기 때문이다.

## 5. 최단거리 분석

최단거리 path를 구하는 문제는 GIS를 구축하는데 필요한 하나의 요소이다. 이러한 최단거리를 2차원 지형에서 구할 때는 시작점과 끝점을 일직선으로 연결해 주면 된다. 하지만 3차원 지형은 어떤 점에서의 고도와 경사도를 같이 고려해야 한다. 이렇게 시작점과 끝점 사이의 점들을 조사하면서 주위의 점의 경사도와 고도를 계산하여 가장 짧은 거리를 찾아내는 것이 문제이다.

최단거리를 구하기 위해서는 출발점과 도착점이 필요하다. 2차원 상에서 주로 사용하는 알고리즘으로 shortest path 알고리즘이 있다. 본 연구에서는 이 알고리즘을 3차원으로 적용하여 최단거리를 계산해 내었다. 이 알고리즘을 적용하기 위해서는 그래프가 형성되어 있어야 하며, 각 노드간의 weight가 구해져야 한다. 출발점과 도착점을 기준으로 사각형의 완전그래프를 형성했으며, weight는 인접한 8방향의 노드를 3차원 상의 거리를 구하고 고도차를 고려하여 구하였다. 이렇게 전처리를 한후에 shortest path 알고리즘을 적용시키면 된다. 알고리즘을 적용하고 나면, 그래프의 노드에는 시작점으로부터의 최소 거리가 들어 가게 된다. 이러한 과정을 정리하면 아래와 같다.

1. 출발점과 도착점을 중심으로 완전 그래프를 형성한다.
2. 8방향의 노드 사이의 weight를 구한다.
3. Shortest path 알고리즘을 적용한다.
  - 3.1 시작점으로부터 인접한 노드에 대해서 weight를 저장한다.
  - 3.2 각각의 노드로부터 다시 인접한 노드로 갈수 있는 경로를 얻는다.
  - 3.3 이 경로중에서 weight를 더하여 그중 최소 weight를 가지는 것을 저장한다.
  - 3.4 다시 각각의 노드에 대해서 인접한 노드의 weight를 구해서 3.2, 3.3 과정을 반복한다.
4. 3번 과정을 전 노드에 대해 반복한다

위와 같은 과정을 거치고 나면, 각각의 노드에는 최소의 weight를 얻을 수 있다. 하

지만 우리가 얻고자 하는 것은 weight외에도 그 경로를 알 수 있어야 한다. 이것은 각각의 노드로 최소의 거리를 가지고 오는 전 노드를 linked-list로 연결하여 알 수 있다. 결국 도착점의 노드로부터 역으로 link를 따라가면 출발점에 도착하여 그 경로를 얻어내는 것이다. 이 알고리즘의 단점은 출발점과 도착점의 최단 거리뿐 아니라 그래프를 형성하는 모든 노드에 관한 출발점까지의 최단 거리를 구해야 함으로 메모리나 계산 시간이 많이 걸린다는 단점을 가지고 있다. 하지만 정확한 최단거리를 구해낼 수 있다는 장점이 있으므로 최단거리를 구할 때 가장 많이 사용되는 알고리즘이다(Fig. 6-16).

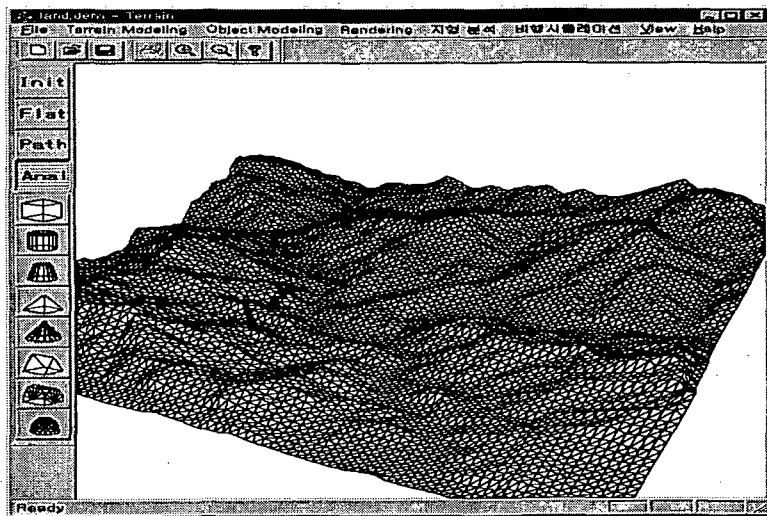


Figure 6-16 Shortest path analysis over terrain

## 6. 등고선 추출

### 가. 사각 그리드 DEM으로부터 등고선 추출

하나의 사각 그리드를 구성하는 두 개의 삼각형에 대하여 각각 세 개의 선분이 존재한다. 이 선분들과 정의된 높이 평면과의 교점 테스트를 수행한다. 찾아낸 두 개의 교점을 연결하여 만들어진 선분을 차례로 모두 연결하면 각 level에서의 등고선이 완성된다(Fig. 6-17).

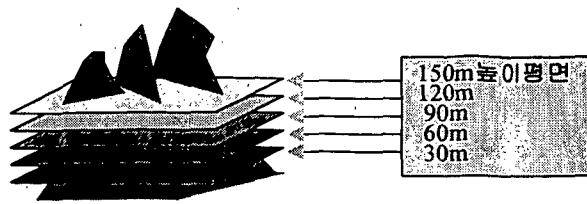


Figure 6-17 Definition of height plane

삼각형과 평면과의 교점 테스트에서 제한해야 할 조건은 크게 두가지로 구분할 수 있다. 테스트 하고자하는 지형이 높이 평면의 영역으로부터 완전히 벗어난 경우에는 교점 테스트를 하지 않고 다음 사각 그리드의 교점을 테스트한다. 이것은 교점 테스트에서의 수행시간을 최소화하기 위한 것이다. 높이 평면의 영역에서 완전히 벗어난 경우는 다음 Fig. 6-18과 같다.

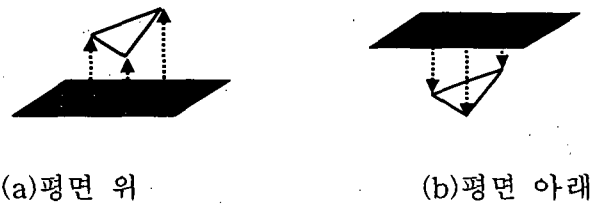


Figure 6-18 Exceptional case in plane intersection

등고선 평면과 삼각형 평면에서의 교점을 테스트하여 교점이 존재한다고 판단되는 경우에 한하여 삼각형의 세 선분과 높이 평면간의 교점을 찾아낸다. 각 높이 평면에 대하여 찾아낸 교점들을 모두 연결하면 하나의 등고선이 완성되며 이러한 등고선들을 모두 화면에 표현하여 전체 등고선을 추출한 것을 보여준다(Fig. 6-19).

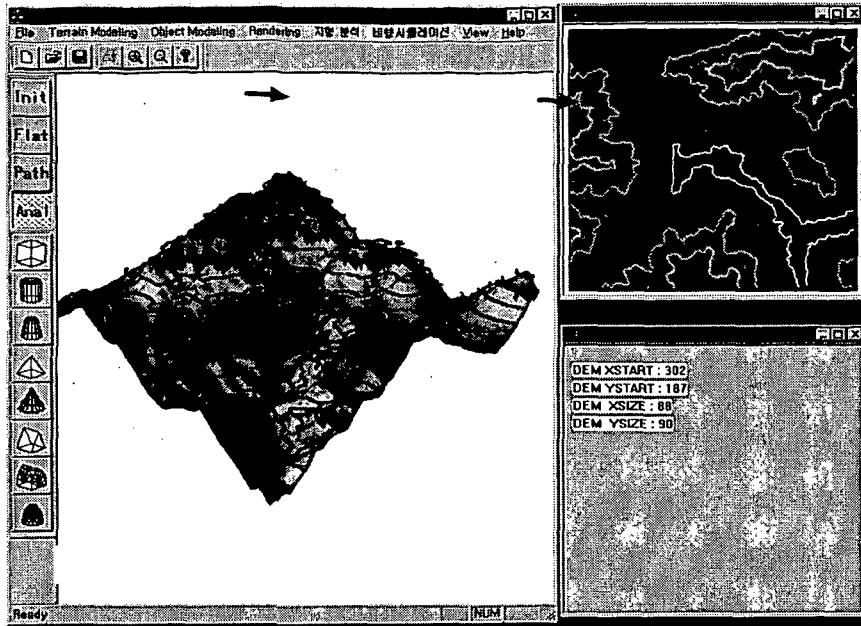
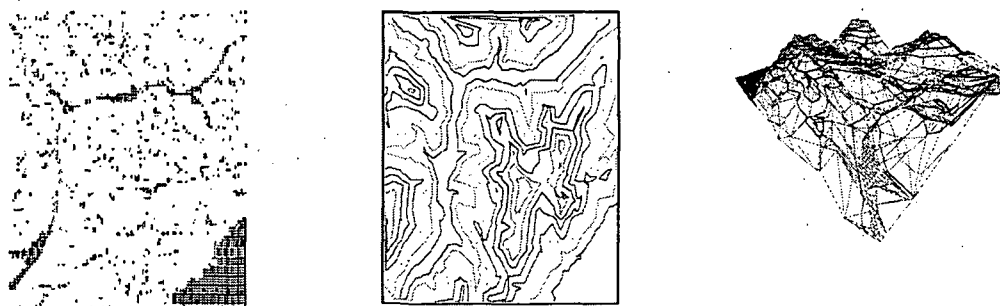


Figure 6-19 Contour extraction from DEM

나. TIN으로부터 등고선 추출

DEM 데이터로부터 등고선을 추출하는 과정과 동일하며 이 경우에는 TIN 데이터를 이용하기 위해 의미점 추출과정을 거쳐야 한다는 점이 추가된다. 다음 Fig. 6-20는 TIN 데이터에서 등고선을 추출하는 과정을 보인 것이다.



a) Random 데이터 (b) 2차원 지형에 매핑한 등고선 (c) 3차원 지형에 매핑한 등고선

Figure 6-20 Contour extraction from TIN

## 제 4 절 비행 시뮬레이션

비행 시뮬레이션에서 비행기의 경로는 조종사의 조종으로 결정된다. 하지만 비행기가 운행되는 경로가 정의되어 있는 경우에는 시스템에서 그 경로에 맞는 방향과 각도를 매 순간 계산해 주어야 한다. 이러한 경로의 설정은 사용자에게 의하여 이루어져야 하는데 경로 전체에 대해 사용자가 모두 지정한다는 것은 매우 어려운 작업이며 이렇게 지정된 경로는 부드러운 곡선의 모양을 가지기가 어려우므로 성공적인 시뮬레이션을 하기에는 부족한 점이 많다. 따라서 사용자의 경로의 형태를 제어할 몇개의 제어점만을 지정하고 시스템에서 이들 제어점의 위치를 사용하여 부드러운 경로를 생성하는 접근이 보다 바람직하다고 하겠다. 여기에서는 몇 개의 제어점을 입력으로 한 경로 생성 모듈을 포함하고 있으며 이 경로 생성 모듈은 비행 시뮬레이션을 보다 자연스럽게 진행시키기 위해서 부드러운 곡선 형태의 경로를 생성한다. 사용자의 이렇게 제어점의 설정으로 이러한 경로를 제어할 수 있으며 또한 그 결과를 쉽게 확인하여 최종적인 경로를 지정할 수 있는 것이다. 이러한 경로 생성 모듈은 우선 제어점에 의해 제어되는 보다 부드러운 곡선의 형태를 지닌 경로를 생성할 수 있는 기능과 이렇게 생성된 경로가 이후의 테스트에서 쉽게 변경시킬 수 있는 기능을 가져야 하며 빠른 수행을 위해서 경로 생성에 많은 시간을 빼앗겨서는 안된다. 이러한 비행 경로 생성을 위해서는 베지어 곡선과 스플라인 곡선의 알고리즘을 사용할수 있으면, 본 연구에서는 스플라인 곡선을 이용하여 비행 경로를 구성하였다. 스플라인 곡선 알고리즘은 곡선을 생성하기 위한 계산이 베지어 곡선에 비해 매우 복잡하고 따라서 곡선 계산에 많은 시간이 요구된다는 단점이 있으나 이러한 계산량이 제어점의 갯수에 영향을 받지 않아 제어점을 자유롭게 증가시킬 수 있으며 또한 하나의 제어점은 그 제어점이 속한 곡선의 일부분에만 영향을 미치므로 경로의 변경이 보다 용이하다는 장점으로 가지고 있다. 그리고 경로가 매우 길고 동시에 많은 제어점이 요구될 경우에 적합한 선택이 될 것이다. Fig. 6-21는 기본점에 의하여 생성된 스플라인 비행경로를 나타낸 것이다.

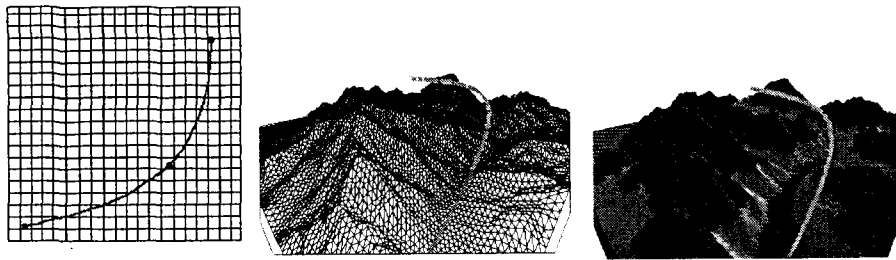
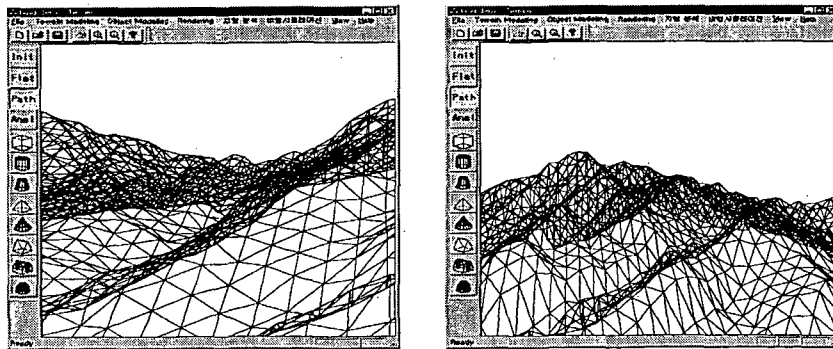


Figure 6-21 Generated Flight path using spline function with control points

## 1. DEM Simulation

비행 시뮬레이션을 DEM을 이용하여 와이어프레임이나 슈도셰이딩, 또는 인공위성 영상을 이용하여 수행하게 된다. 메인 윈도우에서는 실제 지형을 확대하여 비행 시뮬레이션을 수행하는 과정을 보여주며, 속도면에서는 가장 빠른 효과를 나타낸다. Pseudo shading을 선택하면 셰이딩된 지형에서 비행 시뮬레이션을 수행하며, wireframe를 선택한 것 보다 사실감은 있으나 속도는 떨어진다. Texture Mapping을 선택하면 인공위성 영상을 이용하여 실제 지형에 렌더링을 한후 비행 시뮬레이션을 하는 메뉴이다. 이 시뮬레이션은 속도는 가장 느리지만 가장 사실감 있는 영상을 제공하는 비행 시뮬레이션이다(Fig. 6-22, 6-23, 6-24).



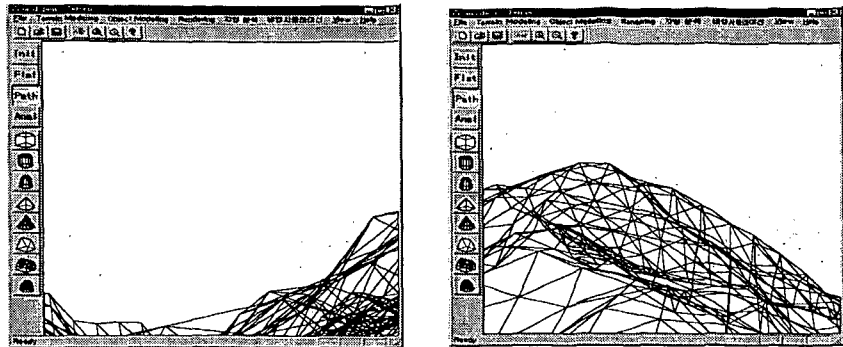


Figure 6-22 Flight simulation using wireframe

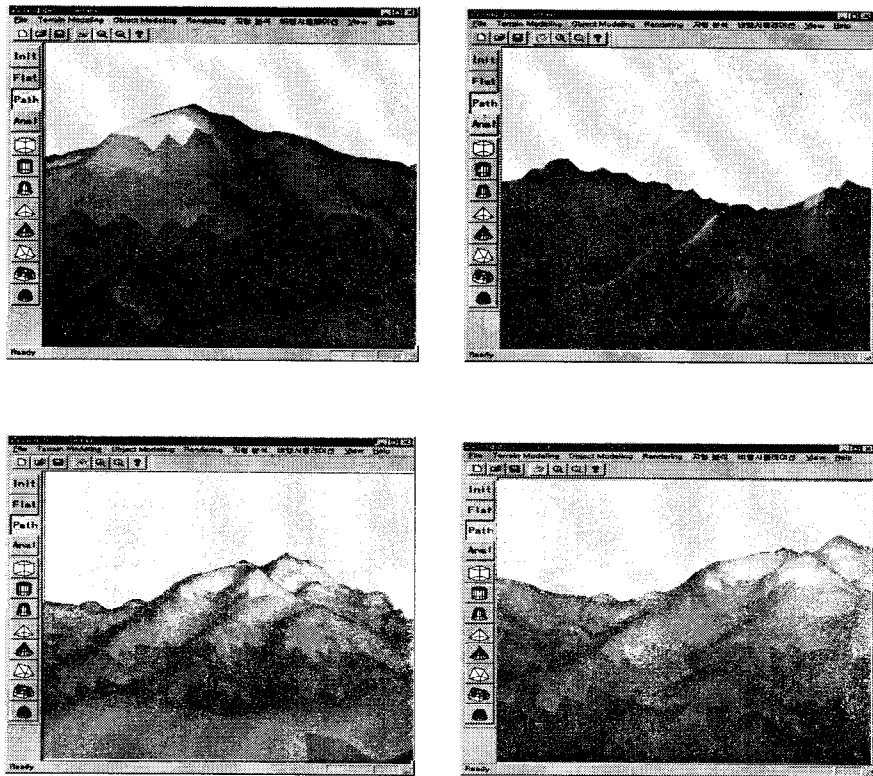


Figure 6-23 Flight simulation using pseudo shading



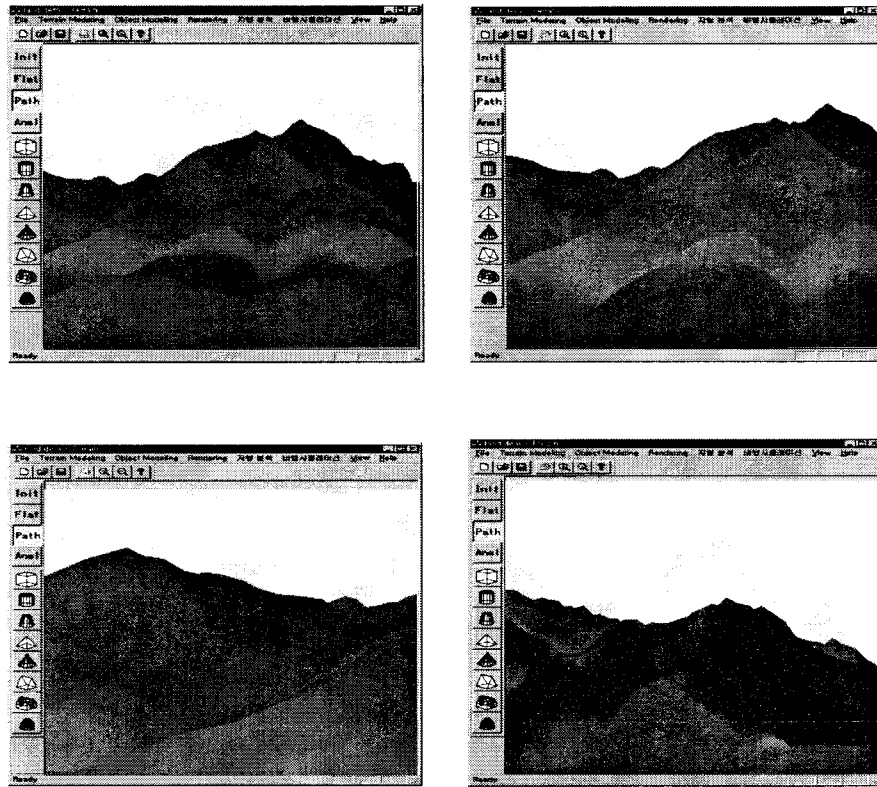


Figure 6-24 Flight simulation using texture mapping of satellite data

## 2. TIN Simulation

TIN을 이용하여 실행함으로써 DEM에 비해서 데이터의 양을 줄여, 전반적인 속도의 증가를 느낄 수 있다. 이외의 동작은 DEM 시뮬레이션과 동일하다(Fig. 6-25, 6-26).

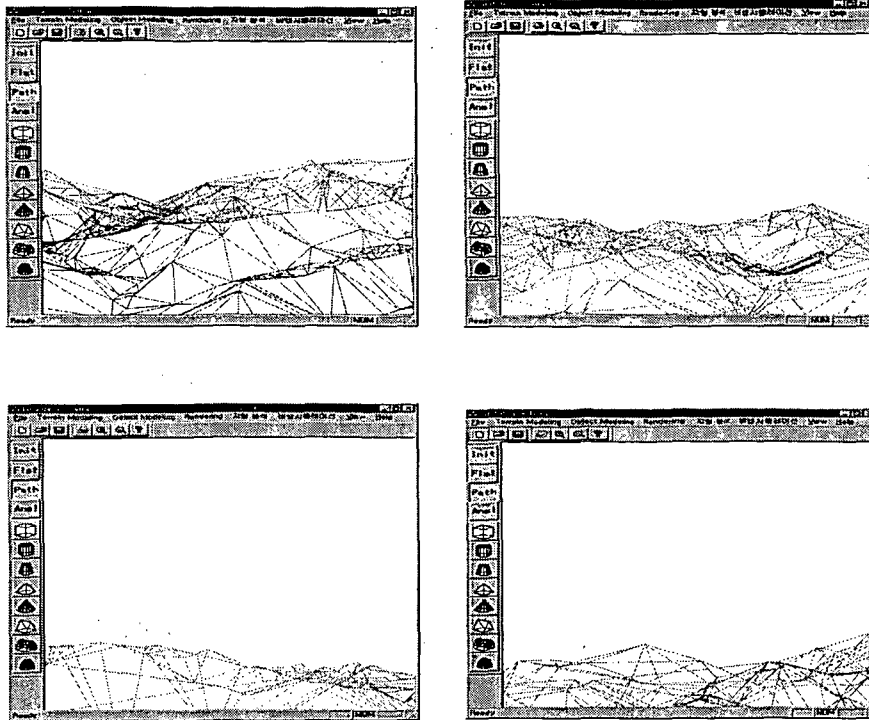


Figure 6-25 Flight simulation using TIN wireframe

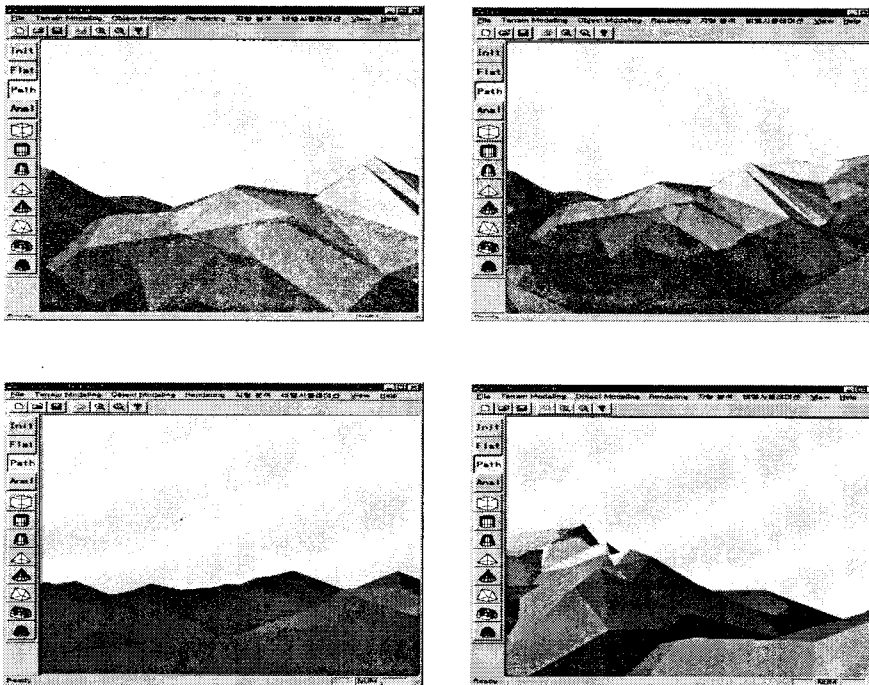
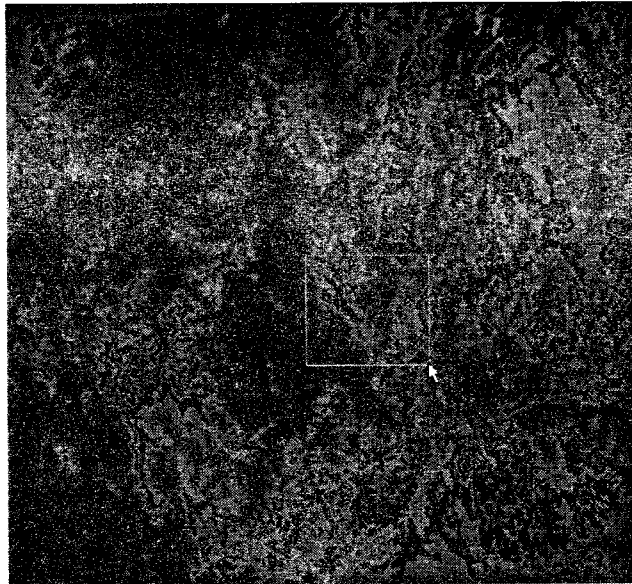


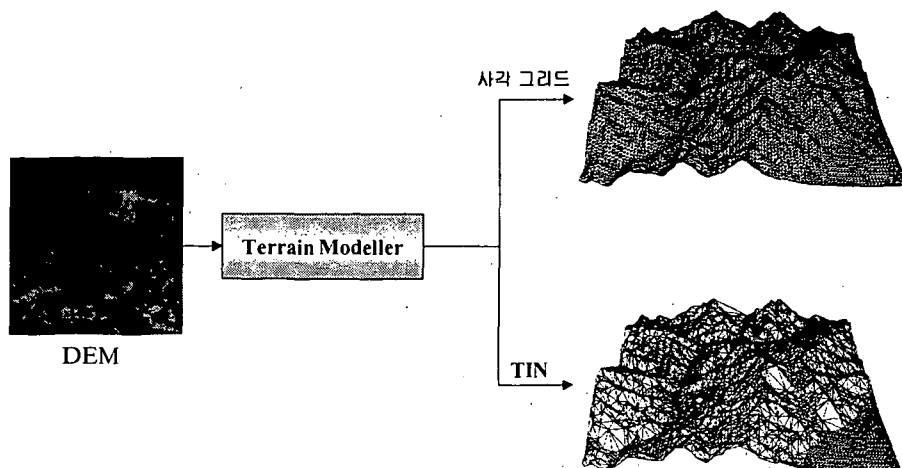
Figure 6-26 Flight simulation using TIN pseudo shading

## 제 5 절 개발된 지형 분석 기능을 이용한 지형 분석

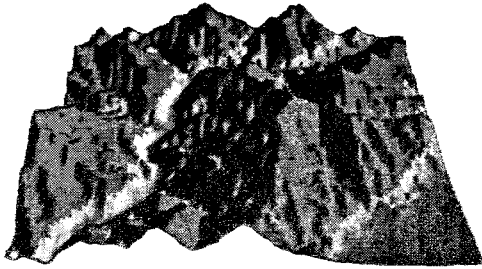
1. 대상 지역 선정 : DEM 파일을 화면 출력한 후 마우스를 이용하여 특정지역을 선택.



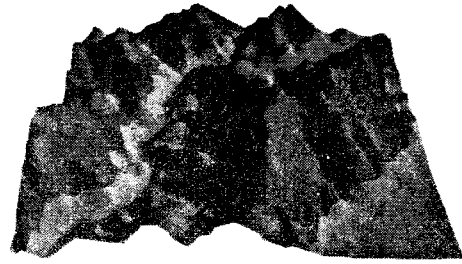
2. 3차원 지형의 모델링 : 선택된 지역에 대하여 원래의 격자형 DEM 자료를 그대로 이용하거나 지형의 특징점들을 추출하여 불규칙 삼각망을 구성한 후 3차원적 지형에 대한 모델링.



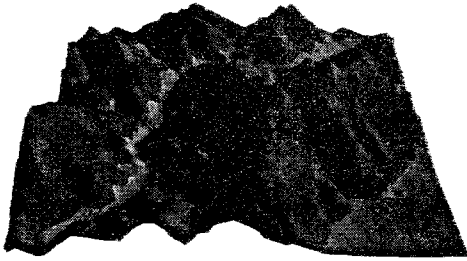
3. 사각 그리드 및 TIN 형성에 의한 지형의 모델링 : 사각 그리드,  
Delaunay TIN, Radial Sweep 알고리즘에 의하여 형성된 TIN을 이용한  
지형의 슈도 셰이딩.



(a) 사각 그리드

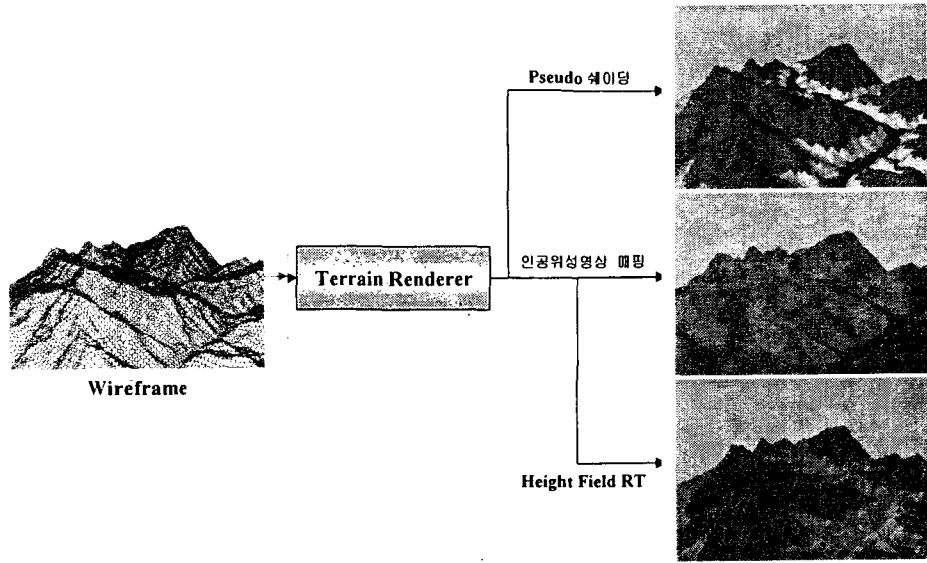


(b) Delaunay Triangulation



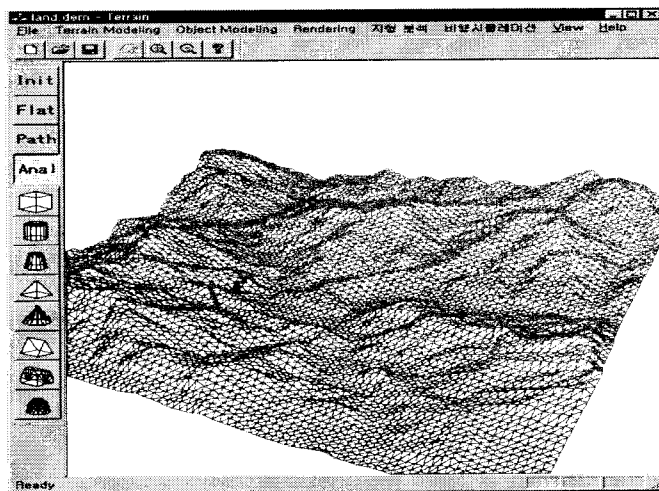
(c) 래디얼 스위프 알고리즘

4. 3차원 지형의 렌더링 : 사각 그리드, TIN에 의하여 표현된 3차원 지형을 와이어  
프레임, 슈도 셰이딩, 인공위성 맵핑등을 통한 입체적 표현

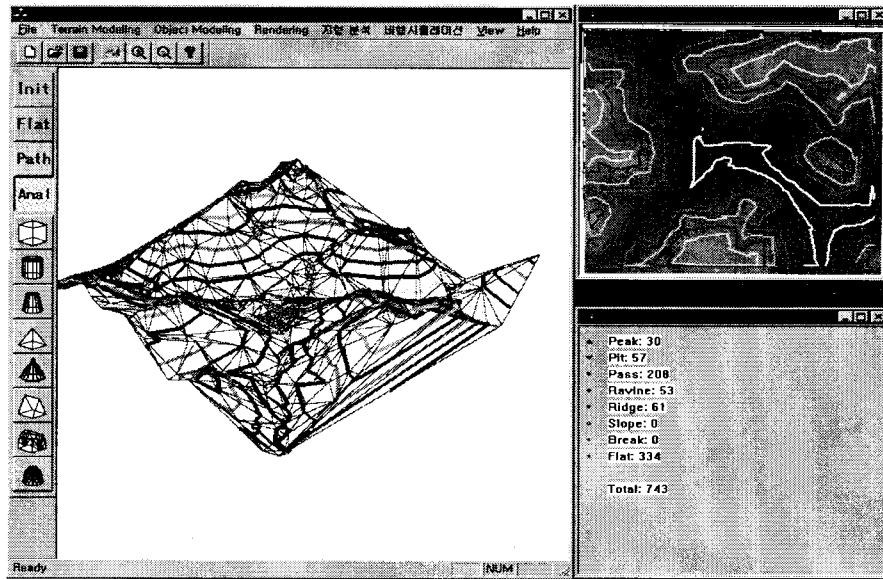


5. 지형분석 : DEM 또는 TIN을 이용한 지형요소의 분석

Slope, Aspect, Shaded Relief, Volume(Cut and Fill), Watershed, Shortest Path, Viewshed, 등고선 추출, 격자 데이터 생성

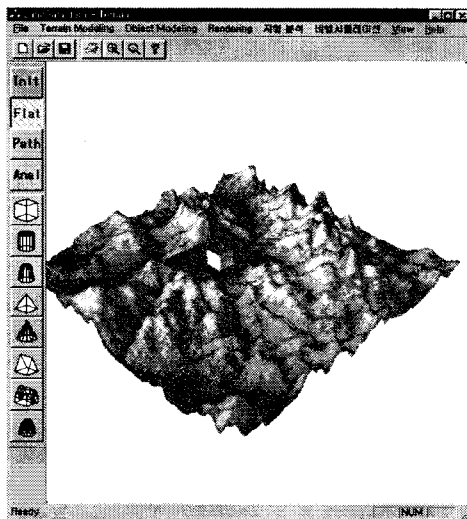


최단거리 구하기

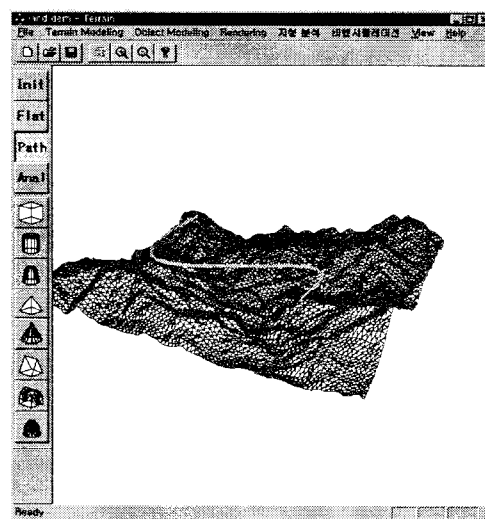


TIN에서의 등고선 추출

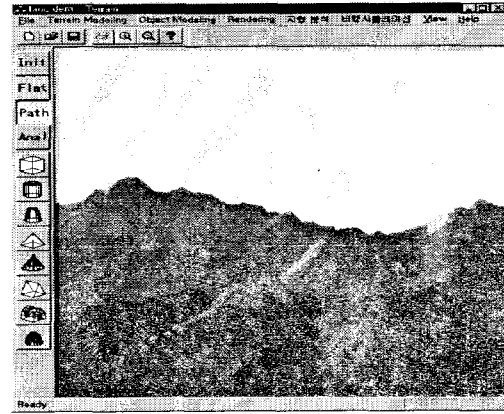
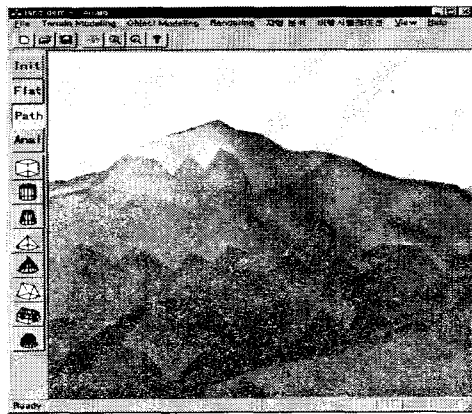
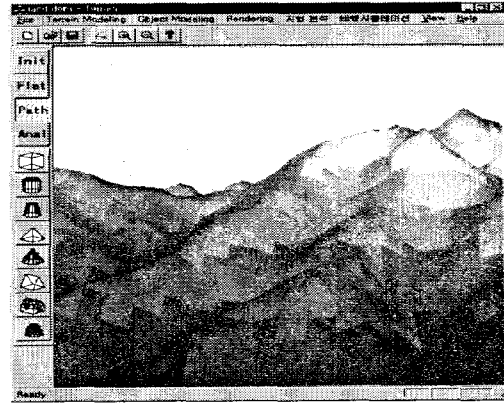
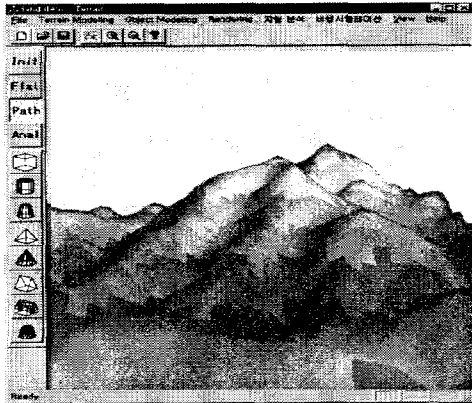
6. 3차원 경관 시뮬레이션 : DEM 또는 TIN을 이용한 비행 경로에 따른 경관의 실시간 시뮬레이션, 인공 구조물의 설치 가능



지형의 임의의 지점에 인공 구조물 삽입



비행 경로 생성



비행경로에 따른 경관 시뮬레이션

## 제 7 장 결 론

본 연구에서의 개발된 영상처리 및 지표면 정보 추출 모듈은 일반적으로 사용되는 기능들을 우선적으로 구현한 것으로 기능을 보완하기 위하여 필요한 부분은 다음과 같다.

1. 최신의 알고리즘에 의한 첨단 자료분석 기능의 구현으로 더욱 정확하고 빠른 자료 처리
2. 수치사진 측량, 레이더 자료처리, 초 다대역 분광 정보 분석 등 차세대 위성정보 처리에 필요한 부가적인 자료처리 모듈 구현
3. 시각화 프로그래밍 도구의 지원으로 보다 유연한 사용자 인터페이스 제공
4. 위성영상으로부터 고부가 정보의 추출에 있어 거치게 되는 여러 자료처리 과정에 대한 기록을 관리할 수 있는 data history 관리 기능
5. 보다 정밀한 전처리 기능, 특히 대기 보정 모듈의 구현
6. 대용량 영상자료의 처리에 필수적인 보다 효율적인 메모리/파일 관리 기능
7. 초고속 정보 통신망 및 인터넷 시대에 대비한 Client/Server 환경에서의 영상처리 서비스 지원 기능
8. Raster GIS의 기능 보완
9. 3차원 지형 분석 모듈의 JAVA 3D API를 활용한 구현으로 순수 JAVA 소스에 의한 S/W 구현.
10. 댐 상단 지형을 고려한 저수량 계산
11. 유수 방향 추적을 이용한 최적 댐 설치 장소 탐색 기능 추가
12. 스플라인 곡선을 이용한 등고선 내삽 기능 추가
13. 비행 시뮬레이션 모델링의 코드 최적화

본 연구의 성과물을 활용하기 위하여는 민간기업에 의한 상품화로 지속적인 S/W 성능의 유지 보수와 관련된 서비스가 가능해져야 한다. 철저한 시장 조사 및 현재 상용화되어 있는 많은 외국산 S/W의 취약한 분야에 대한 틈새시장을 개척하여 점차 상



품의 인지도를 높여야 한다.

아직까지 국내 원격탐사 관련 S/W 시장의 규모는 그다지 크지 않으나, 1999년부터 상용화될 것으로 기대되는 고분해능 위성영상과 다목적 실용위성의 개발 완료로 상당한 신규 사용자가 생길 것으로 예측되며, 이들 수요자는 대부분 단순 원격탐사 자료의 처리 보다는 GIS와 연계된 분석을 원하는 부류로 판단됨으로, 가능하다면 자체적으로 개발인력을 확보하고 있고, 기존의 벡터 기반의 GIS 틀을 가지고 있는 기업과 연계되어 하나의 패키지로써 공급 가능토록 하여야 할 것이다.

## 제 8 장 참고문헌

- 안충현, 신대혁, 위성자료 처리 및 3차원 지형분석 S/W 개발, 한국지형공정  
정보학회, 1997
- GIS World, 1996, Vol.9, No. 6
- ERDAS Field Guide, 1998, ERDAS, Inc.
- S. E. Umbaugh, 1998, *Computer Vision and Image Processing*,  
Prentice-Hall International, Inc. 504p.
- J. R. Jensen, 1986, *Introductory Digital Image Processing - A Remote  
Sensing Perspective*, 379p.
- H.E. Burdick, 1997, *Digital Imaging - Theory and Applications*, 315p.
- R. Crane, 1997, *A simplified approach to image processing - Classical  
and Modern Techniques in C*, 317p.
- B. K. Choi and H. Y. Shin, 1988, "Triangulation of scattered data in 3D  
space", *Computer-aided design*, Vol. 20, No. 5, pp. 239-248.
- McCormick. B. H., T. A. DeFanti, and M. D. Brown, 1987, "Visuaization  
in Scientific Computing", *Computer Graphics*, vol. 21, No. 6.
- L. De Floriani et al., 1993, "Extracting Contours from a Hierarchical  
Surface Model", *Eurographics '93*, Vol. 12, No. 3, pp. c-249-c-260.
- A. Fournier and D. Y. Montuno, 1984, "Triangulating Simple Polygons  
and Equivalent Problems", *ACM Translations on Graphics*, Vol. 3,

- No. 2, pp. 153-174.
- B. Joe, 1991, "Construction of three-dimensional Delaunay triangulations using local transformations", *Computer Aided Geometric Design*, Vol. 8, pp. 123-142.
- D. T. Lee and C. K. Wong, 1980, "Voronoi diagrams in  $L_1(L_\infty)$  metrics with 2-dimensional storage applications", *Siam J. Comput.*, Vol. 9, No. 1, pp. 200-211.
- A. Mirante and N. Weingarten, 1982, "The radial sweep algorithm for constructing triangulated irregular networks", *IEEE Computer Graphics & Application*, Vol. 2, pp. 11-21.
- T. K. Peucker and D. H. Douglas, 1975, "Detection of surface-specific points by local parallel processing of discrete terrain elevation data", *Computer Vision Graphics and Image Processing*, Vol. 4, pp. 375-387.
- F. Schmitt, X. Chen and W. H. Du, 1991, "Geometric Modelling from Range Image Data", *Eurographics '91*, Vol. 10, pp. 317-328.
- R. C. Veltkamp, 1993, "3D Computational Morphology", *Eurographics '93*, Vol. 12, No. 3, pp. c-115-c-127.
- Lee J. 1992, "Comparison of existing methods for building triangulated irregular network models of terrain from grid digital elevation models.", *Int. J. of GIS* 5(3), 267-285.
- Lee Chan, 1997, *The Java Class Libraries*- 2nd ed., Vol.2, Addison

- Wesley, pp.795-844.
- Sun Microsystems, Inc., 1998a, *Java™ Development Kit Version 1.2-Summary of New Features*,  
<http://java.sun.com/products/jdk/1.2/docs/relnotes/features.html>.
- Sun Microsystems, Inc., 1998b, *Java™ 2D*, <http://java.sun.com/products/java-media/2D/index.html>.
- Sun Microsystems, Inc., 1998c, *Java 3D™*,  
<http://java.sun.com/products/java-media/3D/index.html>.
- Sun Microsystems, Inc., 1998d, *Java™ Advanced Imaging API*,  
<http://java.sun.com/products/java-media/jai/index.html>.
- C. Mangione, 1998, "Performance tests show Java as fast as C++",  
Javaworld,  
<http://www.javaworld.com/javaworld/jw-02-1998/jw-02-jperf.html>.
- M. Roulo, 1998, "Accelerate your Java apps!", Javaworld,  
<http://www.javaworld.com/javaworld/jw-09-1998/jw-09-speed.html>.
- E. Armstrong, 1998, "HotSpot: A new breed of virtual machine",  
Javaworld,  
<http://www.javaworld.com/javaworld/jw-03-1998/jw-03-hotspot.html>.
- J. B. Gosling, Joy, and G. Steele, 1996, *The Java Language Specification-version. 1.0*, Addison-Wesley,  
<http://java.sun.com/docs/books/jls/html/6.doc.html#27725>.
- Sun Microsystems, Inc., 1998e, "JNI Enhancements in JDK 1.2", *Java™*

- Development Kit Documentation-JDK™ 1.2*,  
<http://java.sun.com/products/jdk/1.2/docs/guide/jni/jni-12.html>
- D. Lea, 1997, *Concurrent programming in Java: design principles and patterns*, Addison-Wesley, One Jacob Way, Reading, Massachusetts.
- D. H. Shin, C.H. Ahn, S.H. Oh, and T.Y. Kwon, "Satellite Image Processing based on Multithreaded and Data-driven structure", *Proceedings of International Symposium on Remote Sensing*, pp.173-182
- Sun Microsystems, Inc., 1998f, "Performance Enhancement", *Java™ Development Kit Documentation-JDK™ 1.2*,  
<http://java.sun.com/products/jdk/1.2/docs/guide/performance/index.html>.
- Sun Microsystems, Inc., 1997, "Thread Deprecation Changes in Java 1.2" *The Java™ Tutorial*,  
<http://java.sun.com/docs/books/tutorial/post1.0/preview/threads.html>.
- M. Pawlan, 1998, "Reference Objects and Garbage Collection", Sun Microsystems, Inc.,  
<http://developer.java.sun.com/developer/technicalArticles/monicap/RefObj/refobj.htm>

## 부록 1. 상용 S/W와의 기능 비교

표 4.1은 본 연구에서 개발된 GeoPixel과 상용으로 많이 사용하고 있는 PCI EASI/PACE, ENVI, ERDAS Imagine, ERMapper의 기능(GIS World, 1996)을 비교한 것이다. 일반적으로 S/W의 Version Up은 시스템 전반적인 개조 보다는 새로운 모듈의 추가로 이루어짐으로 볼 때 현재의 최신 버전과는 다소 기능이 다를 수 있다. 이러한 기능의 비교로부터 연구 결과물의 상품화를 전제로 할 경우 보강해야 할 기능과 추가적으로 구현하여야 할 기능을 쉽게 판단할 수 있다.

Function	EASI/PACE (6.0.1)	ENVI (2.0)	ERDAS Imagine (8.2)	ERMapper (5.2)	GeoPixel
Data Format					
Native					
- Raster	BIL/BSQ/BIP	BIL/BSQ/BIP	Internal	BIL	BIL
- Vector	Internal	Internal	ARC/INFO	Internal	Internal
Import					
- Raster	36 + UD	24 + UD	17 +UD	98 + UD	7 + UD
- Vector	11	2	7	14	1
- ARC/INFO	y	n	y	y	n
- DGN	y	n	n	y	n
Export					
- Raster	9	n	6	29	2
- Vector	5	n	7	3	n
- ARC/INFO	y	n	y	y	n
- DGN	n	n	n	n	n
Media Support					
- Standard Tape format	y	y	y	y	n
- User Defined Format	y	y	y	y	n
- Taep utilities	y*	y	y	y	n
Graphic Display					
- Single and MultiBand Image	y	y	y	y	y
- Vector display	n	y	y	y	y
- Vector query by attribute	n	n	y	y	n
- Vector edit	y	n	y*	y	n
- Screen display	single	three	single	single	three
Radiometric Correction					
- Destripe/bad lines	y	y	y	y	n
- Atmospheric Effect	y	y	y	y	y(simple)
- Topographic Effect	y	y	y	y	y
Enhancements					
Radiometric					
- Built-in contrast Stretch	y	y	y	y	y
- Arbitrary contrast Stretch	y	y	y	y	y
- Histogram Equalize	y	y	y	y	y
- Histogram Matching	y	y	y	y	y
Spatial					
- Filter support	good	good	good	excellent	average
- Data fusion	y(limited)	n	y(limited)	y(limited)	y(limited)
Frequency					
- FFT forward/inverse	y	y	y	y	y
Spectral					
- PCA	y	y	y	y	y
- Tasselled Cap	y	y	y	y	y
- Vegetation Index	y	y	y	y	y
- RGB to HIS	y	y	y	y	y
- Decorrelation	y	y	y	y	n
- User defined	y	y	y	y	n

Function	EASI/PACE (6.0.1)	ENVI (2.0)	ERDAS Imagine (8.2)	ERMapper (5.2)	GeoPixel
Registration/Rectification /Geometrical Correction					
- Simple Georeferencing	y	y	y	y	y
- Image to Image	y	y	y	y	y
- Image to Map	y	n	y	y	y
- Other	n	y	n	n	n
- GCP Prediction	n	n	n	y	y
- Digitizer registration	y	n	y	n	y
- Vector warp to image	y	n	n	n	n
Mosaicking	y	y	y	y	y
Terrain Analysis	y	y	y	y	y
Classification					
- Statistic reports	good	good	excellent	excellent	good
- Masking	y	y	y	y	y
- Training set def. Interactive	y	y	y	y	y
Ext. raster/vector	n	n	y	y	n
- Unsupervised Method	3	1	2	1	2
- Supervised Method	3	4	5	5	5
- Class merging	y	y	y	y	n
- Class filtering and Sieving	y	y	y	y	n
Raster GIS Functionality	good	minimal	excellent	minimal	poor
Nap Production					
- Graphic file output	y	y	y	y	y
- Graphics window to printer	y	n	y	y	y
- Map composition	y*	n	y	y	n
Customization					
- Batch processing	y	y	y	n	y
- Programming lib.	y	IDL	y*	n	y OpenGL
- Visual Programming	n	n	minimal	minimsl	n
- Algorithm support	n	n	simple	y	n
- Image history save	n	n	n	y	n
Developing Language	C/C++	C/C++	C/C++	C/C++	Java/C++
Price(\$)	4,200 +?	3,350-4,475	6,000 +?	2,900 + ?	



여 백

## 부록 2. 연구 발표 성과

### [학회지]

1. 3차원 지형 모델링을 위한 개선된 radial sweep 알고리즘 개발, 지형공간학회지, 5, 2, 77 - 85, 1998.
2. JAVA를 이용한 위성 영상 처리/ 분석 시스템 개발 - GeoPixel 1.0, 대한원격탐사학회지, 12, 3, 229 - 244, 1996.
3. 3차원 지형 모델링을 위한 래디얼 스위프 알고리즘, 중앙대학교 기술과학연구소 논문집, 26, 133 - 140, 1996.

### [학술발표]

1. Development of Remotely Sensed Image Processing/Analysis System via Internet, '98 Symp. of ASPRS, 1998.
2. 핵심어 탐색엔진을 이용한 위성영상처리 통합 소프트웨어의 사용자 인터페이스, 대한원격탐사학회, 1998.
3. 3차원 지형 모델링을 이용한 지형분석 모듈 개발에 관한 연구, 한국정보과학회, 1998.
4. 수치고도모델로부터 등고선 추출에 관한 연구, 한국정보과학회, 1998.
5. Satellite Image Processing based on Multi-thread and Data-driven structure, 6th Annual Workshop of EMSEA, 1997.
6. 위성자료 처리 및 3차원 지형분석 S/W 개발, 한국지형공간정보학회, 1997
7. JAVA를 이용한 위성영상 처리 소프트웨어 개발, 대한원격탐사학회, 1997
8. Empirical Correction of topographic-induced radiance variation in LANDSAT TM images, 6th Annual Workshop of EMSEA, 1996.
9. Remotely Sensed Image coding using wavelet transform and interband prediction, 5th Annual Workshop of EMSEA, 1996.

### [프로그램 등록]

1. JAVA언어를 이용한 위성영상처리 분석 소프트웨어 - OASIS 1.0, 1997.

2. 3차원 지형분석 S/W - GeoTerrain Modeller, 1998.
3. 3차원 비행 시뮬레이션 S/W - GeoTerrain Flighter, 1998.

[상표 등록]

1. GeoPixel : 국내 상표 등록, 1997.  
미국 상표 등록 신청중, 1997.

### 부록 3. 연구개발목표 달성도 및 대외 기여도

#### 제 1 절 평가의 착안점

착안점	설명	자체 평가
범용성	<ol style="list-style-type: none"> <li>이 부분은 개발된 모듈들이 계획된 기능을 모두 발휘하는가.</li> <li>사용자 인터페이스의 구성 등 사용의 용이성에 대한 실무자들의 평가</li> </ol>	<ol style="list-style-type: none"> <li>모듈 개발 후 다양한 종류의 영상을 사용하여 기능 점검.</li> <li>JAVA Component로 제공되는 Swing, JCLASS등 최신버전의 공개 컴포넌트를 활용.</li> <li>공개 발표회('98 GITA 및 학회 발표)를 통한 전문 사용자들로부터의 평가 받음.</li> </ol>
효율성	<ol style="list-style-type: none"> <li>대용량 데이터는 어느 정도까지 가능한가 ?</li> </ol>	<ol style="list-style-type: none"> <li>처리 대상영상 전체를 메모리에 올리지 않고 사용자가 선택할 수 있도록 지원.</li> <li>처리 대상지역을 사용자가 한정시키거나 파일에서의 처리를 통하여 대용량 자료에 대한 문제점 해결</li> </ol>
용이성	<ol style="list-style-type: none"> <li>사용자가 개발된 시스템을 이용하여 쉽게 다른 응용 모듈의 개발을 지원하는가 ?</li> <li>도움말 기능이 지원되는가 ?</li> </ol>	<ol style="list-style-type: none"> <li>API에 대한 online 사용자 매뉴얼을 제공함으로써 확장 지원.</li> <li>online Help 지원</li> <li>사용자 독자적인 메뉴의 생성 기능 지원</li> </ol>
다양성	<ol style="list-style-type: none"> <li>각 사용자의 요구에 부합하는 기법 들이 제공되었는가?</li> </ol>	<ol style="list-style-type: none"> <li>불특정 다수의 대상이 필요로 하는 위성영상 활용의 대부분 기능들을 선택할 수 있도록 지원. (예 식생지수 13 가지 지원 수치값을 반사휘도로 변환)</li> </ol>

## 제 2 절 연도별 연구개발 목표 및 연구 내용

연도	연구 개발 목표	비고
1차년도 (1995.12. 1. - 1996. 11. 30.)	<ul style="list-style-type: none"> <li>• 국내외 자료 수집</li> <li>• Remote Sensig에 의한 자료획득 및 기본모듈 개발</li> <li>• 도형 및 공간자료의 압축 S/W 설계 및 기본 모듈 구현</li> <li>• 3차원 지형분석 S/W 설계 및 기본모듈 개발</li> </ul>	C/C++언어로 개발
2차년도 (1996.12. 1. - 1997. 11. 30.)	<ul style="list-style-type: none"> <li>• 국내외 자료 수집 및 분석</li> <li>• Remote Sensig에 의한 자료획득 및 기본모듈 개발</li> <li>• 3차원 지형분석 S/W 설계 및 기본모듈 개발</li> </ul>	JAVA언어를 이용한 완전 객체 지향적 개발  컴포넌트화 지향  다중스레드 기반의 영상처리 Porototype S/W
3차년도 (1997.12. 1. - 1998. 11. 30.)	<ul style="list-style-type: none"> <li>• 시스템 최적화 및 안정화                             <ul style="list-style-type: none"> <li>- 최종 개발 S/W의 최적화 및 안정화</li> <li>- 온라인 사용자 도움말 및 보고서 작성</li> </ul> </li> <li>• Remote Sensig에 의한 자료획득 S/W GeoPixel 개발                             <ul style="list-style-type: none"> <li>- 프로토타입의 소프트웨어 기능 보완</li> <li>- 다양한 형식의 자료 입출력 지원</li> <li>- 화면 디지털이징 모듈 구현</li> <li>- 주파수 영역에서의 FFT 필터 구현</li> </ul> </li> <li>• 3차원 지형모델링 및 Viewing S/W 개발                             <ul style="list-style-type: none"> <li>- 3차원 지형 분석 모듈 및 사용자 인터페이스 강화</li> <li>- 3차원 경관 시뮬레이션 모듈 구현</li> <li>- Watershed 분석 모듈 설계 및 구현</li> <li>- Data Interpolation 모듈 설계 및 구현</li> </ul> </li> </ul>	단일스레드 방식 채택  키워드 탐색엔지에 의한 모듈 접근 지원  사용자 메뉴 자동 생성 지원

## 부록 4. GeoPixel API 설명서

[Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)  
[PREV](#) [NEXT](#) [FRAMES](#) [NO FRAMES](#)  
[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

---

### A

[ADD](#) - Static variable in class [UnaryArithmeticFunctionOp](#)

[ADD](#) - Static variable in class [BinaryArithmeticFunctionOp](#)  
덧셈 연산자

[addFeature\(\[Feature\]\(#\)\)](#) - Method in class [GeoVector](#)  
feature를 추가한다.

[addLine\(\[boolean\]\(#\)\[\], \[int\]\(#\)\)](#) - Method in class [AOI.Bit](#)

[addMenuItem\(\[MenuToken\]\(#\)\)](#) - Method in class [MenuToken](#)  
현재 [MenuToken](#) 에 [MenuItem](#) 이나 [Menu](#) 가 될 [MenuToken](#) 을 child 로 붙인다.

[addPolygon\(\[Polygon\]\(#\)\)](#) - Method in class [AOI.Graph](#)

[addVertex\(\[double\]\(#\), \[double\]\(#\)\)](#) - Method in class [DXFtoVectorOp.FeatureProtocol](#)

Vertex를 추가한다.

[addVertex\(\[double\]\(#\), \[double\]\(#\)\)](#) - Method in class [Feature](#)

[addVertex\(\[double\]\(#\), \[double\]\(#\)\)](#) - Method in class [Feature.Point](#)

[addVertex\(\[Feature.Point\]\(#\)\)](#) - Method in class [Feature](#)

[addVertex\(\[Feature.Point\]\(#\)\)](#) - Method in class [Feature.Point](#)

[adjustScrollViewSize\(\[Dimension\]\(#\), \[int\]\(#\), \[int\]\(#\)\)](#) - Static method in class [ViewCalc](#)

[AOI](#) - class [AOI](#).

[AOI\\_ADD](#) - Static variable in class [AOILayer](#)

[AOI\\_DELETE](#) - Static variable in class [AOILayer](#)

[AOI\\_NONE](#) - Static variable in class [AOILayer](#)

[AOI.Bit](#) - class [AOI.Bit](#).

[AOI.Bit\(\)](#) - Constructor for class [AOI.Bit](#)

[AOI.Bit\(\[int\]\(#\)\)](#) - Constructor for class [AOI.Bit](#)

[AOI.Bit\(\[int\]\(#\), \[int\]\(#\)\)](#) - Constructor for class [AOI.Bit](#)

AOI.Graph - class AOI.Graph.

AOI.Graph() - Constructor for class AOI.Graph

AOI.Graph(String) - Constructor for class AOI.Graph

AOI() - Constructor for class AOI

AOILayer - class AOILayer.

AOILayer() - Constructor for class AOILayer

ask(Window, String) - Static method in class GeoEnviron  
사용자 인터럽트에 대하여 질의를 하고 그 결과를 되돌린다.

assert() - Method in class CubeHeader  
영상의 헤더 정보가 정확하게 설정되어 있는지를 확인한다.

---

## B

B\_SPLINE - Static variable in class GeoResampler  
B-Spline 법

BAND\_MODE - Static variable in class Target

Band1 - Variable in class ScattergramOp

Band2 - Variable in class ScattergramOp

bandMax - Variable in class ISODATAClassifyOp

bandMax - Variable in class KMeansClassifyOp

bandMax - Variable in class KMeansUnSupervisedClassifyOp

bandMin - Variable in class ISODATAClassifyOp

bandMin - Variable in class KMeansClassifyOp

bandMin - Variable in class KMeansUnSupervisedClassifyOp

Bands - Variable in class Cube  
영상의 분광대 수를 정의한다.

BILINEAR - Static variable in class GeoResampler  
공이차법

BiLinear(byte[][][], byte[][][], boolean[], float[], float[], int, int) -  
Method in class GeoResampler  
바이트형 배열에 대한 BiLinear 재배열을 실행한다.

BiLinear(float[][][], float[][][], boolean[], float[], float[], int, int) -  
Method in class GeoResampler

4바이트 실수형 배열에 대한 BiLinear 재배열을 실행한다.

BiLinear(int[][][], int[][][], boolean[], float[], float[], int, int) -  
Method in class GeoResampler

4바이트 정수형 배열에 대한 BiLinear 재배열을 실행한다.  
BiLinear(short[[]], short[[]], boolean[], float[], float[], int, int) - Method in class GeoResampler  
 2바이트 정수형 배열에 대한 BiLinear 재배열을 실행한다.  
BilToBipConvertOp - class BilToBipConvertOp.  
 BIL 포맷의 영상을 BIP 포맷의 파일로 변환하여 저장한다.  
BilToBipConvertOp() - Constructor for class BilToBipConvertOp  
 Constructor  
BilToBsqConvertOp - class BilToBsqConvertOp.  
 BIL 포맷의 영상을 BSQ 포맷으로 변환하여 저장한다.  
BilToBsqConvertOp() - Constructor for class BilToBsqConvertOp  
  
BinaryArithmeticFunctionOp - class BinaryArithmeticFunctionOp.  
 두개의 분광대에 대하여 산술 연산을 한다.  
BinaryArithmeticFunctionOp() - Constructor for class BinaryArithmeticFunctionOp  
 Constructor  
BinaryLogicalFunctionOp - class BinaryLogicalFunctionOp.  
 두개의 분광대에 대하여 논리 비교 연산을 한다.  
BinaryLogicalFunctionOp() - Constructor for class BinaryLogicalFunctionOp  
 Constructor  
BinaryPixelValueReplaceOp - class BinaryPixelValueReplaceOp.  
 두개의 분광대에 대하여 논리 비교 연산을 한다.  
BinaryPixelValueReplaceOp() - Constructor for class BinaryPixelValueReplaceOp  
 Constructor  
BipToBilConvertOp - class BipToBilConvertOp.  
 BIP 포맷의 영상을 BIL 포맷으로 변환한다.  
BipToBilConvertOp() - Constructor for class BipToBilConvertOp  
 Constructor  
BIT - Static variable in class AOI  
  
BndOffset - Variable in class Cube  
 영상의 Bands에 대한 오프셋을 정의한다.  
BOUNDARY - Static variable in class UnaryMorphologicalFunctionOp  
  
BSpline(byte[[]], byte[[]], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 바이트형 배열에 대한 B-Spline 재배열을 실행한다.  
BSpline(float[[]], float[[]], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 4바이트 실수형 배열에 대한 B-Spline 재배열을 실행한다.  
BSpline(int[[]], int[[]], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 4바이트 정수형 배열에 대한 B-Spline 재배열을 실행한다.  
BSpline(short[[]], short[[]], boolean[], float[], float[], int, int)  
 -Method in class GeoResampler  
 2바이트 정수형 배열에 대한 B-Spline 재배열을 실행한다.  
BsqToBilConvertOp - class BsqToBilConvertOp.  
 BSQ 포맷의 영상을 BIL 포맷으로 변환한다.  
BsqToBilConvertOp() - Constructor for class BsqToBilConvertOp  
 Constructor  
ButterworthBPFFilteredFrequency(float[[]], int) - Method in class FFTOP



ButterworthBSFilteredFrequency(float[[[]], int) – Method in class FFTOp

ButterworthHPFilteredFrequency(float[[[]], int) – Method in class FFTOp

ButterworthLPFilteredFrequency(float[[[]], int) – Method in class FFTOp

BW – Static variable in class ImageMaker

BYTE – Static variable in class Cube  
바이트형을 정의한다.

---

## C

calculateFrequency(int, byte[], int, int) – Method in class HistogramOp

calculateFrequency(int, float[], int, int) – Method in class HistogramOp

calculateFrequency(int, int[], int, int) – Method in class HistogramOp

calculateFrequency(int, short[], int, int) – Method in class HistogramOp

calculateIndex(double[[[]], int, float[[[]], int, int) – Method in class ChromaticityAnalysisOp

Chromaticity를 계산한다.

calculateTexture(int, byte[], int, float[], int, int, int) – Method in class UnaryTextureFunctionOp

changePixelValue(byte[], int, byte[], int, int) – Method in class UnaryBitwiseFunctionOp

changePixelValue(byte[], int, byte[], int, int) – Method in class HistogramModifyEnhanceOp

changePixelValue(byte[], int, int, byte[], int, int) – Method in class InteactiveEnhanceOp

changePixelValue(byte[], int, int, byte[], int, int) – Method in class MinMaxEnhanceOp

changePixelValue(byte[], int, int, byte[], int, int) – Method in class MinMaxLinearEnhanceOp

changePixelValue(byte[], int, int, byte[], int, int) – Method in class PiecewiseEnhanceOp

changePixelValue(double[[[]], int, float[[[]], int, int) – Method in class TasseledCapTransformOp

changePixelValue(double[], double[], int, byte[], int, int) – Method

in class BinaryLogicalFunctionOp

비교 연산을 수행한다.

changePixelValue(double[], double[], int, float[], int, int) - Method

in class BinaryArithmeticFunctionOp

산술 연산을 수행한다.

changePixelValue(double[], double[], int, float[], int, int) - Method

in class BinaryPixelValueReplaceOp

비교 연산을 수행한다.

changePixelValue(double[], int, byte[], int, int) - Method in class  
LevelSliceOp

changePixelValue(double[], int, float[], int, int) - Method in class  
UnaryArithmeticFunctionOp

changePixelValue(double[], int, float[], int, int) - Method in class  
UnaryMathFunctionOp

changePixelValue(double[], int, int, float[], int, int) - Method in  
class DNTToRadianceOp

changePixelValue(double[], int, int, float[], int, int) - Method in  
class DNTToReflectanceOp

changePixelValue(float[], int, byte[], int, int) - Method in class  
HistogramModifyEnhanceOp

changePixelValue(float[], int, int, byte[], int, int) - Method in class  
MinMaxEnhanceOp

changePixelValue(float[], int, int, byte[], int, int) - Method in class  
MinMaxLinearEnhanceOp

changePixelValue(int[], int, byte[], int, int) - Method in class  
HistogramModifyEnhanceOp

changePixelValue(int[], int, int, byte[], int, int) - Method in class  
MinMaxEnhanceOp

changePixelValue(int[], int, int, byte[], int, int) - Method in class  
MinMaxLinearEnhanceOp

changePixelValue(short[], int, boolean, int, byte[], int, int) -  
Method in class MinMaxEnhanceOp

changePixelValue(short[], int, boolean, int, byte[], int, int) -  
Method in class MinMaxLinearEnhanceOp

changePixelValue(short[], int, byte[], int, int) - Method in class  
HistogramModifyEnhanceOp

changePixelValue(short[], int, int, byte[], int, int) - Method in class  
MinMaxEnhanceOp

changePixelValue(short[], int, int, byte[], int, int) - Method in class  
MinMaxLinearEnhanceOp

ChromaticityAnalysisOp -class ChromaticityAnalysisOp.  
3개의 분광대를 이용하여 영상의 chromaticity X, Y, Z을 계산한다.

ChromaticityAnalysisOp() -Constructor for class ChromaticityAnalysisOp  
Constructor

ClassifyErrorAnalysisOp -class ClassifyErrorAnalysisOp.  
분류 결과에 대한 Error matrix를 계산한다.

ClassifyErrorAnalysisOp() -Constructor for class ClassifyErrorAnalysisOp  
Constructor

classNo -Variable in class ISODATAClassifyOp

classNo -Variable in class KMeansClassifyOp

classNo -Variable in class KMeansUnSupervisedClassifyOp

clear() -Method in class ScattergramOp  
스캐터그램 배열을 초기화한다.

clear() -Method in class AOI.Graph

clear() -Method in class DXFtoVectorOp.FeatureProtocol  
벡터정보를 제거한다.

clear() -Method in class GeoVector  
Feature를 모두 목록에서 제거한다.

clear() -Method in class HistogramOp  
히스토그램을 초기화한다.

clone() -Method in class Cube  
현재의 영상을 복사하여 되돌린다.

clone() -Method in class CubeHeader  
현재의 헤더 정보를 복사한다.

CLOSE -Static variable in class UnaryMorphologicalFunctionOp

close() -Method in class ImageMaker  
화면 구성고 관련된 객체를 제거한다.

CmdTable -Static variable in class GeoEnviron  
명령어 관리 테이블

collectAllGarbages() -Static method in class GeoUtility  
시스템의 카미지 콜렉터를 실행한다.

COLOR -Static variable in class DXFtoVectorOp

COLOR -Static variable in class DXFVertexExtractOp

ColorConvertOp -class ColorConvertOp.  
색 좌표계의 변환을 수행한다.

ColorConvertOp() -Constructor for class ColorConvertOp  
Constructor

CommandLoader -class CommandLoader.  
선택된 명령어에 대한 객체를 가져온다

CommandLoader() -Constructor for class CommandLoader  
Constructor

commandSelected(CommandEvent) -Method in class geopixel

COMMENT -Static variable in class CubeHeader  
영상에 대한 주석

comparePixelValue(byte[], int, byte[], int, int) -Method in class UnaryLogicalFunctionOp

comparePixelValue(float[], int, byte[], int, int) - Method in class UnaryLogicalFunctionOp

comparePixelValue(int[], int, byte[], int, int) - Method in class UnaryLogicalFunctionOp

comparePixelValue(short[], int, byte[], int, int) - Method in class UnaryLogicalFunctionOp

convert(byte[][][], int, float[][][], int, int) - Method in class ColorConvertOp

색좌표를 계산한다.

convert(double[], int, float[], int, int) - Method in class TM6ToTemperatureOp

convertRadiance(double) - Method in class TM6ToTemperatureOp

CONVEX - Static variable in class UnaryMorphologicalFunctionOp

convolve(double[][][]) - Method in class ConvolveOp  
미리 정의된 커널과 입력배열과의 콘볼루션 값을 구한다.

ConvolveOp - class ConvolveOp.

커널과 배열에 대한 콘볼루션을 실행한다.

ConvolveOp(GeoKernel, int) - Constructor for class ConvolveOp  
Constructor

CORRELATION - Static variable in class UnaryTextureFunctionOp

count() - Method in class AOI.Graph

count() - Method in class Feature

count() - Method in class Feature.Point

count() - Method in class GeoVector

목록의 Feature의 수를 되돌린다.

CrossMark - class CrossMark.

CrossMark를 표시한다.

CROSSMARK - Static variable in interface Drawable

CrossMark(int, int) - Constructor for class CrossMark  
Constructor

CrossMark(int, int, int) - Constructor for class CrossMark  
Constructor

Cube - class Cube.

영상에 대한 정보를 관리하는 클래스이다.

CUBE\_MODE - Static variable in class Target

Cube() - Constructor for class Cube  
영상을 선언한다.

Cube(int, int, int, int, int, int, int) - Constructor for class Cube  
주어진 변수에 대한 영상을 선언한다.

CubeFileToMemoryOp - class CubeFileToMemoryOp.

영상을 메모리에 등록한다.

CubeFileToMemoryOp() - Constructor for class CubeFileToMemoryOp

Constructor  
CubeGeometryFlipOp - class CubeGeometryFlipOp.  
 영상에 대하여 수평, 수직 또는 전이를 실행한다.  
CubeGeometryFlipOp() - Constructor for class CubeGeometryFlipOp  
 Constrctor  
CubeGeometryResizeOp - class CubeGeometryResizeOp.  
 영상의 크기를 재조절한다.  
CubeGeometryResizeOp() - Constructor for class CubeGeometryResizeOp  
 Constructor  
CubeGeometryRotateOp - class CubeGeometryRotateOp.  
 영상을 임의의 각도로 회전한다.  
CubeGeometryRotateOp() - Constructor for class CubeGeometryRotateOp  
 Constrctor  
CubeGeometryTranslateOp - class CubeGeometryTranslateOp.  
 영상을 x, y방향으로 이동한다.  
CubeGeometryTranslateOp() - Constructor for class  
CubeGeometryTranslateOp  
 Constrctor  
CubeHeader - class CubeHeader.  
  
CubeHeader() - Constructor for class CubeHeader  
 영상의 헤더를 선언한다.  
CubeHeader(int, int, int, int) - Constructor for class CubeHeader  
 영상의 헤더를 선언한다.  
CubeHeader(String) - Constructor for class CubeHeader  
 영상의 헤더를 선언한다.  
CubeHeader(String, int) - Constructor for class CubeHeader  
 영상의 헤더를 선언한다.  
CubeHeader(String, int, int, int, int) - Constructor for class CubeHeader  
 영상의 헤더를 선언한다.  
CubeListDlg - Static variable in class GeoEnviron  
 영상 목록 관리 창  
CubeMemoryToFileOp - class CubeMemoryToFileOp.  
 메모리 타입의 영상을 파일 타입으로 변환한다.  
CubeMemoryToFileOp() - Constructor for class CubeMemoryToFileOp  
 Constructor  
CubeSubsetOp - class CubeSubsetOp.  
 부분 영상을 추출한다.  
CubeSubsetOp() - Constructor for class CubeSubsetOp  
 Constructor  
CUBIC - Static variable in class GeoResampler  
 Cubic법  
Cubic(byte[][][], byte[][][], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 바이트형 배열에 대한 Cubic 재배열을 실행한다.  
Cubic(float[][][], float[][][], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 4바이트 실수형 배열에 대한 Cubic 재배열을 실행한다.  
Cubic(int[][][], int[][][], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 4바이트 정수형 배열에 대한 Cubic 재배열을 실행한다.  
Cubic(short[][][], short[][][], boolean[], float[], float[], int, int) -  
 Method in class GeoResampler  
 2바이트 정수형 배열에 대한 Cubic 재배열을 실행한다.  
cumFreq - Variable in class HistogramModifyEnhanceOp

CurrentIndex - Variable in class AOILayer

CurrentPath - Static variable in class GeoEnviron  
사용자 디렉토리

---

## D

D\_0 - Static variable in class UnaryTextureFunctionOp

D\_135 - Static variable in class UnaryTextureFunctionOp

D\_180 - Static variable in class UnaryTextureFunctionOp

D\_225 - Static variable in class UnaryTextureFunctionOp

D\_270 - Static variable in class UnaryTextureFunctionOp

D\_315 - Static variable in class UnaryTextureFunctionOp

D\_45 - Static variable in class UnaryTextureFunctionOp

D\_90 - Static variable in class UnaryTextureFunctionOp

degreeToRadian(double) - Method in class UnaryMathFunctionOp

Description - Variable in class Supervisor

DILATE - Static variable in class UnaryMorphologicalFunctionOp

DisplayEnhanceOp - class DisplayEnhanceOp.  
화면에 대한 영상향상을 실행한다.

DisplayEnhanceOp(DisplayEnhanceMethod) - Constructor for class DisplayEnhanceOp  
Constructor

dispose() - Method in class Target

dispose() - Method in class Target.File  
Target을 삭제한다.

dispose() - Method in class Target.Memory  
Target을 삭제한다.

DIVIDE - Static variable in class UnaryArithmeticFunctionOp

DIVIDE - Static variable in class BinaryArithmeticFunctionOp  
나눗셈 연산자

dlgActionPerformed(DlgActionEvent) - Method in class TargetLister

DNToRadianceOp - class DNToRadianceOp.  
영상의 회소값을 휘도값으로 변환한다.

DNToRadianceOp() - Constructor for class DNToRadianceOp  
Constructor

DNToReflectanceOp - class DNToReflectanceOp.

영상의 회소값을 센서의 반사도값으로 변환한다.  
DNToReflectanceOp() - Constructor for class DNToReflectanceOp  
 Constructor  
draw(Graphics, float) - Method in class AOI.Graph  
draw(Graphics, float) - Method in class CrossMark  
 CrossMark를 그린다.  
draw(Graphics, float) - Method in interface Drawable  
draw(Graphics, float) - Method in class LabeledCrossMark  
 CrossMark와 Label을 그린다.  
draw(Graphics, int, int) - Method in class AOI.Graph  
draw(Graphics, int, int) - Method in class CrossMark  
 CrossMark를 그린다.  
draw(Graphics, int, int) - Method in interface Drawable  
draw(Graphics, int, int) - Method in class LabeledCrossMark  
 CrossMark와 Label을 그린다.  
Drawable - interface Drawable.  
DVI - Static variable in class VegetationIndexOp  
dviValue(double[][][], int, float[], int, int) - Method in class VegetationIndexOp  
DXFtoVectorOp - class DXFtoVectorOp.  
 DXF 포맷으로 부터 내부 벡터자료(GeoVector)를 구성한다.  
DXFtoVectorOp.FeatureProtocol - class DXFtoVectorOp.FeatureProtocol.  
 Feature를 관리한다.  
DXFtoVectorOp.FeatureProtocol(DXFtoVectorOp) - Constructor for class DXFtoVectorOp.FeatureProtocol  
 Constructor  
DXFtoVectorOp.VHeader - class DXFtoVectorOp.VHeader.  
 벡터의 헤더정보를 관리한다.  
DXFtoVectorOp.VHeader(DXFtoVectorOp) - Constructor for class DXFtoVectorOp.VHeader  
DXFtoVectorOp() - Constructor for class DXFtoVectorOp  
 Constructor  
DXFVertexExtractOp - class DXFVertexExtractOp.  
 DXF파일에서 Vertex를 추출한다.  
DXFVertexExtractOp() - Constructor for class DXFVertexExtractOp  
 Constructor

---

## E

EDGE\_NO\_OP - Static variable in class SpatialKernelFilterOp  
EDGE\_ZERO\_FILL - Static variable in class SpatialKernelFilterOp  
elements() - Method in class AOI.Graph

elements() - Method in class Feature

elements() - Method in class Feature.Point

elements() - Method in class GeoVector

벡터 element를 되돌린다.

embody() - Method in class Target.Memory

영상을 저장할 메모리 영역을 확보한다.

EmbossingKernel - class EmbossingKernel.

Embossing필터 커널을 관리한다.

EmbossingKernel(String) - Constructor for class EmbossingKernel  
Constructor

ENERGY - Static variable in class UnaryTextureFunctionOp

ENTROPHY - Static variable in class UnaryTextureFunctionOp

EQUAL - Static variable in class UnaryLogicalFunctionOp

EQUAL - Static variable in class BinaryLogicalFunctionOp

= 연산자

EQUAL - Static variable in class BinaryPixelValueReplaceOp

= 연산자

equals(Feature.Point) - Method in class Feature.Point

동일한 객체인지를 검사한다.

equals(Object) - Method in class Target.File

입력 파일과의 동일성 여부를 검사한다.

equals(Object) - Method in class Target.Memory

입력 파일과의 동일성 여부를 검사한다.

equals(Spot) - Method in class Spot

equals(Volume) - Method in class Volume

ERODE - Static variable in class UnaryMorphologicalFunctionOp

execute() - Method in class TargetLister

execute() - Method in class Supervisor

existHDFFile() - Method in class Target.File

헤더 파일이 존재하는가를 검사한다.

ExtXmax - Variable in class DXFtoVectorOp.VHeader

ExtXmin - Variable in class DXFtoVectorOp.VHeader

ExtYmax - Variable in class DXFtoVectorOp.VHeader

ExtYmin - Variable in class DXFtoVectorOp.VHeader

---

## F



Feature -class Feature.  
Feature를 관리한다.

FEATURE\_ID -Static variable in class VectorLayer

Feature.Line -class Feature.Line.

Feature.Line() -Constructor for class Feature.Line

Feature.Point -class Feature.Point.  
Point feature를 관리한다.

Feature.Point() -Constructor for class Feature.Point  
Constructor

Feature.Point(double, double) -Constructor for class Feature.Point  
Point를 설정한다.

Feature.Point(Feature.Point) -Constructor for class Feature.Point  
Point를 설정한다.

Feature.Polygon -class Feature.Polygon.

Feature.Polygon() -Constructor for class Feature.Polygon

Feature() -Constructor for class Feature

FeatureCount -Variable in class DXFtoVectorOp.VHeader

FFT1D(float[], float[], float[], float[], int) -Method in class FFTOP

FFT2D(float[][][], int, int, int) -Method in class FFTOP

FFTOP -class FFTOP.

영상의 fast fourier transform을 수행한다. forward 변환일 경우 대상영상은 메모리형이어야 하고 출력 결과는 real, imagery가 제1분광대, 제2분광대로 저장된다.

FFTOP() -Constructor for class FFTOP  
Constructor

FFTPowerSpectrumOp -class FFTPowerSpectrumOp.

영상의 FFT 결과를 이용하여 파워스펙트럼 영상을 생성한다.

FFTPowerSpectrumOp() -Constructor for class FFTPowerSpectrumOp  
Constructor

FLOAT -Static variable in class Cube  
4바이트 부동소수점형을 정의한다.

FLOAT -Static variable in class GeoKernel

Freq -Variable in class ScattergramOp

---

## G

GCPCorrectOp -class GCPCorrectOp.  
영상의 GCP보정을 실행한다.

GCPCorrectOp() -Constructor for class GCPCorrectOp  
Constructor

GcpRegressor -class GcpRegressor.

GCP를 이용한 변환 계수를 계산한다.  
GcpRegressor(double[][][], int) - Constructor for class GcpRegressor  
 Constructor

GeoArrayTypeConverter - class GeoArrayTypeConverter.

GeoArrayTypeConverter() - Constructor for class GeoArrayTypeConverter  
 주어진 화소형의 배열을 다른 화소형의 배열로 변환한다.

GeoDataInputStream - class GeoDataInputStream.  
 파일 입력 스트림을 정의한다.

GeoDataInputStream(InputStream) - Constructor for class GeoDataInputStream  
 Constructor

GeoDataOutputStream - class GeoDataOutputStream.  
 파일 출력 스트림을 정의한다.

GeoDataOutputStream(OutputStream) - Constructor for class GeoDataOutputStream  
 Constructor

GeoEnviron - class GeoEnviron.

GeoEnviron() - Constructor for class GeoEnviron

GeoException - exception GeoException.  
 Exception과 관련된 메시지를 출력한다.

GeoException() - Constructor for class GeoException  
 Constructor

GeoException(String) - Constructor for class GeoException  
 Constructor

GeoFileIO TextField - class GeoFileIO TextField.  
 영상이외의 파일 객체를 관리한다.

GeoFileIO TextField() - Constructor for class GeoFileIO TextField  
 Constructor

GeoFileIO TextField(File) - Constructor for class GeoFileIO TextField  
 Constructor

GeoKernel - class GeoKernel.  
 필터링과 관련된 커널을 관리한다.

GeoKernel(int, int, float[]) - Constructor for class GeoKernel  
 Create a Kernel object from an array of floats.

GeoKernel(int, int, float[], int) - Constructor for class GeoKernel  
 Constructor

GeoKernel(int, int, int[]) - Constructor for class GeoKernel  
 Constructor

GeoKernel(int, int, int[], int) - Constructor for class GeoKernel  
 Constructor

GeoKeyTable - class GeoKeyTable.  
 키워드를 관리한다.

GeoKeyTable() - Constructor for class GeoKeyTable  
 Constructor

GeoKeyTable(GeoCommandTable) - Constructor for class GeoKeyTable  
 Constructor

GeoKeyTextField - class GeoKeyTextField.

GeoKeyTextField() - Constructor for class GeoKeyTextField

GeoKeyTextField(String) - Constructor for class GeoKeyTextField

GeoLookupTable -class GeoLookupTable.  
화소값의 변환을 위한 참조표를 관리한다.

GeoLookupTable() -Constructor for class GeoLookupTable  
Constructor

GeoLookupTable(String, int[][]) -Constructor for class GeoLookupTable  
Constructor

GeoMenuParser -class GeoMenuParser.

AttributeParser2 class 를 사용해서 Menu Description file 을 parsing 한다

GeoMenuParser(File) -Constructor for class GeoMenuParser  
file name 을 넣어서 GeoMenuParser 를 구동시킨다.

GeoMenuReader -class GeoMenuReader.

메뉴 Description file 을 읽어서 Menu 를 만드는 클래스이다.

GeoMenuReader(MenuBar, int, ActionListener) -Constructor for class GeoMenuReader

GeoMenuWriter -class GeoMenuWriter.

GeoMenuWriter(MenuBar, int) -Constructor for class GeoMenuWriter

geopixel -class geopixel.

geopixel() -Constructor for class geopixel

GeoRectangle -class GeoRectangle.

사각 영역을 정의한다.

GeoRectangle() -Constructor for class GeoRectangle  
Constructor

GeoRectangle(double, double, double, double) -Constructor for class GeoRectangle  
Constructor

Constructor

GeoRectangle(GeoRectangle) -Constructor for class GeoRectangle  
Constructor

GeoResampler -class GeoResampler.

출력 좌표에 대한 화소값의 추정한다.

GeoResampler() -Constructor for class GeoResampler  
Constructor

GeoUtility -class GeoUtility.

응용 클래스를 관리한다.

GeoUtility() -Constructor for class GeoUtility

GeoVector -class GeoVector.

벡터 클래스를 정의한다.

GeoVector() -Constructor for class GeoVector  
Constructor

GeoVector(String) -Constructor for class GeoVector  
Constructor

getAbsoluteError() -Method in class Regressor  
절대 오차를 되돌린다.

getAOI(int) -Method in class AOILayer

getBand() -Method in class ImageLoade  
분광대를 되돌린다.

getBandName(int) -Method in class CubeHeader

영상의 분광대 목록에서 주어진 인덱스의 분광대 이름을 되돌린다.

getBandNameForList(int) -Method in class CubeHeader

영상의 목록에서 주어진 인덱스의 분광대 이름을 되돌린다.

getBias() - Method in class GcpRegressor  
변환 결과의 수평/수직 이동량을 되돌린다.

getBound() - Method in class ScattergramOp  
스캐터그램의 화소값 영역을 되돌린다.

getBounds() - Method in class CrossMark  
영역을 되돌린다.

getBounds() - Method in interface Drawable

getBounds() - Method in class LabeledCrossMark  
CrossMark의 경계를 되돌린다.

getBounds(float) - Method in class CrossMark  
Scaling된 좌표의 영역을 되돌린다.

getBounds(float) - Method in interface Drawable

getBounds(float) - Method in class LabeledCrossMark  
CrossMark의 축척에 대한 경계를 되돌린다.

getBounds(int, int) - Method in class CrossMark  
x, y 방향으로 이동된 좌표의 영역을 되돌린다.

getBounds(int, int) - Method in interface Drawable

getBounds(int, int) - Method in class LabeledCrossMark  
CrossMark의 이동된 경계를 되돌린다.

getBundleByteSize() - Method in class Cube  
영상에 대해 한 라인에 대해 전체 분광대를 처리 할 수 있는 바이트 크기를 되돌린다.

getByteSize() - Method in class Cube  
영상의 전체 바이트 크기를 되돌린다.

getCapacity() - Method in class VectorLayer  
저장된 벡터의 수를 되돌린다.

getCapacity() - Method in class AOILayer

getChildren() - Method in class MenuToken  
현재 구성된 MenuToken 의 Child 가 있는 Vector 를 가져오는데 보통 Menu 또는 MenuItem 을 구성하기위해 쓰거나 Serial 하게 구성되어있는 MenuToken 을 실제 메뉴형태인 Tree 구조로 바꾸기 위해서 특정 MenuToken 과 그 Children 을 가져온다.

getClassname() - Method in class MenuToken  
Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 classname 을 가져온다

getCoef() - Method in class Regressor  
계수를 되돌린다.

getColor(Feature) - Method in class VectorLayer  
객체의 색을 되돌린다.

getColor(int) - Method in class VectorLayer  
인덱스에 해당하는 색을 되돌린다.

getColor(int) - Method in class AOILayer

getCommand(String) - Static method in class GeoEnviron  
이력 ID를 갖는 명령어를 되돌린다.

getCorrel(int, int, int) - Method in class Statistics  
클래스의 분광대에 대한 공상관 값을 되돌린다.

getCorrelation() - Method in class UnaryTextureFunctionOp

getCovar(int, int, int) - Method in class Statistics

클래스의 분광대에 대한 공분산 값을 되돌린다.

getCumulativeFreq() - Method in class HistogramOp  
지정된 분광대에 대한 누적 농도 분포를 되돌린다.

getCumulativeFreq(int) - Method in class HistogramOp  
지정된 분광대의 화소값에 대한 누적 농도 분포를 되돌린다.

getCurrentMainFrame() - Static method in class GeoEnviron  
현재 설정되어 있는 주프레임을 되돌린다.

getDivisor() - Method in class SobelKernel  
콘볼루션 수행에 대한 분모 연산자를 되돌린다.

getDivisor() - Method in class EmbossingKernel  
콘볼루션에 대한 가중치를 되돌린다.

getDivisor() - Method in class GeoKernel  
divisor를 되돌린다.

getDivisor() - Method in class LaplacianKernel  
콘볼루션 수행에 대한 분모 연산자를 되돌린다.

getDivisor() - Method in class OutlierKernel  
콘볼루션 수행에 대한 분모 연산자를 되돌린다.

getDivisor() - Method in class ParmHighPassKernel  
콘볼루션 수행에 대한 분모 연산자를 되돌린다.

getDivisor() - Method in class ParmLowPassKernel  
콘볼루션 수행에 대한 분모 연산자를 되돌린다.

getDivisor() - Method in class PrewittKernel  
콘볼루션 수행에 대한 분모 연산자를 되돌린다.

getDKernel() - Method in class SobelKernel  
Sobel(D) Kernel 배열의 가중치를 되돌린다.

getDownKernel() - Method in class EmbossingKernel  
Down 방향에 대한 필터배열을 되돌린다.

getEdgeHint() - Method in class SpatialKernelFilterOp  
엣지 처리와 관련된 내용을 되돌린다.

getEditState() - Method in class AOILayer

getEKernel() - Method in class PrewittKernel  
Prewitt(E)Kernel 배열의 가중치를 되돌린다.

getEnergy() - Method in class UnaryTextureFunctionOp

getEntropy() - Method in class UnaryTextureFunctionOp

getError() - Method in class Regressor  
오차를 되돌린다.

getErrorMatrix() - Method in class ClassifyErrorAnalysisOp  
Error matrix를 되돌린다.

getEstimate() - Method in class Regressor

getExtendedFilename(String, String) - Static method in class GeoUtility  
파일의 확장자를 확인하여 유효한 파일명을 되돌린다.

getFeature(int) - Method in class GeoVector  
인덱스에 해당하는 feature를 되돌린다.

getFeatures() - Method in class GeoVector  
Feature배열을 되돌린다.

getFile() - Method in class GeoFileIOTextField  
설정된 파일 객체를 되돌린다.

getFreq() - Method in class ScattergramOp  
스캐터그램을 되돌린다.

getFreq() - Method in class HistogramOp  
지정된 인덱스의 히스토그램을 되돌린다.

getFreq(float) - Method in class HistogramOp  
 화소값에 대한 빈도수를 되돌린다.

getFreq(float, float) - Method in class ScattergramOp  
 화소값 x, y에 대한 빈도수를 되돌린다.

getFreq(int) - Method in class HistogramOp  
 인덱스의 히스토그램을 되돌린다.

getFreq(int, int) - Method in class ScattergramOp  
 인덱스 x, y에 대한 빈도수를 되돌린다.

getGCPEstimate() - Method in class GcpRegressor  
 변환식을 이용하여 GCP의 위치를 예측한다.

getGeoCoord() - Method in class CubeHeader  
 영상의 x, y 방향에 대한 최소, 최대값을 갖는 영역을 되돌린다.

getHeight() - Method in class GeoKernel  
 Returns the height

getHKernel() - Method in class LaplacianKernel  
 Laplacian (H)Kernel 배열의 가중치를 되돌린다.

getHomogeneity() - Method in class UnaryTextureFunctionOp

getId() - Method in class MenuItem  
 Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuItem 의 id 값을 가져온다

getID() - Method in class Target

getID() - Method in class Target.File  
 파일의 절대 경로를 되돌린다.

getID() - Method in class Target.Memory  
 Target의 이름을 되돌린다.

getID() - Method in class Feature  
 ID를 되돌린다.

getImage(String) - Method in class ImageProcessor  
 파일로부터 이미지를 생성하여 되돌린다.

getKernel() - Method in class SpatialKernelFilterOp  
 현재 설정되어 있는 커널을 되돌린다.

getKernel() - Method in class SobelKernel  
 현재 설정되어 있는 Sobel Kernel 배열의 가중치를 되돌린다.

getKernel() - Method in class EmbossingKernel  
 필터 배열을 되돌린다.

getKernel() - Method in class LaplacianKernel  
 현재 설정되어 있는 Laplacian Kernel 배열의 가중치를 되돌린다.

getKernel() - Method in class OutlierKernel  
 Outlier Kernel을 되돌린다.

getKernel() - Method in class ParMHighPassKernel  
 커널 배열을 되돌린다.

getKernel() - Method in class ParMLowPassKernel  
 커널 배열을 되돌린다.

getKernel() - Method in class PrewittKernel  
 현재 설정되어 있는 Prewitt Kernel 배열의 가중치를 되돌린다.

getKernelData(float[]) - Method in class GeoKernel  
 Returns the kernel data in row major order.

getKernelData(int[]) - Method in class GeoKernel  
 Returns the kernel data in row major order.

getKey() - Method in class MenuItem  
 Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuItem 의 key 값을 가져온다

getKurtosis() - Method in class HistogramOp

지정된 농도 분포에 대한 Kurtosis값을 되돌린다.  
getKurtosis(int) - Method in class HistogramOp  
 농도 분포에 대한 Kurtosis값을 되돌린다.

getLabel() - Method in class MenuToken  
 Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 label 값을 가져온다

getLeftKernel() - Method in class EmbossingKernel  
 Left 방향에 대한 필터배열을 되돌린다.

getLevel(float) - Method in class HistogramOp  
 퍼센트값에 대한 화소값을 되돌린다.

getLine(boolean[], int, int) - Method in class AOI

getLine(boolean[], int, int) - Method in class AOI.Bit

getLine(boolean[], int, int) - Method in class AOI.Graph

getLKernel() - Method in class SobelKernel  
 Sobel(L) Kernel 배열의 가중치를 되돌린다.

getLocation() - Method in class CrossMark  
 CrossMark의 위치를 되돌린다.

getLocation() - Method in interface Drawable

getLocation() - Method in class LabeledCrossMark  
 CrossMark의 위치를 되돌린다.

getLookupTable(int) - Method in class ImageMaker  
 인덱스에 해당하는 화면출력 변환값을 되돌린다.

getLRKernel() - Method in class LaplacianKernel  
 Laplacian (LR)Kernel 배열의 가중치를 되돌린다.

getMainCommandFrame() - Static method in class GeoEnvir  
 주명령어 프레임을 되돌린다.

getMainFreq(int) - Method in class ImageMaker  
 주 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

getMainImage(Rectangle) - Method in class ImageMaker  
 주 화면의 선택영역에 대한 이미지를 되돌린다.

getMainMenuFrame() - Static method in class GeoEnviron  
 주메뉴 프레임을 되돌린다.

getMainViewFreq(int) - Method in class ImageMaker  
 주 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

getMainViewRegion(Rectangle, float) - Static method in class ViewCalc

getMax() - Method in class Feature

getMax() - Method in class Feature.Point  
 포인트의 최대 좌표값을 되돌린다.

getMax() - Method in class MinMaxOp  
 각 분광대의 최대값 배열을 되돌린다.

getMax(int) - Method in class MinMaxOp  
 특정 분광대의 최대값을 되돌린다.

getMaxFreq() - Method in class ScattergramOp  
 스캐터그램에서 최대 빈도수를 되돌린다.

getMaxFreq() - Method in class HistogramOp  
 최대 빈도수를 되돌린다.

getMaxLimit() - Method in class GeoVector  
 point feature의 최대 영역 범위를 되돌린다.

getMean() - Method in class Statistics

평균을 되돌린다.  
getMean() - Method in class HistogramOp  
 농도 분포의 평균을 되돌린다.  
getMean(int, int) - Method in class Statistics  
 클래스의 분광대에 대한 평균을 되돌린다.  
getMin() - Method in class Feature  
  
getMin() - Method in class Feature.Point  
 포인트의 최소 좌표값을 되돌린다.  
getMin() - Method in class MinMaxOp  
 각 분광대의 최소값 배열을 되돌린다.  
getMin(int) - Method in class MinMaxOp  
 특정 분광대의 최소값을 되돌린다.  
getMinLimit() - Method in class GeoVector  
 point feature의 최소 영역 범위를 되돌린다.  
getMinMaxCoord() - Method in class GcpRegressor  
 최소, 최대 좌표를 되돌린다.  
getMode() - Method in class GeoKernel  
 연산에 관련된 화소형(float/int)를 되돌린다.  
getMode() - Method in class ImageMaker  
 화면 출력 모드를 되돌린다.  
getMoment() - Method in class UnaryTextureFunctionOp  
  
getName() - Method in class Target  
  
getName() - Method in class Target.File  
 Target의 이름을 되돌린다.  
getName() - Method in class Target.Memory  
 Target의 이름을 되돌린다.  
getNEKernel() - Method in class PrewittKernel  
 Prewitt(NE)Kernel 배열의 가중치를 되돌린다.  
getNKernel() - Method in class PrewittKernel  
 Prewitt(N)Kernel 배열의 가중치를 되돌린다.  
getNoOfBands() - Method in class Statistics  
 분광대 수를 되돌린다.  
getNoOfClasses() - Method in class Statistics  
 클래스 수를 되돌린다.  
getNoOfPixels(int) - Method in class Statistics  
 클래스의 화소수를 되돌린다.  
getNWKernel() - Method in class PrewittKernel  
 Prewitt(NW)Kernel 배열의 가중치를 되돌린다.  
getOmniKernel() - Method in class LaplacianKernel  
 Laplacian (Omni)Kernel 배열의 가중치를 되돌린다.  
getOriginalValue(int, int, int[]) - Method in class ImageMaker  
 x, y좌표에서의 원 화소값을 되돌린다.  
getOrigMainPixels(int) - Method in class ImageMaker  
 인덱스에 대한 주화면의 원 화소값을 되돌린다.  
getOrigScrollPixels(int) - Method in class ImageMaker  
 인덱스에 대한 스크롤 화면의 원 화소값을 되돌린다.  
getPalette() - Method in class VectorLayer  
 색상표를 되돌린다.  
getPalette() - Method in class ImageMaker  
 색상표를 되돌린다.  
getParent() - Method in class MenuToken  
 Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 parent



값을 가져온다

getParm() - Method in class ParmHighPassKernel  
가중치 parameter 값을 되돌린다.

getParm() - Method in class ParmLowPassKernel  
가중치 parameter 값을 되돌린다.

getPathOfHDFile() - Method in class Target.File  
헤더 파일의 경로를 되돌린다.

getPixelByteSize(int) - Static method in class Cube  
화소형의 바이트 크기를 되돌린다.

getPixelIndex() - Method in class SpatialProfileViewOp  
각 화소값의 x, y좌표를 되돌린다.

getPixelTypeNo(String) - Static method in class Cube  
영상의 화소형을 되돌린다.

getPixelTypeString(int) - Static method in class Cube  
영상의 화소형을 되돌린다.

getPixelValue() - Method in class SpectrumProfileViewOp  
분광대별 화소값을 되돌린다.

getPixelValue() - Method in class SpatialProfileViewOp  
화소값을 되돌린다.

getPixelValue() - Method in class PixelInteractionViewOp  
지정된 위치에서의 화소값 배열을 되돌린다.

getPixelValue() - Method in class PixelSpectrumViewOp  
지정된 위치에서의 화소값 배열을 되돌린다.

getPolygon() - Method in class AOI.Graph

getPolygon(Feature, float) - Method in class VectorLayer  
축척에 대한 좌표를 갖는 폴리곤을 되돌린다.

getPolygon(Feature, int, int) - Method in class VectorLayer  
영상 좌표를 갖는 폴리곤을 되돌린다.

getPolygon(int) - Method in class AOI.Graph

getPosition(double, double) - Method in class GcpRegressor  
입력 좌표에 대한 변환 좌표를 되돌린다.

getPromotedType(int) - Static method in class Cube  
영상의 화소형을 되돌린다.

getRadius() - Method in class CrossMark  
CrossMark의 반경을 되돌린다.

getRange(int, int) - Method in class Statistics  
클래스의 분광대에 대한 화소값 범위를 되돌린다.

getRightKernel() - Method in class EmbossingKernel  
Right 방향에 대한 필터배열을 되돌린다.

getRKernel() - Method in class SobelKernel  
Sobel(R)Kernel 배열의 가중치를 되돌린다.

getRLKernel() - Method in class LaplacianKernel  
Laplacian (RL)Kernel 배열의 가중치를 되돌린다.

getRMS() - Method in class GcpRegressor  
변환식에 의한 각 좌표의 rms를 되돌린다.

getScaledGeoCoord(Rectangle) - Method in class CubeHeader  
입력 범위에 대하여 새로운 좌표값을 계산하여 되돌린다.

getScreenValue(int, int, int[]) - Method in class ImageMaker  
x, y좌표에서의 화면 화소값을 되돌린다.

getScrollBar(Rectangle, float) - Static method in class ViewCalc

getScrollFreq(int) - Method in class ImageMaker  
스크롤 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

getScrollImage(int) - Method in class ImageMaker  
주어진 축척에 대한 스크롤 이미지를 되돌린다.  
getScrollViewFreq(int) - Method in class ImageMaker  
스크롤 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.  
getSEKernel() - Method in class PrewittKernel  
Prewitt(SE)Kernel 배열의 가중치를 되돌린다.  
getSharedValues(Vector) - Method in class GeoKeyTable

getSize() - Method in class AOI.Bit

getSKernel() - Method in class PrewittKernel  
Prewitt(S)Kernel 배열의 가중치를 되돌린다.  
getSkewness() - Method in class HistogramOp  
지정된 농도 분포에 대한 Skewness 값을 되돌린다.  
getSkewness(int) - Method in class HistogramOp  
농도 분포에 대한 Skewness 값을 되돌린다.  
getStatistic() - Method in class ISODATAClassifyOp  
분류 결과에 대한 통계를 되돌린다.  
getStatistic() - Method in class KMeansClassifyOp  
분류 결과에 대한 통계를 되돌린다.  
getStatistic() - Method in class KMeansUnSupervisedClassifyOp  
분류 결과에 대한 통계를 되돌린다.  
getStatistics() - Method in class StatisticCalculateOp  
통계 결과를 되돌린다.  
getStd() - Method in class Statistics  
표준 편차를 되돌린다.  
getStd() - Method in class HistogramOp  
농도 분포의 표준편차를 되돌린다.  
getStd(int, int) - Method in class Statistics  
클래스의 분광대에 대한 표준편차를 되돌린다.  
getSum(int, int) - Method in class Statistics  
클래스의 분광대에 대한 Sum을 되돌린다.  
getSWKernel() - Method in class PrewittKernel  
Prewitt(SW)Kernel 배열의 가중치를 되돌린다.  
getTarget() - Method in class ImageLoader  
영상을 되돌린다.  
getTarget() - Method in class ImageProcessor  
처리된 영상을 되돌린다.  
getTitle() - Method in class Statistics  
타이틀을 되돌린다.  
getTitle() - Method in class ScattergramOp  
스캐터그램의 타이틀을 되돌린다.  
getTitle() - Method in class AOI

getTitle() - Method in class DXFtoVectorOp  
Title을 되돌린다.  
getTitle() - Method in class GeoLookupTable  
타이틀을 되돌린다.  
getTitle() - Method in class GeoVector  
벡터의 타이틀을 되돌린다.  
getTitle() - Method in class HistogramOp  
히스토그램의 타이틀을 설정한다.  
getTotalFreq() - Method in class ScattergramOp  
전체 빈도수를 되돌린다.  
getTotalFreq() - Method in class HistogramOp

총 화소수를 되돌린다.  
getTotalRMS() - Method in class GcpRegressor  
 변환식에 의한 전체 평균 rms를 되돌린다.  
getType() - Method in class AOI

getType() - Method in class CrossMark  
 CrossMark의 type을 되돌린다.  
getType() - Method in interface Drawable

getType() - Method in class Feature

getType() - Method in class Feature.Point  
 Feature의 타입을 되돌린다.  
getType() - Method in class Feature.Line

getType() - Method in class Feature.Polygon

getType() - Method in class LabeledCrossMark  
 CrossMark의 종류를 되돌린다.  
getUKernel() - Method in class SobelKernel  
 Sobel(U) Kernel 배열의 가중치를 되돌린다.  
getUpKernel() - Method in class EmbossingKernel  
 Up 방향에 대한 필터배열을 되돌린다.  
getVar(int, int) - Method in class Statistics  
 클래스의 분광대에 대한 분산을 되돌린다.  
getVector(int) - Method in class VectorLayer  
 인덱스에 벡터를 되돌린다.  
getVertex(int) - Method in class Feature

getVertex(int) - Method in class Feature.Point

getVertexs() - Method in class Feature

getVertexs() - Method in class Feature.Point

getVKernel() - Method in class LaplacianKernel  
 Laplacian (V)Kernel 배열의 가중치를 되돌린다.  
getVolume() - Method in class Cube  
 현재 정의된 영상과 같은 크기를 갖는 영상을 생성, 되돌린다.  
getWidth() - Method in class GeoKernel  
 Returns the width  
getWKernel() - Method in class PrewittKernel  
 Prewitt(W)Kernel 배열의 가중치를 되돌린다.  
getXmax() - Method in class DXFtoVectorOp.FeatureProtocol  
 X좌표의 최대값을 되돌린다.  
getXmin() - Method in class DXFtoVectorOp.FeatureProtocol  
 X좌표의 최소값을 되돌린다.  
getXOrigin() - Method in class GeoKernel  
 Returns the X origin.  
getYmax() - Method in class DXFtoVectorOp.FeatureProtocol  
 Y좌표의 최대값을 되돌린다.  
getYmin() - Method in class DXFtoVectorOp.FeatureProtocol  
 Y좌표의 최소값을 되돌린다.  
getYOrigin() - Method in class GeoKernel  
 Returns the Y origin.

GRAPH - Static variable in class AOI

GREATER - Static variable in class UnaryLogicalFunctionOp

GREATER - Static variable in class BinaryLogicalFunctionOp  
> 연산자

GREATER - Static variable in class BinaryPixelValueReplaceOp  
> 연산자

GREATER\_THAN - Static variable in class UnaryLogicalFunctionOp

GREATER\_THAN - Static variable in class BinaryLogicalFunctionOp  
>= 연산자

GREATER\_THAN - Static variable in class BinaryPixelValueReplaceOp  
>= 연산자

---

## H

handleEvent(Event) - Method in class SunParameterCanvas

handlePixels(Image, int, int, int, int, int[]) - Method in class ImageProcessor  
이미지 화소 색상값을 배열로 저장한다.

hasChildren() - Method in class MenuItem

현재 MenuItem 의 Object 가 Child 의 Object 가 있는 지를 check 한다

hasMoreTokens() - Method in class GeoMenuParser

현재 부르는 파서에 대해 앞으로 남은 MenuItem 이 남아 있는 지를 넘겨준다  
nextMenuItem() 을 무한히 looping 할때 그 조건으로 넣어주는 함수다

HD - Variable in class Target

height - Variable in class Volume

help(String) - Static method in class GeoEnviron

사용자 입력창에서의 도움말 기능을 지원한다.

HistogramMatchOp - class HistogramMatchOp.

두개의 영상에 대한 히스토그램 일치화를 실행한다.

HistogramMatchOp() - Constructor for class HistogramMatchOp  
Constructor

HistogramMatchOp(int[], int[]) - Constructor for class HistogramMatchOp  
Constructor

Constructor

HistogramModifyEnhanceOp - class HistogramModifyEnhanceOp.

농도 분포 변형을 통한 영상향상을 실행한다.

HistogramModifyEnhanceOp() - Constructor for class HistogramModifyEnhanceOp  
Constructor

Constructor

HistogramOp - class HistogramOp.

영상의 히스토그램을 작성한다.

HistogramOp() - Constructor for class HistogramOp  
Constructor

HistoryCount - Static variable in class GeoEnviron

명령어 저장 크기

HIT\_MISS - Static variable in class UnaryMorphologicalFunctionOp

HOMOGENITY - Static variable in class UnaryTextureFunctionOp

HorizontalFlip(byte[], int, byte[][][], int, int, int) - Method in class CubeGeometryFlipOp

바이트형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(byte[], int, byte[], int) - Method in class CubeGeometryFlipOp

바이트형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(float[], int, float[][][], int, int, int) - Method in class CubeGeometryFlipOp

4바이트 실수형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(float[], int, float[], int) - Method in class CubeGeometryFlipOp

4바이트 실수형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(int[], int, int[][][], int, int, int) - Method in class CubeGeometryFlipOp

4바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(int[], int, int[], int) - Method in class CubeGeometryFlipOp

4바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(short[], int, short[][][], int, int, int) - Method in class CubeGeometryFlipOp

2바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

HorizontalFlip(short[], int, short[], int) - Method in class CubeGeometryFlipOp

2바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

## I

ID - Variable in class DXFtoVectorOp.FeatureProtocol

ID - Variable in class Feature

IdealBPFilteredFrequency(float[][][], int) - Method in class FFTOp

IdealBSFilteredFrequency(float[][][], int) - Method in class FFTOp

IdealHPFilteredFrequency(float[][][], int) - Method in class FFTOp

IdealLPFilteredFrequency(float[][][], int) - Method in class FFTOp

II - Static variable in class VegetationIndexOp

iiValue(double[][][], int, float[], int, int) - Method in class VegetationIndexOp

IllegalAttributeException - exception IllegalAttributeException.

IllegalAttributeException에 대한 메시지를 처리한다.

IllegalAttributeException() - Constructor for class IllegalAttributeException  
Constructor

IllegalAttributeException(String) - Constructor for class

IllegalAttributeException

Constructor

IllegalFormatException - exception IllegalFormatException.

IllegalFormatException에 대한 메시지를 처리한다.

IllegalFormatException() - Constructor for class IllegalFormatException

Constructor

IllegalFormatException(String) - Constructor for class

IllegalFormatException

Constructor

IllegalValueException - exception IllegalValueException.

IllegalFormatException에 대한 메시지를 처리한다.

IllegalValueException() - Constructor for class IllegalValueException

Constructor

IllegalValueException(String) - Constructor for class IllegalValueException

Constructor

IMAGE\_TO\_IMAGE - Static variable in class GCPCorrectOp

IMAGE\_TO\_MAP - Static variable in class GCPCorrectOp

ImageLoader - class ImageLoader.

화면출력시 영상을 읽어 드리는 클래스.

ImageLoader(Target, int) - Constructor for class ImageLoader

Constructor

ImageLoadOp - class ImageLoadOp.

화면출력시 영상을 읽어 드리는 클래스.

ImageLoadOp(Target, int) - Constructor for class ImageLoadOp

Constructor

ImageMaker - class ImageMaker.

화면출력을 위한 이미지를 작성한다.

ImageMaker(Target, int) - Constructor for class ImageMaker

Constructor

ImageMaker(Target, int, Target, int, Target, int) - Constructor for

class ImageMaker

Constructor

ImageProcessor - class ImageProcessor.

그래픽 포맷의 입출력을 관리한다.

ImageProcessor() - Method in class ImageProcessor

Constructor

ImageProcessor() - Constructor for class ImageProcessor

indexOf(Feature.Point) - Method in class Feature

indexOf(Feature.Point) - Method in class Feature.Point

indexOf(Feature) - Method in class GeoVector

Feature의 인덱스를 되돌린다.

indexOf(int, int) - Method in class Feature

indexOf(int, int) - Method in class Feature.Point

indexOf(Polygon) - Method in class AOI.Graph

inflate(int, int, int) - Method in class Volume

inflate(Volume) - Method in class Volume

info() – Method in class Target  
return the content of target header

init() – Method in class GeoVector

initialize() – Static method in class GeoEnviron  
GeoPixel의 각종 환경 변수를 초기화 하고 메뉴프레임과 명령어 프레임을 가동한다.

initialize(double) – Method in class Target.Memory  
영상의 화소값을 초기화 한다.

insertFeature(int, Feature) – Method in class GeoVector  
feature를 삽입한다.

insertPolygon(int, Polygon) – Method in class AOI.Graph

insertVertex(int, double, double) – Method in class Feature

insertVertex(int, double, double) – Method in class Feature.Point

insertVertex(int, Feature.Point) – Method in class Feature

insertVertex(int, Feature.Point) – Method in class Feature.Point

insideRegion(int, int, int) – Method in class CubeHeader  
입력된 화소 위치가 영상의 유효범위에 있는지를 판단한다.

INT – Static variable in class Cube  
4바이트 정수형을 정의한다.

INT – Static variable in class GeoKernel

InteactiveEnhanceOp – class InteactiveEnhanceOp.  
대화형 영상향상을 실행한다.

InteactiveEnhanceOp() – Constructor for class InteactiveEnhanceOp  
Constructor

Intervals – Variable in class HistogramModifyEnhanceOp

IntervalsX – Variable in class ScattergramOp

IntervalsY – Variable in class ScattergramOp

IPVI – Static variable in class VegetationIndexOp

ipviValue(double[[[], int, float[], int, int) – Method in class VegetationIndexOp

isBit(File) – Static method in class AOI

isEmpty() – Method in class CubeHeader  
영상의 헤더정보가 정의 되어 있는지 판단한다.

isEmpty() – Method in class GeoFileIO.TextField  
설정된 파일 객체를 검사한다.

isEmpty(int) – Method in class VectorLayer  
인덱스의 벡터가 비어있는지를 검사한다.

isEmpty(int) – Method in class AOILayer

isLayerOrderColorMode() – Method in class VectorLayer  
layer 순의 색지정 모드인지를 검사한다.

ISODATAClassifyOp - class ISODATAClassifyOp.  
ISODATA 분류를 실행한다.  
ISODATAClassifyOp() - Constructor for class ISODATAClassifyOp  
Constructor  
isPolygonFill() - Method in class VectorLayer  
폴리곤 채움 모드인지를 검사한다.  
isSameVolume(Cube) - Method in class Cube  
입력 영상과 같은 폴리곤 크기를 갖는지 검사한다.  
isShift() - Method in class MenuToken  
Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 shift 의 boolean 값을 가져온다  
isValidPixelType(int) - Static method in class Cube  
화소형의 유효 여부를 검사한다.  
isVisible(int) - Method in class VectorLayer  
인덱스의 레이어가 표시됨 모드인지를 검사한다.  
isVisible(int) - Method in class AOILayer

---

## K

KeyTable - Static variable in class GeoEnviron  
명령어 키워드 관리 테이블  
KMeansClassifyOp - class KMeansClassifyOp.  
K-means 방법에 의한 분류를 실행한다.  
KMeansClassifyOp() - Constructor for class KMeansClassifyOp  
Constructor  
KMeansUnSupervisedClassifyOp - class KMeansUnSupervisedClassifyOp.  
무감독에 의한 K-means 분류를 실행한다.  
KMeansUnSupervisedClassifyOp() - Constructor for class  
KMeansUnSupervisedClassifyOp  
Constructor  
KMeansUnSupervisedClassifyOp(String) - Constructor for class  
KMeansUnSupervisedClassifyOp  
Constructor

---

## L

Label - Variable in class LabeledCrossMark  
LABELED\_CROSSMARK - Static variable in interface Drawable  
LabeledCrossMark - class LabeledCrossMark.  
화면에 출력되는 CrossMark를 관리한다.  
LabeledCrossMark(int, int) - Constructor for class LabeledCrossMark  
x, y 위치에 CrossMark를 출력한다.  
LabeledCrossMark(int, int, String) - Constructor for class  
LabeledCrossMark  
x, y 위치에 입력 Label를 갖는 CrossMark를 출력한다.  
LANDSATTMImportOp - class LANDSATTMImportOp.



Landsat TM 영상을 내부 포맷으로 변환 저장한다.  
LANDSATTMImportOp() - Constructor for class LANDSATTMImportOp  
 Constructor  
LaplacianKernel - class LaplacianKernel.  
 Laplacian Kernel를 관리한다.  
LaplacianKernel(String) - Constructor for class LaplacianKernel  
 Constructor  
LAYER\_ORDER - Static variable in class VectorLayer  
LAYER\_ORDER - Static variable in class DXFtoVectorOp  
LAYER\_ORDER - Static variable in class DXFVertexExtractOp  
LBS - Static variable in class UnaryBitwiseFunctionOp  
length - Variable in class Volume  
LESS - Static variable in class UnaryLogicalFunctionOp  
LESS - Static variable in class BinaryLogicalFunctionOp  
 < 연산자  
LESS - Static variable in class BinaryPixelValueReplaceOp  
 < 연산자  
LESS\_THAN - Static variable in class UnaryLogicalFunctionOp  
LESS\_THAN - Static variable in class BinaryLogicalFunctionOp  
 <= 연산자  
LESS\_THAN - Static variable in class BinaryPixelValueReplaceOp  
 <= 연산자  
LevelSliceOp - class LevelSliceOp.  
 Density level slicing을 실행한다.  
LevelSliceOp() - Constructor for class LevelSliceOp  
 Constructor  
LimitXmax - Variable in class DXFtoVectorOp.VHeader  
LimitXmin - Variable in class DXFtoVectorOp.VHeader  
LimitYmax - Variable in class DXFtoVectorOp.VHeader  
LimitYmin - Variable in class DXFtoVectorOp.VHeader  
LINE - Static variable in class Feature  
Lines - Variable in class AOI.Bit  
Lines - Variable in class Cube  
 영상의 Lines값을 정의한다.  
LnOffset - Variable in class Cube  
 영상의 Lines에 대한 오프셋을 정의한다.  
load(int, byte[]) - Method in class ImageLoader  
 축척에 대한 영상을 읽어 들인다.  
load(Rectangle, byte[]) - Method in class ImageLoader  
 처리 영역에 대한 영상을 읽어 들인다.  
loadClass(String) - Method in class CommandLoader  
 명령어에 대한 클래스를 읽어 들인다.

loadClass(String, boolean) - Method in class CommandLoader  
명령어에 대한 클래스를 읽어들인다.

LookupReplaceOp - class LookupReplaceOp.  
참조표를 이용하여 배열의 값을 재구성한다.

LookupReplaceOp() - Constructor for class LookupReplaceOp  
Constructor

LookupReplaceOp(byte[]) - Constructor for class LookupReplaceOp  
Constructor

---

## M

main(String[]) - Static method in class AOI

main(String[]) - Static method in class GeoKeyTextField

main(String[]) - Static method in class geopixel

MainViewSize - Static variable in class GeoEnviron  
주 윈도우의 초기 크기값

makeD0\_GLCM() - Method in class UnaryTextureFunctionOp

makeD180\_GLCM() - Method in class UnaryTextureFunctionOp

makeD270\_GLCM() - Method in class UnaryTextureFunctionOp

makeD90\_GLCM() - Method in class UnaryTextureFunctionOp

MAP\_TO\_MAP - Static variable in class GCPCorrectOp

Mark - Variable in class LabeledCrossMark

MAX\_SIZE - Static variable in class VectorLayer

MAX\_SIZE - Static variable in class AOILayer

maxFreq - Variable in class ScattergramOp

Mean - Variable in class ISODATAClassifyOp

Mean - Variable in class KMeansClassifyOp

Mean - Variable in class KMeansUnSupervisedClassifyOp

Mean - Variable in class MinDistanceClassifyOp

MenuToken - class MenuToken.

Menu Description file 에서 읽어진 token 을 가지고 MenuToken object 를  
만들게 된다 Menu Display 하는 데이터 구조의 기본 클래스이다.

MenuToken() - Constructor for class MenuToken

message(Window, Exception) - Static method in class GeoEnviron  
Exception 발생에 대하여 메시지를 출력한다.

message(Window, String) – Static method in class GeoEnviron  
 사용자 인터럽트에 대하여 메시지를 출력한다.

MinDistanceClassifyOp – class MinDistanceClassifyOp.  
 최소거리 분류법을 실행한다.

MinDistanceClassifyOp() – Constructor for class MinDistanceClassifyOp  
 Constructor

MinMaxEnhanceOp – class MinMaxEnhanceOp.  
 최소-최대값을 이용한 선형 영상 향상을 실행한다.

MinMaxEnhanceOp() – Constructor for class MinMaxEnhanceOp  
 Constructor

MinMaxLinearEnhanceOp – class MinMaxLinearEnhanceOp.  
 최소-최대값을 이용한 선형 영상 향상을 실행한다.

MinMaxLinearEnhanceOp() – Constructor for class MinMaxLinearEnhanceOp  
 Constructor

MinMaxOp – class MinMaxOp.  
 영상의 각 분광대의 최소, 최대값을 구한다.

MinMaxOp() – Constructor for class MinMaxOp  
 Constructor

MLClassifyOp – class MLClassifyOp.  
 최대 우도법에 의한 분류를 실행한다.

MLClassifyOp() – Constructor for class MLClassifyOp  
 Constructor

MOMENT – Static variable in class UnaryTextureFunctionOp

move(int, int, int) – Method in class Spot

move(Spot) – Method in class Spot

moveInside(Rectangle, Rectangle) – Static method in class ViewCalc

MSAVI – Static variable in class VegetationIndexOp

MSAVI2 – Static variable in class VegetationIndexOp

msaviValue(double[][][], int, float[], int, int) – Method in class VegetationIndexOp

MSI – Static variable in class VegetationIndexOp

msiValue(double[][][], int, float[], int, int) – Method in class VegetationIndexOp

MultiCubeMergeOp – class MultiCubeMergeOp.  
 여러 분광대를 갖는 복수영상을 정렬하여 하나의 영상으로 재 구성한다.

MultiCubeMergeOp() – Constructor for class MultiCubeMergeOp  
 Constructor

MultiImageComposeOp – class MultiImageComposeOp.  
 영상을 다른 영상에 끼워 넣기를 한다.

MultiImageComposeOp() – Constructor for class MultiImageComposeOp  
 Constructor

MULTIPLY – Static variable in class UnaryArithmeticFunctionOp

MULTIPLY – Static variable in class BinaryArithmeticFunctionOp  
 곱셈 연산자

---

## N

NDVI - Static variable in class VegetationIndexOp

ndviValue(double[[[], int, float[], int, int) - Method in class VegetationIndexOp

NEAREST\_NEIGHBOR - Static variable in class GeoResampler

최근린근사법

NearestNeighbor(byte[[[], byte[[[], boolean[], float[], float[], int, int) - Method in class GeoResampler

바이트형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

NearestNeighbor(float[[[], float[[[], boolean[], float[], float[], int, int) - Method in class GeoResampler

4바이트 실수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

NearestNeighbor(int[[[], int[[[], boolean[], float[], float[], int, int) - Method in class GeoResampler

4바이트 정수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

NearestNeighbor(short[[[], short[[[], boolean[], float[], float[], int, int) - Method in class GeoResampler

2바이트 정수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

NeoPoly - Variable in class AOILayer

nextMenuToken() - Method in class GeoMenuParser

File에서 MenuToken을 하나씩 하나씩 얻어낼때 쓴다.

NoFilteredFrequency(float[[[], int) - Method in class FFTOP

NONE - Static variable in class VegetationIndexOp

NONE - Static variable in class UnaryArithmeticFunctionOp

NONE - Static variable in class UnaryMorphologicalFunctionOp

NONE - Static variable in class UnaryLogicalFunctionOp

NONE - Static variable in class UnaryBitwiseFunctionOp

NONE - Static variable in class UnaryTextureFunctionOp

NONE - Static variable in class TasseledCapTransformOp

NONE - Static variable in class BinaryArithmeticFunctionOp

NONE - Static variable in class BinaryLogicalFunctionOp

NONE - Static variable in class BinaryPixelValueReplaceOp

NONE - Static variable in class DNToRadianceOp

NONE - Static variable in class DNToReflectanceOp

NoOfBands - Variable in class MinDistanceClassifyOp

NoOfClasses – Variable in class MinDistanceClassifyOp

NOT – Static variable in class UnaryBitwiseFunctionOp

NOT\_EQUAL – Static variable in class UnaryLogicalFunctionOp

NOT\_EQUAL – Static variable in class BinaryLogicalFunctionOp

NOT\_EQUAL – Static variable in class BinaryPixelValueReplaceOp

NotMatchedException – exception NotMatchedException.

NotMatchedException() – Constructor for class NotMatchedException

NotMatchedException(String) – Constructor for class NotMatchedException

NotMatchedException – exception NotMatchedException.  
type이 맞지 않을 경우에 대한 예외처리 클래스이다.

NotMatchedException() – Constructor for class NotMatchedException

NotMatchedException(String) – Constructor for class NotMatchedException

NullImageMakeOp – class NullImageMakeOp.  
영상을 초기화한 빈 영상을 작성한다.

NullImageMakeOp() – Constructor for class NullImageMakeOp

---

## O

ObjectLayer – class ObjectLayer.

ObjectLayer() – Constructor for class ObjectLayer

on(int, int) – Method in class AOI

on(int, int) – Method in class AOI.Bit

on(int, int) – Method in class AOI.Graph

on(Point) – Method in class AOI

on(Point) – Method in class AOI.Bit

on(Point) – Method in class AOI.Graph

OPEN – Static variable in class UnaryMorphologicalFunctionOp

OpenFileDialog - Static variable in class GeoEnviron  
파일 열기 입력창

operate() - Method in class VegetationIndexOp  
식생 지수를 계산한다.

operate() - Method in class UnaryArithmeticFunctionOp  
입력영상의 분광대에 대한 산술연산을 실행한다.

operate() - Method in class UnaryMorphologicalFunctionOp  
입력영상의 분광대에 대한 Morphology 연산을 실행한다.

operate() - Method in class UnaryMathFunctionOp  
입력영상의 분광대에 대한 수학 연산을 실행한다.

operate() - Method in class UnaryLogicalFunctionOp  
입력영상의 분광대에 대한 논리 비교 연산을 실행한다.

operate() - Method in class UnaryBitwiseFunctionOp  
입력영상의 분광대에 대한 비트 이동 연산을 실행한다.

operate() - Method in class UnaryTextureFunctionOp  
Texture 추출을 실행한다.

operate() - Method in class TM6ToTemperatureOp  
온도 변환을 실행한다.

operate() - Method in class ThinningOp  
세선화를 실행한다.

operate() - Method in class TerrainPropertyOp  
지형 요소를 계산한다.

operate() - Method in class TasseledCapTransformOp  
Tasseled Cap 변환을 실행한다.

operate() - Method in class TopographicDeEffectOp  
반사면 모델을 이용하여 지형 효과를 제거한다.

operate() - Method in class StatisticCalculateOp  
영상에 대한 통계를 구한다.

operate() - Method in class SPOTHRVImportOp  
SPOT HRV 영상을 내부 포맷으로 변환한다.

operate() - Method in class SpectrumProfileViewOp  
분광 프로파일을 작성한다.

operate() - Method in class SpatialProfileViewOp  
두점사이의 분광 프로파일을 작성한다.

operate() - Method in class SpatialKernelFilterOp  
필터링을 실행한다.

operate() - Method in class ScattergramOp  
스캐터그램 작성을 실행한다.

operate() - Method in class SceneRegressNormalizeOp  
영상 정규화를 실행한다.

operate() - Method in class BilToBipConvertOp  
포맷 변환을 실행한다.

operate() - Method in class BilToBsqConvertOp  
포맷 변환을 실행한다.

operate() - Method in class BinaryArithmeticFunctionOp  
입력영상의 두 분광대에 대한 산술연산을 실행한다.

operate() - Method in class BinaryLogicalFunctionOp  
입력영상의 두 분광대에 대한 논리 비교 연산을 실행한다.

operate() - Method in class BinaryPixelValueReplaceOp  
논리 비교 연산을 실행한다.

operate() - Method in class BipToBilConvertOp  
포맷 변환을 실행한다.

operate() - Method in class BsqToBilConvertOp  
포맷 변환을 실행한다.

operate() - Method in class ChromaticityAnalysisOp

3개 분광대를 이용하여 Chromaticity를 계산한다.

operate() - Method in class ClassifyErrorAnalysisOp  
Error matrix를 계산한다.

operate() - Method in class ColorConvertOp  
색좌표 변환을 실행한다.

operate() - Method in class CubeFileToMemoryOp  
입력영상을 메모리형의 영상으로 변환한다.

operate() - Method in class CubeGeometryFlipOp  
입력영상의 Geometry를 변환한다.

operate() - Method in class CubeGeometryResizeOp  
입력영상의 크기를 변환한다.

operate() - Method in class CubeGeometryRotateOp  
영상을 회전한다.

operate() - Method in class CubeGeometryTranslateOp  
영상을 수평, 수직방향으로 이동한다.

operate() - Method in class CubeMemoryToFileOp  
영상을 파일 타입으로 변환한다.

operate() - Method in class CubeSubsetOp  
부분 영상을 추출한다.

operate() - Method in class DNTToRadianceOp  
화소값의 휘도로의 변환을 한다.

operate() - Method in class DNTToReflectanceOp  
화소값의 반사도값으로의 변환을 한다.

operate() - Method in class DXFtoVectorOp  
내부 벡터로의 변환을 실행한다.

operate() - Method in class DXFVertexExtractOp  
Vertex를 추출한다.

operate() - Method in class FFTOp  
변환을 실행한다.

operate() - Method in class FFTPowerSpectrumOp  
파워스펙트럼을 만든다.

operate() - Method in class GCPCorrectOp  
대상 영상을 설정한다.

operate() - Method in class HistogramMatchOp  
히스토그램 일치화를 실행한다.

operate() - Method in class HistogramModifyEnhanceOp  
히스토그램 변형에 의한 영상향상을 실행한다.

operate() - Method in class HistogramOp  
히스토그램 작성을 실행한다.

operate() - Method in class ImageProcessor  
이미지 포맷의 입력을 실행한다.

operate() - Method in class InteactiveEnhanceOp  
사용자 정의에 의한 영상향상을 실행한다.

operate() - Method in class ISODATAClassifyOp  
ISODATA 분류를 실행한다.

operate() - Method in class KMeansClassifyOp  
K-Means 분류를 실행한다.

operate() - Method in class KMeansUnSupervisedClassifyOp  
무감독 K-Means 분류를 실행한다.

operate() - Method in class LANDSATTMImportOp  
Landsat TM 영상을 내부 포맷으로 변환한다.

operate() - Method in class LevelSliceOp  
Density Slicing을 실행한다.

operate() - Method in class MinDistanceClassifyOp  
최소 거리 분류를 실행한다.

operate() - Method in class MinMaxEnhanceOp  
최소-최대값을 이용한 선형 향상을 실행한다.

operate() - Method in class MinMaxLinearEnhanceOp  
최소-최대값을 이용한 선형 향상을 실행한다.

operate() - Method in class MinMaxOp  
최소-최대값 찾기를 실행한다.

operate() - Method in class MLClassifyOp  
최대우도법 분류를 실행한다.

operate() - Method in class MultiCubeMergeOp  
영상의 merge를 실행한다.

operate() - Method in class MultiImageComposeOp  
영상 삽입을 실행한다.

operate() - Method in class NullImageMakeOp  
빈 영상을 작성한다.

operate() - Method in class PCATransformOp  
주성분 분석을 실행한다.

operate() - Method in class PiecewiseEnhanceOp  
단계적 영상향상을 실행한다.

operate() - Method in class PixelInteractionViewOp  
화소값을 읽어 배열에 저장한다.

operate() - Method in class PixelSpectrumViewOp  
화소값을 읽어 배열에 저장한다.

operate() - Method in class PixelTypeConvertOp  
화소형 변환을 실행한다.

operate(byte[]) - Method in class LookupReplaceOp  
배열의 재구성을 실행한다.

operate(byte[], byte[]) - Method in class DisplayEnhanceOp  
마이트형 영상에 대한 향상을 한다.

operate(byte[], byte[], int, int) - Method in class DisplayEnhanceOp  
마이트형 영상에 대한 향상을 한다.

operate(byte[], int, double[], int, int) - Method in class GeoArrayTypeConverter  
마이트형을 8바이트 실수형으로 변환하여 되돌린다.

operate(byte[], int, float[], int, int) - Method in class GeoArrayTypeConverter  
마이트형을 4바이트 실수형으로 변환하여 되돌린다.

operate(byte[], int, int[], int, int) - Method in class GeoArrayTypeConverter  
마이트형을 4바이트 정수형으로 변환하여 되돌린다.

operate(byte[], int, short[], int, int) - Method in class GeoArrayTypeConverter  
마이트형을 2바이트 정수형으로 변환하여 되돌린다.

operate(float[], byte[]) - Method in class DisplayEnhanceOp  
4바이트 실수형 영상에 대한 향상을 한다.

operate(float[], byte[], float, float) - Method in class DisplayEnhanceOp  
4바이트 정수형 영상에 대한 향상을 한다.

operate(float[], int, double[], int, int) - Method in class GeoArrayTypeConverter  
4바이트 실수형을 8바이트 실수형으로 변환하여 되돌린다.

operate(float[], int, float[], int, int) - Method in class GeoArrayTypeConverter  
4바이트 실수형을 4바이트 실수형으로 변환하여 되돌린다.

operate(int[], byte[]) - Method in class DisplayEnhanceOp  
4바이트 정수형 영상에 대한 향상을 한다.

operate(int[], byte[], int, int) - Method in class DisplayEnhanceOp



4바이트 정수형 영상에 대한 향상을 한다.  
operate(int[], int, double[], int, int) - Method in class GeoArrayTypeConverter  
 4바이트 정수형을 8바이트 실수형으로 변환하여 되돌린다.  
operate(int[], int, int[], int, int) - Method in class GeoArrayTypeConverter  
 4바이트 정수형을 4바이트 정수형으로 변환하여 되돌린다.  
operate(int, byte[]) - Method in class ImageLoadOp  
operate(int, float[]) - Method in class ImageLoadOp  
operate(int, int[]) - Method in class ImageLoadOp  
operate(int, short[]) - Method in class ImageLoadOp  
operate(short[], byte[], boolean) - Method in class DisplayEnhanceOp  
 2바이트 정수형 영상에 대한 향상을 한다.  
operate(short[], byte[], boolean, int, int) - Method in class DisplayEnhanceOp  
 2바이트 정수형 영상에 대한 향상을 한다.  
operate(short[], int, boolean, double[], int, int) - Method in class GeoArrayTypeConverter  
 2바이트 정수형을 8바이트 실수형으로 변환하여 되돌린다.  
operate(short[], int, boolean, int[], int, int) - Method in class GeoArrayTypeConverter  
 2바이트 정수형을 4바이트 정수형으로 변환하여 되돌린다.  
Operator - interface Operator.  
 All of operators must implements this class and these methods  
OutlierKernel - class OutlierKernel.  
 Outlier Kernel을 관리한다.  
OutlierKernel() - Constructor for class OutlierKernel  
 Constructor  
OutOfMemoryException - exception OutOfMemoryException.  
 메모리 부족에 대한 예외처리를 한다.  
OutOfMemoryException() - Constructor for class OutOfMemoryException  
 Constructor  
OutOfMemoryException(String) - Constructor for class OutOfMemoryException  
 Constructor  
OVERLAY - Static variable in class BinaryArithmeticFunctionOp  
 중첩 연산자

---

## P

paint(Graphics) - Method in class SunParameterCanvas

Parent - Variable in class Supervisor

ParmHighPassKernel - class ParmHighPassKernel.

Parametric High Pass Filter Kernel 클래스이다.

ParmHighPassKernel() - Constructor for class ParmHighPassKernel  
 Constructor

ParmHighPassKernel(int) - Constructor for class ParmHighPassKernel  
Constructor

ParmLowPassKernel - class ParmLowPassKernel.  
Parametric Low Pass Filter Kernel 클래스이다.

ParmLowPassKernel() - Constructor for class ParmLowPassKernel  
Constructor

ParmLowPassKernel(int) - Constructor for class ParmLowPassKernel  
Constructor

PCATransformOp - class PCATransformOp.  
주성분 분석을 실행한다.

PCATransformOp() - Constructor for class PCATransformOp  
Constructor

PiecewiseEnhanceOp - class PiecewiseEnhanceOp.  
사용자 정의에 의한 변환표를 이용하여 영상 향상을 한다.

PiecewiseEnhanceOp() - Constructor for class PiecewiseEnhanceOp  
Constructor

Pixel - Variable in class Target.File

Pixel - Variable in class Target.Memory

PixelInteractionViewOp - class PixelInteractionViewOp.  
입력 위치로부터 9x9의 화소값을 숫자로 표현한다.

PixelInteractionViewOp() - Constructor for class PixelInteractionViewOp  
Constructor

PixelSpectrumViewOp - class PixelSpectrumViewOp.  
각 분광대의 화소값을 표현한다.

PixelSpectrumViewOp() - Constructor for class PixelSpectrumViewOp  
Constructor

PixelTable - Variable in class SpectrumProfileViewOp

PixelTable - Variable in class SpatialProfileViewOp

PixelTable - Variable in class PixelInteractionViewOp

PixelTable - Variable in class PixelSpectrumViewOp

PixelType - Variable in class Cube  
영상의 화소형을 정의한다.

PixelTypeConvertOp - class PixelTypeConvertOp.  
입력 화소형을 다른 화소형으로 변환한다.

PixelTypeConvertOp() - Constructor for class PixelTypeConvertOp  
Constructor

POINT - Static variable in class Feature

POLYGON - Static variable in class Feature

POW - Static variable in class UnaryArithmeticFunctionOp

PrewittKernel - class PrewittKernel.  
Prewitt Kernel를 관리한다.

PrewittKernel(String) - Constructor for class PrewittKernel  
Constructor

print() - Method in class DXFtoVectorOp.VHeader  
헤더정보를 화면에 출력한다.

print() - Method in class DXFtoVectorOp.FeatureProtocol

벡터 정보(Vertex)를 화면에 출력한다.  
Property - Static variable in class GeoEnviron  
GeoPixel 환경 관리  
PRUNE - Static variable in class UnaryMorphologicalFunctionOp  
pushImageDisplay(CubeViewOp) - Static method in class GeoEnviron  
화면 출력 오퍼레이터를 관리 목록에 삽입하고 목록에서의 위치를 되돌린다.  
putImage(String, Image, String) - Method in class ImageProcessor  
이미지를 원하는 포맷으로 파일에 출력한다.  
PVI - Static variable in class VegetationIndexOp  
pviValue(double[[[], int, float[], int, int) - Method in class VegetationIndexOp

---

## Q

quit() - Method in class geopixel

---

## R

radianToDegree(double) - Method in class UnaryMathFunctionOp  
RBS - Static variable in class UnaryBitwiseFunctionOp  
RBS0 - Static variable in class UnaryBitwiseFunctionOp  
read(File) - Method in class Statistics  
통계를 파일에서 입력한다.  
read(File) - Method in class AOI  
read(File) - Method in class AOI.Bit  
read(File) - Method in class AOI.Graph  
read(File) - Method in class CubeHeader  
영상의 헤더 정보를 읽는다.  
read(File) - Method in class GeoLookupTable  
변환참조표를 파일에서 입력한다.  
read(File) - Method in class GeoMenuReader  
Menu Description 파일을 넘겨주면 그 파일을 통해 새로 구성된 메뉴바를 얻게된다  
read(File) - Method in class GeoVector  
Feature를 파일에서 입력한다.  
read(float[]) - Method in class GeoDataInputStream  
4바이트 실수형의 자료를 입력한다.  
read(float[], int, int) - Method in class GeoDataInputStream  
4바이트 실수형의 자료를 입력한다.

read(int[]) - Method in class GeoDataInputStream  
4바이트 정수형의 자료를 입력한다.

read(int[], int, int) - Method in class GeoDataInputStream  
4바이트 정수형의 자료를 입력한다.

read(short[]) - Method in class GeoDataInputStream  
2바이트 정수형의 자료를 읽는다.

read(short[], int, int) - Method in class GeoDataInputStream  
2바이트 정수형의 자료를 입력한다.

read(String) - Method in class Statistics  
통계를 파일에서 입력한다.

read(String) - Method in class ScattergramOp  
스캐터그램을 파일에서 입력한다.

read(String) - Method in class AOI

read(String) - Method in class AOI.Bit

read(String) - Method in class AOI.Graph

read(String) - Method in class CubeHeader  
영상의 헤더 정보를 읽는다.

read(String) - Method in class GeoLookupTable  
변환참조표를 파일에서 입력한다.

read(String) - Method in class GeoMenuReader  
Menu Description 파일을 넘겨주면 그 파일을 통해 새로 구성된 메뉴바를 얻게된다

read(String) - Method in class GeoVector  
Feature를 파일에서 입력한다.

read(String) - Method in class HistogramOp  
농도 분포를 파일에서 입력한다.

readCommandObject() - Static method in class GeoEnviron  
명령어 객체를 파일로부터 읽는다.

readFixedString(int) - Method in class GeoDataInputStream  
고정된 길이의 문자열을 입력한다.

readLayername() - Method in class DXFtoVectorOp  
Layer에 대한 벡터를 되돌린다.

readLayername() - Method in class DXFVertexExtractOp  
Layer에 대한 벡터를 되돌린다.

readMenuObject() - Static method in class GeoMenuReader

readSubarea(Rectangle, int) - Method in class Target

readSubarea(Rectangle, int) - Method in class Target.File  
Target으로 부터 정해진 영역을 읽는다.

readSubarea(Rectangle, int) - Method in class Target.Memory  
Target으로 부터 정해진 영역을 읽는다.

regionByEnd(float, float, int) - Method in class HistogramModifyEnhanceOp  
히스토그램 작성을 위한 농도 범위를 설정한다.

regionByEnd(float, float, int) - Method in class HistogramOp  
히스토그램 작성을 위한 농도 범위를 설정한다.

regionByEnd(float, float, int, boolean) - Method in class ScattergramOp  
화소값의 영역을 설정한다.

regionByWidth(float, float, int) - Method in class HistogramModifyEnhanceOp  
히스토그램 작성을 위한 농도 범위를 설정한다.

regionByWidth(float, float, int) - Method in class HistogramOp  
 히스토그램 작성을 위한 농도 범위를 설정한다.

regionByWidth(float, float, int, boolean) - Method in class ScattergramOp  
 화소값의 영역을 설정한다.

regress() - Method in class Regressor  
 회귀식을 계산한다.

Regressor - class Regressor.  
 X좌표에서 Y좌표로의 변환을 위한 회귀식을 구한다.

Regressor() - Constructor for class Regressor  
 Constructor

Regressor(double[][], double[][]) - Constructor for class Regressor  
 Constructor

removeAOI(int) - Method in class AOILayer

removeFeature(double, double) - Method in class GeoVector  
 x, y 좌표에 있는 feature를 삭제한다.

removeFeature(Feature) - Method in class GeoVector  
 feature를 삭제한다.

removeFeature(int) - Method in class GeoVector  
 feature를 삭제한다.

removePolygon(int) - Method in class AOI.Graph

removePolygon(int, int) - Method in class AOI

removePolygon(int, int) - Method in class AOI.Bit

removePolygon(int, int) - Method in class AOI.Graph

removePolygon(Polygon) - Method in class AOI.Graph

removeVector(int) - Method in class VectorLayer  
 인덱스에 벡터를 삭제한다.

removeVertex(double, double) - Method in class Feature

removeVertex(double, double) - Method in class Feature.Point

removeVertex(Feature.Point) - Method in class Feature

removeVertex(Feature.Point) - Method in class Feature.Point

removeVertex(int) - Method in class Feature

removeVertex(int) - Method in class Feature.Point

replaceLine(int, boolean[]) - Method in class AOI.Bit

reproduceMainImage() - Method in class ImageMaker  
 주화면에 대한 이미지를 재구성한다.

reproduceScrollImage() - Method in class ImageMaker  
 스크롤 화면에 대한 이미지를 재구성한다.

requireScroll(Rectangle, int, int) - Static method in class ViewCalc

RGB - Static variable in class ImageMaker

runLengthCode(boolean[], int, int) – Method in class AOI.Bit

RVI – Static variable in class VegetationIndexOp

rviValue(double[][][], int, float[], int, int) – Method in class VegetationIndexOp

---

## S

Samples – Variable in class AOI.Bit

Samples – Variable in class Cube  
영상의 Samples값을 정의한다.

SaveFileDialog – Static variable in class GeoEnviron  
파일 저장 입력창

SAVI – Static variable in class VegetationIndexOp

saviValue(double[][][], int, float[], int, int) – Method in class VegetationIndexOp

ScattergramOp – class ScattergramOp.  
스캐터그램을 만든다.

ScattergramOp() – Constructor for class ScattergramOp  
Constructor

SceneRegressNormalizeOp – class SceneRegressNormalizeOp.  
통계적 방법에 의한 영상 정규화를 실행한다.

SceneRegressNormalizeOp() – Constructor for class SceneRegressNormalizeOp  
Constructor

ScrollViewSize – Static variable in class GeoEnviron  
스크롤 윈도우의 초기 크기값

setAlpha(float) – Method in class HistogramModifyEnhanceOp  
히스토그램의 변형에 필요한 alpha값을 설정한다.

setAngle(float) – Method in class CubeGeometryRotateOp  
영상의 회전각도를 설정한다.

setAOI(int, AOI) – Method in class AOILayer

setAOI(String[]) – Method in class ClassifyErrorAnalysisOp  
AOI 파일의 배열을 설정한다.

setBackground(int) – Method in class ThinningOp  
배경 화소값을 설정한다.

setBackground(int) – Method in class GCPCorrectOp  
배경 화소값을 설정한다.

setBackgroundValue(float) – Method in class NullImageMakeOp  
배경의 화소값을 설정한다.

setBand(int) – Method in class UnaryMorphologicalFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBand(int) – Method in class TM6ToTemperatureOp  
대상 분광대를 설정한다.

setBand(int) – Method in class ThinningOp  
대상 분광대를 설정한다.

setBand(int) - Method in class TerrainPropertyOp  
대상 분광대를 설정한다.

setBand(int) - Method in class SpatialKernelFilterOp  
대상 분광대를 설정한다.

setBand(int) - Method in class SceneRegressNormalizeOp  
분광대를 설정한다.

setBand(int) - Method in class FFTOp  
처리 대상 분광대를 설정한다.

setBand(int) - Method in class HistogramMatchOp  
처리하고자 하는 분광대를 설정한다.

setBand(int) - Method in class HistogramModifyEnhanceOp  
대상 분광대를 설정한다.

setBand(int) - Method in class LevelSliceOp  
처리 대상 분광대를 설정한다.

setBand(int) - Method in class PixelInteractionViewOp  
대상 분광대를 설정한다.

setBandIndex(int) - Method in class HistogramOp  
작성된 히스토그램을 얻고자하는 분광대의 인덱스를 설정한다.

setBandName(int, String) - Method in class CubeHeader  
대상 영상의 주어진 분광대의 이름을 정의한다.

setBands(int) - Method in class NullImageMakeOp  
분광대 수를 설정한다.

setBands(int[]) - Method in class VegetationIndexOp  
대상 분광대를 설정한다.

setBands(int[]) - Method in class UnaryArithmeticFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class UnaryMathFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class UnaryLogicalFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class UnaryBitwiseFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class UnaryTextureFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class TopographicDeEffectOp  
대상 분광대를 설정한다.

setBands(int[]) - Method in class StatisticCalculateOp  
대상 분광대를 설정한다.

setBands(int[]) - Method in class SPOTHRImportOp  
분광대를 설정한다.

setBands(int[]) - Method in class SpectrumProfileViewOp  
대상 분광대를 설정한다.

setBands(int[]) - Method in class BilToBipConvertOp  
처리 대상 분광대를 설정한다.

setBands(int[]) - Method in class BilToBsqConvertOp  
처리 대상 분광대를 설정한다.

setBands(int[]) - Method in class ChromaticityAnalysisOp  
처리 대상 분광대를 설정한다.

setBands(int[]) - Method in class ColorConvertOp  
처리 대상 분광대를 설정한다.

setBands(int[]) - Method in class CubeFileToMemoryOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class CubeGeometryFlipOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int[]) - Method in class CubeGeometryResizeOp

입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class CubeGeometryRotateOp  
 입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class CubeGeometryTranslateOp  
 입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class CubeMemoryToFileOp  
 입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class CubeSubsetOp  
 입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class DNToRadianceOp  
 입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class DNToReflectanceOp  
 입력 영상의 대상 분광대를 설정한다.  
setBands(int[]) - Method in class GCPCorrectOp  
 처리 대상 분광대를 설정한다.  
setBands(int[]) - Method in class HistogramOp  
 처리 대상 분광대를 설정한다.  
setBands(int[]) - Method in class ImageProcessor  
 분광대를 설정한다.  
setBands(int[]) - Method in class InteactiveEnhanceOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class ISODATAClassifyOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class KMeansClassifyOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class KMeansUnSupervisedClassifyOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class LANDSATTMImportOp  
 분광대를 설정한다.  
setBands(int[]) - Method in class MinDistanceClassifyOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class MinMaxEnhanceOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class MinMaxLinearEnhanceOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class MinMaxOp  
 처리 대상 분광대를 설정한다.  
setBands(int[]) - Method in class MLClassifyOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class MultiImageComposeOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class PCATransformOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class PiecewiseEnhanceOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class PixelSpectrumViewOp  
 대상 분광대를 설정한다.  
setBands(int[]) - Method in class PixelTypeConvertOp  
 대상 분광대를 설정한다.  
setBands(int[[[]], int[]) - Method in class MultiCubeMergeOp  
 대상 분광대를 설정한다.  
setBands(int, int) - Method in class ScattergramOp  
 분광대를 설정한다.  
setBands(int, int) - Method in class BinaryArithmeticFunctionOp  
 입력 영상의 대상 분광대를 설정한다.



setBands(int, int) - Method in class BinaryLogicalFunctionOp  
입력 영상의 대상 분광대를 설정한다.

setBands(int, int) - Method in class BinaryPixelValueReplaceOp  
입력 영상의 대상 분광대를 설정한다.

setBoundary(int, int, int, int) - Method in class ImageLoadOp  
처리 영역을 설정한다.

setBoundary(Rectangle) - Method in class ImageLoadOp  
처리 영역을 설정한다.

setBounds(double, double, double, double) - Method in class GeoRectangle  
사각 영역을 설정한다.

setBounds(GeoRectangle) - Method in class GeoRectangle  
사각 영역을 설정한다.

setBounds(int[]) - Method in class BipToBilConvertOp  
처리 영역을 설정한다.

setBounds(int[]) - Method in class BsqToBilConvertOp  
처리 영역을 설정한다.

setBounds(int, int) - Method in class NullImageMakeOp  
영상의 크기를 설정한다.

setBounds(Rectangle) - Method in class VegetationIndexOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class UnaryArithmeticFunctionOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class UnaryMorphologicalFunctionOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class UnaryMathFunctionOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class UnaryLogicalFunctionOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class UnaryBitwiseFunctionOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class UnaryTextureFunctionOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class TM6ToTemperatureOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class ThinningOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class TerrainPropertyOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class TasseledCapTransformOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class TopographicDeEffectOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class StatisticCalculateOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class SPOTHRVIImportOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class SpatialKernelFilterOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class ScattergramOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class SceneRegressNormalizeOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class BilToBipConvertOp  
처리 영역을 설정한다.

setBounds(Rectangle) - Method in class BilToBsqConvertOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class BinaryArithmeticFunctionOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class BinaryLogicalFunctionOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class BinaryPixelValueReplaceOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class ChromaticityAnalysisOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class ColorConvertOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeFileToMemoryOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeGeometryFlipOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeGeometryResizeOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeGeometryRotateOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeGeometryTranslateOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeMemoryToFileOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class CubeSubsetOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class DNToRadianceOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class DNToReflectanceOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class FFTOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class HistogramMatchOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class HistogramModifyEnhanceOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class HistogramOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class InteactiveEnhanceOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class ISODATAClassifyOp  
 처리영역을 설정한다.

setBounds(Rectangle) - Method in class KMeansClassifyOp  
 처리영역을 설정한다.

setBounds(Rectangle) - Method in class KMeansUnSupervisedClassifyOp  
 처리영역을 설정한다.

setBounds(Rectangle) - Method in class LANDSATTMImportOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class LevelSliceOp  
 처리 영역을 설정한다.

setBounds(Rectangle) - Method in class MinDistanceClassifyOp  
 처리영역을 설정한다.

setBounds(Rectangle) - Method in class MinMaxEnhanceOp  
 처리영역을 설정한다.

setBounds(Rectangle) - Method in class MinMaxLinearEnhanceOp

처리영역을 설정한다.  
setBounds(Rectangle) - Method in class MinMaxOp  
 처리 영역을 설정한다.  
setBounds(Rectangle) - Method in class MLClassifyOp  
 처리영역을 설정한다.  
setBounds(Rectangle) - Method in class MultimageComposeOp  
 처리영역을 설정한다.  
setBounds(Rectangle) - Method in class PCATransformOp  
 처리영역을 설정한다.  
setBounds(Rectangle) - Method in class PiecewiseEnhanceOp  
 처리영역을 설정한다.  
setBounds(Rectangle) - Method in class PixelTypeConvertOp  
 처리 영역을 설정한다.  
setChangeRate(float) - Method in class KMeansClassifyOp  
 이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다.  
setChangeRate(float) - Method in class KMeansUnsupervisedClassifyOp  
 이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다.  
setClassCount(int) - Method in class ISODATAClassifyOp  
 한 군집이 가질 수 있는 최대 화소수를 설정한다.  
setClassCount(int) - Method in class KMeansClassifyOp  
 분류 군집 수를 설정한다.  
setClassCount(int) - Method in class KMeansUnsupervisedClassifyOp  
 분류 군집 수를 설정한다.  
setClassID(int) - Method in class ISODATAClassifyOp  
  
setClassMean(float[][]) - Method in class KMeansClassifyOp  
 군집의 평균값을 설정한다.  
setClassname(String) - Method in class MenuToken  
 Parser 가 file 로 부터 얻어들이는 classname 을 MenuToken 의 classname 로 Setting 한다.  
setClassNo(int) - Method in class KMeansClassifyOp  
 분류 군집 수를 설정한다.  
setClassStd(float[][]) - Method in class KMeansClassifyOp  
 군집의 표준편차값을 설정한다.  
setCoefficient(double[][][], double[]) - Method in class GCPCorrectOp  
 변환식의 계수를 설정한다.  
setColor(Color) - Method in class SunParameterCanvas  
  
setColor(Color) - Method in class CrossMark  
 CrossMark의 색을 설정한다.  
setColor(Color) - Method in interface Drawable  
  
setColor(Color) - Method in class LabeledCrossMark  
 CrossMark의 색상을 설정한다.  
setColor(int, Color) - Method in class AOILayer  
  
setColorMode(int) - Method in class VectorLayer  
 색지정 모드를 설정한다.  
setConstant(float) - Method in class UnaryArithmeticFunctionOp  
 적용 상수를 설정한다.  
setConstant(float) - Method in class UnaryLogicalFunctionOp  
 비교를 위한 값을 설정한다.  
setConstant(float) - Method in class BinaryPixelValueReplaceOp

비교값이 참인 경우 새로운 화소값으로 대체할 값을 설정한다.

setConstant(int) - Method in class UnaryBitwiseFunctionOp  
적용 상수를 설정한다.

setCoordConverter(GeoCoordConverter) - Method in class VectorLayer  
좌표 변환계를 설정한다.

setCubicOrder(float) - Method in class GeoResampler  
Cubic법의 차수를 설정한다.

setCumulativeFreq() - Method in class HistogramOp  
지정된 분광대에 대한 누적 농도 분포를 구한다.

setCumulHistogram(int, int[]) - Method in class HistogramModifyEnhanceOp  
누적 히스토그램을 설정한다.

setCurrentMainFrame(GeoMainFrame) - Static method in class GeoEnviron  
현재의 주프레임을 설정한다.

setCutDistance(float) - Method in class KMeansClassifyOp  
군집에 포함될 임계 거리값을 설정한다.

setCutDistance(float) - Method in class MinDistanceClassifyOp  
군집에 포함될 임계 거리값을 설정한다.

setDescription(String) - Method in class Supervisor  
Description of executor

setDirection(String) - Method in class UnaryTextureFunctionOp  
조직 추출 방향을 설정한다.

setDistance(int) - Method in class UnaryTextureFunctionOp  
GLCM연산을 위한 거리를 설정한다.

setDistanceMethod(String) - Method in class ISO/DATAClassifyOp  
분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

setDistanceMethod(String) - Method in class KMeansClassifyOp  
분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

setDistanceMethod(String) - Method in class KMeansUnSupervisedClassifyOp  
분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

setDistanceMethod(String) - Method in class MinDistanceClassifyOp  
분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

setDivisor(int) - Method in class SpatialKernelFilterOp  
divisor를 설정한다.

setDivisor(int) - Method in class ConvolveOp  
컨볼루션 결과에 대한 최종값을 결정하기 위한 가중치를 설정한다.

setDXF(File) - Method in class DXFtoVectorOp  
DXF 파일을 설정한다.

setDXF(File) - Method in class DXFVertexExtractOp  
DXF 파일을 설정한다.

setEarthSunDistance(float) - Method in class DNToReflectanceOp  
태양과 지구와의 거리비 값을 설정한다.

setEdgeHint(int) - Method in class SpatialKernelFilterOp  
엣지 처리와 관련된 내용을 설정한다.

setEditState(int) - Method in class AOILayer

setEigenVector(double[][][]) - Method in class PCATransformOp  
고유벡터 행렬을 설정한다.

setElementKernel(int[][]) - Method in class UnaryMorphologicalFunctionOp  
연산을 위한 Kernel을 입력한다.

setFeatureIDMode(int, int) - Method in class DXFtoVectorOp  
Feature의 ID를 세팅한다.

setFile(File) - Method in class GeoFileIOTextField

파일 객체를 설정한다.

setFile(String) - Method in class GeoFileIO TextField  
파일명을 설정한다.

setFilterOrder(float) - Method in class FFTOP  
필터에 사용할 order를 설정한다.

setFilterType(String) - Method in class FFTOP  
역방향 변환에서 사용할 필터를 설정한다.

setGeoBounds(GeoRectangle) - Method in class GCPCorrectOp  
처리 영역을 설정한다.

setGeoCoord(double, double, double, double) - Method in class CubeHeader  
입력 범위에 대하여 영상의 x, y 방향에 대한 최소, 최대값을 설정한다.

setGeoCoord(GeoRectangle) - Method in class CubeHeader  
입력 범위에 대하여 영상의 x, y 방향에 대한 최소, 최대값을 설정한다.

setGreyColorModel() - Method in class VectorLayer  
칼라 모델을 설정한다.

setGreyColorModel() - Method in class ImageMaker  
흑백이미지 구성을 위한 칼라모델을 설정한다.

setGreyLevel(int) - Method in class UnaryTextureFunctionOp  
GLCM연산을 위한 레벨을 설정한다.

setHeader(LANDSAT TMHeader) - Method in class LANDSAT TMImportOp  
LANDSAT TM의 헤더를 설정한다.

setHeader(SPO THRVHeader) - Method in class SPO THRVImportOp  
SPOT HRV의 헤더를 설정한다.

setHighCutFrequency(float) - Method in class FFTOP  
필터의 high cut freq.를 설정한다.

setHistogram(int[], int[], int) - Method in class HistogramMatchOp  
히스토그램을 설정한다.

setHistogramFileName(String) - Method in class HistogramOp  
히스토그램 저장 파일을 설정한다.

setId(int) - Method in class Feature  
ID를 설정한다.

setId(String) - Method in class MenuToken  
Parser 가 file 로 부터 얻어들이는 id 값을 MenuToken 의 id 로 Setting 한다.

setImage() - Method in class SunParameterCanvas

setImageSize(int, int) - Method in class GCPCorrectOp  
영상의 크기를 설정한다.

setInitStatistic(float[[]], float[[]]) - Method in class ISODATAClassifyOp  
초기 통계를 설정한다.

setInitStatistic(float[[]], float[[]]) - Method in class KMeansUnSupervisedClassifyOp

setIteration(int) - Method in class ISODATAClassifyOp  
분류를 반복 수행하기 위한 횟수를 설정한다.

setIteration(int) - Method in class KMeansClassifyOp  
분류를 반복 수행하기 위한 횟수를 설정한다.

setIteration(int) - Method in class KMeansUnSupervisedClassifyOp  
분류를 반복 수행하기 위한 횟수를 설정한다.

setKernel(GeoKernel, int) - Method in class SpatialKernelFilterOp  
커널을 설정한다.

setKey(int) - Method in class MenuToken  
Parser 가 file 로 부터 얻어들이는 key 값을 MenuToken 의 key 로 Setting 한다.

setLabel(String) - Method in class LabeledCrossMark

CrossMark의 Label을 설정한다.

setLabel(String) - Method in class MenuToken  
Parser 가 file 로 부터 얻어들이 Label 값을 MenuToken 의 Label 로 Setting 한다.

setLimit(double, double, double, double) - Method in class GeoVector  
두 point의 최소-최대 좌표를 설정한다.

setLimit(Feature.Point, Feature.Point) - Method in class GeoVector  
두 point feature의 최소-최대 좌표를 설정한다.

setLocate() - Method in class SunParameterPanel

setLocation(int, int) - Method in class CrossMark  
CrossMark의 위치를 설정한다.

setLocation(int, int) - Method in interface Drawable

setLocation(int, int) - Method in class LabeledCrossMark  
CrossMark의 위치를 설정한다.

setLocation(int, int, int) - Method in class Cube  
영상에서의 물리적 위치를 지정한다.

setLookupTable(byte[]) - Method in class LookupReplaceOp  
참조표를 설정한다.

setLookupTable(int, byte[]) - Method in class ImageMaker  
인덱스에 해당하는 화면출력 변환값을 설정한다.

setLowCutFrequency(float) - Method in class FFTOP  
필터의 low cut freq.를 설정한다.

setLUT(int[][]) - Method in class InteactiveEnhanceOp  
사용자에 의하여 정의된 변환표를 설정한다.

setLUT(int[][]) - Method in class PiecewiseEnhanceOp  
영상 향상 변환표를 설정한다.

setMapping(int) - Method in class GCPCCorrectOp  
변환 방법을 설정한다.

setMax(double, double) - Method in class Feature

setMax(double, double) - Method in class Feature.Point

setMax(int, int, double) - Method in class Statistics  
클래스의 분광대에 대한 최대값을 설정한다.

setMaxChangeRate(float) - Method in class ISODATAClassifyOp  
이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다.

setMaxClassSD(float) - Method in class ISODATAClassifyOp  
분류하고자 하는 클래스의 표준편차를 설정한다.

setMaxFreq() - Method in class ScattergramOp  
스캐터그램에서 최대 빈도수를 설정한다.

setMaxFreq() - Method in class HistogramOp  
최대 빈도수를 설정한다.

setMaxLimit(double, double) - Method in class GeoVector  
최대 x, y좌표를 설정한다.

setMaxLimit(Feature.Point) - Method in class GeoVector  
point feature의 최대 좌표를 설정한다.

setMethod(int) - Method in class TM6ToTemperatureOp  
온도 추정 방법을 설정한다.

setMethod(int) - Method in class TasseledCapTransformOp  
변환 계수 설정을 위한 방법을 설정한다.

setMethod(String) - Method in class VegetationIndexOp  
식생지수 계산법을 설정한다.

setMethod(String) - Method in class UnaryArithmeticFunctionOp  
 연산에 대한 방법을 설정한다.

setMethod(String) - Method in class UnaryMorphologicalFunctionOp  
 Morphology 연산자를 설정한다.

setMethod(String) - Method in class UnaryMathFunctionOp  
 수학 연산자를 설정한다.

setMethod(String) - Method in class UnaryLogicalFunctionOp  
 논리 비교 연산자를 설정한다.

setMethod(String) - Method in class UnaryBitwiseFunctionOp  
 연산에 대한 방법을 설정한다.

setMethod(String) - Method in class UnaryTextureFunctionOp  
 Texture 추출 요소를 설정한다.

setMethod(String) - Method in class TM6ToTemperatureOp  
 온도 추정 방법을 설정한다.

setMethod(String) - Method in class TasseledCapTransformOp  
 변환 계수 설정을 위한 방법을 설정한다.

setMethod(String) - Method in class TopographicDeEffectOp  
 반사면 모델을 설정한다.

setMethod(String) - Method in class BinaryArithmeticFunctionOp  
 연산에 대한 방법을 설정한다.

setMethod(String) - Method in class BinaryLogicalFunctionOp  
 논리 비교 연산자를 설정한다.

setMethod(String) - Method in class BinaryPixelValueReplaceOp  
 논리 비교 연산자를 설정한다.

setMethod(String) - Method in class CubeGeometryFlipOp  
 처리 방법을 설정한다.

setMethod(String) - Method in class CubeGeometryResizeOp  
 재배열 방법을 설정한다.

setMethod(String) - Method in class CubeGeometryRotateOp  
 재배열 방법을 설정한다.

setMethod(String) - Method in class CubeGeometryTranslateOp  
 재배열 방법을 설정한다.

setMethod(String) - Method in class FFTOp  
 Forward/Backward 변환을 지정한다.

setMethod(String) - Method in class HistogramModifyEnhanceOp  
 히스토그램 변형법을 설정한다.

setMethod(String) - Method in class MinMaxEnhanceOp  
 최소-최대값 결정 방법을 설정한다.

setMethod(String) - Method in class MinMaxLinearEnhanceOp  
 최소-최대값 결정 방법을 설정한다.

setMethod(String, String) - Method in class ColorConvertOp  
 색 좌표 변환을 위한 입출력 좌표계를 설정한다.

setMin(double, double) - Method in class Feature

setMin(double, double) - Method in class Feature.Point

setMin(int, int, double) - Method in class Statistics  
 클래스의 분광대에 대한 최소값을 설정한다.

setMinClassDistance(float) - Method in class ISODATAClassifyOp  
 군집간의 최소거리를 설정한다.

setMinClassPixelCount(int) - Method in class ISODATAClassifyOp  
 분류되는 군집의 최소 화소수를 설정한다.

setMinLimit(double, double) - Method in class GeoVector  
 최소 x, y 좌표를 설정한다.

setMinLimit(Feature.Point) - Method in class GeoVector

point feature의 최소 좌표를 설정한다.

setMinMax(double[], double[]) - Method in class MinMaxEnhanceOp  
각 분광대의 최소-최대값을 설정한다.

setMinMax(double[], double[]) - Method in class MinMaxLinearEnhanceOp  
각 분광대의 최소-최대값을 설정한다.

setMinMax(int, float, float) - Method in class CubeHeader  
대상 영상의 분광대의 최소값과 최대값을 정의한다

setNoOfPixels(int, int) - Method in class Statistics  
클래스의 화소수를 설정한다.

setOpacity(float) - Method in class MultImageComposeOp  
기존의 화소값에 투명도를 조절하기 위한 가중치를 설정한다.

setOptions(boolean[]) - Method in class TerrainPropertyOp  
처리 옵션을 설정한다.

setOrigin(String) - Method in class CubeGeometryRotateOp  
회전의 중심축을 설정한다

setOutputRange(int, int) - Method in class HistogramModifyEnhanceOp  
히스토그램 변경 결과에 대한 영역을 설정한다.

setOverlayFactor(float) - Method in class BinaryArithmeticFunctionOp  
중첩 연산자의 경우 중첩 비율을 설정한다.

setPalette(Palette) - Method in class VectorLayer  
색상표를 설정한다.

setPalette(Palette) - Method in class ImageMaker  
색상표를 설정한다.

setParameter(double[]) - Method in class TopographicDeEffectOp  
태양 입력 변수등을 설정한다.

setParameter(double[][]) - Method in class LevelSliceOp  
Level slicing 변환표를 설정한다.

setParameter(float[]) - Method in class VegetationIndexOp  
식생 지수 산출과 관련된 Soil Line 변수들을 설정한다.

setParameter(float[]) - Method in class CubeGeometryTranslateOp  
수평, 수직 이동량을 설정한다.

setParameter(float[]) - Method in class DNTToReflectanceOp  
태양 변수를 설정한다.

setParameter(float[]) - Method in class FFTOp  
필터 변수를 설정한다.

setParameter(float[]) - Method in class ISODATAClassifyOp  
분류에 필요한 각 변수값들을 설정한다.

setParameter(float[]) - Method in class KMeansClassifyOp  
분류에 필요한 각 변수값들을 설정한다.

setParameter(float[]) - Method in class KMeansUnSupervisedClassifyOp  
분류에 필요한 각 변수값들을 설정한다.

setParameter(float, float) - Method in class SceneRegressNormalizeOp  
직선식을 설정한다.

setParameter(int[]) - Method in class SpatialProfileViewOp  
선을 정의 하기 위한 두 점을 설정한다.

setParameter(int, int) - Method in class SpectrumProfileViewOp  
화소의 위치를 설정한다.

setParameter(int, int) - Method in class PixelInteractionViewOp  
화소값을 가져올 영상에서의 위치를 설정한다.

setParameter(int, int) - Method in class PixelSpectrumViewOp  
화소값을 가져올 영상에서의 위치를 설정한다.

setParameter(int, int, int) - Method in class PixelInteractionViewOp  
화소값을 가져올 영상에서의 위치를 설정한다.

setParent(String) - Method in class MenuToken  
Parser 가 file 로 부터 얻어들인 Parent 값을 MenuToken 의 parent 로 Setting



한다.

setParentFrame(Frame) - Method in class Supervisor  
parent frame for dilalogs

setParm(int) - Method in class ParmHighPassKernel  
가중치 parameter를 설정한다.

setParm(int) - Method in class ParmLowPassKernel  
가중치 parameter를 설정한다.

setPixelType(int) - Method in class Cube  
화소형을 지정한다.

setPixelType(int) - Method in class PixelTypeConvertOp  
출력 화소형을 설정한다.

setPixelType(String) - Method in class BipToBilConvertOp  
입력 화소형을 설정한다.

setPixelType(String) - Method in class BsqToBilConvertOp  
입력 화소형을 설정한다.

setPixelType(String) - Method in class Cube  
화소형을 지정한다.

setPixelType(String) - Method in class NullImageMakeOp  
영상의 화소형을 설정한다.

setPixelType(String) - Method in class PixelTypeConvertOp  
출력 화소형을 설정한다.

setPixelUnit(float) - Method in class TerrainPropertyOp  
격자의 지상에서의 단위길이를 설정한다.

setPixelUnit(float) - Method in class TopographicDeEffectOp  
격자의 지상에서의 단위길이를 설정한다.

setPoint(Point) - Method in class SunParameterCanvas

setPolygonFill(boolean) - Method in class VectorLayer  
폴리곤 채움 모드를 설정한다.

setPolyOrder(int) - Method in class GCPCorrectOp  
변환식의 차수를 설정한다.

setPosition(int, int, int) - Method in class MultImageComposeOp  
영상을 삽입하기 위한 시작 위치를 설정한다.

setProgressStream(ProgressStream) - Method in class VegetationIndexOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class UnaryArithmeticFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class UnaryMorphologicalFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class UnaryMathFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class UnaryLogicalFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class UnaryBitwiseFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class UnaryTextureFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

TM6ToTemperatureOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class ThinningOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class TerrainPropertyOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class TasseledCapTransformOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class TopographicDeEffectOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class StatisticCalculateOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class SPOTHRVImportOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class SpectrumProfileViewOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class SpatialProfileViewOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class SpatialKernelFilterOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class ScattergramOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class SceneRegressNormalizeOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BilToBipConvertOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BilToBsqConvertOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BinaryArithmeticFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BinaryLogicalFunctionOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BinaryPixelValueReplaceOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BipToBilConvertOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class BsqToBilConvertOp  
프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

ChromaticityAnalysisOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

ClassifyErrorAnalysisOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class ColorConvertOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

CubeFileToMemoryOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

CubeGeometryFlipOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

CubeGeometryResizeOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

CubeGeometryRotateOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

CubeGeometryTranslateOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

CubeMemoryToFileOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class CubeSubsetOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class DNToRadianceOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

DNToReflectanceOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class DXFtoVectorOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

DXFVertexExtractOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class FFTOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

FFTPowerSpectrumOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class GCPCorrectOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

HistogramMatchOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

HistogramModifyEnhanceOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class HistogramOp

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class ImageProcessor

프로그램 진행 상태바를 설정한다.

setProgressStream(ProgressStream) - Method in class

InteactiveEnhanceOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

ISODATAClassifyOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

KMeansClassifyOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

KMeansUnSupervisedClassifyOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

LANDSATTMImportOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

LevelSliceOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MinDistanceClassifyOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MinMaxEnhanceOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MinMaxLinearEnhanceOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MinMaxOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MLClassifyOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MultiCubeMergeOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

MultImageComposeOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

NullImageMakeOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in interface

Operator  
setProgressStream(ProgressStream) - Method in class

PCATransformOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

PiecewiseEnhanceOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

PixelInteractionViewOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

PixelSpectrumViewOp  
프로그램 진행 상태바를 설정한다.  
setProgressStream(ProgressStream) - Method in class

PixelTypeConvertOp  
프로그램 진행 상태바를 설정한다.  
setReferenceHistogram(int[]) - Method in class

HistogramMatchOp  
참조 히스토그램을 설정한다.

setRegion(float[], float[]) - Method in class ScattergramOp  
스캐터그램을 작성을 화소값의 영역을 설정한다.

setResampling(int) - Method in class GCPCorrectOp  
화소 재배열 방법을 설정한다.

setRotate(float) - Method in class MultiImageComposeOp  
영상을 임의의 각도로 회전시켜 삽입하기 위한 회전각을 설정한다.

setSatellite(String) - Method in class TM6ToTemperatureOp  
위성의 종류를 설정한다.

setSatellite(String) - Method in class DNTToRadianceOp  
처리 대상영상을 획득한 위성의 종류를 설정한다.

setSatellite(String) - Method in class DNTToReflectanceOp  
처리 대상영상을 획득한 위성의 종류를 설정한다.

setScale(float) - Method in class CubeGeometryResizeOp  
크기 변환을 위한 스케일을 설정한다.

setScale(float) - Method in class GeoResampler  
축척을 설정한다.

setScattergramFileName(String) - Method in class ScattergramOp  
스캐터그램을 저장할 파일을 설정한다.

setSelectedFeatureType(int) - Method in class DXFtoVectorOp  
Dialog에서 선택된 Feature들의 type을 세팅한다.

setSelectedLayer(String[]) - Method in class DXFtoVectorOp  
Dialog에서 선택한 Layer를 설정한다.

setSelectedLayer(String[]) - Method in class DXFVertexExtractOp  
Dialog에서 선택한 Layer를 설정한다.

setShift(boolean) - Method in class MenuToken  
Parser 가 file 로 부터 얻어들이는 boolean shift 값을 MenuToken 의 shift 값으로 Setting 한다.

setSkip(int) - Method in class ImageLoadOp  
축척에 따른 Skip수를 설정한다.

setSolarZenith(float) - Method in class DNTToReflectanceOp  
태양 천정각을 설정한다.

setStatistic(float, float) - Method in class HistogramModifyEnhanceOp  
히스토그램 변경에 필요한 통계를 설정한다.

setStatisticFile(String) - Method in class KMeansClassifyOp  
표본 통계 파일을 지정한다.

setStatisticsFile(File) - Method in class StatisticCalculateOp  
통계 결과를 저장할 파일을 설정한다.

setStatisticsFile(File) - Method in class MinDistanceClassifyOp  
통계 정보를 갖고 있는 파일을 설정한다.

setStatisticsFile(File) - Method in class MLClassifyOp  
통계 정보를 갖고 있는 파일을 설정한다.

setStatisticsFile(String) - Method in class StatisticCalculateOp  
통계 결과를 저장할 파일을 설정한다.

setSumX(int, int, double) - Method in class Statistics  
클래스와 분광대에 대한 Sum을 설정한다.

setSumXY(int, int, int, double) - Method in class Statistics  
클래스의 분광대간의 Sum을 설정한다.

setSunPositions(double[]) - Method in class TerrainPropertyOp  
태양의 위치 변수를 설정한다.

setSunPositions(double[]) - Method in class TopographicDeEffectOp  
태양의 위치 변수를 구하기 위한 입력값을 설정한다.

setTarget(File, Target.File) - Method in class SPOTHRVIImportOp  
대상 영상을 설정한다.

setTarget(File, Target.File) - Method in class LANDSATTMImportOp  
대상 영상을 설정한다.

setTarget(String[], Target) – Method in class BsqToBilConvertOp  
입출력 영상을 설정한다.

setTarget(String, Target) – Method in class BipToBilConvertOp  
입출력 영상을 설정한다.

setTarget(Target.File, Target.Memory) – Method in class CubeFileToMemoryOp  
입출력 영상을 설정한다.

setTarget(Target.Memory, Target.File) – Method in class CubeMemoryToFileOp  
입출력 영상을 설정한다.

setTarget(Target) – Method in class StatisticCalculateOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class SpectrumProfileViewOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class SpatialProfileViewOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class ClassifyErrorAnalysisOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class HistogramOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class ImageProcessor  
처리할 영상을 설정한다.

setTarget(Target) – Method in class MinMaxOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class NullImageMakeOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class PixelInteractionViewOp  
대상 영상을 설정한다.

setTarget(Target) – Method in class PixelSpectrumViewOp  
대상 영상을 설정한다.

setTarget(Target[], Target) – Method in class MultiCubeMergeOp  
대상 영상을 설정한다.

setTarget(Target, File) – Method in class BilToBipConvertOp  
대상 영상을 설정한다.

setTarget(Target, String) – Method in class BilToBsqConvertOp  
대상 영상을 설정한다.

setTarget(Target, Target) – Method in class VegetationIndexOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class UnaryArithmeticFunctionOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class UnaryMorphologicalFunctionOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class UnaryMathFunctionOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class UnaryLogicalFunctionOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class UnaryBitwiseFunctionOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class UnaryTextureFunctionOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class TM6ToTemperatureOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class ThinningOp  
입출력 영상을 설정한다.

setTarget(Target, Target) – Method in class TerrainPropertyOp

입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class TasseledCapTransformOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class SpatialKernelFilterOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class ScattergramOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class SceneRegressNormalizeOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class ChromaticityAnalysisOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class ColorConvertOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class CubeGeometryFlipOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class CubeGeometryResizeOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class CubeGeometryRotateOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class CubeGeometryTranslateOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class CubeSubsetOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class DNTToRadianceOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class DNTToReflectanceOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target) - Method in class FFTOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class FFTPowerSpectrumOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class GCPCorrectOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class HistogramMatchOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class HistogramModifyEnhanceOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class InteactiveEnhanceOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class ISODATAClassifyOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class KMeansClassifyOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class KMeansUnSupervisedClassifyOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class LevelSliceOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class MinDistanceClassifyOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class MinMaxEnhanceOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class MinMaxLinearEnhanceOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class MLClassifyOp

대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class MultiImageComposeOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class PCATransformOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class PiecewiseEnhanceOp  
 대상 영상을 설정한다.  
setTarget(Target, Target) - Method in class PixelTypeConvertOp  
 대상 영상을 설정한다.  
setTarget(Target, Target, Target) - Method in class TopographicDeEffectOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target, Target) - Method in class BinaryArithmeticFunctionOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target, Target) - Method in class BinaryLogicalFunctionOp  
 입출력 영상을 설정한다.  
setTarget(Target, Target, Target) - Method in class BinaryPixelValueReplaceOp  
 입출력 영상을 설정한다.  
setTargetHistogram(int[]) - Method in class HistogramMatchOp  
 목표 히스토그램을 설정한다.  
setThreshold(int) - Method in class ThinningOp  
 입력 영상에서 임계값을 설정한다.  
setTitle(String) - Method in class Statistics  
 타이틀을 설정한다.  
setTitle(String) - Method in class AOI  
  
setTitle(String) - Method in class CubeHeader  
 헤더 정보의 Title을 설정한다.  
setTitle(String) - Method in class DXFtoVectorOp  
 Title을 설정한다.  
setTitle(String) - Method in class GeoVector  
 타이틀을 설정한다.  
setTotalPixel(int) - Method in class HistogramMatchOp  
 총 화소수를 설정한다.  
setUnClassifyClass(int) - Method in class ISODATAClassifyOp  
 미분류 화소에 대한 균집값을 설정한다.  
setUnClassifyClass(int) - Method in class KMeansClassifyOp  
 미분류 화소에 대한 균집값을 설정한다.  
setUnClassifyClass(int) - Method in class MinDistanceClassifyOp  
 미분류 화소에 대한 균집값을 설정한다.  
setVar(double[[[[]], double[[[[]]]) - Method in class Regressor  
 대상 행렬을 설정한다.  
setVector(File) - Method in class DXFtoVectorOp  
 출력 Vector 파일을 설정한다.  
setVector(int, GeoVector) - Method in class VectorLayer  
 인덱스에 벡터를 설정한다.  
setVisible(int, boolean) - Method in class VectorLayer  
 인덱스의 레이어를 화면에 나타내는 모드를 설정한다.  
setVisible(int, boolean) - Method in class AOILayer  
  
setVolume(int, int, int) - Method in class Cube  
 영상의 크기를 정의한다.



setVolume(int, int, int) - Method in class CubeHeader  
 입력된 Samples, Lines, Bands에 대하여 영상의 메모리 영역을 확보한다.

setVolume(Volume) - Method in class Cube  
 현재 정의된 영상과 같은 크기를 갖는 영상을 생성한다.

setWindowSize(int) - Method in class UnaryTextureFunctionOp  
 window의 크기를 설정한다.

setXYZFile(File) - Method in class DXFVertexExtractOp  
 출력 ASCII 파일을 설정한다.

SHORT - Static variable in class Cube  
 2바이트 정수형 (Signed)을 정의한다.

SKELETON - Static variable in class UnaryMorphologicalFunctionOp

SmpOffset - Variable in class Cube  
 영상의 Samples에 대한 오프셋을 정의한다.

smsaviValue(double[ ][ ], int, float[ ], int, int) - Method in class VegetationIndexOp

SobelKernel - class SobelKernel.  
 Sobel Kernel를 관리한다.

SobelKernel(String) - Constructor for class SobelKernel  
 Constructor

SpatialKernelFilterOp - class SpatialKernelFilterOp.  
 미리 정의된 커널을 이용한 필터링을 수행한다.

SpatialKernelFilterOp() - Constructor for class SpatialKernelFilterOp  
 Constructor

SpatialProfileViewOp - class SpatialProfileViewOp.  
 영상의 주어진 선을 따른 분광대 별 화소값의 변화를 출력한다.

SpatialProfileViewOp() - Constructor for class SpatialProfileViewOp  
 Constructor

SpectrumProfileViewOp - class SpectrumProfileViewOp.  
 영상의 주어진 선을 따른 분광대 별 화소값의 변화를 출력한다.

SpectrumProfileViewOp() - Constructor for class SpectrumProfileViewOp  
 Constructor

Spot - class Spot.

Spot() - Constructor for class Spot

Spot(int, int, int) - Constructor for class Spot

Spot(Spot) - Constructor for class Spot

SPOTHRVImportOp - class SPOTHRVImportOp.  
 SPOTHRVImportOp 영상을 읽어 내부포맷으로 변환한다.

SPOTHRVImportOp() - Constructor for class SPOTHRVImportOp  
 Constructor

Start - Variable in class HistogramModifyEnhanceOp

StartX - Variable in class ScattergramOp

StartY - Variable in class ScattergramOp

StatisticCalculateOp - class StatisticCalculateOp.  
 영상의 통계를 계산한다.

StatisticCalculateOp() - Constructor for class StatisticCalculateOp  
 Constructor

Statistics - class Statistics.  
통계를 관리하는 클래스이다.

Statistics() - Constructor for class Statistics  
Constructor

Statistics(int, int) - Constructor for class Statistics  
Constructor

Std - Variable in class ISODATAClassifyOp

Std - Variable in class KMeansClassifyOp

Std - Variable in class KMeansUnSupervisedClassifyOp

Std - Variable in class MinDistanceClassifyOp

stop() - Method in class VegetationIndexOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class UnaryArithmeticFunctionOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class UnaryMorphologicalFunctionOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class UnaryMathFunctionOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class UnaryLogicalFunctionOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class UnaryBitwiseFunctionOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class UnaryTextureFunctionOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class TM6ToTemperatureOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class ThinningOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class TerrainPropertyOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class TasseledCapTransformOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class TopographicDeEffectOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class StatisticCalculateOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class SPOTHRVImportOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class SpectrumProfileViewOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class SpatialProfileViewOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class SpatialKernelFilterOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class ScattergramOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class SceneRegressNormalizeOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class BilToBipConvertOp  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class BilToBsqConvertOp

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class BinaryArithmeticFunctionOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class BinaryLogicalFunctionOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class BinaryPixelValueReplaceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class BipToBilConvertOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class BsqToBilConvertOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class ChromaticityAnalysisOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class ClassifyErrorAnalysisOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class ColorConvertOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeFileToMemoryOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeGeometryFlipOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeGeometryResizeOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeGeometryRotateOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeGeometryTranslateOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeMemoryToFileOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class CubeSubsetOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class DNToRadianceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class DNToReflectanceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class DXFtoVectorOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class DXFVertexExtractOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class FFTOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class FFTPowerSpectrumOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class GCPCorrectOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class HistogramMatchOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class HistogramModifyEnhanceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class HistogramOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class InteactiveEnhanceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.  
stop() - Method in class ISODATAClassifyOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class KMeansClassifyOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class KMeansUnSupervisedClassifyOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class LANDSATTMImportOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class LevelSliceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MinDistanceClassifyOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MinMaxEnhanceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MinMaxLinearEnhanceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MinMaxOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MLClassifyOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MultiCubeMergeOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class MultiImageComposeOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class NullImageMakeOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in interface Operator

stop() - Method in class PCATransformOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class PiecewiseEnhanceOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class PixelInteractionViewOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class PixelSpectrumViewOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

stop() - Method in class PixelTypeConvertOp  
 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

SUBSTRACT - Static variable in class UnaryArithmeticFunctionOp

SUBTRACT - Static variable in class BinaryArithmeticFunctionOp  
 뺄셈 연산자

SunParameterCanvas - class SunParameterCanvas.

SunParameterCanvas(SunParameterPanel) - Constructor for class SunParameterCanvas

SunParameterPanel - class SunParameterPanel.

SunParameterPanel() - Constructor for class SunParameterPanel

Supervisor - class Supervisor.

Parent class for each command All of command class should inherit this class and implement execute() method.

Supervisor() - Constructor for class Supervisor  
 constructor for class loading

synopsis(Window, Target, int, Rectangle) - Static method in class

GeoEnviron

사용자 입력창에서 Screen Synopsis 기능을 지원한다.

---

## T

Target -class Target.

Target.File -class Target.File.

파일 타입의 Target을 관리한다.

Target.File() -Constructor for class Target.File

Constructor

Target.File(File) -Constructor for class Target.File

Constructor

Target.File(String) -Constructor for class Target.File

Constructor

Target.File(String, String) -Constructor for class Target.File

Constructor

Target.File(Target.File) -Constructor for class Target.File

Constructor

Target.Memory -class Target.Memory.

메모리 타입의 Target을 관리한다.

Target.Memory() -Constructor for class Target.Memory

Constructor

Target.Memory(String) -Constructor for class Target.Memory

Constructor

Target() -Constructor for class Target

TargetLister -class TargetLister.

TargetLister() -Constructor for class TargetLister

TasseledCapTransformOp -class TasseledCapTransformOp.

Tasseled Cap 변환을 수행한다.

TasseledCapTransformOp() -Constructor for class

TasseledCapTransformOp

Constructor

TerrainPropertyOp -class TerrainPropertyOp.

DEM으로 부터 지형 요소를 계산한다.

TerrainPropertyOp() -Constructor for class TerrainPropertyOp

Constructor

TEXT -Static variable in interface Drawable

THICK -Static variable in class UnaryMorphologicalFunctionOp

THIN -Static variable in class UnaryMorphologicalFunctionOp

ThinningOp -class ThinningOp.

이진 영상의 세선화를 실행한다.

ThinningOp() -Constructor for class ThinningOp

Constructor

title -Variable in class ScattergramOp

Title - Variable in class AOI

Title - Variable in class CubeHeader  
영상의 Title

Title - Variable in class DXFtoVectorOp.VHeader

TM6ToTemperatureOp - class TM6ToTemperatureOp.  
LANDSAT TM band 6 영상으로 부터 지표면 온도를 추출한다.

TM6ToTemperatureOp() - Constructor for class TM6ToTemperatureOp  
Constructor

tmpTitle - Variable in class ScattergramOp

TopographicDeEffectOp - class TopographicDeEffectOp.  
지형에 의한 영향을 제거한다.

TopographicDeEffectOp() - Constructor for class TopographicDeEffectOp  
Constructor

toString() - Method in class Target  
Target의 이름을 되돌린다.

toString() - Method in class CubeHeader  
영상의 헤더 정보를 문자열로 출력한다.

toString() - Method in class Feature.Point

toString() - Method in class Feature.Line

toString() - Method in class Feature.Polygon

toString() - Method in class GeoLookupTable

toString() - Method in class GeoVector

toString() - Method in class HistogramOp

totalFreq - Variable in class HistogramModifyEnhanceOp

translate(int, int, int) - Method in class Spot

translate(int, int, int) - Method in class Cube  
영상에서의 물리적 위치를 지정한다.

translate(Spot) - Method in class Spot

Transpose(byte[], byte[], int, int, int) - Method in class CubeGeometryFlipOp  
바이트형 배열의 화소를 전이배열 한다.

Transpose(byte[], int, byte[][][], int, int, int) - Method in class CubeGeometryFlipOp  
바이트형 배열의 화소를 전이배열 한다.

Transpose(byte[], int, byte[], int, int) - Method in class CubeGeometryFlipOp  
바이트형 배열의 화소를 전이배열 한다.

Transpose(float[], float[], int, int, int) - Method in class CubeGeometryFlipOp  
4바이트형 실수형 배열의 화소를 전이배열 한다.

Transpose(float[], int, float[][][], int, int, int) - Method in class CubeGeometryFlipOp  
4바이트 실수형 배열의 화소를 전이배열 한다.

Transpose(float[], int, float[][][], int, int, int) - Method in class CubeGeometryFlipOp  
4바이트 실수형 배열의 화소를 전이배열 한다.

Transpose(float[], int, float[][][], int, int, int) - Method in class CubeGeometryFlipOp  
4바이트 실수형 배열의 화소를 전이배열 한다.

Transpose(float[], int, float[], int, int) - Method in class CubeGeometryFlipOp  
4바이트 정수형 배열의 화소를 전이배열 한다.

Transpose(int[], int[], int, int, int) - Method in class CubeGeometryFlipOp  
4바이트형 정수형 배열의 화소를 전이배열 한다.

Transpose(int[], int, int[][], int, int, int) - Method in class CubeGeometryFlipOp  
4바이트 정수형 배열의 화소를 전이배열 한다.

Transpose(int[], int, int[], int, int) - Method in class CubeGeometryFlipOp  
4바이트 정수형 배열의 화소를 전이배열 한다.

Transpose(short[], int, short[][], int, int, int) - Method in class CubeGeometryFlipOp  
2바이트 정수형 배열의 화소를 전이배열 한다.

Transpose(short[], int, short[], int, int) - Method in class CubeGeometryFlipOp  
2바이트 정수형 배열의 화소를 전이배열 한다.

Transpose(short[], short[], int, int, int) - Method in class CubeGeometryFlipOp  
2바이트형 정수형 배열의 화소를 전이배열 한다.

trimStringForNumeric(String) - Static method in class GeoUtility  
문자열을 유효 숫자로 변환한다.

TSAVI - Static variable in class VegetationIndexOp

tsaviValue(double[][][], int, float[], int, int) - Method in class VegetationIndexOp

TVI - Static variable in class VegetationIndexOp

tviValue(double[][][], int, float[], int, int) - Method in class VegetationIndexOp

Type - Variable in class AOI

Type - Variable in class DXFtoVectorOp.FeatureProtocol

---

## U

UnaryArithmeticFunctionOp - class UnaryArithmeticFunctionOp.  
영상의 단일 분광대에 대하여 산술 연산을 한다.

UnaryArithmeticFunctionOp() - Constructor for class UnaryArithmeticFunctionOp  
Constructor

UnaryBitwiseFunctionOp - class UnaryBitwiseFunctionOp.  
영상의 단일 분광대에 대하여 비트이동 연산을 한다.

UnaryBitwiseFunctionOp() - Constructor for class UnaryBitwiseFunctionOp  
Constructor

UnaryLogicalFunctionOp - class UnaryLogicalFunctionOp.  
영상의 단일 분광대에 대하여 논리 비교 연산을 한다.

UnaryLogicalFunctionOp() - Constructor for class UnaryLogicalFunctionOp

Constructor  
UnaryMathFunctionOp -class UnaryMathFunctionOp.  
 분광대에 대하여 수학적 연산을 한다.  
UnaryMathFunctionOp() -Constructor for class UnaryMathFunctionOp  
 Constructor  
UnaryMorphologicalFunctionOp -class UnaryMorphologicalFunctionOp.  
 단일 분광대 영상(2진 영상)에 대한 Morphology연산을 실행한다.  
UnaryMorphologicalFunctionOp() -Constructor for class  
UnaryMorphologicalFunctionOp  
 Constructor  
UnaryTextureFunctionOp -class UnaryTextureFunctionOp.  
 단일 분광대에 대한 Texture특성을 분석한다.  
UnaryTextureFunctionOp() -Constructor for class  
UnaryTextureFunctionOp  
 Constructor  
UNDEFINED - Static variable in class Cube  
  
update(Graphics) -Method in class SunParameterCanvas  
  
USER\_DEFINED - Static variable in class DXFtoVectorOp  
  
USER\_DEFINED - Static variable in class DXFVertexExtractOp  
  
UserMenuPosition - Static variable in class GeoEnviron  
 사용자 메뉴 목록에서의 초기 위치  
USHORT - Static variable in class Cube  
 2바이트 정수형(Untsigned)을 정의한다.

---

## V

validMinMax -Variable in class CubeHeader  
 분광대의 최소, 최대값 유효성  
VectorLayer -class VectorLayer.  
 벡터 layer를 관리한다.  
VectorLayer() -Constructor for class VectorLayer  
 Constructor  
VegetationIndexOp -class VegetationIndexOp.  
 식생지수를 계산한다.  
VegetationIndexOp() -Constructor for class VegetationIndexOp  
 Constructor  
VertexNo -Variable in class DXFtoVectorOp.FeatureProtocol  
  
VerticalFlip(byte[], int, byte[][][], int, int, int) -Method in class  
CubeGeometryFlipOp  
 바이트형 배열의 화소를 수직방향으로 재배열 한다.  
VerticalFlip(byte[], int, byte[], int) -Method in class  
CubeGeometryFlipOp  
 바이트형 배열의 화소를 수직방향으로 재배열 한다.  
VerticalFlip(float[], int, float[][][], int, int, int) -Method in class  
CubeGeometryFlipOp  
 4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.  
VerticalFlip(float[], int, float[], int) -Method in class



CubeGeometryFlipOp

4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.

VerticalFlip(int[], int, int[][][], int, int, int) - Method in class CubeGeometryFlipOp

4바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

VerticalFlip(int[], int, int[], int) - Method in class CubeGeometryFlipOp

4바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

VerticalFlip(short[], int, short[][][], int, int, int) - Method in class CubeGeometryFlipOp

2바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

VerticalFlip(short[], int, short[], int) - Method in class CubeGeometryFlipOp

2바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

ViewCalc - class ViewCalc.

ViewCalc() - Constructor for class ViewCalc

Volume - class Volume.

Volume() - Constructor for class Volume

Volume(int, int, int) - Constructor for class Volume

Volume(Volume) - Constructor for class Volume

---

## W

WDVI - Static variable in class VegetationIndexOp

wdviValue(double[][][], int, float[], int, int) - Method in class VegetationIndexOp

width - Variable in class Volume

Width - Variable in class HistogramModifyEnhanceOp

WidthX - Variable in class ScattergramOp

WidthY - Variable in class ScattergramOp

windowClosing(WindowEvent) - Method in class geopixel

write(File) - Method in class Statistics

통계를 파일로 출력한다.

write(File) - Method in class AOI

write(File) - Method in class AOI.Bit

write(File) - Method in class AOI.Graph

write(File) - Method in class CubeHeader

영상의 헤더 정보를 파일로 저장한다.  
write(File) - Method in class GeoLookupTable  
 변환참조표를 파일로 출력한다.  
write(File) - Method in class GeoMenuWriter

write(File) - Method in class GeoVector  
 Feature를 파일로 출력한다.  
write(float[]) - Method in class GeoDataOutputStream  
 4바이트 실수형의 자료를 출력한다.  
write(float[], int, int) - Method in class GeoDataOutputStream  
 4바이트 실수형의 자료를 출력한다.  
write(GeoDataOutputStream) - Method in class DXFtoVectorOp.VHeader  
 헤더 정보중(Title, Xmin, Xmax, Ymin, Ymax, FeatureCount)를 파일로  
 출력한다.  
write(GeoDataOutputStream) - Method in class  
DXFtoVectorOp.FeatureProtocol  
 벡터 정보(Vertex)를 파일로 출력한다.  
write(int[]) - Method in class GeoDataOutputStream  
 4바이트 정수형의 자료를 출력한다.  
write(int[], int, int) - Method in class GeoDataOutputStream  
 4바이트 정수형의 자료를 출력한다.  
write(RandomAccessFile) - Method in class DXFtoVectorOp.VHeader  
 헤더 정보중(Title, Xmin, Xmax, Ymin, Ymax, FeatureCount)를 파일로  
 출력한다.  
write(short[]) - Method in class GeoDataOutputStream  
 2바이트 정수형의 자료를 출력한다.  
write(short[], int, int) - Method in class GeoDataOutputStream  
 2바이트 정수형의 자료를 출력한다.  
write(String) - Method in class Statistics  
 통계를 파일로 출력한다.  
write(String) - Method in class ScattergramOp  
 스캐터그램을 파일로 출력한다.  
write(String) - Method in class AOI

write(String) - Method in class AOI.Bit

write(String) - Method in class AOI.Graph

write(String) - Method in class CubeHeader  
 영상의 헤더 정보를 파일로 저장한다.  
write(String) - Method in class GeoLookupTable  
 변환참조표를 파일로 출력한다.  
write(String) - Method in class GeoMenuWriter

write(String) - Method in class GeoVector  
 Feature를 파일로 출력한다.  
write(String) - Method in class HistogramOp  
 농도 분포를 파일로 출력한다.  
writeCommandObject(GeoCommandTable) - Static method in class  
GeoEnviron  
 명령어 객체를 테이블에 출력한다.  
writeFixedString(String, int) - Method in class GeoDataOutputStream  
 고정된 길이의 문자열을 출력한다.  
writeMenuObject(MenuBar) - Static method in class GeoMenuWriter

---

## X

x - Variable in class Spot

x - Variable in class Feature.Point

X - Variable in class DXFtoVectorOp.FeatureProtocol

x86writeInt(int) - Method in class GeoDataOutputStream

32비트인 Int Integer를 OutputStream에 쓰기 위한 메서드

x86writeShort(int) - Method in class GeoDataOutputStream

16비트인 Short Integer를 OutputStream에 쓰기 위한 메서드

xmax - Variable in class GeoRectangle

Xmax - Variable in class CubeHeader

x 방향의 최소값

Xmax - Variable in class DXFtoVectorOp.VHeader

xmin - Variable in class GeoRectangle

Xmin - Variable in class CubeHeader

x 방향의 최대값

Xmin - Variable in class DXFtoVectorOp.VHeader

---

## Y

y - Variable in class Spot

y - Variable in class Feature.Point

Y - Variable in class DXFtoVectorOp.FeatureProtocol

ymax - Variable in class GeoRectangle

Ymax - Variable in class CubeHeader

y 방향의 최대값

Ymax - Variable in class DXFtoVectorOp.VHeader

ymin - Variable in class GeoRectangle

Ymin - Variable in class CubeHeader

y 방향의 최소값

Ymin - Variable in class DXFtoVectorOp.VHeader

---

## Z

z - Variable in class Spot

Zmax - Variable in class CubeHeader  
각 분광대의 최대값

Zmin - Variable in class CubeHeader  
각 분광대의 최소값

ZoomViewScale - Static variable in class GeoEnviron  
확대 비율 초기값

ZoomViewSize - Static variable in class GeoEnviron  
확대 윈도우의 초기 크기값

ZVALUE - Static variable in class DXFtoVectorOp

ZVALUE - Static variable in class DXFVertexExtractOp

---

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [Y](#) [Z](#)

[Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV](#) [NEXT](#)

[FRAMES](#) [NO FRAMES](#)

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class AOI.Bit

```
java.lang.Object
|
+--AOI
    |
    +--AOI.Bit
```

---

```
public static class AOI.Bit
extends AOI
```

---

### Inner classes inherited from class [AOI](#)

[AOI.Bit](#), [AOI.Graph](#)

### Field Summary

	int	<a href="#">Lines</a>
	int	<a href="#">Samples</a>

### Fields inherited from class [AOI](#)

[BIT](#), [GRAPH](#), [Title](#), [Type](#)

### Constructor Summary

[AOI.Bit\(\)](#)

[AOI.Bit\(int smps\)](#)

[AOI.Bit\(int smps, int lns\)](#)

## Method Summary

void	<u>addLine</u> (boolean[] b, int offset)
boolean[]	<u>getLine</u> (boolean[] buf, int ln, int width)
java.awt.Dimension	<u>getSize</u> ()
boolean	<u>on</u> (int x, int y)
boolean	<u>on</u> (java.awt.Point p)
void	<u>read</u> (java.io.File f)
void	<u>read</u> (java.lang.String path)
java.awt.Rectangle	<u>removePolygon</u> (int x, int y)
void	<u>replaceLine</u> (int ln, boolean[] b)
short[]	<u>runLengthCode</u> (boolean[] b, int offset, int width)
void	<u>write</u> (java.io.File f)
void	<u>write</u> (java.lang.String path)

### Methods inherited from class AOI

getTitle, getType, isBit, main, setTitle

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### Samples

```
public int Samples
```

---

### Lines

```
public int Lines
```

## Constructor Detail

### AOI.Bit

```
public AOI.Bit()
```

---

## AOI.Bit

```
public AOI.Bit(int smps)
```

---

## AOI.Bit

```
public AOI.Bit(int smps,  
              int lns)
```

## Method Detail

### getSize

```
public java.awt.Dimension getSize()
```

---

### removePolygon

```
public java.awt.Rectangle removePolygon(int x,  
                                       int y)
```

Overrides:

[removePolygon](#) in class [AOI](#)

---

### on

```
public boolean on(int x,  
                 int y)
```

Overrides:

[on](#) in class [AOI](#)

---

### on

```
public boolean on(java.awt.Point p)
```

Overrides:

[on](#) in class [AOI](#)

---

### getLine

```
public boolean[] getLine(boolean[] buf,  
                        int ln,  
                        int width)
```

Overrides:

[getLine](#) in class [AOI](#)

---

---

## replaceLine

```
public void replaceLine(int ln,  
                        boolean[] b)
```

---

## addLine

```
public void addLine(boolean[] b,  
                   int offset)
```

---

## runLengthCode

```
public short[] runLengthCode(boolean[] b,  
                             int offset,  
                             int width)
```

---

## read

```
public void read(java.lang.String path)  
               throws java.io.IOException
```

### Overrides:

read in class AOI

---

## read

```
public void read(java.io.File f)  
               throws java.io.IOException
```

### Overrides:

read in class AOI

---

## write

```
public void write(java.lang.String path)  
               throws java.io.IOException
```

### Overrides:

write in class AOI

---

## write

```
public void write(java.io.File f)  
               throws java.io.IOException
```



Overrides:  
write in class AOI

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class AOI.Graph

```
java.lang.Object
|
+--AOI
|
+--AOI.Graph
```

---

public static class AOI.Graph  
extends [AOI](#)

---

### Inner classes inherited from class [AOI](#)

[AOI.Bit](#), [AOI.Graph](#)

### Fields inherited from class [AOI](#)

[BIT](#), [GRAPH](#), [Title](#), [Type](#)

### Constructor Summary

[AOI.Graph](#)()

[AOI.Graph](#)(java.lang.String title)

## Method Summary

void	<u>addPolygon</u> (java.awt.Polygon p)
void	<u>clear</u> ()
int	<u>count</u> ()
void	<u>draw</u> (java.awt.Graphics g, float scale)
void	<u>draw</u> (java.awt.Graphics g, int xoffset, int yoffset)
java.util.Enumeration	<u>elements</u> ()
boolean[]	<u>getLine</u> (boolean[] buf, int ln, int width)
java.awt.Polygon[]	<u>getPolygon</u> ()
java.awt.Polygon	<u>getPolygon</u> (int index)
int	<u>indexOf</u> (java.awt.Polygon p)
void	<u>insertPolygon</u> (int index, java.awt.Polygon p)
boolean	<u>on</u> (int x, int y)
boolean	<u>on</u> (java.awt.Point p)
void	<u>read</u> (java.io.File f)
void	<u>read</u> (java.lang.String path)
java.awt.Polygon	<u>removePolygon</u> (int index)
java.awt.Rectangle	<u>removePolygon</u> (int x, int y)
int	<u>removePolygon</u> (java.awt.Polygon p)
void	<u>write</u> (java.io.File f)
void	<u>write</u> (java.lang.String path)

### Methods inherited from class AOI

getTitle, getType, isBit, main, setTitle

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### AOI.Graph

```
public AOI.Graph()
```

---

### AOI.Graph

```
public AOI.Graph(java.lang.String title)
```

## Method Detail

### addPolygon

```
public void addPolygon(java.awt.Polygon p)
```

---

### insertPolygon

```
public void insertPolygon(int index,  
                           java.awt.Polygon p)
```

---

### removePolygon

```
public java.awt.Polygon removePolygon(int index)
```

---

### removePolygon

```
public java.awt.Rectangle removePolygon(int x,  
                                         int y)
```

Overrides:

[removePolygon](#) in class [AOI](#)

---

### removePolygon

```
public int removePolygon(java.awt.Polygon p)
```

---

### elements

```
public java.util.Enumeration elements()
```

---

### getPolygon

```
public java.awt.Polygon[] getPolygon()
```

---

## getPolygon

```
public java.awt.Polygon getPolygon(int index)
```

---

## indexOf

```
public int indexOf(java.awt.Polygon p)
```

---

## count

```
public int count()
```

---

## clear

```
public void clear()
```

---

## on

```
public boolean on(int x,  
                 int y)
```

Overrides:  
    on in class AOI

---

## on

```
public boolean on(java.awt.Point p)
```

Overrides:  
    on in class AOI

---

## getLine

```
public boolean[] getLine(boolean[] buf,  
                        int ln,  
                        int width)
```

Overrides:  
    getLine in class AOI

---

## draw

```
public void draw(java.awt.Graphics g,  
                int xoffset,  
                int yoffset)
```

---

## draw

```
public void draw(java.awt.Graphics g,  
                float scale)
```

---

## read

```
public void read(java.lang.String path)  
    throws java.io.IOException
```

### Overrides:

[read](#) in class [AOI](#)

---

## read

```
public void read(java.io.File f)  
    throws java.io.IOException
```

### Overrides:

[read](#) in class [AOI](#)

---

## write

```
public void write(java.lang.String path)  
    throws java.io.IOException
```

### Overrides:

[write](#) in class [AOI](#)

---

## write

```
public void write(java.io.File f)  
    throws java.io.IOException
```

### Overrides:

[write](#) in class [AOI](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class Tree Deprecated Index Help

PREV CLASS [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class AOI

java.lang.Object

|  
+--AOI

Direct Known Subclasses:

[AOI.Bit](#), [AOI.Graph](#)

---

public abstract class AOI  
extends java.lang.Object

---

### Inner Class Summary

static class	<a href="#">AOI.Bit</a>
static class	<a href="#">AOI.Graph</a>

### Field Summary

static int	<a href="#">BIT</a>
static int	<a href="#">GRAPH</a>
protected java.lang.String	<a href="#">Title</a>
protected int	<a href="#">Type</a>

### Constructor Summary

<a href="#">AOI()</a>	
-----------------------	--

## Method Summary

abstract boolean[]	<u>getLine</u> (boolean[] buf, int ln, int width)
java.lang.String	<u>getTitle</u> ()
int	<u>getType</u> ()
static boolean	<u>isBit</u> (java.io.File f)
static void	<u>main</u> (java.lang.String[] args)
abstract boolean	<u>on</u> (int x, int y)
abstract boolean	<u>on</u> (java.awt.Point p)
abstract void	<u>read</u> (java.io.File f)
abstract void	<u>read</u> (java.lang.String path)
abstract java.awt.Rectangle	<u>removePolygon</u> (int x, int y)
void	<u>setTitle</u> (java.lang.String s)
abstract void	<u>write</u> (java.io.File f)
abstract void	<u>write</u> (java.lang.String path)

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### GRAPH

public static final int GRAPH

---

### BIT

public static final int BIT

---

### Type

protected int Type

---



## Title

```
protected java.lang.String Title
```

## Constructor Detail

### AOI

```
public AOI()
```

## Method Detail

### getType

```
public int getType()
```

---

### setTitle

```
public void setTitle(java.lang.String s)
```

---

### getTitle

```
public java.lang.String getTitle()
```

---

### on

```
public abstract boolean on(int x,  
                           int y)
```

---

### on

```
public abstract boolean on(java.awt.Point p)
```

---

### getLine

```
public abstract boolean[] getLine(boolean[] buf,  
                                  int ln,  
                                  int width)
```

---

### removePolygon

```
public abstract java.awt.Rectangle removePolygon(int x,  
                                                  int y)
```

---

## read

```
public abstract void read(java.lang.String path)
                        throws java.io.IOException
```

---

## read

```
public abstract void read(java.io.File f)
                        throws java.io.IOException
```

---

## write

```
public abstract void write(java.lang.String path)
                        throws java.io.IOException
```

---

## write

```
public abstract void write(java.io.File f)
                        throws java.io.IOException
```

---

## isBit

```
public static boolean isBit(java.io.File f)
                        throws java.io.IOException
```

---

## main

```
public static void main(java.lang.String[] args)
```

---

[Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

---

## Class AOILayer

java.lang.Object  
|  
+--AOILayer

---

```
public class AOILayer  
extends java.lang.Object
```

---

Field Summary	
static int	<u>AOI_ADD</u>
static int	<u>AOI_DELETE</u>
static int	<u>AOI_NONE</u>
int	<u>CurrentIndex</u>
static int	<u>MAX_SIZE</u>
java.awt.Polygon	<u>NeoPoly</u>

Constructor Summary	
<u>AOILayer()</u>	

Method Summary	
AOI	<code>getAOI(int index)</code>
int	<code>getCapacity()</code>
java.awt.Color	<code>getColor(int index)</code>
int	<code>getEditState()</code>
boolean	<code>isEmpty(int index)</code>
boolean	<code>isVisible(int index)</code>
void	<code>removeAOI(int index)</code>
void	<code>setAOI(int index, AOI aoi)</code>
void	<code>setColor(int index, java.awt.Color c)</code>
void	<code>setEditState(int edit)</code>
void	<code>setVisible(int index, boolean b)</code>

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### MAX\_SIZE

```
public static final int MAX_SIZE
```

---

### AOI\_NONE

```
public static final int AOI_NONE
```

---

### AOI\_ADD

```
public static final int AOI_ADD
```

---

### AOI\_DELETE

```
public static final int AOI_DELETE
```

---

## NeoPoly

```
public java.awt.Polygon NeoPoly
```

---

## CurrentIndex

```
public int CurrentIndex
```

<b>Constructor Detail</b>
---------------------------

### AOILayer

```
public AOILayer()
```

<b>Method Detail</b>
----------------------

### setEditState

```
public void setEditState(int edit)
```

---

### getEditState

```
public int getEditState()
```

---

### setColor

```
public void setColor(int index,  
                    java.awt.Color c)
```

---

### getColor

```
public java.awt.Color getColor(int index)
```

---

### setVisible

```
public void setVisible(int index,  
                      boolean b)
```

---

### isVisible

```
public boolean isVisible(int index)
```

---

## isEmpty

```
public boolean isEmpty(int index)
```

---

## setAOI

```
public void setAOI(int index,  
                  AOI aoi)
```

---

## removeAOI

```
public void removeAOI(int index)
```

---

## getAOI

```
public AOI getAOI(int index)
```

---

## getCapacity

```
public int getCapacity()
```

---

### [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class BilToBipConvertOp

java.lang.Object  
|  
+--BilToBipConvertOp

```
public class BilToBipConvertOp
extends java.lang.Object
implements Operator
```

BIL 포맷의 영상을 BIP포맷의 파일로 변환하여 저장한다.

### Constructor Summary

<code>BilToBipConvertOp()</code> Constructor
---

### Method Summary

void	<code>operate()</code> 포맷 변환을 실행한다.
void	<code>setBands(int[] bands)</code> 처리 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target srcTgt, java.io.File destFile)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Constructor Detail

#### BilToBipConvertOp

```
public BilToBipConvertOp()
```

Constructor

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

처리 대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target srcTgt,  
                     java.io.File destFile)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

포맷 변환을 실행한다.

Throws:

java.io.IOException - IOException

java.lang.InterruptedException - InterruptedException

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---



## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class `BilToBsqConvertOp`

`java.lang.Object`  
|  
+--`BilToBsqConvertOp`

---

```
public class BilToBsqConvertOp
extends java.lang.Object
implements Operator
```

BIL 포맷의 영상을 BSQ 포맷으로 변환하여 저장한다.

---

## Constructor Summary

<code>BilToBsqConvertOp()</code>
----------------------------------

## Method Summary

void	<code>operate()</code> 포맷 변환을 실행한다.
void	<code>setBands(int[] bands)</code> 처리 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, java.lang.String f)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `BilToBsqConvertOp`

```
public BilToBsqConvertOp()
```

## Method Detail

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## setBands

```
public void setBands(int[] bands)
```

처리 대상 분광대를 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target stgt,  
                    java.lang.String f)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

포맷 변환을 실행한다.

Throws:

[java.io.IOException](#) – [IOException](#)

[java.lang.InterruptedException](#) – [InterruptedException](#)

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

[Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class BinaryArithmeticFunctionOp

java.lang.Object  
|  
+--BinaryArithmeticFunctionOp

---

```
public class BinaryArithmeticFunctionOp
extends java.lang.Object
implements Operator
```

두개의 분광대에 대하여 산술 연산을 한다. 출력 화소값은 4바이트 실수형으로 정의된다.

---

### Field Summary

static int	<a href="#">ADD</a> 덧셈 연산자
static int	<a href="#">DIVIDE</a> 나눗셈 연산자
static int	<a href="#">MULTIPLY</a> 곱셈 연산자
static int	<a href="#">NONE</a>
static int	<a href="#">OVERLAY</a> 중첩 연산자
static int	<a href="#">SUBTRACT</a>

### Constructor Summary

<a href="#">BinaryArithmeticFunctionOp()</a> Constructor
---

## Method Summary

protected void	<code>changePixelValue(double[] a, double[] b, int aOffset, float[] c, int bOffset, int length)</code> 산술 연산을 수행한다.
void	<code>operate()</code> 입력영상의 두 분광대에 대한 산술연산을 실행한다.
void	<code>setBands(int b1, int b2)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 연산에 대한 방법을 설정한다.
void	<code>setOverlayFactor(float c)</code> 중첩 연산자의 경우 중첩 비율을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target s1, Target s2, Target tgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

public static final int NONE

---

ADD

public static final int ADD

덧셈 연산자

---

SUBTRACT

public static final int SUBTRACT

뺄셈 연산자

---

MULTIPLY

public static final int MULTIPLY

곱셈 연산자

---

## DIVIDE

```
public static final int DIVIDE
```

나눗셈 연산자

---

## OVERLAY

```
public static final int OVERLAY
```

중첩 연산자

## Constructor Detail

### BinaryArithmeticFunctionOp

```
public BinaryArithmeticFunctionOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

연산에 대한 방법을 설정한다.

---

### setOverlayFactor

```
public void setOverlayFactor(float c)
```

중첩 연산자의 경우 중첩 비율을 설정한다. 중첩 비율은 두 분광대에 대하여 다음과 같이 적용된다. 분광대 1 \* 중첩 비율 + 분광대 2 \* (1 - 중첩비율)

---

### setBands

```
public void setBands(int b1,  
                    int b2)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target s1,  
                    Target s2,  
                    Target tgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 두 분광대에 대한 산술연산을 실행한다.

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                double[] b,  
                                int aOffset,  
                                float[] c,  
                                int bOffset,  
                                int length)
```

산술 연산을 수행한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class BinaryLogicalFunctionOp

java.lang.Object  
|  
+--BinaryLogicalFunctionOp

---

```
public class BinaryLogicalFunctionOp
extends java.lang.Object
implements Operator
```

두개의 분광대에 대하여 논리 비교 연산을 한다. 출력 화소값은 바이트 형으로 조건이 참일 경우에는 1, 거짓일 경우에는 0 값이 출력 화소값으로 정의 된다.

---

### Field Summary

static int	<u>EQUAL</u> = 연산자
static int	<u>GREATER</u> > 연산자
static int	<u>GREATER THAN</u> >= 연산자
static int	<u>LESS</u> < 연산자
static int	<u>LESS THAN</u> <= 연산자
static int	<u>NONE</u>
static int	<u>NOT EQUAL</u> !

### Constructor Summary

<u>BinaryLogicalFunctionOp()</u> Constructor
---



## Method Summary

protected void	<code>changePixelValue(double[] a, double[] b, int aOffset, byte[] c, int bOffset, int length)</code> 비교 연산을 수행한다.
void	<code>operate()</code> 입력영상의 두 분광대에 대한 논리 비교 연산을 실행한다.
void	<code>setBands(int b1, int b2)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 논리 비교 연산자를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target s1, Target s2, Target tgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

public static final int NONE

---

EQUAL

public static final int EQUAL

= 연산자

---

NOT\_EQUAL

public static final int NOT\_EQUAL

!= 연산자

---

LESS

public static final int LESS

< 연산자

---

## LESS\_THAN

```
public static final int LESS_THAN
```

<= 연산자

---

## GREATER

```
public static final int GREATER
```

> 연산자

---

## GREATER\_THAN

```
public static final int GREATER_THAN
```

>= 연산자

## Constructor Detail

### BinaryLogicalFunctionOp

```
public BinaryLogicalFunctionOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

논리 비교 연산자를 설정한다.

---

### setBands

```
public void setBands(int b1,  
                    int b2)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target s1,  
                    Target s2,  
                    Target tgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 두 분광대에 대한 논리 비교 연산을 실행한다.

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                double[] b,  
                                int aOffset,  
                                byte[] c,  
                                int bOffset,  
                                int length)
```

비교 연산을 수행한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class BinaryPixelValueReplaceOp

```
java.lang.Object
|
+--BinaryPixelValueReplaceOp
```

```
public class BinaryPixelValueReplaceOp
extends java.lang.Object
implements Operator
```

두개의 분광대에 대하여 논리 비교 연산을 한다. 출력 화소값은 4바이트 실수형으로 조건이 참일 경우에는 새로운 값이, 거짓일 경우에는 입력 분광대 1의 값이 출력 화소값으로 정의 된다.

## Field Summary

static int	<a href="#">EQUAL</a> = 연산자
static int	<a href="#">GREATER</a> > 연산자
static int	<a href="#">GREATER THAN</a> >= 연산자
static int	<a href="#">LESS</a> < 연산자
static int	<a href="#">LESS THAN</a> <= 연산자
static int	<a href="#">NONE</a>
static int	<a href="#">NOT EQUAL</a> !

## Constructor Summary

```
BinaryPixelValueReplaceOp\(\)  
Constructor
```

## Method Summary

protected void	<code>changePixelValue(double[] a, double[] b, int aOffset, float[] c, int bOffset, int length)</code> 비교 연산을 수행한다.
void	<code>operate()</code> 논리 비교 연산을 실행한다.
void	<code>setBands(int b1, int b2)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setConstant(float c)</code> 비교값이 참인 경우 새로운 화소값으로 대체할 값을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 논리 비교 연산자를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target s1, Target s2, Target tgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Field Detail

### NONE

`public static final int NONE`

---

### EQUAL

`public static final int EQUAL`

`=` 연산자

---

### NOT\_EQUAL

`public static final int NOT_EQUAL`

`!=` 연산자

---

### LESS

`public static final int LESS`

< 연산자

---

## LESS\_THAN

```
public static final int LESS_THAN
```

<= 연산자

---

## GREATER

```
public static final int GREATER
```

> 연산자

---

## GREATER\_THAN

```
public static final int GREATER_THAN
```

>= 연산자

---

## Constructor Detail

### BinaryPixelValueReplaceOp

```
public BinaryPixelValueReplaceOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

논리 비교 연산자를 설정한다.

---

### setConstant

```
public void setConstant(float c)
```

비교값이 참인 경우 새로운 화소값으로 대체할 값을 설정한다.

---

### setBands

```
public void setBands(int b1,  
                    int b2)
```

입력 영상의 대상 분광대를 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target s1,  
                    Target s2,  
                    Target tgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

논리 비교 연산을 실행한다.

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                double[] b,  
                                int aOffset,  
                                float[] c,  
                                int bOffset,  
                                int length)
```

비교 연산을 수행한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

## Class BipToBilConvertOp

java.lang.Object  
|--BipToBilConvertOp

public class BipToBilConvertOp  
extends java.lang.Object  
implements [Operator](#)

BIP 포맷의 영상을 BIL 포맷으로 변환한다.

### Constructor Summary

<a href="#">BipToBilConvertOp()</a> Constructor
--

### Method Summary

void	<a href="#">operate()</a> 포맷 변환을 실행한다.
void	<a href="#">setBounds(int[] p)</a> 처리 영역을 설정한다.
void	<a href="#">setPixelType(java.lang.String type)</a> 입력 화소형을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상대바를 설정한다.
void	<a href="#">setTarget(java.lang.String f, <a href="#">Target</a> dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

#### BipToBilConvertOp

public BipToBilConvertOp()

Constructor



## Method Detail

### setBounds

```
public void setBounds(int[] p)
```

처리 영역을 설정한다.

---

### setPixelType

```
public void setPixelType(java.lang.String type)
```

입력 화소형을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(java.lang.String f,  
                      Target dtgt)
```

입출력 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

포맷 변환을 실행한다.

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class BsqToBilConvertOp

```
java.lang.Object
|
|--BsqToBilConvertOp
```

```
public class BsqToBilConvertOp
extends java.lang.Object
implements Operator
```

BSQ 포맷의 영상을 BIL 포맷으로 변환한다.

Constructor Summary	
<a href="#">BsqToBilConvertOp()</a>	Constructor

Method Summary	
void	<a href="#">operate()</a> 포맷 변환을 실행한다.
void	<a href="#">setBounds(int[] p)</a> 처리 영역을 설정한다.
void	<a href="#">setPixelType(java.lang.String type)</a> 입력 화소형을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(java.lang.String[] f, <a href="#">Target</a> dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
<a href="#">clone</a> , <a href="#">equals</a> , <a href="#">finalize</a> , <a href="#">getClass</a> , <a href="#">hashCode</a> , <a href="#">notify</a> , <a href="#">notifyAll</a> , <a href="#">toString</a> , <a href="#">wait</a> , <a href="#">wait</a> , <a href="#">wait</a>

## Constructor Detail

```
BsqToBilConvertOp
public BsqToBilConvertOp()
    Constructor
```

## Method Detail

### setBounds

```
public void setBounds(int[] p)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(java.lang.String[] f,  
                      Target dtgt)
```

입출력 영상을 설정한다.

---

### setPixelType

```
public void setPixelType(java.lang.String type)
```

입력 화소형을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

포맷 변환을 실행한다.

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class ChromaticityAnalysisOp

```
java.lang.Object
|
+--ChromaticityAnalysisOp
```

```
public class ChromaticityAnalysisOp
extends java.lang.Object
implements Operator
```

3개의 분광대를 이용하여 영상의 chromaticity X, Y, Z을 계산한다.  $X = \text{band 1} / (\text{band 1} + \text{band 2} + \text{band 3})$   $Y = \text{band 2} / (\text{band 1} + \text{band 2} + \text{band 3})$   $Z = 1 - (X + Y)$

Constructor Summary	
<a href="#">ChromaticityAnalysisOp()</a>	Constructor

Method Summary	
protected void	<a href="#">calculateIndex</a> (double[][] iPix, int aOffset, float[][] oPix, int bOffset, int length) Chromaticity를 계산한다.
void	<a href="#">operate</a> () 3개 분광대를 이용하여 Chromaticity를 계산한다.
void	<a href="#">setBands</a> (int[] bands) 처리 대상 분광대를 설정한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target stgt, Target dtgt) 대상 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail
<a href="#">ChromaticityAnalysisOp</a>

```
public ChromaticityAnalysisOp()
```

Constructor

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

처리 대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

3개 분광대를 이용하여 Chromaticity를 계산한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

### calculateIndex

```
protected void calculateIndex(double[][] iPix,  
                             int aOffset,  
                             float[][] oPix,  
                             int bOffset,  
                             int length)
```

Chromaticity를 계산한다.

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class `ClassifyErrorAnalysisOp`

```
java.lang.Object
|
|--ClassifyErrorAnalysisOp
```

```
public class ClassifyErrorAnalysisOp
extends java.lang.Object
implements Operator
```

분류 결과에 대한 Error matrix를 계산한다.

<b>Constructor Summary</b>	
<code>ClassifyErrorAnalysisOp()</code>	Constructor

<b>Method Summary</b>	
<code>java.lang.Object[]</code>	<code>getErrorMatrix()</code> Error matrix 를 되돌린다.
<code>void</code>	<code>operate()</code> Error matrix를 계산한다.
<code>void</code>	<code>setAOI(java.lang.String[] path)</code> AOI 파일의 배열을 설정한다.
<code>void</code>	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
<code>void</code>	<code>setTarget(Target tgt)</code> 대상 영상을 설정한다.
<code>void</code>	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

<b>Methods inherited from class <code>java.lang.Object</code></b>
<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

<b>Constructor Detail</b>
<b><code>ClassifyErrorAnalysisOp</code></b>

```
public ClassifyErrorAnalysisOp()
    Constructor
```

## Method Detail

### setAOI

```
public void setAOI(java.lang.String[] path)
    throws java.io.IOException
```

AOI 파일의 배열을 설정한다.

Throws:

java.io.IOException – IOException

---

### getErrorMatrix

```
public java.lang.Object[] getErrorMatrix()
```

Error matrix 를 되돌린다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target tgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()
    throws java.io.IOException,
           java.lang.InterruptedIOException
```

Error matrix를 계산한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedIOException – InterruptedIOException

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---



## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class ColorConvertOp

```
java.lang.Object
|
+--ColorConvertOp
```

```
public class ColorConvertOp
extends java.lang.Object
implements Operator
```

색 좌표계의 변환을 수행한다. 출력 화소형은 4바이트 실수형이다.

## Constructor Summary

<a href="#">ColorConvertOp()</a> Constructor
---

## Method Summary

protected void	<a href="#">convert</a> (byte[][] dPixel, int aOffset, float[][] obPixel, int bOffset, int length) 색좌표를 계산한다.
void	<a href="#">operate</a> () 색좌표 변환을 실행한다.
void	<a href="#">setBands</a> (int[] bands) 처리 대상 분광대를 설정한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<a href="#">setMethod</a> (java.lang.String in, java.lang.String out) 색 좌표 변환을 위한 입출력 좌표계를 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target stgt, Target dtgt) 대상 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructor Detail

[ColorConvertOp](#)

public ColorConvertOp()

Constructor

## Method Detail

### setMethod

public void setMethod(java.lang.String in,  
                      java.lang.String out)

색 좌표 변환을 위한 입출력 좌표계를 설정한다. 지원되는 색좌표계는 RGB, B/W, HSB, HSI, HLS, YCbCr, CYMK, CYM, YIQ, YUV 이다.

---

### setBounds

public void setBounds(java.awt.Rectangle r)

처리 영역을 설정한다.

---

### setBands

public void setBands(int[] bands)

처리 대상 분광대를 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

public void setTarget(Target stgt,  
                      Target dtgt)

대상 영상을 설정한다.

---

### operate

public void operate()  
                  throws java.io.IOException,  
                          java.lang.InterruptedException

색좌표 변환을 실행한다.

---

### convert

```
protected void convert(byte[][] dPixel,  
                       int aOffset,  
                       float[][] obPixel,  
                       int bOffset,  
                       int length)
```

색좌표를 계산한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

# Class Tree ~~Deprecated~~ Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class Command

java.lang.Object  
|  
+--Command

public class Command  
extends java.lang.Object  
implements java.io.Serializable

GeoPixel의 명령어를 관리한다.

See Also:  
[Serialized Form](#)

Field Summary	
java.lang.String	<a href="#">Description</a>
java.lang.String	<a href="#">ID</a>

Constructor Summary	
<a href="#">Command()</a> Constructor	
<a href="#">Command(java.lang.String id, java.lang.String descrip)</a> 명령어 식별자와 설명문을 설정한다.	

Method Summary	
boolean	<a href="#">equals(java.lang.Object obj)</a> 현재의 명령어 객체와의 동일 여부를 판단한다.
java.lang.String	<a href="#">getDescription()</a> 현재의 명령어 설명문을 되돌린다.
java.lang.String	<a href="#">getID()</a> 현재의 명령어 식별자를 되돌린다.
int	<a href="#">hashCode()</a> 현재의 명령어 식별자의 해쉬코드를 되돌린다.
java.lang.String	<a href="#">toString()</a> 현재의 명령어 식별자와 설명문을 문자열로 되돌린다.

Methods inherited from class java.lang.Object	
clone, finalize, getClass, notify, notifyAll, wait, wait, wait	

## Field Detail

### ID

```
public java.lang.String ID
```

---

### Description

```
public java.lang.String Description
```

## Constructor Detail

### Command

```
public Command()
```

Constructor

---

### Command

```
public Command(java.lang.String id,  
               java.lang.String descrip)
```

명령어 식별자와 설명문을 설정한다.

## Method Detail

### getID

```
public java.lang.String getID()
```

현재의 명령어 식별자를 되돌린다.

---

### getDescription

```
public java.lang.String getDescription()
```

현재의 명령어 설명문을 되돌린다.

---

### hashCode

```
public int hashCode()
```

현재의 명령어 식별자의 해쉬코드를 되돌린다.

Overrides:

hashCode in class java.lang.Object

---

### equals

public boolean equals(java.lang.Object obj)

현재의 명령어 객체와의 동일 여부를 판단한다. 동일한 식별자일 경우 true, 아닐 경우 false를 되돌린다.

Overrides:

equals in class java.lang.Object

---

## toString

public java.lang.String toString()

현재의 명령어 식별자와 설명문을 문자열로 되돌린다.

Overrides:

toString in class java.lang.Object

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class CommandLoader

```
java.lang.Object
|
+--java.lang.ClassLoader
|
+--CommandLoader
```

```
public class CommandLoader
extends java.lang.ClassLoader
```

선택된 명령어에 대한 객체를 가져온다

## Constructor Summary

<a href="#">CommandLoader()</a> Constructor
--

## Method Summary

java.lang.Class	<a href="#">loadClass</a> (java.lang.String className) 명령어에 대한 클래스를 읽어들이는다.
java.lang.Class	<a href="#">loadClass</a> (java.lang.String className, boolean resolve) 명령어에 대한 클래스를 읽어들이는다.

## Methods inherited from class java.lang.ClassLoader

[defineClass](#), [defineClass](#), [defineClass](#), [definePackage](#), [findClass](#), [findLibrary](#), [findLoadedClass](#), [findResource](#), [findResources](#), [findSystemClass](#), [getPackage](#), [getPackages](#), [getParent](#), [getResource](#), [getResourceAsStream](#), [getResources](#), [getSystemClassLoader](#), [getSystemResource](#), [getSystemResourceAsStream](#), [getSystemResources](#), [resolveClass](#), [setSigners](#)

## Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructor Detail

### CommandLoader

```
public CommandLoader()
```

Constructor

## Method Detail

### loadClass

```
public java.lang.Class loadClass(java.lang.String className)
    throws java.lang.ClassNotFoundException
```

명령어에 대한 클래스를 읽어들인다.

Overrides:

loadClass in class java.lang.ClassLoader

---

## loadClass

```
public java.lang.Class loadClass(java.lang.String className,
    boolean resolve)
    throws java.lang.ClassNotFoundException
```

명령어에 대한 클래스를 읽어들인다.

Overrides:

loadClass in class java.lang.ClassLoader

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class ConvolveOp

java.lang.Object  
|  
+--ConvolveOp

---

public class ConvolveOp  
extends java.lang.Object

커널과 배열에 대한 컨볼루션을 실행한다.

---

## Constructor Summary

<code>ConvolveOp(GeoKernel k, int d)</code> Constructor
--

## Method Summary

float	<code>convolve(double[][] m)</code> 미리 정의된 커널과 입력배열과의 컨볼루션 값을 구한다.
void	<code>setDivisor(int d)</code> 컨볼루션 결과에 대한 최종값을 결정하기 위한 가중치를 설정한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### ConvolveOp

public ConvolveOp([GeoKernel](#) k,  
int d)

Constructor

## Method Detail

### setDivisor

public void setDivisor(int d)

컨볼루션 결과에 대한 최종값을 결정하기 위한 가중치를 설정한다.

---

## convolve

```
public float convolve(double[][] m)
```

미리 정의된 커널과 입력배열과의 콘볼루션 값을 구한다.

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree ~~Deprecated~~ Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class CoordConvertOp

CoordConvertOp

```
public class CoordConvertOp
```

선정된 영역의 좌표를 구한다.

## Constructor Summary

[CoordConvertOp\(\)](#)  
Constructor

[CoordConvertOp\(int smps, int lns, double xmin, double xmax, double ymin, double ymax\)](#)  
Constructor

## Method Summary

<a href="#">GeoRectangle</a>	<a href="#">getBounds()</a> 선택된 영역을 되돌린다.
int	<a href="#">getImageColumn(double x)</a> 입력 x좌표값에 대한 영상의 Column값을 되돌린다.
int	<a href="#">getImageRow(double y)</a> 입력 y좌표값에 대한 영상의 Row값을 되돌린다.
<a href="#">java.awt.Dimension</a>	<a href="#">getImageSize()</a> 영상의 크기를 되돌린다.
double	<a href="#">getMapX(int col)</a> 입력 Column에 대한 지도의 X좌표값을 되돌린다.
double	<a href="#">getMapY(int row)</a> 입력 Row에 대한 지도의 Y좌표값을 되돌린다.
void	<a href="#">setBounds(double xmin, double xmax, double ymin, double ymax)</a> 영역을 설정한다.
void	<a href="#">setBounds(GeoRectangle r)</a> 영역을 설정한다.
void	<a href="#">setImageSize(int smps, int lns)</a> 영상의 크기를 설정한다.

## Constructor Detail

CoordConvertOp

```
public CoordConvertOp()
```

## Constructor

---

### CoordConvertOp

```
public CoordConvertOp(int smps,  
                      int lns,  
                      double xmin,  
                      double xmax,  
                      double ymin,  
                      double ymax)
```

## Constructor

### Method Detail

#### setImageSize

```
public void setImageSize(int smps,  
                         int lns)
```

영상의 크기를 설정한다.

---

#### setBounds

```
public void setBounds(double xmin,  
                      double xmax,  
                      double ymin,  
                      double ymax)
```

영역을 설정한다.

---

#### setBounds

```
public void setBounds(GeoRectangle r)
```

영역을 설정한다.

---

#### getImageSize

```
public java.awt.Dimension getImageSize()
```

영상의 크기를 되돌린다.

---

#### getBounds

```
public GeoRectangle getBounds()
```

선택된 영역을 되돌린다.

---

## getMapX

`public double getMapX(int col)`

입력 Column에 대한 지도의 X좌표값을 되돌린다.

---

## getMapY

`public double getMapY(int row)`

입력 Row에 대한 지도의 Y좌표값을 되돌린다.

---

## getImageColumn

`public int getImageColumn(double x)`

입력 x좌표값에 대한 영상의 Column값을 되돌린다.

---

## getImageRow

`public int getImageRow(double y)`

입력 y좌표값에 대한 영상의 Row값을 되돌린다.

---

---

### [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class CrossMark

```
java.lang.Object  
|  
+--CrossMark
```

```
public class CrossMark  
extends java.lang.Object  
implements Drawable
```

CrossMark를 표시한다.

<h3>Constructor Summary</h3>
<a href="#">CrossMark</a> (int x, int y) Constructor
<a href="#">CrossMark</a> (int x, int y, int radius) Constructor

## Method Summary

void	<code>draw(java.awt.Graphics g, float scale)</code> CrossMark를 그린다.
void	<code>draw(java.awt.Graphics g, int xoffset, int yoffset)</code> CrossMark를 그린다.
java.awt.Rectangle	<code>getBounds()</code> 영역을 되돌린다.
java.awt.Rectangle	<code>getBounds(float scale)</code> Scaling된 좌표의 영역을 되돌린다.
java.awt.Rectangle	<code>getBounds(int xoffset, int yoffset)</code> x, y 방향으로 이동된 좌표의 영역을 되돌린다.
java.awt.Point	<code>getLocation()</code> CrossMark의 위치를 되돌린다.
int	<code>getRadius()</code> CrossMark의 반경을 되돌린다.
int	<code>getType()</code> CrossMark의 type을 되돌린다.
void	<code>setColor(java.awt.Color color)</code> CrossMark의 색을 설정한다.
void	<code>setLocation(int x, int y)</code> CrossMark의 위치를 설정한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### CrossMark

```
public CrossMark(int x,  
                 int y,  
                 int radius)
```

Constructor

---

### CrossMark

```
public CrossMark(int x,  
                 int y)
```

Constructor

## Method Detail

### getType

```
public int getType()
```

CrossMark의 type을 되돌린다.  
Specified by:  
getType in interface Drawable

---

## setColor

```
public void setColor(java.awt.Color color)
```

CrossMark의 색을 설정한다.  
Specified by:  
setColor in interface Drawable

---

## getLocation

```
public java.awt.Point getLocation()
```

CrossMark의 위치를 되돌린다.  
Specified by:  
getLocation in interface Drawable

---

## setLocation

```
public void setLocation(int x,  
                        int y)
```

CrossMark의 위치를 설정한다.  
Specified by:  
setLocation in interface Drawable

---

## getBounds

```
public java.awt.Rectangle getBounds()
```

영역을 되돌린다.  
Specified by:  
getBounds in interface Drawable

---

## getBounds

```
public java.awt.Rectangle getBounds(int xoffset,  
                                    int yoffset)
```



x, y 방향으로 이동된 좌표의 영역을 되돌린다.  
Specified by:  
[getBounds](#) in interface [Drawable](#)

---

## getBounds

```
public java.awt.Rectangle getBounds(float scale)
```

Scaling된 좌표의 영역을 되돌린다.  
Specified by:  
[getBounds](#) in interface [Drawable](#)

---

## getRadius

```
public int getRadius()
```

CrossMark의 반경을 되돌린다.

---

## draw

```
public void draw(java.awt.Graphics g,  
                 int xoffset,  
                 int yoffset)
```

CrossMark를 그린다.  
Specified by:  
[draw](#) in interface [Drawable](#)

---

## draw

```
public void draw(java.awt.Graphics g,  
                 float scale)
```

CrossMark를 그린다.  
Specified by:  
[draw](#) in interface [Drawable](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Cube

java.lang.Object  
|  
+--Cube

Direct Known Subclasses:  
[CubeHeader](#)

---

```
public class Cube
extends java.lang.Object
implements java.lang.Cloneable
```

영상에 대한 정보를 관리하는 클래스이다.

Version:  
1.0  
Author:  
GeoPixel team

---

## Field Summary

int	<u>Bands</u>	영상의 분광대 수를 정의한다.
int	<u>BndOffset</u>	영상의 Bands에 대한 오프셋을 정의한다.
static int	<u>BYTE</u>	바이트형을 정의한다.
static int	<u>FLOAT</u>	4바이트 부동소수점형을 정의한다.
static int	<u>INT</u>	4바이트 정수형을 정의한다.
int	<u>Lines</u>	영상의 Lines값을 정의한다.
int	<u>LnOffset</u>	영상의 Lines에 대한 오프셋을 정의한다.
int	<u>PixelFormat</u>	영상의 화소형을 정의한다.
int	<u>Samples</u>	영상의 Samples값을 정의한다.
static int	<u>SHORT</u>	2바이트 정수형(Signed)을 정의한다.
int	<u>SmpOffset</u>	영상의 Samples에 대한 오프셋을 정의한다.
static int	<u>UNDEFINED</u>	
static int	<u>USHORT</u>	2바이트 정수형(Unsigned)을 정의한다.

## Constructor Summary

Cube() 영상을 선언한다.

Cube(int type, int soff, int loff, int boff, int smps, int lns, int bnds)  
주어진 변수에 대한 영상을 선언한다.

여 백

## BYTE

```
public static final int BYTE
```

바이트형을 정의한다.

---

## USHORT

```
public static final int USHORT
```

2바이트 정수형 (Unsigned)을 정의한다.

---

## SHORT

```
public static final int SHORT
```

2바이트 정수형 (Signed)을 정의한다.

---

## INT

```
public static final int INT
```

4바이트 정수형을 정의한다.

---

## FLOAT

```
public static final int FLOAT
```

4바이트 부동소수점형을 정의한다.

---

## PixelFormat

```
public int PixelType
```

영상의 화소형을 정의한다.

---

## SmpOffset

```
public int SmpOffset
```

영상의 Samples에 대한 오프셋을 정의한다.

---

## LnOffset

```
public int LnOffset
```

영상의 Lines에 대한 오프셋을 정의한다.

---

## BndOffset

```
public int BndOffset
```

영상의 Bands에 대한 오프셋을 정의한다.

---

## Samples

```
public int Samples
```

영상의 Samples값을 정의한다.

---

## Lines

```
public int Lines
```

영상의 Lines값을 정의한다.

---

## Bands

```
public int Bands
```

영상의 분광대 수를 정의한다.

---

## Constructor Detail

### Cube

```
public Cube()
```

영상을 선언한다.

---

### Cube

```
public Cube(int type,
            int soff,
            int loff,
            int boff,
            int smps,
            int lns,
            int bnds)
    throws java.lang.IllegalArgumentException
```

주어진 변수에 대한 영상을 선언한다.

## Method Detail

### setPixelType

```
public void setPixelType(int type)
```

화소형을 지정한다.

Parameters:

type - 화소형

Returns:

없음

Throws:

없음 -

---

### setPixelType

```
public void setPixelType(java.lang.String type)
```

화소형을 지정한다.

Parameters:

type - 화소형

---

### setLocation

```
public void setLocation(int soff,
                        int loff,
                        int boff)
    throws java.lang.IllegalArgumentException
```

영상에서의 물리적 위치를 지정한다.

Parameters:

soff - Sample에 대한 오프셋

loff - Lines에 대한 오프셋

boff - Bands에 대한 오프셋

Returns:

없음

Throws:

없음 -

## translate

```
public void translate(int dSmp,  
                    int dLn,  
                    int dBnd)  
    throws java.lang.IllegalArgumentException
```

영상에서의 물리적 위치를 지정한다.

### Parameters:

soff - Sample에 대한 오프셋  
lff - Lines에 대한 오프셋  
bff - Bands에 대한 오프셋

### Returns:

없음

### Throws:

없음 -

---

## setVolume

```
public void setVolume(int smps,  
                    int lns,  
                    int bnds)  
    throws java.lang.IllegalArgumentException
```

영상의 크기를 정의한다.

---

## isSameVolume

```
public boolean isSameVolume(Cube cb)
```

입력 영상과 같은 물리적 크기를 갖는지 검사한다.

---

## clone

```
public java.lang.Object clone()
```

현재의 영상을 복사하여 되돌린다.

### Overrides:

clone in class java.lang.Object

---

## setVolume

```
public void setVolume(Volume v)
```

현재 정의된 영상과 같은 크기를 갖는 영상을 생성한다.

---

## getVolume



```
public Volume getVolume()
```

현재 정의된 영상과 같은 크기를 갖는 영상을 생성, 되돌린다.

---

## getPixelTypeNo

```
public static int getPixelTypeNo(java.lang.String type)
```

영상의 화소형을 되돌린다.

---

## getPromotedType

```
public static int getPromotedType(int type)
```

영상의 화소형을 되돌린다.

---

## getPixelTypeString

```
public static java.lang.String getPixelTypeString(int type)
```

영상의 화소형을 되돌린다.

---

## isValidPixelType

```
public static boolean isValidPixelType(int type)
```

화소형의 유효 여부를 검사한다.

---

## getPixelByteSize

```
public static int getPixelByteSize(int type)
```

화소형의 바이트 크기를 되돌린다.

---

## getBundleByteSize

```
public int getBundleByteSize()
```

영상에 대해 한 라인에 대해 전체 분광대를 처리 할 수 있는 바이트 크기를 되돌린다.

---

## getByteSize

```
public int getByteSize()
```

영상의 전체 바이트 크기를 되돌린다.

---

# Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

## Class CubeFileToMemoryOp

java.lang.Object  
|--CubeFileToMemoryOp

public class CubeFileToMemoryOp  
extends java.lang.Object  
implements Operator

영상을 메모리에 등록한다.

### Constructor Summary

CubeFileToMemoryOp()  
Constructor

### Method Summary

void	<u>operate()</u> 입력영상을 메모리형의 영상으로 변환한다.
void	<u>setBands(int[] bands)</u> 입력 영상의 대상 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리 영역을 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setTarget(Target.File srcTgt, Target.Memory destTgt)</u> 입출력 영상을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

CubeFileToMemoryOp

public CubeFileToMemoryOp()

Constructor

## Method Detail

### setTarget

```
public void setTarget(Target.File srcTgt,  
                    Target.Memory destTgt)
```

입출력 영상을 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상을 메모리형의 영상으로 변환한다.

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class CubeGeometryFlipOp

```
java.lang.Object
|
+--CubeGeometryFlipOp
```

```
public class CubeGeometryFlipOp
extends java.lang.Object
implements Operator
```

영상에 대하여 수평, 수직 또는 전이를 실행한다.

## Constructor Summary

<a href="#">CubeGeometryFlipOp()</a> Constructor
---

## Method Summary

protected void	<a href="#">HorizontalFlip</a> (byte[] iPixel, int aOffset, byte[][] oPixel, int index, int bOffset, int smps) 바이트형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (byte[] iPixel, int aOffset, byte[] oPixel, int smps) 바이트형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (float[] iPixel, int aOffset, float[][] oPixel, int index, int bOffset, int smps) 4바이트 실수형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (float[] iPixel, int aOffset, float[] oPixel, int smps) 4바이트 실수형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (int[] iPixel, int aOffset, int[][] oPixel, int index, int bOffset, int smps) 4바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (int[] iPixel, int aOffset, int[] oPixel, int smps) 4바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (short[] iPixel, int aOffset, short[][] oPixel, int index, int bOffset, int smps) 2바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.
protected void	<a href="#">HorizontalFlip</a> (short[] iPixel, int aOffset, short[] oPixel, int smps) 2바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.
void	<a href="#">operate</a> () 입력영상의 Geometry를 변환한다.

void	<u>setMethod</u> (java.lang.String method) 처리 방법을 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.
protected void	<u>Transpose</u> (byte[] iPixel, byte[] oPixel, int lns, int smps, int l) 바이트형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (byte[] iPixel, int aOffset, byte[][] oPixel, int index, int line, int smps) 바이트형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (byte[] iPixel, int aOffset, byte[] oPixel, int line, int smps) 바이트형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (float[] iPixel, float[] oPixel, int lns, int smps, int l) 4바이트형 실수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (float[] iPixel, int aOffset, float[][] oPixel, int index, int line, int smps) 4바이트 실수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (float[] iPixel, int aOffset, float[] oPixel, int line, int smps) 4바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (int[] iPixel, int[] oPixel, int lns, int smps, int l) 4바이트형 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (int[] iPixel, int aOffset, int[][] oPixel, int index, int line, int smps) 4바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (int[] iPixel, int aOffset, int[] oPixel, int line, int smps) 4바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (short[] iPixel, int aOffset, short[][] oPixel, int index, int line, int smps) 2바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (short[] iPixel, int aOffset, short[] oPixel, int line, int smps) 2바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (short[] iPixel, short[] oPixel, int lns, int smps, int l) 2바이트형 정수형 배열의 화소를 전이배열 한다.
protected void	<u>VerticalFlip</u> (byte[] iPixel, int aOffset, byte[][] oPixel, int index, int bOffset, int smps) 바이트형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<u>VerticalFlip</u> (byte[] iPixel, int aOffset, byte[] oPixel, int smps) 바이트형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<u>VerticalFlip</u> (float[] iPixel, int aOffset, float[][] oPixel, int index, int bOffset, int smps) 4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<u>VerticalFlip</u> (float[] iPixel, int aOffset, float[] oPixel, int smps) 4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.

void	<u>setBands</u> (int[] band) 입력 영상의 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<u>setMethod</u> (java.lang.String method) 처리 방법을 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.
protected void	<u>Transpose</u> (byte[] iPixel, byte[] oPixel, int lns, int smps, int l) 바이트형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (byte[] iPixel, int aOffset, byte[][] oPixel, int index, int line, int smps) 바이트형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (byte[] iPixel, int aOffset, byte[] oPixel, int line, int smps) 바이트형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (float[] iPixel, float[] oPixel, int lns, int smps, int l) 4바이트형 실수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (float[] iPixel, int aOffset, float[][] oPixel, int index, int line, int smps) 4바이트 실수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (float[] iPixel, int aOffset, float[] oPixel, int line, int smps) 4바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (int[] iPixel, int[] oPixel, int lns, int smps, int l) 4바이트형 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (int[] iPixel, int aOffset, int[][] oPixel, int index, int line, int smps) 4바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (int[] iPixel, int aOffset, int[] oPixel, int line, int smps) 4바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (short[] iPixel, int aOffset, short[][] oPixel, int index, int line, int smps) 2바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (short[] iPixel, int aOffset, short[] oPixel, int line, int smps) 2바이트 정수형 배열의 화소를 전이배열 한다.
protected void	<u>Transpose</u> (short[] iPixel, short[] oPixel, int lns, int smps, int l) 2바이트형 정수형 배열의 화소를 전이배열 한다.
protected void	<u>VerticalFlip</u> (byte[] iPixel, int aOffset, byte[][] oPixel, int index, int bOffset, int smps) 바이트형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<u>VerticalFlip</u> (byte[] iPixel, int aOffset, byte[] oPixel, int smps) 바이트형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<u>VerticalFlip</u> (float[] iPixel, int aOffset, float[][] oPixel, int index, int bOffset, int smps) 4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.

protected void	<code>VerticalFlip(float[] iPixel, int aOffset, float[] oPixel, int smps)</code> 4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<code>VerticalFlip(int[] iPixel, int aOffset, int[][] oPixel, int index, int bOffset, int smps)</code> 4바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<code>VerticalFlip(int[] iPixel, int aOffset, int[] oPixel, int smps)</code> 4바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<code>VerticalFlip(short[] iPixel, int aOffset, short[][] oPixel, int index, int bOffset, int smps)</code> 2바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.
protected void	<code>VerticalFlip(short[] iPixel, int aOffset, short[] oPixel, int smps)</code> 2바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### CubeGeometryFlipOp

`public CubeGeometryFlipOp()`

Constructor

## Method Detail

### setMethod

`public void setMethod(java.lang.String method)`

처리 방법을 설정한다.

---

### setBounds

`public void setBounds(java.awt.Rectangle r)`

처리 영역을 설정한다.

---

### setBands

`public void setBands(int[] band)`

입력 영상의 대상 분광대를 설정한다.

---

### setProgressStream

`public void setProgressStream(ProgressStream prog)`

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 Geometry를 변환한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(byte[] iPixel,  
                              int aOffset,  
                              byte[] oPixel,  
                              int smps)
```

바이트형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(short[] iPixel,  
                              int aOffset,  
                              short[] oPixel,  
                              int smps)
```

2바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(int[] iPixel,  
                              int aOffset,  
                              int[] oPixel,  
                              int smps)
```

4바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(float[] iPixel,
```



```
int aOffset,  
float[] oPixel,  
int smps)
```

4바이트 실수형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(byte[] iPixel,  
int aOffset,  
byte[][] oPixel,  
int index,  
int bOffset,  
int smps)
```

바이트형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(short[] iPixel,  
int aOffset,  
short[][] oPixel,  
int index,  
int bOffset,  
int smps)
```

2바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(int[] iPixel,  
int aOffset,  
int[][] oPixel,  
int index,  
int bOffset,  
int smps)
```

4바이트 정수형 배열의 화소를 수평방향으로 재배열 한다.

---

## HorizontalFlip

```
protected void HorizontalFlip(float[] iPixel,  
int aOffset,  
float[][] oPixel,  
int index,  
int bOffset,  
int smps)
```

4바이트 실수형 배열의 화소를 수평방향으로 재배열 한다.

---

## Transpose

```
protected void Transpose(byte[] iPixel,  
byte[] oPixel,  
int lns,
```

```
int smps,  
int l)
```

바이트형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(short[] iPixel,  
                          short[] oPixel,  
                          int lns,  
                          int smps,  
                          int l)
```

2바이트형 정수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(int[] iPixel,  
                          int[] oPixel,  
                          int lns,  
                          int smps,  
                          int l)
```

4바이트형 정수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(float[] iPixel,  
                          float[] oPixel,  
                          int lns,  
                          int smps,  
                          int l)
```

4바이트형 실수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(byte[] iPixel,  
                          int aOffset,  
                          byte[][] oPixel,  
                          int index,  
                          int line,  
                          int smps)
```

바이트형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(short[] iPixel,  
                          int aOffset,  
                          short[][] oPixel,  
                          int index,  
                          int line,  
                          int smps)
```

2바이트 정수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(int[] iPixel,  
                          int aOffset,  
                          int[][] oPixel,  
                          int index,  
                          int line,  
                          int smps)
```

4바이트 정수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(float[] iPixel,  
                          int aOffset,  
                          float[][] oPixel,  
                          int index,  
                          int line,  
                          int smps)
```

4바이트 실수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(byte[] iPixel,  
                          int aOffset,  
                          byte[] oPixel,  
                          int line,  
                          int smps)
```

바이트형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(short[] iPixel,  
                          int aOffset,  
                          short[] oPixel,  
                          int line,  
                          int smps)
```

2바이트 정수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(int[] iPixel,  
                          int aOffset,  
                          int[] oPixel,  
                          int line,  
                          int smps)
```

4바이트 정수형 배열의 화소를 전이배열 한다.

---

## Transpose

```
protected void Transpose(float[] iPixel,  
                        int aOffset,  
                        float[] oPixel,  
                        int line,  
                        int smps)
```

4바이트 정수형 배열의 화소를 전이배열 한다.

---

## VerticalFlip

```
protected void VerticalFlip(byte[] iPixel,  
                           int aOffset,  
                           byte[] oPixel,  
                           int smps)
```

바이트형 배열의 화소를 수직방향으로 재배열 한다.

---

## VerticalFlip

```
protected void VerticalFlip(short[] iPixel,  
                           int aOffset,  
                           short[] oPixel,  
                           int smps)
```

2바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

---

## VerticalFlip

```
protected void VerticalFlip(int[] iPixel,  
                           int aOffset,  
                           int[] oPixel,  
                           int smps)
```

4바이트 정수형 배열의 화소를 수직방향으로 재배열 한다.

---

## VerticalFlip

```
protected void VerticalFlip(float[] iPixel,  
                           int aOffset,  
                           float[] oPixel,  
                           int smps)
```

4바이트 실수형 배열의 화소를 수직방향으로 재배열 한다.

---

## VerticalFlip

```
protected void VerticalFlip(byte[] iPixel,  
                           int aOffset,  
                           byte[][] oPixel,  
                           int index,  
                           int bOffset,
```

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class CubeGeometryResizeOp

java.lang.Object  
|  
+--CubeGeometryResizeOp

```
public class CubeGeometryResizeOp
extends java.lang.Object
implements Operator
```

영상의 크기를 재조절한다.

### Constructor Summary

CubeGeometryResizeOp() Constructor
---------------------------------------

### Method Summary

void	<a href="#">operate()</a> 입력영상의 크기를 변환한다.
void	<a href="#">setBands(int[] band)</a> 입력 영상의 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리 영역을 설정한다.
void	<a href="#">setMethod(java.lang.String method)</a> 재배열 방법을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setScale(float scale)</a> 크기 변환을 위한 스케일을 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### CubeGeometryResizeOp

```
public CubeGeometryResizeOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

재배열 방법을 설정한다.

---

### setBands

```
public void setBands(int[] band)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setScale

```
public void setScale(float scale)
```

크기 변환을 위한 스케일을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 크기를 변환한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class CubeGeometryRotateOp

java.lang.Object

└--CubeGeometryRotateOp

```
public class CubeGeometryRotateOp
    extends java.lang.Object
    implements Operator
```

영상을 임의의 각도로 회전한다.

### Constructor Summary

<a href="#">CubeGeometryRotateOp()</a> Constructor
---

### Method Summary

void	<a href="#">operate()</a> 영상을 회전한다.
void	<a href="#">setAngle(float angle)</a> 영상의 회전각도를 설정한다.
void	<a href="#">setBands(int[] band)</a> 입력 영상의 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리 영역을 설정한다.
void	<a href="#">setMethod(java.lang.String method)</a> 재배열 방법을 설정한다.
void	<a href="#">setOrigin(java.lang.String origin)</a> 회전의 중심축을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.



## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### CubeGeometryRotateOp

```
public CubeGeometryRotateOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

재배열 방법을 설정한다.

---

### setBands

```
public void setBands(int[] band)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setOrigin

```
public void setOrigin(java.lang.String origin)
```

회전의 중심축을 설정한다

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## setAngle

```
public void setAngle(float angle)
```

영상의 회전각도를 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상을 회전한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class CubeGeometryTranslateOp

```
java.lang.Object
|
+--CubeGeometryTranslateOp
```

```
public class CubeGeometryTranslateOp
extends java.lang.Object
implements Operator
```

영상을 x, y방향으로 이동한다.

### Constructor Summary

<code>CubeGeometryTranslateOp()</code> Constructor
---

### Method Summary

void	<code>operate()</code> 영상을 수평, 수직방향으로 이동한다.
void	<code>setBands(int[] band)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 재배열 방법을 설정한다.
void	<code>setParameter(float[] p)</code> 수평, 수직 이동량을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### CubeGeometryTranslateOp

```
public CubeGeometryTranslateOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

재배열 방법을 설정한다.

---

### setBands

```
public void setBands(int[] band)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setParameter

```
public void setParameter(float[] p)
```

수평, 수직 이동량을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상을 수평, 수직방향으로 이동한다.

**Throws:**

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

**Specified by:**

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

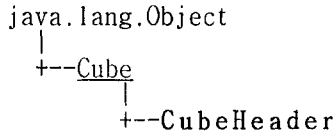
---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class CubeHeader



```
public class CubeHeader
extends Cube
```

Field Summary	
static java.lang.String	<b>COMMENT</b> 영상에 대한 주석
java.lang.String	<b>Title</b> 영상의 Title
boolean[]	<b>validMinMax</b> 분광대의 최소, 최대값 유효성
double	<b>Xmax</b> x 방향의 최소값
double	<b>Xmin</b> x 방향의 최대값
double	<b>Ymax</b> y 방향의 최대값
double	<b>Ymin</b> y 방향의 최소값
double[]	<b>Zmax</b> 각 분광대의 최대값
double[]	<b>Zmin</b> 각 분광대의 최소값

Fields inherited from class <a href="#">Cube</a>
<a href="#">Bands</a> , <a href="#">BndOffset</a> , <a href="#">BYTE</a> , <a href="#">FLOAT</a> , <a href="#">INT</a> , <a href="#">Lines</a> , <a href="#">LnOffset</a> , <a href="#">PixelFormat</a> , <a href="#">Samples</a> , <a href="#">SHORT</a> , <a href="#">SmpOffset</a> , <a href="#">UNDEFINED</a> , <a href="#">USHORT</a>

## Constructor Summary

`CubeHeader()`

영상의 헤더를 선언한다.

`CubeHeader(int type, int smps, int lns, int bnds)`

영상의 헤더를 선언한다.

`CubeHeader(java.lang.String title)`

영상의 헤더를 선언한다.

`CubeHeader(java.lang.String title, int type)`

영상의 헤더를 선언한다.

`CubeHeader(java.lang.String title, int type, int smps, int lns, int bnds)`

영상의 헤더를 선언한다.

Method Summary	
void	<u>assert()</u> 영상의 헤더 정보가 정확하게 설정되어 있는지를 확인한다.
java.lang.Object	<u>clone()</u> 현재의 헤더 정보를 복사한다.
java.lang.String	<u>getBandName(int index)</u> 영상의 분광대 목록에서 주어진 인덱스의 분광대 이름을 되돌린다.
java.lang.String	<u>getBandNameForList(int index)</u> 영상의 목록에서 주어진 인덱스의 분광대 이름을 되돌린다.
GeoRectangle	<u>getGeoCoord()</u> 영상의 x, y 방향에 대한 최소, 최대값을 갖는 영역을 되돌린다.
GeoRectangle	<u>getScaledGeoCoord(java.awt.Rectangle r)</u> 입력 범위에 대하여 새로운 좌표값을 계산하여 되돌린다.
boolean	<u>insideRegion(int smp, int lns, int bnd)</u> 입력된 화소 위치가 영상의 유효범위에 있는지를 판단한다.
boolean	<u>isEmpty()</u> 영상의 헤더정보가 정의 되어 있는지 판단한다.
void	<u>read(java.io.File f)</u> 영상의 헤더 정보를 읽는다.
void	<u>read(java.lang.String path)</u> 영상의 헤더 정보를 읽는다.
void	<u>setBandName(int index, java.lang.String name)</u> 대상 영상의 주어진 분광대의 이름을 정의한다
void	<u>setGeoCoord(double xmin, double xmax, double ymin, double ymax)</u> 입력 범위에 대하여 영상의 x, y 방향에 대한 최소, 최대값을 설정한다.
void	<u>setGeoCoord(GeoRectangle r)</u> 입력 범위에 대하여 영상의 x, y 방향에 대한 최소, 최대값을 설정한다.
void	<u>setMinMax(int band, float min, float max)</u> 대상 영상의 분광대의 최소값과 최대값을 정의한다
void	<u>setTitle(java.lang.String title)</u> 헤더 정보의 Title을 설정한다.
void	<u>setVolume(int smpls, int lns, int bnds)</u> 입력된 Samples, Lines, Bands에 대하여 영상의 메모리 영역을 확보한다.
java.lang.String	<u>toString()</u> 영상의 헤더 정보를 문자열로 출력한다.
void	<u>write(java.io.File f)</u> 영상의 헤더 정보를 파일로 저장한다.
void	<u>write(java.lang.String path)</u> 영상의 헤더 정보를 파일로 저장한다.



## Methods inherited from class `Cube`

`getBundleByteSize`, `getByteSize`, `getPixelByteSize`, `getPixelTypeNo`, `getPixelTypeString`, `getPromotedType`, `getVolume`, `isSameVolume`, `isValidPixelType`, `setLocation`, `setPixelType`, `setPixelType`, `setVolume`, `translate`

## Methods inherited from class `java.lang.Object`

`equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### COMMENT

`public static java.lang.String COMMENT`

영상에 대한 주석

---

### Title

`public java.lang.String Title`

영상의 Title

---

### Xmin

`public double Xmin`

x 방향의 최대값

---

### Xmax

`public double Xmax`

x 방향의 최소값

---

### Ymin

`public double Ymin`

y 방향의 최소값

---

### Ymax

`public double Ymax`

y 방향의 최대값

---

## Zmin

`public double[] Zmin`

각 분광대의 최소값

---

## Zmax

`public double[] Zmax`

각 분광대의 최대값

---

## validMinMax

`public boolean[] validMinMax`

분광대의 최소, 최대값 유효성

## Constructor Detail

### CubeHeader

`public CubeHeader()`

영상의 헤더를 선언한다. Title이외의 헤더 정보는 초기화 된다.

---

### CubeHeader

`public CubeHeader(java.lang.String title)`

영상의 헤더를 선언한다. Title이외의 헤더 정보는 초기화 된다.

---

### CubeHeader

`public CubeHeader(java.lang.String title,  
int type)`

영상의 헤더를 선언한다. Title과 화소형 이외의 헤더 정보는 초기화 된다.

---

## CubeHeader

```
public CubeHeader(java.lang.String title,  
                  int type,  
                  int smps,  
                  int lns,  
                  int bnds)
```

영상의 헤더를 선언한다.

---

## CubeHeader

```
public CubeHeader(int type,  
                  int smps,  
                  int lns,  
                  int bnds)
```

영상의 헤더를 선언한다.

## Method Detail

### setTitle

```
public void setTitle(java.lang.String title)
```

헤더 정보의 Title을 설정한다.

---

### setVolume

```
public void setVolume(int smps,  
                      int lns,  
                      int bnds)
```

입력된 Samples, Lines, Bands에 대하여 영상의 메모리 영역을 확보한다.

Overrides:

[setVolume](#) in class [Cube](#)

---

### clone

```
public java.lang.Object clone()
```

현재의 헤더 정보를 복사한다.

Overrides:

[clone](#) in class [Cube](#)

---

## setGeoCoord

```
public void setGeoCoord(double xmin,  
                        double xmax,  
                        double ymin,  
                        double ymax)
```

입력 범위에 대하여 영상의 x, y 방향에 대한 최소, 최대값을 설정한다.

---

## setGeoCoord

```
public void setGeoCoord(GeoRectangle r)
```

입력 범위에 대하여 영상의 x, y 방향에 대한 최소, 최대값을 설정한다.

---

## getGeoCoord

```
public GeoRectangle getGeoCoord()
```

영상의 x, y 방향에 대한 최소, 최대값을 갖는 영역을 되돌린다.

---

## getScaledGeoCoord

```
public GeoRectangle getScaledGeoCoord(java.awt.Rectangle r)
```

입력 범위에 대하여 새로운 좌표값을 계산하여 되돌린다.

---

## setMinMax

```
public void setMinMax(int band,  
                      float min,  
                      float max)
```

대상 영상의 분광대의 최소값과 최대값을 정의한다

---

## setBandName

```
public void setBandName(int index,  
                        java.lang.String name)
```

대상 영상의 주어진 분광대의 이름을 정의한다

---

## getBandName

```
public java.lang.String getBandName(int index)
```

영상의 분광대 목록에서 주어진 인덱스의 분광대 이름을 되돌린다.

---

## getBandNameForList

```
public java.lang.String getBandNameForList(int index)
```

영상의 목록에서 주어진 인덱스의 분광대 이름을 되돌린다.

---

## read

```
public void read(java.lang.String path)
    throws java.io.IOException,
           IllegalFormatException
```

영상의 헤더 정보를 읽는다.

Throws:

java.io.IOException - IOException  
IllegalFormatException - IllegalFormatException

---

## read

```
public void read(java.io.File f)
    throws java.io.IOException,
           IllegalFormatException
```

영상의 헤더 정보를 읽는다.

Throws:

java.io.IOException - IOException  
IllegalFormatException - IllegalFormatException

---

## write

```
public void write(java.io.File f)
    throws java.io.IOException
```

영상의 헤더 정보를 파일로 저장한다.

Throws:

java.io.IOException - IOException

---

## write

```
public void write(java.lang.String path)
    throws java.io.IOException,
           java.util.NoSuchElementException
```

영상의 헤더 정보를 파일로 저장한다.

Throws:

java.io.IOException – IOException

java.util.NoSuchElementException – NoSuchElementException

---

## toString

```
public java.lang.String toString()
```

영상의 헤더 정보를 문자열로 출력한다.

Overrides:

toString in class java.lang.Object

---

## isEmpty

```
public boolean isEmpty()
```

영상의 헤더 정보가 정의 되어 있는지 판단한다. 헤더중의 Samples, Lines, Bands 중 하나라도 정의되어 있지 않으면 false를 그렇지 않은 경우 true를 되돌린다.

---

## insideRegion

```
public boolean insideRegion(int smp,
                             int ln,
                             int bnd)
```

입력된 화소 위치가 영상의 유효범위에 있는지를 판단한다. 유효범위내에 있을 경우 true, 아닐 경우 false를 되돌린다.

---

## assert

```
public void assert()
    throws AssertionError
```

영상의 헤더 정보가 정확하게 설정되어 있는지를 확인한다.

Throws:

AssertionError – AssertionError

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class CubeMemoryToFileOp

```
java.lang.Object
|
+--CubeMemoryToFileOp
```

```
public class CubeMemoryToFileOp
extends java.lang.Object
implements Operator
```

메모리 타입의 영상을 파일 타입으로 변환한다.

## Constructor Summary

<code>CubeMemoryToFileOp()</code> Constructor
--

## Method Summary

void	<code>operate()</code> 영상을 파일 타입으로 변환한다.
void	<code>setBands(int[] bands)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target.Memory srcTgt, Target.File destTgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### CubeMemoryToFileOp

```
public CubeMemoryToFileOp()
```

Constructor

## Method Detail

### setTarget

```
public void setTarget(Target.Memory srcTgt,  
                    Target.File destTgt)
```

입출력 영상을 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상을 파일 타입으로 변환한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedExcep<sup>tion</sup> – InterruptedExcep<sup>tion</sup>

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---



## Class CubeSubsetOp

java.lang.Object  
|  
+--CubeSubsetOp

public class CubeSubsetOp  
extends java.lang.Object  
implements [Operator](#)

부분 영상을 추출한다.

Constructor Summary	
<a href="#">CubeSubsetOp()</a>	Constructor

Method Summary	
void	<a href="#">operate()</a> 부분 영상을 추출한다.
void	<a href="#">setBands(int[] bands)</a> 입력 영상의 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리 영역을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### CubeSubsetOp

public CubeSubsetOp()

Constructor

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedExcepion
```

부분 영상을 추출한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedExcepion – InterruptedExcepion

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class DisplayEnhanceOp

```
java.lang.Object
|
+--DisplayEnhanceOp
```

```
public class DisplayEnhanceOp
extends java.lang.Object
```

화면에 대한 영상향상을 실행한다. 결과는 바이트이다.

Constructor Summary	
<code>DisplayEnhanceOp(DisplayEnhanceMethod method)</code>	Constructor

Method Summary	
<code>byte[] operate(byte[] src, byte[] dest)</code>	바이트형 영상에 대한 향상을 한다.
<code>byte[] operate(byte[] src, byte[] dest, int min, int max)</code>	바이트형 영상에 대한 향상을 한다.
<code>byte[] operate(float[] src, byte[] dest)</code>	4바이트 실수형 영상에 대한 향상을 한다.
<code>byte[] operate(float[] src, byte[] dest, float min, float max)</code>	4바이트 정수형 영상에 대한 향상을 한다.
<code>byte[] operate(int[] src, byte[] dest)</code>	4바이트 정수형 영상에 대한 향상을 한다.
<code>byte[] operate(int[] src, byte[] dest, int min, int max)</code>	4바이트 정수형 영상에 대한 향상을 한다.
<code>byte[] operate(short[] src, byte[] dest, boolean signed)</code>	2바이트 정수형 영상에 대한 향상을 한다.
<code>byte[] operate(short[] src, byte[] dest, boolean signed, int min, int max)</code>	2바이트 정수형 영상에 대한 향상을 한다.

Methods inherited from class java.lang.Object
<code>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

## Constructor Detail

### DisplayEnhanceOp

```
public DisplayEnhanceOp(DisplayEnhanceMethod method)
```

## Method Detail

### operate

```
public byte[] operate(byte[] src,  
                      byte[] dest)
```

바이트형 영상에 대한 향상을 한다.

---

### operate

```
public byte[] operate(short[] src,  
                      byte[] dest,  
                      boolean signed)
```

2바이트 정수형 영상에 대한 향상을 한다.

---

### operate

```
public byte[] operate(int[] src,  
                      byte[] dest)
```

4바이트 정수형 영상에 대한 향상을 한다.

---

### operate

```
public byte[] operate(float[] src,  
                      byte[] dest)
```

4바이트 실수형 영상에 대한 향상을 한다.

---

### operate

```
public byte[] operate(byte[] src,  
                      byte[] dest,  
                      int min,  
                      int max)
```

바이트형 영상에 대한 향상을 한다.

---

### operate

```
public byte[] operate(short[] src,  
                      byte[] dest,  
                      boolean signed,  
                      int min,  
                      int max)
```

2바이트 정수형 영상에 대한 향상을 한다.

---

## operate

```
public byte[] operate(int[] src,
                     byte[] dest,
                     int min,
                     int max)
```

4바이트 정수형 영상에 대한 향상을 한다.

---

## operate

```
public byte[] operate(float[] src,
                     byte[] dest,
                     float min,
                     float max)
```

4바이트 정수형 영상에 대한 향상을 한다.

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

## Class DNToRadianceOp

```
java.lang.Object
|
+---DNToRadianceOp
```

```
public class DNToRadianceOp
extends java.lang.Object
implements Operator
```

영상의 회소값을 휘도값으로 변환한다. 출력결과는 4바이트 실수형이다.

Field Summary	
static int	<u>NONE</u>

Constructor Summary	
<u>DNToRadianceOp()</u> Constructor	

Method Summary	
protected void	<u>changePixelValue</u> (double[] a, int idx, int aOffset, float[] b, int bOffset, int length)
void	<u>operate</u> () 회소값의 휘도로의 변환을 한다.
void	<u>setBands</u> (int[] bands) 입력 영상의 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setSatellite</u> (java.lang.String satellite) 처리 대상영상을 획득한 위성의 종류를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

## Field Detail

NONE

```
public static final int NONE
```

## Constructor Detail

DNToRadianceOp

```
public DNToRadianceOp()
```

Constructor

## Method Detail

setSatellite

```
public void setSatellite(java.lang.String satellite)
```

처리 대상영상을 획득한 위성의 종류를 설정한다.

---

setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

화소값의 휘도로의 변환을 한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                int idx,  
                                int aOffset,  
                                float[] b,  
                                int bOffset,  
                                int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class DNToReflectanceOp

java.lang.Object

|--DNToReflectanceOp

---

```
public class DNToReflectanceOp
extends java.lang.Object
implements Operator
```

영상의 회소값을 센서의 반사도값으로 변환한다. 출력결과는 4바이트 실수형이다.

---

## Field Summary

static int	<a href="#">NONE</a>
------------	----------------------

## Constructor Summary

<a href="#">DNToReflectanceOp()</a> Constructor	
--	--

## Method Summary

protected void	<code>changePixelValue(double[] a, int idx, int aOffset, float[] b, int bOffset, int length)</code>
void	<code>operate()</code> 화소값의 반사도값으로의 변환을 한다.
void	<code>setBands(int[] bands)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle b)</code> 처리 영역을 설정한다.
void	<code>setEarthSunDistance(float d)</code> 태양과 지구와의 거리비 값을 설정한다.
void	<code>setParameter(float[] p)</code> 태양 변수를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setSatellite(java.lang.String satellite)</code> 처리 대상영상을 획득한 위성의 종류를 설정한다.
void	<code>setSolarZenith(float z)</code> 태양 천정각을 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

public static final int NONE

## Constructor Detail

DNToReflectanceOp

public DNToReflectanceOp()

Constructor

## Method Detail

setSatellite

public void setSatellite(java.lang.String satellite)

처리 대상영상을 획득한 위성의 종류를 설정한다.

## setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

## setParameter

```
public void setParameter(float[] p)
```

태양 변수를 설정한다.

---

## setSolarZenith

```
public void setSolarZenith(float z)
```

태양 천정각을 설정한다.

---

## setEarthSunDistance

```
public void setEarthSunDistance(float d)
```

태양과 지구와의 거리비 값을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

화소값의 반사도값으로의 변환을 한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                int idx,  
                                int aOffset,  
                                float[] b,  
                                int bOffset,  
                                int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class DXFtoVectorOp.FeatureProtocol

java.lang.Object

|--DXFtoVectorOp.FeatureProtocol

---

```
public class DXFtoVectorOp.FeatureProtocol
extends java.lang.Object
```

Feature를 관리한다.

---

Field Summary	
int	<u>ID</u>
int	<u>Type</u>
int	<u>VertexNo</u>
double[]	<u>X</u>
double[]	<u>Y</u>

Constructor Summary	
<u>DXFtoVectorOp.FeatureProtocol()</u>	
Constructor	

## Method Summary

void	<code>addVertex(double Xvertex, double Yvertex)</code> Vertex를 추가한다.
void	<code>clear()</code> 벡터정보를 제거한다.
double	<code>getXmax()</code> X좌표의 최대값을 되돌린다.
double	<code>getXmin()</code> X좌표의 최소값을 되돌린다.
double	<code>getYmax()</code> Y좌표의 최대값을 되돌린다.
double	<code>getYmin()</code> Y좌표의 최소값을 되돌린다.
void	<code>print()</code> 벡터 정보(Vertex)를 화면에 출력한다.
void	<code>write(GeoDataOutputStream out)</code> 벡터 정보(Vertex)를 파일로 출력한다.

## Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Field Detail

### Type

public int Type

---

### ID

public int ID

---

### X

public double[] X

---

### Y

public double[] Y

---

### VertexNo

public int VertexNo

## Constructor Detail

### DXFtoVectorOp.FeatureProtocol

```
public DXFtoVectorOp.FeatureProtocol()
```

Constructor

## Method Detail

### getXmin

```
public double getXmin()
```

X좌표의 최소값을 되돌린다.

---

### getYmin

```
public double getYmin()
```

Y좌표의 최소값을 되돌린다.

---

### getXmax

```
public double getXmax()
```

X좌표의 최대값을 되돌린다.

---

### getYmax

```
public double getYmax()
```

Y좌표의 최대값을 되돌린다.

---

### addVertex

```
public void addVertex(double Xvertex,  
                     double Yvertex)
```

Vertex를 추가한다.

---

### clear

```
public void clear()
```

벡터정보를 제거한다.

---

## print

```
public void print()
```

백터 정보(Vertex)를 화면에 출력한다.

---

## write

```
public void write(GeoDataOutputStream out)
    throws java.io.IOException
```

백터 정보(Vertex)를 파일로 출력한다.

Throws:

java.io.IOException – IOException

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class DXFtoVectorOp

```
java.lang.Object
|
+--DXFtoVectorOp
```

```
public class DXFtoVectorOp
extends java.lang.Object
implements Operator
```

DXF 포맷으로 부터 내부 벡터자료(GeoVector)를 구성한다.

Inner Class Summary	
class	<a href="#">DXFtoVectorOp.FeatureProtocol</a> Feature를 관리한다.
class	<a href="#">DXFtoVectorOp.VHeader</a> 벡터의 헤더정보를 관리한다.

Field Summary	
static int	<a href="#">COLOR</a>
static int	<a href="#">LAYER_ORDER</a>
static int	<a href="#">USER_DEFINED</a>
static int	<a href="#">ZVALUE</a>

Constructor Summary	
<a href="#">DXFtoVectorOp()</a>	Constructor

## Method Summary

java.lang.String	<u>getTitle()</u> Title을 되돌린다.
void	<u>operate()</u> 내부 벡터로의 변환을 실행한다.
java.util.Vector	<u>readLayername()</u> Layer에 대한 벡터를 되돌린다.
void	<u>setDXF(java.io.File dxf)</u> DXF 파일을 설정한다.
void	<u>setFeatureIDMode(int field, int value)</u> Feature의 ID를 세팅한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setSelectedFeatureType(int type)</u> Dialog에서 선택된 Feature들의 type을 세팅한다.
void	<u>setSelectedLayer(java.lang.String[] layer)</u> Dialog에서 선택한 Layer를 설정한다.
void	<u>setTitle(java.lang.String s)</u> Title을 설정한다.
void	<u>setVector(java.io.File vec)</u> 출력 Vector 파일을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### LAYER\_ORDER

```
public static final int LAYER_ORDER
```

---

### ZVALUE

```
public static final int ZVALUE
```

---

### COLOR

```
public static final int COLOR
```

---

### USER\_DEFINED

```
public static final int USER_DEFINED
```

## Constructor Detail

### DXFtoVectorOp

```
public DXFtoVectorOp()
```

Constructor

## Method Detail

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### getTitle

```
public java.lang.String getTitle()
```

Title을 되돌린다.

---

### setTitle

```
public void setTitle(java.lang.String s)
```

Title을 설정한다.

---

### setSelectedFeatureType

```
public void setSelectedFeatureType(int type)
```

Dialog에서 선택된 Feature들의 type을 세팅한다.

---

### setFeatureIDMode

```
public void setFeatureIDMode(int field,  
                             int value)
```

Feature의 ID를 세팅한다.

---

### setSelectedLayer

```
public void setSelectedLayer(java.lang.String[] layer)
```

Dialog에서 선택한 Layer를 설정한다.

---

### setDXF

```
public void setDXF(java.io.File dxf)
```

DXF 파일을 설정한다.

---

## setVector

```
public void setVector(java.io.File vec)
```

출력 Vector 파일을 설정한다.

---

## readLayername

```
public java.util.Vector readLayername()  
    throws java.io.IOException
```

Layer에 대한 벡터를 되돌린다.

Throws:

java.io.IOException – IOException

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedExcepion
```

내부 벡터로의 변환을 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedExcepion – InterruptedExcepion

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class DXFtoVectorOp.VHeader

```
java.lang.Object
|
+--DXFtoVectorOp.VHeader
```

---

```
public class DXFtoVectorOp.VHeader
extends java.lang.Object
```

벡터의 헤더정보를 관리한다.

---

Field Summary	
double	<a href="#">ExtXmax</a>
double	<a href="#">ExtXmin</a>
double	<a href="#">ExtYmax</a>
double	<a href="#">ExtYmin</a>
int	<a href="#">FeatureCount</a>
double	<a href="#">LimitXmax</a>
double	<a href="#">LimitXmin</a>
double	<a href="#">LimitYmax</a>
double	<a href="#">LimitYmin</a>
java.lang.String	<a href="#">Title</a>
double	<a href="#">Xmax</a>
double	<a href="#">Xmin</a>
double	<a href="#">Ymax</a>
double	<a href="#">Ymin</a>

## Constructor Summary

`DXFtoVectorOp.VHeader()`

## Method Summary

`void print()`  
헤더정보를 화면에 출력한다.

`void write(GeoDataOutputStream out)`  
헤더 정보중(Title, Xmin, Xmax, Ymin, Ymax, FeatureCount)를 파일로 출력한다.

`void write(java.io.RandomAccessFile rout)`  
헤더 정보중(Title, Xmin, Xmax, Ymin, Ymax, FeatureCount)를 파일로 출력한다.

## Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Field Detail

### Title

`public java.lang.String Title`

---

### ExtXmin

`public double ExtXmin`

---

### ExtYmin

`public double ExtYmin`

---

### ExtXmax

`public double ExtXmax`

---

### ExtYmax

`public double ExtYmax`

---

### LimitXmin

`public double LimitXmin`

---

## LimitYmin

public double LimitYmin

---

## LimitXmax

public double LimitXmax

---

## LimitYmax

public double LimitYmax

---

## Xmin

public double Xmin

---

## Ymin

public double Ymin

---

## Xmax

public double Xmax

---

## Ymax

public double Ymax

---

## FeatureCount

public int FeatureCount

## Constructor Detail

### DXFtoVectorOp.VHeader

public DXFtoVectorOp.VHeader()

## Method Detail

print

```
public void print()
```

헤더정보를 화면에 출력한다.

---

## write

```
public void write(GeoDataOutputStream out)
    throws java.io.IOException
```

헤더 정보중(Title, Xmin, Xmax, Ymin, Ymax, FeatureCount)를 파일로 출력한다.

Throws:

java.io.IOException - IOException

---

## write

```
public void write(java.io.RandomAccessFile rout)
    throws java.io.IOException
```

헤더 정보중(Title, Xmin, Xmax, Ymin, Ymax, FeatureCount)를 파일로 출력한다.

Throws:

java.io.IOException - IOException

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

---



## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class DXFVertexExtractOp

java.lang.Object

|--DXFVertexExtractOp

---

```
public class DXFVertexExtractOp
extends java.lang.Object
implements Operator
```

DXF파일에서 Vertex를 추출한다. 각 Vertex는 X좌표 Y좌표 Z좌표로 기록되며 각 값은 스페이스로 구분된다. 파일의 끝에는 -999.0 -999.0 -999.0이라는 값이 기록된다.

---

### Field Summary

static int	<a href="#">COLOR</a>
static int	<a href="#">LAYER_ORDER</a>
static int	<a href="#">USER_DEFINED</a>
static int	<a href="#">ZVALUE</a>

### Constructor Summary

<a href="#">DXFVertexExtractOp()</a> Constructor	
---	--

## Method Summary

void	<code>operate()</code> Vertex를 추출한다.
java.util.Vector	<code>readLayername()</code> Layer에 대한 벡터를 되돌린다.
void	<code>setDXF(java.io.File dxf)</code> DXF 파일을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setSelectedLayer(java.lang.String[] layer)</code> Dialog에서 선택한 Layer를 설정한다.
void	<code>setXYZFile(java.io.File xyz)</code> 출력 ASCII 파일을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### LAYER\_ORDER

```
public static final int LAYER_ORDER
```

---

### ZVALUE

```
public static final int ZVALUE
```

---

### COLOR

```
public static final int COLOR
```

---

### USER\_DEFINED

```
public static final int USER_DEFINED
```

## Constructor Detail

### DXFVertexExtractOp

```
public DXFVertexExtractOp()
```

Constructor

## Method Detail

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setSelectedLayer

```
public void setSelectedLayer(java.lang.String[] layer)
```

Dialog에서 선택한 Layer를 설정한다.

---

### setDXF

```
public void setDXF(java.io.File dxf)
```

DXF 파일을 설정한다.

---

### setXYZFile

```
public void setXYZFile(java.io.File xyz)
```

출력 ASCII 파일을 설정한다.

---

### readLayername

```
public java.util.Vector readLayername()  
                           throws java.io.IOException
```

Layer에 대한 벡터를 되돌린다.

Throws:

java.io.IOException – IOException

---

### operate

```
public void operate()  
           throws java.io.IOException,  
                  java.lang.InterruptedException
```

Vertex를 추출한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

### stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class EmbossingKernel

java.lang.Object  
|  
+--EmbossingKernel

public class EmbossingKernel  
extends java.lang.Object

Embossing 필터 커널을 관리한다.

Constructor Summary	
<a href="#">EmbossingKernel</a> (java.lang.String method)	Constructor

Method Summary	
int	<a href="#">getDivisor</a> () 콘볼루션에 대한 가중치를 되돌린다.
int[]	<a href="#">getDownKernel</a> () Down 방향에 대한 필터배열을 되돌린다.
int[]	<a href="#">getKernel</a> () 필터 배열을 되돌린다.
int[]	<a href="#">getLeftKernel</a> () Left 방향에 대한 필터배열을 되돌린다.
int[]	<a href="#">getRightKernel</a> () Right 방향에 대한 필터배열을 되돌린다.
int[]	<a href="#">getUpKernel</a> () Up 방향에 대한 필터배열을 되돌린다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### EmbossingKernel

public EmbossingKernel(java.lang.String method)

Constructor

## Method Detail

### getKernel

```
public int[] getKernel()
```

휠터 배열을 되돌린다.

---

### getUpKernel

```
public int[] getUpKernel()
```

Up 방향에 대한 휠터배열을 되돌린다.

---

### getDownKernel

```
public int[] getDownKernel()
```

Down 방향에 대한 휠터배열을 되돌린다.

---

### getLeftKernel

```
public int[] getLeftKernel()
```

Left 방향에 대한 휠터배열을 되돌린다.

---

### getRightKernel

```
public int[] getRightKernel()
```

Right 방향에 대한 휠터배열을 되돌린다.

---

### getDivisor

```
public int getDivisor()
```

콘블루션에 대한 가중치를 되돌린다.

---

[Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY: INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class EnhanceGraph

```
java.lang.Object
|
+--java.awt.Component
    |
    +--java.awt.Canvas
        |
        +--EnhanceGraph
```

---

```
public class EnhanceGraph
extends java.awt.Canvas
```

대화형 영상향상을 위한 캔버스를 관리한다.

See Also:  
[Serialized Form](#)

---

### Fields inherited from class java.awt.Component

[BOTTOM\\_ALIGNMENT](#), [CENTER\\_ALIGNMENT](#), [LEFT\\_ALIGNMENT](#), [RIGHT\\_ALIGNMENT](#), [TOP\\_ALIGNMENT](#)

### Constructor Summary

<a href="#">EnhanceGraph()</a> Constructor
<a href="#">EnhanceGraph(java.awt.Color col)</a> Constructor
<a href="#">EnhanceGraph(java.awt.Color col, com.objectspace.jgl.DList vList)</a> Constructor

Method Summary	
void	<u>drawBackGround()</u> 배경 히스토그램을 그린다.
void	<u>drawGrid(boolean b)</u> 캔버스에 50화소 간격으로 격자를 그린다.
java.awt.Point	<u>getSelectedVertex()</u> 선택된 점의 좌표를 되돌린다.
com.objectspace.jgl.DList	<u>getVList()</u> 작성된 그래프의 좌표를 되돌린다.
static java.awt.Point	<u>getXY()</u> 마우스의 위치를 되돌린다.
void	<u>initBackGround()</u> 배경을 초기화 한다.
void	<u>inverseVList()</u> 작성된 그래프의 좌표를 반전시킨다.
void	<u>inverseVList(com.objectspace.jgl.DList dList)</u>
void	<u>paint(java.awt.Graphics g)</u> 캔버스에 그래프를 출력한다.
void	<u>setBackground()</u> 배경을 설정한다.
void	<u>setBackground(int[] histo)</u> 배경 히스토그램을 설정한다.
void	<u>setColor(java.awt.Color col)</u> 그래프의 색을 설정한다.
void	<u>setListener()</u>
void	<u>setVList(com.objectspace.jgl.DList vList)</u> 변환 좌표를 설정 또는 초기화한다.
void	<u>update(java.awt.Graphics g)</u>

Methods inherited from class java.awt.Canvas
addNotify



## Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getName, getParent, getPeer, getPreferredSize, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, paramString, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, processComponentEvent, processEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFont, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus, validate

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### EnhanceGraph

```
public EnhanceGraph()
```

Constructor

---

### EnhanceGraph

```
public EnhanceGraph(java.awt.Color col)
```

Constructor

---

### EnhanceGraph

```
public EnhanceGraph(java.awt.Color col,  
                    com.objectspace.jgl.DList vList)
```

Constructor

## Method Detail

### setColor

```
public void setColor(java.awt.Color col)
```

그래프의 색을 설정한다.

---

## setListener

```
public void setListener()
```

---

## setVList

```
public void setVList(com.objectspace.jgl.DList vList)
```

변환 좌표를 설정 또는 초기화한다.

---

## getVList

```
public com.objectspace.jgl.DList getVList()
```

작성된 그래프의 좌표를 되돌린다.

---

## getSelectedVertex

```
public java.awt.Point getSelectedVertex()
```

선택된 점의 좌표를 되돌린다.

---

## inverseVList

```
public void inverseVList(com.objectspace.jgl.DList dList)
```

---

## inverseVList

```
public void inverseVList()
```

작성된 그래프의 좌표를 반전시킨다.

---

## getXY

```
public static java.awt.Point getXY()
```

마우스의 위치를 되돌린다.

---

## update

```
public void update(java.awt.Graphics g)
```

Overrides:  
update in class java.awt.Component

---

## paint

```
public void paint(java.awt.Graphics g)
```

캔버스에 그래프를 출력한다.

Overrides:  
paint in class java.awt.Canvas

---

## initBackGround

```
public void initBackGround()
```

배경을 초기화 한다.

---

## setBackground

```
public void setBackground()
```

배경을 설정한다.

---

## setBackground

```
public void setBackground(int[] histo)
```

배경 히스토그램을 설정한다.

---

## drawBackGround

```
public void drawBackGround()
```

배경 히스토그램을 그린다.

---

## drawGrid

```
public void drawGrid(boolean b)
```

캔버스에 50화소 간격으로 격자를 그린다.

---

# Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Feature

java.lang.Object  
|  
+--Feature

Direct Known Subclasses:  
Feature.MultiVertex, [Feature.Point](#)

---

public abstract class Feature  
extends java.lang.Object

Feature를 관리한다.

---

Inner Class Summary	
static class	<a href="#">Feature.Line</a>
static class	<a href="#">Feature.Point</a> Point feature를 관리한다.
static class	<a href="#">Feature.Polygon</a>

Field Summary	
protected int	<a href="#">ID</a>
static int	<a href="#">LINE</a>
static int	<a href="#">POINT</a>
static int	<a href="#">POLYGON</a>

Constructor Summary	
<a href="#">Feature()</a>	

## Method Summary

abstract void	<u>addVertex</u> (double x, double y)
abstract void	<u>addVertex</u> (Feature.Point p)
abstract int	<u>count</u> ()
abstract java.util.Enumeration	<u>elements</u> ()
int	<u>getID</u> () ID를 되돌린다.
abstract Feature.Point	<u>getMax</u> ()
abstract Feature.Point	<u>getMin</u> ()
abstract int	<u>getType</u> ()
abstract Feature.Point	<u>getVertex</u> (int index)
abstract Feature.Point[]	<u>getVertices</u> ()
abstract int	<u>indexOf</u> (Feature.Point p)
abstract int	<u>indexOf</u> (int x, int y)
abstract void	<u>insertVertex</u> (int index, double x, double y)
abstract void	<u>insertVertex</u> (int index, Feature.Point p)
abstract void	<u>removeVertex</u> (double x, double y)
abstract void	<u>removeVertex</u> (Feature.Point p)
abstract void	<u>removeVertex</u> (int index)
void	<u>setID</u> (int id) ID를 설정한다.
abstract void	<u>setMax</u> (double x, double y)
abstract void	<u>setMin</u> (double x, double y)

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

## POINT

```
public static final int POINT
```

---

## LINE

```
public static final int LINE
```

---

## POLYGON

```
public static final int POLYGON
```

---

## ID

```
protected int ID
```

### Constructor Detail

#### Feature

```
public Feature()
```

### Method Detail

#### getType

```
public abstract int getType()
```

---

#### getID

```
public int getID()
```

ID를 되돌린다.

---

#### setID

```
public void setID(int id)
```

ID를 설정한다.

---

#### getMin

```
public abstract Feature.Point getMin()
```

---

## getMax

```
public abstract Feature.Point getMax()
```

---

## setMin

```
public abstract void setMin(double x,  
                             double y)
```

---

## setMax

```
public abstract void setMax(double x,  
                             double y)
```

---

## addVertex

```
public abstract void addVertex(double x,  
                                double y)
```

---

## addVertex

```
public abstract void addVertex(Feature.Point p)
```

---

## insertVertex

```
public abstract void insertVertex(int index,  
                                   double x,  
                                   double y)
```

---

## insertVertex

```
public abstract void insertVertex(int index,  
                                   Feature.Point p)
```

---

## removeVertex

```
public abstract void removeVertex(int index)
```

---

## removeVertex

```
public abstract void removeVertex(double x,  
                                   double y)
```

---

## removeVertex

public abstract void `removeVertex`([Feature.Point](#) p)

---

## elements

public abstract [java.util.Enumeration](#) `elements`()

---

## getVertices

public abstract [Feature.Point](#)[] `getVertices`()

---

## getVertex

public abstract [Feature.Point](#) `getVertex`(int index)

---

## indexOf

public abstract int `indexOf`(int x,  
int y)

---

## indexOf

public abstract int `indexOf`([Feature.Point](#) p)

---

## count

public abstract int `count`()

---

---

### [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

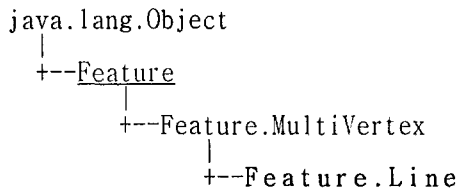
[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class Feature.Line



public static class Feature.Line  
 extends Feature.MultiVertex

### Inner classes inherited from class Feature

[Feature.Line](#), [Feature.Point](#), [Feature.Polygon](#)

### Fields inherited from class Feature.MultiVertex

Max, Min, validMax, validMin, Vertex

### Fields inherited from class Feature

[ID](#), [LINE](#), [POINT](#), [POLYGON](#)

## Constructor Summary

[Feature.Line\(\)](#)

## Method Summary

	int	<a href="#">getType()</a>
	java.lang.String	<a href="#">toString()</a>

### Methods inherited from class Feature.MultiVertex

[addVertex](#), [addVertex](#), [calculateMinMax](#), [count](#), [elements](#), [equals](#), [getMax](#), [getMin](#), [getVertex](#), [getVertexs](#), [indexOf](#), [indexOf](#), [insertVertex](#), [insertVertex](#), [removeVertex](#), [removeVertex](#), [removeVertex](#), [setMax](#), [setMin](#)

### Methods inherited from class Feature

[getID](#), [setID](#)

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

## Constructor Detail

### Feature.Line

```
public Feature.Line()
```

## Method Detail

### getType

```
public int getType()
```

Overrides:

[getType](#) in class [Feature](#)

---

### toString

```
public java.lang.String toString()
```

Overrides:

[toString](#) in class [Feature.MultiVertex](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Feature.Point

java.lang.Object

+--Feature

+--Feature.Point

---

public static class Feature.Point  
extends Feature

Point feature를 관리한다.

---

### Inner classes inherited from class Feature

Feature.Line, Feature.Point, Feature.Polygon

### Field Summary

	double	x
	double	y

### Fields inherited from class Feature

ID, LINE, POINT, POLYGON

### Constructor Summary

Feature.Point()  
Constructor

Feature.Point(double x, double y)  
Point를 설정한다.

Feature.Point(Feature.Point p)  
Point를 설정한다.

Method Summary	
void	<code>addVertex(double x, double y)</code>
void	<code>addVertex(<u>Feature.Point</u> p)</code>
int	<code>count()</code>
java.util.Enumeration	<code>elements()</code>
boolean	<code>equals(<u>Feature.Point</u> p)</code> 동일한 객체인지를 검사한다.
<u>Feature.Point</u>	<code>getMax()</code> 포인트의 최대 좌표값을 되돌린다.
<u>Feature.Point</u>	<code>getMin()</code> 포인트의 최소 좌표값을 되돌린다.
int	<code>getType()</code> Feature의 타입을 되돌린다.
<u>Feature.Point</u>	<code>getVertex(int index)</code>
<u>Feature.Point</u> []	<code>getVertexs()</code>
int	<code>indexOf(<u>Feature.Point</u> p)</code>
int	<code>indexOf(int x, int y)</code>
void	<code>insertVertex(int index, double x, double y)</code>
void	<code>insertVertex(int index, <u>Feature.Point</u> p)</code>
void	<code>removeVertex(double x, double y)</code>
void	<code>removeVertex(<u>Feature.Point</u> p)</code>
void	<code>removeVertex(int index)</code>
void	<code>setMax(double x, double y)</code>
void	<code>setMin(double x, double y)</code>
java.lang.String	<code>toString()</code>

#### Methods inherited from class Feature

`getID`, `setID`

#### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

**x**

```
public double x
```

---

**y**

```
public double y
```

## Constructor Detail

### Feature.Point

```
public Feature.Point()
```

Constructor

---

### Feature.Point

```
public Feature.Point(Feature.Point p)
```

Point를 설정한다.

---

### Feature.Point

```
public Feature.Point(double x,  
                     double y)
```

Point를 설정한다.

## Method Detail

### equals

```
public boolean equals(Feature.Point p)
```

동일한 객체인지를 검사한다.

---

### getType

```
public int getType()
```

Feature의 타입을 되돌린다.

Overrides:

getType in class Feature

---

## getMin

```
public Feature.Point getMin()
```

포인트의 최소 좌표값을 되돌린다.

Overrides:

getMin in class Feature

---

## getMax

```
public Feature.Point getMax()
```

포인트의 최대 좌표값을 되돌린다.

Overrides:

getMax in class Feature

---

## setMin

```
public void setMin(double x,  
                  double y)
```

Overrides:

setMin in class Feature

---

## setMax

```
public void setMax(double x,  
                  double y)
```

Overrides:

setMax in class Feature

---

## addVertex

```
public void addVertex(double x,  
                     double y)
```

Overrides:

addVertex in class Feature

---

## addVertex

```
public void addVertex(Feature.Point p)
```

Overrides:

addVertex in class Feature

---

## insertVertex

```
public void insertVertex(int index,  
                        double x,  
                        double y)
```

Overrides:  
[insertVertex](#) in class [Feature](#)

---

## insertVertex

```
public void insertVertex(int index,  
                        Feature.Point p)
```

Overrides:  
[insertVertex](#) in class [Feature](#)

---

## removeVertex

```
public void removeVertex(int index)
```

Overrides:  
[removeVertex](#) in class [Feature](#)

---

## removeVertex

```
public void removeVertex(double x,  
                        double y)
```

Overrides:  
[removeVertex](#) in class [Feature](#)

---

## removeVertex

```
public void removeVertex(Feature.Point p)
```

Overrides:  
[removeVertex](#) in class [Feature](#)

---

## elements

```
public java.util.Enumeration elements()
```

Overrides:  
[elements](#) in class [Feature](#)

---

## getVertexs

```
public Feature.Point[] getVertexs()
```

Overrides:  
[getVertexs](#) in class [Feature](#)

---

## getVertex

```
public Feature.Point getVertex(int index)
```

Overrides:  
[getVertex](#) in class [Feature](#)

---

## indexOf

```
public int indexOf(int x,  
                  int y)
```

Overrides:  
[indexOf](#) in class [Feature](#)

---

## indexOf

```
public int indexOf(Feature.Point p)
```

Overrides:  
[indexOf](#) in class [Feature](#)

---

## count

```
public int count()
```

Overrides:  
[count](#) in class [Feature](#)

---

## toString

```
public java.lang.String toString()
```

Overrides:  
[toString](#) in class [java.lang.Object](#)

---

[Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)



## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Feature.Polygon

```
java.lang.Object
|
+--Feature
    |
    +--Feature.MultiVertex
        |
        +--Feature.Polygon
```

---

```
public static class Feature.Polygon
extends Feature.MultiVertex
```

---

### Inner classes inherited from class [Feature](#)

[Feature.Line](#), [Feature.Point](#), [Feature.Polygon](#)

### Fields inherited from class [Feature.MultiVertex](#)

[Max](#), [Min](#), [validMax](#), [validMin](#), [Vertex](#)

### Fields inherited from class [Feature](#)

[ID](#), [LINE](#), [POINT](#), [POLYGON](#)

## Constructor Summary

[Feature.Polygon\(\)](#)

## Method Summary

	int	<a href="#">getType()</a>
	java.lang.String	<a href="#">toString()</a>

### Methods inherited from class [Feature.MultiVertex](#)

[addVertex](#), [addVertex](#), [calculateMinMax](#), [count](#), [elements](#), [equals](#), [getMax](#), [getMin](#), [getVertex](#), [getVertexs](#), [indexOf](#), [indexOf](#), [insertVertex](#), [insertVertex](#), [removeVertex](#), [removeVertex](#), [removeVertex](#), [setMax](#), [setMin](#)

### Methods inherited from class [Feature](#)

[getID](#), [setID](#)

### Methods inherited from class [java.lang.Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

## Constructor Detail

### Feature.Polygon

```
public Feature.Polygon()
```

## Method Detail

### getType

```
public int getType()
```

Overrides:

[getType](#) in class [Feature](#)

---

### toString

```
public java.lang.String toString()
```

Overrides:

[toString](#) in class [Feature.MultiVertex](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class FFTOp

```
java.lang.Object
|
+--FFTOp
```

---

```
public class FFTOp
extends java.lang.Object
implements Operator
```

영상의 fast fourier transform을 수행한다. forward 변환일 경우 대상영상은 메모리형이어야 하고 출력 결과는 real, imagery가 제1분광대, 제2분광대로 저장된다. 역방향인 경우 출력 결과는 단일 밴드의 바이트형이다.

---

## Constructor Summary

<a href="#">FFTOp()</a> Constructor	
--	--

## Method Summary

protected void	<u>ButterworthBPF</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>ButterworthBS</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>ButterworthHP</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>ButterworthLP</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>FFT1D</u> (float[] ra, float[] ia, float[] st, float[] ct, int lenth)
protected void	<u>FFT2D</u> (float[][] oPixel, int lns, int smps, int lenth)
protected void	<u>IdealBPF</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>IdealBS</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>IdealHP</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>IdealLP</u> ilteredFrequency(float[][] pix, int length)
protected void	<u>NoFiltered</u> Frequency(float[][] pix, int length)
void	<u>operate</u> () 변환을 실행한다.
void	<u>setBand</u> (int b) 처리 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<u>setFilterOrder</u> (float f) 필터에 사용할 order를 설정한다.
void	<u>setFilterType</u> (java.lang.String type) 역방향 변환에서 사용할 필터를 설정한다.
void	<u>setHighCutFrequency</u> (float f) 필터의 high cut freq.를 설정한다.
void	<u>setLowCutFrequency</u> (float f) 필터의 low cut freq.를 설정한다.
void	<u>setMethod</u> (java.lang.String method) Forward/Backward 변환을 지정한다.
void	<u>setParameter</u> (float[] freq) 필터 변수를 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target sTgt, Target dTgt) 대상 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### FFTOP

```
public FFTOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

Forward/Backward 변환을 지정한다.

---

### setBand

```
public void setBand(int b)
```

처리 대상 분광대를 설정한다.

---

### setFilterType

```
public void setFilterType(java.lang.String type)
```

역방향 변환에서 사용할 필터를 설정한다.

---

### setParameter

```
public void setParameter(float[] freq)
```

필터 변수를 설정한다.

---

### setFilterOrder

```
public void setFilterOrder(float f)
```

필터에 사용할 order를 설정한다.

---

### setLowCutFrequency

```
public void setLowCutFrequency(float f)
```

필터의 low cut freq.를 설정한다.

---

### setHighCutFrequency

```
public void setHighCutFrequency(float f)
```

필터의 high cut freq.를 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target sTgt,  
                     Target dTgt)
```

대상 영상을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

변환을 실행한다.

Throws:

[java.io.IOException](#) – [IOException](#)

[java.lang.InterruptedException](#) – [InterruptedException](#)

---

## FFT1D

```
protected void FFT1D(float[] ra,  
                    float[] ia,  
                    float[] st,  
                    float[] ct,  
                    int lenth)
```

---

## FFT2D

```
protected void FFT2D(float[][] oPixel,  
                    int lns,  
                    int smps,  
                    int lenth)
```

---

---

## IdealBPFfilteredFrequency

```
protected void IdealBPFfilteredFrequency(float[][] pix,  
                                         int length)
```

---

---

## IdealBSFilteredFrequency

```
protected void IdealBSFilteredFrequency(float[][] pix,  
                                         int length)
```

---

---

## IdealHPFilteredFrequency

```
protected void IdealHPFilteredFrequency(float[][] pix,  
                                         int length)
```

---

---

## NoFilteredFrequency

```
protected void NoFilteredFrequency(float[][] pix,  
                                   int length)
```

---

---

## IdealLPFilteredFrequency

```
protected void IdealLPFilteredFrequency(float[][] pix,  
                                         int length)
```

---

---

## ButterworthLPFilteredFrequency

```
protected void ButterworthLPFilteredFrequency(float[][] pix,  
                                               int length)
```

---

---

## ButterworthHPFilteredFrequency

```
protected void ButterworthHPFilteredFrequency(float[][] pix,  
                                               int length)
```

---

---

## ButterworthBSFilteredFrequency

```
protected void ButterworthBSFilteredFrequency(float[][] pix,  
                                               int length)
```

---

---

## ButterworthBPFfilteredFrequency

```
protected void ButterworthBPFfilteredFrequency(float[][] pix,  
                                               int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class FFTPowerSpectrumOp

```
java.lang.Object
|
+--FFTPowerSpectrumOp
```

```
public class FFTPowerSpectrumOp
extends java.lang.Object
implements Operator
```

영상의 FFT 결과를 이용하여 파워스펙트럼 영상을 생성한다. 출력 결과는 단일 밴드의 바이트형이다.

### Constructor Summary

<a href="#">FFTPowerSpectrumOp()</a> Constructor	
---	--

### Method Summary

void	<a href="#">operate()</a> 파워스펙트럼을 만든다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

#### FFTPowerSpectrumOp

```
public FFTPowerSpectrumOp()
    Constructor
```

### Method Detail

#### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedExcepion
```

파워스펙트럼을 만든다.

Throws:

[java.io.IOException](#) – [IOException](#)

[java.lang.InterruptedExcepion](#) – [InterruptedExcepion](#)

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class GPCCorrectOp

```
java.lang.Object
|
+--GPCCorrectOp
```

---

```
public class GPCCorrectOp
extends java.lang.Object
implements Operator
```

영상의 GCP보정을 실행한다. 대상 영상은 메모리형이어야 한다.

---

Field Summary	
static int	<a href="#">IMAGE_TO_IMAGE</a>
static int	<a href="#">IMAGE_TO_MAP</a>
static int	<a href="#">MAP_TO_MAP</a>

Constructor Summary	
<a href="#">GPCCorrectOp()</a>	
Constructor	

Method Summary	
void	<u>operate()</u> 대상 영상을 설정한다.
void	<u>setBackground(int background)</u> 배경 화소값을 설정한다.
void	<u>setBands(int[] bands)</u> 처리 대상 분광대를 설정한다.
void	<u>setCoefficient(double[][] coef, double[] t)</u> 변환식의 계수를 설정한다.
void	<u>setGeoBounds(GeoRectangle r)</u> 처리 영역을 설정한다.
void	<u>setImageSize(int smps, int lns)</u> 영상의 크기를 설정한다.
void	<u>setMapping(int mapping)</u> 변환 방법을 설정한다.
void	<u>setPolyOrder(int degree)</u> 변환식의 차수를 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setResampling(int resmp)</u> 화소 재배열 방법을 설정한다.
void	<u>setTarget(Target srcTgt, Target destTgt)</u> 대상 영상을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### IMAGE\_TO\_IMAGE

public static final int IMAGE\_TO\_IMAGE

---

### IMAGE\_TO\_MAP

public static final int IMAGE\_TO\_MAP

---

### MAP\_TO\_MAP

public static final int MAP\_TO\_MAP

## Constructor Detail

### GCPCorrectOp

public GCPCorrectOp()

Constructor

## Method Detail

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

public void setTarget(Target srcTgt,  
                          Target destTgt)

대상 영상을 설정한다.

---

### setBands

public void setBands(int[] bands)

처리 대상 분광대를 설정한다.

---

### setMapping

public void setMapping(int mapping)

변환 방법을 설정한다.

---

### setResampling

public void setResampling(int resmp)

화소 재배열 방법을 설정한다.

---

### setPolyOrder

public void setPolyOrder(int degree)

변환식의 차수를 설정한다.

---

### setBackground

public void setBackground(int background)

배경 화소값을 설정한다.

---

## setImageSize

```
public void setImageSize(int smps,  
                        int lns)
```

영상의 크기를 설정한다.

---

## setGeoBounds

```
public void setGeoBounds(GeoRectangle r)
```

처리 영역을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

대상 영상을 설정한다.

---

## setCoefficient

```
public void setCoefficient(double[][] coef,  
                          double[] t)
```

변환식의 계수를 설정한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class GcpRegressor

```
java.lang.Object
|
+--Regressor
|
+--GcpRegressor
```

```
public class GcpRegressor
extends Regressor
```

GCP를 이용한 변환 계수를 계산한다.

### Constructor Summary

<code>GcpRegressor(double[][] G, int degree)</code> Constructor
--

### Method Summary

double[]	<code>getBias()</code> 변환 결과의 수평/수직 이동량을 되돌린다.
double[][]	<code>getGCPEstimate()</code> 변환식을 이용하여 GCP의 위치를 예측한다.
double[]	<code>getMinMaxCoord()</code> 최소, 최대 좌표를 되돌린다.
double[]	<code>getPosition(double x, double y)</code> 입력 좌표에 대한 변환 좌표를 되돌린다.
double[]	<code>getRMS()</code> 변환식에 의한 각 좌표의 rms를 되돌린다.
double	<code>getTotalRMS()</code> 변환식에 의한 전체 평균 rms를 되돌린다.

### Methods inherited from class [Regressor](#)

`getAbsoluteError`, `getCoef`, `getError`, `getEstimate`, `regress`, `setVar`

### Methods inherited from class [java.lang.Object](#)

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Detail

`GcpRegressor`

```
public GcpRegressor(double[][] G,  
                    int degree)  
    throws IllegalFormatException
```

Constructor

Throws:

IllegalFormatException – IllegalFormatException

## Method Detail

### getBias

```
public double[] getBias()
```

변환 결과의 수평/수직 이동량을 되돌린다.

---

### getGCPEstimate

```
public double[][] getGCPEstimate()
```

변환식을 이용하여 GCP의 위치를 예측한다.

---

### getRMS

```
public double[] getRMS()
```

변환식에 의한 각 좌표의 rms를 되돌린다.

---

### getTotalRMS

```
public double getTotalRMS()
```

변환식에 의한 전체 평균 rms를 되돌린다.

---

### getPosition

```
public double[] getPosition(double x,  
                             double y)  
    throws NotMatchedSizeException
```

입력 좌표에 대한 변환 좌표를 되돌린다.

---

### getMinMaxCoord

```
public double[] getMinMaxCoord()
```

최소, 최대 좌표를 되돌린다.



## Class GeoArrayTypeConverter

```
java.lang.Object
|
+--GeoArrayTypeConverter
```

```
public class GeoArrayTypeConverter
extends java.lang.Object
```

### Constructor Summary

**GeoArrayTypeConverter()**  
 주어진 화소형의 배열을 다른 화소형의 배열로 변환한다.

### Method Summary

double[]	<b>operate</b> (byte[] src, int srcOffset, double[] dest, int destOffset, int length) 바이트형을 8바이트 실수형으로 변환하여 되돌린다.
float[]	<b>operate</b> (byte[] src, int srcOffset, float[] dest, int destOffset, int length) 바이트형을 4바이트 실수형으로 변환하여 되돌린다.
int[]	<b>operate</b> (byte[] src, int srcOffset, int[] dest, int destOffset, int length) 바이트형을 4바이트 정수형으로 변환하여 되돌린다.
short[]	<b>operate</b> (byte[] src, int srcOffset, short[] dest, int destOffset, int length) 바이트형을 2바이트 정수형으로 변환하여 되돌린다.
double[]	<b>operate</b> (float[] src, int srcOffset, double[] dest, int destOffset, int length) 4바이트 실수형을 8바이트 실수형으로 변환하여 되돌린다.
float[]	<b>operate</b> (float[] src, int srcOffset, float[] dest, int destOffset, int length) 4바이트 실수형을 4바이트 실수형으로 변환하여 되돌린다.
double[]	<b>operate</b> (int[] src, int srcOffset, double[] dest, int destOffset, int length) 4바이트 정수형을 8바이트 실수형으로 변환하여 되돌린다.
int[]	<b>operate</b> (int[] src, int srcOffset, int[] dest, int destOffset, int length) 4바이트 정수형을 4바이트 정수형으로 변환하여 되돌린다.
double[]	<b>operate</b> (short[] src, int srcOffset, boolean signed, double[] dest, int destOffset, int length) 2바이트 정수형을 8바이트 실수형으로 변환하여 되돌린다.
int[]	<b>operate</b> (short[] src, int srcOffset, boolean signed, int[] dest, int destOffset, int length) 2바이트 정수형을 4바이트 정수형으로 변환하여 되돌린다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GeoArrayTypeConverter

```
public GeoArrayTypeConverter()
```

주어진 화소형의 배열을 다른 화소형의 배열로 변환한다.

## Method Detail

### operate

```
public short[] operate(byte[] src,
                       int srcOffset,
                       short[] dest,
                       int destOffset,
                       int length)
```

바이트형을 2바이트 정수형으로 변환하여 되돌린다.

---

### operate

```
public int[] operate(byte[] src,
                     int srcOffset,
                     int[] dest,
                     int destOffset,
                     int length)
```

바이트형을 4바이트 정수형으로 변환하여 되돌린다.

---

### operate

```
public float[] operate(byte[] src,
                       int srcOffset,
                       float[] dest,
                       int destOffset,
                       int length)
```

바이트형을 4바이트 실수형으로 변환하여 되돌린다.

---

### operate

```
public double[] operate(byte[] src,
                        int srcOffset,
                        double[] dest,
                        int destOffset,
                        int length)
```

바이트형을 8바이트 실수형으로 변환하여 되돌린다.

---

### operate

```
public int[] operate(short[] src,
```

```
int srcOffset,  
boolean signed,  
int[] dest,  
int destOffset,  
int length)
```

2바이트 정수형을 4바이트 정수형으로 변환하여 되돌린다.

---

## operate

```
public double[] operate(short[] src,  
int srcOffset,  
boolean signed,  
double[] dest,  
int destOffset,  
int length)
```

2바이트 정수형을 8바이트 실수형으로 변환하여 되돌린다.

---

## operate

```
public int[] operate(int[] src,  
int srcOffset,  
int[] dest,  
int destOffset,  
int length)
```

4바이트 정수형을 4바이트 정수형으로 변환하여 되돌린다.

---

## operate

```
public double[] operate(int[] src,  
int srcOffset,  
double[] dest,  
int destOffset,  
int length)
```

4바이트 정수형을 8바이트 실수형으로 변환하여 되돌린다.

---

## operate

```
public float[] operate(float[] src,  
int srcOffset,  
float[] dest,  
int destOffset,  
int length)
```

4바이트 실수형을 4바이트 실수형으로 변환하여 되돌린다.

---

## operate

```
public double[] operate(float[] src,  
int srcOffset,
```

```
double[] dest,  
int destOffset,  
int length)
```

4바이트 실수형을 8바이트 실수형으로 변환하여 되돌린다.

---

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

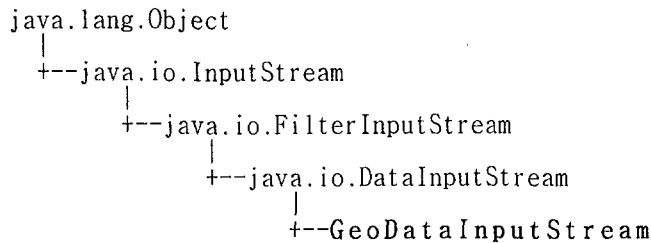
FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

---

## Class `GeoDataInputStream`



```

public class GeoDataInputStream
extends java.io.DataInputStream
    
```

파일 입력 스트림을 정의한다.

Fields inherited from class <code>java.io.FilterInputStream</code>
<code>in</code>

Constructor Summary
<code>GeoDataInputStream</code> ( <code>java.io.InputStream in</code> ) Constructor

Method Summary
<code>int</code> <code>read</code> ( <code>float[] buf</code> ) 4바이트 실수형의 자료를 입력한다.
<code>int</code> <code>read</code> ( <code>float[] buf, int offset, int length</code> ) 4바이트 실수형의 자료를 입력한다.
<code>int</code> <code>read</code> ( <code>int[] buf</code> ) 4바이트 정수형의 자료를 입력한다.
<code>int</code> <code>read</code> ( <code>int[] buf, int offset, int length</code> ) 4바이트 정수형의 자료를 입력한다.
<code>int</code> <code>read</code> ( <code>short[] buf</code> ) 2바이트 정수형의 자료를 읽는다.
<code>int</code> <code>read</code> ( <code>short[] buf, int offset, int length</code> ) 2바이트 정수형의 자료를 입력한다.
<code>java.lang.String</code> <code>readFixedString</code> ( <code>int size</code> ) 고정된 길이의 문자열을 입력한다.

## Methods inherited from class java.io.DataInputStream

read, readBoolean, readByte, readChar, readDouble, readFloat, readFully, readFully, readInt, readLine, readLong, readShort, readUnsignedByte, readUnsignedShort, readUTF, readUTF, skipBytes

## Methods inherited from class java.io.FilterInputStream

available, close, mark, markSupported, read, reset, skip

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GeoDataInputStream

```
public GeoDataInputStream(java.io.InputStream in)
```

Constructor

## Method Detail

### read

```
public int read(short[] buf)
    throws java.io.IOException
```

2바이트 정수형의 자료를 읽는다.

Throws:

java.io.IOException - IOException

---

### read

```
public int read(short[] buf,
                int offset,
                int length)
    throws java.io.IOException
```

2바이트 정수형의 자료를 입력한다.

Throws:

java.io.IOException - IOException

---

### read

```
public int read(int[] buf)
    throws java.io.IOException
```

4바이트 정수형의 자료를 입력한다.

Throws:

java.io.IOException - IOException

---

## read

```
public int read(int[] buf,  
               int offset,  
               int length)  
    throws java.io.IOException
```

4바이트 정수형의 자료를 입력한다.

Throws:  
java.io.IOException - IOException

---

## read

```
public int read(float[] buf)  
    throws java.io.IOException
```

4바이트 실수형의 자료를 입력한다.

Throws:  
java.io.IOException - IOException

---

## read

```
public int read(float[] buf,  
               int offset,  
               int length)  
    throws java.io.IOException
```

4바이트 실수형의 자료를 입력한다.

Throws:  
java.io.IOException - IOException

---

## readFixedString

```
public java.lang.String readFixedString(int size)  
    throws java.io.IOException
```

고정된 길이의 문자열을 입력한다.

Throws:  
java.io.IOException - IOException

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class `GeoDataOutputStream`

```
java.lang.Object
|
+--java.io.OutputStream
    |
    +--java.io.FilterOutputStream
        |
        +--java.io.DataOutputStream
            |
            +--GeoDataOutputStream
```

---

```
public class GeoDataOutputStream
    extends java.io.DataOutputStream
```

파일 출력 스트림을 정의한다.

---

### Fields inherited from class `java.io.DataOutputStream`

written

### Fields inherited from class `java.io.FilterOutputStream`

out

### Constructor Summary

<code>GeoDataOutputStream</code> ( <code>java.io.OutputStream</code> out)
Constructor



## Method Summary

void	<code>write(float[] buffer)</code> 4바이트 실수형의 자료를 출력한다.
void	<code>write(float[] buffer, int offset, int length)</code> 4바이트 실수형의 자료를 출력한다.
void	<code>write(int[] buffer)</code> 4바이트 정수형의 자료를 출력한다.
void	<code>write(int[] buffer, int offset, int length)</code> 4바이트 정수형의 자료를 출력한다.
void	<code>write(short[] buffer)</code> 2바이트 정수형의 자료를 출력한다.
void	<code>write(short[] buffer, int offset, int length)</code> 2바이트 정수형의 자료를 출력한다.
void	<code>writeFixedString(java.lang.String s, int size)</code> 고정된 길이의 문자열을 출력한다.
void	<code>x86writeInt(int v)</code> 32비트인 Int Integer를 OutputStream에 쓰기 위한 메서드
void	<code>x86writeShort(int v)</code> 16비트인 Short Integer를 OutputStream에 쓰기 위한 메서드

## Methods inherited from class java.io.DataOutputStream

flush, size, write, write, writeBoolean, writeByte, writeBytes, writeChar, writeChars, writeDouble, writeFloat, writeInt, writeLong, writeShort, writeUTF

## Methods inherited from class java.io.FilterOutputStream

close, write

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GeoDataOutputStream

```
public GeoDataOutputStream(java.io.OutputStream out)
```

Constructor

## Method Detail

### write

```
public void write(short[] buffer)  
    throws java.io.IOException
```

2바이트 정수형의 자료를 출력한다.

Throws:

java.io.IOException – IOException

## write

```
public void write(short[] buffer,  
                 int offset,  
                 int length)  
    throws java.io.IOException
```

2바이트 정수형의 자료를 출력한다.

Throws:  
java.io.IOException - IOException

---

## write

```
public void write(int[] buffer)  
    throws java.io.IOException
```

4바이트 정수형의 자료를 출력한다.

Throws:  
java.io.IOException - IOException

---

## write

```
public void write(int[] buffer,  
                 int offset,  
                 int length)  
    throws java.io.IOException
```

4바이트 정수형의 자료를 출력한다.

Throws:  
java.io.IOException - IOException

---

## write

```
public void write(float[] buffer)  
    throws java.io.IOException
```

4바이트 실수형의 자료를 출력한다.

Throws:  
java.io.IOException - IOException

---

## write

```
public void write(float[] buffer,  
                 int offset,  
                 int length)  
    throws java.io.IOException
```

4바이트 실수형의 자료를 출력한다.

Throws:  
java.io.IOException - IOException

---

## writeFixedString

```
public void writeFixedString(java.lang.String s, .
                             int size)
                             throws java.io.IOException
```

고정된 길이의 문자열을 출력한다.

Throws:

java.io.IOException - IOException

---

## x86writeShort

```
public void x86writeShort(int v)
                          throws java.io.IOException
```

16비트인 Short Integer를 OutputStream에 쓰기 위한 메서드

Parameters:

v - 스트림에 쓰기 위한 정수

---

## x86writeInt

```
public void x86writeInt(int v)
                      throws java.io.IOException
```

32비트인 Int Integer를 OutputStream에 쓰기 위한 메서드

Parameters:

v - 스트림에 쓰기 위한 정수

---

### [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
 SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
 DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class GeoEnviron

```
java.lang.Object
|
+--GeoEnviron
```

```
public class GeoEnviron
extends java.lang.Object
```

Field Summary	
static <a href="#">GeoCommandTable</a>	<a href="#">CmdTable</a> 명령어 관리 테이블
static <a href="#">GeoCubeListDialog</a>	<a href="#">CubeListDlg</a> 영상 목록 관리 창
static java.lang.String	<a href="#">CurrentPath</a> 사용자 디렉토리
static int	<a href="#">HistoryCount</a> 명령어 저장 크기
static <a href="#">GeoKeyTable</a>	<a href="#">KeyTable</a> 명령어 키워드 관리 테이블
static int	<a href="#">MainViewSize</a> 주 윈도우의 초기 크기값
static java.awt.FileDialog	<a href="#">OpenFileDlg</a> 파일 열기 입력창
static java.lang.String	<a href="#">Property</a> GeoPixel 환경 관리
static java.awt.FileDialog	<a href="#">SaveFileDlg</a> 파일 저장 입력창
static int	<a href="#">ScrollViewSize</a> 스크롤 윈도우의 초기 크기값
static int	<a href="#">UserMenuPosition</a> 사용자 메뉴 목록에서의 초기 위치
static int	<a href="#">ZoomViewScale</a> 확대 비율 초기값
static int	<a href="#">ZoomViewSize</a> 확대 윈도우의 초기 크기값

Constructor Summary	
<a href="#">GeoEnviron()</a>	

Method Summary	
static boolean	<u>ask</u> (java.awt.Window precursor, java.lang.String q) 사용자 인터럽트에 대하여 질의를 하고 그 결과를 되돌린다.
static Command	<u>getCommand</u> (java.lang.String id) 이력 ID를 갖는 명령어를 되돌린다.
static GeoMainFrame	<u>getCurrentMainFrame</u> () 현재 설정되어 있는 주프레임을 되돌린다.
static GeoCommandFrame	<u>getMainCommandFrame</u> () 주명령어 프레임을 되돌린다.
static GeoMenuFrame	<u>getMainMenuFrame</u> () 주메뉴 프레임을 되돌린다.
static void	<u>help</u> (java.lang.String s) 사용자 입력창에서의 도움말 기능을 지원한다.
static void	<u>initialize</u> () GeoPixel의 각종 환경 변수를 초기화 하고 메뉴프레임과 명령어 프레임을 가동한다.
static void	<u>message</u> (java.awt.Window precursor, java.lang.Exception e) Exception 발생에 대하여 메시지를 출력한다.
static void	<u>message</u> (java.awt.Window precursor, java.lang.String msg) 사용자 인터럽트에 대하여 메시지를 출력한다.
static int	<u>pushImageDisplay</u> (CubeViewOp op) 화면 출력 오퍼레이터를 관리 목록에 삽입하고 목록에서의 위치를 되돌린다.
static void	<u>readCommandObject</u> () 명령어 객체를 파일로부터 읽는다.
static void	<u>setCurrentMainFrame</u> (GeoMainFrame f) 현재의 주프레임을 설정한다.
static java.awt.Rectangle	<u>synopsis</u> (java.awt.Window pWindow, Target tgt, int bno, java.awt.Rectangle r) 사용자 입력창에서 Screen Synopsis 기능을 지원한다.
static void	<u>writeCommandObject</u> (GeoCommandTable table) 명령어 객체를 테이블에 출력한다.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### Property

public static final java.lang.String Property

GeoPixel 환경 관리

### CurrentPath

public static java.lang.String CurrentPath

## ScrollViewSize

```
public static int ScrollViewSize
```

스크롤 윈도우의 초기 크기값

---

## MainViewSize

```
public static int MainViewSize
```

주 윈도우의 초기 크기값

---

## ZoomViewSize

```
public static int ZoomViewSize
```

확대 윈도우의 초기 크기값

---

## ZoomViewScale

```
public static int ZoomViewScale
```

확대 비율 초기값

---

## HistoryCount

```
public static int HistoryCount
```

명령어 저장 크기

---

## UserMenuPosition

```
public static int UserMenuPosition
```

사용자 메뉴 목록에서의 초기 위치

---

## CmdTable

```
public static GeoCommandTable CmdTable
```

명령어 관리 테이블

---

## KeyTable

public static [GeoKeyTable](#) KeyTable

명령어 키워드 관리 테이블

---

## CubeListDlg

public static [GeoCubeListDialog](#) CubeListDlg

영상 목록 관리 창

---

## OpenFileDlg

public static [java.awt.FileDialog](#) OpenFileDlg

파일 열기 입력창

---

## SaveFileDlg

public static [java.awt.FileDialog](#) SaveFileDlg

파일 저장 입력창

---

## Constructor Detail

### GeoEnviron

public [GeoEnviron](#)()

## Method Detail

### initialize

public static void [initialize](#)()

GeoPixel의 각종 환경 변수를 초기화 하고 메뉴프레임과 명령어 프레임을 가동한다.

---

## getCurrentMainFrame

public static [GeoMainFrame](#) [getCurrentMainFrame](#)()

현재 설정되어 있는 주프레임을 되돌린다.

---

## setCurrentMainFrame

```
public static void setCurrentMainFrame(GeoMainFrame f)
```

현재의 주프레임을 설정한다.

---

## getMainCommandFrame

```
public static GeoCommandFrame getMainCommandFrame()
```

주명령어 프레임을 되돌린다.

---

## getMainMenuFrame

```
public static GeoMenuFrame getMainMenuFrame()
```

주메뉴 프레임을 되돌린다.

---

## message

```
public static void message(java.awt.Window precursor,  
                           java.lang.String msg)
```

사용자 인터럽트에 대하여 메시지를 출력한다.

---

## message

```
public static void message(java.awt.Window precursor,  
                           java.lang.Exception e)
```

Exception 발생에 대하여 메시지를 출력한다.

---

## ask

```
public static boolean ask(java.awt.Window precursor,  
                          java.lang.String q)
```

사용자 인터럽트에 대하여 질의를 하고 그 결과를 되돌린다.

---

## getCommand

```
public static Command getCommand(java.lang.String id)
```

이력 ID를 갖는 명령어를 되돌린다.

---



## readCommandObject

```
public static void readCommandObject()  
    throws java.io.IOException,  
           java.lang.ClassNotFoundException
```

명령어 객체를 파일로부터 읽는다.

Throws:

java.io.IOException – IOException  
java.lang.ClassNotFoundException – ClassNotFoundException

---

## writeCommandObject

```
public static void writeCommandObject(GeoCommandTable table)  
    throws java.io.IOException
```

명령어 객체를 테이블에 출력한다.

Throws:

java.io.IOException – IOException

---

## pushImageDisplay

```
public static int pushImageDisplay(CubeViewOp op)
```

화면 출력 오퍼레이터를 관리 목록에 삽입하고 목록에서의 위치를 되돌린다.

---

## synopsis

```
public static java.awt.Rectangle synopsis(java.awt.Window pWindow,  
    Target tgt,  
    int bno,  
    java.awt.Rectangle r)
```

사용자 입력창에서 Screen Synopsis 기능을 지원한다.

---

## help

```
public static void help(java.lang.String s)
```

사용자 입력창에서의 도움말 기능을 지원한다.

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class GeoEnviron

java.lang.Object

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class GeoFileIOTextField

java.lang.Object

```
graph TD
    A[+--java.lang.Object] --> B[+--java.awt.Component]
    B --> C[+--java.awt.TextComponent]
    C --> D[+--java.awt.TextField]
    D --> E[+--GeoFileIOTextField]
```

---

```
public class GeoFileIO TextField
extends java.awt.TextField
```

영상이외의 파일 객체를 관리한다.

See Also:  
[Serialized Form](#)

---

### Fields inherited from class java.awt.TextComponent

textListener

### Fields inherited from class java.awt.Component

BOTTOM\_ALIGNMENT, CENTER\_ALIGNMENT, LEFT\_ALIGNMENT, RIGHT\_ALIGNMENT, TOP\_ALIGNMENT

## Constructor Summary

[GeoFileIOTextField\(\)](#)  
Constructor

[GeoFileIOTextField\(java.io.File f\)](#)  
Constructor

## Method Summary

java.io.File	<b>getFile()</b> 설정된 파일 객체를 되돌린다.
boolean	<b>isEmpty()</b> 설정된 파일 객체를 검사한다.
void	<b>setFile(java.io.File f)</b> 파일 객체를 설정한다.
void	<b>setFile(java.lang.String s)</b> 파일명을 설정한다.

## Methods inherited from class java.awt.TextField

addActionListener, addNotify, echoCharIsSet, getColumns, getEchoChar, getMinimumSize, getPreferredSize, getPreferredSize, getPreferredSize, minimumSize, minimumSize, paramString, preferredSize, preferredSize, processActionEvent, processEvent, removeActionListener, setColumns, setEchoChar, setEchoCharacter, setText

## Methods inherited from class java.awt.TextComponent

addTextListener, getCaretPosition, getSelectedText, getSelectionEnd, getSelectionStart, getText, isEditable, processTextEvent, removeNotify, removeTextListener, select, selectAll, setCaretPosition, setEditable, setSelectionEnd, setSelectionStart

## Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getName, getParent, getPeer, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, prepareImage, prepareImage, print, printAll, processComponentEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFont, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus, update, validate

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoFileIOTextField

```
public GeoFileIOTextField()
```

Constructor

---

## GeoFileIOTextField

public GeoFileIOTextField(java.io.File f)

Constructor

### Method Detail

#### setFile

public void setFile(java.io.File f)

파일 객체를 설정한다.

---

#### setFile

public void setFile(java.lang.String s)

파일명을 설정한다.

---

#### getFile

public java.io.File getFile()

설정된 파일 객체를 되돌린다.

---

#### isEmpty

public boolean isEmpty()

설정된 파일 객체를 검사한다.

---

### Class [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

+--GeoEnviron

---

```
public class GeoEnviron
extends java.lang.Object
```

---

Field Summary	
static GeoCommandTable	<u>CmdTable</u> 명령어 관리 테이블
static GeoCubeListDialog	<u>CubeListDlg</u> 영상 목록 관리 창
static java.lang.String	<u>CurrentPath</u> 사용자 디렉토리
static int	<u>HistoryCount</u> 명령어 저장 크기
static <u>GeoKeyTable</u>	<u>KeyTable</u> 명령어 키워드 관리 테이블
static int	<u>MainViewSize</u> 주 윈도우의 초기 크기값
static java.awt.FileDialog	<u>OpenFileDlg</u> 파일 열기 입력창
static java.lang.String	<u>Property</u> GeoPixel 환경 관리
static java.awt.FileDialog	<u>SaveFileDlg</u> 파일 저장 입력창
static int	<u>ScrollViewSize</u> 스크롤 윈도우의 초기 크기값
static int	<u>UserMenuPosition</u> 사용자 메뉴 목록에서의 초기 위치
static int	<u>ZoomViewScale</u> 확대 비율 초기값
static int	<u>ZoomViewSize</u> 확대 윈도우의 초기 크기값

Constructor Summary	
<u>GeoEnviron()</u>	

Method Summary	
static boolean	<u>ask</u> (java.awt.Window precursor, java.lang.String q) 사용자 인터럽트에 대하여 질의를 하고 그 결과를 되돌린다.
static Command	<u>getCommand</u> (java.lang.String id) 이력 ID를 갖는 명령어를 되돌린다.
static GeoMainFrame	<u>getCurrentMainFrame</u> () 현재 설정되어 있는 주프레임을 되돌린다.
static GeoCommandFrame	<u>getMainCommandFrame</u> () 주명령어 프레임을 되돌린다.
static GeoMenuFrame	<u>getMainMenuFrame</u> () 주메뉴 프레임을 되돌린다.
static void	<u>help</u> (java.lang.String s) 사용자 입력창에서의 도움말 기능을 지원한다.
static void	<u>initialize</u> () GeoPixel의 각종 환경 변수를 초기화 하고 메뉴프레임과 명령어 프레임을 가동한다.
static void	<u>message</u> (java.awt.Window precursor, java.lang.Exception e) Exception 발생에 대하여 메시지를 출력한다.
static void	<u>message</u> (java.awt.Window precursor, java.lang.String msg) 사용자 인터럽트에 대하여 메시지를 출력한다.
static int	<u>pushImageDisplay</u> (CubeViewOp op) 화면 출력 오퍼레이터를 관리 목록에 삽입하고 목록에서의 위치를 되돌린다.
static void	<u>readCommandObject</u> () 명령어 객체를 파일로부터 읽는다.
static void	<u>setCurrentMainFrame</u> (GeoMainFrame f) 현재의 주프레임을 설정한다.
static java.awt.Rectangle	<u>synopsis</u> (java.awt.Window pWindow, Target tgt, int bno, java.awt.Rectangle r) 사용자 입력창에서 Screen Synopsis 기능을 지원한다.
static void	<u>writeCommandObject</u> (GeoCommandTable table) 명령어 객체를 테이블에 출력한다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### Property

public static final java.lang.String Property

GeoPixel 환경 관리

### CurrentPath

public static java.lang.String CurrentPath

## ScrollViewSize

```
public static int ScrollViewSize
```

스크롤 윈도우의 초기 크기값

---

## MainViewSize

```
public static int MainViewSize
```

주 윈도우의 초기 크기값

---

## ZoomViewSize

```
public static int ZoomViewSize
```

확대 윈도우의 초기 크기값

---

## ZoomViewScale

```
public static int ZoomViewScale
```

확대 비율 초기값

---

## HistoryCount

```
public static int HistoryCount
```

명령어 저장 크기

---

## UserMenuPosition

```
public static int UserMenuPosition
```

사용자 메뉴 목록에서의 초기 위치

---

## CmdTable

```
public static GeoCommandTable CmdTable
```

명령어 관리 테이블

---

## KeyTable

public static [GeoKeyTable](#) KeyTable

명령어 키워드 관리 테이블

---

## CubeListDlg

public static [GeoCubeListDialog](#) CubeListDlg

영상 목록 관리 창

---

## OpenFileDlg

public static [java.awt.FileDialog](#) OpenFileDlg

파일 열기 입력창

---

## SaveFileDlg

public static [java.awt.FileDialog](#) SaveFileDlg

파일 저장 입력창

---

## Constructor Detail

### GeoEnviron

public [GeoEnviron](#)()

## Method Detail

### initialize

public static void initialize()

GeoPixel의 각종 환경 변수를 초기화 하고 메뉴프레임과 명령어 프레임을 가동한다.

---

## getCurrentMainFrame

public static [GeoMainFrame](#) getCurrentMainFrame()

현재 설정되어 있는 주프레임을 되돌린다.

---

## setCurrentMainFrame



```
public static void setCurrentMainFrame(GeoMainFrame f)
```

현재의 주프레임을 설정한다.

---

## getMainCommandFrame

```
public static GeoCommandFrame getMainCommandFrame()
```

주명령어 프레임을 되돌린다.

---

## getMainMenuFrame

```
public static GeoMenuFrame getMainMenuFrame()
```

주메뉴 프레임을 되돌린다.

---

## message

```
public static void message(java.awt.Window precursor,  
                           java.lang.String msg)
```

사용자 인터럽트에 대하여 메시지를 출력한다.

---

## message

```
public static void message(java.awt.Window precursor,  
                           java.lang.Exception e)
```

Exception 발생에 대하여 메시지를 출력한다.

---

## ask

```
public static boolean ask(java.awt.Window precursor,  
                          java.lang.String q)
```

사용자 인터럽트에 대하여 질의를 하고 그 결과를 되돌린다.

---

## getCommand

```
public static Command getCommand(java.lang.String id)
```

이력 ID를 갖는 명령어를 되돌린다.

---

## readCommandObject

```
public static void readCommandObject()  
    throws java.io.IOException,  
           java.lang.ClassNotFoundException
```

명령어 객체를 파일로부터 읽는다.

Throws:

java.io.IOException – IOException  
java.lang.ClassNotFoundException – ClassNotFoundException

---

## writeCommandObject

```
public static void writeCommandObject(GeoCommandTable table)  
    throws java.io.IOException
```

명령어 객체를 테이블에 출력한다.

Throws:

java.io.IOException – IOException

---

## pushImageDisplay

```
public static int pushImageDisplay(CubeViewOp op)
```

화면 출력 오퍼레이터를 관리 목록에 삽입하고 목록에서의 위치를 되돌린다.

---

## synopsis

```
public static java.awt.Rectangle synopsis(java.awt.Window pWindow,  
    Target tgt,  
    int bno,  
    java.awt.Rectangle r)
```

사용자 입력창에서 Screen Synopsis 기능을 지원한다.

---

## help

```
public static void help(java.lang.String s)
```

사용자 입력창에서의 도움말 기능을 지원한다.

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

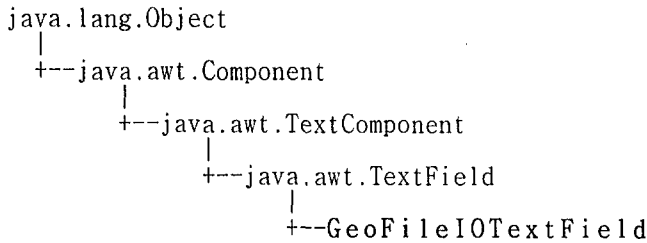
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class GeoFileIOTextField



public class GeoFileIOTextField  
 extends java.awt.TextField

영상이외의 파일 객체를 관리한다.

See Also:  
[Serialized Form](#)

<b>Fields inherited from class java.awt.TextComponent</b>
textListener

<b>Fields inherited from class java.awt.Component</b>
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

<b>Constructor Summary</b>	
<a href="#">GeoFileIOTextField()</a>	Constructor
<a href="#">GeoFileIOTextField(java.io.File f)</a>	Constructor

<b>Method Summary</b>	
java.io.File	<a href="#">getFile()</a> 설정된 파일 객체를 되돌린다.
boolean	<a href="#">isEmpty()</a> 설정된 파일 객체를 검사한다.
void	<a href="#">setFile(java.io.File f)</a> 파일 객체를 설정한다.
void	<a href="#">setFile(java.lang.String s)</a> 파일명을 설정한다.

### Methods inherited from class java.awt.TextField

addActionListener, addNotify, echoCharIsSet, getColumns, getEchoChar, getMinimumSize, getPreferredSize, getPreferredSize, minimumSize, minimumSize, paramString, preferredSize, preferredSize, processActionEvent, processEvent, removeActionListener, setColumns, setEchoChar, setEchoCharacter, setText

### Methods inherited from class java.awt.TextComponent

addTextListener, getCaretPosition, getSelectedText, getSelectionEnd, getSelectionStart, getText, isEditable, processTextEvent, removeNotify, removeTextListener, select, selectAll, setCaretPosition, setEditable, setSelectionEnd, setSelectionStart

### Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getName, getParent, getPeer, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paint, paintAll, postEvent, prepareImage, prepareImage, print, printAll, processComponentEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFont, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus, update, validate

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoFileIOTextField

```
public GeoFileIOTextField()
```

Constructor

---

### GeoFileIOTextField

```
public GeoFileIOTextField(java.io.File f)
```

Constructor

## Method Detail

setFile

```
public void setFile(java.io.File f)
```

파일 객체를 설정한다.

---

## setFile

```
public void setFile(java.lang.String s)
```

파일명을 설정한다.

---

## getFile

```
public java.io.File getFile()
```

설정된 파일 객체를 되돌린다.

---

## isEmpty

```
public boolean isEmpty()
```

설정된 파일 객체를 검사한다.

---

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY: INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class GeoKernel

java.lang.Object

|--GeoKernel

---

```
public class GeoKernel
extends java.lang.Object
```

필터링과 관련된 커널을 관리한다.

---

Field Summary	
static int	<a href="#">FLOAT</a>
static int	<a href="#">INT</a>

Constructor Summary
<a href="#">GeoKernel(int width, int height, float[] data)</a> Create a Kernel object from an array of floats.
<a href="#">GeoKernel(int width, int height, float[] data, int divisor)</a> Constructor
<a href="#">GeoKernel(int width, int height, int[] data)</a> Constructor
<a href="#">GeoKernel(int width, int height, int[] data, int divisor)</a> Constructor

## Method Summary

int	<code>getDivisor()</code> divisor를 되돌린다.
int	<code>getHeight()</code> Returns the height
float[]	<code>getKernelData(float[] data)</code> Returns the kernel data in row major order.
int[]	<code>getKernelData(int[] data)</code> Returns the kernel data in row major order.
int	<code>getMode()</code> 연산에 관련된 화소형(float/int)를 되돌린다.
int	<code>getWidth()</code> Returns the width
int	<code>getXOrigin()</code> Returns the X origin.
int	<code>getYOrigin()</code> Returns the Y origin.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### INT

```
public static final int INT
```

---

### FLOAT

```
public static final int FLOAT
```

## Constructor Detail

### GeoKernel

```
public GeoKernel(int width,  
                 int height,  
                 float[] data)
```

Create a Kernel object from an array of floats. The data array is copied.

#### Parameters:

- width - Width of the kernel.
  - height - Height of the kernel.
  - data - Kernel data in row major order.
- 

### GeoKernel

```
public GeoKernel(int width,  
                 int height,
```

```
float[] data,  
int divisor)
```

Constructor

Parameters:

width - Width of the kernel.  
height - Height of the kernel.  
data - Kernel data in row major order.  
divisor - divisor for convolution result

---

## GeoKernel

```
public GeoKernel(int width,  
int height,  
int[] data)
```

Constructor

Parameters:

width - Width of the kernel.  
height - Height of the kernel.  
data - Kernel data in row major order.

---

## GeoKernel

```
public GeoKernel(int width,  
int height,  
int[] data,  
int divisor)
```

Constructor

Parameters:

width - Width of the kernel.  
height - Height of the kernel.  
data - Kernel data in row major order.  
divisor - divisor for convolution result

## Method Detail

### getDivisor

```
public int getDivisor()
```

divisor를 되돌린다.

---

### getMode

```
public int getMode()
```

연산에 관련된 화소형(float/int)를 되돌린다.

---

### getXOrigin



```
public int getXOrigin()
```

Returns the X origin.

---

## getYOrigin

```
public int getYOrigin()
```

Returns the Y origin.

---

## getWidth

```
public int getWidth()
```

Returns the width

---

## getHeight

```
public int getHeight()
```

Returns the height

---

## getKernelData

```
public float[] getKernelData(float[] data)
```

Returns the kernel data in row major order.

Parameters:

data – If non-null, will contain the returned kernel data.

---

## getKernelData

```
public int[] getKernelData(int[] data)
```

Returns the kernel data in row major order.

Parameters:

data – If non-null, will contain the returned kernel data.

---

[Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class GeoLookupTable

java.lang.Object  
|  
+--GeoLookupTable

public class GeoLookupTable  
extends java.lang.Object

화소값의 변환을 위한 참조표를 관리한다.

Constructor Summary	
<code>GeoLookupTable()</code>	Constructor
<code>GeoLookupTable(java.lang.String title, int[][] data)</code>	Constructor

Method Summary	
java.lang.String	<code>getTitle()</code> 타이틀을 되돌린다.
void	<code>read(java.io.File f)</code> 변환참조표를 파일에서 입력한다.
void	<code>read(java.lang.String path)</code> 변환참조표를 파일에서 입력한다.
java.lang.String	<code>toString()</code>
void	<code>write(java.io.File f)</code> 변환참조표를 파일로 출력한다.
void	<code>write(java.lang.String path)</code> 변환참조표를 파일로 출력한다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoLookupTable

public GeoLookupTable()

Constructor

---

## GeoLookupTable

```
public GeoLookupTable(java.lang.String title,  
                      int[][] data)
```

Constructor

### Method Detail

#### getTitle

```
public java.lang.String getTitle()
```

타이틀을 되돌린다.

---

#### read

```
public void read(java.lang.String path)  
    throws java.io.IOException,  
           IllegalFormatException
```

변환참조표를 파일에서 입력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException

---

#### read

```
public void read(java.io.File f)  
    throws java.io.IOException,  
           IllegalFormatException
```

변환참조표를 파일에서 입력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException

---

#### write

```
public void write(java.lang.String path)  
    throws java.io.IOException
```

변환참조표를 파일로 출력한다.

Throws:

java.io.IOException – IOException

---

#### write

```
public void write(java.io.File f)
```

throws java.io.IOException

변환참조표를 파일로 출력한다.

Throws:

java.io.IOException – IOException

---

## toString

public java.lang.String toString()

Overrides:

toString in class java.lang.Object

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class `GeoMainImageCanvas`

```
java.lang.Object
|
+--java.awt.Component
|   |
|   +--java.awt.Canvas
|       |
|       +--GeoMainImageCanvas
```

---

```
public class GeoMainImageCanvas
    extends java.awt.Canvas
```

주 영상 화면을 관리한다.

See Also:  
[Serialized Form](#)

---

### Fields inherited from class `java.awt.Component`

`BOTTOM_ALIGNMENT`, `CENTER_ALIGNMENT`, `LEFT_ALIGNMENT`, `RIGHT_ALIGNMENT`, `TOP_ALIGNMENT`

### Constructor Summary

<code>GeoMainImageCanvas(CubeViewOp op)</code> Constructor
---

## Method Summary

void	<u>add</u> (java.awt.PopupMenu menu) 캔버스에 팝업 메뉴를 추가한다.
boolean	<u>adjustBoxArea</u> (java.awt.Rectangle r) 선택영역의 크기를 조절한다.
void	<u>drawAOILine</u> () 화면에 AOI의 경계선을 그린다.
void	<u>drawImage</u> (java.awt.Image img) 화면에 이미지를 출력한다.
void	<u>drawVectorLayer</u> () 벡터 레이어를 그린다.
void	<u>flush</u> () 이미지 버퍼를 Flushing한다.
java.awt.Rectangle	<u>getBoxArea</u> () 선택 영역을 되돌린다.
java.awt.Color	<u>getBoxColor</u> () 화면의 영역을 지정하는 경계색을 되돌린다.
java.awt.Image	<u>getImage</u> () 이미지 버퍼를 되돌린다.
java.awt.Rectangle	<u>getRegion</u> () 화면에 표시된 영역을 되돌린다.
void	<u>linkAOILayer</u> (AOILayer layer) 화면에 AOI 레이어를 연결한다.
void	<u>linkObjectLayer</u> (ObjectLayer layer) 화면에 객체 레이어를 연결한다.
void	<u>linkVectorLayer</u> (VectorLayer layer) 화면에 벡터 레이어를 연결한다.
void	<u>moveBoxArea</u> (java.awt.Rectangle newBox) 박스로 표시된 영역의 이동을 관리한다.
void	<u>paint</u> (java.awt.Graphics g)
void	<u>processMouseEvent</u> (java.awt.event.MouseEvent e) 마우스 이벤트를 받아 팝업 메뉴를 동작시킨다.
void	<u>redrawImage</u> () 영역부분을 다시 그린다.
void	<u>redrawImage</u> (java.awt.Rectangle r) 영역부분을 다시 그린다.
void	<u>setBoxColor</u> (java.awt.Color color) 선택영역의 색을 설정한다.
void	<u>setGrid</u> (int i) 화면에 격자망을 표시하기 위한 격자간격을 설정한다.
void	<u>setRegion</u> (java.awt.Rectangle r) 화면에 영역을 설정한다.
void	<u>update</u> (java.awt.Graphics g)

### Methods inherited from class java.awt.Canvas

addNotify

## Methods inherited from class java.awt.Component

action, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getName, getParent, getPeer, getPreferredSize, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, paramString, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, processComponentEvent, processEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFont, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus, validate

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoMainImageCanvas

```
public GeoMainImageCanvas(CubeViewOp op)
```

Constructor

## Method Detail

### update

```
public void update(java.awt.Graphics g)
```

Overrides:

update in class java.awt.Component

---

### paint

```
public void paint(java.awt.Graphics g)
```

Overrides:

paint in class java.awt.Canvas

---

### getImage

```
public java.awt.Image getImage()
```

이미지 버퍼를 되돌린다.

---

## flush

```
public void flush()
```

이미지 버퍼를 Flushing한다.

---

## getBoxColor

```
public java.awt.Color getBoxColor()
```

화면의 영역을 지정하는 경계색을 되돌린다.

---

## getRegion

```
public java.awt.Rectangle getRegion()
```

화면에 표시된 영역을 되돌린다.

---

## setRegion

```
public void setRegion(java.awt.Rectangle r)
```

화면에 영역을 설정한다.

---

## drawAOIline

```
public void drawAOIline()
```

화면에 AOI의 경계선을 그린다.

---

## redrawImage

```
public void redrawImage()
```

영역부분을 다시 그린다.

---

## redrawImage

```
public void redrawImage(java.awt.Rectangle r)
```



영역부분을 다시 그린다.

---

## drawImage

```
public void drawImage(java.awt.Image img)
```

화면에 이미지를 출력한다.

---

## linkObjectLayer

```
public void linkObjectLayer(ObjectLayer layer)
```

화면에 객체 레이어를 연결한다.

---

## linkVectorLayer

```
public void linkVectorLayer(VectorLayer layer)
```

화면에 벡터 레이어를 연결한다.

---

## linkAOILayer

```
public void linkAOILayer(AOILayer layer)
```

화면에 AOI 레이어를 연결한다.

---

## drawVectorLayer

```
public void drawVectorLayer()
```

벡터 레이어를 그린다.

---

## setGrid

```
public void setGrid(int i)
```

화면에 격자망을 표시하기 위한 격자간격을 설정한다.

---

## setBoxColor

```
public void setBoxColor(java.awt.Color color)
```

선택영역의 색을 설정한다.

---

## getBoxArea

```
public java.awt.Rectangle getBoxArea()
```

선택 영역을 되돌린다.

---

## adjustBoxArea

```
public boolean adjustBoxArea(java.awt.Rectangle r)
```

선택영역의 크기를 조절한다.

---

## moveBoxArea

```
public void moveBoxArea(java.awt.Rectangle newBox)
```

박스로 표시된 영역의 이동을 관리한다.

---

## add

```
public void add(java.awt.PopupMenu menu)
```

캔버스에 팝업 메뉴를 추가한다.

Overrides:

add in class java.awt.Component

---

## processMouseEvent

```
public void processMouseEvent(java.awt.event.MouseEvent e)
```

마우스 이벤트를 받아 팝업 메뉴를 동작시킨다.

Overrides:

processMouseEvent in class java.awt.Component

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

## Class GeoMenuParser

java.lang.Object  
|  
+--GeoMenuParser

---

```
public class GeoMenuParser  
extends java.lang.Object
```

AttributeParser2 class 를 사용해서 Menu Description file 을 parsing 한다

Version:

Author:  
GeoPixel team

---

### Constructor Summary

<code>GeoMenuParser(java.io.File f)</code> file name 을 넣어서 GeoMenuParser 를 구동시킨다.
--

### Method Summary

boolean	<code>hasMoreTokens()</code> 현재 부르는 파서에게 앞으로 남은 MenuItem 이 남아 있는지를 넘겨준다 nextMenuItem() 을 무한히 looping 할때 그 조건으로 넣어주는 함수다
MenuItem	<code>nextMenuItem()</code> File 에서 MenuItem 을 하나씩 하나씩 얻어낼때 쓴다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### GeoMenuParser

```
public GeoMenuParser(java.io.File f)  
throws java.io.FileNotFoundException
```

file name 을 넣어서 GeoMenuParser 를 구동시킨다.

Parameters:

f - MenuDescription File

## Method Detail

### nextMenuToken

```
public MenuToken nextMenuToken()  
    throws IllegalFormatException,  
           java.io.IOException
```

File 에서 MenuToken 을 하나씩 하나씩 얻어낼때 쓴다.

Returns:

parse 되어 만들어진 MenuToken 을 넘긴다.

Throws:

[IllegalFormatException](#) - Menu Description file 이 잘못 되었을때  
[java.io.IOException](#) - File 과 IO 시 발생하는 error 에 대처

---

### hasMoreTokens

```
public boolean hasMoreTokens()
```

현재 부르는 파서에게 앞으로 남은 MenuToken 이 남아 있는지 를 넘겨준다  
nextMenuToken() 을 무한히 looping 할때 그 조건으로 넣어주는 함수다

Returns:

MenuToken 을 더 읽을수 있는 지를 boolean 으로 넘긴다.

---

## [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class GeoResampler

java.lang.Object  
|  
+--GeoResampler

---

```
public class GeoResampler  
extends java.lang.Object
```

출력 좌표에 대한 화소값의 추정한다.

---

Field Summary	
static int	<a href="#">B_SPLINE</a> B-Spline 법
static int	<a href="#">BILINEAR</a> 공이차법
static int	<a href="#">CUBIC</a> Cubic법
static int	<a href="#">NEAREST_NEIGHBOR</a> 최근린근사법

---

Constructor Summary	
<a href="#">GeoResampler()</a> Constructor	

Method Summary	
void	<u>BiLinear</u> (byte[][] bPixel, byte[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 바이트형 배열에 대한 BiLinear 재배열을 실행한다.
void	<u>BiLinear</u> (float[][] bPixel, float[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 실수형 배열에 대한 BiLinear 재배열을 실행한다.
void	<u>BiLinear</u> (int[][] bPixel, int[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 정수형 배열에 대한 BiLinear 재배열을 실행한다.
void	<u>BiLinear</u> (short[][] bPixel, short[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 2바이트 정수형 배열에 대한 BiLinear 재배열을 실행한다.
void	<u>BSpline</u> (byte[][] bPixel, byte[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 바이트형 배열에 대한 B-Spline 재배열을 실행한다.
void	<u>BSpline</u> (float[][] bPixel, float[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 실수형 배열에 대한 B-Spline 재배열을 실행한다.
void	<u>BSpline</u> (int[][] bPixel, int[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 정수형 배열에 대한 B-Spline 재배열을 실행한다.
void	<u>BSpline</u> (short[][] bPixel, short[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 2바이트 정수형 배열에 대한 B-Spline 재배열을 실행한다.
void	<u>Cubic</u> (byte[][] bPixel, byte[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 바이트형 배열에 대한 Cubic 재배열을 실행한다.
void	<u>Cubic</u> (float[][] bPixel, float[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 실수형 배열에 대한 Cubic 재배열을 실행한다.
void	<u>Cubic</u> (int[][] bPixel, int[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 정수형 배열에 대한 Cubic 재배열을 실행한다.
void	<u>Cubic</u> (short[][] bPixel, short[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 2바이트 정수형 배열에 대한 Cubic 재배열을 실행한다.
void	<u>NearestNeighbor</u> (byte[][] bPixel, byte[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 바이트형 배열에 대한 Nearest Neighbor 재배열을 실행한다.
void	<u>NearestNeighbor</u> (float[][] bPixel, float[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 실수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.
void	<u>NearestNeighbor</u> (int[][] bPixel, int[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 4바이트 정수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.
void	<u>NearestNeighbor</u> (short[][] bPixel, short[][] ibPixel, boolean[] isSelectedBand, float[] x, float[] y, int lns, int smps) 2바이트 정수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.
void	<u>setCubicOrder</u> (float a) Cubic법의 차수를 설정한다.
void	<u>setScale</u> (float s) 축척을 설정한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### NEAREST\_NEIGHBOR

public static final int NEAREST\_NEIGHBOR

최근린근사법

---

### BILINEAR

public static final int BILINEAR

공이차법

---

### CUBIC

public static final int CUBIC

Cubic법

---

### B\_SPLINE

public static final int B\_SPLINE

B-Spline 법

## Constructor Detail

### GeoResampler

public GeoResampler()

Constructor

## Method Detail

### setScale

public void setScale(float s)

축척을 설정한다.

---

### setCubicOrder

public void setCubicOrder(float a)

Cubic법의 차수를 설정한다.

---

## BSpline

```
public void BSpline(byte[][] bPixel,  
                    byte[][] ibPixel,  
                    boolean[] isSelectedBand,  
                    float[] x,  
                    float[] y,  
                    int lns,  
                    int smps)
```

바이트형 배열에 대한 B-Spline 재배열을 실행한다.

---

## BSpline

```
public void BSpline(short[][] bPixel,  
                    short[][] ibPixel,  
                    boolean[] isSelectedBand,  
                    float[] x,  
                    float[] y,  
                    int lns,  
                    int smps)
```

2바이트 정수형 배열에 대한 B-Spline 재배열을 실행한다.

---

## BSpline

```
public void BSpline(int[][] bPixel,  
                    int[][] ibPixel,  
                    boolean[] isSelectedBand,  
                    float[] x,  
                    float[] y,  
                    int lns,  
                    int smps)
```

4바이트 정수형 배열에 대한 B-Spline 재배열을 실행한다.

---

## BSpline

```
public void BSpline(float[][] bPixel,  
                    float[][] ibPixel,  
                    boolean[] isSelectedBand,  
                    float[] x,  
                    float[] y,  
                    int lns,  
                    int smps)
```

4바이트 실수형 배열에 대한 B-Spline 재배열을 실행한다.

---

## Cubic



```
public void Cubic(byte[][] bPixel,
                 byte[][] ibPixel,
                 boolean[] isSelectedBand,
                 float[] x,
                 float[] y,
                 int lns,
                 int smps)
```

바이트형 배열에 대한 Cubic 재배열을 실행한다.

---

## Cubic

```
public void Cubic(short[][] bPixel,
                 short[][] ibPixel,
                 boolean[] isSelectedBand,
                 float[] x,
                 float[] y,
                 int lns,
                 int smps)
```

2바이트 정수형 배열에 대한 Cubic 재배열을 실행한다.

---

## Cubic

```
public void Cubic(int[][] bPixel,
                 int[][] ibPixel,
                 boolean[] isSelectedBand,
                 float[] x,
                 float[] y,
                 int lns,
                 int smps)
```

4바이트 정수형 배열에 대한 Cubic 재배열을 실행한다.

---

## Cubic

```
public void Cubic(float[][] bPixel,
                 float[][] ibPixel,
                 boolean[] isSelectedBand,
                 float[] x,
                 float[] y,
                 int lns,
                 int smps)
```

4바이트 실수형 배열에 대한 Cubic 재배열을 실행한다.

---

## BiLinear

```
public void BiLinear(byte[][] bPixel,
                   byte[][] ibPixel,
                   boolean[] isSelectedBand,
                   float[] x,
                   float[] y,
                   int lns,
                   int smps)
```

바이트형 배열에 대한 BiLinear 재배열을 실행한다.

---

## BiLinear

```
public void BiLinear(short[][] bPixel,
                    short[][] ibPixel,
                    boolean[] isSelectedBand,
                    float[] x,
                    float[] y,
                    int lns,
                    int smps)
```

2바이트 정수형 배열에 대한 BiLinear 재배열을 실행한다.

---

## BiLinear

```
public void BiLinear(int[][] bPixel,
                    int[][] ibPixel,
                    boolean[] isSelectedBand,
                    float[] x,
                    float[] y,
                    int lns,
                    int smps)
```

4바이트 정수형 배열에 대한 BiLinear 재배열을 실행한다.

---

## BiLinear

```
public void BiLinear(float[][] bPixel,
                    float[][] ibPixel,
                    boolean[] isSelectedBand,
                    float[] x,
                    float[] y,
                    int lns,
                    int smps)
```

4바이트 실수형 배열에 대한 BiLinear 재배열을 실행한다.

---

## NearestNeighbor

```
public void NearestNeighbor(byte[][] bPixel,
                            byte[][] ibPixel,
                            boolean[] isSelectedBand,
                            float[] x,
                            float[] y,
                            int lns,
                            int smps)
```

바이트형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

---

## NearestNeighbor

```
public void NearestNeighbor(short[][] bPixel,
                            short[][] ibPixel,
```

```
boolean[] isSelectedBand,  
float[] x,  
float[] y,  
int lns,  
int smps)
```

2바이트 정수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

---

## NearestNeighbor

```
public void NearestNeighbor(int[][] bPixel,  
int[][] ibPixel,  
boolean[] isSelectedBand,  
float[] x,  
float[] y,  
int lns,  
int smps)
```

4바이트 정수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

---

## NearestNeighbor

```
public void NearestNeighbor(float[][] bPixel,  
float[][] ibPixel,  
boolean[] isSelectedBand,  
float[] x,  
float[] y,  
int lns,  
int smps)
```

4바이트 실수형 배열에 대한 Nearest Neighbor 재배열을 실행한다.

---

### [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY: INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class GeoScrollImageCanvas

```
java.lang.Object
|
+--java.awt.Component
|   |
|   +--java.awt.Canvas
|       |
|       +--GeoScrollImageCanvas
```

---

```
public class GeoScrollImageCanvas
extends java.awt.Canvas
```

스크롤 영상 화면을 관리한다.

See Also:

[Serialized Form](#)

---

### Fields inherited from class java.awt.Component

[BOTTOM\\_ALIGNMENT](#), [CENTER\\_ALIGNMENT](#), [LEFT\\_ALIGNMENT](#), [RIGHT\\_ALIGNMENT](#), [TOP\\_ALIGNMENT](#)

### Constructor Summary

<a href="#">GeoScrollImageCanvas</a> (CubeViewOp op)
Constructor

## Method Summary

boolean	<u>adjustBoxArea</u> (java.awt.Rectangle r) 선택영역의 크기를 조절한다.
void	<u>drawAOILine</u> () 화면에 AOI의 경계선을 그린다.
void	<u>drawImage</u> (java.awt.Image img) 화면에 이미지를 출력한다.
void	<u>drawVectorLayer</u> () 벡터 레이어를 그린다.
void	<u>flush</u> () 이미지 버퍼를 Flushing한다.
java.awt.Rectangle	<u>getBoxArea</u> () 선택 영역을 되돌린다.
java.awt.Image	<u>getImage</u> () 이미지 버퍼를 되돌린다.
float	<u>getScale</u> () 스크를 화면 구성에 대한 축척을 되돌린다.
void	<u>linkAOILayer</u> (AOILayer layer) 화면에 AOI 레이어를 연결한다.
void	<u>linkObjectLayer</u> (ObjectLayer layer) 화면에 객체 레이어를 연결한다.
void	<u>linkVectorLayer</u> (VectorLayer layer) 화면에 벡터 레이어를 연결한다.
void	<u>moveBoxArea</u> (java.awt.Rectangle newBox) 박스로 표시된 영역의 이동을 관리한다.
void	<u>paint</u> (java.awt.Graphics g)
void	<u>redrawImage</u> () 영역부분을 다시 그린다.
void	<u>redrawImage</u> (java.awt.Rectangle r) 영역부분을 다시 그린다.
void	<u>setBoxColor</u> (java.awt.Color color) 선택영역의 색을 설정한다.
void	<u>setScale</u> (float scale) 스크를 화면 구성에 대한 축척을 설정한다.
void	<u>update</u> (java.awt.Graphics g)

## Methods inherited from class java.awt.Canvas

addNotify

## Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getName, getParent, getPeer, getPreferredSize, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, paramString, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, processComponentEvent, processEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFont, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus, validate

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoScrollImageCanvas

```
public GeoScrollImageCanvas(CubeViewOp op)
```

Constructor

## Method Detail

### update

```
public void update(java.awt.Graphics g)
```

Overrides:

update in class java.awt.Component

---

### paint

```
public void paint(java.awt.Graphics g)
```

Overrides:

paint in class java.awt.Canvas

---

### getImage

```
public java.awt.Image getImage()
```

이미지 버퍼를 되돌린다.

---

## **flush**

```
public void flush()
```

이미지 버퍼를 Flushing한다.

---

## **drawAOIline**

```
public void drawAOIline()
```

화면에 AOI의 경계선을 그린다.

---

## **drawImage**

```
public void drawImage(java.awt.Image img)
```

화면에 이미지를 출력한다.

---

## **redrawImage**

```
public void redrawImage(java.awt.Rectangle r)
```

영역부분을 다시 그린다.

---

## **redrawImage**

```
public void redrawImage()
```

영역부분을 다시 그린다.

---

## **getScale**

```
public float getScale()
```

스크롤 화면 구성에 대한 축척을 되돌린다.

---

## **setScale**

```
public void setScale(float scale)
```

스크롤 화면 구성에 대한 축척을 설정한다.

---

## linkVectorLayer

```
public void linkVectorLayer(VectorLayer layer)
```

화면에 벡터 레이어를 연결한다.

---

## linkObjectLayer

```
public void linkObjectLayer(ObjectLayer layer)
```

화면에 객체 레이어를 연결한다.

---

## drawVectorLayer

```
public void drawVectorLayer()
```

벡터 레이어를 그린다.

---

## linkAOILayer

```
public void linkAOILayer(AOILayer layer)
```

화면에 AOI 레이어를 연결한다.

---

## setBoxColor

```
public void setBoxColor(java.awt.Color color)
```

선택영역의 색을 설정한다.

---

## getBoxArea

```
public java.awt.Rectangle getBoxArea()
```

선택 영역을 되돌린다.

---

## adjustBoxArea

```
public boolean adjustBoxArea(java.awt.Rectangle r)
```

선택영역의 크기를 조절한다.

---



## moveBoxArea

```
public void moveBoxArea(java.awt.Rectangle newBox)
```

박스로 표시된 영역의 이동을 관리한다.

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree Deprecated Index Help

PREV CLASS    NEXT CLASS  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#)    [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class GeoTargetTree

```
java.lang.Object
|
+--java.awt.Component
   |
   +--java.awt.Container
      |
      +--com.sun.java.swing.JComponent
         |
         +--com.sun.java.swing.JTree
            |
            +--GeoTargetTree
```

---

```
public class GeoTargetTree
extends com.sun.java.swing.JTree
```

영상 관리자를 관리한다.

See Also:  
[Serialized Form](#)

---

### Inner classes inherited from class com.sun.java.swing.JTree

```
com.sun.java.swing.JTree.AccessibleJTree,
com.sun.java.swing.JTree.DynamicUtilTreeNode,
com.sun.java.swing.JTree.EmptySelectionModel,
com.sun.java.swing.JTree.TreeSelectionRedirector
```

### Inner classes inherited from class com.sun.java.swing.JComponent

```
com.sun.java.swing.JComponent.AccessibleJComponent
```

### Fields inherited from class com.sun.java.swing.JTree

```
CELL_EDITOR_PROPERTY, CELL_RENDERER_PROPERTY, cellEditor, cellRenderer, editable,
EDITABLE_PROPERTY, INVOKES_STOP_CELL_EDITING_PROPERTY, invokesStopCellEditing,
LARGE_MODEL_PROPERTY, largeModel, ROOT_VISIBLE_PROPERTY, rootVisible, ROW_HEIGHT_PROPERTY,
rowHeight, SELECTION_MODEL_PROPERTY, selectionModel, selectionRedirector,
SHOWS_ROOT_HANDLES_PROPERTY, showsRootHandles, TREE_MODEL_PROPERTY, treeModel,
VISIBLE_ROW_COUNT_PROPERTY, visibleRowCount
```

### Fields inherited from class com.sun.java.swing.JComponent

```
accessibleContext, listenerList, TOOL_TIP_TEXT_KEY, ui, UNDEFINED_CONDITION,
WHEN_ANCESTOR_OF_FOCUSED_COMPONENT, WHEN_FOCUSED, WHEN_IN_FOCUSED_WINDOW
```

### Fields inherited from class java.awt.Component

```
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT
```

## Constructor Summary

`GeoTargetTree()`  
Constructor

`GeoTargetTree(java.lang.String title)`  
Constructor

## Method Summary

void	<code>deleteAllTargets()</code> 모든 영상을 삭제한다.
void	<code>deleteSelectedTarget()</code> 선택된 영상을 삭제한다.
void	<code>deleteTarget(Target tgt)</code> 영상을 목록에서 제거한다.
boolean	<code>existTarget(Target tgt)</code> 선택된 영상이 이미 존재하는지를 검사한다.
int	<code>getSelectedBandIndex()</code> 선택된 영상의 분광대 인덱스를 되돌린다.
int	<code>getSelectedCubeIndex()</code> 선택된 영상의 인덱스를 되돌린다.
Target	<code>getSelectedTarget()</code> 선택된 영상을 되돌린다.
boolean	<code>isBandSelected()</code> 선택된 영상의 모드를 검사한다.
boolean	<code>isCubeSelected()</code> 선택된 영상의 모드를 검사한다.
boolean	<code>isEmpty()</code> 영상목록이 비어 있는지를 검사한다.
void	<code>pushTarget(Target tgt)</code> 영상목록에 추가한다.

## Methods inherited from class com.sun.java.swing.JTree

`addSelectionInterval`, `addSelectionPath`, `addSelectionPaths`, `addSelectionRow`,  
`addSelectionRows`, `addTreeExpansionListener`, `addTreeSelectionListener`, `clearSelection`,  
`collapsePath`, `collapseRow`, `convertValueToText`, `createTreeModel`, `expandPath`, `expandRow`,  
`fireTreeCollapsed`, `fireTreeExpanded`, `fireValueChanged`, `getAccessibleContext`, `getCellEditor`,  
`getCellRenderer`, `getClosestPathForLocation`, `getClosestRowForLocation`, `getDefaultTreeModel`,  
`getEditingPath`, `getInvokesStopCellEditing`, `getLastSelectedPathComponent`,  
`getLeadSelectionPath`, `getLeadSelectionRow`, `getMaxSelectionRow`, `getMinSelectionRow`,  
`getModel`, `getPathBetweenRows`, `getPathBounds`, `getPathForLocation`, `getPathForRow`,  
`getPreferredScrollableViewportSize`, `getRowBounds`, `rowCount`, `getRowForLocation`,  
`getRowForPath`, `getRowHeight`, `getScrollableBlockIncrement`,  
`getScrollableTracksViewportHeight`, `getScrollableTracksViewportWidth`,  
`getScrollableUnitIncrement`, `getSelectionCount`, `getSelectionModel`, `getSelectionPath`,  
`getSelectionPaths`, `getSelectionRows`, `getShowsRootHandles`, `getToolTipText`, `getUI`,  
`getUIClassID`, `getVisibleRowCount`, `isCollapsed`, `isExpanded`, `isEditable`, `isEditing`,  
`isExpanded`, `isExpanded`, `isFixedRowHeight`, `isLargeModel`, `isOpaque`, `isPathEditable`,  
`isPathSelected`, `isRootVisible`, `isRowSelected`, `isSelectionEmpty`, `isVisible`, `makeVisible`,  
`removeSelectionInterval`, `removeSelectionPath`, `removeSelectionPaths`, `removeSelectionRow`,  
`removeSelectionRows`, `removeTreeExpansionListener`, `removeTreeSelectionListener`,  
`scrollPathToVisible`, `scrollRowToVisible`, `setCellEditor`, `setCellRenderer`, `setEditable`,  
`setInvokesStopCellEditing`, `setLargeModel`, `setModel`, `setRootVisible`, `setRowHeight`,  
`setSelectionInterval`, `setSelectionModel`, `setSelectionPath`, `setSelectionPaths`,  
`setSelectionRow`, `setSelectionRows`, `setShowsRootHandles`, `setUI`, `setVisibleRowCount`,  
`startEditingAtPath`, `stopEditing`, `treeDidChange`, `updateUI`

## Methods inherited from class com.sun.java.swing.JComponent

addAncestorListener, addNotify, addPropertyChangeListener, addVetoableChangeListener, computeVisibleRect, contains, createToolTip, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, firePropertyChange, fireVetoableChange, getActionForKeyStroke, getAlignmentX, getAlignmentY, getAutoscrolls, getBorder, getBounds, getClientProperty, getComponentGraphics, getConditionForKeyStroke, getDebugGraphicsOptions, getGraphics, getHeight, getInsets, getLocation, getMaximumSize, getMinimumSize, getNextFocusableComponent, getPreferredSize, getRegisteredKeyStrokes, getRootPane, getSize, getToolTipLocation, getToolTipText, getTopLevelAncestor, getVisibleRect, getWidth, getX, getY, grabFocus, hasFocus, isDoubleBuffered, isFocusCycleRoot, isFocusTraversable, isLightweightComponent, isManagingFocus, isOptimizedDrawingEnabled, isPaintingTile, isRequestFocusEnabled, isValidRoot, paint, paintBorder, paintChildren, paintComponent, paintImmediately, paintImmediately, processComponentKeyEvent, processFocusEvent, processKeyEvent, processMouseEvent, putClientProperty, registerKeyboardAction, registerKeyboardAction, removeAncestorListener, removeNotify, removePropertyChangeListener, removeVetoableChangeListener, repaint, repaint, requestDefaultFocus, requestFocus, resetKeyboardActions, reshape, revalidate, scrollRectToVisible, setAlignmentX, setAlignmentY, setAutoscrolls, setBorder, setBounds, setDebugGraphicsOptions, setDoubleBuffered, setMaximumSize, setMinimumSize, setNextFocusableComponent, setOpaque, setPreferredSize, setRequestFocusEnabled, setToolTipText, setUI, setVisible, unregisterKeyboardAction, update

## Methods inherited from class java.awt.Container

add, add, add, add, add, addContainerListener, addImpl, countComponents, deliverEvent, doLayout, findComponentAt, findComponentAt, getComponent, getComponentAt, getComponentAt, getComponentCount, getComponents, getLayout, insets, invalidate, isAncestorOf, layout, list, list, locate, minimumSize, paintComponents, paramString, preferredSize, print, printComponents, processContainerEvent, processEvent, remove, remove, removeAll, removeContainerListener, setFont, setLayout, validate, validateTree

## Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, createImage, createImage, disable, disableEvents, dispatchEvent, enable, enable, enableEvents, enableInputMethods, getBackground, getBounds, getColorModel, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocationOnScreen, getName, getParent, getPeer, getSize, getToolkit, getTreeLock, gotFocus, handleEvent, hide, imageUpdate, inside, isDisplayable, isEnabled, isLightweight, isShowing, isValid, isVisible, keyDown, keyUp, list, list, list, location, lostFocus, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, postEvent, prepareImage, prepareImage, printAll, processComponentEvent, processInputMethodEvent, processMouseEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removePropertyChangeListener, repaint, repaint, repaint, resize, resize, setBackground, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, show, show, size, toString, transferFocus

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoTargetTree

```
public GeoTargetTree()
```

Constructor

## GeoTargetTree

```
public GeoTargetTree(java.lang.String title)
```

Constructor

### Method Detail

#### pushTarget

```
public void pushTarget(Target tgt)
```

영상목록에 추가한다.

---

#### deleteTarget

```
public void deleteTarget(Target tgt)
```

영상을 목록에서 제거한다.

---

#### getSelectedTarget

```
public Target getSelectedTarget()
```

선택된 영상을 되돌린다.

---

#### getSelectedCubeIndex

```
public int getSelectedCubeIndex()
```

선택된 영상의 인덱스를 되돌린다.

---

#### getSelectedBandIndex

```
public int getSelectedBandIndex()
```

선택된 영상의 분광대 인덱스를 되돌린다.

---

#### isBandSelected

```
public boolean isBandSelected()
```

선택된 영상의 모드를 검사한다. 분광대가 선택된 경우 true를 되돌린다.

---

#### isCubeSelected

```
public boolean isCubeSelected()
```

선택된 영상의 모드를 검사한다. 영상이 선택된 경우 true를 되돌린다.

---

## existTarget

```
public boolean existTarget(Target tgt)
```

선택된 영상이 이미 존재하는지를 검사한다. 존재하는 경우 true를 되돌린다.

---

## deleteAllTargets

```
public void deleteAllTargets()
```

모든 영상을 삭제한다.

---

## deleteSelectedTarget

```
public void deleteSelectedTarget()
```

선택된 영상을 삭제한다.

---

## isEmpty

```
public boolean isEmpty()
```

영상목록이 비어 있는지를 검사한다.

---

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

## Class GeoUtility

java.lang.Object  
|  
+--GeoUtility

---

public class GeoUtility  
extends java.lang.Object

응용 클래스를 관리한다.

---

### Constructor Summary

<a href="#">GeoUtility()</a>	
------------------------------	--

### Method Summary

static void	<a href="#">collectAllGarbages()</a> 시스템의 카비지 콜렉터를 실행한다.
static java.lang.String	<a href="#">getExtendedFilename(java.lang.String base, java.lang.String ext)</a> 파일의 확장자를 확인하여 유효한 파일명을 되돌린다.
static java.lang.String	<a href="#">trimStringForNumeric(java.lang.String s)</a> 문자열을 유효 숫자로 변환한다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### GeoUtility

public GeoUtility()

### Method Detail

#### collectAllGarbages

public static void collectAllGarbages()

시스템의 카비지 콜렉터를 실행한다.

---

## trimStringForNumeric

```
public static java.lang.String trimStringForNumeric(java.lang.String s)
```

문자열을 유효 숫자로 변환한다.

---

## getExtendedFilename

```
public static java.lang.String getExtendedFilename(java.lang.String base,  
                                                    java.lang.String ext)
```

파일의 확장자를 확인하여 유효한 파일명을 되돌린다.

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class GeoVector

```
java.lang.Object
|
+--GeoVector
```

```
public class GeoVector
extends java.lang.Object
```

벡터 클래스를 정의한다.

Constructor Summary	
<code>GeoVector()</code>	Constructor
<code>GeoVector(java.lang.String title)</code>	Constructor

Method Summary	
void	<code>addFeature(<a href="#">Feature</a> f)</code> feature를 추가한다.
void	<code>clear()</code> Feature를 모두 목록에서 제거한다.
int	<code>count()</code> 목록의 Feature의 수를 되돌린다.
java.util.Enumeration	<code>elements()</code> 벡터 element를 되돌린다.
<a href="#">Feature</a>	<code>getFeature(int index)</code> 인덱스에 해당하는 feature를 되돌린다.
<a href="#">Feature[]</a>	<code>getFeatures()</code> Feature배열을 되돌린다.
<a href="#">Feature.Point</a>	<code>getMaxLimit()</code> point feature의 최대 영역 범위를 되돌린다.
<a href="#">Feature.Point</a>	<code>getMinLimit()</code> point feature의 최소 영역 범위를 되돌린다.
java.lang.String	<code>getTitle()</code> 벡터의 타이틀을 되돌린다.
int	<code>indexOf(<a href="#">Feature</a> f)</code> Feature의 인덱스를 되돌린다.
void	<code>init()</code>
void	<code>insertFeature(int index, <a href="#">Feature</a> f)</code>

	feature를 삽입한다.
void	<u>read</u> (java.io.File f) Feature를 파일에서 입력한다.
void	<u>read</u> (java.lang.String path) Feature를 파일에서 입력한다.
void	<u>removeFeature</u> (double x, double y) x, y 좌표에 있는 feature를 삭제한다.
void	<u>removeFeature</u> (Feature f) feature를 삭제한다.
void	<u>removeFeature</u> (int index) feature를 삭제한다.
void	<u>setLimit</u> (double x1, double x2, double y1, double y2) 두 point의 최소-최대 좌표를 설정한다.
void	<u>setLimit</u> (Feature.Point min, Feature.Point max) 두 point feature의 최소-최대 좌표를 설정한다.
void	<u>setMaxLimit</u> (double x, double y) 최대 x, y좌표를 설정한다.
void	<u>setMaxLimit</u> (Feature.Point p) point feature의 최대 좌표를 설정한다.
void	<u>setMinLimit</u> (double x, double y) 최소 x, y좌표를 설정한다.
void	<u>setMinLimit</u> (Feature.Point p) point feature의 최소 좌표를 설정한다.
void	<u>setTitle</u> (java.lang.String title) 타이틀을 설정한다.
java.lang.String	<u>toString</u> ()
void	<u>write</u> (java.io.File f) Feature를 파일로 출력한다.
void	<u>write</u> (java.lang.String path) Feature를 파일로 출력한다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoVector

public GeoVector()

Constructor

### GeoVector

public GeoVector(java.lang.String title)

Constructor

## Method Detail

### getTitle

```
public java.lang.String getTitle()
```

벡터의 타이틀을 되돌린다.

---

### setTitle

```
public void setTitle(java.lang.String title)
```

타이틀을 설정한다.

---

### getMinLimit

```
public Feature.Point getMinLimit()
```

point feature의 최소 영역 범위를 되돌린다.

---

### getMaxLimit

```
public Feature.Point getMaxLimit()
```

point feature의 최대 영역 범위를 되돌린다.

---

### setMinLimit

```
public void setMinLimit(double x,  
                        double y)
```

최소 x, y좌표를 설정한다.

---

### setMinLimit

```
public void setMinLimit(Feature.Point p)
```

point feature의 최소 좌표를 설정한다.

---

### setMaxLimit

```
public void setMaxLimit(double x,  
                        double y)
```

최대 x, y좌표를 설정한다.

---

## setMaxLimit

```
public void setMaxLimit(Feature.Point p)
```

point feature의 최대 좌표를 설정한다.

---

## setLimit

```
public void setLimit(Feature.Point min,  
                    Feature.Point max)
```

두 point feature의 최소-최대 좌표를 설정한다.

---

## setLimit

```
public void setLimit(double x1,  
                    double x2,  
                    double y1,  
                    double y2)
```

두 point의 최소-최대 좌표를 설정한다.

---

## addFeature

```
public void addFeature(Feature f)
```

feature를 추가한다.

---

## insertFeature

```
public void insertFeature(int index,  
                          Feature f)
```

feature를 삽입한다.

---

## removeFeature

```
public void removeFeature(int index)
```

feature를 삭제한다.

---

## removeFeature

```
public void removeFeature(double x,
```

double y)

x, y 좌표에 있는 feature를 삭제한다.

---

## removeFeature

public void removeFeature(Feature f)

feature를 삭제한다.

---

## elements

public java.util.Enumeration elements()

백터 element를 되돌린다.

---

## getFeatures

public Feature[] getFeatures()

Feature배열을 되돌린다.

---

## getFeature

public Feature getFeature(int index)

인덱스에 해당하는 feature를 되돌린다.

---

## indexOf

public int indexOf(Feature f)

Feature의 인덱스를 되돌린다.

---

## count

public int count()

목록의 Feature의 수를 되돌린다.

---

## clear

public void clear()

Feature를 모두 목록에서 제거한다.

---

## read

```
public void read(java.lang.String path)
    throws java.io.IOException,
           IllegalFormatException
```

Feature를 파일에서 입력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException

---

## read

```
public void read(java.io.File f)
    throws java.io.FileNotFoundException,
           java.io.IOException,
           IllegalFormatException
```

Feature를 파일에서 입력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException  
java.io.FileNotFoundException – FileNotFoundException

---

## write

```
public void write(java.lang.String path)
    throws java.io.IOException
```

Feature를 파일로 출력한다.

Throws:

java.io.IOException – IOException

---

## write

```
public void write(java.io.File f)
    throws java.io.IOException
```

Feature를 파일로 출력한다.

Throws:

java.io.IOException – IOException

---

## init

```
public void init()
```

---

## toString

public java.lang.String toString()

Overrides:  
toString in class java.lang.Object

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

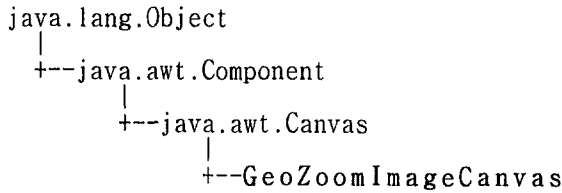
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class GeoZoomImageCanvas



```

public class GeoZoomImageCanvas
extends java.awt.Canvas
    
```

확대 영상 화면을 관리한다.

See Also:  
[Serialized Form](#)

Fields inherited from class java.awt.Component
BOTTOM_ALIGNMENT, CENTER_ALIGNMENT, LEFT_ALIGNMENT, RIGHT_ALIGNMENT, TOP_ALIGNMENT

Constructor Summary	
<code>GeoZoomImageCanvas(CubeViewOp op)</code>	Constructor

Method Summary	
void	<code>drawImage(java.awt.Image img, java.awt.Rectangle r)</code> 이미지 버퍼의 영역을 화면창에 출력한다.
java.awt.Image	<code>getImage()</code> 이미지 버퍼를 되돌린다.
int	<code>getScale()</code> 확대 화면 구성에 대한 축척을 되돌린다.
void	<code>paint(java.awt.Graphics g)</code>
void	<code>setCrossColor(java.awt.Color color)</code> 화면의 cross hair에 대한 색을 설정한다.
void	<code>setCrossPivot(int x, int y)</code> 화면의 cross hair의 위치를 설정한다.
void	<code>setScale(int scale)</code> 확대 화면 구성에 대한 축척을 설정한다.
void	<code>update(java.awt.Graphics g)</code>



## Methods inherited from class java.awt.Canvas

addNotify

## Methods inherited from class java.awt.Component

action, add, addComponentListener, addFocusListener, addInputMethodListener, addKeyListener, addMouseListener, addMouseMotionListener, addPropertyChangeListener, addPropertyChangeListener, bounds, checkImage, checkImage, coalesceEvents, contains, contains, createImage, createImage, deliverEvent, disable, disableEvents, dispatchEvent, doLayout, enable, enable, enableEvents, enableInputMethods, firePropertyChange, getAlignmentX, getAlignmentY, getBackground, getBounds, getBounds, getColorModel, getComponentAt, getComponentAt, getComponentOrientation, getCursor, getDropTarget, getFont, getFontMetrics, getForeground, getGraphics, getHeight, getInputContext, getInputMethodRequests, getLocale, getLocation, getLocation, getLocationOnScreen, getMaximumSize, getMinimumSize, getName, getParent, getPeer, getPreferredSize, getSize, getSize, getToolkit, getTreeLock, getWidth, getX, getY, gotFocus, handleEvent, hasFocus, hide, imageUpdate, inside, invalidate, isDisplayable, isDoubleBuffered, isEnabled, isFocusTraversable, isLightweight, isOpaque, isShowing, isValid, isVisible, keyDown, keyUp, layout, list, list, list, list, list, locate, location, lostFocus, minimumSize, mouseDown, mouseDrag, mouseEnter, mouseExit, mouseMove, mouseUp, move, nextFocus, paintAll, paramString, postEvent, preferredSize, prepareImage, prepareImage, print, printAll, processComponentEvent, processEvent, processFocusEvent, processInputMethodEvent, processKeyEvent, processMouseEvent, processMouseMotionEvent, remove, removeComponentListener, removeFocusListener, removeInputMethodListener, removeKeyListener, removeMouseListener, removeMouseMotionListener, removeNotify, removePropertyChangeListener, removePropertyChangeListener, repaint, repaint, repaint, repaint, requestFocus, reshape, resize, resize, setBackground, setBounds, setBounds, setComponentOrientation, setCursor, setDropTarget, setEnabled, setFont, setForeground, setLocale, setLocation, setLocation, setName, setSize, setSize, setVisible, show, show, size, toString, transferFocus, validate

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### GeoZoomImageCanvas

```
public GeoZoomImageCanvas(CubeViewOp op)
```

Constructor

## Method Detail

### update

```
public void update(java.awt.Graphics g)
```

Overrides:

update in class java.awt.Component

---

### paint

```
public void paint(java.awt.Graphics g)
```

Overrides:

paint in class java.awt.Canvas

---

## getImage

```
public java.awt.Image getImage()
```

이미지 버퍼를 되돌린다.

---

## setScale

```
public void setScale(int scale)
```

확대 화면 구성에 대한 축척을 설정한다.

---

## getScale

```
public int getScale()
```

확대 화면 구성에 대한 축척을 되돌린다.

---

## setCrossColor

```
public void setCrossColor(java.awt.Color color)
```

화면의 cross hair에 대한 색을 설정한다.

---

## setCrossPivot

```
public void setCrossPivot(int x,  
                          int y)
```

화면의 cross hair의 위치를 설정한다.

---

## drawImage

```
public void drawImage(java.awt.Image img,  
                      java.awt.Rectangle r)
```

이미지 버퍼의 영역을 화면창에 출력한다.

---

[Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class HistogramMatchOp

java.lang.Object

└--HistogramMatchOp

public class HistogramMatchOp  
extends java.lang.Object  
implements [Operator](#)

두개의 영상에 대한 히스토그램 일치화를 실행한다. 이 작업은 가급적 히스토그램의 분포 모양이 유사한 동일 분광대에 대하여 동일 계열에 획득된 영상을 이용하여야 효과가 있다.

## Constructor Summary

[HistogramMatchOp\(\)](#)  
Constructor

[HistogramMatchOp\(int\[\] t, int\[\] s\)](#)  
Constructor

## Method Summary

void [operate\(\)](#)  
히스토그램 일치화를 실행한다.

void [setBand\(int b\)](#)  
처리하고자 하는 분광대를 설정한다.

void [setBounds\(java.awt.Rectangle r\)](#)  
처리 영역을 설정한다.

void [setHistogram\(int\[\] t, int\[\] s, int n\)](#)  
히스토그램을 설정한다.

void [setProgressStream\(ProgressStream prog\)](#)  
프로그램 진행 상태바를 설정한다.

void [setReferenceHistogram\(int\[\] s\)](#)  
참조 히스토그램을 설정한다.

void [setTarget\(Target stgt, Target dtgt\)](#)  
대상 영상을 설정한다.

void [setTargetHistogram\(int\[\] t\)](#)  
목표 히스토그램을 설정한다.

void [setTotalPixel\(int n\)](#)  
총 화소수를 설정한다.

void [stop\(\)](#)  
사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### HistogramMatchOp

```
public HistogramMatchOp()
```

Constructor

---

### HistogramMatchOp

```
public HistogramMatchOp(int[] t,  
                        int[] s)
```

Constructor

## Method Detail

### setBand

```
public void setBand(int b)
```

처리하고자 하는 분광대를 설정한다.

---

### setHistogram

```
public void setHistogram(int[] t,  
                        int[] s,  
                        int n)
```

히스토그램을 설정한다.

---

### setTotalPixel

```
public void setTotalPixel(int n)
```

총 화소수를 설정한다.

---

### setTargetHistogram

```
public void setTargetHistogram(int[] t)
```

목표 히스토그램을 설정한다.

---

### setReferenceHistogram

```
public void setReferenceHistogram(int[] s)
```

참조 히스토그램을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

히스토그램 일치화를 실행한다.

Throws:

[java.io.IOException](#) - [IOException](#)

[java.lang.InterruptedException](#) - [InterruptedException](#)

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class HistogramModifyEnhanceOp

java.lang.Object

|--HistogramModifyEnhanceOp

---

public class HistogramModifyEnhanceOp  
extends java.lang.Object  
implements [Operator](#)

농도 분포 변형을 통한 영상향상을 실행한다.

---

Field Summary	
float[]	<a href="#">cumFreq</a>
int	<a href="#">Intervals</a>
float	<a href="#">Start</a>
int	<a href="#">totalFreq</a>
float	<a href="#">Width</a>

Constructor Summary	
<a href="#">HistogramModifyEnhanceOp()</a>	Constructor

## Method Summary

protected void	<u>changePixelValue</u> (byte[] a, int aOffset, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (float[] a, int aOffset, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (int[] a, int aOffset, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (short[] a, int aOffset, byte[] b, int bOffset, int length)
void	<u>operate</u> () 히스토그램 변형에 의한 영상향상을 실행한다.
void	<u>regionByEnd</u> (float start, float end, int intervals) 히스토그램 작성을 위한 농도 범위를 설정한다.
void	<u>regionByWidth</u> (float start, float width, int intervals) 히스토그램 작성을 위한 농도 범위를 설정한다.
void	<u>setAlpha</u> (float a) 히스토그램의 변형에 필요한 alpha값을 설정한다.
void	<u>setBand</u> (int band) 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<u>setCumulHistogram</u> (int total, int[] cf) 누적 히스토그램을 설정한다.
void	<u>setMethod</u> (java.lang.String method) 히스토그램 변형법을 설정한다.
void	<u>setOutputRange</u> (int mi, int mx) 히스토그램 변경 결과에 대한 영역을 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setStatistic</u> (float m, float st) 히스토그램 변경에 필요한 통계를 설정한다.
void	<u>setTarget</u> (Target sTgt, Target dTgt) 대상 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### cumFreq

public float[] cumFreq

### totalFreq

```
public int totalFreq
```

---

## Start

```
public float Start
```

---

## Width

```
public float Width
```

---

## Intervals

```
public int Intervals
```

## Constructor Detail

### HistogramModifyEnhanceOp

```
public HistogramModifyEnhanceOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

히스토그램 변형법을 설정한다.

---

### setBand

```
public void setBand(int band)
```

대상 분광대를 설정한다.

---

### setAlpha

```
public void setAlpha(float a)
```

히스토그램의 변형에 필요한 alpha값을 설정한다.

---

### setOutputRange

```
public void setOutputRange(int mi,  
                           int mx)
```

히스토그램 변경 결과에 대한 영역을 설정한다.

---



## setStatistic

```
public void setStatistic(float m,  
                        float st)
```

히스토그램 변경에 필요한 통계를 설정한다.

---

## setCumulHistogram

```
public void setCumulHistogram(int total,  
                              int[] cf)
```

누적 히스토그램을 설정한다.

---

## regionByEnd

```
public void regionByEnd(float start,  
                       float end,  
                       int intervals)
```

히스토그램 작성을 위한 농도 범위를 설정한다.

---

## regionByWidth

```
public void regionByWidth(float start,  
                          float width,  
                          int intervals)
```

히스토그램 작성을 위한 농도 범위를 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target sTgt,  
                    Target dTgt)
```

대상 영상을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

히스토그램 변형에 의한 영상향상을 실행한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## changePixelValue

```
protected void changePixelValue(byte[] a,  
                                 int aOffset,  
                                 byte[] b,  
                                 int bOffset,  
                                 int length)
```

---

## changePixelValue

```
protected void changePixelValue(short[] a,  
                                 int aOffset,  
                                 byte[] b,  
                                 int bOffset,  
                                 int length)
```

---

## changePixelValue

```
protected void changePixelValue(int[] a,  
                                 int aOffset,  
                                 byte[] b,  
                                 int bOffset,  
                                 int length)
```

---

## changePixelValue

```
protected void changePixelValue(float[] a,  
                                 int aOffset,  
                                 byte[] b,  
                                 int bOffset,  
                                 int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

## Class HistogramOp

```
java.lang.Object
|
+--HistogramOp
```

```
public class HistogramOp
extends java.lang.Object
implements Operator
```

영상의 히스토그램을 작성한다.

Constructor Summary	
<a href="#">HistogramOp()</a>	Constructor

Method Summary	
protected void	<a href="#">calculateFrequency</a> (int bandIndex, byte[] pix, int aOffset, int length)
protected void	<a href="#">calculateFrequency</a> (int bandIndex, float[] pix, int aOffset, int length)
protected void	<a href="#">calculateFrequency</a> (int bandIndex, int[] pix, int aOffset, int length)
protected void	<a href="#">calculateFrequency</a> (int bandIndex, short[] pix, int aOffset, int length)
void	<a href="#">clear</a> () 히스토그램을 초기화한다.
int[]	<a href="#">getCumulativeFreq</a> () 지정된 분광대에 대한 누적 농도 분포를 되돌린다.
int	<a href="#">getCumulativeFreq</a> (int index) 지정된 분광대의 화소값에 대한 누적 농도 분포를 되돌린다.
int[]	<a href="#">getFreq</a> () 지정된 인덱스의 히스토그램을 되돌린다.
int	<a href="#">getFreq</a> (float value) 화소값에 대한 빈도수를 되돌린다.
int	<a href="#">getFreq</a> (int index) 인덱스의 히스토그램을 되돌린다.
float	<a href="#">getKurtosis</a> () 지정된 농도 분포에 대한 Kurtosis값을 되돌린다.
float	<a href="#">getKurtosis</a> (int index)

	농도 분포에 대한 Kurtosis값을 되돌린다.
float	<u>getLevel</u> (float percent) 퍼센트값에 대한 화소값을 되돌린다.
int	<u>getMaxFreq</u> () 최대 빈도수를 되돌린다.
float	<u>getMean</u> () 농도 분포의 평균을 되돌린다.
float	<u>getSkewness</u> () 지정된 농도 분포에 대한 Skewness값을 되돌린다.
float	<u>getSkewness</u> (int index) 농도 분포에 대한 Skewness값을 되돌린다.
float	<u>getStd</u> () 농도 분포의 표준편차를 되돌린다.
java.lang.String	<u>getTitle</u> () 히스토그램의 타이틀을 설정한다.
int	<u>getTotalFreq</u> () 총 화소수를 되돌린다.
void	<u>operate</u> () 히스토그램 작성을 실행한다.
void	<u>read</u> (java.lang.String path) 농도 분포를 파일에서 입력한다.
void	<u>regionByEnd</u> (float start, float end, int intervals) 히스토그램 작성을 위한 농도 범위를 설정한다.
void	<u>regionByWidth</u> (float start, float width, int intervals) 히스토그램 작성을 위한 농도 범위를 설정한다.
void	<u>setBandIndex</u> (int i) 작성된 히스토그램을 얻고자하는 분광대의 인덱스를 설정한다.
void	<u>setBands</u> (int[] band) 처리 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<u>setCumulativeFreq</u> () 지정된 분광대에 대한 누적 농도 분포를 구한다.
void	<u>setHistogramFileName</u> (java.lang.String f) 히스토그램 저장 파일을 설정한다.
void	<u>setMaxFreq</u> () 최대 빈도수를 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target srcTgt) 대상 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.
java.lang.String	<u>toString</u> ()
void	<u>write</u> (java.lang.String path) 농도 분포를 파일로 출력한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### HistogramOp

```
public HistogramOp()
```

Constructor

## Method Detail

### setBands

```
public void setBands(int[] band)
```

처리 대상 분광대를 설정한다.

---

### setTarget

```
public void setTarget(Target srcTgt)
```

대상 영상을 설정한다.

---

### setHistogramFileName

```
public void setHistogramFileName(java.lang.String f)
```

히스토그램 저장 파일을 설정한다. 확장자는 hgm이다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### getTitle

```
public java.lang.String getTitle()
```

히스토그램의 타이틀을 설정한다.

---

## regionByEnd

```
public void regionByEnd(float start,
                        float end,
                        int intervals)
    throws IllegalArgumentException
```

히스토그램 작성을 위한 농도 범위를 설정한다.

---

## regionByWidth

```
public void regionByWidth(float start,
                          float width,
                          int intervals)
    throws IllegalArgumentException
```

히스토그램 작성을 위한 농도 범위를 설정한다.

---

## operate

```
public void operate()
    throws java.io.IOException,
           java.lang.InterruptedException
```

히스토그램 작성을 실행한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## calculateFrequency

```
protected void calculateFrequency(int bandIndex,
                                  byte[] pix,
                                  int aOffset,
                                  int length)
```

---

## calculateFrequency

```
protected void calculateFrequency(int bandIndex,
                                  short[] pix,
                                  int aOffset,
                                  int length)
```

---

## calculateFrequency

```
protected void calculateFrequency(int bandIndex,
                                  int[] pix,
                                  int aOffset,
                                  int length)
```

---

## calculateFrequency

```
protected void calculateFrequency(int bandIndex,  
                                float[] pix,  
                                int aOffset,  
                                int length)
```

---

## setBandIndex

```
public void setBandIndex(int i)
```

작성된 히스토그램을 얻고자하는 분광대의 인덱스를 설정한다.

---

## getTotalFreq

```
public int getTotalFreq()
```

총 화소수를 되돌린다.

---

## setMaxFreq

```
public void setMaxFreq()
```

최대 빈도수를 설정한다.

---

## getMaxFreq

```
public int getMaxFreq()
```

최대 빈도수를 되돌린다.

---

## getFreq

```
public int[] getFreq()
```

지정된 인덱스의 히스토그램을 되돌린다.

---

## getFreq

```
public int getFreq(int index)
```

인덱스의 히스토그램을 되돌린다.

---

## getFreq

```
public int getFreq(float value)
```

화소값에 대한 빈도수를 되돌린다.

---

## setCumulativeFreq

```
public void setCumulativeFreq()
```

지정된 분광대에 대한 누적 농도 분포를 구한다.

---

## getCumulativeFreq

```
public int[] getCumulativeFreq()
```

지정된 분광대에 대한 누적 농도 분포를 되돌린다.

---

## getCumulativeFreq

```
public int getCumulativeFreq(int index)
```

지정된 분광대의 화소값에 대한 누적 농도 분포를 되돌린다.

---

## getKurtosis

```
public float getKurtosis()
```

지정된 농도 분포에 대한 Kurtosis값을 되돌린다.

---

## getKurtosis

```
public float getKurtosis(int index)
```

농도 분포에 대한 Kurtosis값을 되돌린다.

---

## getSkewness

```
public float getSkewness()
```

지정된 농도 분포에 대한 Skewness값을 되돌린다.

---

## getSkewness

```
public float getSkewness(int index)
```

농도 분포에 대한 Skewness값을 되돌린다.

---

## getLevel

```
public float getLevel(float percent)
```

퍼센트값에 대한 화소값을 되돌린다.

---

## getMean

```
public float getMean()
```



농도 분포의 평균을 되돌린다.

---

## getStd

```
public float getStd()
```

농도 분포의 표준편차를 되돌린다.

---

## read

```
public void read(java.lang.String path)
    throws java.io.IOException
```

농도 분포를 파일에서 입력한다.

Throws:

java.io.IOException – IOException

---

## write

```
public void write(java.lang.String path)
    throws java.io.IOException
```

농도 분포를 파일로 출력한다.

Throws:

java.io.IOException – IOException

---

## toString

```
public java.lang.String toString()
```

Overrides:

toString in class java.lang.Object

---

## clear

```
public void clear()
```

히스토그램을 초기화한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class ImageLoader

```
java.lang.Object
|
+--ImageLoader
```

```
public class ImageLoader
extends java.lang.Object
```

화면출력시 영상을 읽어 드리는 클래스.

### Constructor Summary

<a href="#">ImageLoader</a> ( <a href="#">Target</a> tgt, int band)
Constructor

### Method Summary

int	<a href="#">getBand</a> () 분광대를 되돌린다.
<a href="#">Target</a>	<a href="#">getTarget</a> () 영상을 되돌린다.
byte[]	<a href="#">load</a> (int scale, byte[] pixel) 축척에 대한 영상을 읽어 들인다.
byte[]	<a href="#">load</a> (java.awt.Rectangle r, byte[] pixel) 처리 영역에 대한 영상을 읽어 들인다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### ImageLoader

```
public ImageLoader(Target tgt,
                  int band)
    throws java.io.IOException
```

Constructor

## Method Detail

### getTarget

```
public Target getTarget()
```

영상을 되돌린다.

---

### getBand

```
public int getBand()
```

분광대를 되돌린다.

---

### load

```
public byte[] load(java.awt.Rectangle r,  
                  byte[] pixel)  
    throws java.io.IOException
```

처리 영역에 대한 영상을 읽어 들인다.

Throws:

java.io.IOException - IOException

---

### load

```
public byte[] load(int scale,  
                  byte[] pixel)  
    throws java.io.IOException
```

축척에 대한 영상을 읽어 들인다.

Throws:

java.io.IOException - IOException

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class ImageLoadOp

java.lang.Object  
|  
+--ImageLoadOp

---

public class ImageLoadOp  
extends java.lang.Object

화면출력시 영상을 읽어 드리는 클래스.

---

## Constructor Summary

<code>ImageLoadOp(Target bw, int no)</code> Constructor
--

## Method Summary

byte[]	<code>operate(int index, byte[] pixel)</code>
float[]	<code>operate(int index, float[] pixel)</code>
int[]	<code>operate(int index, int[] pixel)</code>
short[]	<code>operate(int index, short[] pixel)</code>
void	<code>setBoundary(int x, int y, int width, int height)</code> 처리 영역을 설정한다.
void	<code>setBoundary(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setSkip(int skip)</code> 축척에 따른 Skip수를 설정한다.

## Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

ImageLoadOp

```
public ImageLoadOp(Target bw,  
                  int no)
```

Constructor

## Method Detail

### setBoundary

```
public void setBoundary(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBoundary

```
public void setBoundary(int x,  
                       int y,  
                       int width,  
                       int height)
```

처리 영역을 설정한다.

---

### setSkip

```
public void setSkip(int skip)
```

축척에 따른 Skip수를 설정한다.

---

### operate

```
public byte[] operate(int index,  
                     byte[] pixel)
```

---

### operate

```
public short[] operate(int index,  
                      short[] pixel)
```

---

### operate

```
public int[] operate(int index,  
                    int[] pixel)
```

---

## Class [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class ImageMaker

```
java.lang.Object
|
+--ImageMaker
```

---

```
public class ImageMaker
extends java.lang.Object
```

화면출력을 위한 이미지를 작성한다.

---

Field Summary	
static int	<a href="#">BW</a>
static int	<a href="#">RGB</a>

---

Constructor Summary	
<a href="#">ImageMaker</a> ( <a href="#">Target</a> tgt, int band)	Constructor
<a href="#">ImageMaker</a> ( <a href="#">Target</a> r, int rno, <a href="#">Target</a> g, int gno, <a href="#">Target</a> b, int bno)	Constructor

Method Summary	
void	<u>close()</u> 화면 구성고 관련된 객체를 제거한다.
byte[]	<u>getLookupTable(int index)</u> 인덱스에 해당하는 화면출력 변환값을 되돌린다.
int[]	<u>getMainFreq(int index)</u> 주 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.
java.awt.Image	<u>getMainImage(java.awt.Rectangle r)</u> 주화면의 선택영역에 대한 이미지를 되돌린다.
int[]	<u>getMainViewFreq(int index)</u> 주 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.
int	<u>getMode()</u> 화면 출력 모드를 되돌린다.
void	<u>getOriginalValue(int x, int y, int[] rgb)</u> x, y좌표에서의 원 화소값을 되돌린다.
byte[]	<u>getOrigMainPixels(int index)</u> 인덱스에 대한 주화면의 원 화소값을 되돌린다.
byte[]	<u>getOrigScrollPixels(int index)</u> 인덱스에 대한 스크롤 화면의 원 화소값을 되돌린다.
Palette	<u>getPalette()</u> 색상표를 되돌린다.
void	<u>getScreenValue(int x, int y, int[] rgb)</u> x, y좌표에서의 화면 화소값을 되돌린다.
int[]	<u>getScrollFreq(int index)</u> 스크롤 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.
java.awt.Image	<u>getScrollImage(int scale)</u> 주어진 축척에 대한 스크롤 이미지를 되돌린다.
int[]	<u>getScrollViewFreq(int index)</u> 스크롤 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.
java.awt.Image	<u>reproduceMainImage()</u> 주화면에 대한 이미지를 재구성한다.
java.awt.Image	<u>reproduceScrollImage()</u> 스크롤 화면에 대한 이미지를 재구성한다.
void	<u>setGreyColorModel()</u> 흑백이미지 구성을 위한 칼라모델을 설정한다.
void	<u>setLookupTable(int index, byte[] lookup)</u> 인덱스에 해당하는 화면출력 변환값을 설정한다.
void	<u>setPalette(Palette cm)</u> 색상표를 설정한다.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### BW

```
public static final int BW
```

---

### RGB

```
public static final int RGB
```

## Constructor Detail

### ImageMaker

```
public ImageMaker(Target tgt,  
                 int band)  
    throws java.io.IOException
```

Constructor

---

### ImageMaker

```
public ImageMaker(Target r,  
                 int rno,  
                 Target g,  
                 int gno,  
                 Target b,  
                 int bno)  
    throws java.io.IOException
```

Constructor

---

## Method Detail

### getMode

```
public int getMode()
```

화면 출력 모드를 되돌린다.

---

### getOriginalValue

```
public void getOriginalValue(int x,  
                             int y,  
                             int[] rgb)
```

x, y좌표에서의 원 화소값을 되돌린다.

---



## getScreenValue

```
public void getScreenValue(int x,  
                           int y,  
                           int[] rgb)
```

x, y좌표에서의 화면 화소값을 되돌린다.

---

## setPalette

```
public void setPalette(Palette cm)
```

색상표를 설정한다.

---

## getPalette

```
public Palette getPalette()
```

색상표를 되돌린다.

---

## getLookupTable

```
public byte[] getLookupTable(int index)
```

인덱스에 해당하는 화면출력 변환값을 되돌린다.

---

## setLookupTable

```
public void setLookupTable(int index,  
                           byte[] lookup)
```

인덱스에 해당하는 화면출력 변환값을 설정한다.

---

## setGreyColorModel

```
public void setGreyColorModel()
```

흑백이미지 구성을 위한 칼라모델을 설정한다.

---

## getMainImage

```
public java.awt.Image getMainImage(java.awt.Rectangle r)
```

throws java.io.IOException

주화면의 선택영역에 대한 이미지를 되돌린다.

---

## reproduceMainImage

```
public java.awt.Image reproduceMainImage()
```

주화면에 대한 이미지를 재구성한다.

---

## getScrollImage

```
public java.awt.Image getScrollImage(int scale)
    throws java.io.IOException
```

주어진 축척에 대한 스크롤이미지를 되돌린다.

---

## reproduceScrollImage

```
public java.awt.Image reproduceScrollImage()
```

스크롤 화면에 대한 이미지를 재구성한다.

---

## getOrigMainPixels

```
public byte[] getOrigMainPixels(int index)
```

인덱스에 대한 주화면의 원 화소값을 되돌린다.

---

## getOrigScrollPixels

```
public byte[] getOrigScrollPixels(int index)
```

인덱스에 대한 스크롤 화면의 원 화소값을 되돌린다.

---

## getMainFreq

```
public int[] getMainFreq(int index)
```

주 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

---

## getScrollFreq

```
public int[] getScrollFreq(int index)
```

스크롤 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

---

## getMainViewFreq

```
public int[] getMainViewFreq(int index)
```

주 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

---

## getScrollViewFreq

```
public int[] getScrollViewFreq(int index)
```

스크롤 화면의 인덱스에 대한 화소값의 빈도를 되돌린다.

---

## close

```
public void close()
```

화면 구성고 관련된 객체를 제거한다.

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class ImageProcessor

```
java.lang.Object
|
+--ImageProcessor
```

```
public class ImageProcessor
extends java.lang.Object
```

그래픽 포맷의 입출력을 관리한다.

Constructor Summary	
<a href="#">ImageProcessor()</a>	

Method Summary	
java.awt.Image	<a href="#">getImage</a> (java.lang.String file) 파일로 부터 이미지를 생성하여 되돌린다.
<a href="#">Target</a>	<a href="#">getTarget</a> () 처리된 영상을 되돌린다.
void	<a href="#">handlePixels</a> (java.awt.Image img, int x, int y, int w, int h, int[] pix) 이미지 화소 색상값을 배열로 저장한다.
void	<a href="#">ImageProcessor</a> () Constructor
void	<a href="#">operate</a> () 이미지 포맷의 입력을 실행한다.
void	<a href="#">putImage</a> (java.lang.String file, java.awt.Image img, java.lang.String format) 이미지를 원하는 포맷으로 파일에 출력한다.
void	<a href="#">setBands</a> (int[] bands) 분광대를 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target tgt) 처리할 영상을 설정한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### ImageProcessor

```
public ImageProcessor()
```

## Method Detail

### ImageProcessor

```
public void ImageProcessor()
```

Constructor

---

### setTarget

```
public void setTarget(Target tgt)
```

처리할 영상을 설정한다.

---

### getTarget

```
public Target getTarget()
```

처리된 영상을 되돌린다.

---

### setBands

```
public void setBands(int[] bands)
```

분광대를 설정한다.

---

### getImage

```
public java.awt.Image getImage(java.lang.String file)
```

파일로부터 이미지를 생성하여 되돌린다.

---

## putImage

```
public void putImage(java.lang.String file,  
                    java.awt.Image img,  
                    java.lang.String format)
```

이미지를 원하는 포맷으로 파일에 출력한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

---

## operate

```
public void operate()
```

이미지 포맷의 입력을 실행한다.

---

## handlePixels

```
public void handlePixels(java.awt.Image img,  
                        int x,  
                        int y,  
                        int w,  
                        int h,  
                        int[] pix)
```

이미지 화소 색상값을 배열로 저장한다.

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class InteactiveEnhanceOp

```
java.lang.Object
|
+--InteactiveEnhanceOp
```

---

```
public class InteactiveEnhanceOp
extends java.lang.Object
implements Operator
```

대화형 영상향상을 실행한다.

---

## Constructor Summary

<a href="#">InteactiveEnhanceOp()</a> Constructor
--

## Method Summary

protected void	<a href="#">changePixelValue</a> (byte[] a, int aOffset, int index, byte[] b, int bOffset, int length)
void	<a href="#">operate</a> () 사용자 정의에 의한 영상향상을 실행한다.
void	<a href="#">setBands</a> (int[] bands) 대상 분광대를 설정한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<a href="#">setLUT</a> (int[][] lut) 사용자에 의하여 정의된 변환표를 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target stgt, Target dtgt) 대상 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructor Detail

### InteactiveEnhanceOp

public InteractiveEnhanceOp()

Constructor

## Method Detail

### setBands

public void setBands(int[] bands)

대상 분광대를 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle b)

처리 영역을 설정한다.

---

### setLUT

public void setLUT(int[][] lut)

사용자에 의하여 정의된 변환표를 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

public void setTarget(Target stgt,  
                  Target dtgt)

대상 영상을 설정한다.

---

### operate

public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException

사용자 정의에 의한 영상향상을 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---



## changePixelValue

```
protected void changePixelValue(byte[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class ISODATAClassifyOp

```
java.lang.Object
|
+--ISODATAClassifyOp
```

---

```
public class ISODATAClassifyOp
extends java.lang.Object
implements Operator
```

ISODATA 분류를 실행한다.

---

Field Summary	
float[][]	<a href="#">bandMax</a>
float[][]	<a href="#">bandMin</a>
int	<a href="#">classNo</a>
float[][]	<a href="#">Mean</a>
float[][]	<a href="#">Std</a>

Constructor Summary	
<a href="#">ISODATAClassifyOp()</a>	
Constructor	

Method Summary	
StatisticOld	<u>getStatistic()</u> 분류 결과에 대한 통계를 되돌린다.
void	<u>operate()</u> ISODATA 분류를 실행한다.
void	<u>setBands(int[] bands)</u> 대상 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리영역을 설정한다.
void	<u>setClassCount(int s)</u> 한 군집이 가질 수 있는 최대 화소수를 설정한다.
void	<u>setClassID(int id)</u>
void	<u>setDistanceMethod(java.lang.String method)</u> 분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.
void	<u>setInitStatistic(float[][] mean, float[][] std)</u> 초기 통계를 설정한다.
void	<u>setIteration(int i)</u> 분류를 반복 수행하기 위한 횟수를 설정한다.
void	<u>setMaxChangeRate(float p)</u> 이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다.
void	<u>setMaxClassSD(float s)</u> 분류하고자 하는 클래스의 표준편차를 설정한다.
void	<u>setMinClassDistance(float s)</u> 군집간의 최소거리를 설정한다.
void	<u>setMinClassPixelCount(int s)</u> 분류되는 군집의 최소 화소수를 설정한다.
void	<u>setParameter(float[] p)</u> 분류에 필요한 각 변수값들을 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setTarget(Target stgt, Target dtgt)</u> 대상 영상을 설정한다.
void	<u>setUnClassifyClass(int c)</u> 미분류 화소에 대한 군집값을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

**classNo**

public int classNo

---

## bandMax

```
public float[][] bandMax
```

---

## bandMin

```
public float[][] bandMin
```

---

## Mean

```
public float[][] Mean
```

---

## Std

```
public float[][] Std
```

## Constructor Detail

### ISODATAClassifyOp

```
public ISODATAClassifyOp()
```

Constructor

## Method Detail

### setDistanceMethod

```
public void setDistanceMethod(java.lang.String method)
```

분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

---

### setClassID

```
public void setClassID(int id)
```

---

### getStatistic

```
public StatisticOld getStatistic()
```

분류 결과에 대한 통계를 되돌린다.

---

### setUnClassifyClass

```
public void setUnClassifyClass(int c)
```

미분류 화소에 대한 군집값을 설정한다.

---

## setIteration

```
public void setIteration(int i)
```

분류를 반복 수행하기 위한 횟수를 설정한다.

---

## setMaxClassSD

```
public void setMaxClassSD(float s)
```

분류하고자 하는 클래스의 표준편차를 설정한다.

---

## setMinClassDistance

```
public void setMinClassDistance(float s)
```

군집간의 최소거리를 설정한다.

---

## setMinClassPixelCount

```
public void setMinClassPixelCount(int s)
```

분류되는 군집의 최소 화소수를 설정한다. 이 이하일 경우 다른 군집과 합해진다.

---

## setClassCount

```
public void setClassCount(int s)
```

한 군집이 가질 수 있는 최대 화소수를 설정한다. 이 이상이 될 경우 두개의 군집으로 분리된다.

---

## setMaxChangeRate

```
public void setMaxChangeRate(float p)
```

이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다. 이 이하일 경우 반복수행을 종료한다.

---

## setParameter

```
public void setParameter(float[] p)
```

분류에 필요한 각 변수값들을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

## setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

ISODATA 분류를 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## setInitStatistic

```
public void setInitStatistic(float[][] mean,  
                            float[][] std)
```

초기 통계를 설정한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class KMeansClassifyOp

java.lang.Object

|--KMeansClassifyOp

---

```
public class KMeansClassifyOp
extends java.lang.Object
implements Operator
```

K-means방법에 의한 분류를 실행한다.

---

Field Summary	
float[][]	<a href="#">bandMax</a>
float[][]	<a href="#">bandMin</a>
int	<a href="#">classNo</a>
float[][]	<a href="#">Mean</a>
float[][]	<a href="#">Std</a>

Constructor Summary	
<a href="#">KMeansClassifyOp()</a>	Constructor

## Method Summary

Statistic	<u>getStatistic()</u> 분류 결과에 대한 통계를 되돌린다.
void	<u>operate()</u> K-Means 분류를 실행한다.
void	<u>setBands(int[] bands)</u> 대상 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리영역을 설정한다.
void	<u>setChangeRate(float s)</u> 이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다.
void	<u>setClassCount(int n)</u> 분류 군집 수를 설정한다.
void	<u>setClassMean(float[][] m)</u> 군집의 평균값을 설정한다.
void	<u>setClassNo(int n)</u> 분류 군집 수를 설정한다.
void	<u>setClassStd(float[][] m)</u> 군집의 표준편차값을 설정한다.
void	<u>setCutDistance(float c)</u> 군집에 포함될 임계 거리값을 설정한다.
void	<u>setDistanceMethod(java.lang.String method)</u> 분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.
void	<u>setIteration(int i)</u> 분류를 반복 수행하기 위한 횟수를 설정한다.
void	<u>setParameter(float[] p)</u> 분류에 필요한 각 변수값들을 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setStatisticFile(java.lang.String f)</u> 표본 통계 파일을 지정한다.
void	<u>setTarget(Target stgt, Target dtgt)</u> 대상 영상을 설정한다.
void	<u>setUnClassifyClass(int c)</u> 미분류 화소에 대한 군집값을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### classNo

public int classNo



---

## bandMax

```
public float[][] bandMax
```

---

## bandMin

```
public float[][] bandMin
```

---

## Mean

```
public float[][] Mean
```

---

## Std

```
public float[][] Std
```

## Constructor Detail

### KMeansClassifyOp

```
public KMeansClassifyOp()
```

Constructor

## Method Detail

### setDistanceMethod

```
public void setDistanceMethod(java.lang.String method)
```

분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

---

### getStatistic

```
public Statistic getStatistic()
```

분류 결과에 대한 통계를 되돌린다.

---

### setClassNo

```
public void setClassNo(int n)
```

분류 군집 수를 설정한다.

---

### setClassCount

```
public void setClassCount(int n)
```

분류 군집 수를 설정한다.

---

## setClassMean

```
public void setClassMean(float[][] m)
```

군집의 평균값을 설정한다.

---

## setClassStd

```
public void setClassStd(float[][] m)
```

군집의 표준편차값을 설정한다.

---

## setUnClassifyClass

```
public void setUnClassifyClass(int c)
```

미분류 화소에 대한 군집값을 설정한다.

---

## setCutDistance

```
public void setCutDistance(float c)
```

군집에 포함될 임계 거리값을 설정한다.

---

## setIteration

```
public void setIteration(int i)
```

분류를 반복 수행하기 위한 횟수를 설정한다.

---

## setChangeRate

```
public void setChangeRate(float s)
```

이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다. 이 이하일 경우 반복수행을 종료한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

## setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

## setStatisticFile

```
public void setStatisticFile(java.lang.String f)
    throws java.io.IOException,
           IllegalArgumentException
```

표본 통계 파일을 지정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,
                     Target dtgt)
```

대상 영상을 설정한다.

---

## setParameter

```
public void setParameter(float[] p)
```

분류에 필요한 각 변수값들을 설정한다.

---

## operate

```
public void operate()
    throws java.io.IOException,
           java.lang.InterruptedException
```

K-Means 분류를 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class `KMeansUnSupervisedClassifyOp`

`java.lang.Object`

|--`KMeansUnSupervisedClassifyOp`

---

```
public class KMeansUnSupervisedClassifyOp
```

```
extends java.lang.Object
```

```
implements Operator
```

무감독에 의한 K-means 분류를 실행한다.

---

Field Summary	
<code>float[][]</code>	<a href="#">bandMax</a>
<code>float[][]</code>	<a href="#">bandMin</a>
<code>int</code>	<a href="#">classNo</a>
<code>float[][]</code>	<a href="#">Mean</a>
<code>float[][]</code>	<a href="#">Std</a>

Constructor Summary	
<a href="#">KMeansUnSupervisedClassifyOp()</a>	Constructor
<a href="#">KMeansUnSupervisedClassifyOp(java.lang.String method)</a>	Constructor

## Method Summary

Statistic	<code>getStatistic()</code> 분류 결과에 대한 통계를 되돌린다.
void	<code>operate()</code> 무감독 K-Means 분류를 실행한다.
void	<code>setBands(int[] bands)</code> 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리영역을 설정한다.
void	<code>setChangeRate(float s)</code> 이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다.
void	<code>setClassCount(int s)</code> 분류 군집 수를 설정한다.
void	<code>setDistanceMethod(java.lang.String method)</code> 분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.
void	<code>setInitStatistic(float[][] mean, float[][] std)</code>
void	<code>setIteration(int i)</code> 분류를 반복 수행하기 위한 횟수를 설정한다.
void	<code>setParameter(float[] p)</code> 분류에 필요한 각 변수값들을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### classNo

public int classNo

---

### bandMax

public float[][] bandMax

---

### bandMin

public float[][] bandMin

---

## Mean

```
public float[][] Mean
```

---

## Std

```
public float[][] Std
```

## Constructor Detail

### KMeansUnSupervisedClassifyOp

```
public KMeansUnSupervisedClassifyOp()
```

Constructor

---

### KMeansUnSupervisedClassifyOp

```
public KMeansUnSupervisedClassifyOp(java.lang.String method)
```

Constructor

## Method Detail

### setDistanceMethod

```
public void setDistanceMethod(java.lang.String method)
```

분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

---

### getStatistic

```
public Statistic getStatistic()
```

분류 결과에 대한 통계를 되돌린다.

---

### setIteration

```
public void setIteration(int i)
```

분류를 반복 수행하기 위한 횟수를 설정한다.

---

### setClassCount

```
public void setClassCount(int s)
```

분류 군집 수를 설정한다.

---

## setChangeRate

```
public void setChangeRate(float s)
```

이전에 분류된 각 클래스의 내용이 다른 군집으로 변경된 화소수의 비율을 설정한다. 이 이하일 경우 반복수행을 종료한다.

---

## setParameter

```
public void setParameter(float[] p)
```

분류에 필요한 각 변수값들을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

## setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedExcepion
```

무감독 K-Means 분류를 실행한다.

Throws:

java.io.IOException - IOException

java.lang.InterruptedExcepion - InterruptedExcepion

---

## setInitStatistic

```
public void setInitStatistic(float[][] mean,  
                             float[][] std)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---



## Class LabeledCrossMark

```
java.lang.Object
|
+--LabeledCrossMark
```

---

```
public class LabeledCrossMark
extends java.lang.Object
implements Drawable
```

화면에 출력되는 CrossMark를 관리한다.

---

Field Summary	
Text	<a href="#">Label</a>
<a href="#">CrossMark</a>	<a href="#">Mark</a>

Constructor Summary	
<a href="#">LabeledCrossMark(int x, int y)</a> x, y 위치에 CrossMark를 출력한다.	
<a href="#">LabeledCrossMark(int x, int y, java.lang.String label)</a> x, y 위치에 입력 Label를 갖는 CrossMark를 출력한다.	

## Method Summary

void	<code>draw(java.awt.Graphics g, float scale)</code> CrossMark와 Label을 그린다.
void	<code>draw(java.awt.Graphics g, int xoffset, int yoffset)</code> CrossMark와 Label을 그린다.
java.awt.Rectangle	<code>getBounds()</code> CrossMark의 경계를 되돌린다.
java.awt.Rectangle	<code>getBounds(float scale)</code> CrossMark의 축척에 대한 경계를 되돌린다.
java.awt.Rectangle	<code>getBounds(int xoffset, int yoffset)</code> CrossMark의 이동된 경계를 되돌린다.
java.awt.Point	<code>getLocation()</code> CrossMark의 위치를 되돌린다.
int	<code>getType()</code> CrossMark의 종류를 되돌린다.
void	<code>setColor(java.awt.Color color)</code> CrossMark의 색상을 설정한다.
void	<code>setLabel(java.lang.String label)</code> CrossMark의 Label을 설정한다.
void	<code>setLocation(int x, int y)</code> CrossMark의 위치를 설정한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### Label

public Text Label

---

### Mark

public CrossMark Mark

## Constructor Detail

### LabeledCrossMark

public LabeledCrossMark(int x,  
int y)

x, y 위치에 CrossMark를 출력한다.

---

### LabeledCrossMark

public LabeledCrossMark(int x,  
int y,

java.lang.String label)

x, y 위치에 입력 Label를 갖는 CrossMark를 출력한다.

## Method Detail

### getType

public int `getType()`

CrossMark의 종류를 되돌린다.

Specified by:

`getType` in interface `Drawable`

---

### setColor

public void `setColor(java.awt.Color color)`

CrossMark의 색상을 설정한다.

Specified by:

`setColor` in interface `Drawable`

---

### setLabel

public void `setLabel(java.lang.String label)`

CrossMark의 Label을 설정한다.

---

### getLocation

public java.awt.Point `getLocation()`

CrossMark의 위치를 되돌린다.

Specified by:

`getLocation` in interface `Drawable`

---

### setLocation

public void `setLocation(int x,  
int y)`

CrossMark의 위치를 설정한다.

Specified by:

`setLocation` in interface `Drawable`

---

### getBounds

public java.awt.Rectangle `getBounds()`

CrossMark의 경계를 되돌린다.  
Specified by:  
[getBounds](#) in interface [Drawable](#)

---

## getBounds

```
public java.awt.Rectangle getBounds(int xoffset,  
                                       int yoffset)
```

CrossMark의 이동된 경계를 되돌린다.  
Specified by:  
[getBounds](#) in interface [Drawable](#)

---

## getBounds

```
public java.awt.Rectangle getBounds(float scale)
```

CrossMark의 축척에 대한 경계를 되돌린다.  
Specified by:  
[getBounds](#) in interface [Drawable](#)

---

## draw

```
public void draw(java.awt.Graphics g,  
                int xoffset,  
                int yoffset)
```

CrossMark와 Label을 그린다.  
Specified by:  
[draw](#) in interface [Drawable](#)

---

## draw

```
public void draw(java.awt.Graphics g,  
                float scale)
```

CrossMark와 Label을 그린다.  
Specified by:  
[draw](#) in interface [Drawable](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class LANDSATTMImportOp

```
java.lang.Object
|
+--LANDSATTMImportOp
```

```
public class LANDSATTMImportOp
extends java.lang.Object
implements Operator
```

Landsat TM 영상을 내부 포맷으로 변환 저장한다.

## Constructor Summary

LANDSATTMImportOp() Constructor
------------------------------------

## Method Summary

void	<code>operate()</code> Landsat TM 영상을 내부 포맷으로 변환한다.
void	<code>setBands(int[] bands)</code> 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setHeader(LANDSATTMHeader hd)</code> LANDSAT TM의 헤더를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(java.io.File f, Target.File tgt)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### LANDSATTMImportOp

```
public LANDSATTMImportOp()
```

## Method Detail

### setHeader

```
public void setHeader(LANDSATMHeader hd)
```

LANDSAT TM의 헤더를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(java.io.File f,  
                      Target.File tgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

Landsat TM 영상을 내부 포맷으로 변환한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class LaplacianKernel

```
java.lang.Object
|
+--LaplacianKernel
```

```
public class LaplacianKernel
    extends java.lang.Object
```

Laplacian Kernel를 관리한다.

### Constructor Summary

LaplacianKernel(java.lang.String method)	
Constructor	

### Method Summary

int	<a href="#">getDivisor()</a> 콘볼루션 수행에 대한 분모 연산자를 되돌린다.
int[]	<a href="#">getHKernel()</a> Laplacian (H)Kernel 배열의 가중치를 되돌린다.
int[]	<a href="#">getKernel()</a> 현재 설정되어 있는 Laplacian Kernel 배열의 가중치를 되돌린다.
int[]	<a href="#">getLRKernel()</a> Laplacian (LR)Kernel 배열의 가중치를 되돌린다.
int[]	<a href="#">getOmniKernel()</a> Laplacian (Omni)Kernel 배열의 가중치를 되돌린다.
int[]	<a href="#">getRLKernel()</a> Laplacian (RL)Kernel 배열의 가중치를 되돌린다.
int[]	<a href="#">getVKernel()</a> Laplacian (V)Kernel 배열의 가중치를 되돌린다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

LaplacianKernel



public LaplacianKernel(java.lang.String method)

Constructor

## Method Detail

### getKernel

public int[] getKernel()

현재 설정되어 있는 Laplacian Kernel 배열의 가중치를 되돌린다.

---

### getLRKernel

public int[] getLRKernel()

Laplacian (LR)Kernel 배열의 가중치를 되돌린다.

---

### getOmniKernel

public int[] getOmniKernel()

Laplacian (Omni)Kernel 배열의 가중치를 되돌린다.

---

### getHKernel

public int[] getHKernel()

Laplacian (H)Kernel 배열의 가중치를 되돌린다.

---

### getVKernel

public int[] getVKernel()

Laplacian (V)Kernel 배열의 가중치를 되돌린다.

---

### getRLKernel

public int[] getRLKernel()

Laplacian (RL)Kernel 배열의 가중치를 되돌린다.

## Class LevelSliceOp

```
java.lang.Object
|
+--LevelSliceOp
```

```
public class LevelSliceOp
extends java.lang.Object
implements Operator
```

Density level slicing을 실행한다.

Constructor Summary	
<code>LevelSliceOp()</code>	Constructor

Method Summary	
protected void	<code>changePixelValue(double[] a, int aOffset, byte[] b, int bOffset, int length)</code>
void	<code>operate()</code> Density Slicing을 실행한다.
void	<code>setBand(int band)</code> 처리 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setParameter(double[][][] mtx)</code> Level slicing 변환표를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

## Constructor Detail

### LevelSliceOp

```
public LevelSliceOp()
```

Constructor

## Method Detail

### setBand

```
public void setBand(int band)
```

처리 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setParameter

```
public void setParameter(double[][][] mtx)
```

Level slicing 변환표를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

Density Slicing을 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                int aOffset,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class LookupReplaceOp

java.lang.Object  
|  
+--LookupReplaceOp

```
public class LookupReplaceOp  
extends java.lang.Object
```

참조표를 이용하여 배열의 값을 재구성한다.

### Constructor Summary

<code>LookupReplaceOp()</code> Constructor
<code>LookupReplaceOp(byte[] lookup)</code> Constructor

### Method Summary

void	<code>operate(byte[] pixel)</code> 배열의 재구성을 실행한다.
void	<code>setLookupTable(byte[] lookup)</code> 참조표를 설정한다.

### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Detail

#### LookupReplaceOp

```
public LookupReplaceOp()  
    Constructor
```

#### LookupReplaceOp

public LookupReplaceOp(byte[] lookup)

Constructor

## Method Detail

### setLookupTable

public void setLookupTable(byte[] lookup)

참조표를 설정한다.

---

### operate

public void operate(byte[] pixel)

배열의 재구성을 실행한다.

---

#### [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class MenuToken

java.lang.Object  
|  
+--MenuToken

---

```
public class MenuToken
extends java.lang.Object
```

Menu Description file 에서 읽어진 token 을 가지고 MenuToken object 를 만들게 된다  
Menu Display 하는 데이터 구조의 기본 클래스이다. 이 MenuToken 을 보고 실제 Menu  
혹은 MenuItem 을 만들게 된다.

Version:  
1.0  
Author:  
GeoPixel team

---

## Constructor Summary

<a href="#">MenuToken()</a>	
-----------------------------	--

Method Summary	
void	<u>addItem(MenuToken item)</u> 현재 MenuToken 에 MenuItem 이나 Menu 가 될 MenuToken 을 child 로 붙인다.
java.util.Vector	<u>getChildren()</u> 현재 구성된 MenuToken 의 Child 가 있는 Vector 를 가져오는데 보통 Menu 또는 MenuItem 을 구성하기 위해 쓰거나 Serial 하게 구성되어있는 MenuToken 을 실제 메뉴형태인 Tree 구조로 바꾸기 위해서 특정 MenuToken 과 그 Children 을 가져온다.
java.lang.String	<u>getClassname()</u> Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 classname 을 가져온다
java.lang.String	<u>getId()</u> Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 id 값을 가져온다
int	<u>getKey()</u> Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 key 값을 가져온다
java.lang.String	<u>getLabel()</u> Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 label 값을 가져온다
java.lang.String	<u>getParent()</u> Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 parent 값을 가져온다
boolean	<u>hasChildren()</u> 현재 MenuToken 의 Object 가 Child 의 Object 가 있는 지를 check 한다
boolean	<u>isShift()</u> Menu 또는 MenuItem 을 구성하기 위해서 현재 구성된 MenuToken 의 shift 의 boolean 값을 가져온다
void	<u>setClassname(java.lang.String className)</u> Parser 가 file 로 부터 얻어들이는 classname 을 MenuToken 의 classname 로 Setting 한다.
void	<u>setId(java.lang.String id)</u> Parser 가 file 로 부터 얻어들이는 id 값을 MenuToken 의 id 로 Setting 한다.
void	<u>setKey(int key)</u> Parser 가 file 로 부터 얻어들이는 key 값을 MenuToken 의 key 로 Setting 한다.
void	<u>setLabel(java.lang.String label)</u> Parser 가 file 로 부터 얻어들이는 Label 값을 MenuToken 의 Label 로 Setting 한다.
void	<u>setParent(java.lang.String parent)</u> Parser 가 file 로 부터 얻어들이는 Parent 값을 MenuToken 의 parent 로 Setting 한다.
void	<u>setShift(boolean isShift)</u> Parser 가 file 로 부터 얻어들이는 boolean shift 값을 MenuToken 의 shift 값으로 Setting 한다.



## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### MenuToken

```
public MenuToken()
```

## Method Detail

### hasChildren

```
public boolean hasChildren()
```

현재 MenuToken 의 Object 가 Child 의 Object 가 있는 지를 check 한다

Returns:

이 object 가 child 를 가지고 있는 지 여부

---

### addItem

```
public void addItem(MenuToken item)
```

현재 MenuToken 에 MenuItem 이나 Menu 가 될 MenuToken 을 child 로 붙인다.

Parameters:

item - child 로 가질 MenuToken

---

### setId

```
public void setId(java.lang.String id)
```

Parser 가 file 로 부터 얻어들인 id 값을 MenuToken 의 id 로 Setting 한다.

Parameters:

id - Description file 로 부터 읽은 id 값

---

### setParent

```
public void setParent(java.lang.String parent)
```

Parser 가 file 로 부터 얻어들인 Parent 값을 MenuToken 의 parent 로 Setting 한다.

Parameters:

id - Description file 로 부터 읽은 parent 값

---

## setLabel

```
public void setLabel(java.lang.String label)
```

Parser 가 file 로 부터 얻어들이 Label 값을 MenuToken 의 Label 로 Setting 한다.

**Parameters:**

id -Description file 로 부터 읽은 label 값

---

## setKey

```
public void setKey(int key)
```

Parser 가 file 로 부터 얻어들이 key 값을 MenuToken 의 key 로 Setting 한다.

**Parameters:**

id -Description file 로 부터 읽은 key 값

---

## setShift

```
public void setShift(boolean isShift)
```

Parser 가 file 로 부터 얻어들이 boolean shift 값을 MenuToken 의 shift 값으로 Setting 한다.

**Parameters:**

id -Description file 로 부터 읽은 shift 의 boolean 값

---

## setClassname

```
public void setClassname(java.lang.String className)
```

Parser 가 file 로 부터 얻어들이 classname 을 MenuToken 의 classname 로 Setting 한다.

**Parameters:**

id -Description file 로 부터 읽은 classname

---

## getId

```
public java.lang.String getId()
```

Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 id 값을 가져온다

**Returns:**

현재 MenuToken 의 id

---

## getParent

```
public java.lang.String getParent()
```

Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 parent 값을 가져온다

Returns:  
현재 MenuToken 의 parent

---

## getLabel

```
public java.lang.String getLabel()
```

Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 label 값을 가져온다

Returns:  
현재 MenuToken 의 label

---

## getKey

```
public int getKey()
```

Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 key 값을 가져온다

Returns:  
현재 MenuToken 의 key

---

## isShift

```
public boolean isShift()
```

Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 shift 의 boolean 값을 가져온다

Returns:  
현재 MenuToken 의 Shift 의 boolean 값

---

## getClassName

```
public java.lang.String getClassName()
```

Menu 또는 MenuItem 을 구성하기위해서 현재 구성된 MenuToken 의 classname 을 가져온다

Returns:  
현재 MenuToken 의 classname

---

---

## Class MinDistanceClassifyOp

java.lang.Object  
|  
+--MinDistanceClassifyOp

---

public class MinDistanceClassifyOp  
extends java.lang.Object  
implements [Operator](#)

최소거리 분류법을 실행한다.

---

Field Summary	
double[][]	<a href="#">Mean</a>
int	<a href="#">NoOfBands</a>
int	<a href="#">NoOfClasses</a>
double[][]	<a href="#">Std</a>

Constructor Summary	
<a href="#">MinDistanceClassifyOp()</a>	Constructor

## Method Summary

void	<u>operate()</u> 최소 거리 분류를 실행한다.
void	<u>setBands(int[] bands)</u> 대상 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리영역을 설정한다.
void	<u>setCutDistance(float c)</u> 군집에 포함될 임계 거리값을 설정한다.
void	<u>setDistanceMethod(java.lang.String method)</u> 분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setStatisticsFile(java.io.File f)</u> 통계 정보를 갖고 있는 파일을 설정한다.
void	<u>setTarget(Target stgt, Target dtgt)</u> 대상 영상을 설정한다.
void	<u>setUnClassifyClass(int c)</u> 미분류 화소에 대한 군집값을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### NoOfClasses

```
public int NoOfClasses
```

---

### NoOfBands

```
public int NoOfBands
```

---

### Mean

```
public double[][] Mean
```

---

### Std

public double[][] Std

## Constructor Detail

### MinDistanceClassifyOp

public MinDistanceClassifyOp()

Constructor

## Method Detail

### setDistanceMethod

public void setDistanceMethod(java.lang.String method)

분류에 있어 각 클래스에 대한 거리를 결정하는 방법을 설정한다.

---

### setCutDistance

public void setCutDistance(float c)

군집에 포함될 임계 거리값을 설정한다.

---

### setUnClassifyClass

public void setUnClassifyClass(int c)

미분류 화소에 대한 군집값을 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle r)

처리영역을 설정한다.

---

### setBands

public void setBands(int[] bands)

대상 분광대를 설정한다.

---

### setStatisticsFile

public void setStatisticsFile(java.io.File f)  
throws java.io.IOException,  
IllegalFormatException,  
VisualNumerics.math.MathException

통계 정보를 갖고 있는 파일을 설정한다.

### Throws:

[java.io.IOException](#) – [IOException](#)  
[IllegalFormatException](#) – [IllegalFormatException](#)  
[VisualNumerics.math.MathException](#) – [MathException](#)

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

최소 거리 분류를 실행한다.

Throws:

[java.io.IOException](#) – [IOException](#)  
[java.lang.InterruptedException](#) – [InterruptedException](#)

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class MinMaxEnhanceOp

java.lang.Object

|--MinMaxEnhanceOp

---

```
public class MinMaxEnhanceOp
```

```
extends java.lang.Object
```

```
implements Operator
```

최소-최대값을 이용한 선형 영상 향상을 실행한다.

---

### Constructor Summary

<a href="#">MinMaxEnhanceOp()</a> Constructor	
--	--



Method Summary	
protected void	<code>changePixelValue(byte[] a, int aoffset, int index, byte[] b, int boffset, int length)</code>
protected void	<code>changePixelValue(float[] a, int aoffset, int index, byte[] b, int boffset, int length)</code>
protected void	<code>changePixelValue(int[] a, int aoffset, int index, byte[] b, int boffset, int length)</code>
protected void	<code>changePixelValue(short[] a, int aoffset, boolean f, int index, byte[] b, int boffset, int length)</code>
protected void	<code>changePixelValue(short[] a, int aoffset, int index, byte[] b, int boffset, int length)</code>
void	<code>operate()</code> 최소-최대값을 이용한 선형 향상을 실행한다.
void	<code>setBands(int[] bands)</code> 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리영역을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 최소-최대값 결정 방법을 설정한다.
void	<code>setMinMax(double[] bmin, double[] bmax)</code> 각 분광대의 최소-최대값을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

**Methods inherited from class java.lang.Object**  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### MinMaxEnhanceOp

public MinMaxEnhanceOp()

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

최소-최대값 결정 방법을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

## setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

## setMinMax

```
public void setMinMax(double[] bmin,  
                      double[] bmax)
```

각 분광대의 최소-최대값을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

최소-최대값을 이용한 선형 향상을 실행한다.

Throws:

java.io.IOException - IOException

## changePixelValue

```
protected void changePixelValue(byte[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## changePixelValue

```
protected void changePixelValue(short[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## changePixelValue

```
protected void changePixelValue(short[] a,  
                                int aOffset,  
                                boolean f,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## changePixelValue

```
protected void changePixelValue(int[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## changePixelValue

```
protected void changePixelValue(float[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## Class MinMaxLinearEnhanceOp

java.lang.Object

|--MinMaxLinearEnhanceOp

---

```
public class MinMaxLinearEnhanceOp
    extends java.lang.Object
    implements Operator
```

최소-최대값을 이용한 선형 영상 향상을 실행한다.

---

### Constructor Summary

<a href="#">MinMaxLinearEnhanceOp()</a> Constructor	
--	--

## Method Summary

protected void	<u>changePixelValue</u> (byte[] a, int aOffset, int index, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (float[] a, int aOffset, int index, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (int[] a, int aOffset, int index, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (short[] a, int aOffset, boolean f, int index, byte[] b, int bOffset, int length)
protected void	<u>changePixelValue</u> (short[] a, int aOffset, int index, byte[] b, int bOffset, int length)
void	<u>operate</u> () 최소-최대값을 이용한 선형 향상을 실행한다.
void	<u>setBands</u> (int[] bands) 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle r) 처리영역을 설정한다.
void	<u>setMethod</u> (java.lang.String method) 최소-최대값 결정 방법을 설정한다.
void	<u>setMinMax</u> (double[] bmin, double[] bmax) 각 분광대의 최소-최대값을 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 대상 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### MinMaxLinearEnhanceOp

```
public MinMaxLinearEnhanceOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

최소-최대값 결정 방법을 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

## setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

## setMinMax

```
public void setMinMax(double[] bmin,  
                      double[] bmax)
```

각 분광대의 최소-최대값을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

최소-최대값을 이용한 선형 향상을 실행한다.

Throws:

java.io.IOException - IOException

### changePixelValue

```
protected void changePixelValue(byte[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

### changePixelValue

```
protected void changePixelValue(short[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

### changePixelValue

```
protected void changePixelValue(short[] a,  
                                int aOffset,  
                                boolean f,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

### changePixelValue

```
protected void changePixelValue(int[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

### changePixelValue

```
protected void changePixelValue(float[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## Class MinMaxOp

```
java.lang.Object
|
+--MinMaxOp
```

```
public class MinMaxOp
extends java.lang.Object
implements Operator
```

영상의 각 분광대의 최소, 최대값을 구한다.

### Constructor Summary

<u>MinMaxOp()</u> Constructor
----------------------------------

### Method Summary

double[]	<u>getMax()</u> 각 분광대의 최대값 배열을 되돌린다.
double	<u>getMax(int b)</u> 특정 분광대의 최대값을 되돌린다.
double[]	<u>getMin()</u> 각 분광대의 최소값 배열을 되돌린다.
double	<u>getMin(int b)</u> 특정 분광대의 최소값을 되돌린다.
void	<u>operate()</u> 최소-최대값 찾기를 실행한다.
void	<u>setBands(int[] bands)</u> 처리 대상 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리 영역을 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setTarget(Target stgt)</u> 대상 영상을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.



## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### MinMaxOp

```
public MinMaxOp()
```

Constructor

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

처리 대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태마를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt)
```

대상 영상을 설정한다.

---

### getMin

```
public double[] getMin()
```

각 분광대의 최소값 배열을 되돌린다.

---

## getMax

```
public double[] getMax()
```

각 분광대의 최대값 배열을 되돌린다.

---

## getMin

```
public double getMin(int b)
```

특정 분광대의 최소값을 되돌린다.

---

## getMax

```
public double getMax(int b)
```

특정 분광대의 최대값을 되돌린다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

최소-최대값 찾기를 실행한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS  
SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES  
DETAIL: FIELD | CONSTR | METHOD

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class MLClassifyOp

java.lang.Object  
|  
|--MLClassifyOp

public class MLClassifyOp  
extends java.lang.Object  
implements [Operator](#)

최대 우도법에 의한 분류를 실행한다.

## Constructor Summary

MLClassifyOp() Constructor
-------------------------------

## Method Summary

void	<a href="#">operate()</a> 최대우도법 분류를 실행한다.
void	<a href="#">setBands(int[] bands)</a> 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리영역을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setStatisticsFile(java.io.File f)</a> 통계 정보를 갖고 있는 파일을 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

## Constructor Detail

### MLClassifyOp

public MLClassifyOp()

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

### setStatisticsFile

```
public void setStatisticsFile(java.io.File f)
    throws java.io.IOException,
           IllegalFormatException,
           VisualNumerics.math.MathException
```

통계 정보를 갖고 있는 파일을 설정한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException  
VisualNumerics.math.MathException – MathException

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,
                    Target dtgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()
    throws java.io.IOException,
           java.lang.InterruptedException
```

최대우도법 분류를 실행한다.

Throws:

java.io.IOException – IOException

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class MultiCubeMergeOp

java.lang.Object  
|  
+--MultiCubeMergeOp

public class MultiCubeMergeOp  
extends java.lang.Object  
implements [Operator](#)

여러 분광대를 갖는 복수영상을 정렬하여 하나의 영상으로 재 구성한다. 입력 영상의 화소형, 행, 열의 크기는 전부 동일하여야 한다.

Constructor Summary	
<a href="#">MultiCubeMergeOp()</a>	Constructor

Method Summary	
void	<a href="#">operate()</a> 영상의 merge를 실행한다.
void	<a href="#">setBands(int[][] bandA, int[] order)</a> 대상 분광대를 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target[] srcTgtA, Target destTgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### MultiCubeMergeOp

public MultiCubeMergeOp()  
Constructor

## Method Detail

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

## setTarget

```
public void setTarget(Target[] srcTgtA,  
                     Target destTgt)
```

대상 영상을 설정한다.

---

## setBands

```
public void setBands(int[][] bandA,  
                   int[] order)
```

대상 분광대를 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상의 merge를 실행한다.

Throws:

[java.io.IOException](#) – [IOException](#)

[java.lang.InterruptedException](#) – [InterruptedException](#)

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class MultiImageComposeOp

```
java.lang.Object
|
|--MultiImageComposeOp
```

```
public class MultiImageComposeOp
extends java.lang.Object
implements Operator
```

영상을 다른 영상에 끼워 넣기를 한다. 여기서 출력 영상은 메모리형이어야 하며 화소형은 동일하여야 한다.

Constructor Summary	
<a href="#">MultiImageComposeOp()</a>	Constructor

Method Summary	
void	<a href="#">operate()</a> 영상 삽입을 실행한다.
void	<a href="#">setBands(int[] bands)</a> 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리영역을 설정한다.
void	<a href="#">setOpacity(float o)</a> 기존의 화소값에 투명도를 조절하기 위한 가중치를 설정한다.
void	<a href="#">setPosition(int s, int l, int b)</a> 영상을 삽입하기 위한 시작 위치를 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setRotate(float r)</a> 영상을 임의의 각도로 회전시켜 삽입하기 위한 회전각을 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



## Constructor Detail

### MultiImageComposeOp

```
public MultiImageComposeOp()
```

Constructor

## Method Detail

### setPosition

```
public void setPosition(int s,  
                       int l,  
                       int b)
```

영상을 삽입하기 위한 시작 위치를 설정한다.

---

### setRotate

```
public void setRotate(float r)
```

영상을 임의의 각도로 회전시켜 삽입하기 위한 회전각을 설정한다.

---

### setOpacity

```
public void setOpacity(float o)
```

기존의 화소값에 투명도를 조절하기 위한 가중치를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상 삽입을 실행한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class NullImageMakeOp

```
java.lang.Object
|
+--NullImageMakeOp
```

```
public class NullImageMakeOp
extends java.lang.Object
implements Operator
```

영상을 초기화한 빈 영상을 작성한다.

### Constructor Summary

[NullImageMakeOp\(\)](#)

### Method Summary

void	<a href="#">operate()</a> 빈 영상을 작성한다.
void	<a href="#">setBackgroundValue(float f)</a> 배경의 화소값을 설정한다.
void	<a href="#">setBands(int b)</a> 분광대 수를 설정한다.
void	<a href="#">setBounds(int s, int l)</a> 영상의 크기를 설정한다.
void	<a href="#">setPixelType(java.lang.String pt)</a> 영상의 화소형을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target dtgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

[NullImageMakeOp](#)

```
public NullImageMakeOp()
```

## Method Detail

### setBounds

```
public void setBounds(int s,  
                     int l)
```

영상의 크기를 설정한다.

---

### setBands

```
public void setBands(int b)
```

분광대 수를 설정한다.

---

### setBackgroundValue

```
public void setBackgroundValue(float f)
```

배경의 화소값을 설정한다.

---

### setPixelType

```
public void setPixelType(java.lang.String pt)
```

영상의 화소형을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(Target dtgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

빈 영상을 작성한다.

Throws:

[java.io.IOException](#) - IOException

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class OutlierKernel

java.lang.Object  
|  
+--OutlierKernel

public class OutlierKernel  
extends java.lang.Object

Outlier Kernel을 관리한다.

Constructor Summary	
<code>OutlierKernel()</code>	Constructor

Method Summary	
int	<code>getDivisor()</code> 큰분류선 수행에 대한 분모 연산자를 되돌린다.
int[]	<code>getKernel()</code> Outlier Kernel을 되돌린다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### OutlierKernel

public OutlierKernel()  
Constructor

### Method Detail

#### getKernel

public int[] getKernel()  
Outlier Kernel을 되돌린다.

public int getDivisor()

콘볼루션 수행에 대한 분모 연산자를 되돌린다.

---

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

FRAMES NO FRAMES

SUMMARY: INNER | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

---

---

## Class ParmHighPassKernel

```
java.lang.Object
|
+--ParmHighPassKernel
```

```
public class ParmHighPassKernel
extends java.lang.Object
```

Parametric High Pass Filter Kernel 클래스이다.

### Constructor Summary

[ParmHighPassKernel\(\)](#)  
 Constructor

[ParmHighPassKernel\(int b\)](#)  
 Constructor

### Method Summary

int	<a href="#">getDivisor()</a> 콘볼루션 수행에 대한 분모 연산자를 되돌린다.
int[]	<a href="#">getKernel()</a> 커널 배열을 되돌린다.
int	<a href="#">getParm()</a> 가중치 parameter값을 되돌린다.
void	<a href="#">setParm(int b)</a> 가중치 parameter를 설정한다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

#### ParmHighPassKernel

```
public ParmHighPassKernel()
    Constructor
```

#### ParmHighPassKernel



```
public ParmHighPassKernel(int b)
```

Constructor

## Method Detail

### setParm

```
public void setParm(int b)
```

가중치 parameter를 설정한다.

---

### getParm

```
public int getParm()
```

가중치 parameter값을 되돌린다.

---

### getKernel

```
public int[] getKernel()
```

커널 배열을 되돌린다.

---

### getDivisor

```
public int getDivisor()
```

콘블루션 수행에 대한 분모 연산자를 되돌린다.

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class ParmLowPassKernel

```

java.lang.Object
|
+--ParmLowPassKernel

```

```

public class ParmLowPassKernel
extends java.lang.Object

```

Parametric Low Pass Filter Kernel 클래스이다.

Constructor Summary	
<code>ParmLowPassKernel()</code>	Constructor
<code>ParmLowPassKernel(int b)</code>	Constructor

Method Summary	
int	<code>getDivisor()</code> 콘볼루션 수행에 대한 분모 연산자를 되돌린다.
int[]	<code>getKernel()</code> 커널 배열을 되돌린다.
int	<code>getParam()</code> 가중치 parameter 값을 되돌린다.
void	<code>setParam(int b)</code> 가중치 parameter를 설정한다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail
<b>ParmLowPassKernel</b>
public ParmLowPassKernel() Constructor

ParmLowPassKernel

public ParmLowPassKernel(int b)

Constructor

## Method Detail

### setParm

public void setParm(int b)

가중치 parameter를 설정한다.

---

### getParm

public int getParm()

가중치 parameter값을 되돌린다.

---

### getKernel

public int[] getKernel()

커널 배열을 되돌린다.

---

### getDivisor

public int getDivisor()

콘블루션 수행에 대한 분모 연산자를 되돌린다.

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class PCATransformOp

java.lang.Object  
|  
+--PCATransformOp

public class PCATransformOp  
extends java.lang.Object  
implements [Operator](#)

주성분 분석을 실행한다.

### Constructor Summary

<a href="#">PCATransformOp()</a> Constructor
---

### Method Summary

void	<a href="#">operate()</a> 주성분 분석을 실행한다.
void	<a href="#">setBands(int[] bands)</a> 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리영역을 설정한다.
void	<a href="#">setEigenvector(double[][][] eigen)</a> 고유벡터 행렬을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### PCATransformOp

public PCATransformOp()

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

대상 영상을 설정한다.

---

### setEigenVector

```
public void setEigenVector(double[][][] eigen)
```

고유벡터 행렬을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

주성분 분석을 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

### stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

FRAMES NO FRAMES

SUMMARY: INNER | FIELD | CONSTR | METHOD

DETAIL: FIELD | CONSTR | METHOD

---

## Class PiecewiseEnhanceOp

```
java.lang.Object
|
+--PiecewiseEnhanceOp
```

```
public class PiecewiseEnhanceOp
extends java.lang.Object
implements Operator
```

사용자 정의에 의한 변환표를 이용하여 영상 향상을 한다.

### Constructor Summary

<a href="#">PiecewiseEnhanceOp()</a> Constructor
---

### Method Summary

protected void	<a href="#">changePixelValue</a> (byte[] a, int aOffset, int index, byte[] b, int bOffset, int length)
void	<a href="#">operate</a> () 단계적 영상향상을 실행한다.
void	<a href="#">setBands</a> (int[] bands) 대상 분광대를 설정한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle b) 처리영역을 설정한다.
void	<a href="#">setLUT</a> (int[][] lut) 영상 향상 변환표를 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target sgt, Target dtgt) 대상 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

[PiecewiseEnhanceOp](#)

public PiecewiseEnhanceOp()

Constructor

## Method Detail

### setBands

public void setBands(int[] bands)

대상 분광대를 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle b)

처리영역을 설정한다.

---

### setLUT

public void setLUT(int[][] lut)

영상 향상 변환표를 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

public void setTarget(Target stgt,  
Target dtgt)

대상 영상을 설정한다.

---

### operate

public void operate()  
throws java.io.IOException,  
java.lang.InterruptedException

단계적 영상향상을 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---



## changePixelValue

```
protected void changePixelValue(byte[] a,  
                                int aOffset,  
                                int index,  
                                byte[] b,  
                                int bOffset,  
                                int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class PixelInteractionViewOp

```
java.lang.Object
|
+--PixelInteractionViewOp
```

```
public class PixelInteractionViewOp
extends java.lang.Object
implements Operator
```

입력 위치로 부터 9x9의 화소값을 숫자로 표현한다.

Field Summary	
java.lang.String[][]	<a href="#">PixelTable</a>

Constructor Summary	
<a href="#">PixelInteractionViewOp()</a>	Constructor

Method Summary	
java.lang.String[][]	<a href="#">getPixelValue()</a> 지정된 위치에서의 화소값 배열을 되돌린다.
void	<a href="#">operate()</a> 화소값을 읽어 배열에 저장한다.
void	<a href="#">setBand(int bnd)</a> 대상 분광대를 설정한다.
void	<a href="#">setParameter(int smps, int lns)</a> 화소값을 가져올 영상에서의 위치를 설정한다.
void	<a href="#">setParameter(int bnds, int smps, int lns)</a> 화소값을 가져올 영상에서의 위치를 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target tgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

## Field Detail

### PixelTable

```
public java.lang.String[][] PixelTable
```

## Constructor Detail

### PixelInteractionViewOp

```
public PixelInteractionViewOp()
```

Constructor

## Method Detail

### setBand

```
public void setBand(int bnd)
```

대상 분광대를 설정한다.

---

### setParameter

```
public void setParameter(int smps,  
                          int lns)
```

화소값을 가져올 영상에서의 위치를 설정한다.

---

### setParameter

```
public void setParameter(int bnds,  
                          int smps,  
                          int lns)
```

화소값을 가져올 영상에서의 위치를 설정한다.

---

### setTarget

```
public void setTarget(Target tgt)
```

대상 영상을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## getPixelValue

```
public java.lang.String[][] getPixelValue()
```

지정된 위치에서의 화소값 배열을 되돌린다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

화소값을 읽어 배열에 저장한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

### [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class PixelSpectrumViewOp

```
java.lang.Object
|
+--PixelSpectrumViewOp
```

```
public class PixelSpectrumViewOp
extends java.lang.Object
implements Operator
```

각 분광대의 화소값을 표현한다.

Field Summary	
java.lang.String[][][]	<a href="#">PixelTable</a>

Constructor Summary	
<a href="#">PixelSpectrumViewOp()</a>	Constructor

Method Summary	
java.lang.String[][][]	<a href="#">getPixelValue()</a> 지정된 위치에서의 화소값 배열을 되돌린다.
void	<a href="#">operate()</a> 화소값을 읽어 배열에 저장한다.
void	<a href="#">setBands(int[] bands)</a> 대상 분광대를 설정한다.
void	<a href="#">setParameter(int smps, int lns)</a> 화소값을 가져올 영상에서의 위치를 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target tgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### PixelTable

```
public java.lang.String[][][] PixelTable
```

## Constructor Detail

### PixelSpectrumViewOp

```
public PixelSpectrumViewOp()
```

Constructor

## Method Detail

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

### setParameter

```
public void setParameter(int smps,  
                          int lns)
```

화소값을 가져올 영상에서의 위치를 설정한다.

---

### setTarget

```
public void setTarget(Target tgt)
```

대상 영상을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### getPixelValue

```
public java.lang.String[][][] getPixelValue()
```

지정된 위치에서의 화소값 배열을 되돌린다.

---

### operate

```
public void operate()
```

throws java.io.IOException,  
java.lang.InterruptedException

화소값을 읽어 배열에 저장한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class PixelTypeConvertOp

```
java.lang.Object  
|  
+--PixelTypeConvertOp
```

```
public class PixelTypeConvertOp  
extends java.lang.Object  
implements Operator
```

입력 화소형을 다른 화소형으로 변환한다.

### Constructor Summary

<a href="#">PixelTypeConvertOp()</a> Constructor
---

### Method Summary

void	<a href="#">operate()</a> 화소형 변환을 실행한다.
void	<a href="#">setBands(int[] bands)</a> 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리 영역을 설정한다.
void	<a href="#">setPixelType(int type)</a> 출력 화소형을 설정한다.
void	<a href="#">setPixelType(java.lang.String type)</a> 출력 화소형을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target srcTgt, Target destTgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

[PixelTypeConvertOp](#)



public PixelTypeConvertOp()

Constructor

## Method Detail

### setPixelType

public void setPixelType(int type)

출력 화소형을 설정한다.

---

### setPixelType

public void setPixelType(java.lang.String type)

출력 화소형을 설정한다.

---

### setTarget

public void setTarget(Target srcTgt,  
                  · Target destTgt)

대상 영상을 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle r)

처리 영역을 설정한다.

---

### setBands

public void setBands(int[] bands)

대상 분광대를 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### operate

public void operate()  
          throws java.io.IOException,  
                  java.lang.InterruptedException

화소형 변환을 실행한다.

**Throws:**

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

**stop**

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

**Specified by:**

stop in interface Operator

---

Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class PrewittKernel

```
java.lang.Object
|
+--PrewittKernel
```

```
public class PrewittKernel
extends java.lang.Object
```

Prewitt Kernel를 관리한다.

Constructor Summary	
<code>PrewittKernel</code>	( <code>java.lang.String</code> method) Constructor

Method Summary	
<code>int</code>	<code>getDivisor()</code> 콘블루션 수행에 대한 분모 연산자를 되돌린다.
<code>int[]</code>	<code>getEKernel()</code> Prewitt(E)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getKernel()</code> 현재 설정되어 있는 Prewitt Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getNEKernel()</code> Prewitt(NE)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getNKernel()</code> Prewitt(N)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getNWKernel()</code> Prewitt(NW)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getSEKernel()</code> Prewitt(SE)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getSKernel()</code> Prewitt(S)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getSWKernel()</code> Prewitt(SW)Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getWKernel()</code> Prewitt(W)Kernel 배열의 가중치를 되돌린다.

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

## Constructor Detail

### PrewittKernel

```
public PrewittKernel(java.lang.String method)
```

Constructor

## Method Detail

### getKernel

```
public int[] getKernel()
```

현재 설정되어 있는 Prewitt Kernel 배열의 가중치를 되돌린다.

---

### getNKernel

```
public int[] getNKernel()
```

Prewitt(N)Kernel 배열의 가중치를 되돌린다.

---

### getSKernel

```
public int[] getSKernel()
```

Prewitt(S)Kernel 배열의 가중치를 되돌린다.

---

### getEKernel

```
public int[] getEKernel()
```

Prewitt(E)Kernel 배열의 가중치를 되돌린다.

---

### getWKernel

```
public int[] getWKernel()
```

Prewitt(W)Kernel 배열의 가중치를 되돌린다.

---

### getNWKernel

```
public int[] getNWKernel()
```

Prewitt(NW)Kernel 배열의 가중치를 되돌린다.

---

## getNEKernel

```
public int[] getNEKernel()
```

Prewitt(NE)Kernel 배열의 가중치를 되돌린다.

---

## getSWKernel

```
public int[] getSWKernel()
```

Prewitt(SW)Kernel 배열의 가중치를 되돌린다.

---

## getSEKernel

```
public int[] getSEKernel()
```

Prewitt(SE)Kernel 배열의 가중치를 되돌린다.

---

## getDivisor

```
public int getDivisor()
```

콘볼루션 수행에 대한 분모 연산자를 되돌린다.

---

### [Class](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Regressor

```
java.lang.Object
|
+--Regressor
```

Direct Known Subclasses:  
[GcpRegressor](#)

```
public class Regressor
extends java.lang.Object
```

X좌표에서 Y좌표로의 변환을 위한 회귀식을 구한다.

Constructor Summary	
<a href="#">Regressor()</a>	Constructor
<a href="#">Regressor(double[][] X, double[][] Y)</a>	Constructor

Method Summary	
double[][]	<a href="#">getAbsoluteError()</a> 절대 오차를 되돌린다.
double[][]	<a href="#">getCoef()</a> 계수를 되돌린다.
double[][]	<a href="#">getError()</a> 오차를 되돌린다.
double[][]	<a href="#">getEstimate()</a>
void	<a href="#">regress()</a> 회귀식을 계산한다.
void	<a href="#">setVar(double[][] X, double[][] Y)</a> 대상 행렬을 설정한다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### Regressor

`public Regressor()`

Constructor

---

## Regressor

`public Regressor(double[][] X,  
double[][] Y)`

Constructor

## Method Detail

### setVar

`public void setVar(double[][] X,  
double[][] Y)`

대상 행열을 설정한다.

---

### regress

`public void regress()  
throws VisualNumerics.math.MathException`

회귀식을 계산한다.

---

### getCoef

`public double[][] getCoef()`

계수를 되돌린다.

---

### getEstimate

`public double[][] getEstimate()`

---

### getError

`public double[][] getError()`

오차를 되돌린다.

---

### getAbsoluteError

`public double[][] getAbsoluteError()`

절대 오차를 되돌린다.

---

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

---



## Class ScattergramOp

```
java.lang.Object
|
+--ScattergramOp
```

```
public class ScattergramOp
extends java.lang.Object
implements Operator
```

스캐터그램을 만든다.

Field Summary	
int	<a href="#">Band1</a>
int	<a href="#">Band2</a>
int[][]	<a href="#">Freq</a>
int	<a href="#">IntervalsX</a>
int	<a href="#">IntervalsY</a>
int	<a href="#">maxFreq</a>
float	<a href="#">StartX</a>
float	<a href="#">StartY</a>
java.lang.String	<a href="#">title</a>
java.lang.StringBuffer	<a href="#">tmpTitle</a>
float	<a href="#">WidthX</a>
float	<a href="#">WidthY</a>

Constructor Summary	
<a href="#">ScattergramOp()</a>	Constructor

## Method Summary

void	<u>clear()</u> 스캐터그램 배열을 초기화한다.
float[][]	<u>getBound()</u> 스캐터그램의 화소값 영역을 되돌린다.
int[][]	<u>getFreq()</u> 스캐터그램을 되돌린다.
int	<u>getFreq(float valueX, float valueY)</u> 화소값 x, y에 대한 빈도수를 되돌린다.
int	<u>getFreq(int indexX, int indexY)</u> 인덱스 x, y에 대한 빈도수를 되돌린다.
int	<u>getMaxFreq()</u> 스캐터그램에서 최대 빈도수를 되돌린다.
java.lang.String	<u>getTitle()</u> 스캐터그램의 타이틀을 되돌린다.
int	<u>getTotalFreq()</u> 전체 빈도수를 되돌린다.
void	<u>operate()</u> 스캐터그램 작성을 실행한다.
void	<u>read(java.lang.String path)</u> 스캐터그램을 파일에서 입력한다.
void	<u>regionByEnd(float start, float end, int intervals, boolean flg)</u> 화소값의 영역을 설정한다.
void	<u>regionByWidth(float start, float width, int intervals, boolean flg)</u> 화소값의 영역을 설정한다.
void	<u>setBands(int b1, int b2)</u> 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리 영역을 설정한다.
void	<u>setMaxFreq()</u> 스캐터그램에서 최대 빈도수를 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setRegion(float[] r1, float[] r2)</u> 스캐터그램을 작성을 화소값의 영역을 설정한다.
void	<u>setScattergramFileName(java.lang.String f)</u> 스캐터그램을 저장할 파일을 설정한다.
void	<u>setTarget(Target tgt1, Target tgt2)</u> 대상 영상을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.
void	<u>write(java.lang.String path)</u> 스캐터그램을 파일로 출력한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### StartX

```
public float StartX
```

---

### StartY

```
public float StartY
```

---

### WidthX

```
public float WidthX
```

---

### WidthY

```
public float WidthY
```

---

### IntervalsX

```
public int IntervalsX
```

---

### IntervalsY

```
public int IntervalsY
```

---

### Band1

```
public int Band1
```

---

### Band2

```
public int Band2
```

---

### Freq

```
public int[][] Freq
```

---

### maxFreq

```
public int maxFreq
```

---

## title

```
public java.lang.String title
```

---

## tmpTitle

```
public java.lang.StringBuffer tmpTitle
```

## Constructor Detail

### ScattergramOp

```
public ScattergramOp()
```

Constructor

## Method Detail

### setBands

```
public void setBands(int b1,  
                    int b2)
```

분광대를 설정한다.

---

### setTarget

```
public void setTarget(Target tgt1,  
                    Target tgt2)
```

대상 영상을 설정한다.

---

### setScattergramFileName

```
public void setScattergramFileName(java.lang.String f)
```

스캐터그램을 저장할 파일을 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setRegion

```
public void setRegion(float[] r1,  
                     float[] r2)  
    throws IllegalArgumentException
```

스캐터그램을 작성을 화소값의 영역을 설정한다.

---

## regionByEnd

```
public void regionByEnd(float start,  
                       float end,  
                       int intervals,  
                       boolean flg)  
    throws IllegalArgumentException
```

화소값의 영역을 설정한다.

---

## regionByWidth

```
public void regionByWidth(float start,  
                          float width,  
                          int intervals,  
                          boolean flg)  
    throws IllegalArgumentException
```

화소값의 영역을 설정한다.

---

## getBound

```
public float[][] getBound()
```

스캐터그램의 화소값 영역을 되돌린다.

---

## getTitle

```
public java.lang.String getTitle()
```

스캐터그램의 타이틀을 되돌린다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

스캐터그램 작성을 실행한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

---

## getTotalFreq

```
public int getTotalFreq()
```

전체 빈도수를 되돌린다.

---

## setMaxFreq

```
public void setMaxFreq()
```

스캐터그램에서 최대 빈도수를 설정한다.

---

## getMaxFreq

```
public int getMaxFreq()
```

스캐터그램에서 최대 빈도수를 되돌린다.

---

## getFreq

```
public int[][] getFreq()
```

스캐터그램을 되돌린다.

---

## getFreq

```
public int getFreq(int indexX,  
                  int indexY)
```

인덱스 x, y에 대한 빈도수를 되돌린다.

---

## getFreq

```
public int getFreq(float valueX,  
                  float valueY)
```

화소값 x, y에 대한 빈도수를 되돌린다.

---

## read

```
public void read(java.lang.String path)  
              throws java.io.IOException
```

스캐터그램을 파일에서 입력한다.

---

## write

```
public void write(java.lang.String path)  
              throws java.io.IOException
```

스캐터그램을 파일로 출력한다.

---

## clear

public void clear()

스캐터그램 배열을 초기화한다.

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class SceneRegressNormalizeOp

```
java.lang.Object
|
+--SceneRegressNormalizeOp
```

```
public class SceneRegressNormalizeOp
    extends java.lang.Object
    implements Operator
```

통계적 방법에 의한 영상 정규화를 실행한다. 이를 위하여는 두 영상에서 밝은 지역, 어두운 지역을 연결하는 직선식을 구하거나 영상 전체에 대한 평균, 표준편차를 이용한 식을 작성하여야 한다.

Constructor Summary	
<a href="#">SceneRegressNormalizeOp()</a>	Constructor

Method Summary	
void	<a href="#">operate()</a> 영상 정규화를 실행한다.
void	<a href="#">setBand(int b)</a> 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리 영역을 설정한다.
void	<a href="#">setParameter(float s, float o)</a> 직선식을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail
<a href="#">SceneRegressNormalizeOp</a>



```
public SceneRegressNormalizeOp()
```

Constructor

## Method Detail

### setBand

```
public void setBand(int b)
```

분광대를 설정한다.

---

### setParameter

```
public void setParameter(float s,  
                          float o)
```

직선식을 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상 정규화를 실행한다.

Throws:

[java.io.IOException](#) – [IOException](#)

[java.lang.InterruptedException](#) – [InterruptedException](#)

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

---

### [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

## Class SobelKernel

```
java.lang.Object
|
+--SobelKernel
```

```
public class SobelKernel
extends java.lang.Object
```

Sobel Kernel를 관리한다.

Constructor Summary	
<code>SobelKernel</code>	( <code>java.lang.String</code> method) Constructor

Method Summary	
<code>int</code>	<code>getDivisor()</code> 콘볼루션 수행에 대한 분모 연산자를 되돌린다.
<code>int[]</code>	<code>getDKernel()</code> Sobel(D) Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getKernel()</code> 현재 설정되어 있는 Sobel Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getLKernel()</code> Sobel(L) Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getRKernel()</code> Sobel(R) Kernel 배열의 가중치를 되돌린다.
<code>int[]</code>	<code>getUKernel()</code> Sobel(U) Kernel 배열의 가중치를 되돌린다.

Methods inherited from class java.lang.Object	
<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>	

Constructor Detail	
<code>SobelKernel</code>	
<code>public SobelKernel</code> ( <code>java.lang.String</code> method)	
Constructor	

## Method Detail

### getKernel

```
public int[] getKernel()
```

현재 설정되어 있는 Sobel Kernel 배열의 가중치를 되돌린다.

---

### getUKernel

```
public int[] getUKernel()
```

Sobel(U)Kernel 배열의 가중치를 되돌린다.

---

### getDKernel

```
public int[] getDKernel()
```

Sobel(D)Kernel 배열의 가중치를 되돌린다.

---

### getLKernel

```
public int[] getLKernel()
```

Sobel(L)Kernel 배열의 가중치를 되돌린다.

---

### getRKernel

```
public int[] getRKernel()
```

Sobel(R)Kernel 배열의 가중치를 되돌린다.

---

### getDivisor

```
public int getDivisor()
```

콘볼루션 수행에 대한 분모 연산자를 되돌린다.

---

[Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

## Class SpatialKernelFilterOp

java.lang.Object  
|  
+--SpatialKernelFilterOp

---

```
public class SpatialKernelFilterOp  
extends java.lang.Object  
implements Operator
```

미리 정의된 커널을 이용한 필터링을 수행한다.

---

### Field Summary

static int	<a href="#">EDGE_NO_OP</a>
static int	<a href="#">EDGE_ZERO_FILL</a>

### Constructor Summary

<a href="#">SpatialKernelFilterOp()</a> Constructor
--

## Method Summary

int	<u>getEdgeHint()</u> 엣지 처리와 관련된 내용을 되돌린다.
GeoKernel	<u>getKernel()</u> 현재 설정되어 있는 커널을 되돌린다.
void	<u>operate()</u> 필터링을 실행한다.
void	<u>setBand(int b)</u> 대상 분광대를 설정한다.
void	<u>setBounds(java.awt.Rectangle r)</u> 처리 영역을 설정한다.
void	<u>setDivisor(int b)</u> divisor를 설정한다.
void	<u>setEdgeHint(int h)</u> 엣지 처리와 관련된 내용을 설정한다.
void	<u>setKernel(GeoKernel k, int d)</u> 커널을 설정한다.
void	<u>setProgressStream(ProgressStream prog)</u> 프로그램 진행 상태바를 설정한다.
void	<u>setTarget(Target stgt, Target dtgt)</u> 대상 영상을 설정한다.
void	<u>stop()</u> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### EDGE\_ZERO\_FILL

public static final int EDGE\_ZERO\_FILL

### EDGE\_NO\_OP

public static final int EDGE\_NO\_OP

## Constructor Detail

### SpatialKernelFilterOp

public SpatialKernelFilterOp()

Constructor

## Method Detail

### setKernel

```
public void setKernel(GeoKernel k,  
                    int d)
```

커널을 설정한다.

---

## getEdgeHint

```
public int getEdgeHint()
```

엣지 처리와 관련된 내용을 되돌린다.

---

## setEdgeHint

```
public void setEdgeHint(int h)
```

엣지 처리와 관련된 내용을 설정한다.

---

## getKernel

```
public GeoKernel getKernel()
```

현재 설정되어 있는 커널을 되돌린다.

---

## setBand

```
public void setBand(int b)
```

대상 분광대를 설정한다.

---

## setDivisor

```
public void setDivisor(int b)
```

divisor를 설정한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedExcepion
```

필터링을 실행한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedExcepion – InterruptedExcepion

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---



## Class SpatialProfileViewOp

```
java.lang.Object
|
+--SpatialProfileViewOp
```

```
public class SpatialProfileViewOp
extends java.lang.Object
implements Operator
```

영상의 주어진 선을 따른 분광대 별 화소값의 변화를 출력한다.

Field Summary	
float[][]	<a href="#">PixelTable</a>

Constructor Summary	
SpatialProfileViewOp()	Constructor

Method Summary	
int[][]	<a href="#">getPixelIndex()</a> 각 화소값의 x, y좌표를 되돌린다.
float[][]	<a href="#">getPixelValue()</a> 화소값을 되돌린다.
void	<a href="#">operate()</a> 두점사이의 분광 프로파일을 작성한다.
void	<a href="#">setParameter(int[] p)</a> 선을 정의 하기 위한 두 점을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target tgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object	
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait	

## Field Detail

### PixelTable

```
public float[][] PixelTable
```

## Constructor Detail

### SpatialProfileViewOp

```
public SpatialProfileViewOp()
```

Constructor

## Method Detail

### setParameter

```
public void setParameter(int[] p)
```

선을 정의 하기 위한 두 점을 설정한다.

---

### setTarget

```
public void setTarget(Target tgt)
```

대상 영상을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### getPixelValue

```
public float[][] getPixelValue()
```

화소값을 되돌린다.

---

### getPixelIndex

```
public int[][] getPixelIndex()
```

각 화소값의 x, y좌표를 되돌린다.

---

### operate

```
public void operate()  
    throws java.io.IOException,
```

java.lang.InterruptedEception

두점사이의 분광 프로파일을 작성한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedEception – InterruptedEception

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class SpectrumProfileViewOp

java.lang.Object  
|  
+--SpectrumProfileViewOp

public class SpectrumProfileViewOp  
extends java.lang.Object  
implements [Operator](#)

영상의 주어진 선을 따른 분광대 별 화소값의 변화를 출력한다.

### Field Summary

float[][][]	<a href="#">PixelTable</a>
-------------	----------------------------

### Constructor Summary

<a href="#">SpectrumProfileViewOp()</a> Constructor
--

### Method Summary

float[][][]	<a href="#">getPixelValue()</a> 분광대별 화소값을 되돌린다.
void	<a href="#">operate()</a> 분광 프로파일을 작성한다.
void	<a href="#">setBands(int[] bands)</a> 대상 분광대를 설정한다.
void	<a href="#">setParameter(int smps, int lns)</a> 화소의 위치를 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target tgt)</a> 대상 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### PixelTable

```
public float[][][] PixelTable
```

## Constructor Detail

### SpectrumProfileViewOp

```
public SpectrumProfileViewOp()
```

Constructor

## Method Detail

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

### setParameter

```
public void setParameter(int smps,  
                          int lns)
```

화소의 위치를 설정한다.

---

### setTarget

```
public void setTarget(Target tgt)
```

대상 영상을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### getPixelValue

```
public float[][][] getPixelValue()
```

분광대별 화소값을 되돌린다.

---

### operate

```
public void operate()
```

throws java.io.IOException,  
java.lang.InterruptedException

분광 프로파일을 작성한다.

Throws:

java.io.IOException – IOException

java.lang.InterruptedException – InterruptedException

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) · [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class SPOTHRVImportOp

```
java.lang.Object
|
+--SPOTHRVImportOp
```

```
public class SPOTHRVImportOp
extends java.lang.Object
implements Operator
```

SPOT HRV 영상을 읽어 내부포맷으로 변환한다.

## Constructor Summary

<code>SPOTHRVImportOp()</code> Constructor
---

## Method Summary

void	<code>operate()</code> SPOT HRV 영상을 내부 포맷으로 변환한다.
void	<code>setBands(int[] bands)</code> 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setHeader(SPOTHRVHeader hd)</code> SPOT HRV의 헤더를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(java.io.File f, Target.File tgt)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### SPOTHRVImportOp

```
public SPOTHRVImportOp()
```

## Method Detail

### setHeader

```
public void setHeader(SPOTHRVHeader hd)
```

SPOT HRV의 헤더를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

분광대를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(java.io.File f,  
                     Target.File tgt)
```

대상 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

SPOT HRV 영상을 내부 포맷으로 변환한다.

Throws:

[java.io.IOException](#) – [IOException](#)

[java.lang.InterruptedException](#) – [InterruptedException](#)

---

### stop

```
public void stop()
```



사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
[SUMMARY: INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
[DETAIL: FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class `StatisticCalculateOp`

```
java.lang.Object
|
+--StatisticCalculateOp
```

```
public class StatisticCalculateOp
extends java.lang.Object
implements Operator
```

영상의 통계를 계산한다

### Constructor Summary

<code>StatisticCalculateOp()</code> Constructor
--

### Method Summary

Statistics	<code>getStatistics()</code> 통계 결과를 되돌린다.
void	<code>operate()</code> 영상에 대한 통계를 구한다.
void	<code>setBands(int[] band)</code> 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setStatisticsFile(java.io.File f)</code> 통계 결과를 저장할 파일을 설정한다.
void	<code>setStatisticsFile(java.lang.String path)</code> 통계 결과를 저장할 파일을 설정한다.
void	<code>setTarget(Target tgt)</code> 대상 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### StatisticCalculateOp

```
public StatisticCalculateOp()
```

Constructor

## Method Detail

### setStatisticsFile

```
public void setStatisticsFile(java.lang.String path)
```

통계 결과를 저장할 파일을 설정한다.

---

### setStatisticsFile

```
public void setStatisticsFile(java.io.File f)
```

통계 결과를 저장할 파일을 설정한다.

---

### setBands

```
public void setBands(int[] band)
```

대상 분광대를 설정한다.

---

### getStatistics

```
public Statistics getStatistics()
```

통계 결과를 되돌린다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setTarget

```
public void setTarget(Target tgt)
```

대상 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

영상에 대한 통계를 구한다.

Throws:

java.io.IOException – IOException  
java.lang.InterruptedException – InterruptedException

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Statistics

```
java.lang.Object
|
+--Statistics
```

---

```
public class Statistics
extends java.lang.Object
```

통계를 관리하는 클래스이다.

---

### Constructor Summary

<code>Statistics()</code> Constructor
--

<code>Statistics(int classes, int bands)</code> Constructor
--

## Method Summary

double	<u>getCorrel</u> (int c, int i, int j) 클래스의 분광대에 대한 공상관 값을 되돌린다.
double	<u>getCovar</u> (int c, int i, int j) 클래스의 분광대에 대한 공분산 값을 되돌린다.
double[][]	<u>getMean</u> () 평균을 되돌린다.
double	<u>getMean</u> (int c, int i) 클래스의 분광대에 대한 평균을 되돌린다.
int	<u>getNoOfBands</u> () 분광대 수를 되돌린다.
int	<u>getNoOfClasses</u> () 클래스 수를 되돌린다.
int	<u>getNoOfPixels</u> (int c) 클래스의 화소수를 되돌린다.
double	<u>getRange</u> (int c, int i) 클래스의 분광대에 대한 화소값 범위를 되돌린다.
double[][]	<u>getStd</u> () 표준 편차를 되돌린다.
double	<u>getStd</u> (int c, int i) 클래스의 분광대에 대한 표준편차를 되돌린다.
double	<u>getSum</u> (int c, int i) 클래스의 분광대에 대한 Sum을 되돌린다.
java.lang.String	<u>getTitle</u> () 타이틀을 되돌린다.
double	<u>getVar</u> (int c, int i) 클래스의 분광대에 대한 분산을 되돌린다.
void	<u>read</u> (java.io.File f) 통계를 파일에서 입력한다.
void	<u>read</u> (java.lang.String path) 통계를 파일에서 입력한다.
void	<u>setMax</u> (int c, int i, double max) 클래스의 분광대에 대한 최대값을 설정한다.
void	<u>setMin</u> (int c, int i, double min) 클래스의 분광대에 대한 최소값을 설정한다.
void	<u>setNoOfPixels</u> (int c, int n) 클래스의 화소수를 설정한다.
void	<u>setSumX</u> (int c, int i, double sum) 클래스와 분광대에 대한 Sum을 설정한다.
void	<u>setSumXY</u> (int c, int i, int j, double sumXY) 클래스의 분광대간의 Sum을 설정한다.
void	<u>setTitle</u> (java.lang.String s) 타이틀을 설정한다.
void	<u>write</u> (java.io.File f) 통계를 파일로 출력한다.
void	<u>write</u> (java.lang.String path) 통계를 파일로 출력한다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### Statistics

```
public Statistics()
```

Constructor

---

### Statistics

```
public Statistics(int classes,  
                 int bands)
```

Constructor

## Method Detail

### setTitle

```
public void setTitle(java.lang.String s)
```

타이틀을 설정한다.

---

### setSumX

```
public void setSumX(int c,  
                   int i,  
                   double sum)
```

클래스와 분광대에 대한 Sum을 설정한다.

---

### setSumXY

```
public void setSumXY(int c,  
                    int i,  
                    int j,  
                    double sumXY)
```

클래스의 분광대간의 Sum을 설정한다.

---

### setMin

```
public void setMin(int c,  
                  int i,  
                  double min)
```

클래스의 분광대에 대한 최소값을 설정한다.

---

## setMax

```
public void setMax(int c,  
                  int i,  
                  double max)
```

클래스의 분광대에 대한 최대값을 설정한다.

---

## setNoOfPixels

```
public void setNoOfPixels(int c,  
                           int n)
```

클래스의 화소수를 설정한다.

---

## getTitle

```
public java.lang.String getTitle()
```

타이틀을 되돌린다.

---

## getNoOfBands

```
public int getNoOfBands()
```

분광대 수를 되돌린다.

---

## getNoOfClasses

```
public int getNoOfClasses()
```

클래스 수를 되돌린다.

---

## getNoOfPixels

```
public int getNoOfPixels(int c)
```

클래스의 화소수를 되돌린다.

---

## getRange

```
public double getRange(int c,  
                       int i)
```



클래스의 분광대에 대한 화소값 범위를 되돌린다.

---

## getSum

```
public double getSum(int c,  
                    int i)
```

클래스의 분광대에 대한 Sum을 되돌린다.

---

## getMean

```
public double getMean(int c,  
                     int i)
```

클래스의 분광대에 대한 평균을 되돌린다.

---

## getMean

```
public double[][] getMean()
```

평균을 되돌린다.

---

## getStd

```
public double[][] getStd()
```

표준 편차를 되돌린다.

---

## getStd

```
public double getStd(int c,  
                    int i)
```

클래스의 분광대에 대한 표준편차를 되돌린다.

---

## getVar

```
public double getVar(int c,  
                    int i)
```

클래스의 분광대에 대한 분산을 되돌린다.

---

## getCovar

```
public double getCovar(int c,  
                      int i,  
                      int j)
```

클래스의 분광대에 대한 공분산 값을 되돌린다.

---

## getCorrel

```
public double getCorrel(int c,  
                      int i,  
                      int j)
```

클래스의 분광대에 대한 공상관 값을 되돌린다.

---

## read

```
public void read(java.lang.String path)  
    throws java.io.IOException,  
           IllegalFormatException
```

통계를 파일에서 입력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException

---

## read

```
public void read(java.io.File f)  
    throws java.io.IOException,  
           IllegalFormatException
```

통계를 파일에서 입력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException

---

## write

```
public void write(java.lang.String path)  
    throws java.io.IOException
```

통계를 파일로 출력한다.

Throws:

java.io.IOException – IOException  
IllegalFormatException – IllegalFormatException

---

## write

```
public void write(java.io.File f)  
    throws java.io.IOException
```

통계를 파일로 출력한다.

Throws:

[java.io.IOException](#) - IOException

[IllegalFormatException](#) - IllegalFormatException

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Target.File

```
java.lang.Object
|
+--Target
|
+--Target.File
```

public static class Target.File  
extends Target

파일 타입의 Target을 관리한다.

### Inner classes inherited from class Target

Target.File, Target.Memory

### Field Summary

java.io.File	<u>Pixel</u>
--------------	--------------

### Fields inherited from class Target

BAND MODE, CUBE MODE, HD

### Constructor Summary

Target.File()  
Constructor

Target.File(java.io.File f)  
Constructor

Target.File(java.lang.String path)  
Constructor

Target.File(java.lang.String dir, java.lang.String name)  
Constructor

Target.File(Target.File tgt)  
Constructor

## Method Summary

void	<code>dispose()</code> Target을 삭제한다.
boolean	<code>equals(java.lang.Object tgt)</code> 입력 파일과의 동일성 여부를 검사한다.
boolean	<code>existHDFFile()</code> 헤더 파일이 존재하는가를 검사한다.
protected java.lang.String	<code>getID()</code> 파일의 절대 경로를 되돌린다.
java.lang.String	<code>getName()</code> Target의 이름을 되돌린다.
java.lang.String	<code>getPathOfHDFFile()</code> 헤더 파일의 경로를 되돌린다.
Target.Memory	<code>readSubarea(java.awt.Rectangle r, int band)</code> Target으로 부터 정해진 영역을 읽는다.

## Methods inherited from class Target

`info`, `toString`

## Methods inherited from class java.lang.Object

`clone`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### Pixel

```
public java.io.File Pixel
```

## Constructor Detail

### Target.File

```
public Target.File()
```

Constructor

---

### Target.File

```
public Target.File(java.lang.String path)
```

Constructor

---

### Target.File

```
public Target.File(java.lang.String dir,  
                  java.lang.String name)
```

Constructor

**Parameters:**

dir – the name of directory  
filename – the name of file

---

## Target.File

public Target.File(java.io.File f)

Constructor

Parameters:

f – File Object

---

## Target.File

public Target.File(Target.File tgt)

Constructor

---

## Method Detail

### getName

public java.lang.String getName()

Target의 이름을 되돌린다.

Overrides:

getName in class Target

---

### readSubarea

public Target.Memory readSubarea(java.awt.Rectangle r,  
int band)

Target으로 부터 정해진 영역을 읽는다.

Overrides:

readSubarea in class Target

---

### getID

protected java.lang.String getID()

파일의 절대 경로를 되돌린다.

Overrides:

getID in class Target

---

### existHDFFile

public boolean existHDFFile()

헤더 파일이 존재하는가를 검사한다.

---

## getPathOfHDFFile

```
public java.lang.String getPathOfHDFFile()
```

헤더 파일의 경로를 되돌린다.

---

## dispose

```
public void dispose()
```

Target을 삭제한다.

Overrides:

[dispose](#) in class [Target](#)

---

## equals

```
public boolean equals(java.lang.Object tgt)
```

입력 파일과의 동일성 여부를 검사한다.

Overrides:

[equals](#) in class [java.lang.Object](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Target

```
java.lang.Object  
|  
+--Target
```

Direct Known Subclasses:  
[Target.File](#), [Target.Memory](#)

---

```
public abstract class Target  
extends java.lang.Object
```

Version:  
1.0  
Author:  
GeoPixel team

---

Inner Class Summary	
static class	<a href="#">Target.File</a> 파일 타입의 Target을 관리한다.
static class	<a href="#">Target.Memory</a> 메모리 타입의 Target을 관리한다.

Field Summary	
static int	<a href="#">BAND_MODE</a>
static int	<a href="#">CUBE_MODE</a>
<a href="#">CubeHeader</a>	<a href="#">HD</a>

Constructor Summary	
<a href="#">Target()</a>	



## Method Summary

abstract void	<code>dispose()</code>
protected abstract java.lang.String	<code>getID()</code>
abstract java.lang.String	<code>getName()</code>
java.lang.String	<code>info()</code> return the content of target header
abstract <u>Target.Memory</u>	<code>readSubarea(java.awt.Rectangle r, int band)</code>
java.lang.String	<code>toString()</code> Target의 이름을 되돌린다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### CUBE\_MODE

```
public static final int CUBE_MODE
```

---

### BAND\_MODE

```
public static final int BAND_MODE
```

---

### HD

```
public CubeHeader HD
```

## Constructor Detail

### Target

```
public Target()
```

## Method Detail

### getName

```
public abstract java.lang.String getName()
```

---

### readSubarea

```
public abstract Target.Memory readSubarea(java.awt.Rectangle r,  
int band)
```

---

## getID

protected abstract java.lang.String `getID()`

---

## dispose

public abstract void `dispose()`

---

## info

public java.lang.String `info()`

return the content of target header

---

## toString

public java.lang.String `toString()`

Target의 이름을 되돌린다.

**Overrides:**

`toString` in class java.lang.Object

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

## Class Target.Memory

```
java.lang.Object
|
+--Target
|
+--Target.Memory
```

---

```
public static class Target.Memory
extends Target
```

메모리 타입의 Target을 관리한다.

---

Inner classes inherited from class <a href="#">Target</a>
---

<a href="#">Target.File</a> , <a href="#">Target.Memory</a>
---

### Field Summary

java.lang.Object	<a href="#">Pixel</a>
------------------	-----------------------

Fields inherited from class <a href="#">Target</a>
--

<a href="#">BAND_MODE</a> , <a href="#">CUBE_MODE</a> , <a href="#">HD</a>
--

### Constructor Summary

<a href="#">Target.Memory</a> () Constructor
---

<a href="#">Target.Memory</a> (java.lang.String s) Constructor
---

## Method Summary

void	<code>dispose()</code> Target을 삭제한다.
void	<code>embody()</code> 영상을 저장할 메모리 영역을 확보한다.
boolean	<code>equals(java.lang.Object tgt)</code> 입력 파일과의 동일성 여부를 검사한다.
protected java.lang.String	<code>getID()</code> Target의 이름을 되돌린다.
java.lang.String	<code>getName()</code> Target의 이름을 되돌린다.
void	<code>initialize(double value)</code> 영상의 화소값을 초기화 한다.
Target.Memory	<code>readSubarea(java.awt.Rectangle r, int band)</code> Target으로 부터 정해진 영역을 읽는다.

## Methods inherited from class Target

`info`, `toString`

## Methods inherited from class java.lang.Object

`clone`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### Pixel

```
public java.lang.Object Pixel
```

## Constructor Detail

### Target.Memory

```
public Target.Memory()
```

Constructor

### Target.Memory

```
public Target.Memory(java.lang.String s)
```

Constructor

## Method Detail

### getName

```
public java.lang.String getName()
```

Target의 이름을 되돌린다.

Overrides:

getName in class Target

---

## getID

protected java.lang.String getID()

Target의 이름을 되돌린다.

Overrides:

getID in class Target

---

## readSubarea

public Target.Memory readSubarea(java.awt.Rectangle r,  
int band)

Target으로 부터 정해진 영역을 읽는다.

Overrides:

readSubarea in class Target

---

## embody

public void embody()

영상을 저장할 메모리 영역을 확보한다.

---

## initialize

public void initialize(double value)

영상의 화소값을 초기화 한다.

---

## dispose

public void dispose()

Target을 삭제한다.

Overrides:

dispose in class Target

---

## equals

public boolean equals(java.lang.Object tgt)

입력 파일과의 동일성 여부를 검사한다.

Overrides:

equals in class java.lang.Object

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class TasseledCapTransformOp

java.lang.Object  
|  
+--TasseledCapTransformOp

public class TasseledCapTransformOp  
extends java.lang.Object  
implements [Operator](#)

Tasseled Cap 변환을 수행한다.

### Field Summary

static int	<a href="#">NONE</a>
------------	----------------------

### Constructor Summary

<a href="#">TasseledCapTransformOp()</a> Constructor
---

### Method Summary

protected void	<a href="#">changePixelValue</a> (double[][] a, int aOffset, float[][] b, int bOffset, int length)
void	<a href="#">operate</a> () Tasseled Cap 변환을 실행한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<a href="#">setMethod</a> (int t) 변환 계수 설정을 위한 방법을 설정한다.
void	<a href="#">setMethod</a> (java.lang.String satellite) 변환 계수 설정을 위한 방법을 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

```
public static final int NONE
```

## Constructor Detail

TasseledCapTransformOp

```
public TasseledCapTransformOp()
```

Constructor

## Method Detail

setMethod

```
public void setMethod(java.lang.String satellite)
```

변환 계수 설정을 위한 방법을 설정한다.

---

setMethod

```
public void setMethod(int t)
```

변환 계수 설정을 위한 방법을 설정한다.

---

setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

Tasseled Cap 변환을 실행한다.

---

## changePixelValue

```
protected void changePixelValue(double[][] a,  
                                int aOffset,  
                                float[][] b,  
                                int bOffset,  
                                int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---



## Class TerrainPropertyOp

```
java.lang.Object
|
+--TerrainPropertyOp
```

```
public class TerrainPropertyOp
extends java.lang.Object
implements Operator
```

DEM으로 부터 지형 요소를 계산한다.

Constructor Summary	
<a href="#">TerrainPropertyOp()</a>	Constructor

Method Summary	
void	<a href="#">operate()</a> 지형 요소를 계산한다.
void	<a href="#">setBand(int b)</a> 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle r)</a> 처리 영역을 설정한다.
void	<a href="#">setOptions(boolean[] opt)</a> 처리 옵션을 설정한다.
void	<a href="#">setPixelUnit(float u)</a> 격자의 지상에서의 단위길이를 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setSunPositions(double[] p)</a> 태양의 위치 변수를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### TerrainPropertyOp

```
public TerrainPropertyOp()
```

Constructor

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBand

```
public void setBand(int b)
```

대상 분광대를 설정한다.

---

### setPixelUnit

```
public void setPixelUnit(float u)
```

격자의 지상에서의 단위길이를 설정한다.

---

### setSunPositions

```
public void setSunPositions(double[] p)
```

태양의 위치 변수를 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

### setOptions

```
public void setOptions(boolean[] opt)
```

처리 옵션을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

지형 요소를 계산한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

## Class ThinningOp

```
java.lang.Object  
|  
+--ThinningOp
```

```
public class ThinningOp  
extends java.lang.Object  
implements Operator
```

이진 영상의 세선화를 실행한다. 대상 영상은 메모리형이어야 한다.

### Constructor Summary

<a href="#">ThinningOp()</a> Constructor
---

### Method Summary

void	<a href="#">operate()</a> 세선화를 실행한다.
void	<a href="#">setBackground(int b)</a> 배경 화소값을 설정한다.
void	<a href="#">setBand(int band)</a> 대상 분광대를 설정한다.
void	<a href="#">setBounds(java.awt.Rectangle b)</a> 처리 영역을 설정한다.
void	<a href="#">setProgressStream(ProgressStream prog)</a> 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget(Target stgt, Target dtgt)</a> 입출력 영상을 설정한다.
void	<a href="#">setThreshold(int t)</a> 입력 영상에서 임계값을 설정한다.
void	<a href="#">stop()</a> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

### Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

### Constructor Detail

[ThinningOp](#)

public ThinningOp()

Constructor

## Method Detail

### setBand

public void setBand(int band)

대상 분광대를 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle b)

처리 영역을 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

public void setTarget([Target](#) stgt,  
[Target](#) dtgt)

입출력 영상을 설정한다.

---

### setBackground

public void setBackground(int b)

배경 화소값을 설정한다.

---

### setThreshold

public void setThreshold(int t)

입력 영상에서 임계값을 설정한다.

---

### operate

public void operate()  
throws [java.io.IOException](#),  
[java.lang.InterruptedException](#)

세션화를 실행한다.

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class TM6ToTemperatureOp

```
java.lang.Object
|
+--TM6ToTemperatureOp
```

```
public class TM6ToTemperatureOp
extends java.lang.Object
implements Operator
```

LANDSAT TM band 6 영상으로 부터 지표면 온도를 추출한다.

### Constructor Summary

<a href="#">TM6ToTemperatureOp()</a> Constructor
---

### Method Summary

protected void	<a href="#">convert</a> (double[] a, int aoffset, float[] b, int boffset, int length)
protected double	<a href="#">convertRadiance</a> (double a)
void	<a href="#">operate</a> () 온도 변환을 실행한다.
void	<a href="#">setBand</a> (int b) 대상 분광대를 설정한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<a href="#">setMethod</a> (int i) 온도 추정 방법을 설정한다.
void	<a href="#">setMethod</a> (java.lang.String method) 온도 추정 방법을 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setSatellite</a> (java.lang.String satellite) 위성의 종류를 설정한다.
void	<a href="#">setTarget</a> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### TM6ToTemperatureOp

```
public TM6ToTemperatureOp()
```

Constructor

## Method Detail

### setSatellite

```
public void setSatellite(java.lang.String satellite)
```

위성의 종류를 설정한다.

---

### setMethod

```
public void setMethod(java.lang.String method)
```

온도 추정 방법을 설정한다.

---

### setMethod

```
public void setMethod(int i)
```

온도 추정 방법을 설정한다.

---

### setBand

```
public void setBand(int b)
```

대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator



---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

온도 변환을 실행한다.

---

## convertRadiance

```
protected double convertRadiance(double a)
```

---

## convert

```
protected void convert(double[] a,  
                       int aOffset,  
                       float[] b,  
                       int bOffset,  
                       int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

### Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class TopographicDeEffectOp

```
java.lang.Object
|
+--TopographicDeEffectOp
```

```
public class TopographicDeEffectOp
    extends java.lang.Object
    implements Operator
```

지형에 의한 영향을 제거한다. 현재는 입출력 모두 파일에 대하여만 가능하다. DEM파일은 2바이트 정수형이어야 한다.

Constructor Summary	
<code>TopographicDeEffectOp()</code>	Constructor

Method Summary	
void	<code>operate()</code> 반사면 모델을 이용하여 지형 효과를 제거한다.
void	<code>setBands(int[] bands)</code> 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle r)</code> 처리 영역을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 반사면 모델을 설정한다.
void	<code>setParameter(double[] p)</code> 태양 입력 변수등을 설정한다.
void	<code>setPixelUnit(float u)</code> 격자의 지상에서의 단위길이를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setSunPositions(double[] p)</code> 태양의 위치 변수를 구하기 위한 입력값을 설정한다.
void	<code>setTarget(Target stgt, Target s2tgt, Target dtgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
<code>clone</code> , <code>equals</code> , <code>finalize</code> , <code>getClass</code> , <code>hashCode</code> , <code>notify</code> , <code>notifyAll</code> , <code>toString</code> , <code>wait</code> , <code>wait</code> , <code>wait</code>

## Constructor Detail

### TopographicDeEffectOp

```
public TopographicDeEffectOp()
```

Constructor

## Method Detail

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

### setPixelUnit

```
public void setPixelUnit(float u)
```

격자의 지상에서의 단위길이를 설정한다.

---

### setSunPositions

```
public void setSunPositions(double[] p)
```

태양의 위치 변수를 구하기 위한 입력값을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상대바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target s2tgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## setParameter

```
public void setParameter(double[] p)
```

태양 입력 변수등을 설정한다.

---

## setMethod

```
public void setMethod(java.lang.String method)
```

반사면 모델을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

반사면 모델을 이용하여 지형 효과를 제거한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

## Class UnaryArithmeticFunctionOp

java.lang.Object  
|--UnaryArithmeticFunctionOp

public class UnaryArithmeticFunctionOp  
extends java.lang.Object  
implements [Operator](#)

영상의 단일 분광대에 대하여 산술 연산을 한다. 출력 화소값은 4바이트 실수형으로 정의된다.

Field Summary	
static int	<a href="#">ADD</a>
static int	<a href="#">DIVIDE</a>
static int	<a href="#">MULTIPLY</a>
static int	<a href="#">NONE</a>
static int	<a href="#">POW</a>
static int	<a href="#">SUBTRACT</a>

Constructor Summary	
<a href="#">UnaryArithmeticFunctionOp()</a>	
Constructor	

## Method Summary

protected void	<code>changePixelValue(double[] a, int aOffset, float[] b, int bOffset, int length)</code>
void	<code>operate()</code> 입력영상의 분광대에 대한 산술연산을 실행한다.
void	<code>setBands(int[] bands)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle b)</code> 처리 영역을 설정한다.
void	<code>setConstant(float c)</code> 적용 상수를 설정한다.
void	<code>setMethod(java.lang.String method)</code> 연산에 대한 방법을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

public static final int NONE

---

ADD

public static final int ADD

---

SUBSTRACT

public static final int SUBSTRACT

---

MULTIPLY

public static final int MULTIPLY

---

DIVIDE

public static final int DIVIDE

---

## POW

public static final int POW

## Constructor Detail

### UnaryArithmeticFunctionOp

public UnaryArithmeticFunctionOp()

Constructor

## Method Detail

### setMethod

public void setMethod(java.lang.String method)

연산에 대한 방법을 설정한다.

---

### setBands

public void setBands(int[] bands)

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle b)

처리 영역을 설정한다.

---

### setConstant

public void setConstant(float c)

적용 상수를 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 분광대에 대한 산술연산을 실행한다.

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                 int aOffset,  
                                 float[] b,  
                                 int bOffset,  
                                 int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---



## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class UnaryBitwiseFunctionOp

java.lang.Object

└--UnaryBitwiseFunctionOp

---

```
public class UnaryBitwiseFunctionOp
  extends java.lang.Object
  implements Operator
```

영상의 단일 분광대에 대하여 비트이동 연산을 한다. 출력 화소값은 4바이트 실수형으로 정의된다.

---

Field Summary	
static int	<a href="#">LBS</a>
static int	<a href="#">NONE</a>
static int	<a href="#">NOT</a>
static int	<a href="#">RBS</a>
static int	<a href="#">RBS0</a>

Constructor Summary	
<a href="#">UnaryBitwiseFunctionOp()</a>	
Constructor	

Method Summary	
protected void	<code>changePixelValue(byte[] a, int aOffset, byte[] b, int bOffset, int length)</code>
void	<code>operate()</code> 입력영상의 분광대에 대한 비트 이동 연산을 실행한다.
void	<code>setBands(int[] bands)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle b)</code> 처리 영역을 설정한다.
void	<code>setConstant(int c)</code> 적용 상수를 설정한다.
void	<code>setMethod(java.lang.String method)</code> 연산에 대한 방법을 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

public static final int NONE

---

LBS

public static final int LBS

---

RBS

public static final int RBS

---

RBS0

public static final int RBS0

---

NOT

public static final int NOT

## Constructor Detail

### UnaryBitwiseFunctionOp

public UnaryBitwiseFunctionOp()

Constructor

## Method Detail

### setMethod

public void setMethod(java.lang.String method)

연산에 대한 방법을 설정한다.

---

### setBands

public void setBands(int[] bands)

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

public void setBounds(java.awt.Rectangle b)

처리 영역을 설정한다.

---

### setConstant

public void setConstant(int c)

적용 상수를 설정한다.

---

### setProgressStream

public void setProgressStream(ProgressStream prog)

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

### setTarget

public void setTarget(Target stgt,  
Target dtgt)

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 분광대에 대한 비트 이동 연산을 실행한다.

---

## changePixelValue

```
protected void changePixelValue(byte[] a,  
    int aOffset,  
    byte[] b,  
    int bOffset,  
    int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class UnaryLogicalFunctionOp

java.lang.Object  
|  
+--UnaryLogicalFunctionOp

public class UnaryLogicalFunctionOp  
extends java.lang.Object  
implements Operator

영상의 단일 분광대에 대하여 논리 비교 연산을 한다. 출력 화소값은 4바이트 실수형으로 정의된다.

Field Summary	
static int	<u>EQUAL</u>
static int	<u>GREATER</u>
static int	<u>GREATER THAN</u>
static int	<u>LESS</u>
static int	<u>LESS THAN</u>
static int	<u>NONE</u>
static int	<u>NOT EQUAL</u>

Constructor Summary	
<u>UnaryLogicalFunctionOp()</u>	Constructor

## Method Summary

protected void	<code>comparePixelValue(byte[] a, int aOffset, byte[] b, int bOffset, int length)</code>
protected void	<code>comparePixelValue(float[] a, int aOffset, byte[] b, int bOffset, int length)</code>
protected void	<code>comparePixelValue(int[] a, int aOffset, byte[] b, int bOffset, int length)</code>
protected void	<code>comparePixelValue(short[] a, int aOffset, byte[] b, int bOffset, int length)</code>
void	<code>operate()</code> 입력영상의 분광대에 대한 논리 비교 연산을 실행한다.
void	<code>setBands(int[] bands)</code> 입력 영상의 대상 분광대를 설정한다.
void	<code>setBounds(java.awt.Rectangle b)</code> 처리 영역을 설정한다.
void	<code>setConstant(float t)</code> 비교를 위한 값을 설정한다.
void	<code>setMethod(java.lang.String method)</code> 논리 비교 연산자를 설정한다.
void	<code>setProgressStream(ProgressStream prog)</code> 프로그램 진행 상태바를 설정한다.
void	<code>setTarget(Target stgt, Target dtgt)</code> 입출력 영상을 설정한다.
void	<code>stop()</code> 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### NONE

public static final int NONE

---

### EQUAL

public static final int EQUAL

---

### NOT\_EQUAL

public static final int NOT\_EQUAL

---

### LESS

```
public static final int LESS
```

---

## LESS\_THAN

```
public static final int LESS_THAN
```

---

## GREATER

```
public static final int GREATER
```

---

## GREATER\_THAN

```
public static final int GREATER_THAN
```

## Constructor Detail

### UnaryLogicalFunctionOp

```
public UnaryLogicalFunctionOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

논리 비교 연산자를 설정한다.

---

### setConstant

```
public void setConstant(float t)
```

비교를 위한 값을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 분광대에 대한 논리 비교 연산을 실행한다.

---

## comparePixelValue

```
protected void comparePixelValue(byte[] a,  
                                  int aOffset,  
                                  byte[] b,  
                                  int bOffset,  
                                  int length)
```

---

## comparePixelValue

```
protected void comparePixelValue(short[] a,  
                                  int aOffset,  
                                  byte[] b,  
                                  int bOffset,  
                                  int length)
```

---

## comparePixelValue

```
protected void comparePixelValue(int[] a,  
                                  int aOffset,  
                                  byte[] b,  
                                  int bOffset,  
                                  int length)
```

---

## comparePixelValue

```
protected void comparePixelValue(float[] a,  
                                  int aOffset,  
                                  byte[] b,
```



```
int bOffset,  
int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class UnaryMathFunctionOp

```
java.lang.Object
|
+--UnaryMathFunctionOp
```

```
public class UnaryMathFunctionOp
    extends java.lang.Object
    implements Operator
```

분광대에 대하여 수학적 연산을 한다. 출력 화소값은 4바이트 실수형이다.

Constructor Summary	
<a href="#">UnaryMathFunctionOp()</a>	Constructor

Method Summary	
protected void	<a href="#">changePixelValue</a> (double[] a, int aoffset, float[] b, int boffset, int length)
protected double	<a href="#">degreeToRadian</a> (double a)
void	<a href="#">operate</a> () 입력영상의 분광대에 대한 수학 연산을 실행한다.
protected double	<a href="#">radianToDegree</a> (double a)
void	<a href="#">setBands</a> (int[] bands) 입력 영상의 대상 분광대를 설정한다.
void	<a href="#">setBounds</a> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<a href="#">setMethod</a> (java.lang.String method) 수학 연산자를 설정한다.
void	<a href="#">setProgressStream</a> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<a href="#">setTarget</a> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<a href="#">stop</a> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### UnaryMathFunctionOp

```
public UnaryMathFunctionOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

수학 연산자를 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

### setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

[setProgressStream](#) in interface [Operator](#)

---

### setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

### operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 분광대에 대한 수학 연산을 실행한다.

---

## degreeToRadian

protected double degreeToRadian(double a)

---

## radianToDegree

protected double radianToDegree(double a)

---

## changePixelValue

```
protected void changePixelValue(double[] a,  
                                int aOffset,  
                                float[] b,  
                                int bOffset,  
                                int length)
```

---

## stop

public void stop()

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

### [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class UnaryMorphologicalFunctionOp

```
java.lang.Object
|
|--UnaryMorphologicalFunctionOp
```

```
public class UnaryMorphologicalFunctionOp
extends java.lang.Object
implements Operator
```

단일 분광대 영상(2진 영상)에 대한 Morphology연산을 실행한다.

Field Summary	
static int	<a href="#">BOUNDARY</a>
static int	<a href="#">CLOSE</a>
static int	<a href="#">CONVEX</a>
static int	<a href="#">DILATE</a>
static int	<a href="#">ERODE</a>
static int	<a href="#">HIT_MISS</a>
static int	<a href="#">NONE</a>
static int	<a href="#">OPEN</a>
static int	<a href="#">PRUNE</a>
static int	<a href="#">SKELETON</a>
static int	<a href="#">THICK</a>
static int	<a href="#">THIN</a>

Constructor Summary	
<a href="#">UnaryMorphologicalFunctionOp()</a>	Constructor

## Method Summary

void	<u>operate</u> () 입력영상의 분광대에 대한 Morphology 연산을 실행한다.
void	<u>setBand</u> (int band) 입력 영상의 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<u>setElementKernel</u> (int[][] kernel) 연산을 위한 Kernel을 입력한다.
void	<u>setMethod</u> (java.lang.String method) Morphology 연산자를 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### NONE

public static final int NONE

---

### DILATE

public static final int DILATE

---

### ERODE

public static final int ERODE

---

### OPEN

public static final int OPEN

---

### CLOSE

public static final int CLOSE

---

## HIT\_MISS

```
public static final int HIT_MISS
```

---

## BOUNDARY

```
public static final int BOUNDARY
```

---

## CONVEX

```
public static final int CONVEX
```

---

## SKELETON

```
public static final int SKELETON
```

---

## THIN

```
public static final int THIN
```

---

## PRUNE

```
public static final int PRUNE
```

---

## THICK

```
public static final int THICK
```

## Constructor Detail

### UnaryMorphologicalFunctionOp

```
public UnaryMorphologicalFunctionOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

Morphology 연산자를 설정한다.

---

## setBand

```
public void setBand(int band)
```

입력 영상의 대상 분광대를 설정한다.

---

## setElementKernel

```
public void setElementKernel(int[][] kernel)
```

연산을 위한 Kernel을 입력한다.

---

## setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

입력영상의 분광대에 대한 Morphology 연산을 실행한다.

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---



## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

[SUMMARY](#): [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

[DETAIL](#): [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class UnaryTextureFunctionOp

java.lang.Object

|--UnaryTextureFunctionOp

---

public class UnaryTextureFunctionOp

extends java.lang.Object

implements [Operator](#)

단일 분광대에 대한 Texture특성을 분석한다.

---

Field Summary	
static int	<a href="#">CORRELATION</a>
static int	<a href="#">D_0</a>
static int	<a href="#">D_135</a>
static int	<a href="#">D_180</a>
static int	<a href="#">D_225</a>
static int	<a href="#">D_270</a>
static int	<a href="#">D_315</a>
static int	<a href="#">D_45</a>
static int	<a href="#">D_90</a>
static int	<a href="#">ENERGY</a>
static int	<a href="#">ENTROPHY</a>
static int	<a href="#">HOMOGENITY</a>
static int	<a href="#">MOMENT</a>
static int	<a href="#">NONE</a>

## Constructor Summary

UnaryTextureFunctionOp()  
Constructor

## Method Summary

protected void	<u>calculateTexture</u> (int line, byte[] a, int aoffset, float[] b, int boffset, int length, int smps)
protected float	<u>getCorrelation</u> ()
protected float	<u>getEnergy</u> ()
protected float	<u>getEntropy</u> ()
protected float	<u>getHomogeneity</u> ()
protected float	<u>getMoment</u> ()
protected void	<u>makeD0_GLCM</u> ()
protected void	<u>makeD180_GLCM</u> ()
protected void	<u>makeD270_GLCM</u> ()
protected void	<u>makeD90_GLCM</u> ()
void	<u>operate</u> () Texture 추출을 실행한다.
void	<u>setBands</u> (int[] bands) 입력 영상의 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle b) 처리 영역을 설정한다.
void	<u>setDirection</u> (java.lang.String type) 조직 추출 방향을 설정한다.
void	<u>setDistance</u> (int d) GLCM연산을 위한 거리를 설정한다.
void	<u>setGreyLevel</u> (int d) GLCM연산을 위한 레벨을 설정한다.
void	<u>setMethod</u> (java.lang.String method) Texture추출 요소를 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
void	<u>setWindowSize</u> (int d) window의 크기를 설정한다.
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

NONE

public static final int NONE

---

D\_0

public static final int D\_0

---

D\_90

public static final int D\_90

---

D\_180

public static final int D\_180

---

D\_270

public static final int D\_270

---

D\_45

public static final int D\_45

---

D\_135

public static final int D\_135

---

D\_225

public static final int D\_225

---

D\_315

```
public static final int D_315
```

---

## ENERGY

```
public static final int ENERGY
```

---

## ENTROPHY

```
public static final int ENTROPHY
```

---

## MOMENT

```
public static final int MOMENT
```

---

## HOMOGENITY

```
public static final int HOMOGENITY
```

---

## CORRELATION

```
public static final int CORRELATION
```

## Constructor Detail

### UnaryTextureFunctionOp

```
public UnaryTextureFunctionOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

Texture추출 요소를 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

입력 영상의 대상 분광대를 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle b)
```

처리 영역을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                     Target dtgt)
```

입출력 영상을 설정한다.

---

## setDirection

```
public void setDirection(java.lang.String type)
```

조직 추출 방향을 설정한다.

---

## setDistance

```
public void setDistance(int d)
```

GLCM연산을 위한 거리를 설정한다.

---

## setGreyLevel

```
public void setGreyLevel(int d)
```

GLCM연산을 위한 레벨을 설정한다.

---

## setWindowSize

```
public void setWindowSize(int d)
```

window의 크기를 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

Texture 추출을 실행한다.

---

## **makeD0\_GLCM**

protected void makeD0\_GLCM()

---

## **makeD90\_GLCM**

protected void makeD90\_GLCM()

---

## **makeD180\_GLCM**

protected void makeD180\_GLCM()

---

## **makeD270\_GLCM**

protected void makeD270\_GLCM()

---

## **getEnergy**

protected float getEnergy()

---

## **getEntropy**

protected float getEntropy()

---

## **getMoment**

protected float getMoment()

---

## **getHomogeneity**

protected float getHomogeneity()

---

## **getCorrelation**

protected float getCorrelation()

---

## **calculateTexture**

protected void calculateTexture(int line,

```
byte[] a,  
int aOffset,  
float[] b,  
int bOffset,  
int length,  
int smps)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

[stop](#) in interface [Operator](#)

---

## [Class Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

# Class VectorLayer

```
java.lang.Object
|
+--VectorLayer
```

---

```
public class VectorLayer
extends java.lang.Object
```

벡터 layer를 관리한다.

---

### Field Summary

static int	<a href="#">FEATURE_ID</a>
static int	<a href="#">LAYER_ORDER</a>
static int	<a href="#">MAX_SIZE</a>

### Constructor Summary

<a href="#">VectorLayer()</a> Constructor
--



## Method Summary

int	<u>getCapacity()</u> 저장된 벡터의 수를 되돌린다.
java.awt.Color	<u>getColor(Feature obj)</u> 객체의 색을 되돌린다.
java.awt.Color	<u>getColor(int index)</u> 인덱스에 해당하는 색을 되돌린다.
Palette	<u>getPalette()</u> 색상표를 되돌린다.
java.awt.Polygon	<u>getPolygon(Feature obj, float scale)</u> 축척에 대한 좌표를 갖는 폴리곤을 되돌린다.
java.awt.Polygon	<u>getPolygon(Feature obj, int xorigin, int yorigin)</u> 영상 좌표를 갖는 폴리곤을 되돌린다.
<u>GeoVector</u>	<u>getVector(int index)</u> 인덱스에 벡터를 되돌린다.
boolean	<u>isEmpty(int index)</u> 인덱스의 벡터가 비어있는지를 검사한다.
boolean	<u>isLayerOrderColorMode()</u> layer 순의 색지정 모드인지를 검사한다.
boolean	<u>isPolygonFill()</u> 폴리곤 채움 모드인지를 검사한다.
boolean	<u>isVisible(int index)</u> 인덱스의 레이어가 표시됨 모드인지를 검사한다.
void	<u>removeVector(int index)</u> 인덱스에 벡터를 삭제한다.
void	<u>setColorMode(int mode)</u> 색지정 모드를 설정한다.
void	<u>setCoordConverter(GeoCoordConverter op)</u> 좌표 변환계를 설정한다.
void	<u>setGreyColorModel()</u> 칼라 모델을 설정한다.
void	<u>setPalette(Palette cm)</u> 색상표를 설정한다.
void	<u>setPolygonFill(boolean b)</u> 폴리곤 채움 모드를 설정한다.
void	<u>setVector(int index, GeoVector v)</u> 인덱스에 벡터를 설정한다.
void	<u>setVisible(int index, boolean b)</u> 인덱스의 레이어를 화면에 나타내는 모드를 설정한다.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### MAX\_SIZE

```
public static final int MAX_SIZE
```

---

## LAYER\_ORDER

```
public static final int LAYER_ORDER
```

---

## FEATURE\_ID

```
public static final int FEATURE_ID
```

---

## Constructor Detail

### VectorLayer

```
public VectorLayer()
```

Constructor

## Method Detail

### setCoordConverter

```
public void setCoordConverter(GeoCoordConverter op)
```

좌표 변환계를 설정한다.

---

### setGreyColorModel

```
public void setGreyColorModel()
```

칼라 모델을 설정한다.

---

### setPalette

```
public void setPalette(Palette cm)
```

색상표를 설정한다.

---

### getPalette

```
public Palette getPalette()
```

색상표를 되돌린다.

---

### setColorMode

```
public void setColorMode(int mode)
```

색지정 모드를 설정한다.

---

## **isLayerOrderColorMode**

```
public boolean isLayerOrderColorMode()
```

layer 순의 색지정 모드인지를 검사한다.

---

## **isPolygonFill**

```
public boolean isPolygonFill()
```

폴리곤 채움 모드인지를 검사한다.

---

## **setPolygonFill**

```
public void setPolygonFill(boolean b)
```

폴리곤 채움 모드를 설정한다.

---

## **getColor**

```
public java.awt.Color getColor(int index)
```

인덱스에 해당하는 색을 되돌린다.

---

## **getColor**

```
public java.awt.Color getColor(Feature obj)
```

객체의 색을 되돌린다.

---

## **setVisible**

```
public void setVisible(int index,  
                       boolean b)
```

인덱스의 레이어를 화면에 나타내는 모드를 설정한다.

---

## **isVisible**

```
public boolean isVisible(int index)
```

인덱스의 레이어가 표시됨 모드인지를 검사한다.

---

## isEmpty

```
public boolean isEmpty(int index)
```

인덱스의 벡터가 비어있는지를 검사한다.

---

## setVector

```
public void setVector(int index,  
                     GeoVector v)
```

인덱스에 벡터를 설정한다.

---

## removeVector

```
public void removeVector(int index)
```

인덱스에 벡터를 삭제한다.

---

## getVector

```
public GeoVector getVector(int index)
```

인덱스에 벡터를 되돌린다.

---

## getPolygon

```
public java.awt.Polygon getPolygon(Feature obj,  
                                   int xorigin,  
                                   int yorigin)
```

영상 좌표를 갖는 폴리곤을 되돌린다.

---

## getPolygon

```
public java.awt.Polygon getPolygon(Feature obj,  
                                   float scale)
```

축척에 대한 좌표를 갖는 폴리곤을 되돌린다.

---

## getCapacity

public int getCapacity()

저장된 벡터의 수를 되돌린다.

---

---

### Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

---

# Class Tree Deprecated Index Help

[PREV CLASS](#) [NEXT CLASS](#)  
SUMMARY: [INNER](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#)  
DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

---

## Class VegetationIndexOp

java.lang.Object  
|  
+--VegetationIndexOp

---

public class VegetationIndexOp  
extends java.lang.Object  
implements Operator

식생지수를 계산한다.

---

Field Summary	
static int	<u>DVI</u>
static int	<u>II</u>
static int	<u>IPVI</u>
static int	<u>MSAVI</u>
static int	<u>MSAVI2</u>
static int	<u>MSI</u>
static int	<u>NDVI</u>
static int	<u>NONE</u>
static int	<u>PVI</u>
static int	<u>RVI</u>
static int	<u>SAVI</u>
static int	<u>TSAVI</u>
static int	<u>TVI</u>
static int	<u>WDVI</u>

# Constructor Summary

VegetationIndexOp()  
Constructor

# Method Summary

protected void	<u>dviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>iiValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>ipviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>msaviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>msiValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>ndviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
void	<u>operate</u> () 식생 지수를 계산한다.
protected void	<u>pviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>rviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>saviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
void	<u>setBands</u> (int[] bands) 대상 분광대를 설정한다.
void	<u>setBounds</u> (java.awt.Rectangle r) 처리 영역을 설정한다.
void	<u>setMethod</u> (java.lang.String method) 식생지수 계산법을 설정한다.
void	<u>setParameter</u> (float[] p) 식생 지수 산출과 관련된 Soil Line 변수들을 설정한다.
void	<u>setProgressStream</u> (ProgressStream prog) 프로그램 진행 상태바를 설정한다.
void	<u>setTarget</u> (Target stgt, Target dtgt) 입출력 영상을 설정한다.
protected void	<u>smsaviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
void	<u>stop</u> () 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.
protected void	<u>tsaviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>tviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)
protected void	<u>wdviValue</u> (double[][] a, int aOffset, float[] b, int bOffset, int length)

## Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### NONE

public static final int NONE

---

### NDVI

public static final int NDVI

---

### TVI

public static final int TVI

---

### DVI

public static final int DVI

---

### WDVI

public static final int WDVI

---

### SAVI

public static final int SAVI

---

### TSAVI

public static final int TSAVI

---

### MSAVI

public static final int MSAVI

---

### MSAVI2



```
public static final int MSAVI2
```

---

## PVI

```
public static final int PVI
```

---

## RVI

```
public static final int RVI
```

---

## IPVI

```
public static final int IPVI
```

---

## II

```
public static final int II
```

---

## MSI

```
public static final int MSI
```

## Constructor Detail

### VegetationIndexOp

```
public VegetationIndexOp()
```

Constructor

## Method Detail

### setMethod

```
public void setMethod(java.lang.String method)
```

식생지수 계산법을 설정한다.

---

### setBounds

```
public void setBounds(java.awt.Rectangle r)
```

처리 영역을 설정한다.

---

### setBands

```
public void setBands(int[] bands)
```

대상 분광대를 설정한다.

---

## setParameter

```
public void setParameter(float[] p)
```

식생 지수 산출과 관련된 Soil Line 변수들을 설정한다.

---

## setProgressStream

```
public void setProgressStream(ProgressStream prog)
```

프로그램 진행 상태바를 설정한다.

Specified by:

setProgressStream in interface Operator

---

## setTarget

```
public void setTarget(Target stgt,  
                    Target dtgt)
```

입출력 영상을 설정한다.

---

## operate

```
public void operate()  
    throws java.io.IOException,  
           java.lang.InterruptedException
```

식생 지수를 계산한다.

---

## ndviValue

```
protected void ndviValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## tviValue

```
protected void tviValue(double[][] a,  
                       int aOffset,  
                       float[] b,  
                       int bOffset,  
                       int length)
```

---

## dviValue

```
protected void dviValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## rviValue

```
protected void rviValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## pviValue

```
protected void pviValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## ipviValue

```
protected void ipviValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## msiValue

```
protected void msiValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## iiValue

```
protected void iiValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,  
                        int length)
```

---

## wdviValue

```
protected void wdviValue(double[][] a,  
                        int aOffset,  
                        float[] b,  
                        int bOffset,
```

```
int length)
```

---

## saviValue

```
protected void saviValue(double[][] a,  
                          int aOffset,  
                          float[] b,  
                          int bOffset,  
                          int length)
```

---

## tsaviValue

```
protected void tsaviValue(double[][] a,  
                           int aOffset,  
                           float[] b,  
                           int bOffset,  
                           int length)
```

---

## msaviValue

```
protected void msaviValue(double[][] a,  
                           int aOffset,  
                           float[] b,  
                           int bOffset,  
                           int length)
```

---

## smsaviValue

```
protected void smsaviValue(double[][] a,  
                            int aOffset,  
                            float[] b,  
                            int bOffset,  
                            int length)
```

---

## stop

```
public void stop()
```

사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.

Specified by:

stop in interface Operator

---

## Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS

SUMMARY: INNER | FIELD | CONSTR | METHOD

FRAMES NO FRAMES

DETAIL: FIELD | CONSTR | METHOD

---

여 백

## 부록 5. 기하보정을 위한 GeoPixel 소스 코드

```
/** @author Shin Daehyok */

import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.Vector;
import VisualNumerics.math.MathException;

/**
Class reading Cube File to Cube Memory
*/
public class GCPCorrecter extends Supervisor implements DlgActionListener {
    private GeoGCPCorrectDialog CorrectDlg = null;
    private GCPCorrectOp Op = null;

    private Target srcTgt = null;
    private Target destTgt = null;
    private int[] Bands = null;

    private int Degree = 1;
    private int Resampling = GeoResampler.NEAREST_NEIGHBOR;
    private int Mapping = GCPCorrectOp.IMAGE_TO_MAP;
    private int Background = 0;
    private int Samples = 0;
    private int Lines = 0;
    private GeoRectangle GeoBounds = null;

    private Matrix GCP = null;

    public GCPCorrecter() {
        super();
    }

    //command execute
    public void execute() {
        CorrectDlg = new GeoGCPCorrectDialog(Parent, this.Description, false);//not modal
        CorrectDlg.setPrecursor(Parent);
        CorrectDlg.addDlgActionListener(this);
        CorrectDlg.setVisible(true);
    }

    //handler for dialog action event
    public void dlgActionPerformed(DlgActionEvent e) {
        Object obj = e.getSource();
        //Importer Dialog
        if (obj == CorrectDlg) {
            CorrectDlg.dispose();//setVisible(false) in HDDlg
            String choice = e.getChoice();

            if(choice == "OK") {
                Object[] arg = (Object[])e.getArgument();
                this.srcTgt = (Target)arg[0];
                this.Bands = (int[])arg[1];
                this.destTgt = (Target)arg[2];
                this.GCP = (Matrix)arg[3];

                this.Mapping = ((Integer)arg[4]).intValue();
                this.Degree = ((Integer)arg[5]).intValue();
                this.Resampling = ((Integer)arg[6]).intValue();
                this.Background = ((Integer)arg[7]).intValue();

                int[] temp = (int[])arg[8];
                this.Samples = temp[0];
            }
        }
    }
}
```

```

        this.Lines = temp[1];

        this.GeoBounds = (GeoRectangle)arg[9];

        new Thread(this).start();
    }
}

public void run() {
    GeoProgressDialog ProgDlg
        = new GeoProgressDialog(GeoEnviron.getCurrentMainFrame(), this.Description)
    ProgDlg.addWindowListener(this);
    ProgDlg.setVisible(true);

    if(srcTgt instanceof Target.File) {
        ProgDlg.setMin(0);
        ProgDlg.setMax(srcTgt.HD.Lines*this.Bands.length+Lines*(this.Bands.length+1));
    }else if(srcTgt instanceof Target.Memory) {
        ProgDlg.setMin(0);
        ProgDlg.setMax(Lines);
    }

    Target[] tgtA = null;
    try {
        GcpRegressor reg = new GcpRegressor(this.GCP.e, this.Degree);
        reg.regress();
        double[][] A = reg.getCoef();
        double[] T = reg.getBias();

        Op = new GCPCorrectOp();
        Op.setProgressStream(ProgDlg);
        Op.setMapping(this.Mapping);
        Op.setResampling(this.Resampling);
        Op.setBackground(this.Background);
        Op.setPolyOrder(this.Degree);
        Op.setImageSize(this.Samples, this.Lines);
        Op.setGeoBounds(this.GeoBounds);
        Op.setCoefficient(A, T);

        if (srcTgt instanceof Target.File) {
            CubeFileToMemoryOp loader = new CubeFileToMemoryOp();
            loader.setProgressStream(ProgDlg);

            Vector cubeList = new Vector();
            Target.Memory memTgt
                = new Target.Memory("Mem : "+((Target.File)srcTgt).Pixel.getName());
            Target.File tmpTgt = null;
            int[] b = new int[1];
            for (int i = 0; i < srcTgt.HD.Bands; i++) {
                for (int j = 0; j < this.Bands.length; j++) {
                    if (i == this.Bands[j]) {
                        b[0] = i;
                        loader.setTarget((Target.File)srcTgt, (Target.Memory)memTgt);
                        loader.setBands(b);
                        loader.operate();

                        tmpTgt = new Target.File(GeoEnviron.CurrentPath,
                            "tmp###" + i + "." + System.getProperty("geopixel.cube.ext"));
                        b[0] = 0;
                        Op.setTarget(memTgt, tmpTgt);
                        Op.setBands(b);
                    }
                }
            }
        }
    }
}

```

```

        Op.operate();

        cubeList.addElement(tmpTgt);
        break;
    }
}
loader = null;
memTgt = null;tmpTgt = null;

//merging

MultiCubeMergeOp merger = new MultiCubeMergeOp();
tgtA = new Target[cubeList.size()];
for(int i=0; i<tgtA.length; i++)
    tgtA[i] = (Target)cubeList.elementAt(i);
int[][] bandA = new int[this.Bands.length][1];
int[] order = new int[this.Bands.length];
for(int i=0; i<this.Bands.length; i++) {
    bandA[i][0] = 0;
    order[i] = i;
}
merger.setTarget(tgtA, destTgt);
merger.setBands(bandA, order);
merger.setProgressStream(ProgDlg);
merger.operate();

}else if(srcTgt instanceof Target.Memory) {
    Op.setTarget(srcTgt, destTgt);
    Op.setBands(this.Bands);
    Op.operate();
}
if (destTgt instanceof Target.File) {
    destTgt.HD.write(((Target.File)destTgt).getPathOfHDFFile());
}
GeoEnviron.CubeListDlg.add(destTgt);
}catch(InterruptedException e) {
    GeoEnviron.message(GeoEnviron.getCurrentMainFrame(), "User Interruption");
}catch(IllegalArgumentException e) {
    GeoEnviron.message(GeoEnviron.getCurrentMainFrame(), "Not enough GCP points");
}catch(MathException e) {
    GeoEnviron.message(GeoEnviron.getCurrentMainFrame(), "Regression mathematical error")
}catch(IOException evt) {
    GeoEnviron.message(GeoEnviron.getCurrentMainFrame(), evt);
}finally {
    if(ProgDlg != null) ProgDlg.setVisible(false);
    this.Op = null;
    this.srcTgt = null; this.destTgt = null;
    if(tgtA != null) {
        for (int i = 0; i < tgtA.length; i++)
            tgtA[i].dispose();
    }
}
}

public void windowClosing(WindowEvent e) {
    xyz.print("CubeMemoryToFileSAver = > WindowClosing");
    Op.stop();
}
}

/** @author Chung-Hyun, AHN */
import java.awt.*;

```



```

import java.io.*;
import VisualNumerics.math.DoubleVector;

/**
 * 영상의 GCP보정을 실행한다. 대상 영상은 메모리형이어야 한다.
 */
public class GPCCorrectOp implements Operator {
    private ProgressStream ProgStream = null;
    private boolean StopRequest = false;

    public final static int IMAGE_TO_IMAGE = 1;
    public final static int IMAGE_TO_MAP = 2;
    public final static int MAP_TO_MAP = 3;

    private Target srcTgt = null;
    private Target destTgt = null;
    private int[] Bands = null;

    private int Degree = 1;
    private int Resampling = GeoResampler.NEAREST_NEIGHBOR;
    private int Mapping = IMAGE_TO_MAP;
    private double[][] Coef = null;
    private double[] T = null;
    private int Background = 0;

    private Dimension srcDim = new Dimension();
    private GeoRectangle srcGeoBounds = new GeoRectangle();
    private double srcDs = 0.;
    private double srcDl = 0.;

    private Dimension destDim = new Dimension();
    private GeoRectangle destGeoBounds = new GeoRectangle();
    private GeoCoordConverter destCoordOp = new GeoCoordConverter()

    double[] m = null;

    GeoResampler Resampler = null;

    /**
     * Constructor
     */
    public GPCCorrectOp() {
        Resampler = new GeoResampler();
    }

    /**
     * 프로그램 진행 상태바를 설정한다.
     * @parm prog 프로그래스스트림
     */
    public void setProgressStream(ProgressStream prog) {
        ProgStream = prog;
    }

    /**
     * 대상 영상을 설정한다.
     * @parm srcTgt 입력 영상
     * @parm destTgt 출력 영상
     */
    public void setTarget(Target srcTgt, Target destTgt) {
        this.srcTgt = srcTgt;
        this.destTgt = destTgt;
    }

    /**
     * 처리 대상 분광대를 설정한다.
     * @parm bands 처리 대상 분광대 배열

```

```

*/
public void setBands(int[] bands) {
    this.Bands = new int[bands.length];
    System.arraycopy(bands, 0, this.Bands, 0, bands.length);
}

/**
 * 변환 방법을 설정한다.
 * @param mapping 변환방법 IMAGE_TO_IMAGE = 1, IMAGE_TO_MAP = 2, MAP_TO_MAP = 3;
 */
public void setMapping(int mapping) {
    this.Mapping = mapping;
}

/**
 * 화소 재배열 방법을 설정한다.
 * @param resmp 재배열법
 * GeoResampler.NEAREST_NEIGHBOR, GeoResampler.BILINEAR, GeoResampler.CUBIC, GeoResampler.B_SPLINE
 */
public void setResampling(int resmp) {
    this.Resampling = resmp;
}

/**
 * 변환식의 차수를 설정한다.
 * @param degree 차수
 */
public void setPolyOrder(int degree) {
    this.Degree = degree;
    m = new double[((Degree+1)*(Degree+2))/2];
}

/**
 * 배경 화소값을 설정한다.
 * @param background 배경 화소값
 */
public void setBackground(int background) {
    this.Background = background;
}

/**
 * 영상의 크기를 설정한다.
 * @param smps 영상의 samples
 * @param lns 영상의 lns
 */
public void setImageSize(int smps, int lns) {
    destDim.setSize(smps, lns);
}

/**
 * 처리 영역을 설정한다.
 * @param r 처리 영역
 */
public void setGeoBounds(GeoRectangle r) {
    destGeoBounds.setBounds(r);
}

/**
 * 대상 영상을 설정한다.
 * @param srcTgt 입력 영상
 * @param destFile 출력 파일
 */
public void operate()
    throws IOException, InterruptedException
{
    srcDim.width = srcTgt.HD.Samples;
}

```

```

srcDim.height = srcTgt.HD.Lines;
srcGeoBounds.setBounds(srcTgt.HD.getGeoCoord());
srcDs = (double)srcDim.width/(srcGeoBounds.xmax - srcGeoBounds.ymin);
srcDl = (double)srcDim.height/(srcGeoBounds.ymax - srcGeoBounds.ymin);

destCoordOp.setImageSize(destDim.width, destDim.height);
destCoordOp.setBounds(destGeoBounds);

try{
    if (srcTgt instanceof Target.Memory && destTgt instanceof Target.Memory)
        this.correct((Target.Memory)srcTgt, (Target.Memory)destTgt);
    else if (srcTgt instanceof Target.Memory && destTgt instanceof Target.File) {
        this.correct((Target.Memory)srcTgt, (Target.File)destTgt);
        destTgt.HD.write(((Target.File)destTgt).getPathOfHDFFile());
    }else {
        GeoEnviron.message(GeoEnviron.getCurrentMainFrame(), "Sorry, only memory file supported")
    }
}catch(InterruptedException e) {
    destTgt.dispose();
    destTgt.HD = null;
    throw e;
}catch(IOException e) {
    destTgt.dispose();
    destTgt.HD = null;
    throw e;
}
}

private void getPixelCoord(double[] xy, float[] xp, float[] yp, int index) {
    xp[index] = (float)(srcDs*(xy[0]-srcGeoBounds.xmin)-0.5);
    yp[index] = (float)(-srcDl*(xy[1]-srcGeoBounds.ymax)-0.5);
}

/**
 * 변환식의 계수를 설정한다.
 * @param coef X, Y에 대한 변환식 계수 배열
 * @param t 변환결과 의 수평, 수직이동값
 */
public void setCoefficient(double[][] coef, double[] t) {
    Coef = coef;
    T = t;
}

private void getPosition(double a, double b, double[] xy)
{
    int index = 0;
    double x = a;
    double y = b;
    x -= T[2];
    y -= T[3];
    m[index++] = 1.;
    m[index++] = x;
    m[index++] = y;
    if(Degree > 1) {
        m[index++] = x * x;
        m[index++] = x * y;
        m[index++] = y * y;
    }

    if(Degree > 2) {
        m[index++] = x * x * x;
        m[index++] = x * x * y;
        m[index++] = x * y * y;
        m[index++] = y * y * y;
    }
}

```

```

        xy[0] = DoubleVector.innerProduct(m, Coef[0])+T[0];
        xy[1] = DoubleVector.innerProduct(m, Coef[1])+T[1];
    }

private void setHeader(Target srcTgt, Target destTgt) {
    destTgt.HD = new CubeHeader(srcTgt.HD.PixelType, destDim.width, destDim.height, Bands.length);
    destTgt.HD.Title = "Geometrical Corrected Image from "+srcTgt.getName();
    destTgt.HD.setGeoCoord(destGeoBounds);

    for (int i = 0; i < Bands.length; i++)
        destTgt.HD.validMinMax[i] = false;

    for (int i = 0; i < Bands.length; i++)
        destTgt.HD.setBandName(i, ("Band " + (i + 1)));
}

// 메모리에서 메모리로
private void correct(Target.Memory srcTgt, Target.Memory destTgt)
    throws IOException, InterruptedException
{
    setHeader(srcTgt, destTgt);
    destTgt.embody();

    final int smps = destTgt.HD.Samples;
    final int lns = destTgt.HD.Lines;
    final int bnds = Bands.length;

    boolean[] isSelectedBand = new boolean[srcTgt.HD.Bands];

    for (int i = 0; i < srcTgt.HD.Bands; i++) {
        for (int j = 0; j < bnds; j++) {
            if (i == Bands[j]) isSelectedBand[i] = true;
        }
    }

    int offset = 0;

    Resampler.setScale(1.f);
    double[] a, b, c;

    double x, y;
    float[] xp = new float[smps];
    float[] yp = new float[smps];
    switch(srcTgt.HD.PixelType) {
        case Cube.BYTE :
            {
                byte[][] bundle = new byte[bnds][smps];
                byte[][] oPixel = (byte[][])destTgt.Pixel;
                byte[][] iPixel = (byte[][])srcTgt.Pixel;
                for (int i = 0; i < lns; i++) {
                    if(StopRequest) throw new InterruptedException();
                    getMappedCoord(i, smps, xp, yp);
                    switch (Resampling) {
                        case GeoResampler.NEAREST_NEIGHBOR:
                            Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                            break;
                        case GeoResampler.BILINEAR:
                            Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                            break;
                        case GeoResampler.CUBIC:
                            Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height

```

```

srcDim.width);
        break;
    }
    for (int j = 0; j < bnds; j++) {
        System.arraycopy(bundle[j], 0, oPixel[j], offset, smps);
        offset += smps;
    }
    if(ProgStream != null) ProgStream.progress(1);
}
break;
}
case Cube.SHORT :
case Cube.USHORT:
{
    short[][] bundle = new short[bnds][smps];
    short[][] oPixel = (short[][])destTgt.Pixel;
    short[][] iPixel = (short[][])srcTgt.Pixel;
    for (int i = 0; i < lns; i++) {
        if(StopRequest) throw new InterruptedException();
        getMappedCoord(i, smps, xp, yp);
        switch (Resampling) {
            case GeoResampler.NEAREST_NEIGHBOR:Resampler.NearestNeighbor(iPixel, bundle
isSelectedBand, xp, yp, srcDim.height, srcDim.width);
                break;
            case GeoResampler.BILINEAR:Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp
srcDim.height, srcDim.width);
                break;
            case GeoResampler.CUBIC:Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp
srcDim.height, srcDim.width);
                break;
        }
        for (int j = 0; j < bnds; j++) {
            System.arraycopy(bundle[j], 0, oPixel[j], offset, smps);
            offset += smps;
        }
        if(ProgStream != null) ProgStream.progress(1);
    }
    break;
}
case Cube.INT :
{
    int[][] bundle = new int[bnds][smps];
    int[][] oPixel = (int[][])destTgt.Pixel;
    int[][] iPixel = (int[][])srcTgt.Pixel;
    for (int i = 0; i < lns; i++) {
        if(StopRequest) throw new InterruptedException();
        getMappedCoord(i, smps, xp, yp);
        switch (Resampling) {
            case GeoResampler.NEAREST_NEIGHBOR:
                Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                break;
            case GeoResampler.BILINEAR:
                Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                break;
            case GeoResampler.CUBIC:
                Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                break;
        }
        for (int j = 0; j < bnds; j++) {
            System.arraycopy(bundle[j], 0, oPixel[j], offset, smps);
            offset += smps;
        }
    }
}

```

```

        }
        if(ProgStream != null) ProgStream.progress(1);
    }
    break;
}
case Cube.FLOAT:
{
    float[][] bundle = new float[bnds][smpls];
    float[][] oPixel = (float[][])destTgt.Pixel;
    float[][] iPixel = (float[][])srcTgt.Pixel;
    for (int i = 0; i < lns; i++) {
        if(StopRequest) throw new InterruptedException();
        getMappedCoord(i, smpls, xp, yp);
        switch (Resampling) {
            case GeoResampler.NEAREST_NEIGHBOR:
                Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                break;
            case GeoResampler.BILINEAR:
                Resampler.Bilinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                break;
            case GeoResampler.CUBIC:
                Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                break;
        }
        for (int j = 0; j < bnds; j++) {
            System.arraycopy(bundle[j], 0, oPixel[j], offset, smpls);
            offset += smpls;
        }
        if(ProgStream != null) ProgStream.progress(1);
    }
    break;
}
}
}

```

```

private void correct(Target.Memory srcTgt, Target.File destTgt)
    throws IOException, InterruptedException
{
    setHeader(srcTgt, destTgt);

    final int smpls = destTgt.HD.Samples;
    final int lns = destTgt.HD.Lines;
    final int bnds = Bands.length;

    boolean[] isSelectedBand = new boolean[srcTgt.HD.Bands];

    for (int i = 0; i < srcTgt.HD.Bands; i++) {
        for (int j = 0; j < bnds; j++) {
            if (i == Bands[j]) isSelectedBand[i] = true;
        }
    }

    int offset = 0;

    Resampler.setScale(1.f);
    double[] a, b, c;

    GeoDataOutputStream out = null;
    try {
        out = new GeoDataOutputStream(new BufferedOutputStream(
            new FileOutputStream(destTgt.Pixel), destTgt.HD.getBundleByteSize()));
    }
}

```

```

double x, y;
float[] xp = new float[smps];
float[] yp = new float[smps];
double[] l = new double[2];
double[] m = new double[2];

switch(srcTgt.HD.PixelType) {
    case Cube.BYTE :
    {
        byte[][] bundle = new byte[bnds][smps];
        byte[][] iPixel = (byte[][])srcTgt.Pixel;
        for (int i = 0; i < lns; i++) {
            if(StopRequest) throw new InterruptedException();
            getMappedCoord(i, smps, xp, yp);
            switch (Resampling) {
                case GeoResampler.NEAREST_NEIGHBOR:
                    Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp
srcDim.height, srcDim.width);
                    break;
                case GeoResampler.BILINEAR:
                    Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                    break;
                case GeoResampler.CUBIC:
                    Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                    break;
            }
            for (int j = 0; j < bnds; j++) {
                out.write(bundle[j]);
            }
            if(ProgStream != null) ProgStream.progress(1);
        }
        break;
    }
    case Cube.SHORT:
    case Cube.USHORT:
    {
        short[][] bundle = new short[bnds][smps];
        short[][] iPixel = (short[][])srcTgt.Pixel;
        for (int i = 0; i < lns; i++) {
            if(StopRequest) throw new InterruptedException();
            getMappedCoord(i, smps, xp, yp);
            switch (Resampling) {
                case GeoResampler.NEAREST_NEIGHBOR:
                    Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp
srcDim.height, srcDim.width);
                    break;
                case GeoResampler.BILINEAR:
                    Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                    break;
                case GeoResampler.CUBIC:
                    Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height
srcDim.width);
                    break;
            }
            for (int j = 0; j < bnds; j++) {
                out.write(bundle[j]);
            }
            if(ProgStream != null) ProgStream.progress(1);
        }
        break;
    }
}
}

```

```

    case Cube.INT:
    {
        int[][] bundle = new int[bnds][smpls];
        int[][] iPixel = (int[][])srcTgt.Pixel;
        for (int i = 0; i < lns; i++) {
            if(StopRequest) throw new InterruptedException();
            getMappedCoord(i, smpls, xp, yp);
            switch (Resampling) {
                case GeoResampler.NEAREST_NEIGHBOR:
                    Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp,
srcDim.height, srcDim.width);
                    break;
                case GeoResampler.BILINEAR:
                    Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height,
srcDim.width);
                    break;
                case GeoResampler.CUBIC:
                    Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height,
srcDim.width);
                    break;
            }
            for (int j = 0; j < bnds; j++) {
                out.write(bundle[j]);
            }
            if(ProgStream != null) ProgStream.progress(1);
        }
        break;
    }
    case Cube.FLOAT :
    {
        float[][] bundle = new float[bnds][smpls];
        float[][] iPixel = (float[][])srcTgt.Pixel;
        for (int i = 0; i < lns; i++) {
            if(StopRequest) throw new InterruptedException();
            getMappedCoord(i, smpls, xp, yp);
            switch (Resampling) {
                case GeoResampler.NEAREST_NEIGHBOR:
                    Resampler.NearestNeighbor(iPixel, bundle, isSelectedBand, xp, yp,
srcDim.height, srcDim.width);
                    break;
                case GeoResampler.BILINEAR:
                    Resampler.BiLinear(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height,
srcDim.width);
                    break;
                case GeoResampler.CUBIC:
                    Resampler.Cubic(iPixel, bundle, isSelectedBand, xp, yp, srcDim.height,
srcDim.width);
                    break;
            }
            for (int j = 0; j < bnds; j++) {
                out.write(bundle[j]);
            }
            if(ProgStream != null) ProgStream.progress(1);
        }
        break;
    }
}
}
} catch(IOException e) {
    throw e;
} finally {
    try {
        if(out != null) out.close();
    } catch(IOException e) {
        throw e;
    }
}
}
}

```



```

    }
}

private void getMappedCoord(int ln, int smps, float[] xp, float[] yp) {
    double[] tempo = new double[2];
    if(Mapping == MAP_TO_MAP) {
        double x,y;
        for(int i=0; i<smps; i++) {
            x = destCoordOp.getMapX(i);
            y = destCoordOp.getMapY(ln);
            getPosition(x, y,tempo);
            getPixelCoord(tempo,xp,yp,i);
        }
    }else if(Mapping == IMAGE_TO_MAP) {
        double x,y;
        for(int i=0; i<smps; i++) {
            x = destCoordOp.getMapX(i);
            y = destCoordOp.getMapY(ln);
            getPosition(x, y,tempo);
            xp[i] = (float)tempo[0]; yp[i] = (float)tempo[1];
        }
    }else if(Mapping == IMAGE_TO_IMAGE) {
        for(int i=0; i<smps; i++) {
            getPosition((double)i, (double)ln, tempo);
            xp[i] = (float)tempo[0]; yp[i] = (float)tempo[1];
        }
    }
}

/**
 * 사용자 인터럽트가 요청되었을 때 처리를 종료시킨다.
 */
public void stop() {
    StopRequest = true;
}

/*
    A basic extension of the java.awt.Dialog class
*/
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import GeoEnviron;
import java.awt.Frame;
import java.util.Date;
import GeoDialog;
import GeoMessageDialog;

import symantec.itools.awt.BorderPanel;
import VisualNumerics.math.MathException;

//public class GeoGCPCorrectDialog extends Dialog
public class GeoGCPCorrectDialog extends Dialog implements DigActionListener
{
    private Matrix Mtx = null;

    public GeoGCPCorrectDialog(Frame parent, boolean modal)
    {
        super(parent, modal);

        // This code is automatically generated by Visual Cafe when you add
        // components to the visual environment. It instantiates and initializes
        // the components. To modify the code, only use code syntax that matches

```

```

// what Visual Cafe can generate, or Visual Cafe may be unable to back
// parse your Java file into its visual environment.
//{{INIT_CONTROLS
setLayout(null);
setVisible(false);
setSize(367,433);
setBackground(new Color(12632256));
InputLabel = new java.awt.Label("Input Image :",Label.RIGHT);
InputLabel.setBounds(12,6,78,24);
InputLabel.setFont(new Font("Dialog", Font.BOLD, 12));
add(InputLabel);
InputBrowseButton = new java.awt.Button();
InputBrowseButton.setLabel("...");
InputBrowseButton.setBounds(336,6,24,24);
InputBrowseButton.setFont(new Font("Dialog", Font.BOLD, 12));
add(InputBrowseButton);
OutputFileLabel = new java.awt.Label("Output File :",Label.RIGHT);
OutputFileLabel.setBounds(12,72,78,24);
OutputFileLabel.setFont(new Font("Dialog", Font.BOLD, 12));
add(OutputFileLabel);
OutputBrowseButton = new java.awt.Button();
OutputBrowseButton.setLabel("...");
OutputBrowseButton.setBounds(336,72,24,24);
OutputBrowseButton.setFont(new Font("Dialog", Font.BOLD, 12));
add(OutputBrowseButton);
OutputInfoBorderPanel = new symantec.itools.awt.BorderPanel();
try {
    OutputInfoBorderPanel.setLabel("Output Parameter");
}
catch(java.beans.PropertyVetoException e) { }
try {
    OutputInfoBorderPanel.setAlignStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT)
}
catch(java.beans.PropertyVetoException e) { }
OutputInfoBorderPanel.setLayout(null);
OutputInfoBorderPanel.setBounds(12,282,350,114);
OutputInfoBorderPanel.setFont(new Font("Dialog", Font.BOLD, 12));
add(OutputInfoBorderPanel);
ColumnLabel = new java.awt.Label("Columns :",Label.RIGHT);
ColumnLabel.setBounds(32,3,60,24);
ColumnLabel.setFont(new Font("Dialog", Font.BOLD, 12));
OutputInfoBorderPanel.add(ColumnLabel);
ColumnsTextField = new java.awt.TextField();
ColumnsTextField.setText("1000");
ColumnsTextField.setBounds(98,3,54,18);
OutputInfoBorderPanel.add(ColumnsTextField);
RowLabel = new java.awt.Label("Rows :",Label.RIGHT);
RowLabel.setBounds(200,3,42,24);
RowLabel.setFont(new Font("Dialog", Font.BOLD, 12));
OutputInfoBorderPanel.add(RowLabel);
RowsTextField = new java.awt.TextField();
RowsTextField.setText("1000");
RowsTextField.setBounds(248,3,54,18);
OutputInfoBorderPanel.add(RowsTextField);
XminLabel = new java.awt.Label("Xmin :",Label.RIGHT);
XminLabel.setBounds(2,30,90,18);
XminLabel.setFont(new Font("Dialog", Font.BOLD, 12));
OutputInfoBorderPanel.add(XminLabel);
YminLabel = new java.awt.Label("Ymin :",Label.RIGHT);
YminLabel.setBounds(2,54,90,18);
YminLabel.setFont(new Font("Dialog", Font.BOLD, 12));

```

```

OutputInfoBorderPanel.add(YminLabel);
XminTextField = new java.awt.TextField();
XminTextField.setText("17000");
XminTextField.setBounds(98,30,54,18);
OutputInfoBorderPanel.add(XminTextField);
YminTextField = new java.awt.TextField();
YminTextField.setText("40000");
YminTextField.setBounds(98,57,54,18);
OutputInfoBorderPanel.add(YminTextField);
XmaxLabel = new java.awt.Label("Xmax :",Label.RIGHT);
XmaxLabel.setBounds(158,30,84,18);
XmaxLabel.setFont(new Font("Dialog", Font.BOLD, 12));
OutputInfoBorderPanel.add(XmaxLabel);
YmaxLabel = new java.awt.Label("Ymax :",Label.RIGHT);
YmaxLabel.setBounds(164,54,78,18);
YmaxLabel.setFont(new Font("Dialog", Font.BOLD, 12));
OutputInfoBorderPanel.add(YmaxLabel);
XmaxTextField = new java.awt.TextField();
XmaxTextField.setText("20000");
XmaxTextField.setBounds(248,30,54,18);
OutputInfoBorderPanel.add(XmaxTextField);
YmaxTextField = new java.awt.TextField();
YmaxTextField.setText("43000");
YmaxTextField.setBounds(248,54,54,18);
OutputInfoBorderPanel.add(YmaxTextField);
OkButton = new java.awt.Button();
OkButton.setLabel("Ok");
OkButton.setBounds(126,402,74,24);
OkButton.setFont(new Font("Dialog", Font.BOLD, 12));
add(OkButton);
CancelButton = new java.awt.Button();
CancelButton.setActionCommand("OK");
CancelButton.setLabel("Cancel");
CancelButton.setBounds(204,402,74,24);
CancelButton.setFont(new Font("Dialog", Font.BOLD, 12));
add(CancelButton);
HelpButton = new java.awt.Button();
HelpButton.setLabel("Help");
HelpButton.setBounds(282,402,74,24);
HelpButton.setFont(new Font("Dialog", Font.BOLD, 12));
add(HelpButton);
InputGCPLabel = new java.awt.Label("Input GCP :",Label.RIGHT);
InputGCPLabel.setBounds(12,36,78,24);
InputGCPLabel.setFont(new Font("Dialog", Font.BOLD, 12));
add(InputGCPLabel);
InputGCPButton = new java.awt.Button();
InputGCPButton.setLabel("...");
InputGCPButton.setBounds(336,36,24,24);
InputGCPButton.setFont(new Font("Dialog", Font.BOLD, 12));
add(InputGCPButton);
MemoryCheckbox = new java.awt.Checkbox("Image to Memory");
MemoryCheckbox.setBounds(96,108,126,17);
MemoryCheckbox.setFont(new Font("Dialog", Font.BOLD, 11));
add(MemoryCheckbox);
CorrectionPanel = new symantec.itools.awt.BorderPanel();
try {
    CorrectionPanel.setLabel("Correction Parameter");
}
catch(java.beans.PropertyVetoException e) { }

```

```

try {
    CorrectionPanel.setAlignmentStyle(symantec.itools.awt.BorderPanel.ALIGN_LEFT);
}
catch(java.beans.PropertyVetoException e) { }
CorrectionPanel.setLayout(null);
CorrectionPanel.setBounds(12,126,350,156);
CorrectionPanel.setFont(new Font("Dialog", Font.BOLD, 12));
add(CorrectionPanel);
TargetGroup = new CheckboxGroup();
ImageToMapRadioButton = new java.awt.Checkbox("Image To Map", TargetGroup, true);
ImageToMapRadioButton.setBounds(14,30,102,24);
ImageToMapRadioButton.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(ImageToMapRadioButton);
ImageToImageRadioButton = new java.awt.Checkbox("Image To Image", TargetGroup, false);
ImageToImageRadioButton.setBounds(116,30,114,24);
ImageToImageRadioButton.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(ImageToImageRadioButton);
MapToMapRadioButton = new java.awt.Checkbox("Map To Map", TargetGroup, false);
MapToMapRadioButton.setBounds(230,30,90,24);
MapToMapRadioButton.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(MapToMapRadioButton);
TargetLabel = new java.awt.Label("Corrected Target :");
TargetLabel.setBounds(14,6,108,24);
TargetLabel.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(TargetLabel);
Order = new java.awt.Label("Polynomial Order :");
Order.setBounds(14,60,108,24);
Order.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(Order);
OrderGroup = new CheckboxGroup();
FirstRadioButton = new java.awt.Checkbox("1st", OrderGroup, true);
FirstRadioButton.setBounds(134,60,42,24);
FirstRadioButton.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(FirstRadioButton);
SecondRadioButton = new java.awt.Checkbox("2nd", OrderGroup, false);
SecondRadioButton.setBounds(206,60,42,24);
SecondRadioButton.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(SecondRadioButton);
ThirdRadioButton = new java.awt.Checkbox("3rd", OrderGroup, false);
ThirdRadioButton.setBounds(278,60,42,24);
ThirdRadioButton.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(ThirdRadioButton);
BackValueLabel = new java.awt.Label("Background Value :",Label.RIGHT);
BackValueLabel.setBounds(146,90,114,18);
BackValueLabel.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(BackValueLabel);
BackValueTextField = new java.awt.TextField(3);
BackValueTextField.setText("0");
BackValueTextField.setBounds(266,90,54,18);
BackValueTextField.setFont(new Font("Dialog", Font.BOLD, 12));
CorrectionPanel.add(BackValueTextField);
OperatorChoice = new java.awt.Choice();
OperatorChoice.addItem("Nearest Neighbor");
OperatorChoice.addItem("Bi-Linear");
OperatorChoice.addItem("Cubic");
try {
    OperatorChoice.select(0);
}
catch (IllegalArgumentException e) { }
CorrectionPanel.add(OperatorChoice);

```

```

OperatorChoice.setBounds(14,90,126,18);
OperatorChoice.setFont(new Font("Dialog", Font.BOLD, 12));
setTitle("GeoGCPCorrectDialog");
setResizable(false);
//})

InputTextField = new GeoCubeIOTextField();
InputTextField.setEditable(false);
InputTextField.setBounds(96,6,234,24);
// InputTextField.setBounds(insets().left + 90,insets().top + 6,240,24);
InputTextField.setFont(new Font("Dialog", Font.PLAIN, 12));
add(InputTextField);
InputGCPTextField = new GeoFileIOTextField();
InputGCPTextField.setEditable(false);
InputGCPTextField.setBounds(96,36,234,24);
add(InputGCPTextField);
OutputTextField = new GeoCubeIOTextField();
OutputTextField.setEditable(false);
// OutputTextField.setBounds(insets().left + 90,insets().top + 36,240,24);
OutputTextField.setBounds(96,72,234,24);
OutputTextField.setFont(new Font("Dialog", Font.PLAIN, 12));
add(OutputTextField);

//{(REGISTER_LISTENERS
SymWindow aSymWindow = new SymWindow();
this.addWindowListener(aSymWindow);
SymAction ISymAction = new SymAction();
InputBrowseButton.addActionListener(ISymAction);
InputGCPButton.addActionListener(ISymAction);
OutputBrowseButton.addActionListener(ISymAction);
OkButton.addActionListener(ISymAction);
CancelButton.addActionListener(ISymAction);
HelpButton.addActionListener(ISymAction);
SymItem ISymItem = new SymItem();
MemoryCheckbox.addItemListener(ISymItem);
ImageToMapRadioButton.addItemListener(ISymItem);
ImageToImageRadioButton.addItemListener(ISymItem);
MapToMapRadioButton.addItemListener(ISymItem);
//})
}

public GeoGCPCorrectDialog(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

public void addNotify()
{
    // Record the size of the window prior to calling parents addNotify.
    Dimension d = getSize();

    super.addNotify();

    if (fComponentsAdjusted)
        return;

    // Adjust components according to the insets
    setSize(insets().left + insets().right + d.width, insets().top + insets().bottom + d.height)
    Component components[] = getComponents();
    for (int i = 0; i < components.length; i++)
    {

```

```

        Point p = components[i].getLocation();
        p.translate(insets().left, insets().top);
        components[i].setLocation(p);
    }
    fComponentsAdjusted = true;
}

// Used for addNotify check.
boolean fComponentsAdjusted = false;

//{{DECLARE_CONTROLS
java.awt.Label InputImageLabel;
java.awt.Button InputBrowseButton;
java.awt.Label OutputFileLabel;
java.awt.Button OutputBrowseButton;
symantec.itools.awt.BorderPanel OutputInfoBorderPanel;
java.awt.Label ColumnLabel;
java.awt.TextField ColumnsTextField;
java.awt.Label RowLabel;
java.awt.TextField RowsTextField;
java.awt.Label XminLabel;
java.awt.Label YminLabel;
java.awt.TextField XminTextField;
java.awt.TextField YminTextField;
java.awt.Label XmaxLabel;
java.awt.Label YmaxLabel;
java.awt.TextField XmaxTextField;
java.awt.TextField YmaxTextField;
java.awt.Button OkButton;
java.awt.Button CancelButton;
java.awt.Button HelpButton;
java.awt.Label InputGCPLLabel;
java.awt.Button InputGCPButton;
java.awt.Checkbox MemoryCheckbox;
symantec.itools.awt.BorderPanel CorrectionPanel;
java.awt.Checkbox ImageToMapRadioButton;
CheckboxGroup TargetGroup;
java.awt.Checkbox ImageToImageRadioButton;
java.awt.Checkbox MapToMapRadioButton;
java.awt.Label TargetLabel;
java.awt.Label Order;
java.awt.Checkbox FirstRadioButton;
CheckboxGroup OrderGroup;
java.awt.Checkbox SecondRadioButton;
java.awt.Checkbox ThirdRadioButton;
java.awt.Label BackValueLabel;
java.awt.TextField BackValueTextField;
java.awt.Choice OperatorChoice;
//}}

GeoCubeIOTextField InputTextField;
GeoFileIOTextField InputGCPTextField;
GeoCubeIOTextField OutputTextField;

class SymWindow extends java.awt.event.WindowAdapter
{
    public void windowClosing(java.awt.event.WindowEvent event)
    {
        Object object = event.getSource();
        if (object == GeoGCPCorrectDialog.this)

```

```

        GeoGCPCorrectDialog_WindowClosing(event);
    }
}

void GeoGCPCorrectDialog_WindowClosing(java.awt.event.WindowEvent event)
{
    postDlgActionEvent("CANCEL");
}

class SymAction implements java.awt.event.ActionListener
{
    public void actionPerformed(java.awt.event.ActionEvent event)
    {
        Object object = event.getSource();
        if (object == InputBrowseButton)
            InputBrowseButton_ActionPerformed(event);
        else if (object == InputGCPButton)
            InputGCPButton_ActionPerformed(event);
        else if (object == OutputBrowseButton)
            OutputBrowseButton_ActionPerformed(event);
        else if (object == OkButton)
            OkButton_ActionPerformed(event);
        else if (object == CancelButton)
            CancelButton_ActionPerformed(event);
        else if (object == HelpButton)
            HelpButton_ActionPerformed(event);
    }
}

void InputBrowseButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    GeoEnviron.CubeListDlg.addDlgActionListener(this);
    GeoEnviron.CubeListDlg.setVisible(true);
}

void InputGCPButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    GeoEnviron.OpenFileDialog.setTitle("Select matrix file");
    GeoEnviron.OpenFileDialog.setDirectory(GeoEnviron.CurrentPath);
    GeoEnviron.OpenFileDialog.setFile("*."+System.getProperty("geopixel.matrix.ext"));
    GeoEnviron.OpenFileDialog.setVisible(true);
    GeoEnviron.CurrentPath = GeoEnviron.OpenFileDialog.getDirectory();
    String fname = GeoEnviron.OpenFileDialog.getFile();
    if(fname == null) return;
    java.io.File f = new java.io.File(GeoEnviron.CurrentPath, fname);
    try {
        this.Mtx = new Matrix();
        this.Mtx.read(f);
        if(this.Mtx.getColumns() != 4)
            throw new IllegalArgumentException();
        InputGCPTextField.setFile(f);
    }catch(IllegalArgumentException e) {
        this.Mtx = null;
        GeoEnviron.message(this, "Illegal Format of GCP Matrix");
    }catch(IOException e) {
        this.Mtx = null;
        GeoEnviron.message(this, "Cannot read GCP Matrix File");
    }
}

class SymItem implements java.awt.event.ItemListener
{
    public void itemStateChanged(java.awt.event.ItemEvent event)

```

```

        {
            Object object = event.getSource();
            if (object == MemoryCheckbox)
                MemoryCheckbox_ItemStateChanged(event);
            else if (object == ImageToMapRadioButton)
                ImageToMapRadioButton_ItemStateChanged(event);
            else if (object == ImageToImageRadioButton)
                ImageToImageRadioButton_ItemStateChanged(event);
            else if (object == MapToMapRadioButton)
                MapToMapRadioButton_ItemStateChanged(event);
        }
    }

    void OutputBrowseButton_ActionPerformed(java.awt.event.ActionEvent event)
    {
        GeoEnviron.SaveFileDialog.setDirectory(GeoEnviron.CurrentPath);
        GeoEnviron.SaveFileDialog.setFile("*"+System.getProperty("geopixel.cube.ext"));
        GeoEnviron.SaveFileDialog.setVisible(true);
        GeoEnviron.CurrentPath = GeoEnviron.SaveFileDialog.getDirectory();
        String fname = GeoEnviron.SaveFileDialog.getFile();
        if(fname == null) return;
        fname = GeoUtility.getExtendedFilename(fname, System.getProperty("geopixel.cube.ext"));
        OutputTextField.setMode(Target.CUBE_MODE);
        OutputTextField.setTarget(
            new Target.File(GeoEnviron.CurrentPath, fname));
    }

    private int getDegree() {
        int degree = 1;
        if(FirstRadioButton.getState())
            degree = 1;
        else if(SecondRadioButton.getState())
            degree = 2;
        else if(ThirdRadioButton.getState())
            degree = 3;
        return degree;
    }

    private int getMapping() {
        int mapping = GCPCorrectOp.IMAGE_TO_MAP;
        if(ImageToMapRadioButton.getState()) {
            mapping = GCPCorrectOp.IMAGE_TO_MAP;
        }else if(ImageToImageRadioButton.getState()) {
            mapping = GCPCorrectOp.IMAGE_TO_IMAGE;
        }else if(MapToMapRadioButton.getState()) {
            mapping = GCPCorrectOp.MAP_TO_MAP;
        }
        return mapping;
    }

    private int getResampling() {
        int Resampling = GeoResampler.NEAREST_NEIGHBOR;
        String method = OperatorChoice.getSelectedItem();
        if (method.equalsIgnoreCase("NEAREST NEIGHBOR"))
            Resampling = GeoResampler.NEAREST_NEIGHBOR;
        else if (method.equalsIgnoreCase("BI-LINEAR"))
            Resampling = GeoResampler.BILINEAR;
        else if (method.equalsIgnoreCase("CUBIC"))
            Resampling = GeoResampler.CUBIC;
        else if (method.equalsIgnoreCase("B-SPLINE"))
            Resampling = GeoResampler.B_SPLINE;
        return Resampling;
    }

```



```

}

private void setMemoryTarget() {
    String fname = OutputTextField.getText();
    if(fname.length() > 0) {
        fname = GeoUtility.getExtendedFilename(fname, System.getProperty("geopixel.cube.ext"));
        Target tgt = new Target.Memory(fname);
        OutputTextField.setMode(Target.CUBE_MODE);
        OutputTextField.setTarget(tgt);
    }else {
        OutputTextField.clear();
    }
}

void OkButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    if(MemoryCheckbox.getState()) {
        setMemoryTarget();
    }

    Object[] arg = new Object[10];

    arg[0] = InputTextField.getTarget();
    if(arg[0] == null) {
        GeoEnviron.message(this, "No input image");
        return;
    }

    int[] band = null;
    if (InputTextField.isBandMode()) {
        band = new int[1];
        band[0] = InputTextField.getBandIndex();
    }
    else {
        int b = ((Target)arg[0]).HD.Bands;
        band = new int[b];
        for (int i = 0; i < band.length; i++) band[i] = i;
    }
    arg[1] = band;

    arg[3] = this.Mtx;
    if(arg[3] == null) {
        GeoEnviron.message(this,"GCP Matrix file not specified.");
        return;
    }

    arg[2] = OutputTextField.getTarget();
    if(arg[2] == null) {
        GeoEnviron.message(this, "No output image");
        return;
    }

    if(arg[0].equals(arg[2])) {
        GeoEnviron.message(this, "Output image is equal to input image");
        return;
    }

    arg[4] = new Integer(this.getMapping());
    arg[5] = new Integer(this.getDegree());
    arg[6] = new Integer(this.getResampling());
    arg[7] = new Integer(BackValueTextField.getText());

    int[] destPixel = new int[2];
    int smps = new Integer(ColumnsTextField.getText()).intValue();
}

```

```

int lns = new Integer(RowsTextField.getText()).intValue();
    destPixel[0] = smps; destPixel[1] = lns;
    arg[8] = destPixel;

GeoRectangle destBounds = new GeoRectangle();
destBounds.xmin =new Double(XminTextField.getText()).doubleValue();
    destBounds.ymin =new Double(YminTextField.getText()).doubleValue()
    if(this.getMapping() == GCPCorrectOp.IMAGE_TO_IMAGE) {
    destBounds.xmax = destBounds.xmin+smps;
    destBounds.ymax = destBounds.ymin+lns;
    }else {
    destBounds.xmax =new Double(XmaxTextField.getText()).doubleValue();
    destBounds.ymax =new Double(YmaxTextField.getText()).doubleValue();
    }
    arg[9] = destBounds;

postDlgActionEvent("OK",arg);
setVisible(false);
}

void CancelButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    postDlgActionEvent("CANCEL",null);
}

void HelpButton_ActionPerformed(java.awt.event.ActionEvent event)
{
    // to do: code goes here.
}

class SymMouse extends java.awt.event.MouseAdapter
{
    public void mousePressed(java.awt.event.MouseEvent event)
    {
        Object object = event.getSource();
        if (object == MemoryCheckbox)
            MemoryCheckbox_MousePressed(event);
    }
}

void MemoryCheckbox_MousePressed(java.awt.event.MouseEvent event)
{
    OutputTextField.setEnabled(true);
    if(MemoryCheckbox.getState())
        OutputBrowseButton.setEnabled(true);
    else
        OutputBrowseButton.setEnabled(false);
}

public void dlgActionPerformed(DlgActionEvent e) {
    Object obj = e.getSource();
    if(obj == GeoEnviron.CubeListDlg) {
        String choice = e.getChoice();
        if(choice == "SELECT") {
            Object arg = e.getArgument();
            Target tgt = null;
            if(arg instanceof Target) {
                tgt = (Target)arg;
                InputTextField.setMode(Target.CUBE_MODE);
                InputTextField.setTarget(tgt);
            }else {
                Object[] a = (Object[])arg;
                tgt = (Target)a[0];
            }
        }
    }
}

```

```

        InputTextField.setMode(Target.BAND_MODE);
        InputTextField.setTarget(tgt);
        InputTextField.setBandIndex(((Integer)a[1]).intValue());
    }
//    updateOutputParameter();
}
}
}

/***** Common Part of Dialog *****/
//focus transfer and dialog event listener
protected Window pWindow = null;
protected DlgActionListener dlgActionListener = null;

//set precursor
public void setPrecursor(Window precursor) {
    pWindow = precursor;
}

//posting Dialog Event
public synchronized void addDlgActionListener(DlgActionListener l) {
    dlgActionListener = l;
}

public synchronized void removeDlgActionListener(DlgActionListener l) {
    if(dlgActionListener == l) dlgActionListener = null;
}

protected void processEvent(AWTEvent e) {
    if (e instanceof DlgActionEvent) {
        processDlgActionEvent((DlgActionEvent)e);
    }else
        super.processEvent(e);
}

protected synchronized void processDlgActionEvent(DlgActionEvent e) {
    if(dlgActionListener != null)
        dlgActionListener.dlgActionPerformed(e);
}

protected void postDlgActionEvent(String cmd) {
    postDlgActionEvent(cmd, (Object)null);
}

protected void postDlgActionEvent(String cmd, Object dlgAction) {
    DlgActionEvent e = new DlgActionEvent(this, cmd, dlgAction);
    Toolkit toolkit = getToolkit();
    EventQueue queue = toolkit.getSystemEventQueue();
    queue.postEvent(e);
}

//show dialog at the center of screen
public void setVisible(boolean b) {
    if(b) {
        Rectangle component = bounds();
        Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
        setLocation((screen.width - component.width)>>1,
            (screen.height - component.height)>>1);
        super.setVisible(b);
    }else {
        super.setVisible(b);
        if(pWindow != null)
            pWindow.requestFocus();
    }
}
}

```

```

/***** End of Common Part *****/

void MemoryCheckbox_ItemStateChanged(java.awt.event.ItemEvent event)
{
    if(event.getStateChange() == ItemEvent.SELECTED) {
        OutputTextField.setEditable(true);
        OutputBrowseButton.setEnabled(false);
    }else {
        OutputTextField.setEditable(false);
        OutputBrowseButton.setEnabled(true);
    }
}

void ImageToMapRadioButton_ItemStateChanged(java.awt.event.ItemEvent event)
{
    if(event.getStateChange() == ItemEvent.SELECTED) {
        setLabelForMap();
    }
}

void ImageToImageRadioButton_ItemStateChanged(java.awt.event.ItemEvent event)
{
    if(event.getStateChange() == ItemEvent.SELECTED) {
        setLabelForImage();
    }
}

void MapToMapRadioButton_ItemStateChanged(java.awt.event.ItemEvent event)
{
    // to do: code goes here.
    if(event.getStateChange() == ItemEvent.SELECTED) {
        setLabelForMap();
    }
}

private void setLabelForMap() {
    XminLabel.setText("Xmin :");
    XmaxLabel.setText("Xmax :");
    YminLabel.setText("Ymin :");
    YmaxLabel.setText("Ymax :");
    XmaxTextField.setEditable(true);
    YmaxTextField.setEditable(true);
    XmaxTextField.setForeground(XmaxLabel.getForeground());
    YmaxTextField.setForeground(YmaxLabel.getForeground());
}

private void setLabelForImage() {
    XminLabel.setText("Start Column :");
    XmaxLabel.setText("End Column :");
    YminLabel.setText("Start Row :");
    YmaxLabel.setText("End Row :");
    XmaxTextField.setEditable(false);
    YmaxTextField.setEditable(false);
    XmaxTextField.setForeground(XmaxTextField.getBackground());
    YmaxTextField.setForeground(YmaxTextField.getBackground());
}

private void updateOutputParameter() {
    Target tgt = InputTextField.getTarget();
    if(tgt == null) return;
    ColumnsTextField.setText(String.valueOf(tgt.HD.Samples));
    RowsTextField.setText(String.valueOf(tgt.HD.Lines));
    if(ImageToImageRadioButton.getState()) {

```

```
XminTextField.setText(String.valueOf(0));
XmaxTextField.setText(String.valueOf(tgt.HD.Samples-1))
YminTextField.setText(String.valueOf(0));
YmaxTextField.setText(String.valueOf(tgt.HD.Lines-1));
)else {
    XminTextField.setText(String.valueOf(tgt.HD.Xmin));
    XmaxTextField.setText(String.valueOf(tgt.HD.Xmax));
    YminTextField.setText(String.valueOf(tgt.HD.Ymin));
    YmaxTextField.setText(String.valueOf(tgt.HD.Ymax));
}
)
}
```

## 부록 6. GeoPixel-Terrain API 설명서

지형분석 소프트웨어는 CMainFrame Class, CTerrainApp Class, CTerrainDoc Class, CTerrainView Class, CViewDlg1, CViewDlg2, 그외 32개의 class로 구성 되어 있다.

### ① CTerrainApp Class

Terrain.cpp와 Terrain.h의 2개의 파일로 구성되어 있다. CTerrainDoc, CMainFrame, CTerrain View class등 모든 클래스를 서로 연결시켜주는 프로그램의 중심적인 역할을 하는 클래스이다.

### ② CMainFrame Class

mainfrm.cpp와 mainfrm.h의 2개의 파일로 구성된다. 메인 윈도우와 툴바와 같은 메뉴의 체계에 대한 화면구성을 하는 클래스이다.

### ③ CTerrainDoc Class

Terraindoc.cpp와 Terraindoc.h의 2개의 파일로 구성된다. Viewing 소프트웨어의 사용되는 주된 데이터 구조를 정의하고 자료의 저장 및 처리를 하는 클래스이다.

### ④ CTerrainView Class

TerrainView.cpp와 TerrainView.h의 2개의 파일로 구성된다. 메인 윈도우에 화면에 대한 출력을 담당하는 클래스이다.

### ⑤ CViewDlg1 Class

ViewDlg1.cpp와 ViewDlg1.h의 2개의 파일로 구성된다. View1의 화면에 대한 출력과 제어를 하는 클래스이다.

### ⑥ CViewDlg2 Class

ViewDlg2.cpp와 ViewDlg2.h의 2개의 파일로 구성된다. View2의 화면에 대한 출력과 제어를 하는 클래스이다.

⑦ 그외의 클래스

AboutDlg, ALPHADialog, CAnalHeight, CAnalHeight1, CAnalHeight2, CBestPath, CCollisionDialog, CConeDialog, CContour, CCylinderDialog, CDcylinderDialog, CDomeDialog, CLoadProgress, CNoAreaDialog, CNoExtractDlg, CNoPositionDialog, CNoSelectDem, CObjectBrowserDialog, CPathNum, CProgress, CPyramidDialog, CRoofDialog, CSelectRenType, CSelectSimType, CSplashWnd, CSroofDialog, CThreshold, CubicDialog, CViewPathDialog, CZHEIGHTDialog, CZheightDir 등의 클래스로 구성된다. Viewing 소프트웨어의 입력을 받는 다이얼로그와 에러를 출력하는 다이얼로그들등 각종 다이얼로그에 대한 정보를 관리하는 역할을 한다.

◎ TerrainApp

```
virtual BOOL InitInstance();  
afx_msg void OnAppAbout();  
afx_msg void OnFileOpen();  
afx_msg void OnFileOpenDem();  
afx_msg void OnFileOpenSc();
```

◎ MainFrm

```
OnCreate(LPCREATESTRUCT lpCreateStruct);  
PreCreateWindow(CREATESTRUCT & cs);
```

◎ TerrainDoc

```
int LoadHeader();
```

DEM 데이터의 전처리 파일을 읽어 들이는 함수.

```
int ReadAllDEM();
```

사용자가 선택한 DEM file의 내용을 읽어 들이는 함수.

```
int LoadSatImg();
```

사용자가 선택한 DEM file의 인공위성 영상 데이터를 읽어 들이는 함수.

void ReadSelectDEM();

사용자가 선택한 DEM file중에서 일부분을 선택했을 경우 그 부분을 읽어 들이는 함수.

void Make\_GLlist();

OpenGL을 사용하여 지형의 3차원 wireframe을 구성하는 함수. 여기에서 지형 분석에 관계된 wireframe을 같이 구성한다.

void Make\_ShadeGLlist();

OpenGL을 사용하여 셰이딩된 지형의 3차원 구조를 만드는 함수.

Zpoint cnormal(Zpoint a, Zpoint b, Zpoint c);

각 포인트의 노말을 계산하는 함수.

void GetNormal();

셰이딩시 필요한 평면의 노말을 계산하는 함수.

◎ TIN 구성에 이용되는 모듈들

void Extract();

void rad(void);

void save\_pt();

void read\_specific();

void get\_centroid();

void sort\_pt();

int Compare( twopt p1, twopt p2);

double Area(twopt a,twopt b, twopt c);

twopt sub2d(twopt a,twopt b);

double length2d(twopt r);

double dot2d(twopt a,twopt b);

void radial(void);

void get\_boundary(list\_ptr node);

void get\_tr();

void push(int \*top,int item);



```

void pt2tri(int a,int b, int c, tran_ptr tri);
int pop(int *top);
void concave();
void deleteli(list_ptr *ptr, list_ptr trail, list_ptr node);
void show();
void convex1();
void convex2();
void convex3();
int check_nb(tran_ptr p1, tran_ptr p2);
int WhichSide(twopt p,twopt q,twopt r);
void ReadTIN();
void Make_TINlist();
double norm3d(Zpoint p);

```

◎ Height Field Ray Tracing에 이용되는 함수들.

```

Zpoint Init_Ray(double u, double v, Zpoint N, double d);
Zpoint SetPoint(double x, double y, double z);
Zcolor HeightFieldRT(Zpoint start, Zpoint ray_vector, int depth,int row,int col);
Zcolor SetColor(int r,int g,int b);
double Shading(Zpoint view, Zpoint hit_point, Zpoint h_vector, double ks, double kd,
double ka, int np);
double GetIntensity(Zpoint h_vector, Zpoint view, Zpoint hit_point, double ks, double
kd, double ka, int np);
Zpoint Reflection(Zpoint view, Zpoint hit_point, Zpoint h_vector);
Zpoint HitTest(Zpoint start, Zpoint ray_vector, Zpoint *h_vector, int depth);
int CheckBoundingBox(Zpoint start, Zpoint Ray);
Zpoint DDDA(void);
int TestCell(Zpoint Near,Zpoint Far, Zpoint c1,Zpoint c2, Zpoint c3, Zpoint c4);
Zpoint TestPoly(Zpoint Near, Zpoint Far, Zpoint Cell1, Zpoint Cell2, Zpoint Cell3,

```

Zpoint Cell4);

int InsideTest(Zpoint c1, Zpoint c2, Zpoint c3, Zpoint p,Zpoint pn);

void Make\_Texturelist(void);

void Make\_TINTextlist(void);

void transflat();

View1의 윈도우와 3차원 지형 사이의 매핑을 위해서 2차원과 3차원 사이의 화면상의 좌표 매핑을 계산하는 함수.

void EvalAngle();

지형분석의 경사각을 계산하는 함수.

void GetMinMax();

지형분석시 선택된 지형의 DEM 데이터중에서 최대의 고도값과 최소의 고도 값을 구하는 함수.

void eval\_distance8();

최단거리 분석시 중심점으로부터 주위의 거리를 구하는 함수.

double eval\_weight(int x1, int y1, int x2, int y2);

최단거리 분석시 점과 점의 거리에 경사도를 이용하여 거리에 가중치를 주는 함수.

void shortestpath(int start\_node, double \*cost,int node\_num,int end\_node);

최단거리 분석시 구성된 노드들의 거리에 최소값들을 저장하는 함수.

int choose(double distance[],int n, short int found[]);

최단거리 분석시 인접한 노드로 가는 최소의 거리를 가지는 노드를 선택하는 함수.

void Make\_GLlist\_BestPath();

OpenGL을 사용하여 최단거리를 3차원 상에 매핑하기 위한 함수.

void Viewshed\_test();

가시권 분석시 선택된 점으로부터 다른 점의 visibility를 검사하는 함수.

void sort\_view\_area();

가시권 분석시 설정된 previewpoint로부터 가장 많은 가시 영역순으로 정렬하

는 함수.

```
void eval_view_area(int index);
```

가시권 분석시 보이는 영역을 얻어내는 함수.

```
void Pick_view_point();
```

가시권 분석시 정의된 최대의 previewpoint의 개수만큼 viewpoint를 설정하는 함수.

```
void Make_GLlist_View();
```

OpenGL을 사용하여 가시영역을 3차원 상에 매핑하기 위한 구조를 설정하는 함수.

```
void LoadScriptFile(CArchive& ar);
```

ScriptFile의 내용을 읽어 오는 함수.

```
void StoreScriptFile(CArchive& ar);
```

ScriptFile을 만들어 저장하는 함수.

```
double makeplane(CPoint, CPoint);
```

지형의 평탄화를 하는 함수.

```
int makepath();
```

사용자가 입력한 점들로부터 비행경로를 생성해내는 함수.

```
void CrossMat(double A[], int a1,int a2, double B[], int b1,int b2, double C[]);
```

행렬의 cross 연산을 수행하는 함수.

```
void InverseMat(double Ag[], double Bg[]);
```

행렬의 inverse를 계산하는 함수.

```
void Collision();
```

비행 경로를 생성했을 경우에 지형에 비행경로가 충돌을 하는지를 검사하는 함수.

```
//등고선...////////////////////
```

```
void MainRoutine(int);
```

전체 등고선의 모듈을 통합하여 등고선을 구성하는 함수.

```
void InitializeColor(void);
```

각 등고선마다 부여하는 색깔을 정의하는 함수

void InitializeView(void);

view reference point를 정의하는 함수

void ContourTest(void);

DEM에서 높이평면과 지형의 교점을 테스트하기 위해 DEM을 삼각형화한 후  
교점 테스트하는 함수를 호출하는 함수

void IntersectionTest(void);

실제 높이 평면과 지형의 교점이 존재하는지를 테스트하고 교점을 구하는 함수

Zpoint ReturnDEM(int,int);

DEM의 index를 넘겨주면 DEM을 구성하는 point를 리턴하는 함수

Zpoint equalTo(Zpoint);

하나의 point를 다른 변수에 넘겨주는 함수

void sameTest(int);

삼각형의 세 직선에서 구해진 교점이 동일점인지를 테스트하는 함수

int compare(Zpoint,Zpoint);

sameTest() 함수에서 교점간의 비교를 위한 함수

void TinContourTest();

ContourTest()함수를 TIN에서 동일하게 처리하는 함수

void TinMain(int);

MainRoutine()함수를 TIN에서 동일하게 처리하는 함수

void ReadContourTIN(void);

Contour생성을 위해 TIN을 file로부터 읽어들이는 함수

Zpoint ReturnTIN(int);

TIN의 index를 넘겨주면 TIN을 구성하는 point를 리턴하는 함수

void Initparameter();

DISTANCE, EYE, CRC point를 초기화하는 함수

void Make\_GList\_DEMcontour(int);

contour를 구성하기 위한 DEM의 wireframe make\_glist()를 만드는 함수

void Make\_GList\_TINcontour(int);

```

    contour를 구성하기 위한 TIN의 make_gllist()를 만드는 함수
void Make_GLlist_DEMcontour1(int k);
    contour를 구성하기 위한 DEM의 shading make_gllist()를 만드는 함수

////////Tin to DEM////////////////////////////////////
void Make_GLlist_TINtoDEM(int);
    DEM으로 변환하기 위해 TIN의 make_gllist()를 만드는 함수
double minT(double,double,double);
    삼각형의 point중 높이가 가장 작은 point를 리턴하는 함수
double maxT(double,double,double);
    삼각형의 point중 높이가 가장 큰 point를 리턴하는 함수
int InsideTest(int,double,double,Zpoint,Zpoint,Zpoint);
    삼각형의 내부에 point가 존재하는지를 결정하는 함수
Zpoint ComputeMid(Zpoint,Zpoint);
    두 point의 중간값을 계산하는 함수
double linearLagrange(int,double,double);
    2차 lagrange함수값을 계산하는 함수
double quadraticLagrange(int,double,double);
    4차 lagrange함수값을 계산하는 함수
double basis0(Zpoint,Zpoint,Zpoint,double,double);
    삼각형의 세 직선중 하나에 대한 basis function을 구성하는 함수
double basis1(Zpoint,Zpoint,Zpoint,double,double);
    삼각형의 세 직선중 하나에 대한 basis function을 구성하는 함수
double basis2(Zpoint,Zpoint,Zpoint,double,double);
    삼각형의 세 직선중 하나에 대한 basis function을 구성하는 함수
double dot2d(Zpoint,Zpoint);
    2차원적인 dot product를 계산하는 함수
Zpoint normalize2d(Zpoint);
    2차원적인 normalize를 계산하는 함수

```

int compareXY(double,double,Zpoint);

point의 x,y와 동일한 값인지를 비교하는 함수

int equationLine(double,double,double,double,double,double);

line 방정식의 함수값을 계산하는 함수

void biquadratic(void);

TIN의 내부에서 DEM의 각 point를 보간하는 함수

double averageZ(int,int);

높이값 z의 평균값을 구하는 함수

int findIndex(int,int);

index값이 XCOUNT 또는 YCOUNT의 범위내에 속하는지를 테스트하는 함수

void WriteDem();

DEM 데이터를 파일에 저장하는 함수

#### ◎ TerrainView

void DrawAllDEM(CDC \*pDC);

사용자가 선택한 DEM file을 수치고도에 따라서 화면에 출력해주는 함수.

void RedrawLine(CDC \*pDC);

사용자가 선택한 DEM file중에 일부분을 선택할 때의 사각형을 그려주는 함수.

void WireTIN(void);

TIN을 2차원 상의 wireframe으로 출력해주는 함수.

void ShadeTIN(void);

OpenGL을 사용하여 만든 TIN shading 구조를 출력할 때 사용하는 함수.

void StartHeightRT(CDC \*pDC);

void DrawSatImage(CDC \*pDC);

인공위성 영상을 출력하는 함수.

void DrawPath();

비행경로를 출력해주는 함수.

void SimulWireSceneGivenPath();

wireframe 비행시물레이션을 출력해주는 함수.

void SimulTexture();

Texture mapping된 지형에서 비행시물레이션을 출력해주는 함수.

void SimulShadeScene();

Shading된 지형에서 비행시물레이션을 출력해주는 함수.

void GLSetupRC2();

void GLSetupRC3();

OpenGL의 초기값을 설정해주는 함수.

void DrawScene1();

void DrawScene2();

OpenGL을 사용하여 화면의 출력을 구성한 부분을 출력해주는 함수.

#### ◎ ViewDlg1

void draw2dDEM(CDC \*pDC);

사용자가 선택한 DEM을 수치고도 모델로 출력하는 함수.

void OneViewCase(int num,CDC \*pDC);

View1의 모든 경우를 가지고 필요한 함수를 호출할 때 사용하는 함수.

void InitDraw2dDEM(CDC \*pDC);

전체 DEM에서 사용자가 선택한 부분만 화면의 크기에 매핑시켜 수치고도 모델로 출력하는 함수.

void DrawArea(CDC \*pDC);

사용자가 선택하는 부분을 사각형으로 그려주는 함수.

void initBuffer();

화면에 출력할 때 깜박임을 없애기 위해 사용하는 buffer을 초기화 하는 함수.

void DrawBestPath(CDC \*pDC);

View1에서 2차원 적으로 최단 거리를 그려주는 함수.

void DrawPosition(CDC \*pDC);

View1에서 사용자가 선택한 점을 그려주는 함수.

void DrawPath(CDC \*pDC);

View1에서 사용자가 입력한 비행경로의 기본점으로 2차원 곡선으로 비행경로를 그려주는 함수.

void DrawContour2d(CDC \*);

View1에서 추출된 등고선을 수치고도 모델에 매핑시켜 2차원으로 그려주는 함수.

void Wire2DTIN(CDC \*pDC);

의미점 추출이 끝나고 난후에 3각형화를 2차원 적으로 보여 주는 함수.

### ◎ ViewDlg2

void TwoViewCase(int num);

View2에서 사용하는 함수를 제어하고 호출하는 함수.

void DrawTinIcon();

의미점 추출을 한후에 의미점에 대응하는 아이콘과 의미점의 개수를 출력하는 함수를 호출하는 함수.

void peak\_icon(int , int, CDC \*);

peak 아이콘을 그리고 추출된 peak의 수를 출력한다.

void pit\_icon(int , int, CDC \*);

pit 아이콘을 그리고 추출된 pit의 수를 출력한다.

void pass\_icon(int , int, CDC \* );

pass 아이콘을 그리고 추출된 pass의 수를 출력한다.

void ravine\_icon(int , int, CDC \* );

ravine 아이콘을 그리고 추출된 ravine의 수를 출력한다.

void ridge\_icon(int , int, CDC \* );

ridge 아이콘을 그리고 추출된 ridge의 수를 출력한다.

void break\_icon(int , int, CDC \* );

break 아이콘을 그리고 추출된 break의 수를 출력한다.

void slope\_icon(int , int, CDC \* );

slope 아이콘을 그리고 추출된 slope의 수를 출력한다.



void flat\_icon(int , int, CDC \* );

flat 아이콘을 그리고 추출된 flat의 수를 출력한다.

void TextAngle\_Gradient();

경사각 분석을 했을 때 정보를 출력하는 함수.

void TextMinMax();

지역을 선택했을 때 그 지역의 최대 고도와 최소 고도등의 정보를 출력하는 함수.

void TextMoveClay();

절토량에 대한 정보를 출력하는 함수.

void TextFillClay();

성토량에 대한 정보를 출력하는 함수.

void TextFillWater();

저수량에 대한 정보를 출력하는 함수.

void Display\_Watershed();

Watershed의 결과를 2차원적으로 출력하는 함수.

void TextPath();

최단거리의 결과를 출력하는 함수.

void TextViewarea();

가시권 분석의 결과를 출력하는 함수.

void GLSetupRCtwo();

OpenGL을 사용하기 위해서 환경설정을 하는 함수.

void ShadeCurrentPath();

셰이딩 비행 시뮬레이션시 비행 경로를 화면에 출력하는 함수.

void WireCurrentPath();

와이어프레임 비행 시뮬레이션시 비행 경로를 화면에 출력하는 함수.

void DrawScene();

OpenGL로 구성된 함수를 화면에 출력하는데 사용하는 함수.

void GLSetupRC();

OpenGL을 사용하기 위해서 환경설정을 하는 함수.

void DrawObj();

건물을 위치 시켰을 때 그 건물의 편집을 위하여 건물을 그려주는 함수.

void DrawCoord();

사용자가 건물을 선택하고 수정을 할 때 변하는 것을 그려주는 함수.

## 주 의

1. 이 보고서는 과학기술부에서 시행한 특정연구개발사업의 연구보고서입니다.
2. 이 보고서 내용을 발표할 때에는 반드시 과학기술부에서 시행한 특정연구개발사업의 연구결과임을 밝혀야 합니다.
3. 국가과학기술 기밀유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 아니됩니다.