

차세대 디지털 동영상 기술개발

Development of Advanced Digital Moving Image Technology

연구기관
한국과학기술연구원

과 학 기 술 처

제출문

과학기술처장관 귀하

본 보고서를 “차세대 디지털 동영상 기술 개발” 과제의 3차년도 보고서로 제출합니다.

1997년 12월 31일

주관기관명: 한국과학기술연구원

연구책임자: 김 형곤

선임연구원: 안 상철

연구원: 이 제호

김 상욱

학연학생: 김 남규

김 남호

함 상진

이 광도

조 오영

여 백

I 제목

차세대 디지털 동영상 기술 개발

II 연구개발의 목적 및 중요성

1. 목적

차세대 디지털 동영상관련 핵심 알고리즘 연구와 이들을 효율적으로 실시간 환경에서 구현하여 실제 사용 가능한 응용예를 보이는 것을 목적으로 한다. 또한 향후 그 중요성이 크게 부각될 것으로 사료되는 지능형 인간-기계 접속 기술의 핵심인 대화형 가상공간에서 효율적으로 사용 가능한 기반기술을 구축 하고져 하며, 3차원 공간의 물체 입력기술과 사람을 중심으로하는 상호 대화형 기술 개발을 목적으로 하였다.

2. 연구개발의 중요성

디지털 동영상 기술은 멀티미디어 기술의 도래와 함께 그 중요성이 크게 부각된 분야이다. 특히 문자, 소리, 컴퓨터 그래픽스 등의 기술이 혼합화 하여 새로운 미디어 기술을 이루면서 다른 미디어와 비교하여 상대적으로 큰 연산량과 컴퓨터 하드웨어를 요구하는 동영상 기술은 저장 및 전송의 효율성을 위하여 영상 압축 및 복원기술이 핵심으로 연구되어 왔다. JPEG 및 MPEG 연구는 이러한 연구의 중요성이 잘 인식된 결과이다. 표준화 기술의 정착으로 압축 및 복원기술이 어느정도 정착된 상황에서 이들의 효율적인 구현을 목적으로하는 대규모 집적회로 시스

템 기술이 활발한 연구를 이루었으며 MPEG-2 부호화/복호화 처리기, 움직임 검출기 등 영상통신을 가능케 하는 기반기술들이 표준화 작업의 결과로 활발한 연구가 최근 수년간 진행 되었다.

최근들어 부상되는 차세대 디지털 동영상 기술에는 스테레오 영상의 효율적인 표시와 전송 및 저장을 위한 스테레오 영상기술, 디지털 영상을 이용하여 다양한 영상효과 및 편집 시스템을 가능케 하는 디지털 영상 저작 및 생성기술, 가상현실과 컴퓨터 그래픽스 등을 실제 영상과 혼합하여 새로운 가상의 공간을 생성하고 이러한 가상의 공간과 상호 반응하는 대화형 가상공간 기술 등이 새로운 분야로 대두되고 있다. 특히 미국의 디지털 텔레비전 방송 개시와 함께 방송 기술의 디지털화는 영상 시스템의 응용분야가 영화산업, 광고산업, 텔레비전 방송산업, 케이블 텔레비전 등 모든 분야가 영상의 형태 및 제작의 상호관계 등 새로운 대대적인 표준화 작업이 밀물처럼 홍수를 이룰 것으로 예상된다.

Ⅲ. 연구개발 범위

본 연구를 마감하는 3차 연도의 연구는 1차 및 2차 연도에서 연구 되어진 알고리즘과 기법들을 효율적으로 실시간 환경에서 구현하여 실제 사용 가능한 응용예를 보이는 것을 목적으로 하였다. 또한 향후 그 중요성이 크게 부각될 것으로 사료되는 지능형 인간-기계 접속 기술의 핵심인 대화형 가상공간에서 효율적으로 사용 가능한 기반기술을 구축 하고져 하여 3차원 공간의 물체 입력기술과 사람을 중심으로하는 상호 대화형 기술을 개발 하였다. 이들을 위한 구체적인 연구 개발의 범위는 다음과 같다.

- 스테레오 영상 입력 시스템

- 고성능 스테레오 정합 알고리즘
- 실시간 얼굴 검출 및 추적 시스템
- 얼굴 및 제스처 인식 시스템

IV. 연구개발의 결과 및 활용에 대한 건의

고성능 스테레오 영상획득기술, 스테레오 정합 알고리즘 및 이의 실시간 구현기술, 얼굴영역 추출기술 및 추적기술, 얼굴 식별 및 제스처 인식기술 등은 본연구에서 기본적인 연구를 수행하였을 따름이지 결코 이와 관련된 연구가 마무리 되었다고 말할수 없다. 하지만 본 연구를 통하여 개발되고 축적된 기술들은 향후 이 분야의 연구에 기반이 되어서 많은 응용분야가 개발되길 바란다. 특히 본 연구를 통해 개발된 실시간 처리 능력을 갖는 스테레오 정합 처리기 기술과 병렬 DSP를 이용한 얼굴 영역 추적기술등은 산업화가 즉시 가능한 분야로 기업화를 위한 노력이 필요하며 이를위한 지원을 기대한다. 또한 개발된 관련 핵심 기술들이 향후 연구되어지는 대화형 가상환경기술 등에서 효율적으로 사용될수 있도록 기대해 보며 이를위해 지속적인 연구가 진행 되어야 할 것이다. 또한 최근 디지털 영상 미디어에 대해 고조되는 관심에 실제적으로 부응할 수 있는 연구 결과의 활용에 기대를 걸어본다. 특히 얼굴영상 추적기술은 방송사의 얼굴 마스크 기술 등에 응용이 가능하므로 방송사와의 공동연구를 통해 실제적인 응용분야 개발이 필요할 것으로 사료된다.

여 백

-Summary-

I Title

Development of advanced digital moving image technologies (III)

II Purpose and Significance of the Study

1. Purpose

The purpose of this project is to study core technologies in the field of the advanced digital moving image to implement efficient real-time applications.

2. Significance of the Study

The importance of the real-time intelligent man-machine interface technology can not be over-emphasized nowadays with the flooding of data everywhere. The future of the information technology depends on the novel interface technology based on the computer vision technology. The vision-based interactive technology is expected to become core technology for the future.

III. Scope of the study

For the development of the advanced digital moving image technologies, following subjects are studied.

- Acquisition System of Stereo Image
- High-Performance Stereo Matching Algorithm

- Real-Time Face Detection and Tracking System
- Face and Gesture Recognition System

IV. Results and Discussion

Each research subjects are summarized as follows.

- Acquisition System of Stereo Image: In this field of study, the technics for the calibration of stereo Camera are studied first. For the calibration of the camera in software, the registration of the stereo Images are also studied and implemented. Using the suggested calibration technics, the range measurement with stereo vision have been implemented.

- High-Performance Stereo Matching Algorithm: MPC stereo matching algorithm have suggested for the high-performance stereo matching. This algorithm results high accuracy matching especially in the area of the object boundary, and enables high throughput rates. The automatic reconstruction of the face surface using disparity map are also studied although not complete yet. The hardware structure for the real-time processing of the suggested MPC algorithm is also described.

- Real-Time Face Detection and Tracking System: For the real-time face detection, the concept of moving color have introduced. Facial feature extraction algorithm using moving color have suggested. Real-time face tracking system based on the general purpose DSP board are also described.

- Face and Gesture Recognition System: For the implementation of the face and gesture recognition, face recognition algorithms are studied first. The face

recognition algorithm using PCA features and BRF neural net algorithm for face recognition are described with some results.

The results of this research is only the basics of the related research fields, and should be extended for the real application. Especially, the virtual environments for the future information interface could be one of the most eminent application of the developed technology. Also, the VLSI architecture for the MPC disparity estimation and facial feature extraction/tracking could be used for the immediate industrial application.

여 백

- 목차 -

제 1장 서론	15
제 2장 스테레오 영상 입력 시스템	19
1절 스테레오 카메라 교정 (Calibration)	19
2절 스테레오 영상 보정	30
3절 스테레오 비전 거리 측정	41
제 3장 고성능 스테레오 정합 알고리즘	45
1절 MPC 스테레오 정합 알고리즘	45
2절 고성능 MPC 구현 알고리즘	50
3절 얼굴표면 자동생성 (Reconstruction)	78
4절 실시간 구현을 위한 하드웨어 구조	81
제 4장 실시간 얼굴 검출 및 추적 시스템	97
1절 움직임 색상정보 알고리즘	99
2절 움직임 색상을 이용한 얼굴 특징점 추출	117
3절 실시간 얼굴 추적 시스템	135
4절 결론 및 향후 연구방향	139
제 5장 얼굴 및 제스처 인식 시스템	143
1절 얼굴인식 알고리즘	143
2절 PCA 특징값을 이용한 얼굴 인식	146
3절 BRN 신경망 알고리즘	159
제 6장 결론	173

여 백

- Contents -

Chapter 1. Introduction	15
Chapter 2. Aquisition System of Stereo Image	19
2.1. Calibration of Stereo Camera	19
2.2. Registeration of the Stereo Images	30
2.3. Range Measurement with Stereo Vision	41
Chapter 3. High-Performance Stereo Matching Algorithm	45
3.1. MPC Stereo Matching	45
3.2. High-Performance MPC Algorithm	50
3.3 Automatic Reconstruction of the Face Surface	78
3.4. Hardware Structure for the Real-Time Processing	81
Chapter 4. Real-Time Face Detection and Tracking System	97
4.1. Moving Color Algorithm	99
4.2. Facial Feature Extraction Using Moving Color	117
4.3. Real-Time Face Tracking System	135
Chapter 5. Face and Gesture Recognition System	143
5.1. Face Recognition Algorithm	143
5.2. Face Recognition Using PCA Features	146
5.3. BRF Neural Net for Face Recognition	159
Chapter 6. Conclusions	173

여 백

제 1 장. 서론

디지털 동영상 기술은 멀티미디어 기술의 도래와 함께 그 중요성이 크게 부각된 분야이다. 컴퓨터의 특수 효과를 이용한 주라기 공원, 스타트랙, 등의 첨단 영상 산업의 활성화가 국가 경제에 미치는 영향은 미국의 경우를 보면 그 효과를 충분히 짐작하게 된다. 특히 문자, 소리, 컴퓨터 그래픽스 등의 기술이 혼합화 하여 새로운 미디어 기술을 이루면서 다른 미디어와 비교하여 상대적으로 큰 연산량과 컴퓨터 하드웨어를 요구하는 동영상 기술은 저장 및 전송의 효율성을 위하여 영상 압축 및 복원기술이 핵심으로 연구되어 왔다. JPEG 및 MPEG 연구는 이러한 연구의 중요성이 잘 인식된 결과이다.

표준화 기술의 정착으로 압축 및 복원기술이 어느정도 정착된 상황에서 이들의 효율적인 구현을 목적으로하는 대규모 집적회로 시스템 기술이 활발한 연구를 이루었으며 DVD, Set-Top-Box, HDTV, Web을 기반으로한 멀티미디어 정보 활용 기술, MPEG-2 부호화 및 복호화 처리기, 움직임 검출기 등 영상통신을 가능케 하는 기반기술들이 표준화 작업의 결과로 활발한 연구가 최근 수년간 진행 되었다.

이러한 기술을 바탕으로 하여 최근들어 부상되는 차세대 디지털 동영상 기술에는 스테레오 영상의 효율적인 표시와 전송 및 저장을 위한 스테레오 영상기술, 디지털 영상을 이용하여 다양한 영상효과 및 편집 시스템을 가능케 하는 디지털 영상 저작 및 생성기술, 가상현실과 컴퓨터 그래픽스 등을 실제 영상과 혼합하여 새로운 가상의 공간을 생성하고 이러한 가상의 공간과 상호 반응하는 대화형 가상공간 기술 등이 새로운 분야로 대두되고 있다. VRML (Virtual Reality Modeling Language)를 이용한 인터넷상의 3차원 그래픽스와 가상공간을 이용하는 새로운

정보환경은 전자오락, 출판 및 저작, 교육, 가전산업, 시각기반 컴퓨터 등의 광범위한 분야에 영향을 미칠것이 확실시 되며, 이들을 뒷받침 하는 시각기반 컴퓨터 기술에는 실시간 운영체제 (Real-Time Operating System), 3차원 장면 표시기술, 정보의 압축 및 복원 뿐만 아니라 3차원 그래픽스를 이용한 가상의 합성영상 처리를 가능케하여 기능성을 강조하여 표준안을 제공하는 MPEG4 표준안 기술, 3차원 그래픽스 가속기를 기반으로하는 영상 렌더링 (Image Rendering), Affine 변환 등의 3차원 변환을 효율적으로 처리하는 기술이 필요하게 되었다.

특히 미국의 디지털 텔레비전 방송 개시와 함께 방송 기술의 디지털화는 영상 시스템의 응용분야가 영화산업, 광고산업, 텔레비전 방송산업, 케이블 텔레비전 등 모든 분야가 영상의 형태 및 제작의 상호관계 등 새로운 대대적인 표준화 작업이 밀물처럼 홍수를 이룰 것으로 예상된다.

본 연구의 1차 연도에는 초 저전송율 영상 전송을 위한 차세대 영상 압축 알고리즘을 주로 연구 하였다. 1차 연도의 주요 연구 내용은 차세대 고효율 영상 압축 알고리즘 개발, 이들의 실시간 구현을 위한 초고집적회로 (VLSI) 구조 연구 및 극저전송율 비디오 코덱 시스템의 구현 등이다. 극저전송율 비디오 압축 알고리즘 연구는 표준 알고리즘으로 대두되고 있는 H.263 알고리즘과 이들을 향상 시키기 위한 영역분할 (Segmentation) 에 근거한 고압축 알고리즘, 웨이브렛 변환 (Wavelet Transform)을 이용한 고압축 알고리즘 등에 대해 조사 하였으며, 차세대 부호화 기술이라는 모델기반 부호화 기법에 대해 연구 하였다. 실시간 구현을 위한 초고집적회로 (VLSI) 구조 연구에서는 파이프라인 분산연산 처리를 이용하는 이산 역변환 (Discrete Cosine Transform) 처리기 및 이의 역변환 (Inverse Transform) 처리기 구조를 제시하고 설계 하였다. 또한 비트 수준의 파이프라인 처리 구조를 갖는 양자화기 및 역 양자화기의 설계도 수행 하였다. 극저전송율 비

디오 코덱 시스템의 구현과 관련해서는 병렬 디지털 신호 처리기 시스템 구조와 이를 이용한 비디오 코덱 시스템 구현을 하여 극저 전송율 비디오 코덱 시스템의 성능을 검토 하였다. 4개의 TMS320C40 DSP 처리기를 병렬로 사용하였고 비디오 입출력으로 NTSC 입출력 보드를 이용하여 더블버퍼링을 이용하여 병목현상을 제거 하였다.

본 연구의 2차 연도에는 차세대 정보환경을 가능케 하는 디지털 동영상 정보의 추출, 처리, 분석 등의 관련 핵심 원천 기술을 주로 연구 하였다. 2차 연도의 주요 연구 내용은 3차원 모델 생성, 관심영역 검출과 응용기술 및 표시기술과 3차원 공간상의 물체의 움직임 검출 및 추적 기술을 다루었다. 먼저, 3차원 물체의 복원을 위한 스테레오 변위 추정기술에 대한 실험을 수행하였으며 이를 이용한 표면 재구성에 관한 이론적인 고찰을 수행 하였다. 관심영역 검출기법으로는 대칭성을 이용한 얼굴의 특징점 추출과 얼굴영역 분할 및 이의 고속 구현을 위한 시스템 환경 등을 다루었다. 3차원 공간상의 물체의 움직임 검출 및 추적 기술과 관련하여서는 Optical Flow를 이용한 얼굴의 3차원 움직임 추적 기술과 동영상으로부터 얻은 특징점들의 위치를 구하고 이들로부터 순차형 Singular Value Decomposition (SVD) 기법을 이용해 물체의 3차원 재구성과 3차원 움직임 정보를 분리하는 알고리즘을 개발 하였다.

본 연구를 마감하는 3차 연도의 연구는 1차 및 2차 연도에서 연구 되어진 알고리즘과 기법들을 효율적으로 실시간 환경에서 구현하여 실제 사용 가능한 응용예를 보이는 것을 목적으로 하였다. 또한 향후 그 중요성이 크게 부각될 것으로 사료되는 지능형 인간-기계 접속 기술의 핵심인 대화형 가상공간에서 효율적으로 사용 가능한 기반기술을 구축 하고자 하여 3차원 공간의 물체 입력기술과 사람을 중심으로하는 상호 대화형 기술을 개발 하였다.

본 보고서는 다음과 같이 구성 되어 있다. 먼저, 서론에 이어 제 2장에서는 스테레오 영상 입력 시스템과 관련하여 스테레오 카메라 교정 (Calibration), 스테레오 영상 보정 (Registration), 스테레오 비전을 이용한 거리 측정 등을 설명한다. 제 3장에서는 고성능 스테레오 정합 알고리즘인 MPC 스테레오 정합 알고리즘을 설명하고, 중복성을 제거한 고성능 알고리즘 구현과 이의 실시간 구현을 위한 하드웨어 구조를 설명한다. 또한 이를 이용한 얼굴표면의 자동생성 (Reconstruction) 알고리즘에 대하여 설명한다. 제 4장에서는 실시간 얼굴 검출 및 추적 시스템과 관련하여 먼저 움직임 색상정보의 개념을 설명하고 이를 이용한 얼굴 특징점 추출 알고리즘을 설명한후, 이를 얼굴 추적 시스템에 적용하여 실시간 구현한 시스템에 대해 소개한다. 제 5장에서는 얼굴 및 제스처어 인식 시스템 구현을 위한 얼굴인식 알고리즘에 대해 소개하고, 본연구에서 수행한 PCA 특징값을 이용한 얼굴 인식 결과와 BRN 신경망 알고리즘을 이용해 구현한 얼굴인식 결과를 설명한후 제 6장에서 결론을 논한다.

본 연구를 통해 개발된 핵심 기술들이 향후 연구되어지는 대화형 가상환경기술 등에서 효율적으로 사용될수 있도록 기대해보며 이를위해 지속적인 연구가 진행되어야 할 것이다. 또한 최근 디지털 영상 미디어에 대해 고조되는 관심에 실제적으로 부응할 수 있는 연구 결과의 활용에 기대를 걸어본다. 향후 컴퓨터를 이용한 첨단 디지털 영상 제작기술 분야의 국내 기술발전에 도움이 되었으면 기대한다.

제 2 장 스테레오 영상 입력 시스템

1절. 스테레오 카메라 교정(Calibration)

가. 카메라 캘리브레이션(calibration)의 목적

두 개 이상의 카메라를 사용하는 스테레오 카메라 시스템은 이들 카메라 사이의 위치 관계를 형성하기 이전에 먼저 각각의 카메라에 대한 3차원 공간상의 위치에 대한 2차원 영상간의 위치 관계와, 이러한 상이 실제 화면에 표시되는 경우 화소수와의 관계 설정이 이루어 져야 한다. 그러므로 각각의 카메라 캘리브레이션은 컴퓨터 비전에서 가장 기본적인 과정이며, 3차원 공간과 카메라 화상 평면 (image plane)과의 대응 관계로부터 카메라 내부의 기하학적 · 광학적 특성을 나타내는 내부 파라미터 (intrinsic parameter)와 기준 좌표계에 대한 카메라 좌표계의 위치와 방향을 표시하는 외부 파라미터 (extrinsic parameter)를 결정 하는 과정을 의미한다. 카메라의 캘리브레이션 데이터는 크게보면 3차원 공간의 특징점과 2차원 화상 평면 (image plane) 사이의 대응 관계로부터 얻어진다. 이를 다시 세분하면 기준 좌표계에 대한 카메라의 공간적 위치와 방향에 관한 3차원 정보 관계를 설정하는 외부 변수의 관계와 이를 이용하여 실제 화면상에서 화소의 센서구조에 따라 결정되는 화소수의 단위로 표현되는 내부 파라미터를 구하는 과정으로 나누어 진다. 본 절에서는 카메라 캘리브레이션의 원리와 실제 교정방법 및 실험 결과를 설명한다.

나. 캘리브레이션의 원리

a. 카메라 모델

카메라 모델은 카메라 내부 변수 (intrinsic parameters)와 카메라 외부 변수 (extrinsic parameters)로 나누어 나타낼 수 있다. 카메라 내부 변수는 카메라의 실제 하드웨어에 기인한 변수이고, 카메라 외부 변수는 카메라 좌표계의 기준 좌표계에 대한 회전량 (rotation)과 이동량 (translation)에 관한 변수가 된다. 여기서 카메라 내부 변수의 종류로는 초점 거리 (effective focal length), 렌즈 왜곡 (lens distortion), 카메라 센서의 가로 방향과 세로 방향 크기, 카메라 센서의 가로 방향과 세로 방향 센서의 개수, 스케일 양(s_x) 등이다. 카메라 내부 변수는 카메라 제작자가 제공하는 카메라 명세서 (camera specification)에 나오는 것으로 사용 할 수 있다.

카메라 외부 변수는 캘리브레이션의 실제 출력 값이다. 이 외부 변수는 카메라의 3차원 위치와 방향을 나타내는 변수로서 카메라의 방향을 나타내는 회전 (rotation) 성분과 위치를 나타내는 이동 (translation) 성분으로 구성 된다. 기준 좌표계와 카메라 좌표계 사이의 관계는 외부 변수를 사용해서 다음과 같이 나타낼 수 있다.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \quad (2.1)$$

위에서 (x, y, z) 와 (x_w, y_w, z_w) 는 각각 카메라 좌표계와 기준 좌표계에서의 좌표이고, R 은 기준 좌표계에 대한 카메라 좌표계의 회전을 나타내는 양으로서, x 축에 대한 회전량인 피치(ϕ , pitch)와 y 축에 대한 회전량인 요(θ , yaw) 그리고 z 축에 대한 회전량인 로울(ψ , roll)로 구성되는 매트릭스이다. 회전 매트릭스 R 과 이동

벡터 T 는 다음과 같다.

$$R = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \phi + \cos \psi \sin \theta \cos \phi & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \cos \theta \sin \phi \\ \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (2.2)$$

$$T = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (2.3)$$

b. 3차원 기준 좌표계로부터 영상평면으로 변환과정

3차원 기준 좌표계와 영상 평면사이의 대응 관계를 구하기 전에 먼저 3차원 기준 좌표계에서의 임의의 좌표가 영상평면으로 변환 되어가는 과정을 살펴 볼 필요가 있다. 우선 3차원 기준 좌표계에서의 좌표는 카메라 좌표계의 좌표로 변환이 되어야 한다. 이 과정에서 필요한 변수는 카메라 좌표계와 기준 좌표계사이의 회전과 이동량을 나타내는 R 과 T 이고 식 (2. 1)을 사용해서 변환 시켜 줄 수 있다. 이제 카메라 좌표계의 좌표로 변환된 좌표는 perspective transform을 이용해서 영상 평면으로 투영되는 것이다. 이 과정에서는 카메라 내부 변수가 고려되어야 한다. 위의 과정을 그림 2. 1에 다시 나타내었다.

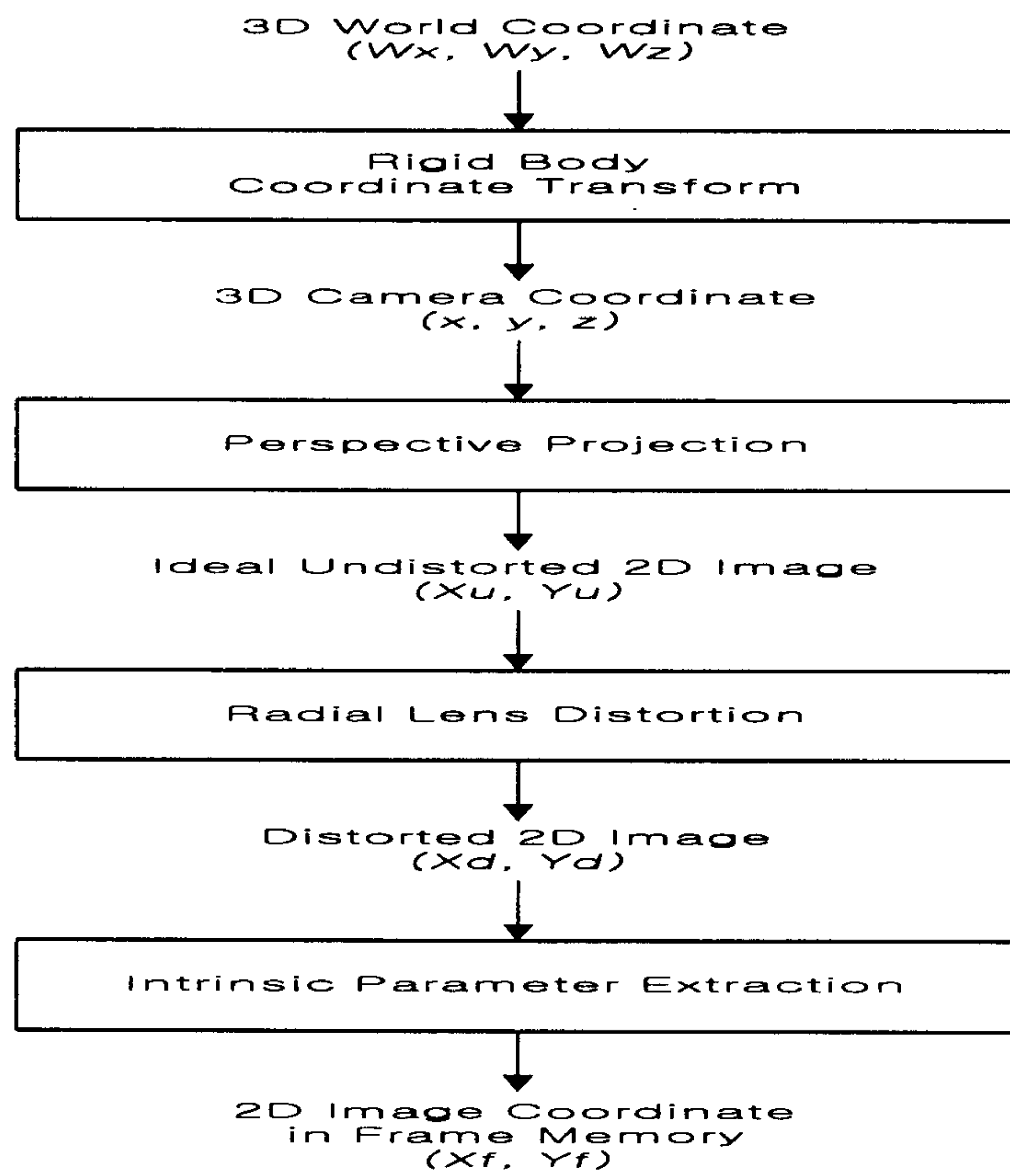


그림 2.1 카메라 캘리브레이션 과정 흐름도

다. 2 단계 카메라 캘리브레이션

카메라 캘리브레이션을 위해 Tsai 가 제안한 2단계 camera calibration 방법을 사용하였으며, 입력으로는 그림 2.2 과 같은 특징점들을 컴퓨터로 입력하여 소프트웨어적으로 각 특징점의 중심 좌표를 추출하여 사용하였다.

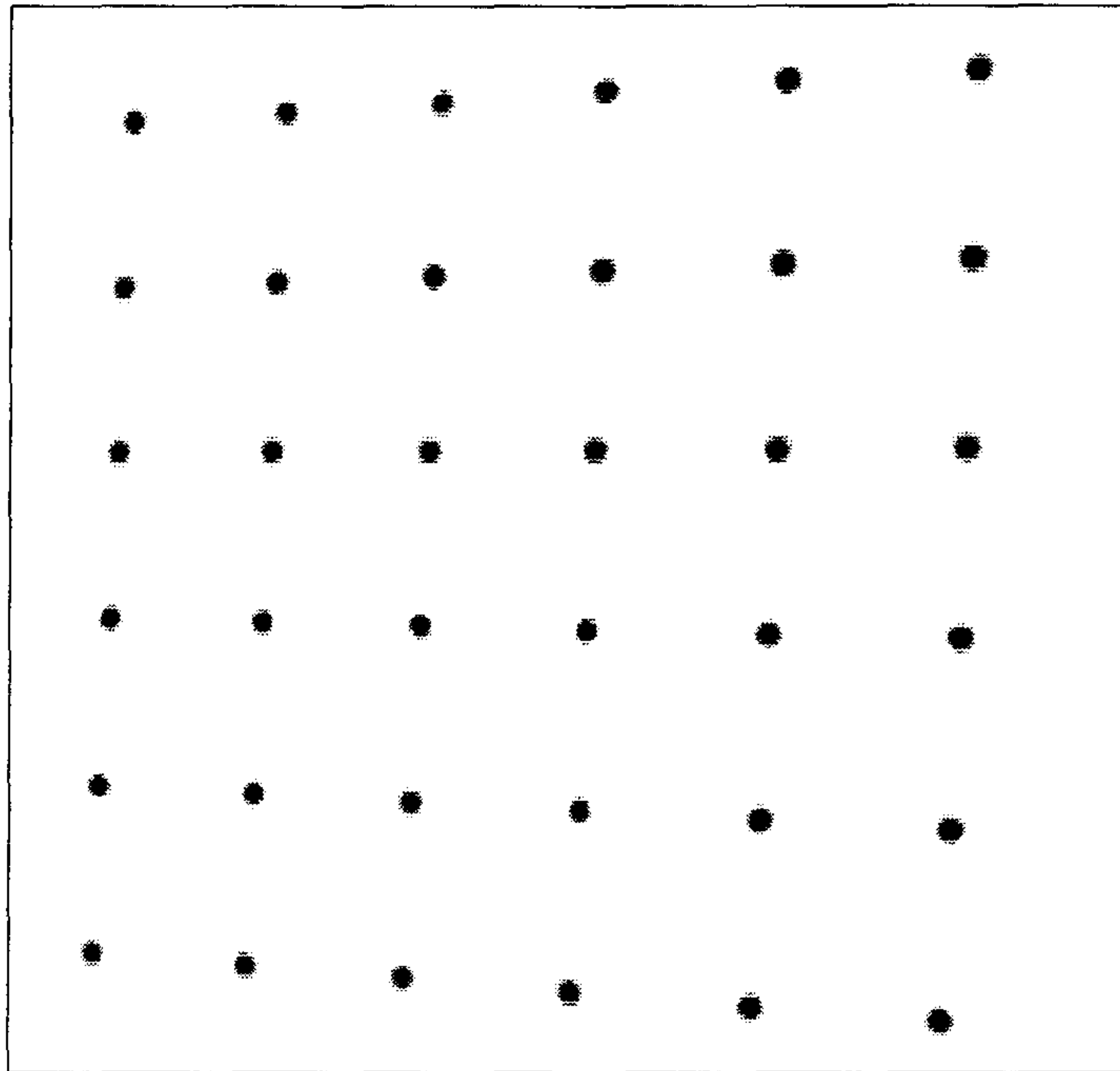


그림 2.2 캘리브레이션의 입력 특징점

2 단계 카메라 캘리브레이션은 다음과 같다. 먼저, 첫째 단계에서 회전 성분과 이동 성분중 Tx와 Ty를 구한다. 첫째 단계는 아래와 같이 크게 세가지 과정으로 나누어질 수 있다. 둘째 단계는 초점 거리, 렌즈 왜곡률, Tz에 대한 근사치를 구하는 과정과 이를 반복해서 오차를 줄여나가 최적의 값을 찾는 과정으로 나눌 수 있다.

a. 1 단계 에서는 3D orientation 과 position (Tx와 Ty)을 다음의 과정으로 구한다.

(1) 다음 식으로부터 camera lens에 의해 왜곡된 image coordinate를 구한다.

$$\begin{aligned} X_{di} &= s_x^{-1} d_x' (X_{fi} - C_x) \\ Y_{di} &= d_y (Y_{fi} - C_y) \end{aligned} \quad (2.4)$$

위 식(2.4)는 camera model로부터 나오는 식이다. 이때 (C_x, C_y) 는 카메라 좌표계에서 중심의 좌표이고, (X_{fi}, Y_{fi}) 는 ith 특징점의 프레임 메모리상의 좌표이다. 또 (X_{di}, Y_{di}) 는 렌즈 왜곡을 고려한 특징점의 좌표가 된다.

(2) $T_y^{-1}r_1, T_y^{-1}r_2, T_y^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5$ 를 구한다. 구하는 방법은 perspective transform 과 RAC(Radial Alignment Constraint)로부터 다음 식을 얻을 수 있다.

$$[Y_{di}x_{wi} \ Y_{di}y_{wi} \ Y_{di} \ -X_{di}x_{wi} \ -X_{di}y_{wi}] \begin{bmatrix} T_y^{-1}r_1 \\ T_y^{-1} \\ T_y^{-1}T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \end{bmatrix} = X_{di} \quad (2.5)$$

위의 다섯 개의 변수를 구하기 위해서는 5 개의 특징점만 있으면 되지만 오차를 줄이기 위해 이보다 많은 특징점을 입력으로 사용한다. RAC (Radial Alignment Constraint)란 그림 2.3. 에서 선분 o_1p_d 와 선분 $p_{oz}p$ 가 평행인 조건을 말한다. 이로부터 $X_d \cdot y - Y_d \cdot x = 0$ 인 조건을 얻을 수 있다. perspective transform과 RAC로부터 얻어진 $X_d \cdot y - Y_d \cdot x = 0$ 로부터 식 (2.5)를 얻을 수 있다.

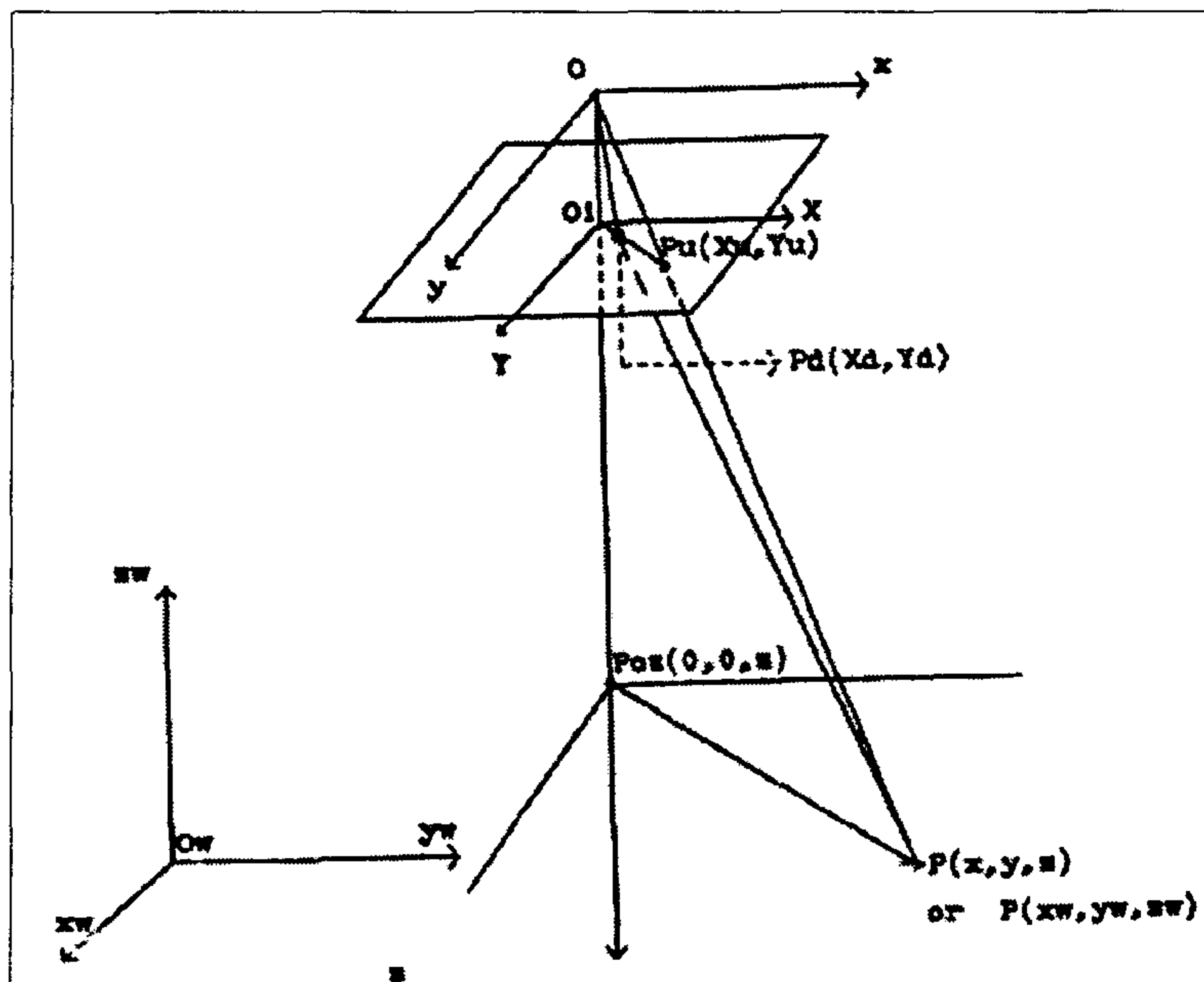


그림 2.3 RAC 조건 $o_1p_d \parallel p_{oz}p$

(3) 앞의 (2)에서 계산한 $(T_y^{-1}r_1, T_y^{-1}r_2, T_y^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5)$ 로부터 rotation matrix와 T_x, T_y 를 다음과 같은 과정으로 구한다.

i) $(T_y^{-1}r_1, T_y^{-1}r_2, T_y^{-1}T_x, T_y^{-1}r_4, T_y^{-1}r_5)$ 로부터 $|T_y|$ 를 구한다.

procedure: matrix C를 rotation matrix R의 2 x 2 submatrix로 한다면, 즉,

$$C \equiv \begin{bmatrix} r_1' & r_2' \\ r_4' & r_5' \end{bmatrix} \equiv \begin{bmatrix} r_1/T_y & r_2/T_y \\ r_4/T_y & r_5/T_y \end{bmatrix} \quad (2.6)$$

로 표현 한다면 다음과 같다.

$$T_y^2 = \frac{S_r - [S_r^2 - 4(r_1'r_5' - r_4'r_2')^2]^{1/2}}{2(r_1'r_5' - r_4'r_2')^2} \quad (2.7)$$

where $S_r = r_1'^2 + r_2'^2 + r_4'^2 + r_5'^2$

ii) T_y 의 부호를 결정한다.

procedure:

-> 임의의 한 특징점을 선택한다.

-> 우선 T_y 의 부호를 +1로 한다.

-> r_1, r_2, r_4, r_5, T_x 와 camera coordinate에서의 좌표 x, y 를 계산한다.

$$r_1 = (T_y^{-1}r_1) * T_y, \quad r_2 = (T_y^{-1}r_2) * T_y, \quad r_4 = (T_y^{-1}r_4) * T_y$$

$$r_5 = (T_y^{-1}r_5) * T_y, \quad T_x = (T_y^{-1}T_x) * T_y$$

$$x = r_1x_w + r_2y_w + T_x, \quad y = r_4x_w + r_5y_w + T_y$$

-> 위에서 계산한 x, y 와 camera image coordinate에서의 좌표 X, Y 가 같은 부호이면 $\text{sign}(T_y) = +1$, 그렇지 않으면 $\text{sign}(T_y) = -1$ 로 한다.

iii) T_y 로부터 rotation matrix와 T_x 를 구한다.

$$r_1 = (T_y^{-1}r_1) * T_y, \quad r_2 = (T_y^{-1}r_2) * T_y, \quad r_4 = (T_y^{-1}r_4) * T_y$$

$$r_5 = (T_y^{-1} r_5) * T_y, \quad T_x = (T_y^{-1} T_x) * T_y$$

$$R_1 = \begin{bmatrix} r_1 & r_2 & (1 - r_1^2 - r_2^2)^{1/2} \\ r_4 & r_5 & s(1 - r_4^2 - r_5^2)^{1/2} \\ r_7 & r_8 & r_9 \end{bmatrix} \quad \text{또는} \quad R_2 = \begin{bmatrix} r_1 & r_2 & -(1 - r_1^2 - r_2^2)^{1/2} \\ r_4 & r_5 & -s(1 - r_4^2 - r_5^2)^{1/2} \\ -r_7 & -r_8 & r_9 \end{bmatrix}$$

여기서 $s = -\text{sgn}(r_1 r_4 + r_2 r_5)$ 이다.

rotation matrix의 결정은 focal length의 부호를 가지고 결정한다. focal length가 (+)부호를 가지면 R1을 선택하고, (-)부호를 가지면 R2를 택한다.

b. 2 단계 (초점거리 f와 왜곡상수 Tz를 계산함)

(1) f와 Tz의 근사치를 계산한다. 이때 lens distortion은 무시하고 계산한다.

$$[y_i - d_y Y_i] \begin{bmatrix} f \\ T_z \end{bmatrix} = w_i d_y Y_i \quad (2.8)$$

$$\text{where } \begin{aligned} y_i &= r_4 x_{wi} + r_5 y_{wi} + r_6 \cdot 0 + T_y \\ w_i &= r_7 x_{wi} + r_8 y_{wi} + r_9 \cdot 0 \end{aligned}$$

(2) perspective transform식을 만족하는 f, Tz, distortion coefficient k1 를 구한다.

perspective transform:

$$d_y Y + d_y Y k_1 r^2 = f \frac{r_4 x_w + r_5 y_w + r_6 z_w + T_y}{r_7 x_w + r_8 y_w + r_9 z_w + T_z} \quad (2.9)$$

$$r = \sqrt{(s_x^{-1} d_x' X)^2 + (d_y Y)^2}$$

다. 카메라 calibration 실험 및 결과.

캘리브레이션 실험을 위해 우선 특징점을 만들어야 한다. 이것은 앞의 그림 2.2와 같은 영상이 되고 이 특징점들의 중심의 3차원 좌표와 이에 대응하는 이미지 평면상에서의 좌표가 캘리브레이션의 입력이 된다.

캘리브레이션 데이터의 측정오차를 알아보기 위해 처음 영상을 찍고 둘째 영상은 처음 카메라 위치로부터 z방향으로 15mm 움직이어서 영상을 얻었다. 두 카메라의 Tz방향의 차이는 14.664036으로 아주 정확한 값이 나왔다. 또 calibration 오차를 측정할 수 있는 방법으로 왜곡 화면오차 (distorted image plane error), 비왜곡 화면오차, 물체 공간오차가 있다.

이 방법은 calibration 데이터를 사용해서 image plane의 특징점의 좌표와 실제 공간상의 특징점의 좌표를 계산한후 이값의 차이를 구한 것이다. calibration 데이터로부터 2차원 영상면으로부터 3 차원 공간상의 좌표를 예측한다고 볼 때, 이러한 방법으로 calibration 오차를 계산하는 것은 아주 적절한 방법이라고 볼 수 있다. 왜곡 화면오차 (distorted image plane error)는 렌즈 왜곡상수를 고려해서 오차를 구한 것이고 비왜곡 화면오차는 렌즈 왜곡을 고려하지 않은 것이다. 물체 공간오차는 camera 중심으로부터 image plane 상의 특징점을 지나가는 광선과 3차원 실제 공간에서의 특징점과의 거리를 구한 것이다.

$t_z = 15$ [mm]

data file: tz1.out

$f = 15.775734$ [mm]

$T_x = -176.521295, T_y = -352.608628, T_z = 1035.870313$ [mm]

Rx = -142.538301, Ry = 1.572475, Rz = 91.319322 [deg]

R
-0.023016 0.793934 -0.607568
0.999358 0.001590 -0.035780
0.027441 -0.608002 -0.793461

distorted image plane error:

mean= 0.495674, stddev= 0.291781, max= 1.151857 [pix], sse= 9.83971 [pix²]

undistorted image plane error:

mean= 0.498144, stddev= 0.293458, max= 1.162616 [pix], sse= 9.9418 [pix²]

object space error:

mean= 0.244507, stddev= 0.140140, max= 0.506790 [mm], sse= 2.3630 [mm²]

data file: tz2.out

f = 15.773324 [mm]

Tx = -176.278400, Ty = -352.712449, Tz = 1021.206277 [mm]

Rx = -142.529203, Ry = 1.572426, Rz = 91.346097 [deg]

R
-0.023483 0.793837 -0.607677
0.999348 0.001955 -0.036064
-0.027441 -0.608128 -0.793365

distorted image plane error:

mean= 0.518074, stddev= 0.286741, max= 1.012915 [pix], sse= 10.4364 [pix²]

undistorted image plane error:

mean= 0.520943, stddev= 0.288299, max= 1.019671 [pix], sse= 10.5518 [pix²]

object space error:

mean= 0.252676, stddev= 0.137799, max= 0.515391 [mm], sse= 2.4660 [mm²]

|Tz1 - Tz2| = 14.664036

(15 - |Tz1 - Tz2|) / Tz2 = 3.29e-04

2절. 영상 보정(Rectification) 기술

가. 영상 보정의 정의와 필요성

영상 보정이란 영상을 우리가 원하는 형태로 만드는 일련의 과정이라 할 수 있다. 여기서 우리가 원하는 형태란 EPI (epipolar plane image)를 만족하는 영상으로 한정한다. 에피폴라 플레인 이미지란 기준 좌표계에 대한 카메라 좌표계의 원점이 같은 거리에 존재하고 높이에 해당하는 좌표가 같은 상태에서 얻은 영상을 말한다. 이렇게 해서 얻은 영상은 대응점의 위치가 같은 행에 존재하게 되어 영상 처리 시간의 단축 등 많은 잇점을 얻을 수 있다.

영상 보정을 위해서 우선 선행되어야 하는 것으로 카메라 캘리브레이션 과정이었다. 이 과정에서 카메라의 외부 변수(extrinsic parameter)와 내부 변수(intrinsic parameter)를 얻을 수 있다. 외부 변수를 사용해서 카메라의 3 차원 상의 위치를 알 수 있으나 실제 사람의 손으로 직접 원하는 위치에 맞게 카메라를 이동 시키는 일은 비효율적일 뿐만 아니라 원하는 위치로 카메라를 이동 시킨다 하더라도 오차가 너무 많게 된다. 카메라에서 원하는 영상을 얻는 것이 이처럼 쉽지 않기 때문에 많은 곳에서 로봇을 이용하거나 또는 쓰쿠바 대학처럼 영상 데이터 베이스를 만들어 놓고 다른 연구소에서 사용하도록 하기도 한다. 따라서 로봇을 사용하지 않은 이상 카메라에서 얻은 실제 영상을 가지고 실험을 하기 위해서는 영상 보정 과정이 반드시 필요하게 되는 것이다.

나. 영상 보정의 원리

a. 영상 보정과 카메라 캘리브레이션과의 관계

영상 보정을 하기 위해서는 먼저 카메라 캘리브레이션을 해야 한다. 카메라 캘리브레이션에서 나온 카메라의 외부변수를 사용해서 보정을 하게 되는 것이다. 카메라 캘리브레이션과 영상 보정의 일련의 과정을 나타내면 그림 2.4와 같다.

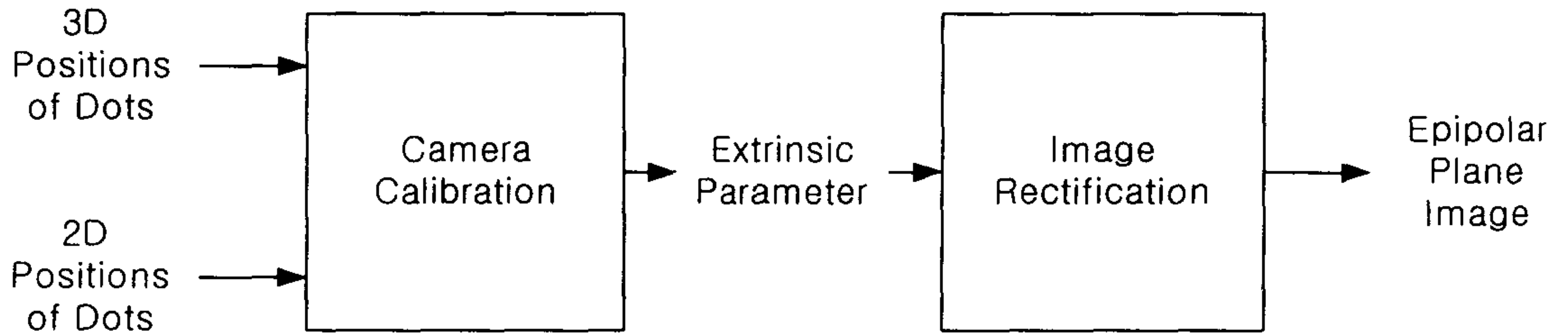


그림 2.4 카메라 교정과 보정과의 관계도

b. 영상 보정의 원리

영상 보정은 우선은 먼저 외부 변수(회전량과 이동량)를 사용해서 보정 평면을 결정해야 한다. 보정 평면은 당연히 에피플라 플레인 조건을 만족해야 할 것이다. 카메라 좌표계, 영상 평면, 보정 평면 사이의 관계는 그림 2.5에서와 같다.

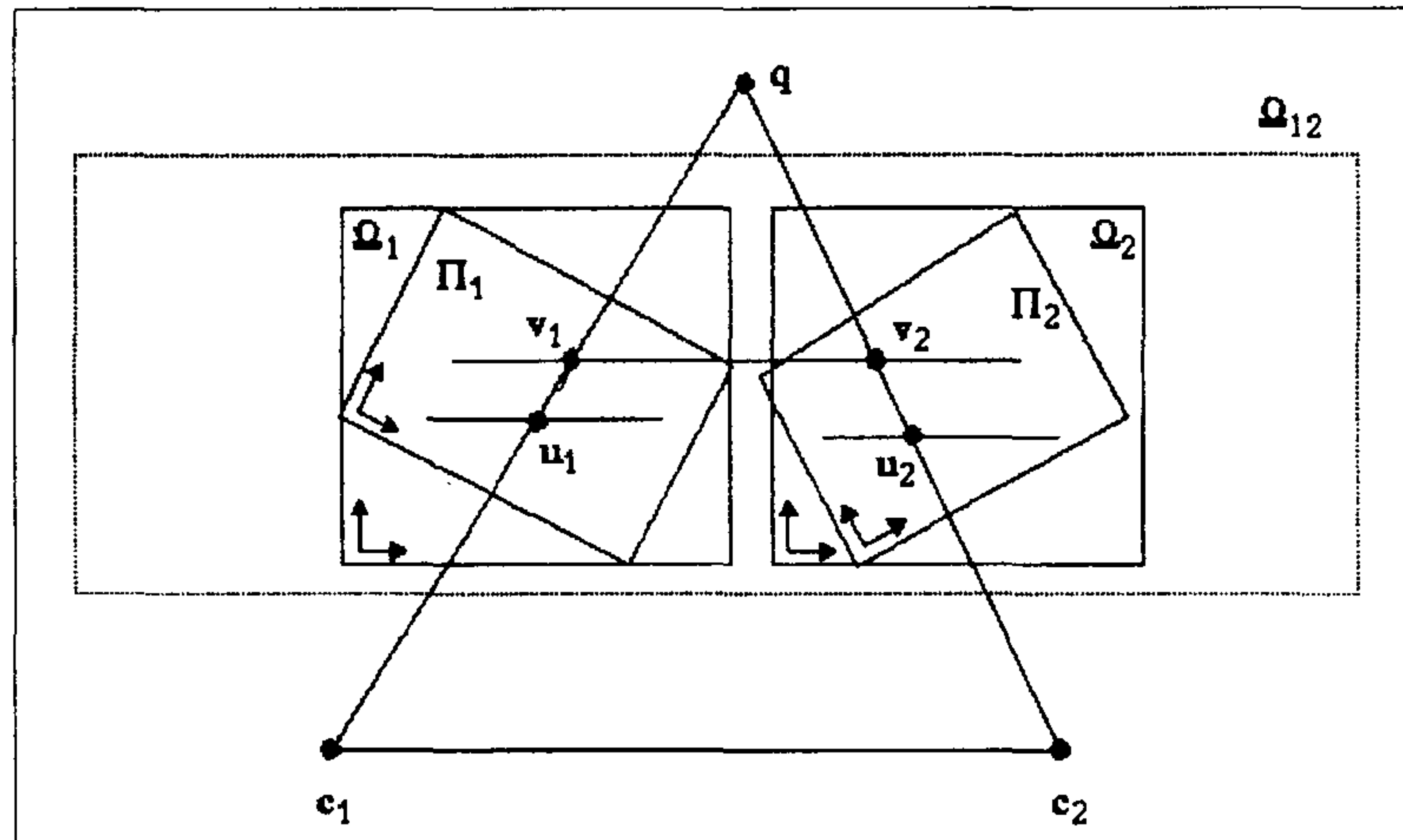


그림2.5 카메라 좌표계와 보정 평면사이의 관계

여기서 q 는 3차원 공간상의 임의의 한 점이고, u_1, u_2 는 q 점의 영상 평면 상의 위치, v_1, v_2 는 보정 평면상에서의 q 점의 위치가 된다. Π_1, Π_2 는 보정하기전의 영상 평면이고 Ω_1, Ω_2 는 보정 평면이다. Ω_{12} 는 보정 평면이 공통적으로 포함 하는 평면이다. 즉 보정 평면들은 하나의 공통된 평면내에 존재하게 된다. 따라서 영상 보정이란 u_1, u_2 와 v_1, v_2 사이의 변환관계를 구하는 과정이라 할 수 있다.

(1) 원근 변환 (Perspective transform)

원근 변환이란 3차원 공간의 점을 영상 평면으로 투영시키는 변환이다. 즉, 3차원 좌표와 영상 평면에서 대응점의 좌표사이의 관계를 나타내는 변환이라 할 수 있다. 카메라 좌표계에서 좌표값이 (x, y, z) 이고 영상 평면에서 좌표값이 (X, Y) 이고 초점 거리를 f 라 하면, 영상 평면에서 좌표값과 카메라 좌표계사이에는 삼각법의 해 다음과 같은 원근 변환이 존재하게 된다.

$$\begin{aligned} X &= f \cdot \frac{x}{z} \\ Y &= af \cdot \frac{y}{z} \end{aligned} \quad (2.10)$$

a 는 카메라 좌표계의 좌표가 영상 평면의 좌표로 변화 될 때, 영상 평면에서 한 픽셀의 가로와 세로 방향 길이의 비로 aspect ratio라 한다. 실제로 공간상의 점은 기준 좌표계에 대한 좌표값이 되므로 카메라 좌표계의 좌표값을 사용하기 위해서는 캘리브레이션에서 구한 회전량과 이동량을 이용해서 (x, y, z) 값을 먼저 구해야 한다. 따라서 최종적으로 3차원 상의 임의의 한점과 영상 평면 사이의 대응점 사이에는 다음과 같은 관계가 성립한다.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & af & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1^T & t_x \\ r_2^T & t_y \\ r_3^T & t_z \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = P \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.11)$$

식 (2.11)에서 P 를 원근 변환(perspective transform)이라 한다.

(2) 보정 원근 변환 (Rectified Perspective transform)

보정 원근 변환은 3차원상의 좌표를 보정 평면으로 가져오는 원근 변환이다. 3차원 공간의 한 점을 보정 평면 Ω_1 으로 투영시키는 보정 원근 변환을 M, 보정 평면 Ω_2 로 투영시키는 보정 원근 변환을 N이라 할때, 보정 원근 변환 M과 N은 식 (2.11)에 의해 다음과 같이 나타낼 수 있다.

$$M = \begin{bmatrix} \overrightarrow{m_1} & m_{14} \\ \overrightarrow{m_2} & m_{24} \\ \overrightarrow{m_3} & m_{34} \end{bmatrix} \quad \text{and} \quad N = \begin{bmatrix} \overrightarrow{n_1} & n_{14} \\ \overrightarrow{n_2} & n_{24} \\ \overrightarrow{n_3} & n_{34} \end{bmatrix} \quad (2.12)$$

식 (2.12)에서 먼저 m_{34} , n_{34} 의 크기를 1로 두고, $\overrightarrow{m_3} = \overrightarrow{n_3}$ 인 조건과 c_1 과 c_2 는 카메라 좌표계의 중심이라는 조건으로부터 다음식을 얻을 수 있다.

$$\overrightarrow{m_3}^T c_1 + 1 = 0 \quad , \quad \overrightarrow{m_3}^T c_2 + 1 = 0 \quad (2.13)$$

이제 보정 영상 평면의 좌표축을 결정해야 하는데, 그 좌표축은 두 카메라 좌표계의 중심을 잇는 방향 벡터와 평행하다는 조건이 있다. 이 조건으로부터 보정 원근 변환의 $\overrightarrow{m_3}$, $\overrightarrow{n_3}$ 성분을 다음과 같이 구할 수 있다.

$$\overrightarrow{m_3} = \overrightarrow{n_3} = -\frac{\overrightarrow{g}}{g^T c_1} = -\frac{\overrightarrow{g}}{g^T c_2} \quad (2.14)$$

이제, $\overrightarrow{m_2}$ 는 $\|\overrightarrow{m_2}\| = \left| \frac{af}{T_z} \right|$ 와 $\overrightarrow{m_2} = \overrightarrow{m_3} \times \overrightarrow{d_{12}}$ 조건을 이용해서 구할 수 있

고, $\overrightarrow{m_1}$ 은 $\|\overrightarrow{m_1}\| = \left| \frac{f}{T_z} \right|$ 와 $\overrightarrow{m_1} = \overrightarrow{m_2} \times \overrightarrow{m_3}$ 조건으로 구할 수 있다.

따라서, 원래의 영상 평면에서 보정 영상 평면으로 변환되는 변환식은 식 (2.15)와 같다.

$$H_1 = \begin{bmatrix} \overline{m_1^T} \\ \overline{m_2^T} \\ \overline{m_3^T} \end{bmatrix} [(\overline{p_{12}} \times \overline{p_{13}}) \quad (\overline{p_{13}} \times \overline{p_{11}}) \quad (\overline{p_{11}} \times \overline{p_{12}})] \quad (2.15)$$

마지막으로 식 (2.10)에 의해서

$$\begin{aligned} \therefore v_1 &= H_1 u_1 \\ v_{1col} &= \frac{v_{1x}}{v_{1z}} \\ v_{1row} &= \frac{v_{1y}}{v_{1z}} \end{aligned} \quad (2.16)$$

즉, 보정 평면 Ω_2 으로 변환시키는 변환도 같은 방법으로 구할 수 있다.

$$H_2 = \begin{bmatrix} \overline{n_1^T} \\ \overline{n_2^T} \\ \overline{n_3^T} \end{bmatrix} [(\overline{p_{22}} \times \overline{p_{23}}) \quad (\overline{p_{23}} \times \overline{p_{21}}) \quad (\overline{p_{21}} \times \overline{p_{22}})] \quad (2.17)$$

$$\begin{aligned} \therefore v_2 &= H_2 u_2 \\ v_{2col} &= \frac{v_{2x}}{v_{2z}} \\ v_{2row} &= \frac{v_{2y}}{v_{2z}} \end{aligned} \quad (2.18)$$

c. 영상 보정 실험

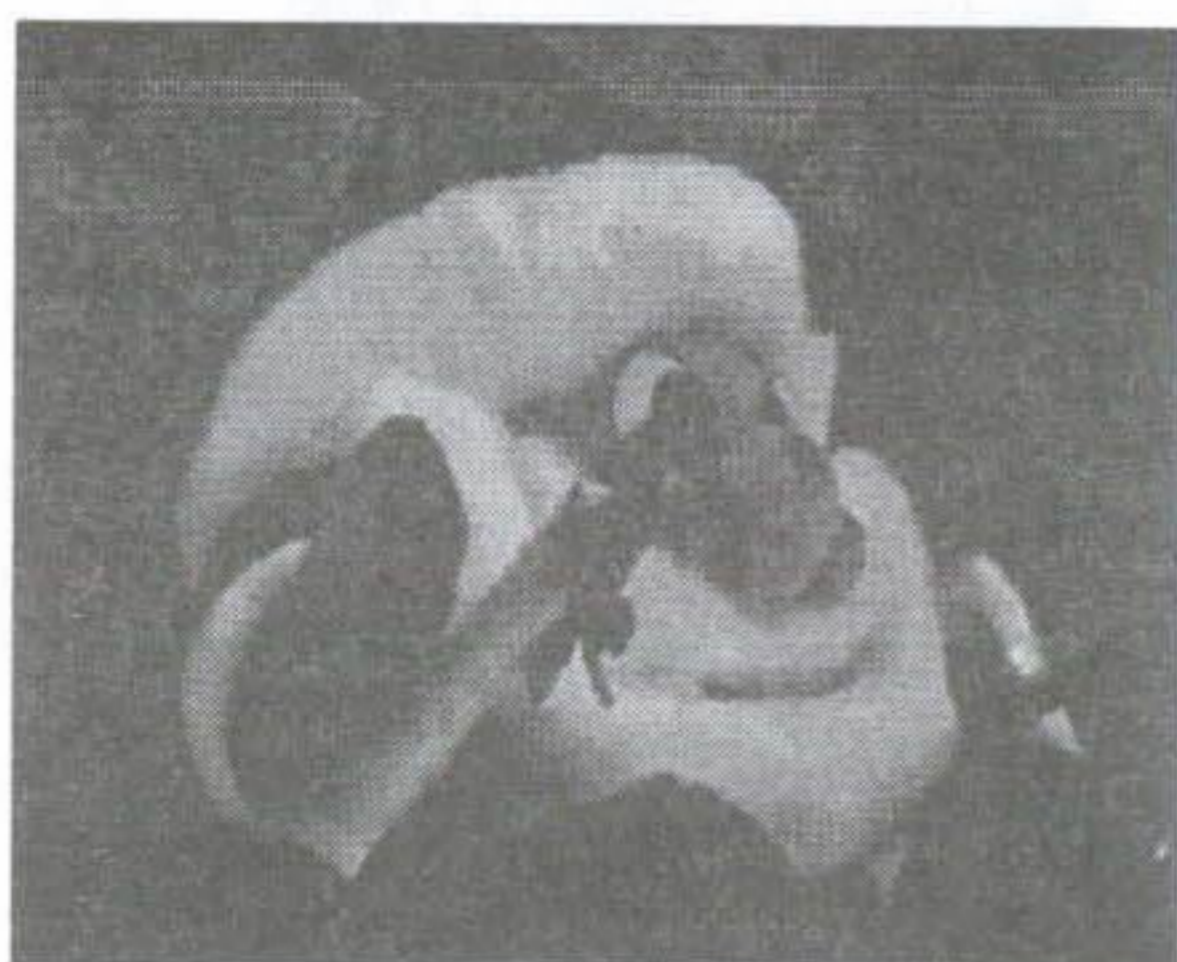
영상 보정의 실험은 아래와 같은 여러 실험 영상에 대해 적용했다. 우선 그림 2.6은 산타 영상으로 일본 쓰쿠바 대학에서 만들어 놓은 영상 데이터 베이스를 사용했다. 그림 2.7은 산타 영상을 보정 평면으로 투영 했을때의 보정 영상이 된다.

그림 2.8과 그림 2.9는 실제 본 연구실에서 영상을 얻고 캘리브레이션을 적용해서 실

제 외부 변수를 얻은 뒤 보정을 한 영상이다. 연구실 영상의 보정 영상을 확인 하기 위해 그림 2.10에 나타낸 바와 같이 영역 기반 방법에 의한 변위도(disparity map)을 구해 보았다. 윈도우를 크게 잡아서 에지부분이 선명하지는 않지만 다른 곳에서는 깊이 정보가 뚜렷이 잘 나온 것을 확인 할 수 있었다. 그림 2.11 과 그림 2.12는 회전 성분을 고려해서 영상을 얻었다. 여기서는 R_y 성분이 약 5° 정도 차이가 나도록 영상을 얻었다. 보정하기 전의 영상을 보면 픽셀의 y성분이 많은 차이가 나지만 보정 후에는 이 차이가 확연히 보정 된 것을 볼 수 있다. 각 영상에 대한 캘리브레이션 데이터는 표 2.1에 나타내었다.

표 2.1 실험 영상에 대한 캘리브레이션 데이터.

	R_x	R_y	R_z	f	T_x	T_y	T_z
Santa1	0.	0.	0.	10.	0.	0.	1000.
Santa2	0.	0.	0.	10.	20.	20.	1000.
Lab.1 translation	-143.9	1.81	90.5	15.03	-191.4	-215.8	1024.3
Lab.2 translation	-144.0	1.56	90.6	14.8	-212.2	-196.9	1188.6
Lab.3 translation & rotation	-143.9	2.7	91.4	14.8	-236.6	-214.8	1095.9
Lab.4 translation & rotation	-143.7	7.1	89.9	14.7	-234.8	-131.5	1104.3

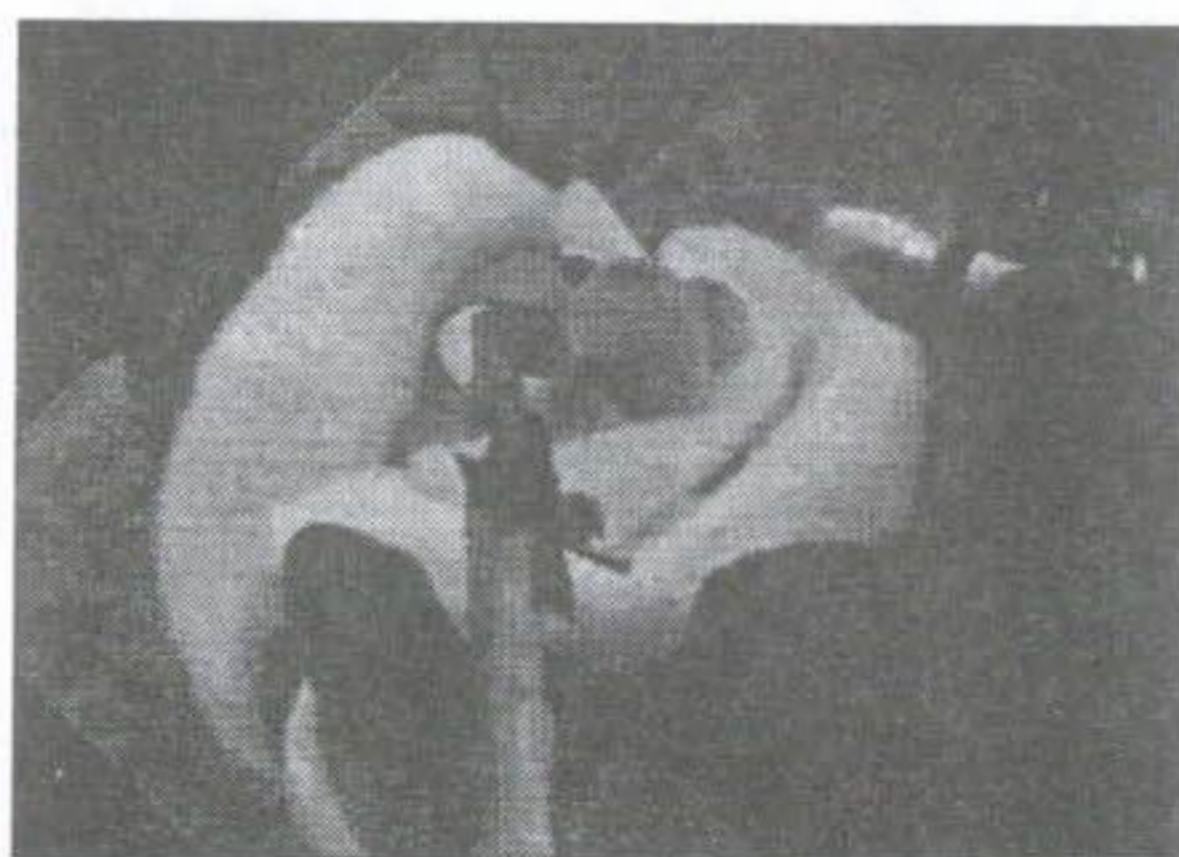


santa 1

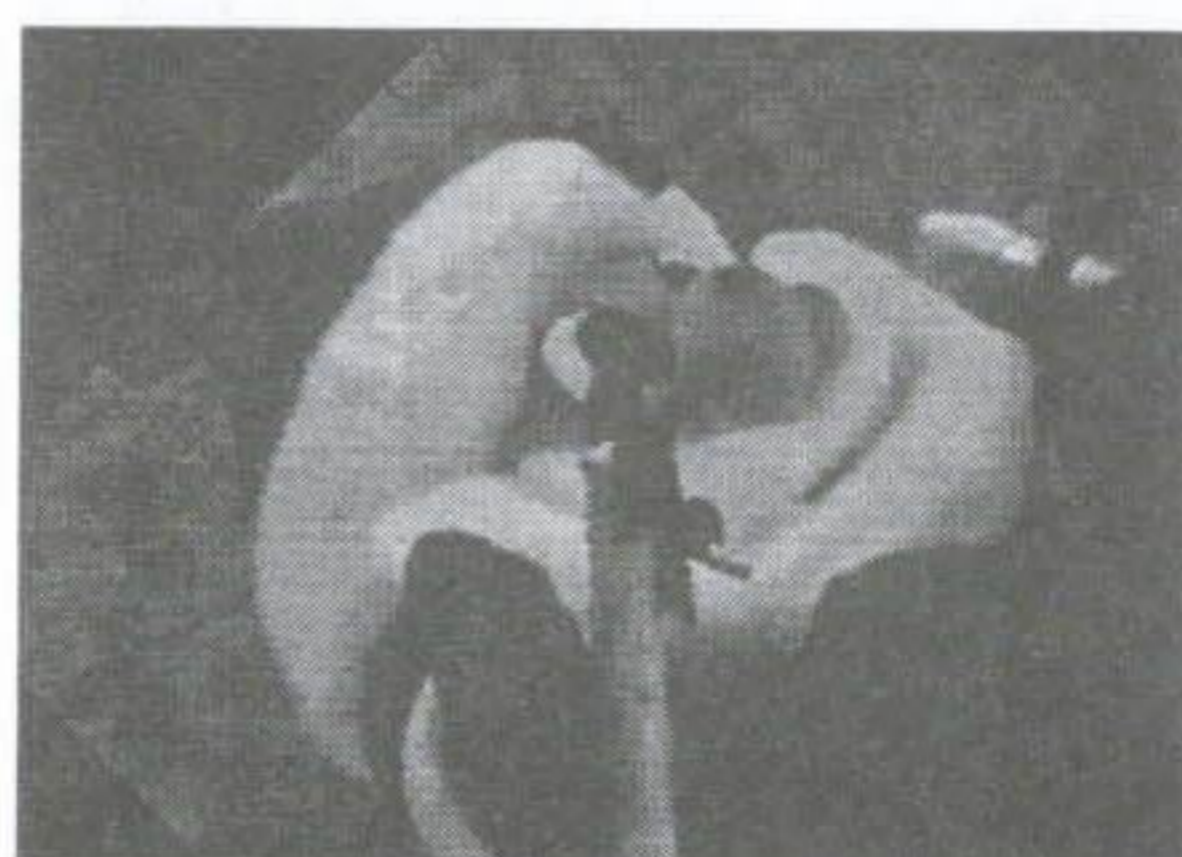


santa 2

그림 2.6 산타 영상.



santa 1의 보정 영상

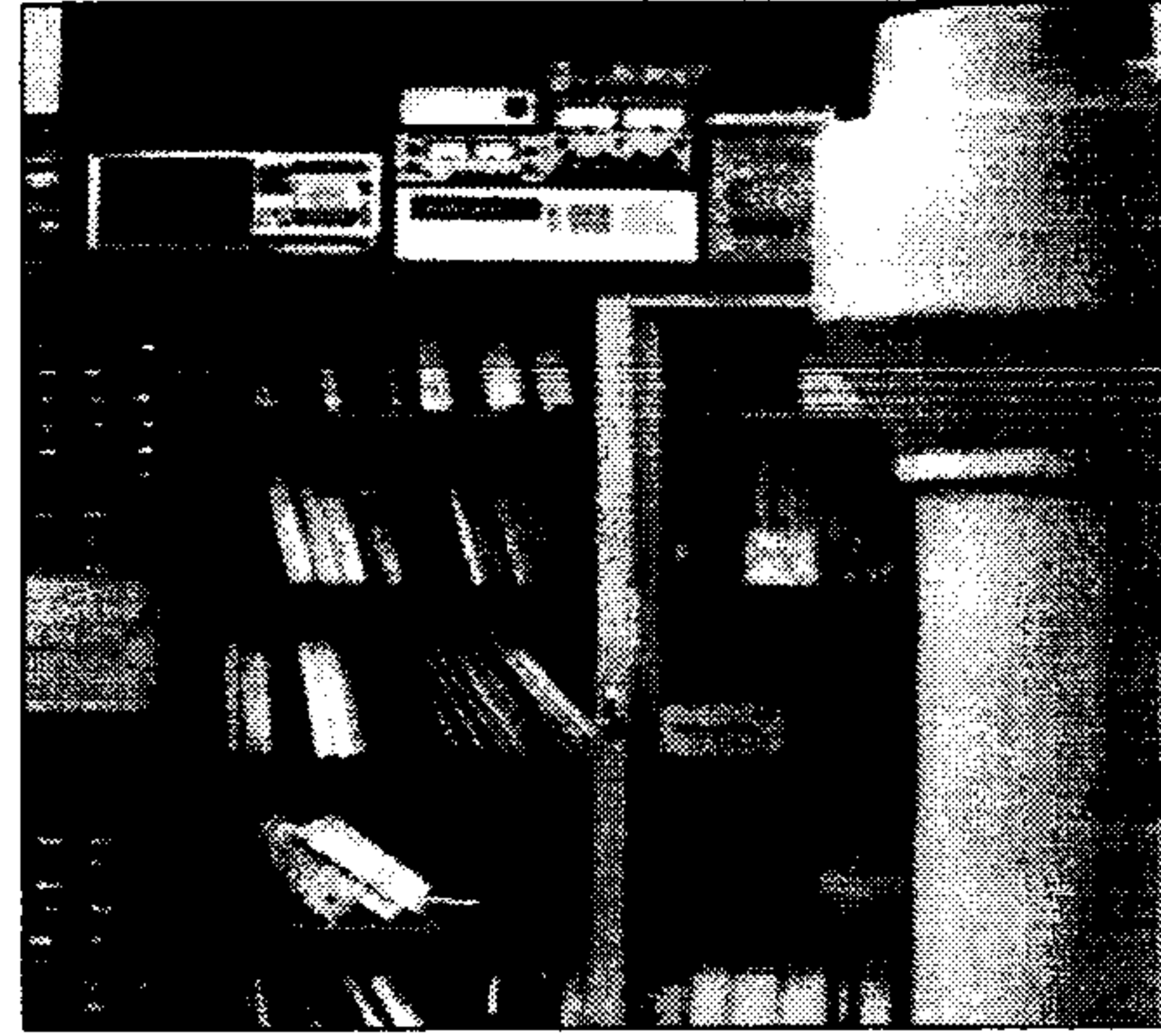


santa 2의 보정 영상

그림 2.7 산타 영상의 보정 영상.



Lab 1



Lab 2

그림 2.8 연구실 영상 : 카메라의 이동 성분만 포함



Lab 2의 보정 영상



Lab 1의 보정 영상

그림 2.9 연구실 영상의 보정 영상

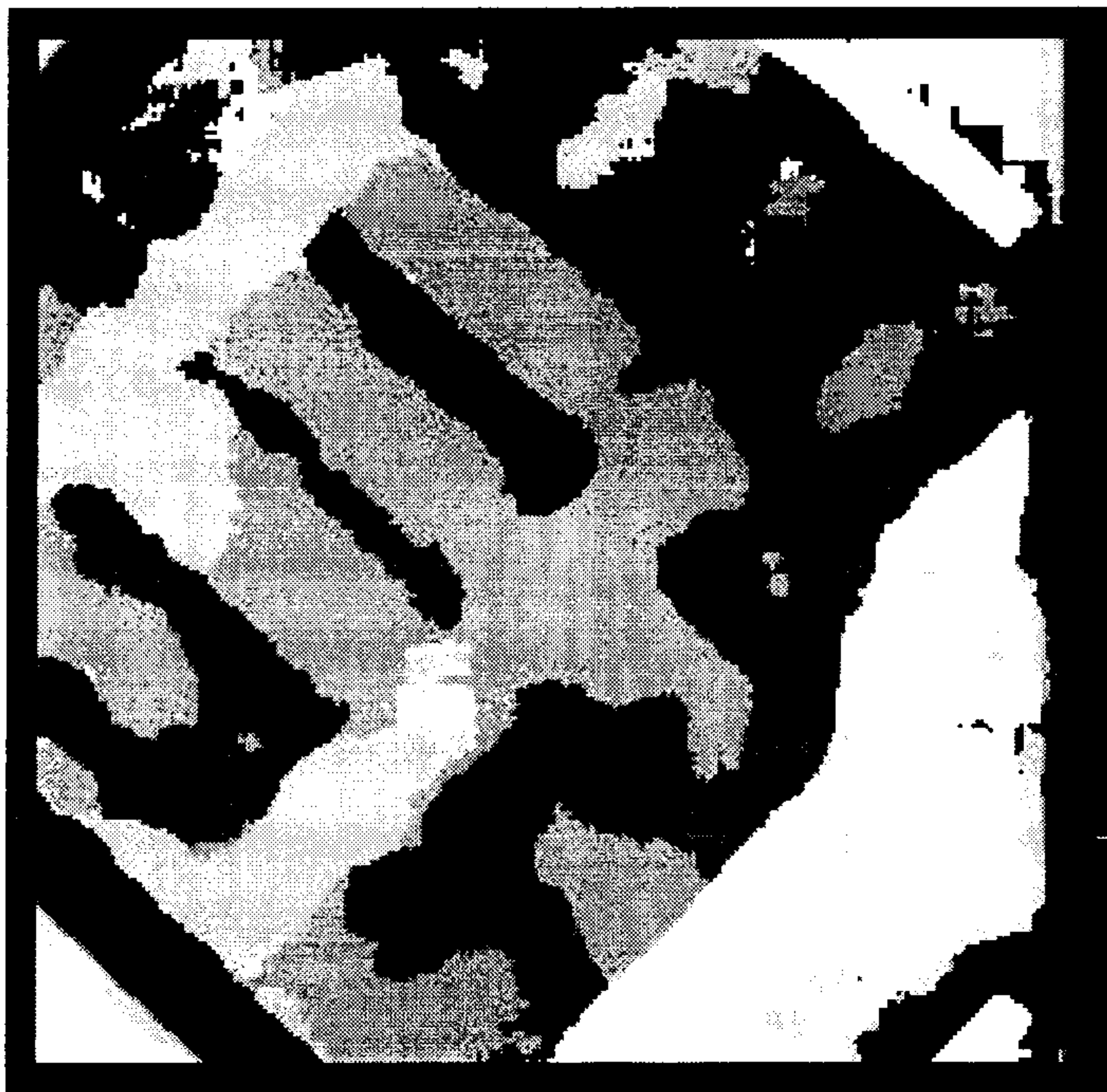
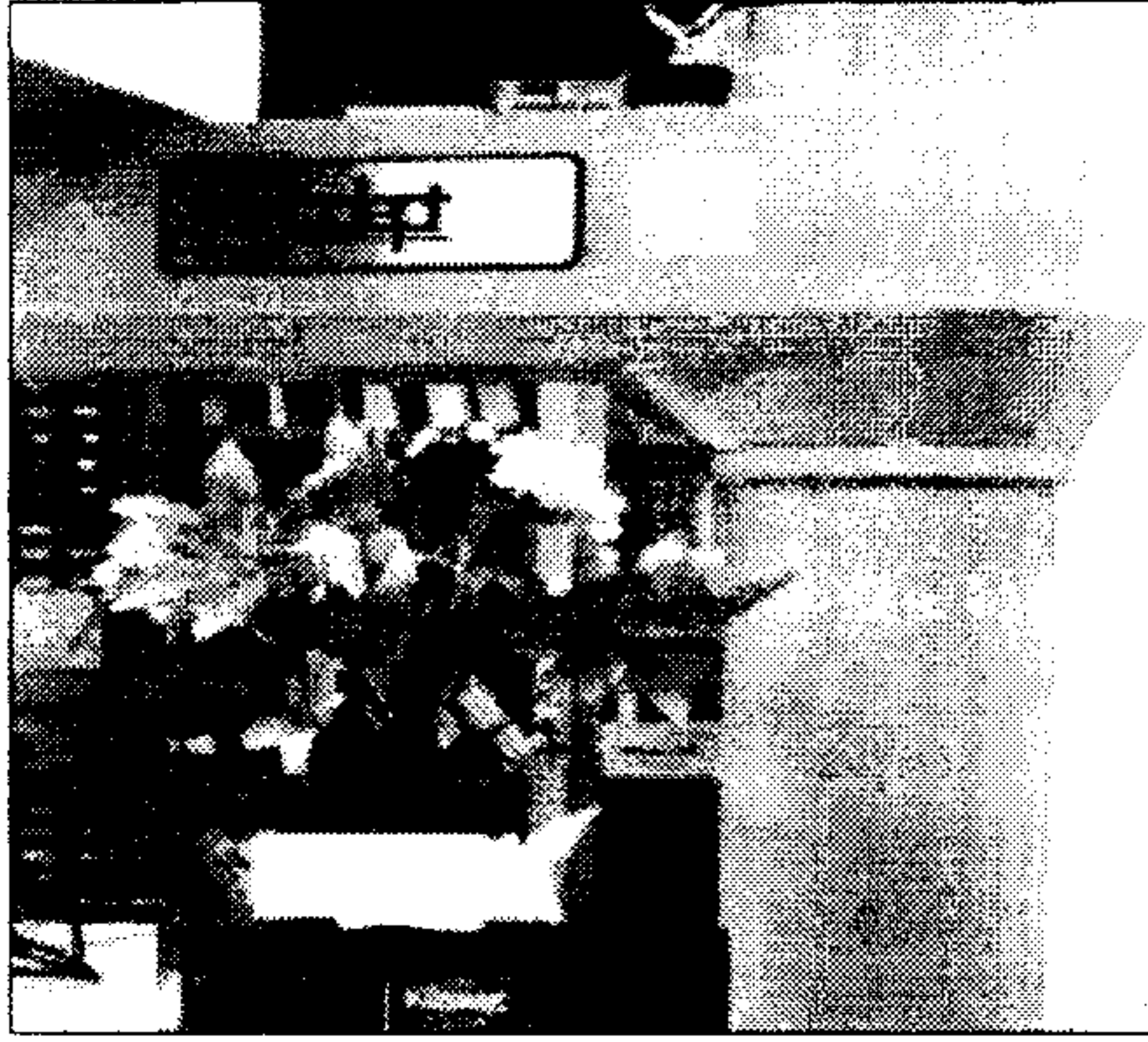
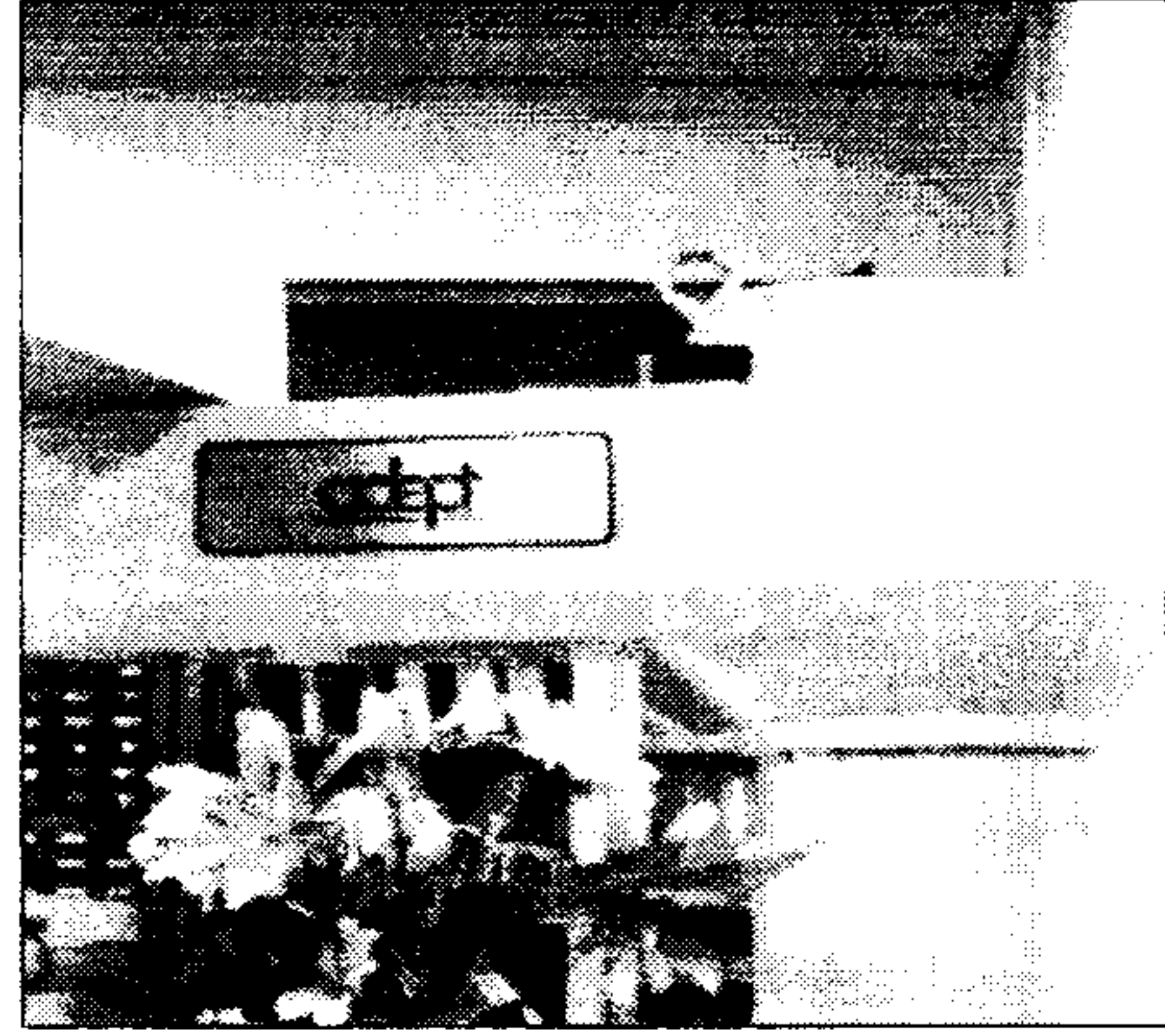


그림2.10 연구실 영상의 보정 변위도



Lab 3



Lab 4

그림 2.11 연구실 영상: 회전 성분과 이동 성분 포함



Lab 3의 보정 영상



Lab 4의 보정 영상

그림 2.12 회전 성분과 이동 성분 모두 보정한 영상

3절. 스테레오 비전 거리 측정

거리 정보를 나타내는 일반적인 식에서 z 는 mm 단위이고 d 는 화소(pixel) 단위임으로 단위 거리당 화소수를 나타내는 Δp 를 첨가하여 완전한 식을 만들면 다음과 같다.

$$z = \frac{bf\Delta p}{d} \quad (2.19)$$

여기서, $\Delta p = \text{pixels}/mm$ 이다. 변위와 거리와의 관계를 알아보기 위하여 일정 거리마다 물체를 위치시켜서 스테레오 영상을 획득하고 변위를 계산하였다. 카메라간 거리 $b=100mm$ 이고, 카메라의 초점거리 $f=16mm$ 이다. 식 (2.19)에서 변위와 거리와의 관계는 쌍곡선 모양의 반비례 관계임을 알 수 있다. 구해진 변위와 거리와의 관계를 가장 잘 만족시키는 쌍곡선의 방정식을 구한 결과 $\Delta p=132.2$ 를 얻을 수 있었다. 표 2.2에 카메라로부터 물체의 측정 거리와 이때 구해진 변위 그리고, 구해진 변위를 사용하여 식 (2.19)에 의해 계산된 거리를 나타내었다. 또한 측정 거리와 계산된 거리와의 오차(측정거리 - 계산된 거리)와 오차의 비율(오차/측정거리)도 나타내었다. 변위와 거리와의 관계를 그림 2.13에 나타내었다. 실선은 식 (2.19)의 쌍곡선의 방정식을 나타내며 동그라미는 측정거리와 변위를 나타낸다. 실제 식과 측정된 값들이 약간의 오차를 갖지만 매우 잘 일치함을 볼 수 있다. 오차와 오차의 비율을 각각 그림 2.14와 그림 2.15에 거리에 따라 나타내었다. 약 3,000mm까지는 $\pm 1\%$ 오차내외의 정확도까지 거리를 구할 수 있고 약 5,000mm까지는 $\pm 5\%$ 오차 범위내에서의 정확한 거리를 구할 수 있음을 알 수 있다. 카메라부터의 거리가 가까울수록 정확한 변위를 구할 수 있음을 알 수 있다. 이는 거리가 멀어질수록 변위의 변화량에 따른 거리의 변화량이 커지기 때문에 화소 단위의 변위로는 정확한 거리 정보를 얻을 수 없기 때문이다.

표 2.2 변위와 거리의 관계

z(측정거리)[mm]	d(변위)[pixel]	z'(계산된변위)	오차(error=z-z')	오차비율(error/z)
800	264	801.1	-1.1	-0.0014
850	250	846	4	0.0047
900	238	888.7	11.3	0.0126
950	222	952.7	-2.7	-0.0029
1000	212	997.6	-2.4	0.0023
1050	200	1057.5	-7.5	-0.0072
1100	192	1101.6	-1.6	-0.0014
1200	174	1215.5	-15.5	-0.0129
1250	166	1274.1	-24.1	-0.0193
1300	162	1305.6	-5.6	-0.0043
1350	156	1355.8	-5.8	-0.0043
1400	150	1410	-10	-0.0072
1450	146	1448.6	1.4	0.0009
1500	142	1489.4	10.6	0.0070
1550	136	1555.1	-5.1	-0.0033
1600	132	1602.3	-2.3	-0.0014
1650	128	1652.3	-2.3	-0.0014
1700	126	1678.6	21.4	0.0126
1750	120	1762.5	-12.5	-0.0072
1800	118	1792.4	7.6	0.0042
1850	114	1855.3	-5.3	-0.0029
1900	110	1922.7	-22.7	-0.0120
1950	108	1958.3	-8.3	-0.0043
2000	107	1976.6	23.4	0.0117
2050	104	2033.7	16.3	0.0080
2100	102	2073.5	26.5	0.0126
2150	98	2158.2	-8.2	-0.0038
2548	82	2579.3	-31.8	-0.0124
2648	78	2711.5	-63.9	-0.0242
2748	76	2782.9	-35.3	-0.0129
2848	74	2858.1	-10.5	-0.0037
2948	70	3021.4	-73.8	-0.0251
3048	70	3021.4	26.2	0.0086

3148	68	3110.3	37.3	0.0118
3248	66	3204.5	43.0	0.0132
3348	62	3411.3	-63.7	-0.0190
3448	62	3411.3	36.3	0.0105
3548	62	3411.3	136.3	0.0384
4400	46	4597.8	-197.8	-0.0450
4600	46	4597.8	2.2	0.0005
4800	42	5035.7	-235.7	-0.0491
5100	44	4806.8	293.2	0.0575
5300	40	5287.5	12.5	0.0023
5500	42	5035.7	464.3	0.0844

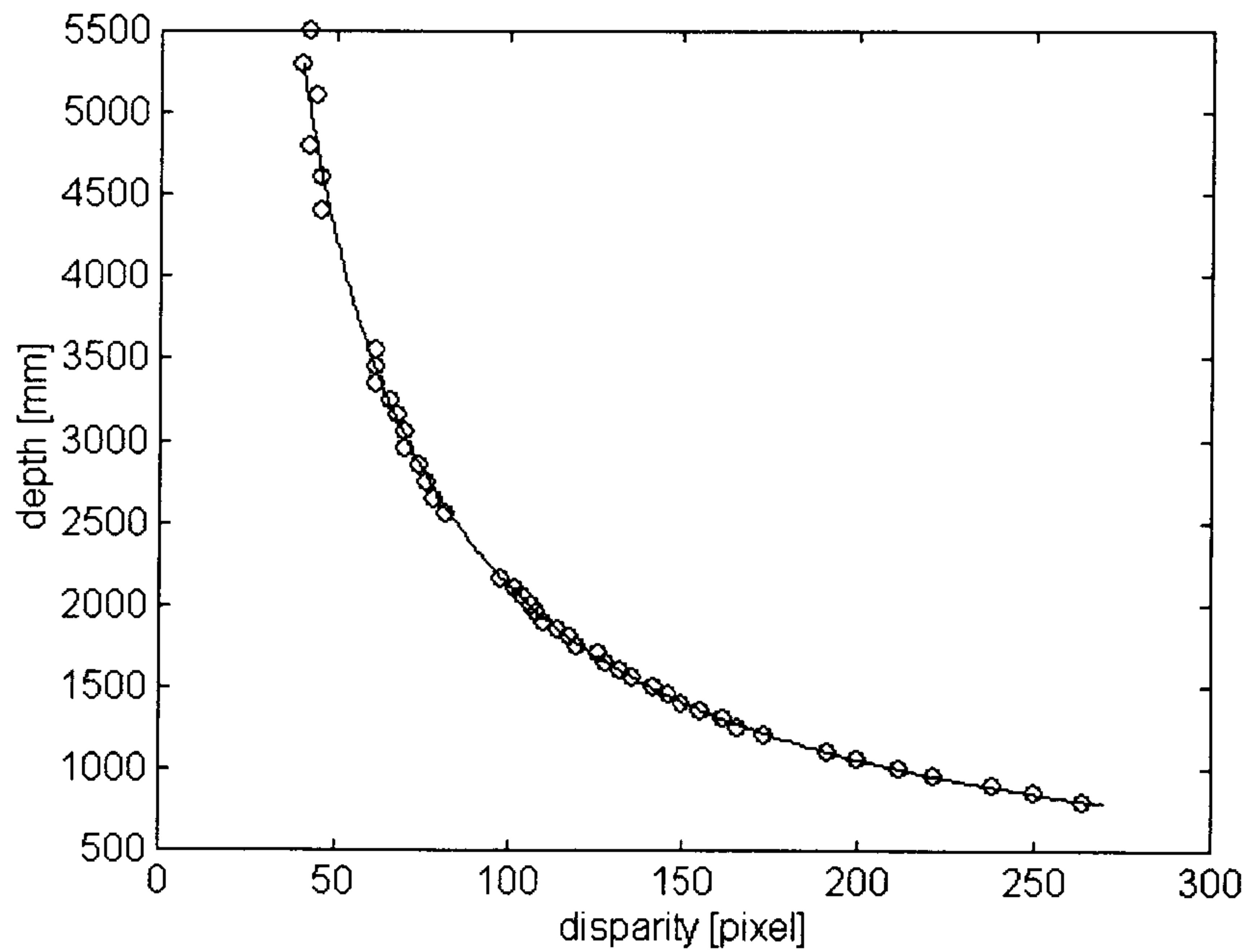


그림 2.13 변위와 거리와의 관계.

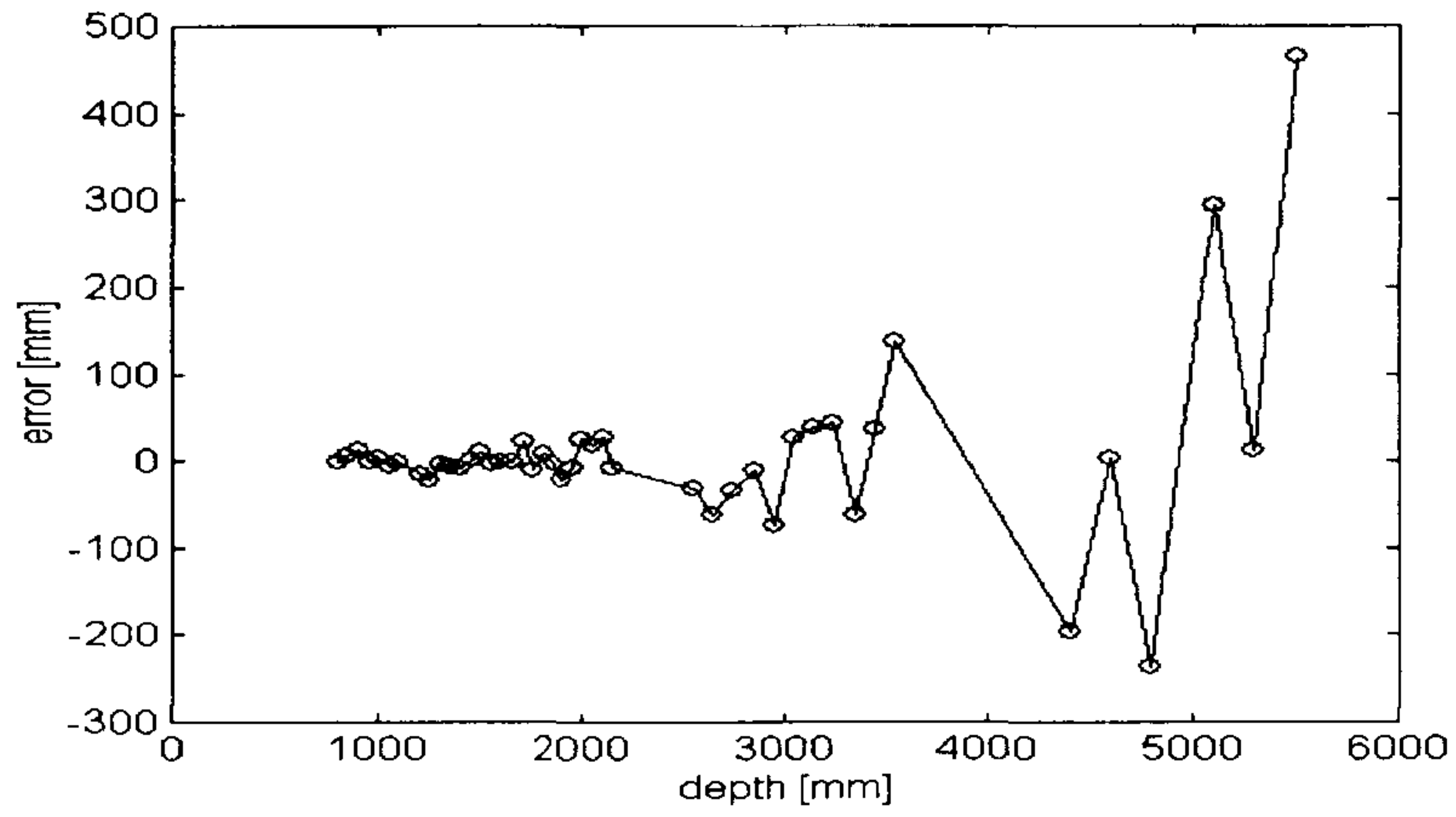


그림 2.14 거리에 따른 측정오차

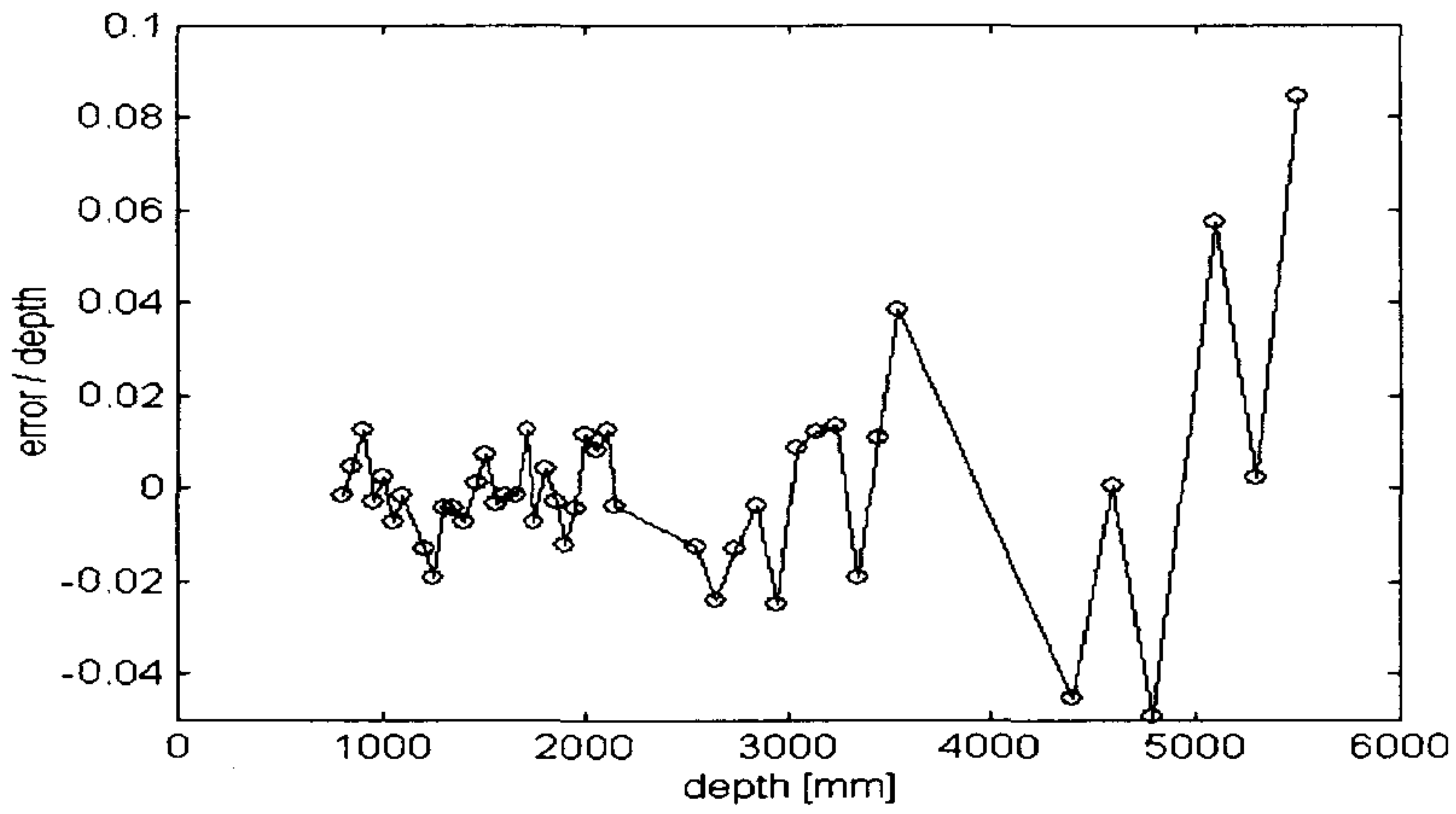


그림 2.15 거리에 따른 거리측정 오차 비율

제 3 장 고성능 스테레오 정합 알고리즘

스테레오 비전은 레이저나 초음파와 같은 능동 감지센서를 사용하지 않고 서로 다른 위치에서 얻은 2차원 영상만으로 3차원 거리 정보를 얻을 수 있는 효율적인 방법으로서, 서로 다른 위치의 카메라로부터 좌우 영상의 획득, 대응점 정합을 통한 변위정보의 추출, 삼각 측량법에 의한 거리정보 연산 등의 과정을 통해서 얻을 수 있다. 이러한 스테레오 비전을 사용한 거리정보 획득 과정 중 대응점을 알아내어 변위정보를 구하는 대응점 정합 과정이 높은 정밀도의 요구와 많은 연산 시간으로 인하여 실시간 구현에 장애가 되는 단계이다[1][2].

1절. MPC를 이용한 영역 기반 스테레오 정합 방법

본 절에서는 에피폴라 구속 조건을 만족시키면서 스테레오 영상을 얻을 수 있는 카메라 배치에 대한 설명과 스테레오 영상에서 대응점을 찾기 위해 영역 기반 방법에서 일반적으로 쓰이는 유사도 측정 방법들과 본 논문에서 사용하는 MPC 유사도 측정 방법에 대해 설명하고 MPC 방법의 우수함을 보인다.

가. 스테레오 카메라 배치

본 연구에서는 좌, 우 두 카메라를 배치함에 있어서 그림 3.1과 같이 x 축 상에 광축을 서로 평행하게 하고, x 좌표 값의 차(base line)는 b 가 되도록 배치하였다. 이와 같이 카메라를 배치함으로써 대응점을 찾는 탐색 범위가 동일 선상으로 제한될 수 있게 하

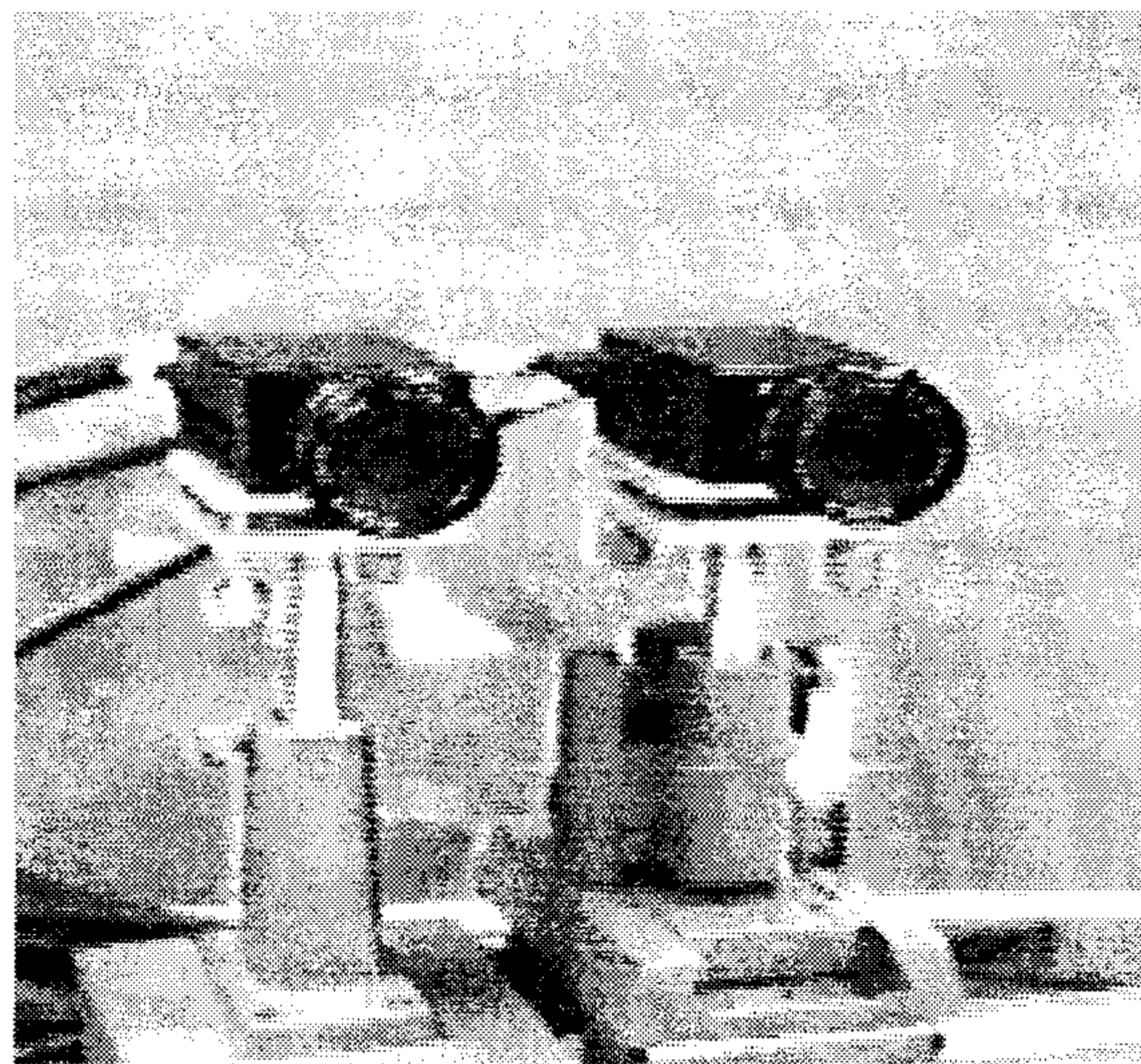
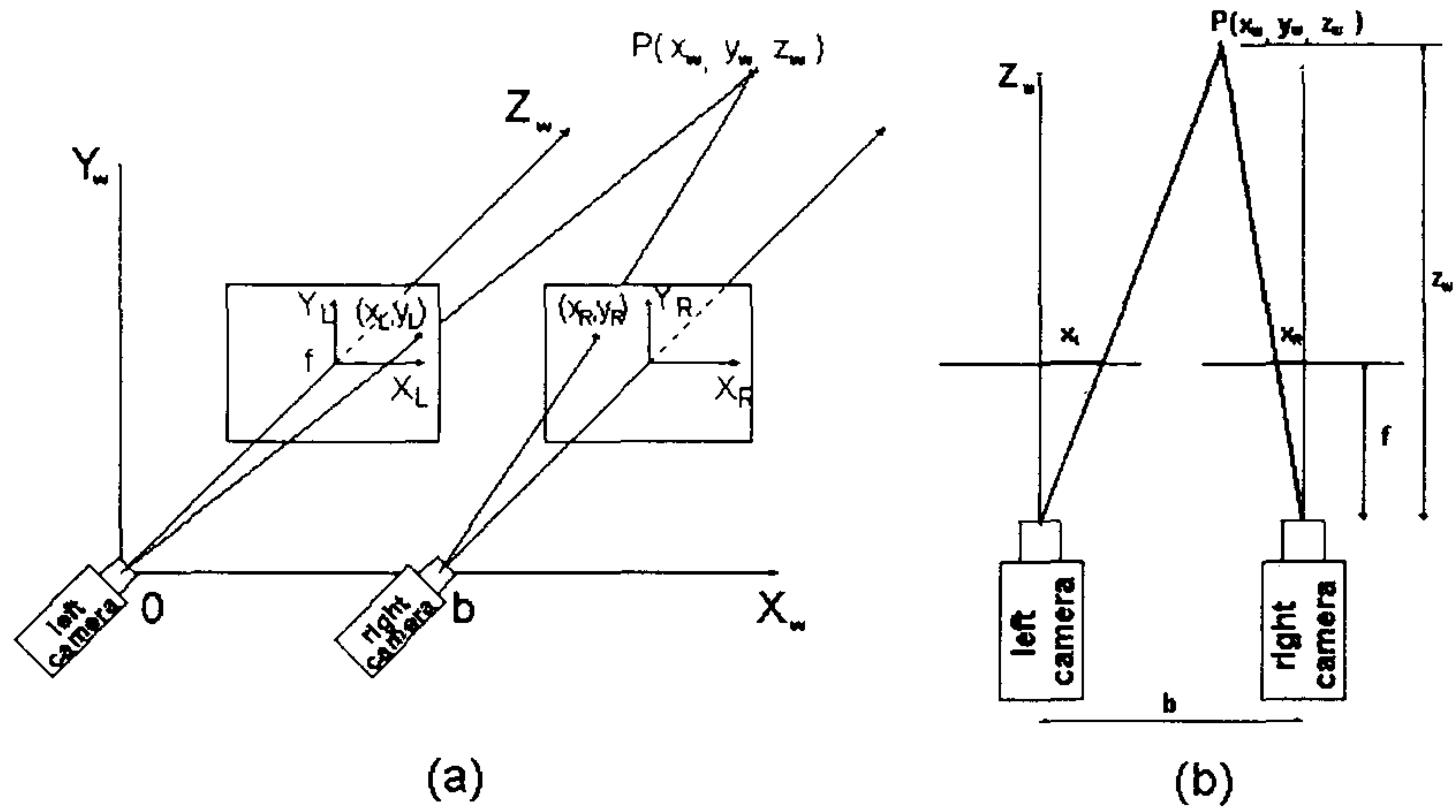
는 에피폴라 구속 조건을 만족하게 하였으며, 이로부터 3차원 거리 정보도 쉽게 알 수 있다[7]. 좌, 우 카메라의 초점 거리를 f 카메라 사이의 거리를 b , 3차원 공간상의 임의의 한 점 P 의 좌표를 (x_w, y_w, z_w) , P 가 좌우 카메라의 영상 면에 투영된 점의 좌표를 각각 (x_L, y_L) , (x_R, y_R) 이라 한다면, 에피폴라 구속조건에 따라 $y_L = y_R$ 이며, x 축 좌표는 삼각형의 비례관계에 의하여 다음과 같은 식으로 나타낼 수 있다.

$$\begin{aligned} x_L &= \frac{fx_w}{z_w} \\ x_R &= \frac{f(x_w - b)}{z_w} \end{aligned} \quad (3.1)$$

식 (3.1)을 사용하여 z_w 를 구하면 다음과 같다.

$$z_w = \frac{bf}{x_L - x_R} = \frac{bf}{d} \quad (3.2)$$

이 때 $d(=x_L - x_R)$ 를 변위라고 하며 깊이 정보에 반비례함을 알 수 있다. 즉, 좌 우 카메라에 투영된 대응점의 x 좌표의 차로부터 변위를 구하고 식 (3.2)에 따라 3차원 거리 정보를 얻을 수 있다. 변위가 클수록 카메라로부터 가깝고 작을수록 카메라로부터 먼 거리에 있다.



(c)

그림 3.1 스테레오 영상을 획득하기 위한 카메라 배치도. (a)평행 광축의 기하학적 배치. (b)위에서 본 배치. (c)실제 카메라 배치 사진.

나. 영역 기반 방법에서의 유사도 측정 방법

영역 기반 정합 방법은 기준 화면의 모든 화소에 대해 대응점을 구함으로 연산시간이 많이 걸리고, 한쪽 영상에서는 보이지만 다른 한쪽 영상에서는 보이지 않는 폐색(Occlusion) 영역이나 화소간의 밝기 변화가 미미한 영역에 대해서는 올바른 결과가 나오지 않지만, 계산이 단순하고 조밀한 변위를 구할 수 있다는 장점 때문에 스테레오 정합 방법으로 많이 쓰이고 있다[2][3][9]. 대응점 정합은 기준 화면의 찾고자 하는 화소를 중심으로 일정한 크기의 창 내의 화소 값들과 탐색 화면에서 대응점이라고 간주되어지는 후보 위치의 화소를 중심으로 같은 크기의 창 내의 화소들간의 유사도를 계산하여 가장 유사도가 큰 값을 갖는 화소를 정합된 대응점이라고 한다. 이때, 모든 정합 후보 위치의 범위는 탐색 범위에 의해 결정된다. 유사도를 계산하는 방법은 화소간의 밝기 차를 보는 SSD(Sum of Squared Distance), SAD(Sum of Absolute Distance)방법과 상관 관계를 보는 NCC(Normalized Correlation Coefficient)방법 등이 있다. 우 영상의 (x, y) 위치의 화소와 x 축 상으로 d 만큼의 거리에 있는 좌 영상의 $(x+d, y)$ 위치의 화소간의 유사도는 각각 다음과 같이 정의된다.

$$SSD(x, y, d) = \sum_W [R_{x,y} - L_{x+d,y}]^2 \quad (3.3)$$

$$SAD(x, y, d) = \sum_W |R_{x,y} - L_{x+d,y}| \quad (3.4)$$

$$NCC(x, y, d) = \frac{\sum_W [R_{x,y} - \overline{R_{x,y}}] \times [L_{x+d,y} - \overline{L_{x+d,y}}]}{\sqrt{\sum_W [R_{x,y} - \overline{R_{x,y}}]^2} \times \sqrt{\sum_W [L_{x+d,y} - \overline{L_{x+d,y}}]^2}} \quad (3.5)$$

이때, $R_{x,y}$, $L_{x,y}$ 는 각각 우 영상 및 좌 영상의 (x, y) 위치에서 화소의 밝기 값을 나타내고, W 는 우 영상의 위치 (x, y) 와 이에 대응하는 좌 영상의 위치 $(x+d, y)$ 를 중심

으로 하는 일정한 크기의 정합 창을 나타내며, $\overline{R_{x,y}}$ 는 우 영상에서 (x, y) 를 중심으로 하는 일정한 크기의 창 내의 화소들의 평균값을 나타낸다. SSD와 SAD는 화소들간의 밝기 차만을 보는 단순한 정합방법으로 유사도가 클수록 작은 결과 값을 갖는다. NCC는 화소들간의 상관 관계를 보는 방법으로 상관 관계가 없을 때 NCC 는 0 이며 정방향의 상관 관계가 있을 때 최대값 1, 역 방향의 상관 관계가 있을 때 최소값 -1을 갖는다. 상관 관계를 사용하는 NCC는 조명과 카메라 고유의 특징이 좌, 우 영상에 달리 작용함으로 인한 잡음 등에 SSD나 SAD보다 강인한 특징을 보인다[10].

SSD, SAD, NCC 등의 유사도 측정 방법은 정합 창 내에서 한 화소가 유사도에 미치는 영향이 화소의 밝기 값에 따라 다르기 때문에 화소간의 밝기 변화가 큰 영역이 작은 영역으로 변위의 경계선이 연장되는 Boundary Overreach 현상[4]이 생기고, 이 때문에 경계 부분에서 잘못된 결과를 나타낸다. 유사도를 측정하는 방법으로 화소간의 밝기 값을 직접 이용하지 않고 정합 창 내에서 밝기가 비슷한 화소의 개수를 계산하는 MPC(Matching Pixel Count)를 사용한다. MPC 정합 방식은 영상의 움직임 정보를 구하기 위해 제안된 정합 방법으로[11] 거리에 의한 변위정보 추출에 적용하는 경우, 정합창 내의 한 화소가 유사도에 미치는 영향이 밝기 값에 관계없이 동일하게 함으로써 경계 부분에서도 좋은 결과를 얻을 수 있다. 우좌 영상의 두 점 (x, y) , $(x+d, y)$ 를 중심으로 하는 창 내에서 밝기가 비슷한 화소의 개수인 $MPC(x, y, d)$ 값은 다음과 같이 나타낼 수 있다.

$$MPC(x, y, d) = \sum_{\mathbb{W}} P(x, y, d)$$

$$where, P(x, y, d) = \begin{cases} 1, & \text{if } |R_{x,y} - L_{x+d,y}| \leq Th \\ 0, & \text{otherwise} \end{cases} \quad (3.6)$$

$P(x, y, d)$ 는 우좌 영상의 두점 (x, y) , $(x+d, y)$ 에서의 화소 밝기가 비슷하면 1, 비슷하지 않으면 0의 값을 갖는다. Th 는 미리 정의된 문턱값으로, 1을 사용하였다. (x, y) 위

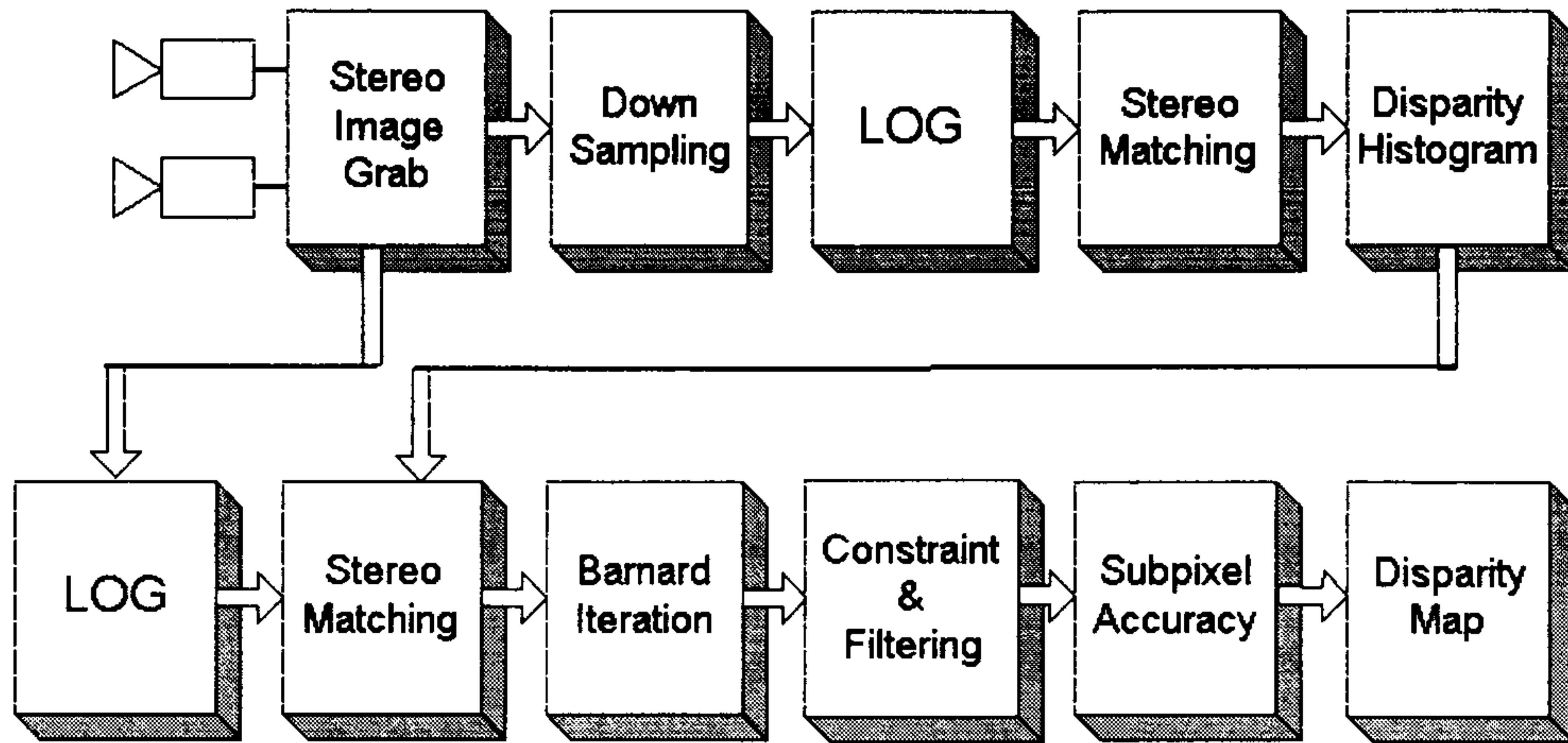
치에서의 변위 $DM(x, y)$ 는 비슷한 화소의 개수가 최대가 될 때의 d 값으로 다음과 같이 나타낼 수 있다.

$$DM(x, y) = \arg \max_d MPC(x, y, d) \quad (3.7)$$

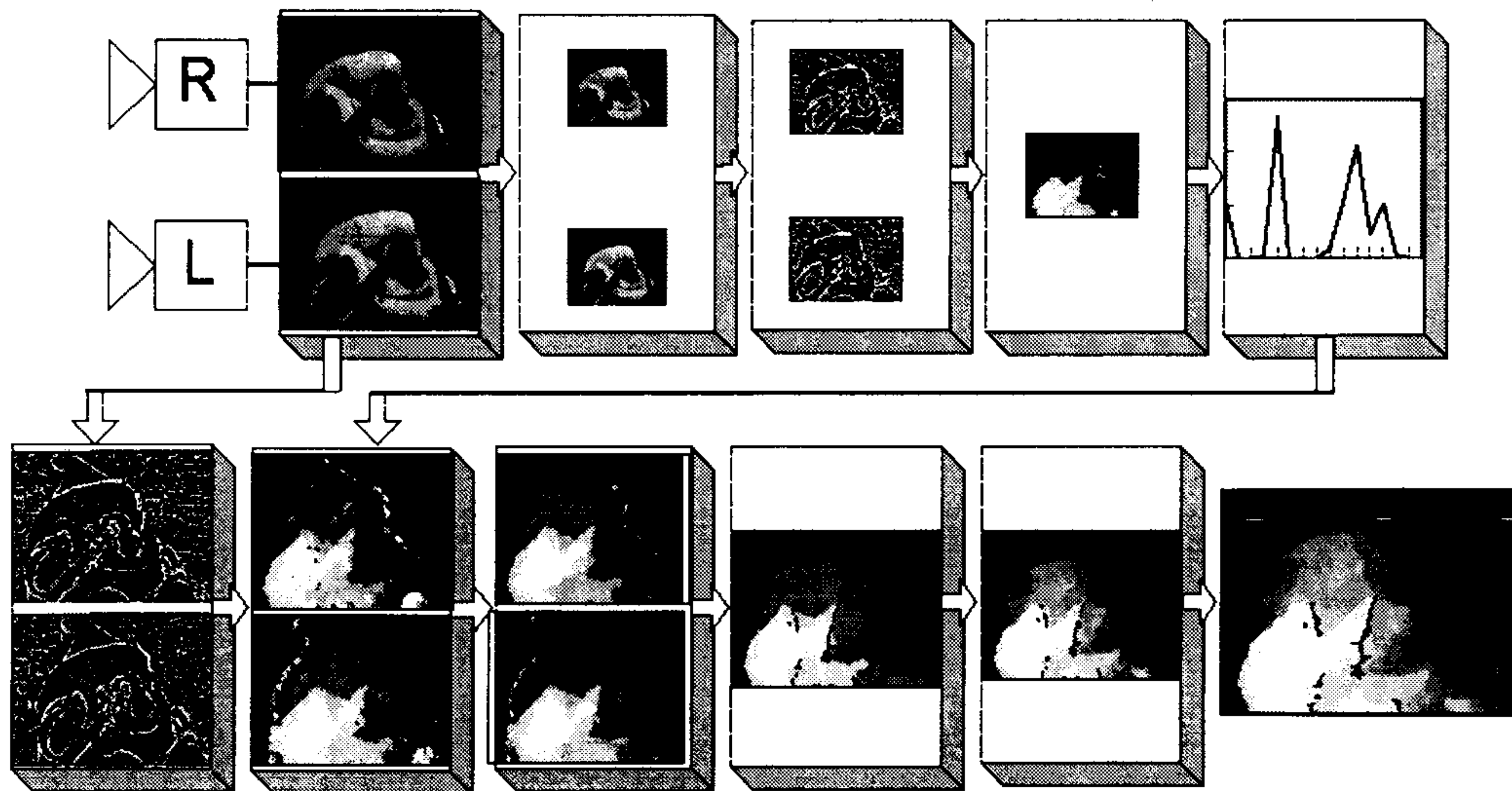
2절. 고성능 MPC 구현 알고리즘

본 절에서는 낮은 해상도에서 계산된 변위 히스토그램을 이용하여 원 해상도에서 탐색 범위를 제한함으로써 경계 부분에서도 좋은 결과를 보이면서 연산 시간도 단축시킬 수 있는 2단계 변위정보 추출 알고리즘과 이의 고속 구현이 가능하도록 중복 연산이 제거된 MPC 정합 방법에 대해 설명한다. 전체적인 알고리즘의 흐름도는 그림 3.2의 (a)와 같고 Santa영상에 대한 실제 예를 그림 3.2의 (b)에 나타내었다. 스테레오 영상 입력 장치를 통하여 얻은 원 해상도의 영상과 이의 다운 샘플링을 통하여 저 해상도의 스테레오 영상을 만든다. 이때 aliasing을 억제하기 위해 다운 샘플링을 하기 전에 평균 필터링을 원 영상에 대해 수행하였다. 이 두 영상에 대해 전처리 과정으로 LOG(Laplacian of Gaussian) 연산을 수행함으로써 영상의 특징들을 강조하고 서로 다른 조명과 카메라 이득으로 인한 영향을 제거해 준다. 저 해상도의 스테레오 영상에 중복 계산이 없는 고속 MPC 정합 방법을 사용하여 저 해상도 변위를 얻는다. 낮은 해상도에서 계산된 변위 영상으로부터 변위의 빈도를 나타내는 변위 히스토그램을 구하면 변위와 해당 변위의 빈도로부터 배경과 물체에 해당하는 변위를 알 수 있다. 이를 이용하여 원 해상도의 탐색 범위와 탐색 영역이 제한된 상태에서 고속 MPC 정합 방법을 원 해상도의 영상에 적용함으로써 초기 변위를 고속으로 구할 수 있다. 구해진 초기 변위에 반복 후원, 좌 우 정합 결과 일치 제한 조건(Two-Views Constraint), 폐색 영역 탐색, 시그마 필터링(Sigma Filtering), 소수점 화소 단위 정확도 등의 후처리 과정을

통해 최종 변위 영상을 얻을 수 있다.



(a)



(b)

그림 3.2 제안하는 알고리즘의 흐름도. (a)흐름도. (b)Santa 영상에 대한 실제 예.

가. 변위 히스토그램을 이용한 2단계 변위 추출 알고리즘

변위 영상에서 특정 변위값을 가지는 화소의 개수를 변위 히스토그램이라 하며, 일반적인 스테레오 영상에서 변위정보는 전체 탐색 범위에 고르게 분포되어 있지 않고 배경과 물체의 변위에 해당하는 값만을 갖는다. 이러한 관점에 착안하여 낮은 해상도에서 연산된 변위 히스토그램을 이용하여 원 해상도에서 물체 단위로 전체 탐색 범위를 제한함으로써 경계 부분에서도 좋은 결과를 보이고 계산 시간도 단축시킬 수 있다. 변위 히스토그램을 이용한 2단계 변위 추출 알고리즘의 개념을 그림 3.3의 Santa 스테레오 영상으로 설명하면 다음과 같다. 그림 3.3의 (a)와 (b)는 4:1의 다운 샘플링을 통해 얻은 160*120 크기의 저 해상도 좌우 스테레오 영상이며, (c)는 이들로부터 구한 변위 영상을 나타낸다. 그림 3.3의 (d)와 (e)는 320*240 크기의 원 해상도 좌우 스테레오 영상이며, (f)는 이들로부터 구한 변위 영상이다. 그림 3.3의 (g)와 (h)는 각각 저 해상도 및 원 해상도 변위 영상으로부터 구한 변위 히스토그램을 나타낸다. 변위 히스토그램의 x 좌표는 변위를 나타내고, y 좌표는 해당 변위의 빈도 수를 나타낸다. 작은 변위에서 나타나는 하나의 극점은 배경에 해당하는 변위값들이고 비교적 많은 개수이면서 서로 모여있는 변위값들은 물체에 해당하는 변위임을 알 수 있다. 즉, 그림 3.3의 (g)에서 배경에 해당하는 변위는 4, 물체에 해당하는 변위는 8~12임을 알 수 있고, 이는 그림 3.3의 (h)에 나타나는 원 해상도에 있어서 배경의 변위값인 7과, 물체의 변위값 범위인 17~25를 잘 반영한다. 이와 같이 원 해상도의 변위 영상을 구할 때 낮은 해상도의 변위 히스토그램을 이용하여 배경과 물체를 분리해 낼 수 있고 배경과 물체의 변위값 범위도 알 수 있다. 이를 적용하여 원 해상도에서는 관심 있는 물체에 해당하는 영역에 대해서만 정합을 수행함으로써 탐색 영역을 제한 할 수 있고, 탐색 범위도 전체 탐색을

하지 않고 배경과 물체에 해당하는 변위로만 제한할 수 있다. 이 경우, 탐색 범위가 제한됨으로써 전체 탐색보다 오정합에 빠질 확률도 줄어들고, 전체 계산량도 다음에 설명할 중복 계산을 제거하는 방법을 이용하여 $(O_x \times O_y) \times (Sr')$ 로 줄어든다. 여기서 $(O_x \times O_y)$ 는 제한된 탐색 영역을 나타내며, Sr' 는 제한된 탐색 범위를 나타낸다. 예를 들어 Santa 스테레오 영상의 경우 1/4 해상도의 영상으로부터 구한 변위 히스토그램으로 얻어진 배경과 물체의 변위값이 각각 4 및 8~12 인 경우, 물체에 해당하는 탐색 범위를 $15(2 \times 8 - 1)$ 에서 $25(2 \times 12 + 1)$ 로 제한할 수 있다. 고속 연산이 요구되는 실시간 구현을 목적으로 하는 경우 1단계 처리 영상의 해상도를 1/16 혹은 1/64로 사용하여 처리 속도와 정확성을 상호 보완하여 사용할 수 있으며 1단계에서 구한 결과를 2단계의 원 해상도에 이용함으로써 빠른 시간에 초기 변위를 얻을 수 있다. 그러나, 변위값이 전체 탐색 범위에 모두 나타나는 경우에는 처리 속도의 향상이 없고 1단계 처리 영상의 해상도를 너무 낮게 할 경우 빈도가 낮은 변위는 사라질 수 있다는 단점이 있다. 기존의 계층적 스테레오 정합 방법이 낮은 해상도의 정합 결과를 원 해상도에 이용할 때 화소 단위로 이용하여 원 해상도에서 각 화소의 탐색 범위는 낮은 해상도의 해당 위치에서의 변위값을 참조하여 정하였으나[6], 본 알고리즘은 낮은 해상도의 정합 결과로 얻은 변위 히스토그램을 이용하여 원 해상도의 제한된 탐색 영역의 모든 화소가 동일한 탐색 범위를 갖되 탐색 범위를 제한시킴으로써 다음 절에서 설명할 중복 계산을 제거하는 MPC 정합 방법을 사용할 수 있고, 또한 화소 단위로 이용함으로써 생기는 경계부근에서의 오정합을 피할 수 있다.

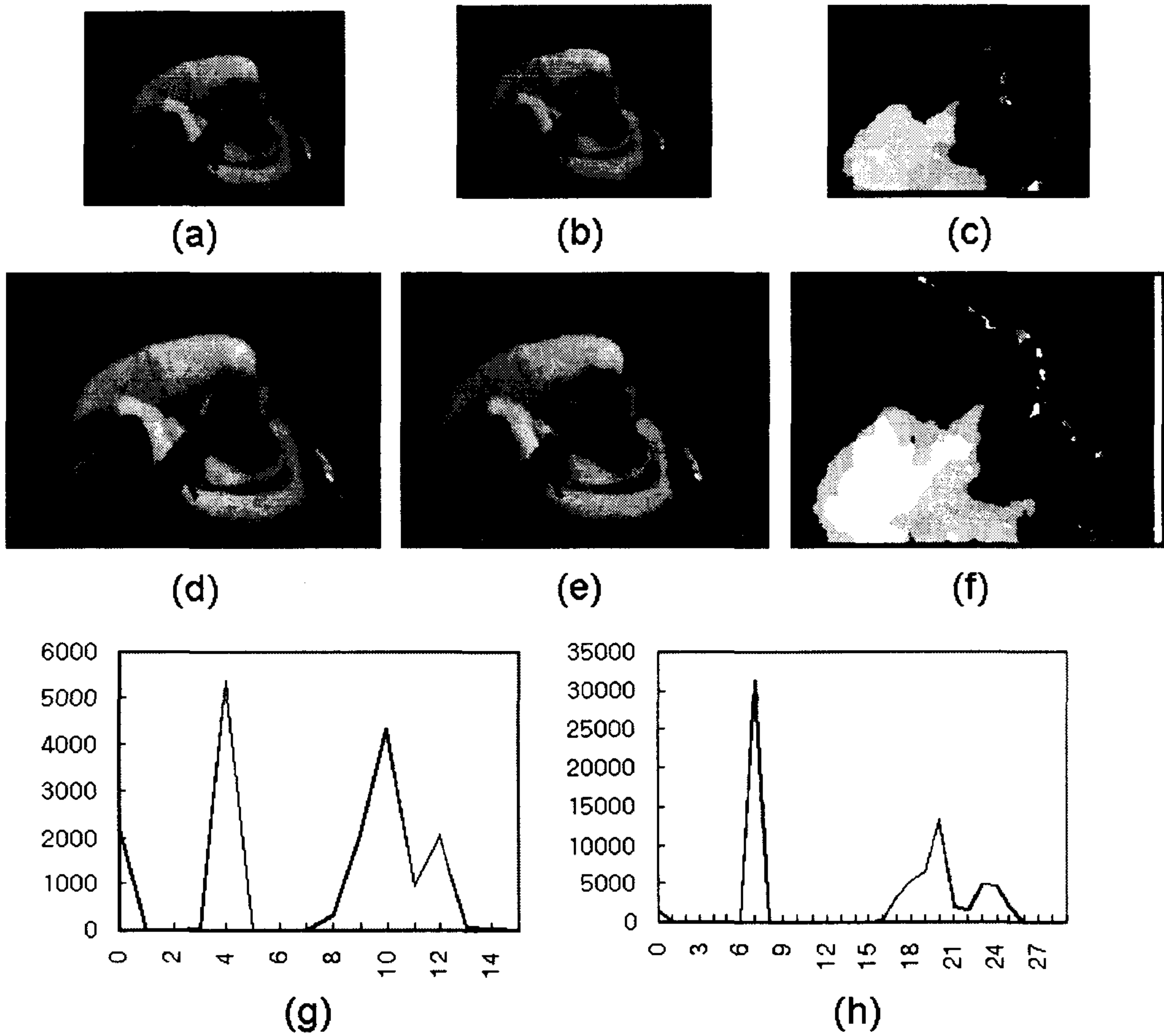


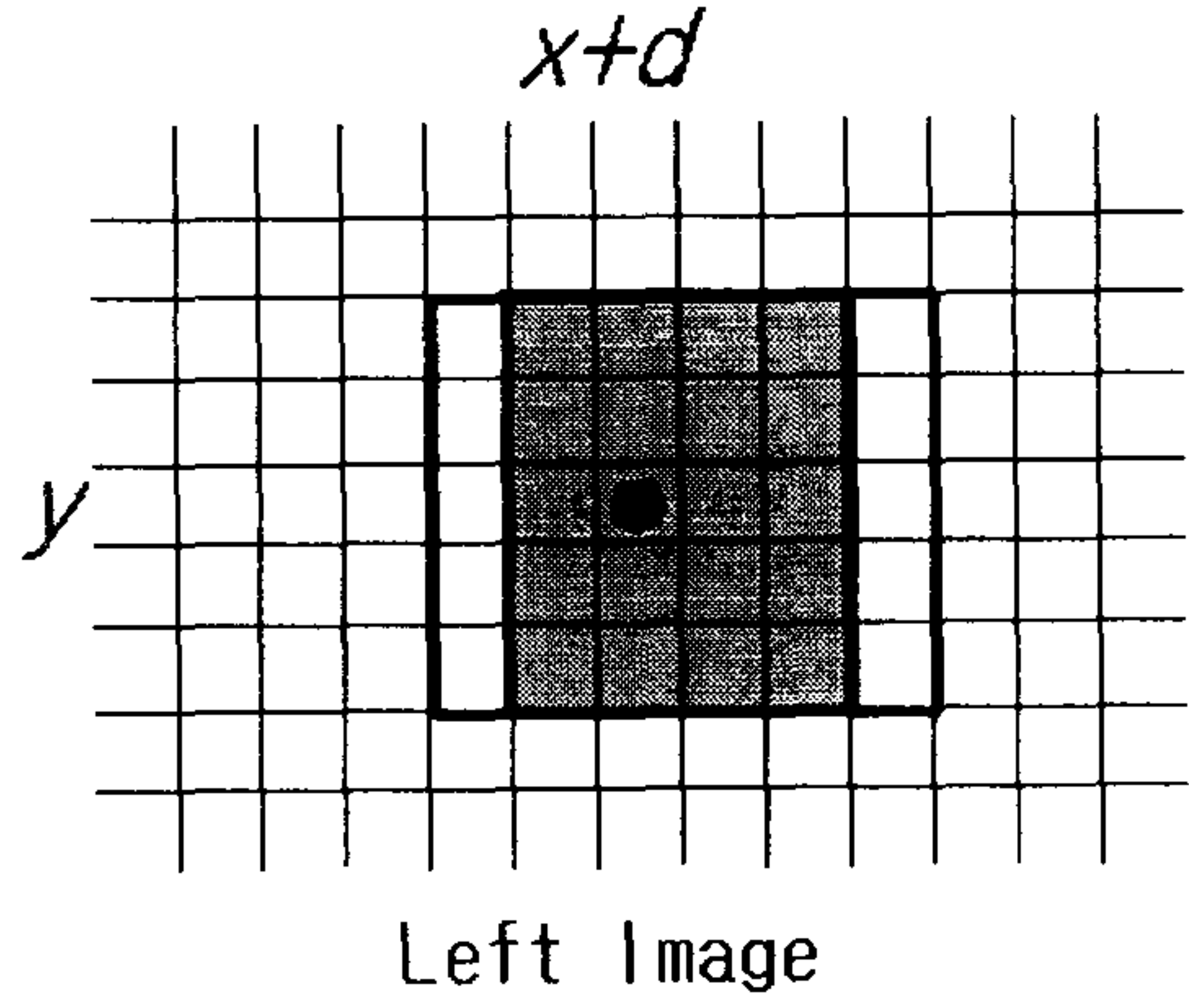
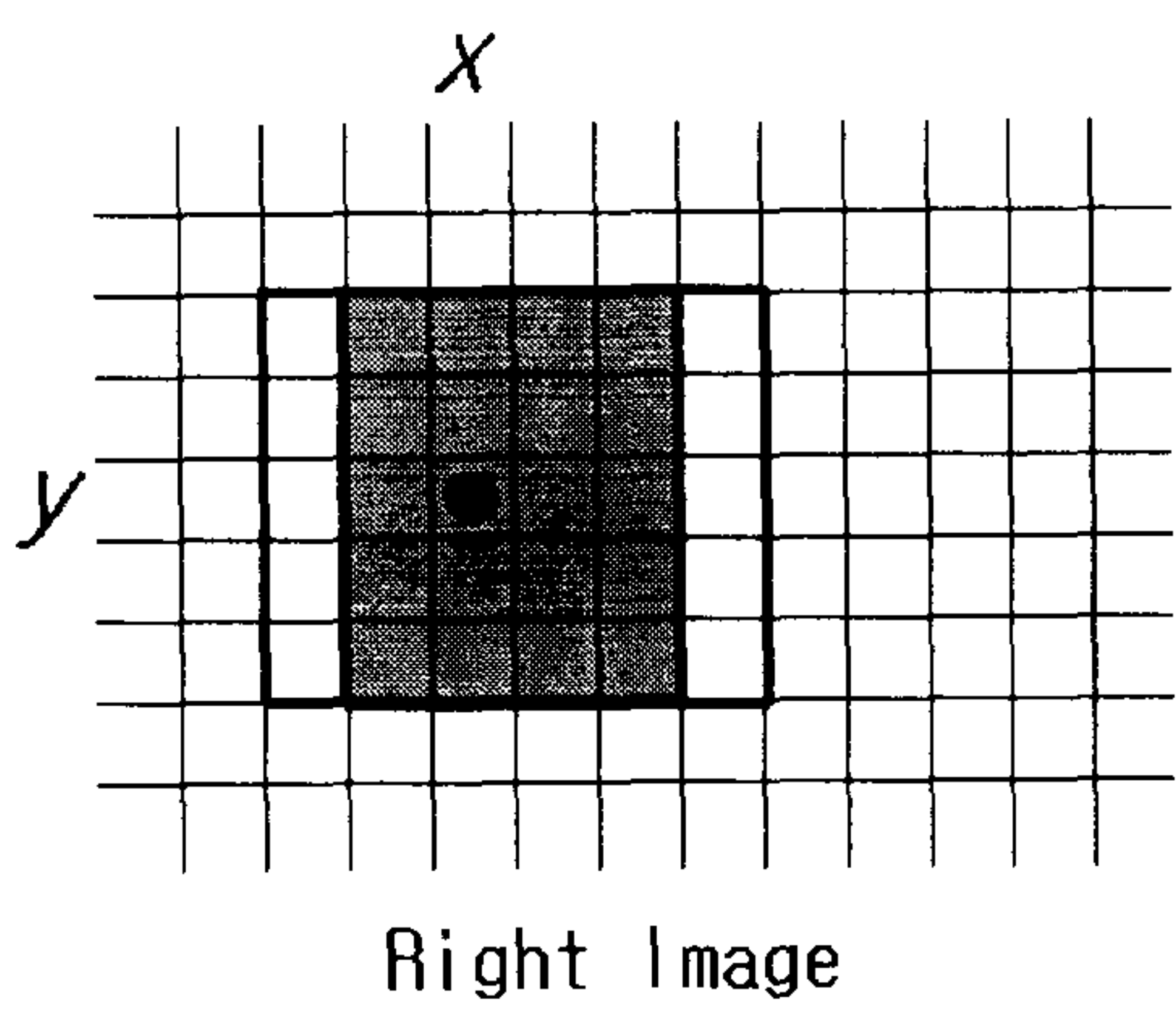
그림 3.3 Santa 스테레오 영상에 대한 변위 히스토그램. 저 해상도(160*120)의 좌(a) 우(b) 스테레오 영상과 변위 영상(c), 원 해상도(320*240)의 좌(d) 우(e) 스테레오 영상과 변위 영상(f), 저 해상도의 변위 히스토그램(g)과 고해상도의 변위 히스토그램(h)

나. 중복 계산이 제거된 MPC 정합 방법

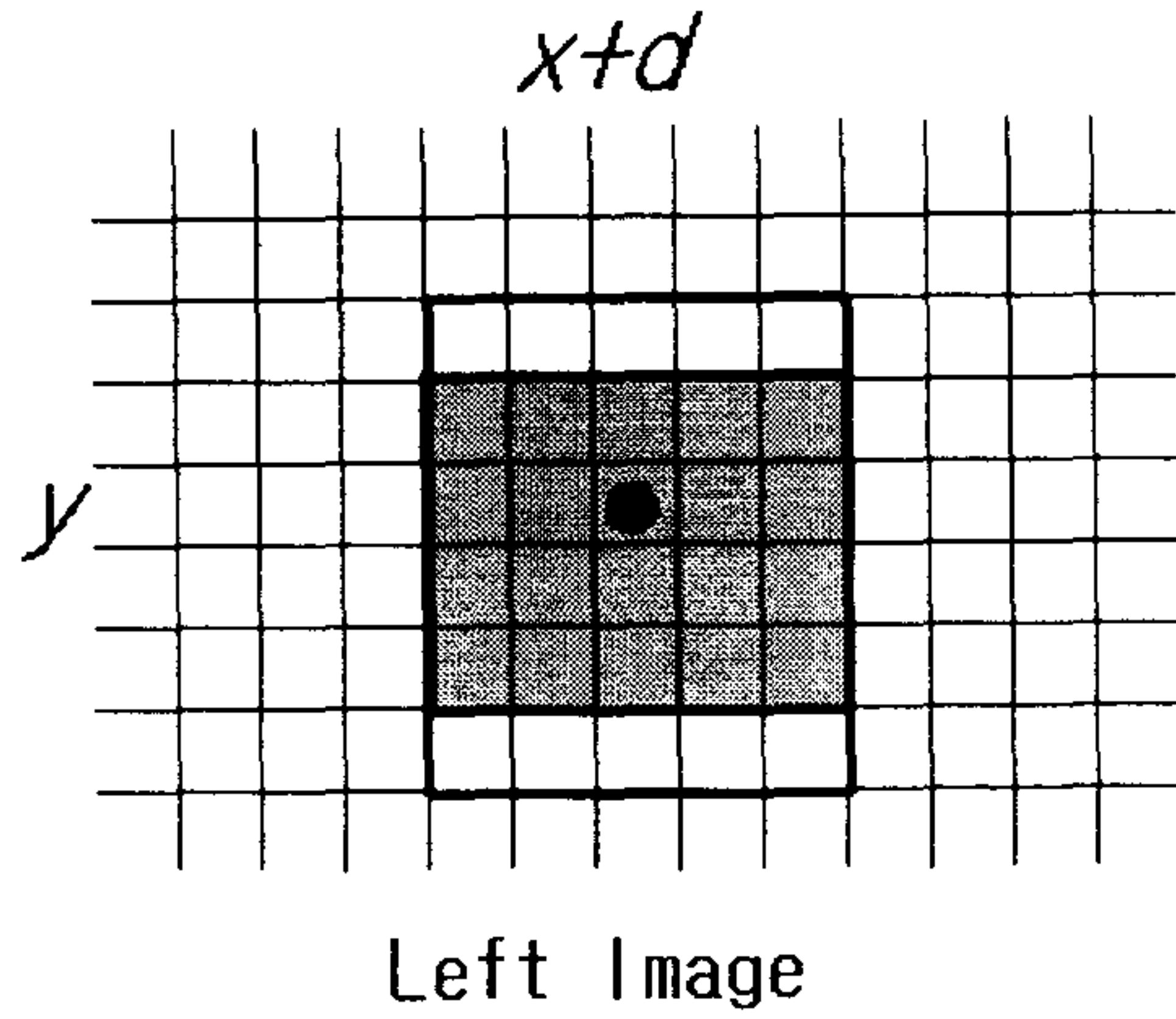
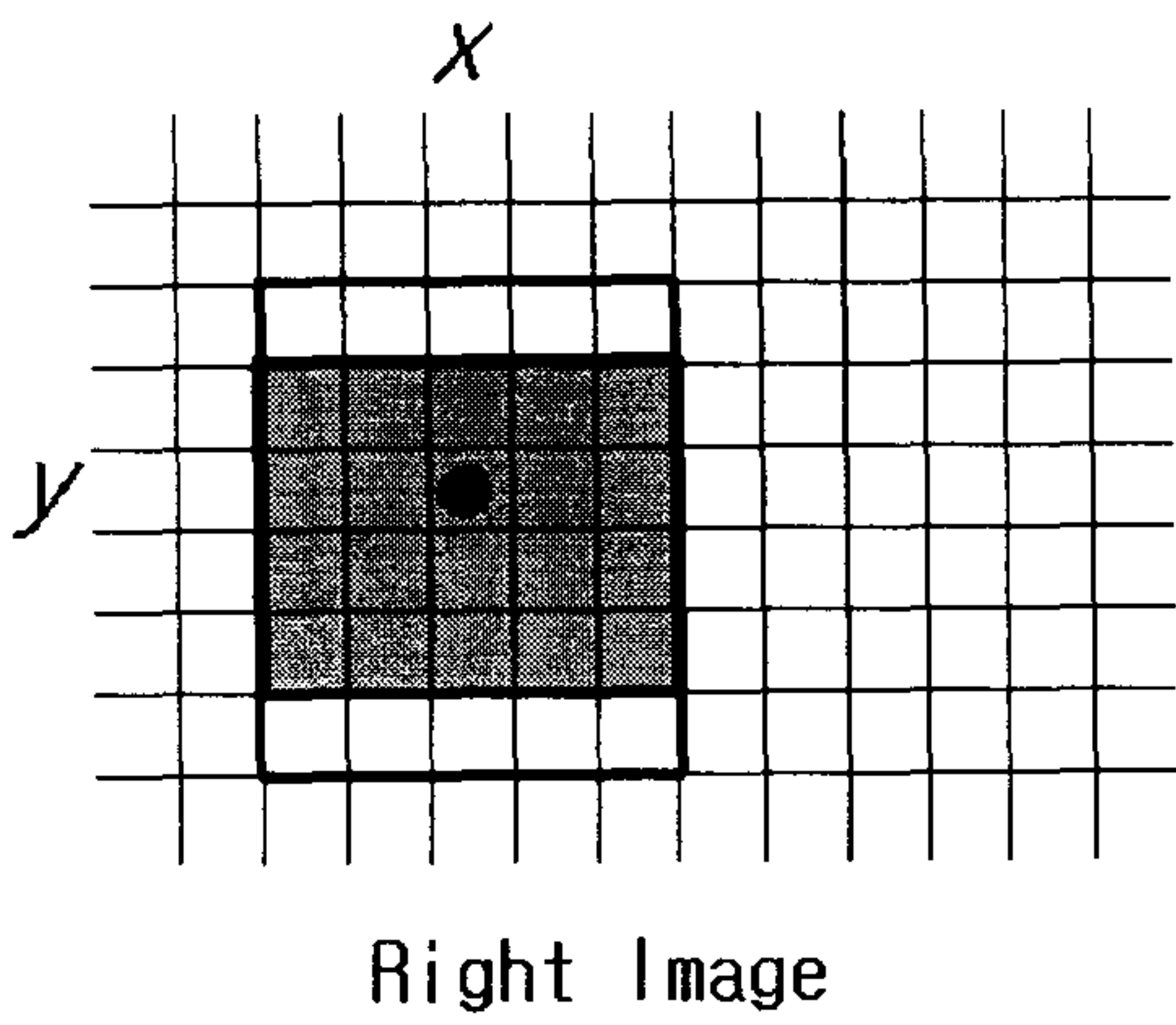
2단계 변위정보 추출 알고리즘의 고속 구현이 가능하게 하는 중복 연산이 제거된 MPC 정합 방법에 대해 설명한다. 일반적으로 영상의 해상도를 $(I_x \times I_y)$, 정합창의 크기를 $(W_x \times W_y)$ 탐색 범위를 Sr 이라면, 영역 기반 정합 방법의 전체 계산량은 $(I_x \times I_y) \times (W_x \times W_y) \times Sr$ 에 비례하는 방대한 계산량이 필요하다. 이처럼 많은 계산량이 필요한 이유는 영역 기반 방법에 내재하고 있는 정합창의 이동에 따른 중복 연산에 의한 것으로 이를 그림 3.4에 나타내었다. 기준 영상인 우 영상의 $R(x, y)$ 화소와 탐색 영상인 좌 영상의 $L(x+d, y)$ 화소간 유사도가 계산되었다면 $R(x+1, y)$ 과 $L(x+1+d, y)$ 의 유사도를 계산할 때 그림 3.4의 (a)에서와 같이 수평 방향으로 정합창이 중복되는 빗금친 부분만큼 중복 계산을 한다는 것을 볼 수 있다. 또한 $R(x, y+1)$ 과 $L(x+d, y+1)$ 의 유사도 계산 시 수직 방향으로 중복 계산을 하고 있는 것을 그림 3.4의 (b)에서 볼 수 있다. 영역 기반 방법에 내재하고 있는 중복 계산을 효율적으로 제거함으로써 전체 계산량을 창 크기에 영향을 받지 않는 $(I_x \times I_y) \times Sr$ 에 비례하는 양으로 줄일 수 있다[10].

변위 히스토그램을 이용한 2단계 변위 추출 알고리즘을 효율적으로 구현하기 위한 중복 계산이 제거된 MPC 정합 방법은 다음과 같다. 먼저, V 가 정합창의 높이에 해당하는 W_y 크기의 수직선간의 정합을, MPC 는 창간의 정합을 나타내고 w_x, w_y 가 창의 중심으로부터 창의 수평 수직 거리를 나타낸다면, 중복 연산을 제거한 MPC 연산 방법은 다음과 같이 표시할 수 있다.

$$w_x = \frac{W_x - 1}{2}, \quad w_y = \frac{W_y - 1}{2} \quad (3.8)$$



(a)



(b)

그림 3.4 영역 기반 방법에 내재된 중복 연산 개념도.

$$V(x, wy, d) = \sum_{j=-wy}^{wy} P(x, wy+j, d) \quad (3.9)$$

$$V(x, y+1, d) = V(x, y, d) + P(x, y+1+wy, d) - P(x, y-wy, d) \quad (3.10)$$

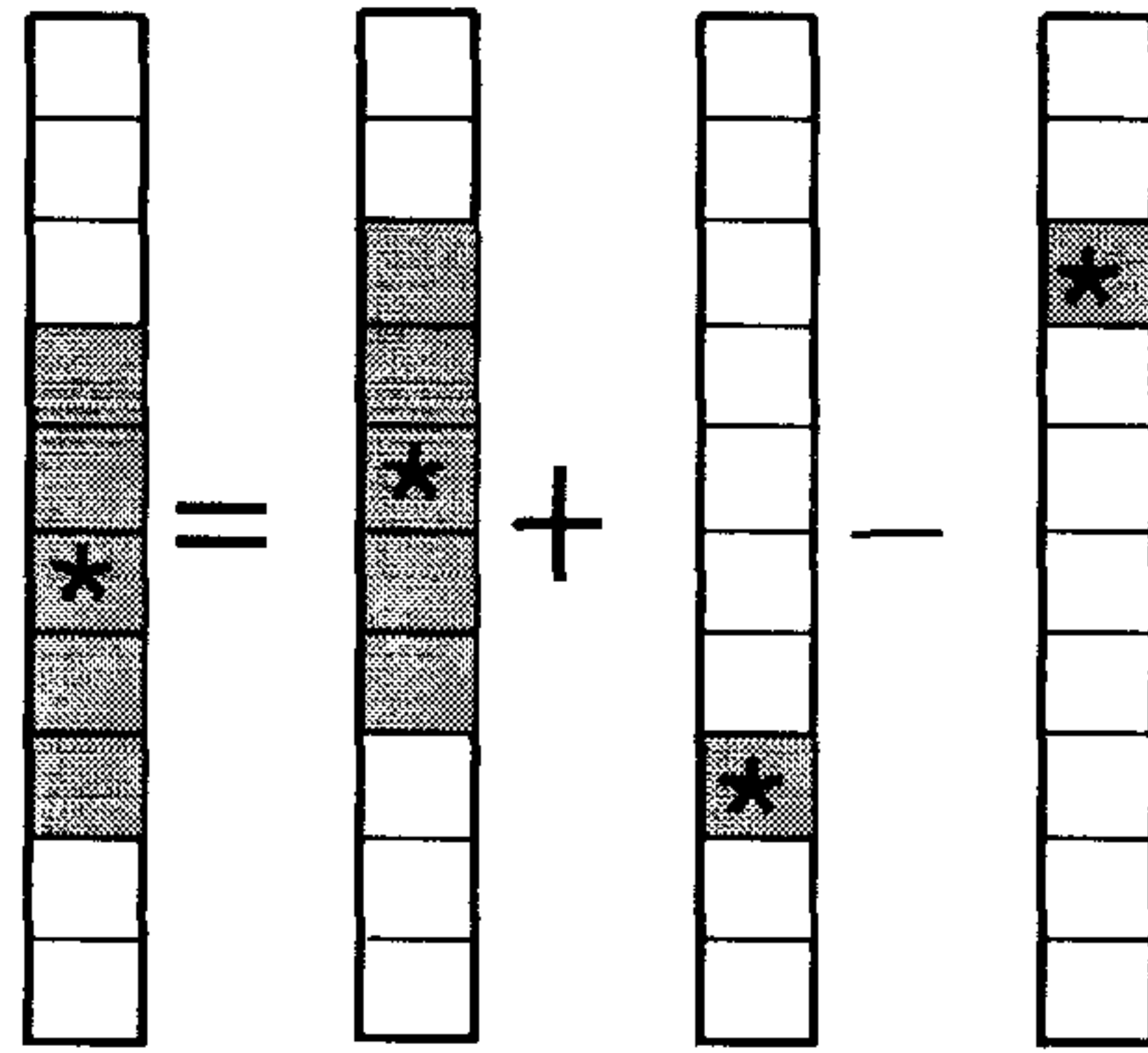
$$MPC(wx, y, d) = \sum_{i=-wx}^{wx} V(wx+i, y, d) \quad (3.11)$$

$$MPC(x+1, y, d) = MPC(x, y, d) + V(x+1+wx, y, d) - V(x-wx, y, d) \quad (3.12)$$

식 (3.9)의 $V(x, wy, d)$ 는 기준 화면의 wy 번째 행에 있는 화소 좌표 (x, wy) 와 탐색 화면의 화소 좌표 $(x+d, wy)$ 를 중심으로 하는 수직선간의 정합을 나타내며 모든 x, d 값에 대해 먼저 구한다. 즉, 각각의 x 값에 대해서 탐색 범위만큼의 V 값인 $V(x, wy, 0), V(x, wy, 1), V(x, wy, 2), \dots, V(x, wy, Sr)$ 를 구하게 된다. 이때, $P(x, j, d)$ 는 식 (3.6)에서와 같이 두 화면의 좌표 (x, j) 및 $(x+d, j)$ 에서의 화소 밝기가 비슷하면 1, 그렇지 않으면 0의 값을 갖는다. 다음 행의 화소를 중심으로 하는 수직선간의 정합인 식 (3.10)의 $V(x, y+1, d)$ 는 그림 3.5의 (a)와 같이 이전 행의 수직선간의 정합 $V(x, y, d)$ 에서 새로 첨가될 화소간의 정합 $P(x, y+1+wy, d)$ 를 더하고 제외되는 화소간의 정합 $P(x, y-wy, d)$ 를 빼 줌으로써 중복 연산 없이 계산될 수 있다. 식 (3.11)의 $MPC(wx, y, d)$ 는 기준 화면의 wx 번째 줄의 화소인 (wx, y) 와 탐색 화면의 화소 좌표 $(wx+d, y)$ 를 중심으로 하는 창간의 정합으로 수직선간의 정합 $V(wx+i, y, d)$ 를 $i=-wx$ 부터 wx 까지 더함으로써 계산된다. 다음 줄의 화소를 중심으로 하는 창간의 정합인 식 (3.12)의 $MPC(x+1, y, d)$ 는 그림 3.5의 (b)와 같이 이전 줄의 창간의 정합 $MPC(x, y, d)$ 에서 새로운 수직선간의 정합 $V(x+1+wx, y, d)$ 를 더하고 창에서 제외되는 수직선간의 정합 $V(x-wx, y, d)$ 를 빼 줌으로써 계산될 수 있다. 이러한 방법을 이용하면 wy 번째 행의

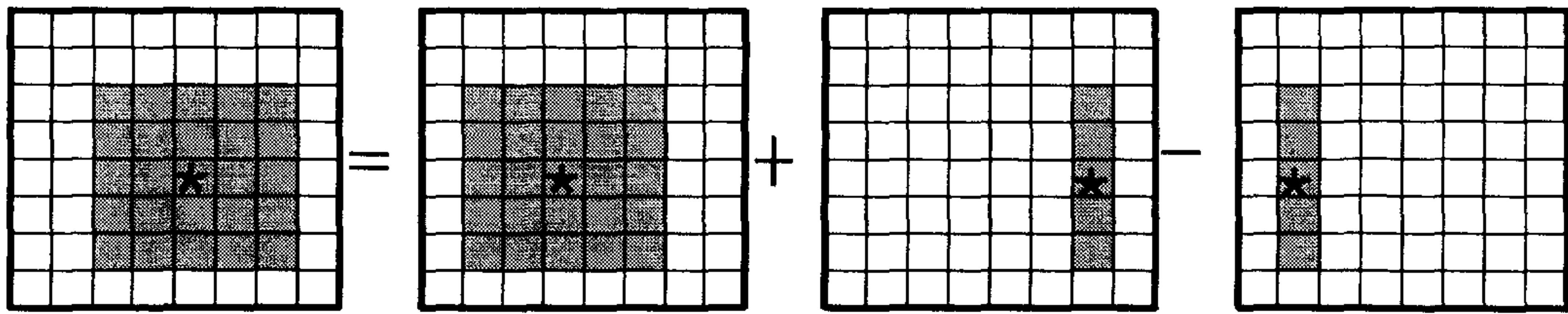
화소를 중심으로 하는 수직선간의 정합과 wx 번째 줄의 화소를 중심으로 하는 창간의 정합만 계산이 되면 창간의 정합은 창의 크기에 관계없이 식 (3.12)에 의해 한번에 구할 수 있다. 그림 3.6에 중복 계산이 제거된 MPC 스테레오 변위 추출 알고리즘의 흐름도를 나타내었다.

식 (3.9), (3.10), (3.11), (3.12)에서 보듯이 모든 화소의 탐색 범위가 동일해야만 중복 계산이 제거된 MPC 정합 방법을 사용 할 수 있다. 본 알고리즘은 낮은 해상도에서 변위 히스토그램을 사용하여 원 해상도의 탐색 범위를 제한하되 모든 화소의 탐색 범위를 동일하게 하여 중복 계산이 제거된 MPC 정합 방법을 1단계뿐만 아니라 2단계에서도 사용함으로써 고속 연산이 가능하도록 하였다.



$$V(x, y+1, d) = V(x, y, d) + T(x, y+1+wy, d) - T(x, y-wy, d)$$

(a)



$$MPC(x+1, y, d) = MPC(x, y, d) + V(x+1+wx, y, d) - V(x-wx, y, d)$$

(b)

그림 3.5 중복 계산을 제거하는 고속 MPC 연산 방법 (a)수직선간 정합 (b)창간 정합

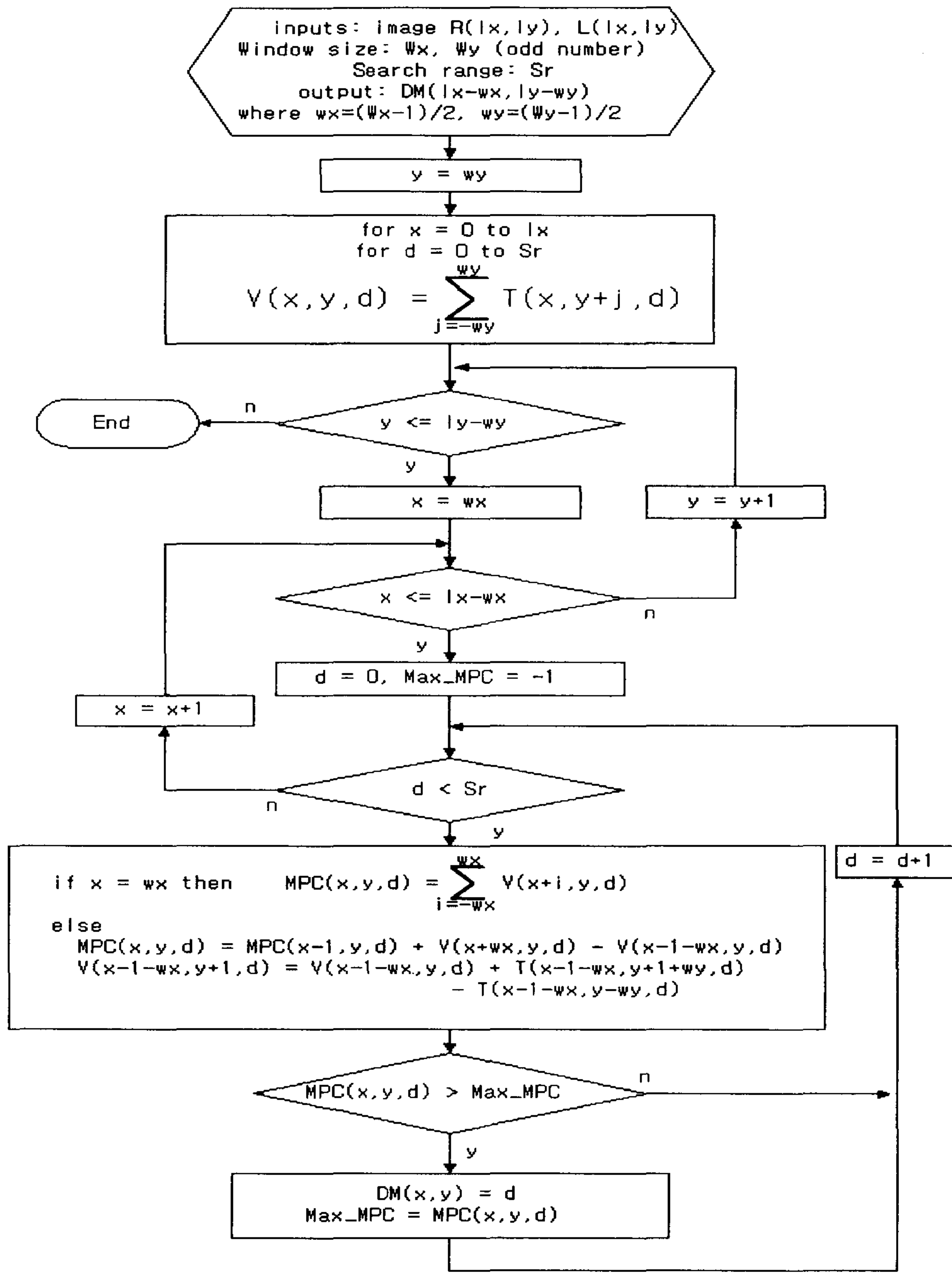


그림 3.6 중복 계산이 제거된 MPC 스테레오 변위 추출 알고리즘 흐름도.

다. 후처리 과정

고속 MPC 정합방법을 적용하여 구해진 초기 변위정보는 반복 후원, 좌우 정합 결과가 일치해야 한다는 Two-Views 제약 조건, 폐색 영역 탐색, 시그마 필터링 (Sigma Filtering), 소수점 화소 단위 정확도 등의 후처리 과정을 통해 정밀한 최종 변위 영상을 얻을 수 있다. 인접한 화소의 변위값은 서로 비슷하다는 smoothness 제한 조건을 이용하여 비슷한 값을 갖는 인접한 변위들은 안정한 상태에 도달할 때까지 또는 미리 설정된 횟수만큼 서로 반복 후원시킨다. 반복 후원으로 초기 변위가 어느 정도 개선되었지만 한쪽 영상에는 보이지만 다른 한쪽 영상에서는 보이지 않는 폐색 영역이나 밝기 변화가 빈약한 영역이 큰 곳에서는 여전히 잘못된 결과를 보인다. 이러한 정합 오류들은 제약 조건 적용 및 필터링단계를 통해 개선시킬 수 있고 소수점 화소 단위 정확도 단계에서는 화소 단위의 정확도를 갖는 변위값을 소수점 단위의 정확도를 갖도록 개선시킨다. 이들 각 과정의 설명은 다음과 같다.

a. 반복 후원

초기 변위 결과를 개선시키기 위해 Barnard가 제안한 반복 후원을 사용하였다. 반복 후원을 하기 위해서는 유사도 측정값을 유사도가 클수록 큰 값을 갖고 최대값 1, 최소값 0을 갖는 확률적인 형태로 변환을 시켜줘야 한다. Barnard가 사용한 유사도 측정 방법인 SSD는 유사도가 클수록 SSD값은 작기 때문에 확률 값으로 변환시키기 위한 반비례 관계의 식이 별도로 필요하다[12]. 본 연구에서 사용하는 MPC 유사도 측정값을 확률 값으로 변환시켜주기 위해서는 MPC 값을 정합창의 화소의 개수로 나누어주기만

하면 된다. (x, y) 위치에서의 화소가 변위 d 를 가질 가능성 $M_{x,y}(d)$ 는 다음과 같다.

$$M_{x,y}(d) = \frac{MPC(x, y, d)}{W_x \times W_y} \quad (3.13)$$

(x, y) 위치의 화소가 대응점을 찾지 못할 초기 확률 $P_{x,y}^0(d^*)$ 는 다음과 같다.

$$P_{x,y}^0(d^*) = 1 - \max_{d \neq d^*} M_{x,y}(d) \quad (3.14)$$

변위가 d 가 될 초기 확률 $P_{x,y}^0(d)$ 은 Bayes 정리를 이용하면 다음과 같다.

$$P_{x,y}^0(d) = P_{x,y}(d|x, y) \times (1 - P_{x,y}^0(d^*)), \quad d \neq d^* \quad (3.15)$$

여기서 $(1 - P_{x,y}^0(d^*))$ 는 (x, y) 위치의 화소가 정합 가능할 확률이고, $P_{x,y}(d|x, y)$ 는 (x, y) 위치의 화소가 정합 가능할 때 변위 d 를 가질 조건부 확률로서 다음과 같이 나타낸다.

$$P_{x,y}(d|x, y) = \frac{M_{x,y}(d)}{\sum_{d \neq d^*} M_{x,y}(d')} \quad (3.16)$$

식 (3.13), (3.14), (3.15), (3.16)를 사용하여 영상내 모든 화소가 탐색 범위내의 모든 가능한 변위를 가질 초기 확률값들을 계산한다. 즉, 영상내 (x, y) 위치의 화소가 변위 d 를 가질 확률값은 모든 x, y, d 값에 대해 계산되고 그림 3.7에서 나타나듯이 3차원 확률 공간 (x, y, d) 에 저장된다. 저장된 초기 확률값은 주변 화소들의 비슷한 변위들간의 후원을 통해 개선된다. 본 연구에서는 동일한 변위들끼리만 서로 후원하도록 하였다. $P_{x,y}^k(d)$ 를 후원을 통해서 개선시킨 $P_{x,y}^{k+1}(d)$ 는 (x, y) 주변에 변위 d 를 가질 확률이 높은 화소가 많으면 증가한다. 후원이 되는 정도 $Q_{x,y}^k(d)$ 는 다음과 같이 나타낼 수 있다.

$$Q_{x,y}^k(d) = \sum_{(x', y') \in N} P_{x', y'}^k(d), \quad d \neq d^* \quad (3.17)$$

여기서 N 은 (x, y) 주변의 화소들을 나타내며 본 연구에서는 정합창의 크기와 동일하게

$N=W_x \times W_y$ 로 하였다. 개선된 확률 $\widehat{P}_{x,y}^{k+1}(d)$, $\widehat{P}_{x,y}^{k+1}(d^*)$ 은 다음과 같다.

$$\widehat{P}_{x,y}^{k+1}(d) = P_{x,y}^k(d) \times (A + B \times Q_{x,y}^k(d)), \quad d \neq d^* \quad (3.18)$$

$$\widehat{P}_{x,y}^{k+1}(d^*) = P_{x,y}^k(d^*) \quad (3.19)$$

A 와 B 는 안정한 상태로 수렴해 나가는데 영향을 미치는 양의 상수이며 $A=0.3$, $B=3$ 을 사용하였다. 정규화 과정을 통한 새로운 확률 $P_{x,y}^{k+1}(d)$ 은 다음과 같다.

$$P_{x,y}^{k+1}(d) = \frac{\widehat{P}_{x,y}^{k+1}(d)}{\sum_d \widehat{P}_{x,y}^{k+1}(d)} \quad (3.20)$$

식 (3.13),(3.14),(3.15),(3.16)를 사용하여 모든 화소들이 가능한 변위를 가질 초기 확률값들이 계산되고, 식 (3.17),(3.18),(3.19),(3.20)을 사용하는 반복적인 후원을 통해 초기 확률값들은 점차 개선된다.

b. 제약 조건 적용 및 필터링

폐색 영역과 밝기변화가 빈약한 영역이 큰 곳에서는 여전히 잘못된 결과일 확률이 높음으로 Two-Views 제약 조건, 폐색 영역 탐색, 시그마 필터링 등의 단계를 통하여 정합 오류들은 개선될 수 있다.

① Two-Views 제약 조건

Two-Views 제약 조건[2]이란 우측 영상을 기준으로 계산된 변위와 좌측 영상을 기준으로 계산된 대응점의 변위는 서로 일치해야 한다는 조건으로서 이를 만족하지 못하는 변위를 갖는 화소는 정합에서 제외시켰다. 즉, 우 영상의 (x, y) 위치에서의 변위가 d 라면 좌 영상의 $(x+d, y)$ 위치의 변위도 d 가 되어야 한다. Two-Views 제약 조건을 식으로 나타내면 다음과 같다.

$$DM_R(x, y) = DM_L(x + DM_R(x, y), y) \quad (3.21)$$

식 (3.21)에서 $DM_R(x, y), DM_L(x, y)$ 는 각각 우좌 영상의 (x, y) 위치에서의 변위값을 나타낸다.

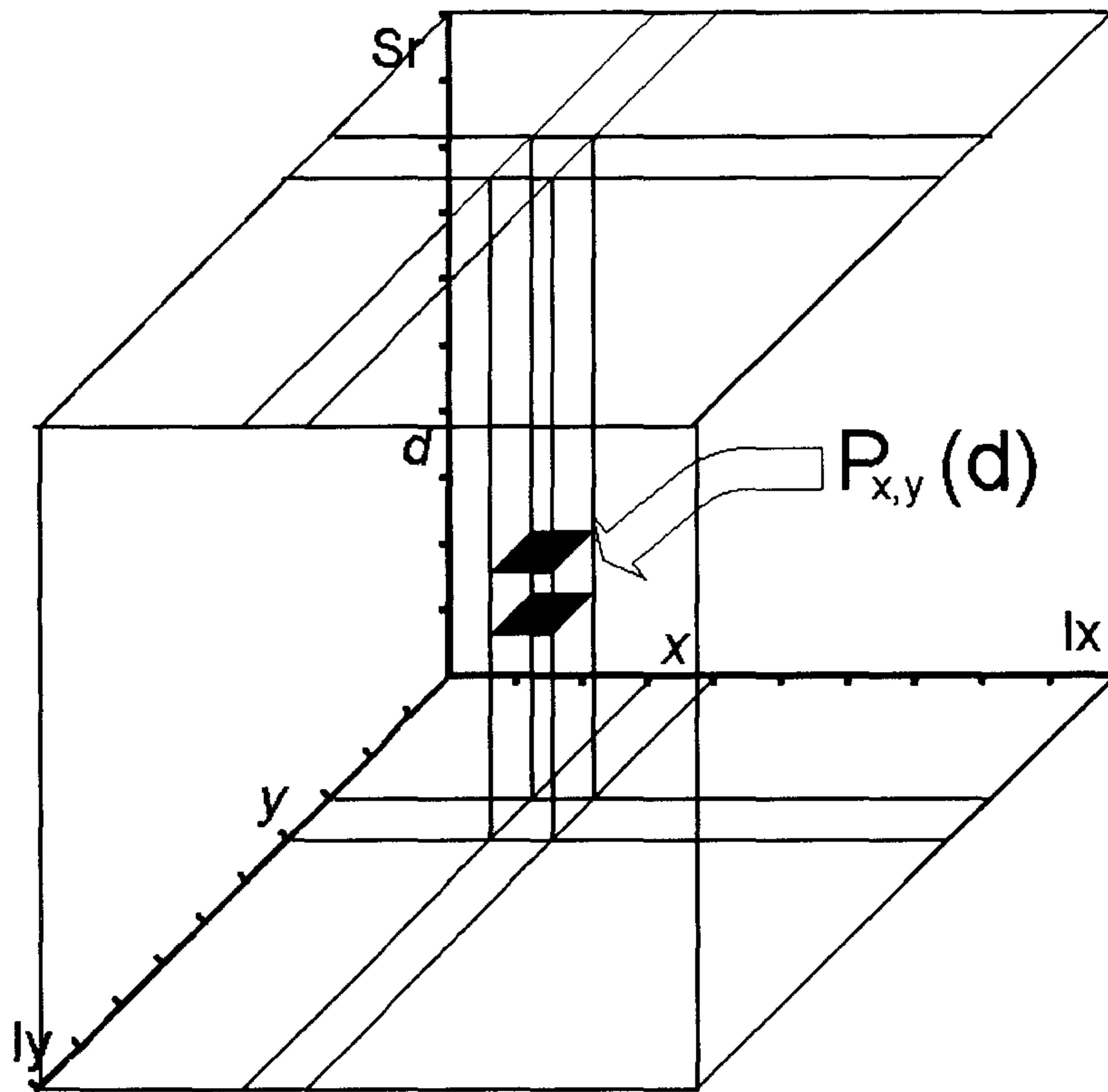


그림 3.7 (x, y) 위치의 화소 변위가 d 가 될 3 차원 확률 공간.

② 폐색 영역 탐색

폐색 영역이란 한쪽 영상에서는 보이지만 다른 쪽 영상에서는 보이지 않는 영역을 말한다. 우 영상의 변위 영상에서는 변위값이 x 축 상에서 좌에서 우로 급격히 작아지는 경계 부분이 폐색 영역이라고 할 수 있으므로 우 영상을 기준으로 계산된 변위 영상에서 Two-Views 제약 조건으로 제외된 화소들 중에서 제외된 화소를 중심으로 왼쪽에 있는 제외되지 않은 화소의 변위값이 오른쪽에 있는 제외되지 않은 화소의 변위값과 비교하여 미리 정의된 값(본 연구에서는 2를 사용)보다 클 때 폐색 영역이라고 지정하였다.

③ 시그마 필터링

Two-Views 제약 조건으로 제외됐지만 폐색 영역이라고 지정되지 않은 영역을 보간하는 방법으로 시그마 필터링을 사용하였다. 시그마 필터링[13]이란 보간하고자 하는 화소를 중심으로 일정한 크기의 창 내의 화소들의 변위값의 평균과 표준편차를 구하고 평균을 중심으로 표준편차 범위내의 변위값들에 대해서만 다시 평균을 구함으로써 경계가 보존되면서 보간하는 방법이다.

c. 소수점 화소 단위 정확도 개선

화소 단위의 정확도를 갖는 변위를 최대 확률 값을 갖는 변위를 중심으로 2차함수 근사화를 시키고 극점의 좌표를 계산함으로써 그림 3.8과 같이 소수점 단위 정확도의 변위를 얻을 수 있다[3]. 2차 함수의 식은 다음과 같다.

$$y = ax^2 + bx + c \quad (3.22)$$

(x, y) 위치의 화소가 변위 d_{max} 에서 최대의 확률값 $P_{x,y}(d_{max})$ 를 가진다고 하면, d_{max} 주변의 값들로 식 (3.22)를 만들고 행렬로 표현하면 다음과 같다.

$$Ax = b \quad (3.23)$$

여기서,

$$A = \begin{bmatrix} (d_{max} - 1)^2 & (d_{max} - 1) & 1 \\ d_{max}^2 & d_{max} & 1 \\ (d_{max} + 1)^2 & (d_{max} + 1) & 1 \end{bmatrix}, \quad x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad b = \begin{bmatrix} P_{x,y}(d_{max} - 1) \\ P_{x,y}(d_{max}) \\ P_{x,y}(d_{max} + 1) \end{bmatrix}$$

x 는 Pseudo-Inverse 방법으로 다음과 같이 구할 수 있다[14].

$$x = (A^T A)^{-1} A^T b$$

2차 함수의 정확한 극점인 $-\frac{b}{2a}$ 가 소수점 정확도의 변위값이 된다.

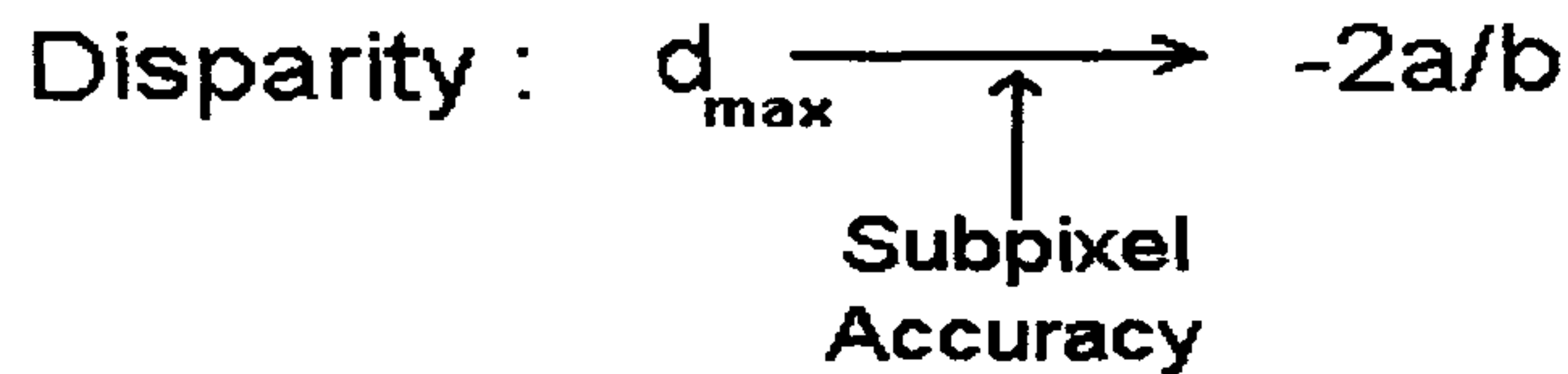
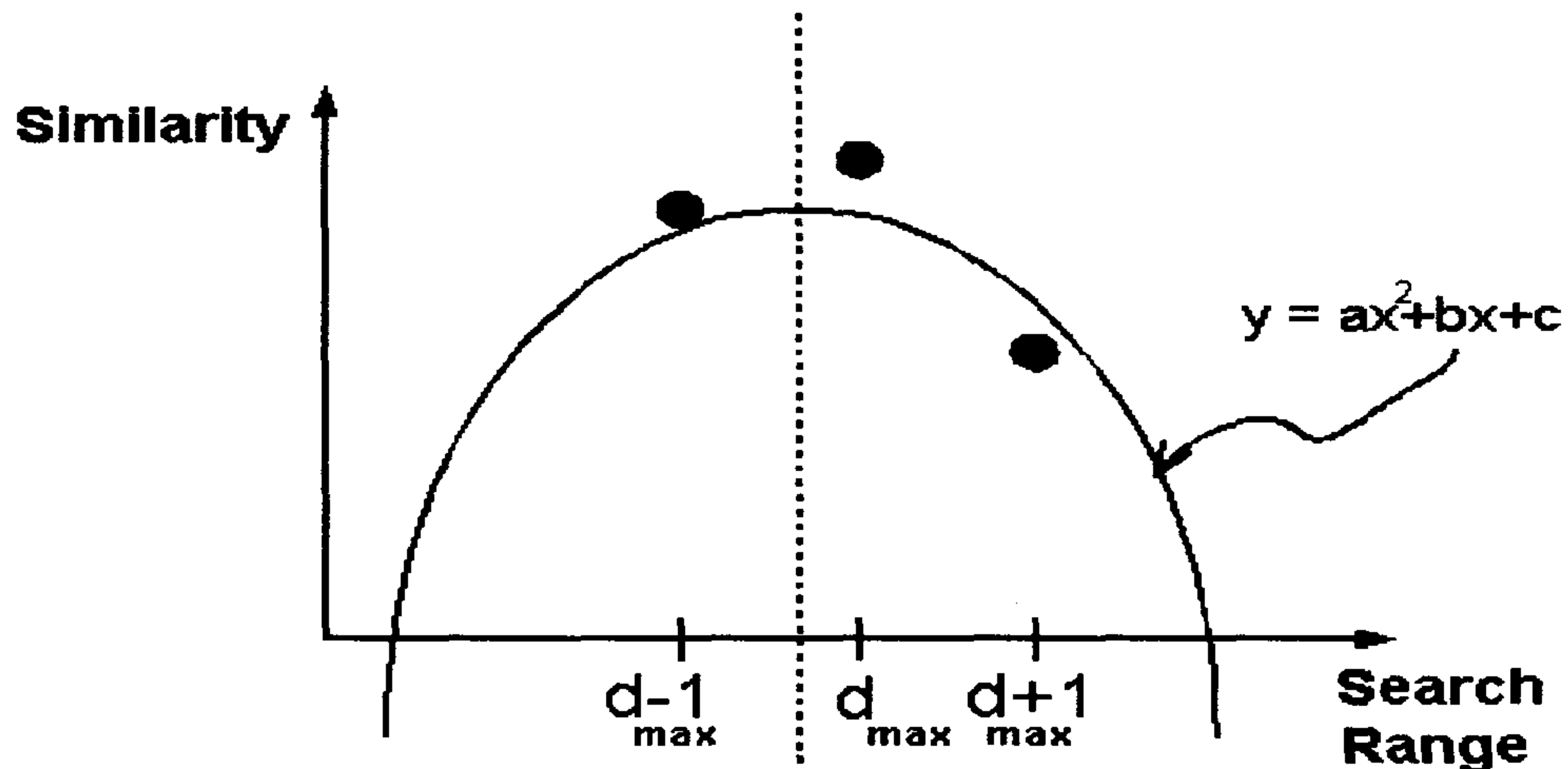


그림 3.8 소수점 정확도의 변위.

다. 실험 결과

실험실에서 스테레오 영상을 획득한 방법을 설명하고 무작위점 스테레오 영상과 일본 쓰쿠바 대학의 다시점 영상 데이터베이스중 널리 사용되는 Santa 영상, City 영상, Kid 영상, Plant 영상 그리고, 본 실험실에서 얻은 연구원, 인형, 우산 스테레오 영상에 대해 실험한 결과를 보인다. 모든 실험에 대해 정합창의 크기는 9×9 로 하였으며 Solaris 2.5 운영 체제하의 Sun Ultra Sparc 1에서 실험하였다. 그리고 변위와 거리와의 관계를 이용하여 변위로부터 거리 정보를 실제 얻을 수 있는지를 실험한 결과를 보인다.

a. 카메라 캘리브레이션과 스테레오 영상 획득

본 실험실에서는 2대의 흑백 카메라(CS8320)를 그림 3.1의 (c)와 같이 동일 직선 상에 평행한 광축을 갖도록 배치하여 스테레오 영상을 획득하였다. 이때 두 대의 카메라를 동일 직선 상에 평행한 광축을 갖도록 배치시키기 위해 5개의 점을 포함하는 평면을 카메라 앞에 위치 시켜서 좌, 우 영상을 얻은 후 대응점의 y 좌표인 y_L, y_R 이 동일하게 되도록 카메라의 위치를 조절하였다. 5개 점의 y 좌표가 모두 동일한 값을 가질 때까지 위 실험을 반복하였으며 최종적으로 Tsai의 캘리브레이션[8]으로 두 카메라의 외부 파라미터를 비교해 봄으로써 동일 직선 상에 평행한 광축을 갖도록 카메라가 배치되었음을 확인하였고 표 3.1에 캘리브레이션 결과를 요약하였다. 표 3.1의 T_x, T_y, T_z 는 각각 x, y, z 축 방향의 카메라 이동을 나타내고 R_x, R_y, R_z 는 각각의 축에 대한 회전을 나타낸다. T_x 를 제외한 외부 파라미터들이 거의 같다는 것을 알 수 있다. TI사의 C40 DSP칩을 사용하는 칼라 영상 획득 보드를 사용하여 3개의 흑백 카메라를 동시

에 입력할 수 있으며, 본 실험에서는 좌 카메라를 Green신호 단자에 입력시키고 우 카메라를 Red신호 단자에 입력시킴으로써 동시에 스테레오 영상을 얻을 수 있었다.

표 3.1 카메라 캘리브레이션 결과

	Tx[mm]	Ty[mm]	Tz[mm]	Rx[deg]	Ry[deg]	Rz[deg]
Left Camera	-124.3	-281.8	1227.6	-129.6	1.5	92.4
Right Camera	-228.2	-284.3	1214.5	-130.7	1.4	91.5

b. 무작위점 스테레오 영상

256×256 크기의 결과 변위 영상이 2단 웨딩 케이크 모양이 되는 무작위점 (Randomdot) 스테레오 영상을 배경의 변위는 0, 1단의 변위는 10, 2단의 변위는 12가 되도록 그림 3.9의 (a) (b)와 같이 만들었고 실제 변위 영상을 (c)에서 보이고 있다. (c)에서 흰색 사각형 테두리는 성능 분석의 대상이 되는 영역을 표시한 것이고 오른쪽 경계 부근의 검은 영역은 폐색 영역을 나타낸다. MPC 유사도 측정 방법의 성능을 비교하기 위해 무작위점 스테레오 영상에 대해 SAD, NCC, MPC 정합 알고리즘을 적용하여 얻은 우 영상을 기준으로 하는 변위 영상들을 그림 3.9의 (d) (e) (f)에 각각 나타내었고 20%의 salt and pepper 잡음을 첨가한 스테레오 영상에 대한 결과를 (g) (h) (i)에 각각 나타내었다. 표 3.2는 각각에 대해 그림 3.9의 (c)에 보이는 사각형 테두리 내의 화소들에 대한 오정합된 화소 수와 정정합된 화소의 비율을 나타내고 있다. SAD 및 NCC 알고리즘의 경우 폐색 영역으로 인하여 결과 변위 영상의 오른쪽 경계 부근에 오정합된 화소들이 관찰되고 경계선 연장 현상으로 인한 경계 부근에서의 오정합이 관찰되지만 MPC 알고리즘의 경우 폐색 영역에서의 오정합은 관찰되지만 경계 부근에서는 좋은 정합결과를 보임을 알 수 있다. 잡음을 20% 첨가한 경우 SAD와 NCC의 경우 오

정합이 많이 관찰되는 것을 볼 수 있지만 MPC 방법의 경우 상당히 강인한 특징을 보이는 것을 알 수 있다. 이는 MPC 방법에서는 정합창내의 한 화소가 유사도에 미치는 영향이 화소의 밝기값에 관계없이 동일하기 때문이다. 표 3.2에서 잡음을 첨가하지 않은 경우와 첨가한 경우 모두 MPC 정합 방법에서 정정합의 화소 비율이 가장 높음을 알 수 있다. 제안하는 알고리즘의 성능 비교를 위해 낮은 해상도의 정합 결과를 원 해상도에서 화소 단위로 이용하는 2단계 계층적인 정합 방법과 제안하는 변위 히스토그램을 이용하는 방법에 대해 후처리 과정을 포함하는 경우와 포함하지 않는 경우에 대해 각각 실험한 결과를 그림 3.10과 표 3.3에 보이고 있다. 계층적 방법과 제안하는 방법 모두 유사도 측정 방법으로 MPC를 사용하였다. 계층적인 방법은 낮은 해상도의 한 화소가 원 해상도에서 여러 화소에 영향을 줌으로 인해 세밀한 정보를 요구하는 경계 부근에서 더 나쁜 결과를 얻음을 알 수 있다. 계층적 방법과 비교하여 후처리 과정을 제외한 제안하는 방법은 변위 히스토그램을 사용하여 낮은 해상도의 정합 결과를 물체 단위로 이용함으로써 탐색 범위를 줄이되 모든 화소가 동일한 탐색 범위를 갖게되어 경계 부근에서도 좋은 결과를 얻을 수 있고 중복 연산을 제거하는 방법을 사용할 수 있어 연산 시간도 빠름을 알 수 있다. 여러 후처리 과정을 통해 결과가 개선됨을 그림 3.10과 표 3.2에서 볼 수 있다. 제안하는 방법은 Two-Views 제약 조건을 적용하기 위해 좌우 영상 모두를 기준으로 변위 영상을 구하고 좌우 각기 2번의 반복 후원을 함에도 탐색 범위를 효과적으로 줄이고 중복 계산을 제거함으로써 고속 연산이 가능함을 볼 수 있다. 또한 정합창의 크기를 크게 하면 계층적인 방법의 경우 연산 시간의 증가를 가져오지만 제안하는 방법은 정합창의 크기에 영향을 받지 않기 때문에 연산 시간의 변화가 없다.

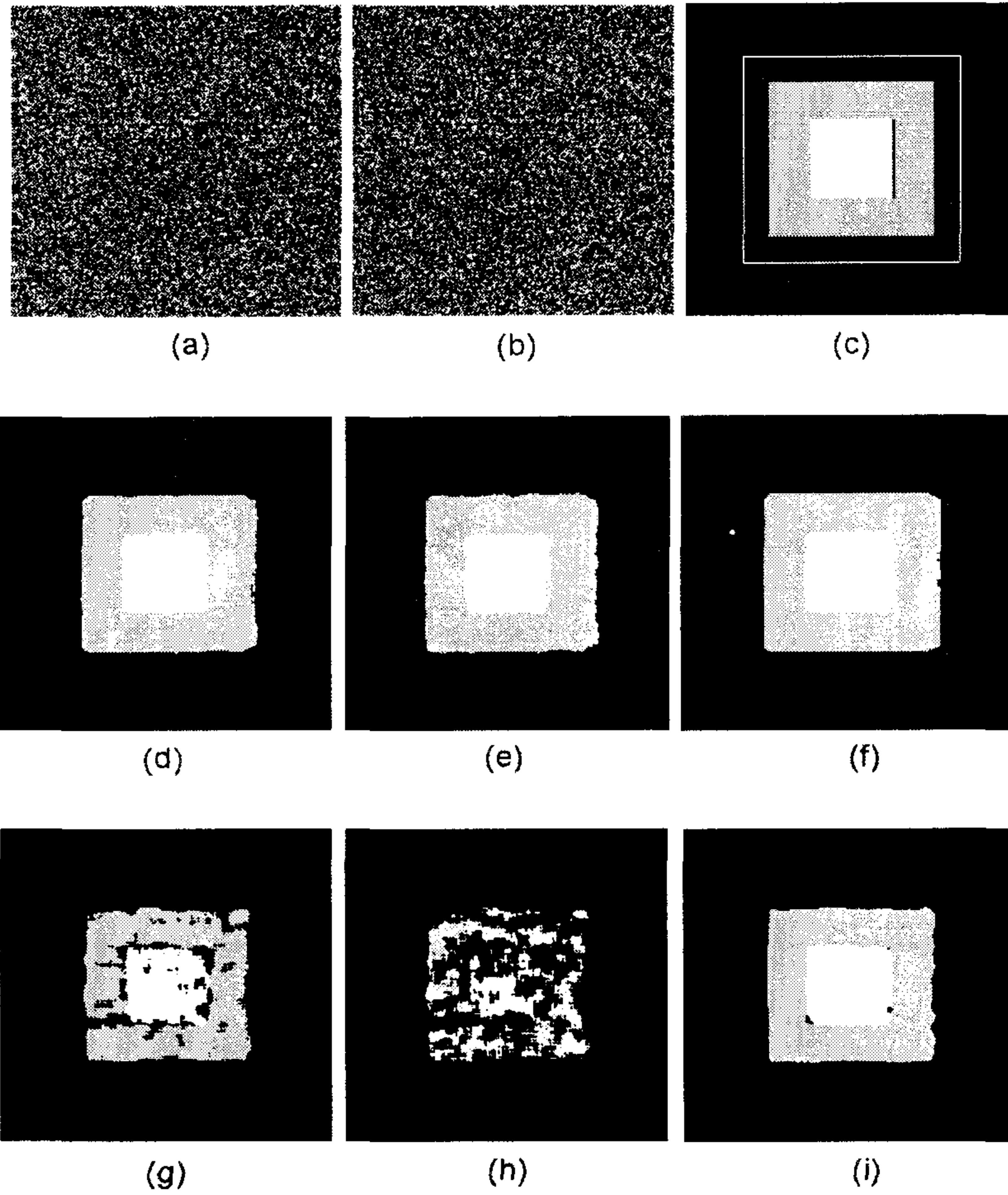


그림 3.9 무작위점 스테레오 영상에 대한 유사도 측정 방법의 성능 비교 (a)좌 영상 (b)우 영상 (c)실제 변위 영상 (d)SAD 결과(0% 잡음) (e)NCC 결과(0% 잡음) (f)MPC 결과(0% 잡음) (g)SAD 결과(20% 잡음) (h)NCC 결과(20% 잡음) (i)MPC 결과(20% 잡음).

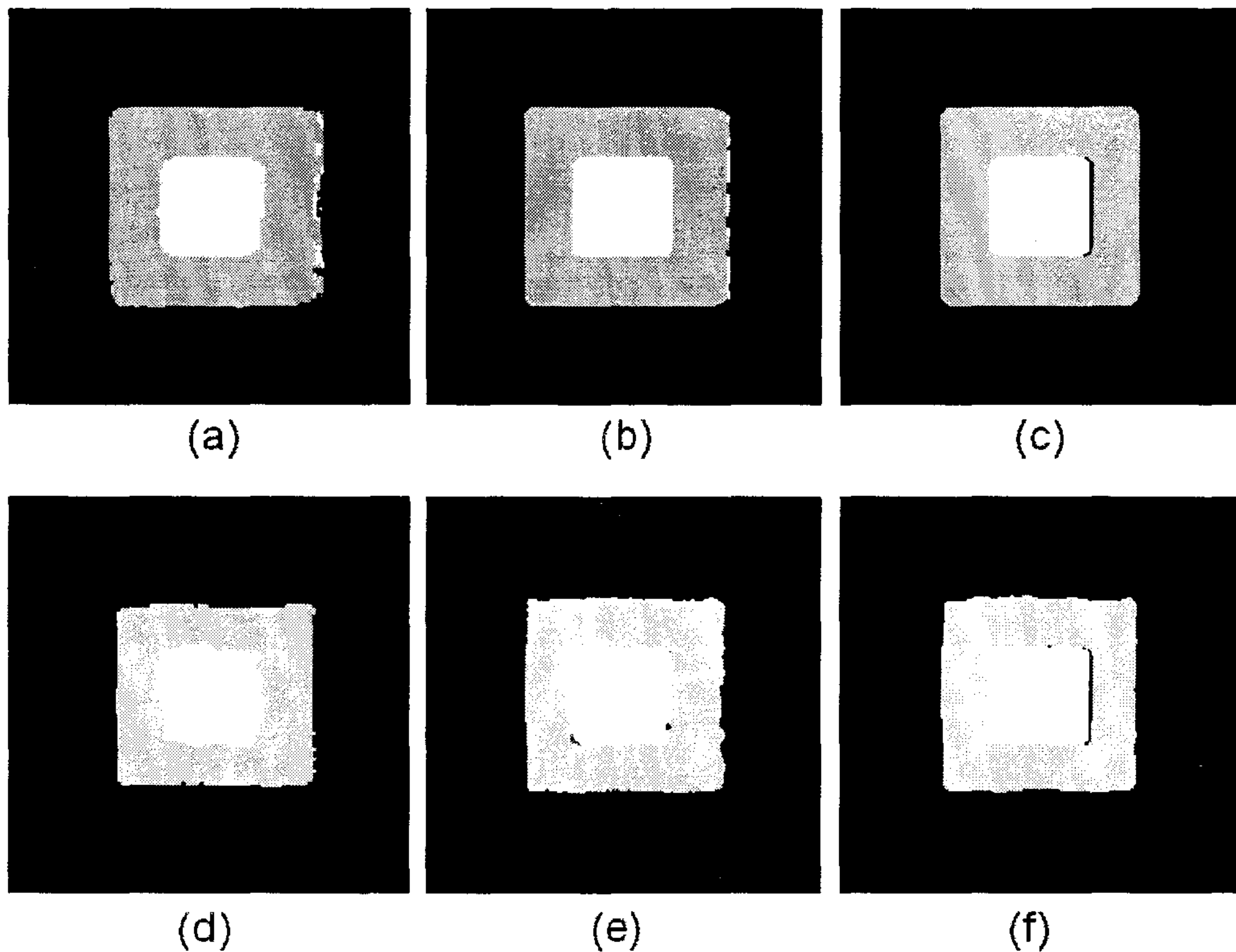


그림 3.10 무작위점 스테레오 영상에 대한 알고리즘 성능 비교 (a)계층적 방법(0% 잡음) (b)제안하는 방법(0% 잡음, 후처리 제외) (c)제안하는 방법(0% 잡음, 후처리 포함) (d)계층적 방법(20% 잡음) (e)제안하는 방법(20% 잡음, 후처리 제외) (f)제안하는 방법(20% 잡음, 후처리 포함).

표 3.2 유사도 성능 비교

salt and pepper noise		SAD	NCC	MPC
		오정합 화소 수	1537	1662
0%	정정합 화소 비	94.55%	94.11%	94.86%
	오정합 화소 수	4535	11110	2208
20%	정정합 화소 비	83.93%	60.64%	92.18%

표 3.3 알고리즘 성능 비교

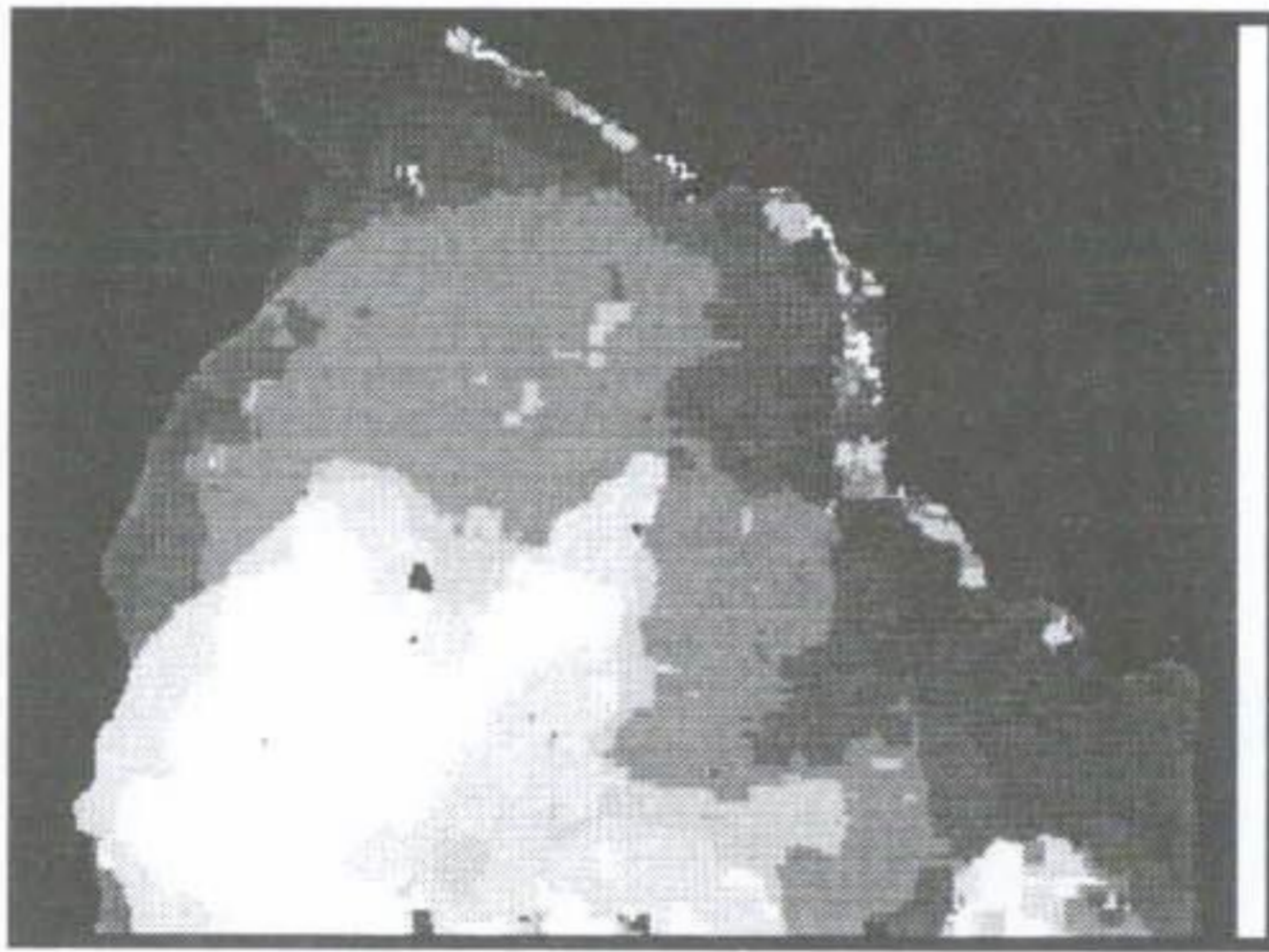
salt and pepper noise		계층적 방법	제안하는 방법	
			후처리 제외	후처리 포함
0%	오정합 화소 수	1648	1450	126
	정정합 화소 비	94.16%	98.86%	99.55%
20%	오정합 화소 수	4168	2208	1397
	정정합 화소 비	85.23%	92.18%	95.05%
연산 시간		5 secs	1 secs	3 secs

c. 실제 영상

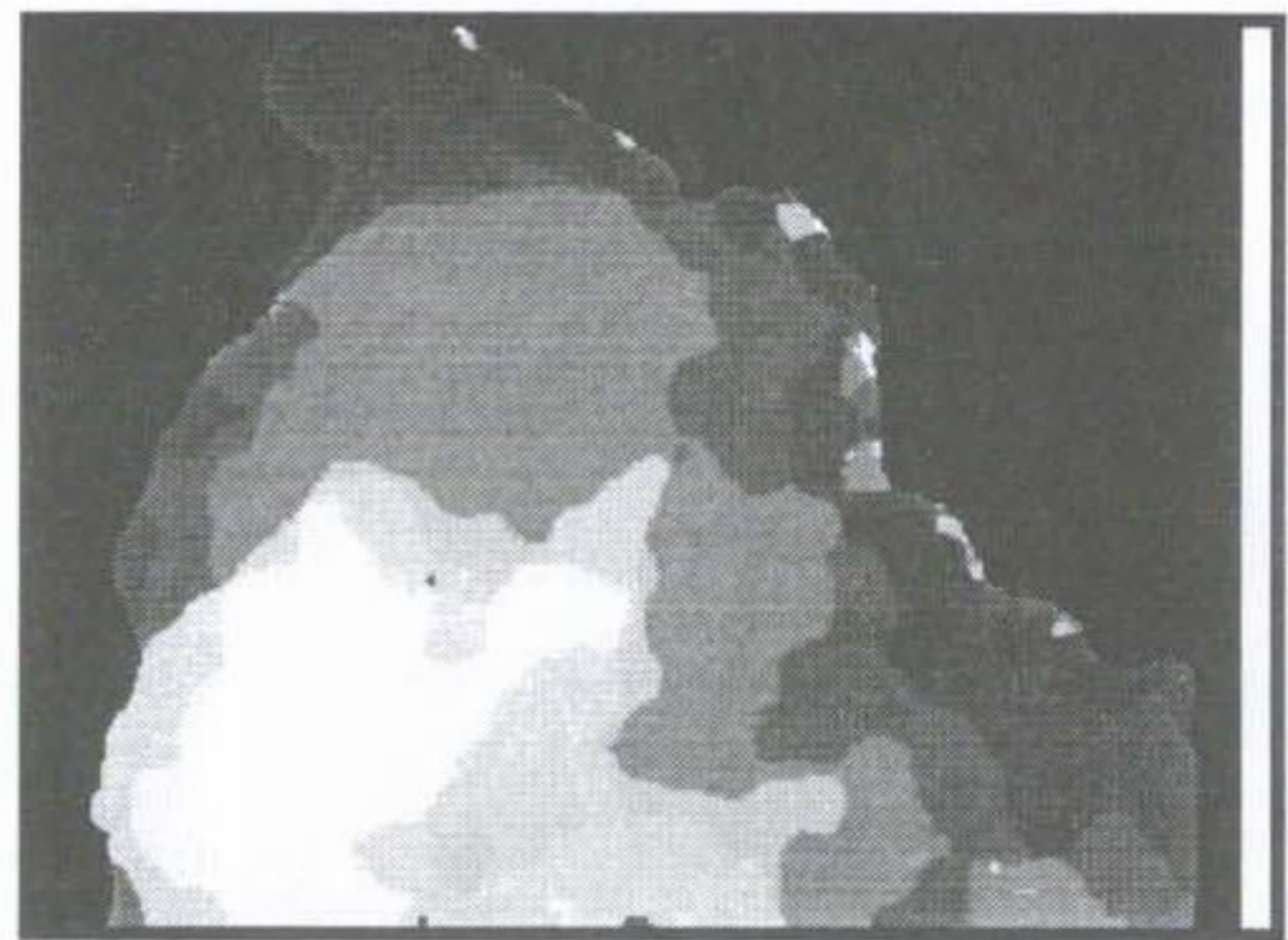
실험에 사용된 스테레오 영상 중 Santa 영상에 대한 각 단계별 결과 변위 영상을 그림 3.11에 나타내었다. 그림 3.11의 (a)는 변위 히스토그램을 이용하여 중복 계산이 없는 MPC 정합 방법으로 얻은 초기 변위 영상을 나타내며 기존의 영역기반 정합 방법 보다는 정확한 변위 정보를 나타내나 밝기 변화가 빈약한 우하 부분의 Santa 발 영역과 Santa와 배경과의 오른쪽 경계 부분에 나타나는 폐색 영역에서는 여전히 오정합을 나타내고 있다. 그림 3.11의 (b)는 반복 후원을 통해 개선된 변위 영상을 나타낸다. 밝기 변화가 빈약함으로 인한 오정합은 많이 제거되었지만 폐색 영역에서는 여전히 잘못된 결과를 나타낸다. 그림 3.11의 (c)는 제약 조건 적용 및 필터링을 적용한 후의 변위 영상으로 경계 부근의 검은 영역은 폐색 영역을 나타낸다. (d)는 변위값의 정확도를 소수점 단위까지 계산된 변위 영상을 보인다.

나머지 스테레오 영상과 이들의 실험 결과들을 쓰쿠바 대학의 영상들에 대해서는 그림 3.12에 실험실에서 얻은 영상에 대해서는 그림 3.13에 보이고 있다. 그림 3.12, 그림 3.13의 첫 번째 줄은 순서대로 City 영상, Kid 영상, Plant 영상과 연구원 영상, 인형

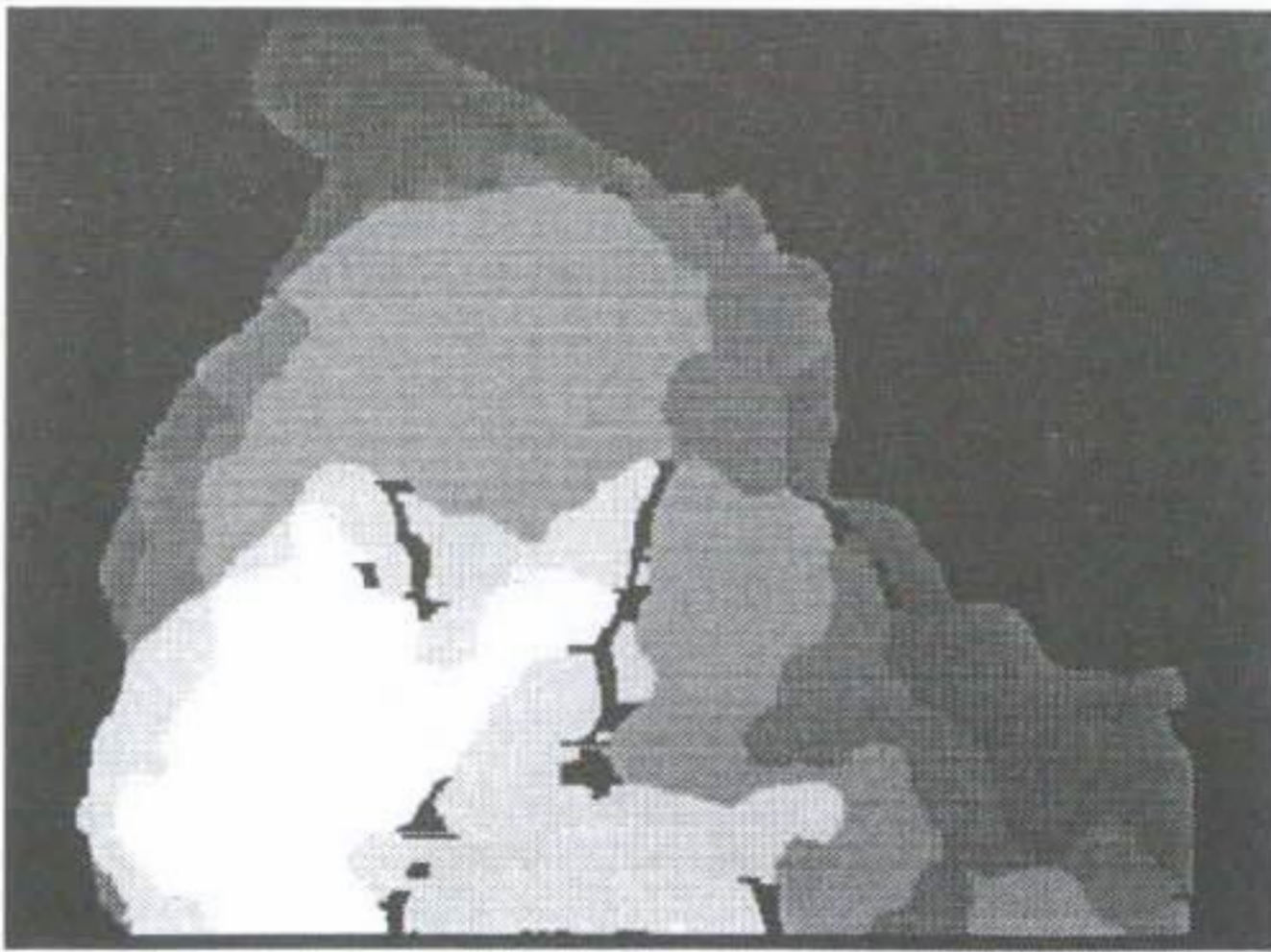
영상, 우산 영상의 스테레오 영상중 우 영상을 보이고 있다. 그림 3.12, 그림 3.13의 두 번째 줄은 각 영상에 대해 제약 조건 적용 및 필터링 단계 후의 변위 영상을 나타내며, 세 번째 줄은 소수점 단위 정확도 연산을 수행한 후의 변위 영상을 나타낸다. 사용된 스테레오 영상의 특징들을 표 3.4에, 각 단계별로 소요된 계산 시간을 표 3.5에 나타내었다. 표 3.4에서와 같이 Santa 영상의 경우 원 영상의 해상도는 320×240 이며, 저 해상도 변위 히스토그램을 구하는데 사용된 영상의 해상도는 160×120 임을 알 수 있다. 탐색 범위는 각각의 스테레오 영상의 전체 탐색 범위인 S_r 을 나타내고 물체 변위는 변위 히스토그램을 사용하여 계산된 물체의 변위 분포로서 제한된 탐색 범위 S_r' 을 나타낸다. Santa 영상의 경우 물체의 변위는 15에서 25 사이에 분포함을 알 수 있다. 표 3.5에서 초기 변위 처리 시간은 전처리 과정인 LOG, 낮은 해상도에서의 정합, 변위 히스토그램 계산과 이를 이용한 원 해상도에서의 정합으로 초기 변위가 계산되는데 걸리는 모든 시간을 나타내며, 반복 후원 처리 시간은 원 해상도의 영상에서 Barnard의 반복 후원을 2번 수행한 시간이다. Two-Views 제약, 폐색 영역 탐색, 시그마 필터링, 소수점 화소 단위의 정확도 등을 계산하는데 걸린 시간은 후처리 시간에 나타내었다. Two-Views 제약 조건을 적용하기 위해서는 우 영상을 기준으로한 변위 영상과 좌 영상을 기준으로한 변위 영상이 필요함으로 초기 변위와 반복 후원은 두 가지 경우를 모두 포함한 계산 시간을 나타낸다. 본 알고리즘의 전체 계산 시간은 저 해상도에서 구해진 변위 히스토그램의 결과에 따라 각각 다르게 구해지는 탐색 영역과 탐색 범위로 인해 가변적이며 256×240 크기의 영상에 대해 본 알고리즘을 적용하는 경우 약 5초 내지 8초 정도 걸림을 알 수 있다.



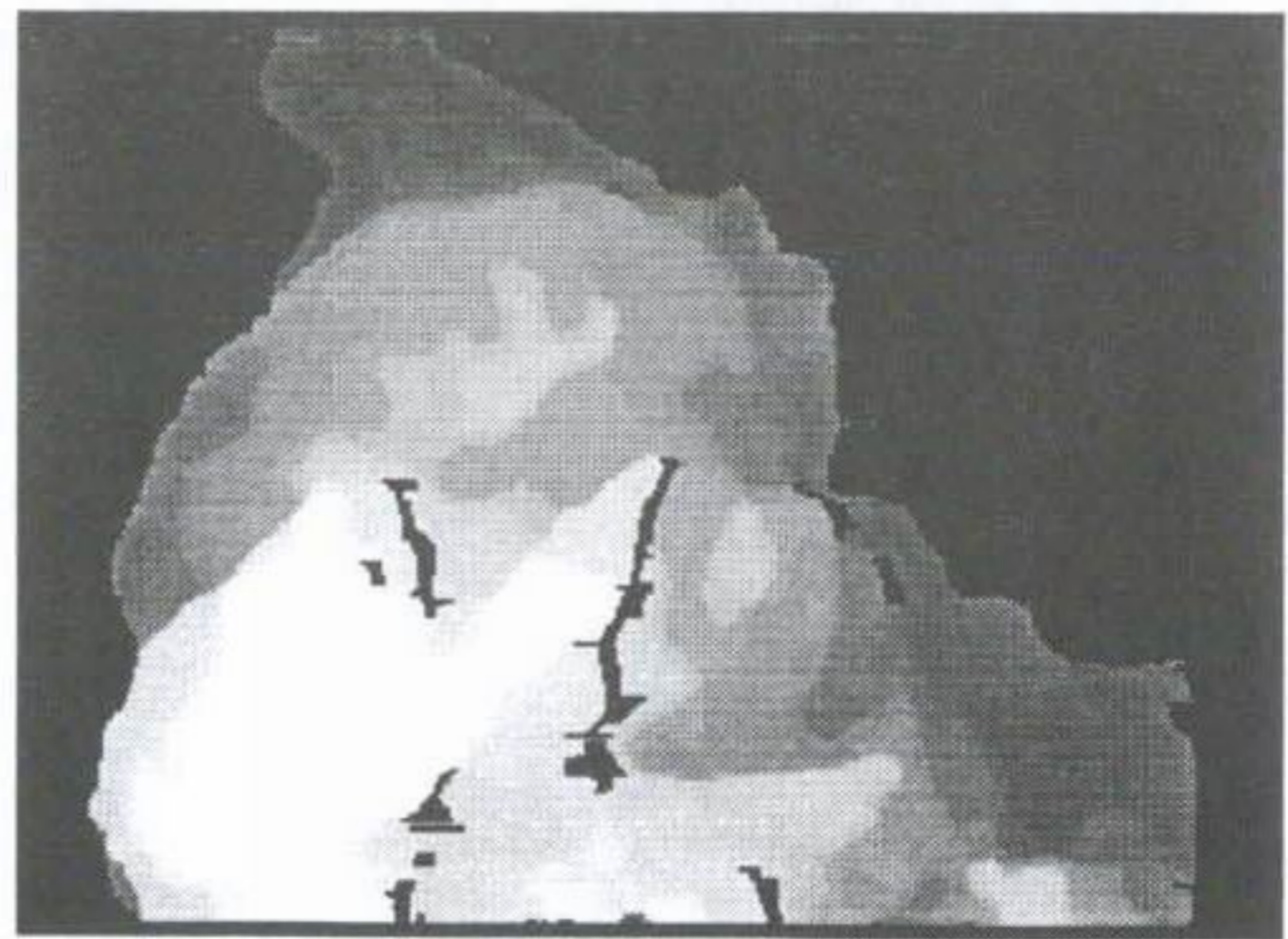
(a)



(b)



(c)



(d)

그림 3.11 Santa 영상의 단계별 변위 영상 (a)초기변위 (b)반복 후원후 (c)제약 조건 적용 및 필터링후 (d)소수점 화소 정확도 처리후

표 3.4 스테레오 영상들의 특징

	원 해상도	저 해상도	탐색 범위	물체 변위
Santa	320*240	160*120	30 pixels	15-25
City	256*240	128*120	22 pixels	5-21
Kid	256*240	128*120	33 pixels	15-21
Plant	256*240	128*120	30 pixels	13-21
연구원	256*240	128*120	30 pixels	17-23
인형	256*240	128*120	66 pixels	53-63
우산	256*240	128*120	60 pixels	41-53

표 3.5 스테레오 영상들의 단계별 연산 시간

	초기 변위	반복 후원	후처리	전체 시간
Santa	2 secs	6 secs	2 secs	10 secs
City	1 secs	5 secs	1 secs	7 secs
Kid	1 secs	4 secs	1 secs	6 secs
Plant	2 secs	5 secs	1 secs	8 secs
연구원	1 secs	3 secs	1 secs	5 secs
인형	1 secs	4 secs	1 secs	6 secs
우산	1 secs	4 secs	1 secs	6 secs

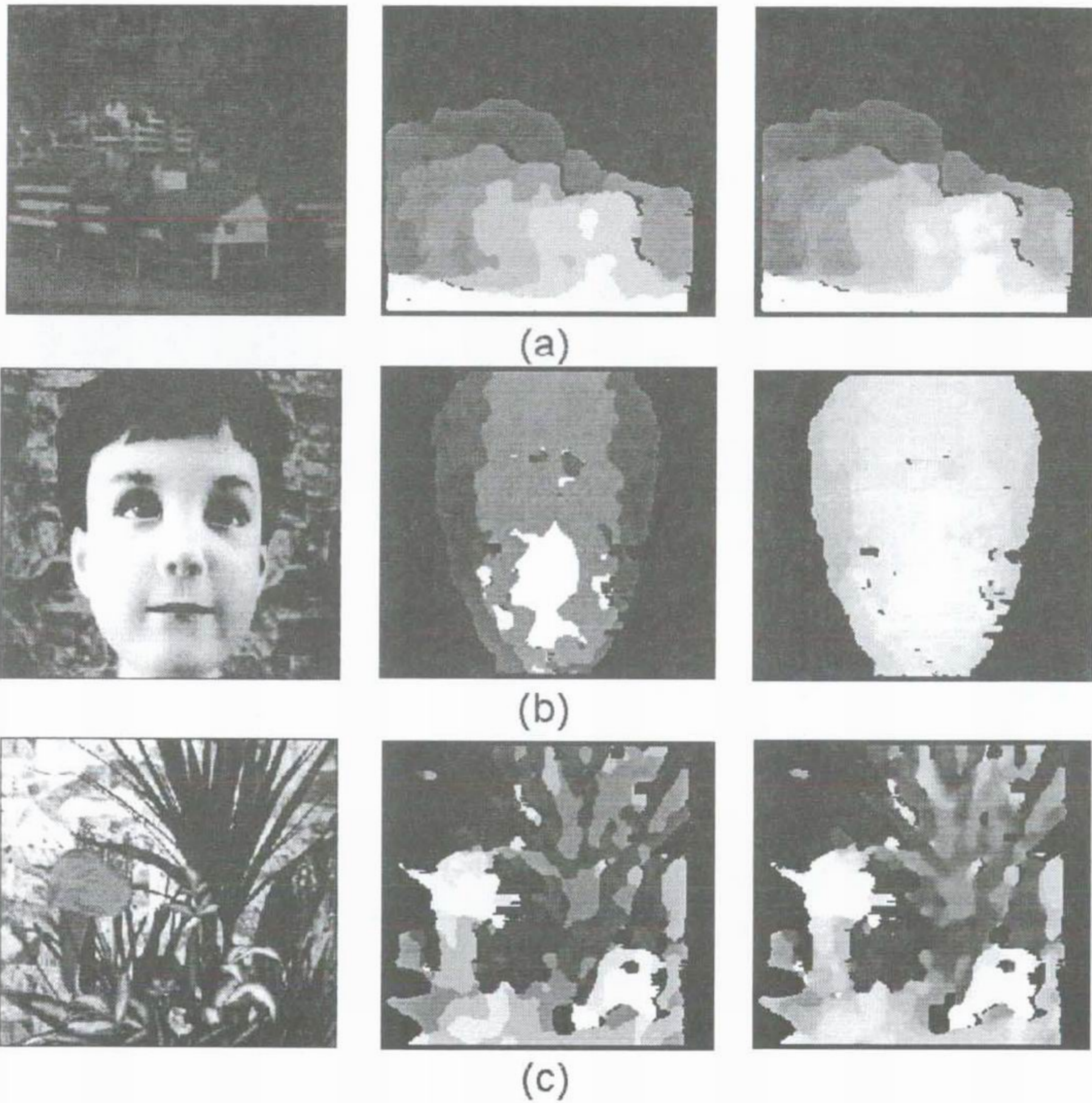


그림 3.12 쓰쿠바 대학 스테레오 영상들의 실험 결과. 첫 번째 줄: 우 영상, 두 번째 줄: 제약 조건 적용 및 필터링후의 변위 영상, 세 번째 줄: 소수점 정확도의 변위 영상 (a)City (b)Kid (c)Plant

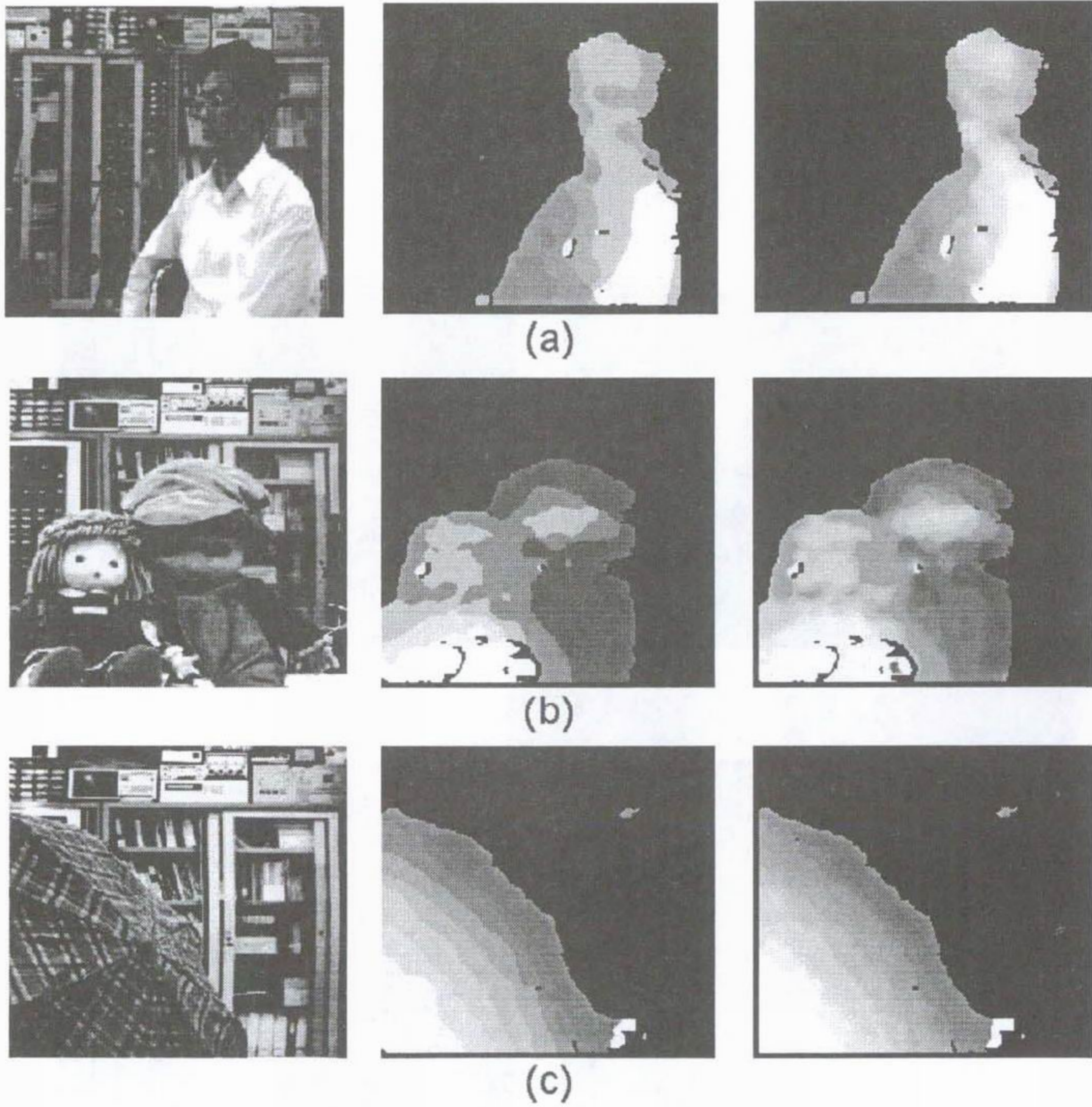


그림 3.13 실험실 스테레오 영상들의 실험 결과. 첫 번째 줄: 우 영상, 두 번째 줄: 제약 조건 적용 및 필터링후의 변위 영상, 세 번째 줄: 소수점 정확도의 변위 영상 (a)연구원 (b)인형 (c)우산

3절. 표면 재구성 (surface reconstruction)

영상으로부터 3 차원 표면 재구성을 하는 것은 변위도를 구한 뒤의 다음 과정으로 생각할 수 있다. 입력 스테레오 영상으로부터 3 차원 모델을 만드는 것은 영상 합성, 가상 현실등 여러 분야에 응용될 수 있을 것이다. 우선 입력 스테레오 영상은 쓰쿠바 대학에서 만든 영상 데이터 베이스중 산타영상과 키드(kid)영상을 사용했다. 원 영상 키드 영상과 산타 영상은 다음과 같다.

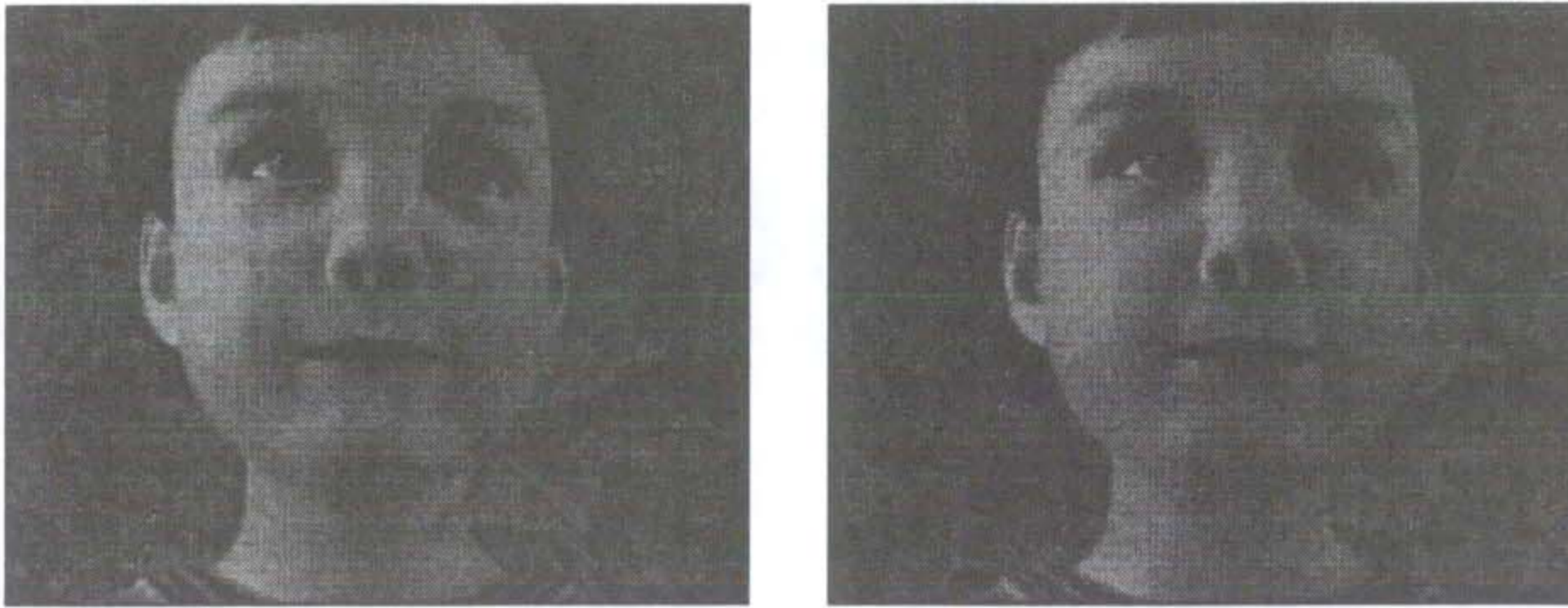


그림 3.14 키드 스테레오 영상의 좌영상과 우영상

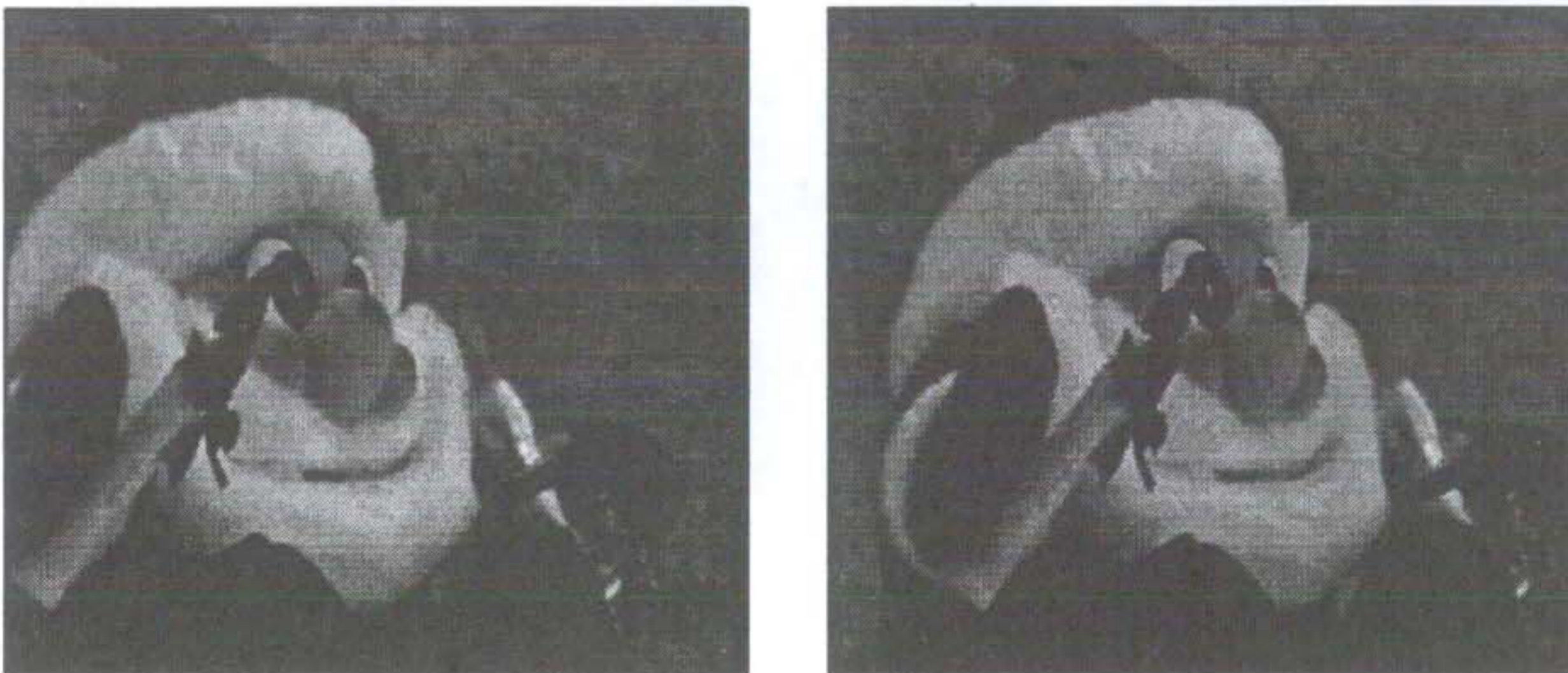


그림 3.15 산타 스테레오 영상의 좌영상과 우영상

키드 영상의 크기는 320x240이고 산타 영상은 256x240이다. 각 영상의 좌영상과 우영상의 베이스 라인 길이는 40mm로 선택했으며, 각 영상에 대해서 고성능 MPC 알고리즘을 이용해서 변위도를 구했다. 이 결과 영상은 다음 그림과 같다.

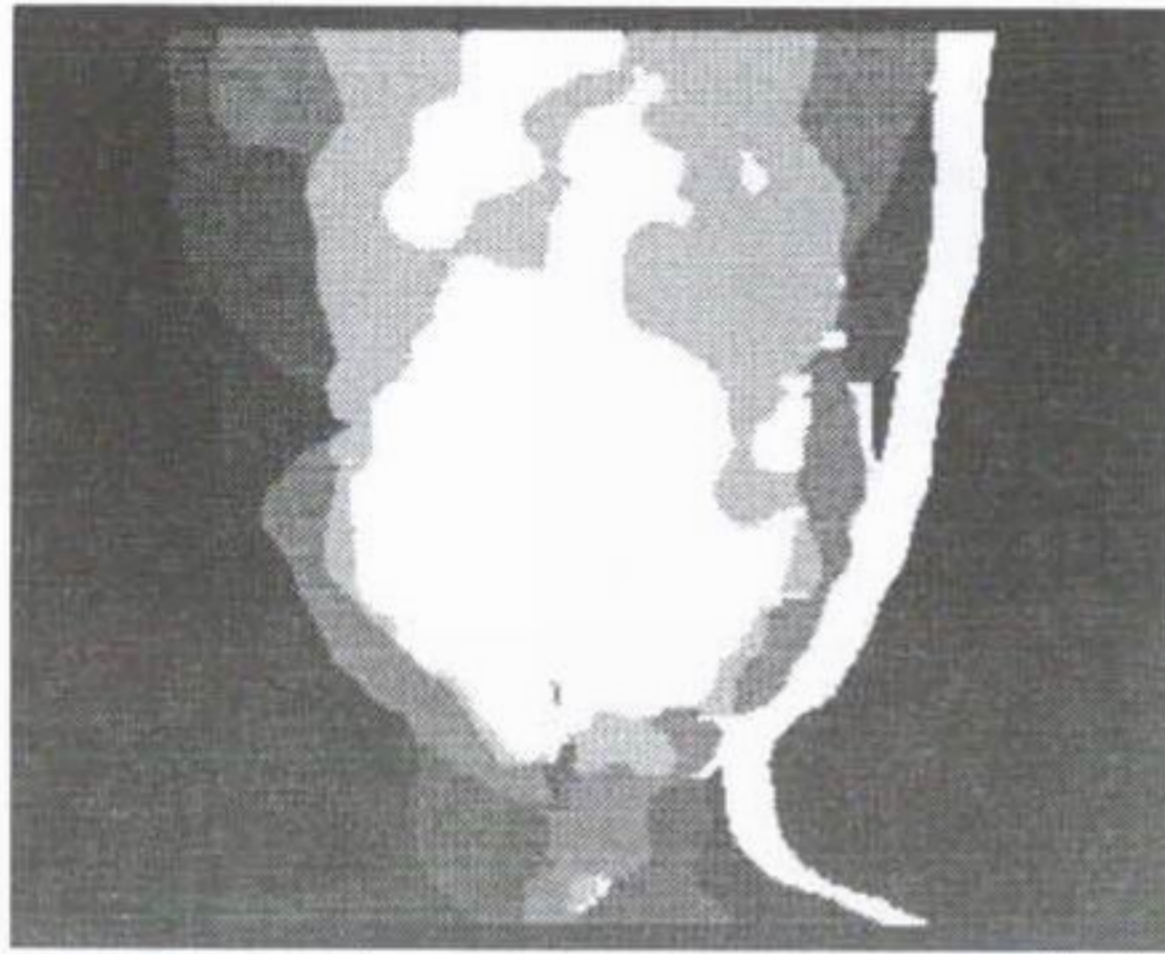


그림 3.16 키드 영상의 변위도



그림 3.17 산타 영상의 변위도

키드 영상은 얼굴 정면 부분에 텍스처가 부족하여 변위도가 얼굴의 모양을 정확히 반영하지 못하고 있는 것을 볼 수 있다. 그림 3.18과 그림 3.19는 이 두 개의 변위도를 사용해서 격자점(Grid)를 이용한 3 차원 매쉬를 재구성한 영상이다.

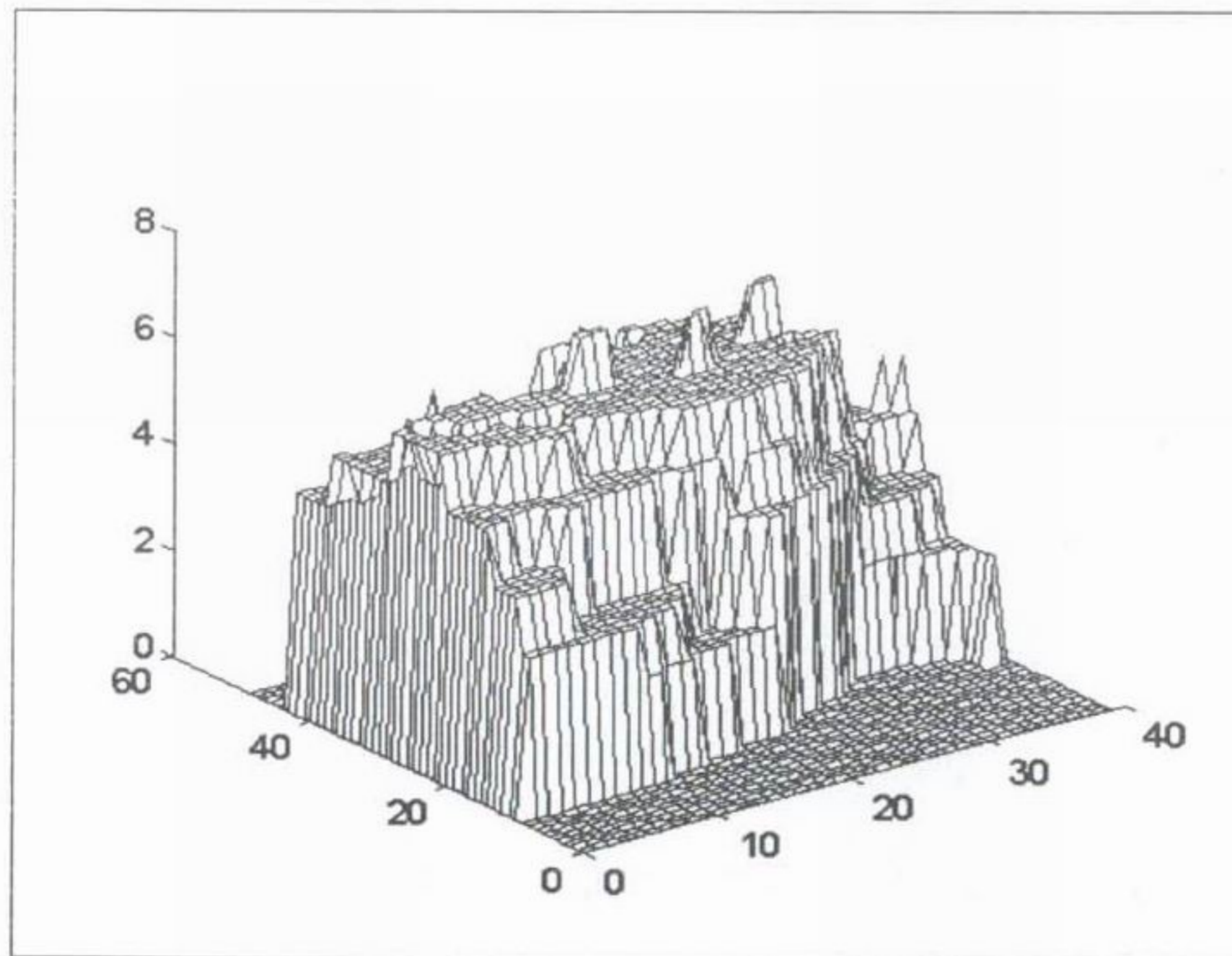


그림 3.18 키드영상의 변위도를 이용한 삼차원 모델

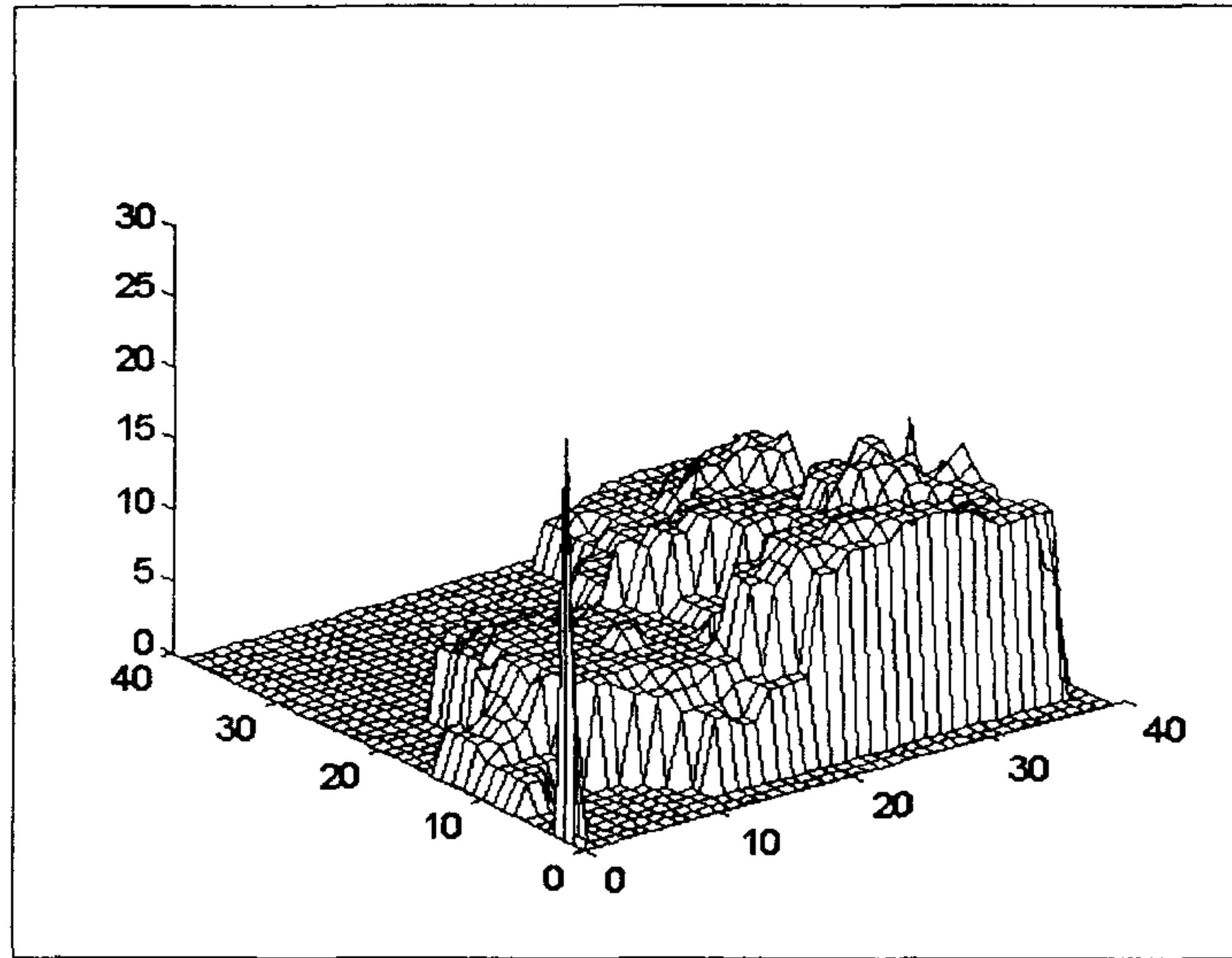


그림 3.19 산타영상의 변위도를 이용한 삼차원 모델

3 차원 모델은 변위도를 가로축과 세로축 데이터를 각각 40x40으로 샘플링한 값을 사용해서 구했다. z 축은 샘플링한 값을 나타낸다.

3 차원 키드 영상과 산타 영상에 대한 모델 모두 3 차원 모델의 대략적인 모양은 나타내고 있는 것을 볼 수 있다. 즉 키드 영상은 코 부분과 입 부분의 값이 크고 코와 입에서 멀어 질수록 작은 값을 가지는 것을 볼 수 있고, 산타 영상은 손 부분과 양초 부분이 큰 값을 가지고 모자 끝부분이 작은 값을 가지는 것을 볼 수 있다. 산타 영상의 (0, 0)위치에 나타난 피크값은 오정합 값이 샘플링되었기 때문이다. 그러나 이것만으로는 3 차원 모델을 만들었다고는 할 수 없다. 이것은 단순히 모델의 3 차원 깊이 정보만 나타내고 있을 뿐이다. 또 그림 3.18와 그림 3.19의 모델값은 불연속적으로 변하고 있는 것을 볼 수 있다. 즉 모델값의 분해능(resolution)의 한계 내에서 불연속적인 깊이 정보를 나타내고 있는 것이다. 앞으로 좀더 많은 분야에 응용하기 위해서는 분해능을 좀더 높은 상태에서 변위도를 구하는 방법과 연속적인 값을 가지는 3 차원 모델을 구하는 방법이 연구되어야 할 것이다.

4절. 실시간 구현을 위한 하드웨어 구조

스테레오 변위 추출은 2차원 영상으로부터 3차원 정보를 구하기 위해 가장 널리 사용되는 방법의 일환으로, 현재까지도 많은 연구가 활발하게 진행되고 있는 분야이다. 주된 연구의 방향은 크게, 정밀한 변위를 추출하기 위한 알고리즘의 개발이라는 측면과 실시간 응용에 적용할 수 있는 고속 알고리즘을 개발하는 측면으로 나눌 수 있다. 본 절에서는 1절에서 기술한 MPC 스테레오 정합 알고리즘에 기반한 고속의 스테레오 변위 추출기에 대해 기술한다. 앞에서 언급한 바와 같이 제안된 스테레오 변위 유사도 측정 방법은 화소간의 밝기 값을 직접 이용하지 않고 정합 창 내에서 밝기가 비슷한 화소의 개수를 사용함으로써 스테레오 비전 분야에서 널리 이용되는 영역 기반 스테레오 변위를 실시간 처리가 가능케 하는 하드웨어 구조를 효율적으로 구현할 수 있다.

가. 스테레오 변위 추출기

그림 3.20은 스테레오 변위 추출을 위한 하드웨어의 구조를 나타내며, 크게 나누어 P-BUFFER, MPC-Processor 및 V-BUFFER로 구성된다. 좌 영상 및 우 영상의 화소값들이 이 하드웨어의 입력으로 들어가고, MPC를 계산하기 위한 문턱값 Th 도 입력으로 주어진다. 출력으로는 우 영상의 좌표 (x,y) 에서의 최대 $MPC(x,y,d)(=MaxV)$ 와 그 때의 변위 $d(=DM(x,y))$ 이 나오게 된다. 이 외에 이 하드웨어 동작의 기준이 되는 Clock과 HSync, VSync 신호들이 입력되게 된다. 그림 3.20은 또한 P-UNIT, MPC-UNIT, Max_MPC 선택, CONTROL

-UNIT의 내부부분으로 구성되는 MPC-Processor의 구조를 보여주고 있다. MPC-Processor에는 MPC-UNIT가 $(Sr+1)(=64)$ 개 있어서 정합 탐색 범위에 대해서 병렬로 처리하고 있다는 것을 알 수 있다. 전체적으로 보면 영상의 화소값들이 MPC-Processor의 P-UNIT 로 입력되어 식 3.6에 나타낸 $P(x,y,d)$ 를 계산하게 된다. P-UNIT 에서는 기준영상 $R(x,y)$ 와 탐색영상 $L(x,y)$ 을 받아서 병렬로 $(Sr+1)$ 개의 $P(x,y,d)$, $d=0\sim Sr$, 값을 계산한다. 이렇게 계산된 값은 다음 연산을 위해 P-BUFFER에 저장된다. 이렇게 저장된 $P(x,y,d)$ 값은 다시 $(Sr+1)$ 개의 MPC-UNIT으로 입력되어 $MPC(x,y,d)$ 를 계산하게 된다. 이 때, $V(x,y,d)$ 값을 사용하게 되는데, 중복을 제거하고 연산하기 위해서 $V(x,y,d)$ 를 V-BUFFER에 저장한다. MPC-UNIT에서는 각 변위(d) 별로 P-BUFFER에서 $Pv(d)$ 를 받아서 $V(x,y,d)$ 를 생성하고, V_BUFFER로 부터는 기존에 저장되어 있던 $V(x,y,d)$ 를 받아서 각 변위에 대한 MPC값인 W 를 계산하여 출력한다. 각 변위 별로 나뉘어진 MPC-UNIT에서 계산된 MPC값들은 Max_MPC Select부분에 입력된다. Max_MPC Select부에서는 변위 $0\sim 63(=Sr)$ 에 대해서 나온 MPC값을 비교하여 제일 큰 값을 선택하고, 그 때의 변위를 $DM(x,y)$ 에 출력한다. 그림에서는 정합창의 크기를 $7\times 7(=Wx\times Wy)$ 로 하고 탐색영역의 크기 Sr 을 63으로 하여 설계된 하드웨어를 보이고 있다.

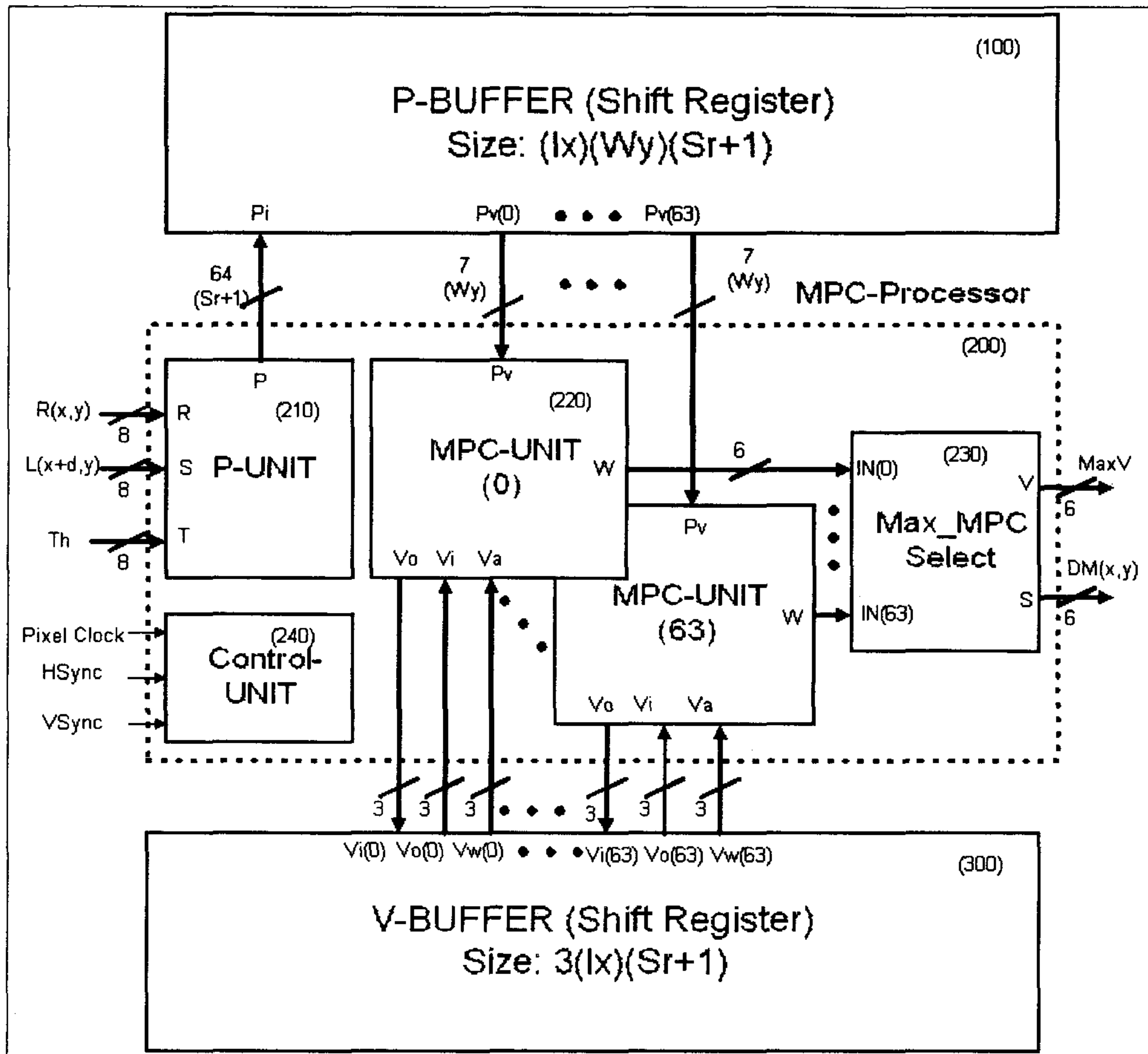


그림 3.20 병렬 MPC 스테레오 변위 추출기 구성도.

그림 3.21의 (a)에서는 P-BUFFER의 구조를 보여주고 있다. P-BUFFER는 MPC-Processor의 P-UNIT에서 계산한 $P(x,y,d)$ 값을 저장하고 적절한 시기에 다시 MPC-Processor에 있는 $(Sr + 1)$ 개의 MPC-UNIT(220)으로 제공하는 역할을 하며 $(Sr + 1)$ 개의 병렬 $(Ix \times Wy)$ -bit Shift register로 구성된다. 그림에서 P_R(x,y)로 표시된 각각의 블록은 병렬로 들어오는 64개($d=0 \sim 63$)의

1-bit 의 $P(x,y,d)$ 값을 저장하게 되고, 이 값은 clock에 맞춰서 화살표 방향으로 shift된다. 그림에서 각 행은 영상의 각 행에 대응하고, 각 열도 영상의 각 열에 대응한다. 따라서, 그림의 좌측에서 출력되는 값들은 영상의 하나의 열에 대한 $P(x,0,d) \sim P(x,Wy-1,d)$ 값이 나오게 되는데 각각이 변위 $d=0\sim 63$ 에 대해 64개의 0혹은 1 값을 가지게 된다. P-BUFFER에서는 이 값들을 다시 변위 d 에 대해서 정렬하여 각 변위에 대해 정합 창의 열 방향 길이인 $7(=Wy)$ 개 크기의 벡터 $Pv(0) \sim Pv(63)$ 을 출력한다. P-BUFFER는 Shift register를 사용하기 때문에 register가 다 차는 처음 $I_x \times W_y$ clock 이후부터 원하는 출력이 나오게 되지만, 그 이후는 시간 지연 없이 연속해서 영상을 처리할 수 있게 된다. 여기서 출력된 $Pv(d)$ 는 $V(i)$ 의 연산을 위해서 다시 MPC-Processor에 제공된다. 그림 3.21은 또한 V-BUFFER의 구조를 나타내고 있다. V-BUFFER는 MPC-Processor에 있는 S_r 개의 MPC-UNIT에서 계산된 $V(x,y,d)$ 값을 일시적으로 저장하는데 사용된다. 그림에서 $V_R(i)$ 로 표시된 각각의 블록은 병렬로 들어오는 $64(d=0\sim 63)$ 개의 3 bit $V(x,y,d)$ 값을 저장하고, clock에 맞춰서 화살표 방향의 다음 블록으로 shift한다. V-BUFFER의 출력은 $0\sim 63$ 의 각각의 변위에 따라 $V_R(W_x), V_R(0)$ 값이 되는데, 이는 정합 창이 수평방향(x 방향)으로 움직이면서 MPC를 계산할 때 각각 더해지고 빼지는 값이 된다. V-BUFFER도 $I_x - W_x$ clock 만큼의 초기 시간 지연을 가지게 된다. V-BUFFER의 출력은 MPC를 계산하는데 사용되기 위해서 MPC-UNIT에 입력된다.

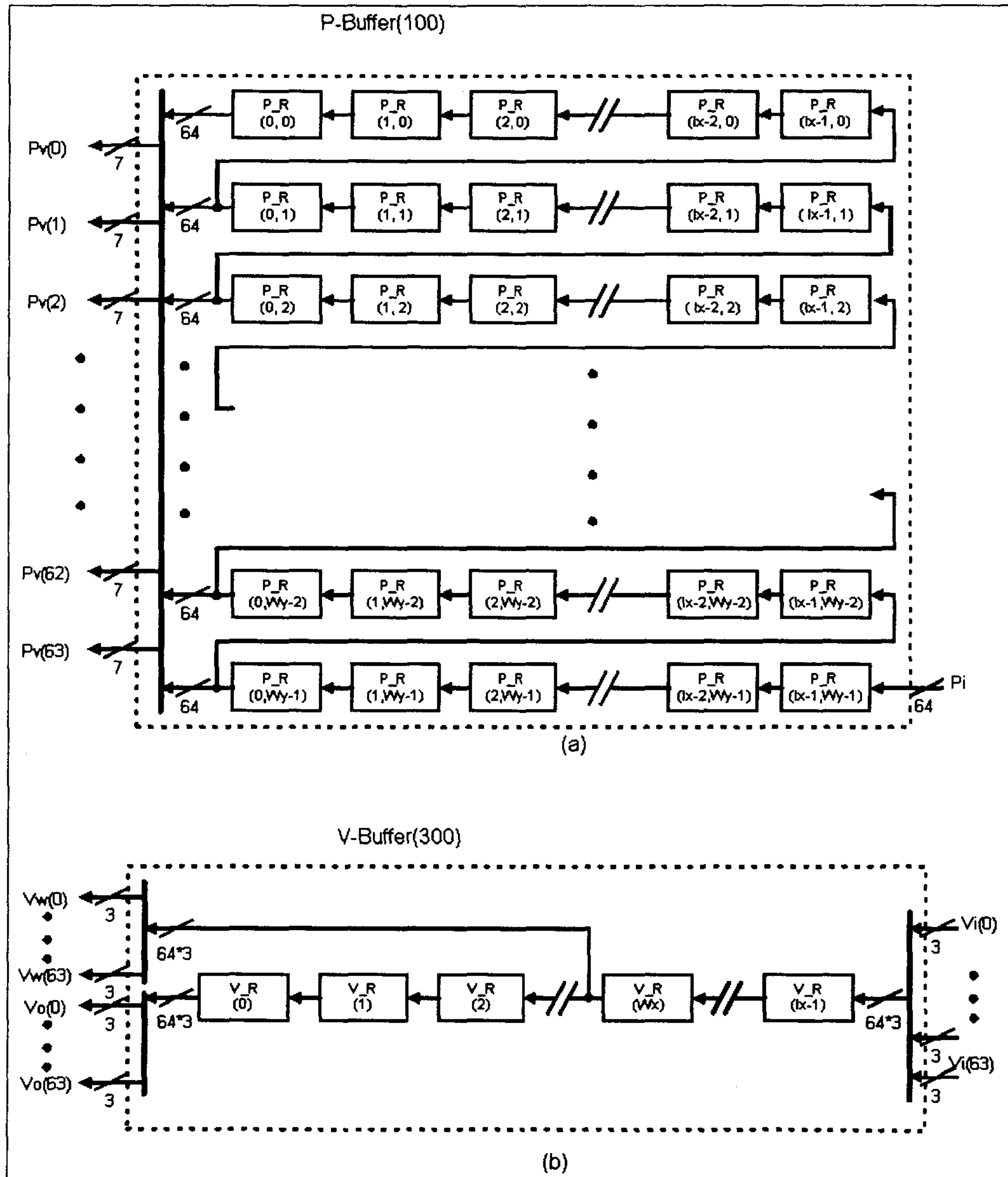


그림 3.21 P 버퍼와 V 버퍼의 구조도. (a) P 버퍼 구조도. (b) V 버퍼 구조도.

그림 3.22의 (a)는 그림 3.20의 MPC-Processor 중 P-UNIT의 구조를 보여주고 있다. P-UNIT의 입력은 R, S 와 Th가 있는데 Th는 $P(x,y,d)$ 값을 계산하기 위한 문턱값이다. R과 S는 각각 기준이 되는 우 영상과 탐색에 사용되는 좌 영상의 화소값이 들어오는 부분이다. 그림에서는 화소값이 $0 \sim 255(2^8)$ 인 영상을 가정하였기 때문에 R과 S가 8 bit씩 할당되어 있다. P-UNIT에서는 기준 영상 한 화소에 대해서 탐색 영상에서 탐색 범위(Sr)+1 개 만큼의 화소를 받아 식 3.6에 주어진 $P(x,y,d)$ 를 계산한다. D_P(d)로 표시된 처리부에서 기준영상 한 화소와 탐색영상에서 d 만큼 떨어진 화소에 대해 $P(x,y,d)$ 값을 계산하여 준다.

그림 3.22의 (b)는 D_P 처리부의 자세한 논리 회로를 나타낸다. 그림의 좌측에 있는 덧셈기(Adder)는 R에서 S값을 빼주는 역할을 하고, 우측에 있는 Carry 발생기(Carry Gen.)에서는 문턱값인 T에서 R-S의 절대값을 빼주어 그 때 Carry를 발생시키는 역할을 한다. 이 때 발생하는 Carry D가 1이면 문턱값 T가 앞에서 나온 결과보다 더 크다는 것을 나타낸다. 참고로 가운데에 있는 XOR Gate의 출력은 앞단에서의 결과가 양수일 경우 2의 보수(2's Complement)를 출력하고, 음수일 경우에는 이를 그대로 통과시키게 되어 그림의 우측단에서 빼기가 되게 한다. 이러한 D_P(d) 처리부는 ($Sr+1$)개 만큼 존재하기 때문에 ($Sr+1$)개의 $P(x,y,d)$ 값을 병렬로 계산할 수 있게 된다. P-UNIT에서 기준영상은 한 화소씩 사용되고 버려지는 반면, 탐색영상은 입력된 화소가 shift되면서 ($Sr+1$)번 만큼 계산에 사용된다. D_R(d)로 표시된 블록은 영상의 화소값을 담을 수 있는 register이고, clock에 따라 다음 D_R 블록으로 shift된다. P-UNIT도 Shift register를 사용하기 때문에 초기에 ($Sr+1$) clock 만큼 시간 지연이 있지만, 그 다음부터는 연속적으로 동작하게 되어있다.

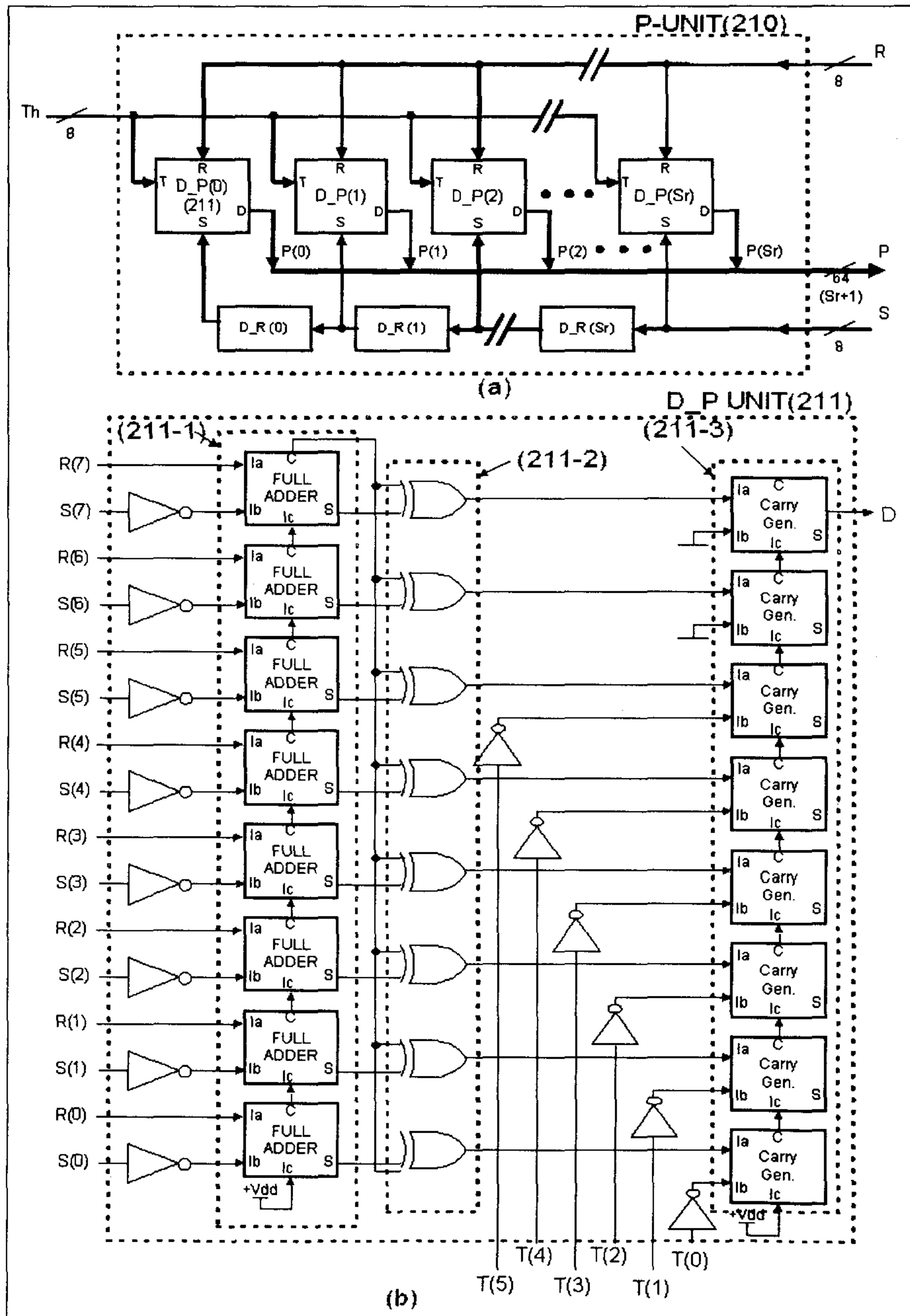


그림 3.22 P-UNIT 구조도 및 회로도. (a) P-UNIT의 구조. (b) D_P UNIT의 회로도.

그림 3.23은 그림 3.20에 나타낸 MPC-Processor의 MPC-UNIT 구조를 보여주고 있다. MPC-UNIT은 기능상 식 3.9로 표시된 $V(x,y,d)$ 를 계산해주는 V_MP Counter 및 V_MP Update 부분과, 식 3.11로 표시된 $MPC(x,y,d)$ 를 계산하는 W_MP Count & Update 로 나눌 수 있다. 이중에서 V_MP Counter는 계산 초기에 입력되는 Pv값을 더해서 $V(x,y,d)$ 값을 얻는데 사용되고, V_MP Update는 초기 일정시간이 지난 후 V-BUFFER로부터 Vi가 입력되기 시작하면 중복계산을 없애고 $V(x,y,d)$ 값을 얻는데 사용된다.

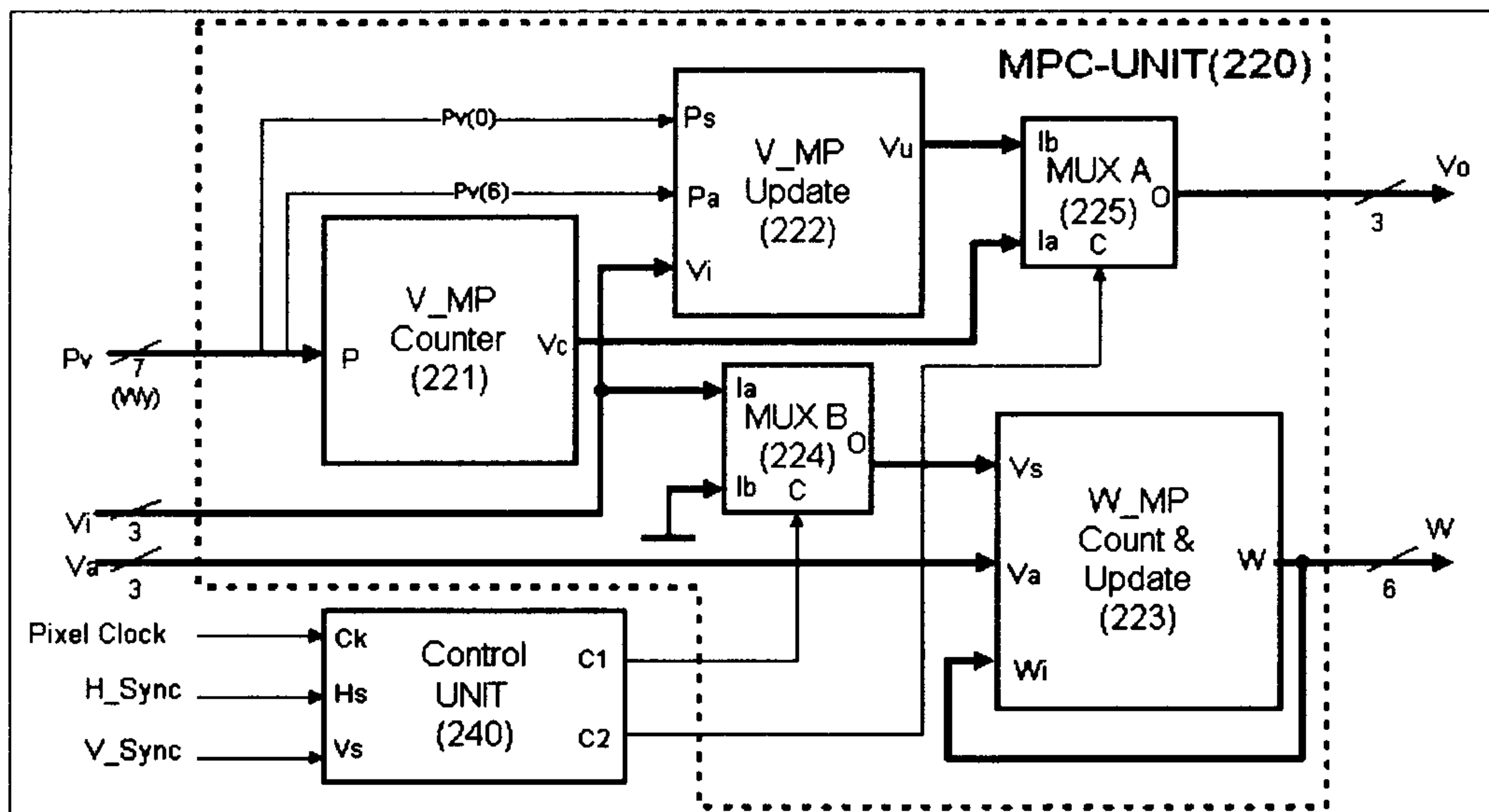


그림 3.23 MPC-UNIT의 구성도.

그림 3.24는 MPC-UNIT의 자세한 논리 회로도를 나타낸다. V_MP Counter 에서는 초기에 P-BUFFER로부터 들어오는 탐색창 높이 만큼의 Pv값을 4개의 전가산기로 더해서 2진수인 $V(x,y,d)$ 를 출력시킨다. 그리고, 이 값은 MUX A를 통과하여 V-BUFFER로 출력된다. MUX A의 C2 제어 입력은 V_MP Update의 출력이 유효해질 때 까지 V_MP Counter의 출력을 선택하게

된다. V-BUFFER에 입력된 $V(x,y,d)$ 값은 I_x clock이 지난 다음부터 출력되어, MPC-UNIT의 V_i 에 입력된다. 그러면, V_MP Update 에서는 $V(x,y,d)$ 를 중복계산을 피해서 계산할 수 있게 $P_v(0)$ 와 $P_v(6)$ 의 상태에 따라 논리 게이트를 사용하여 전가산기에 1을 더하거나 빼게 된다. 그렇게 되면, MUX A의 C2 제어 입력은 V_MP Update의 출력을 선택한다.

한편, V-BUFFER에서 MPC-UNIT으로 입력된 V_i , V_a 는 식 3.12에서와 같이 MPC 값을 계산하는데 사용된다. W_MP Count & Update 의 좌측에 있는 가산기에서는 $V_i - V_a$ 를 계산하고 이 값은 우측에 있는 가산기에서 먼저 있던 MPC 값에 더해져서 새로운 MPC를 계산하게 된다. MUX B에서는 V-BUFFER 에서 출력되는 V_i 가 유효한 값을 가질 때 까지 0을 입력시키도록 선택해준다.

그림 3.15은 MPC-Processor의 Max_MPC Select 구조를 나타낸다. Max_MPC Select 는 $(S_r + 1)$ 개의 MPC-UNIT 에서 나온 각 변위에 대한 MPC 값들을 비교하여 최대의 값을 선택하고, 그 때의 변위값을 출력하는 부분이다. Max_MPC Select 구성의 기본을 이루고 있는 것은 C&A(n) Cell로 이것은 입력을 비교하여 큰 값을 출력하고 그것의 위치를 만들어주는 부분이다.

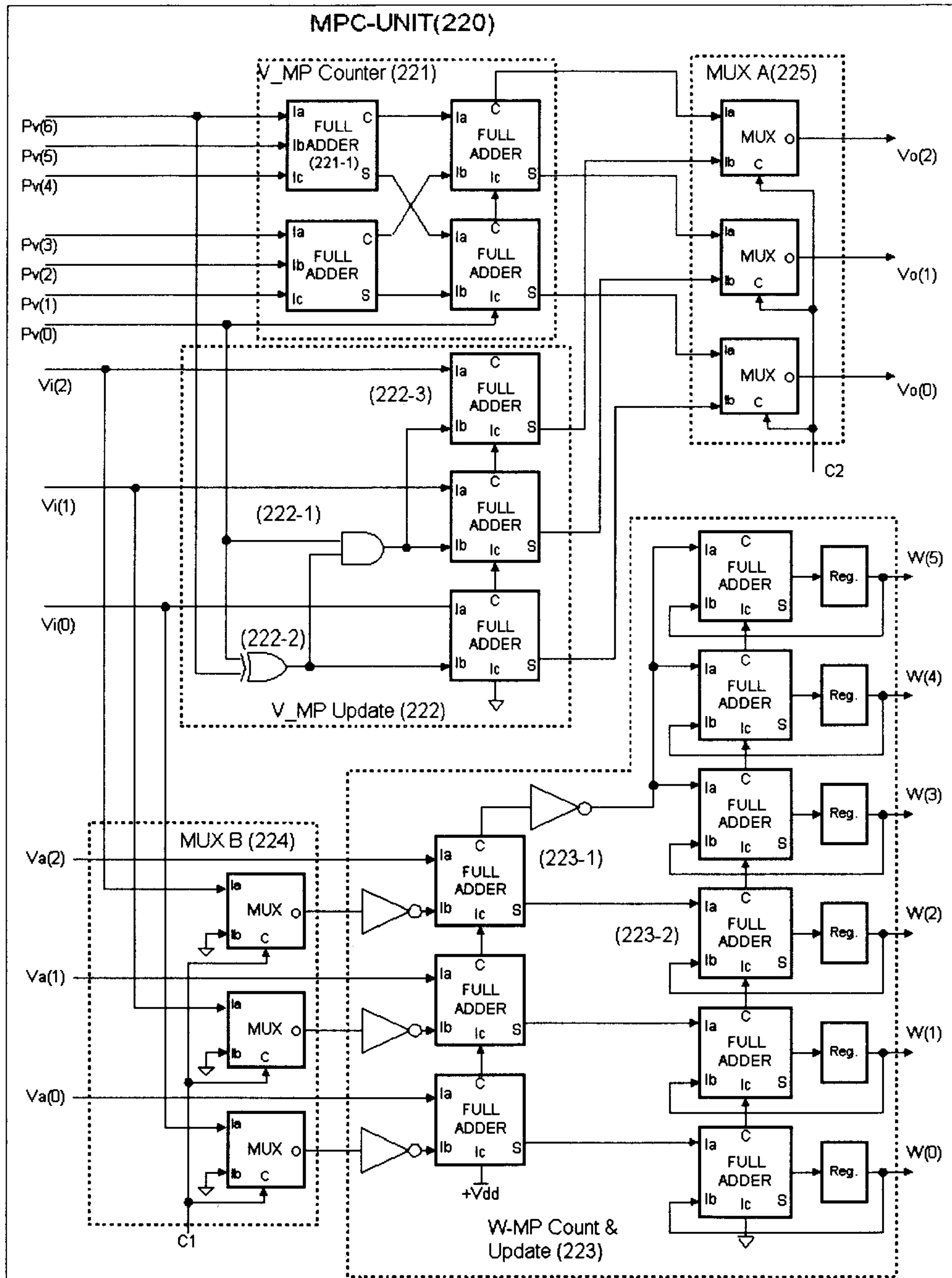


그림 3.24 MPC-UNIT의 논리 회로도.

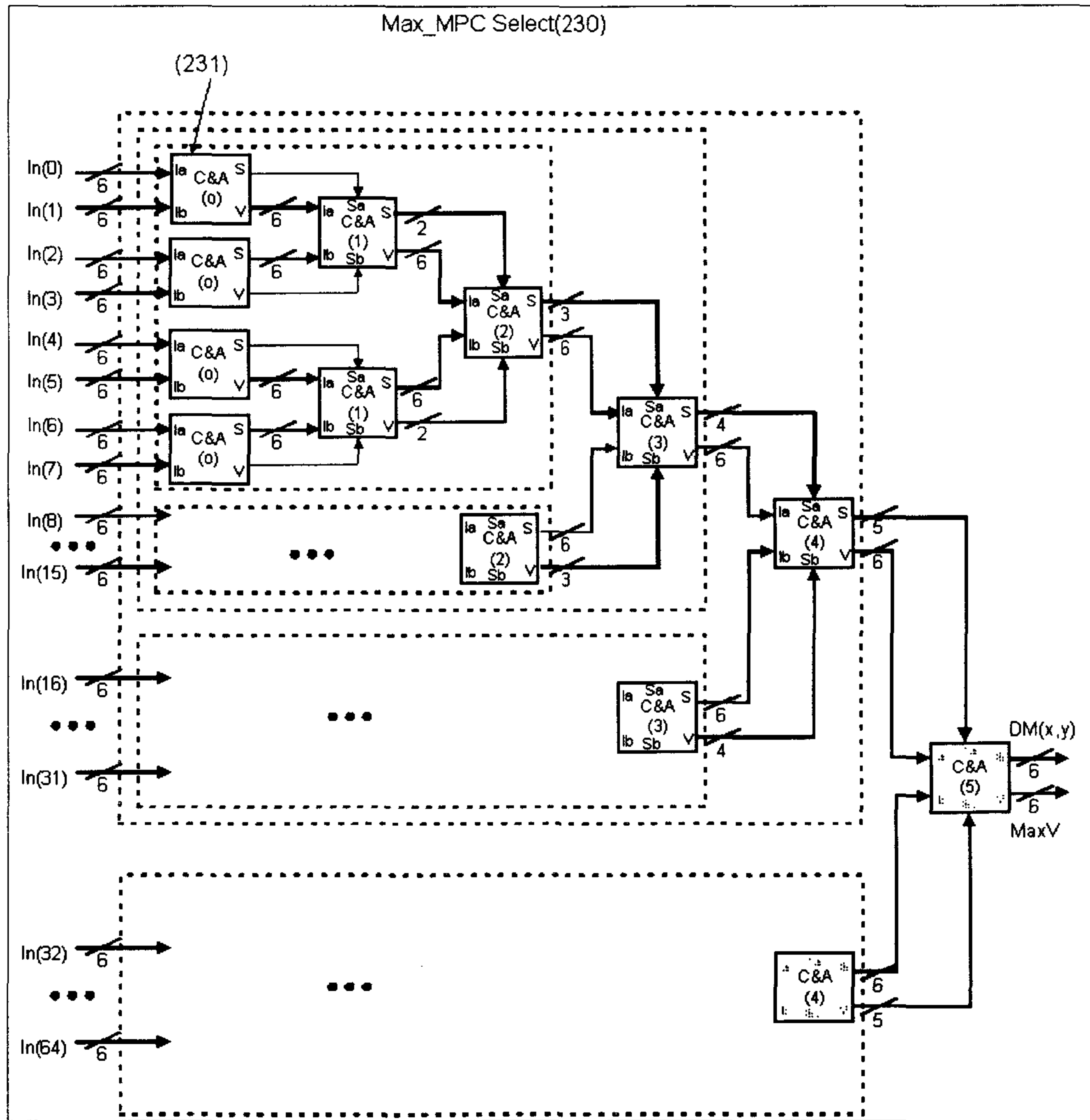


그림 3.25 Max_MPC Select의 구성도.

그림 3.26은 C&A(n) Cell의 논리회로를 나타낸다. C&A(n) Cell 에서 Carry Gen.은 Carry로 Ib가 Ia보다 작으면 0을, 아니면 1을 발생시켜서 우측 열에 있는 MUX에서 큰값을 선택하여 출력하도록 한다. 또한, 최대 값의 위치 정보를 알 아내기 위해서 Carry와 Carry Gen. 위쪽에 있는 MUX 들을 사용하게 되는데, C&A(n) Cell의 구조는 n값에 따라서 사용되는 MUX의 개수가 바뀌게 된다. 그림에서 Carry Gen. 위쪽에 있는 MUX(231-2)의 개수가 n개가 됨을 알 수 있다. 이것은 입력으로 들어온 앞단의 위치정보 Sa, Sb중에서 원하는 것을 선택하기 위한 것이다. 이는 그림에서 볼 수 있는데 오른쪽으로 갈수록 C&A(n) Cell에 입력 되는 Sa, Sb의 bit수가 늘어나는 것을 볼 수 있다. 이렇게 해서 마지막 단에 나오는 MaxV와 DM(x,y)는 기준 영상의 화소 (x,y)에 대한 최대 MPC값과 그 때의 변위를 나타내게 된다.

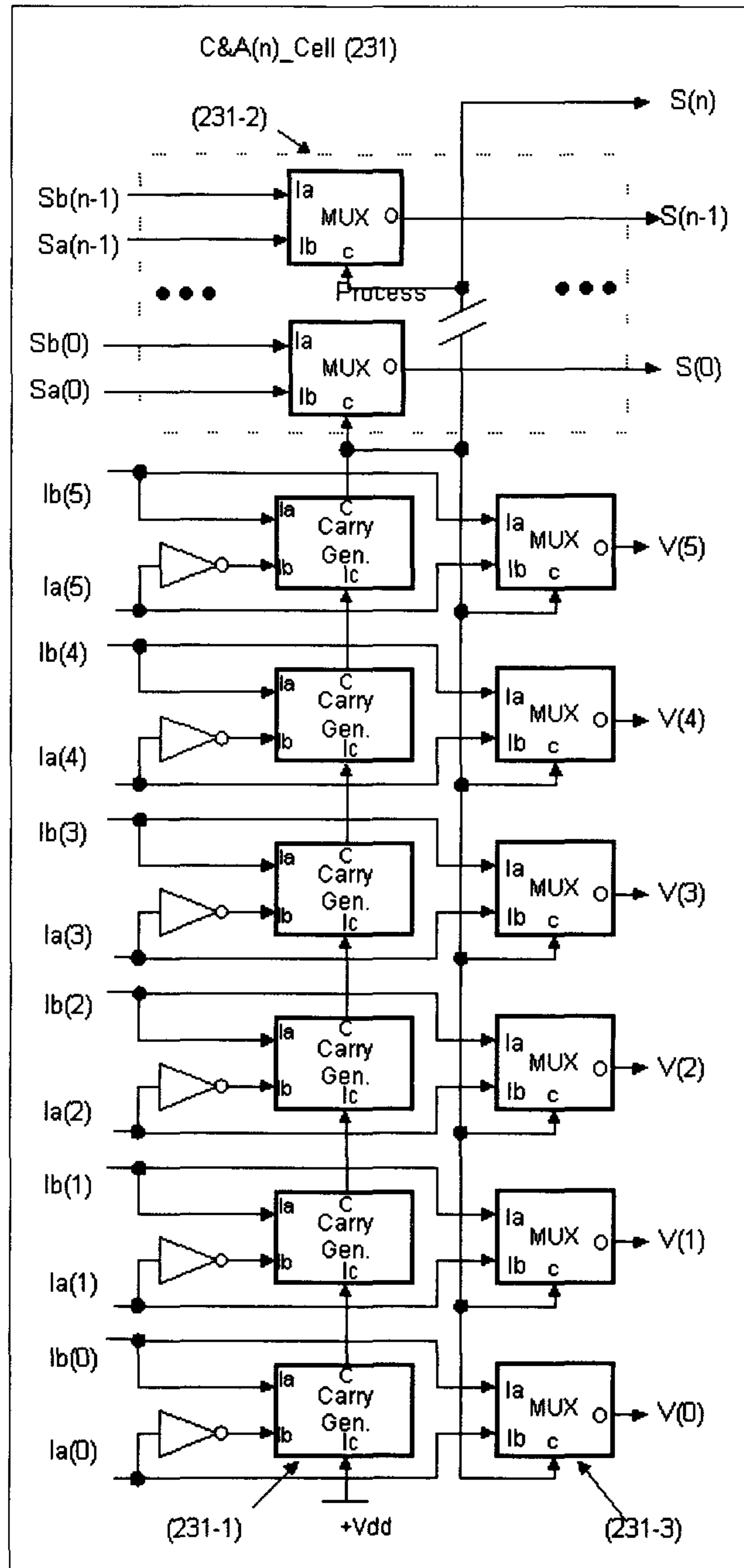


그림 3.26 C&A(n) Cell의 논리 회로도.

여 백

3장 참고문헌

- [1] 최영우, 박래홍, "Stereo vision 및 응용", 전자공학회지, 제 21 권 제 10 호, pp.36-46, 1994년 10월
- [2] S. D. Cochran and G. Medioni, "3-D Surface Description from Binocular Stereo", *IEEE Trans. PAMI*, vol. 14, no. 10, pp.981-994, Oct. 1992.
- [3] T. Kanade et al., "A Stereo Machine for Video-Rate Dense Depth Mapping and Its New Applications", *Proceedings of Computer Vision and Pattern Recognition Conference*, San Francisco, June 1996.
- [4] 박종일, 이노우에 세이키, "여러 대의 카메라를 이용한 계층적 깊이정보 추출 알고리즘", 방송공학회지, 제 2 권 제 1 호, pp.45-56, 1997년 3월
- [5] P. Anandan, "A Computational Framework and an Algorithm for the Measurement of Visual Motion", *International Journal of Computer Vision*, vol. 2, pp.283-310, 1989.
- [6] C. Menard and N. Brandle, "Hierarchical Area-Based Stereo Algorithm for 3D Acquisition", *International Workshop on Stereoscopic and Three Dimensional Imaging*, pp.195-201, Sep. 1995.
- [7] U. R. Dhond and J. K. Aggarwal, "Structure from Stereo-A Review", *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 6, pp.1489-1510, Nov./Dec. 1989.
- [8] R. Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses", *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 4, pp.323-344, Aug.

1987.

- [9] M. Okotumi and T. Kanade, "A Multiple Baseline Stereo", *IEEE Trans. PAMI*, vol. 15, no. 4, pp.353-363, 1993.
- [10] O. Faugeras et al., "Real time correlation-based stereo : algorithm, implementations and applications", Technical Report, INRIA, 1993.
- [11] H. Gharavi and M. Mills, "Blockmatching Motion Estimation Algorithms-New Results", *IEEE Trans. Circuits Syst.*, vol. 37, no. 5, pp.649-651, May 1990.
- [12] S. T. Barnard and W. B. Thompson, "Disparity Analysis of Images", *IEEE Trans. PAMI*, vol. 2, no. 4, pp.333-340, July 1980.
- [13] J. S. Lee, "Digital Image Smoothing and the Sigma Filter", *Comput. Graphics Image Processing*, vol. 24, pp.255-269, 1983.
- [14] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, 1982.

제 4 장 실시간 얼굴 검출 및 추적 시스템

최근 들어 얼굴 정보에 대한 관심이 증대되고 있다. 얼굴 정보는 범죄자 검색 시스템, 개인용 컴퓨터의 접근 제어, 공공 기관에서의 보안 시스템, 컴퓨터 게임, 3D TV의 시청자 검색 시스템 등의 분야에서 널리 이용되는 기술이다. 특히 최근 하드웨어 기술의 발달로 컴퓨터 기술이 보편화되고 컴퓨터 그래픽스 등의 기술이 발달하면서 보다 친숙하고 현실감 있는 사용자 인터페이스에 관한 요구가 증대되고 있어 얼굴 검출 및 얼굴 특징점에 대한 관심이 증대되고 있는 상황이다[1]. 얼굴 정보는 크게 얼굴 영역 검출, 얼굴 특징점 추출, 그리고 얼굴 분류 등의 세 가지 기술로 크게 나눌 수 있다[1]. 얼굴 분류기술은 Eigen face[2], SVD, 신경망[6], Garbor wavelet을 사용한 graph matching방법[17]등 여러 각도에서 연구가 활발히 진행되고 있다. 하지만 얼굴 분류 기술의 전처리 단계라 할 수 있는 얼굴 영역 검출과 얼굴 특징점 추출은 연구가 초기 단계에 머무르고 있어 대상이 되는 영상이 단순한 배경을 가지고 있어야 하며 영상에서 얼굴이 차지하는 크기나 얼굴의 방향 등에 대한 제약을 가지게 되었다[1]. 보다 폭넓은 응용을 위해 얼굴 분류기술의 전처리 단계라 할 수 있는 얼굴 영역 검출과 특징점 추출 기술은 앞으로 활발히 연구가 진행되어야 할 분야이다.

얼굴 영역을 검출하는 데 사용될 수 있는 정보는 크게 모양 정보, 색상 정보, 움직임 정보 등으로 나눌 수 있다[1]. 모양 정보는 명암 영상에서 주로 에지(edge)나 명암의 편차 등을 추출하고 얼굴의 사전 지식에 적용하여 얼굴 영역을 추출하는 데 사용된다. 모양 정보를 이용해서 얼굴 영역을 검출하는 방법에는 모자이크를 사용하는 방법[5], 신경망을 사용하는 방법[7], K-L변환을 이용하는 방법[15]등이 얼굴 영역 검출 방법으로

제안되었다. 하지만 이들 방법들은 얼굴이 거의 정면인 경우 적용이 가능하고 얼굴의 크기와 방향, 얼굴이외의 배경에 대해 결과가 많은 영향을 받는 단점을 가지고 있다. 색상 정보는 인간의 얼굴이 가지는 고유의 색상을 이용하는 것으로 인종이나 개인마다 편차를 가지고 있고 영상이 획득되는 하드웨어에 따라 변화가 심하다는 한계를 가지고 있다. 하지만 명암 영상을 사용하는 방법에 비해서 많은 정보량을 가지고 있어 최근 하드웨어의 발달과 함께 이용 가치가 높아지고 있다. 움직임 정보는 동영상에서 물체의 움직임으로 인한 정보를 사용하는 것으로 응용에 따라 유용하게 사용된다.

기존의 얼굴 특징점 추출 방법으로는 템플릿 매칭, 변형가능 템플릿 매칭[3], 대칭 변환[4]등이 있다. 변형가능 템플릿 매칭은 기존의 템플릿 매칭이 갖는 대상 물체의 크기나 방향, 개인적인 편차 등을 극복할 수 있으나 계산시간이 많이 걸리고 초기 설정지역을 제대로 지정하지 못하면 올바른 결과를 얻기 어려운 단점이 있어 응용이 제한적이다. 대칭변환의 경우, 정형화된 모델을 가지지 않고서도 눈, 코, 입의 위치 등을 찾을 수 있는 장점이 있으나 눈의 중심점이나 끝점과 같은 정확한 특징점들을 결정하기는 어려우며 얼굴의 크기 변화에 상당히 민감하고 역시 계산 시간이 많이 걸리는 단점을 가지고 있다. 마스크를 이용한 방법은 다른 방법에 비해 고속의 연산이 가능하지만 영상내의 얼굴의 위치나 방향 등에 대한 제한이 많아 넓은 범위의 응용이 불가능하다.

본 장에서는 컬러 비디오 시퀀스 상에서 배경의 영향, 얼굴의 크기와 방향, 표정 등에 덜 민감하고 보다 빠른 연산이 가능한 얼굴 특징점 추출 방법을 제안한다. 색상 정보와 움직임 정보를 복합적으로 사용함으로써 복잡한 배경에서는 물론 얼굴의 크기나 방향, 그리고 조명의 영향에 대해서도 강한 특성을 가지게 된다. 색상 정보는 정규화된 색상모델을 사용하여 조명의 변화에 따른 색상의 변화를 최소화하였다. 그리고 살색을 지니는 배경이나 살색 영역이외의 움직임은 영역을 제거하기 위해 살색분포를 이용한 색상 변환된 영상에서 움직임을 효율적으로 강조하며, 움직임의 민감도를 살색의 가능

성 정도에 따라 변화하도록 적응적 문턱값을 가지는 움직임 정보를 사용한다. 이 방법을 사용함으로써 특정 색상을 가지는 움직임 영역만을 민감하게 추출할 수 있다. 색상 움직임 정보를 갖는 영역에 대해 Top-hat 변환을 하고 눈의 색상 분포를 이용해 눈 후보영역을 찾은 후 최종적인 눈 영역은 눈과 눈썹의 지정학적인 관계를 후보 영역에 적용하여 결정한다. 입의 색상 분포는 개인적인 편차가 심하기 때문에 눈 영역의 위치를 이용한 입 후보 영역을 추출하고 이의 히스토그램을 이용해 정확한 입 영역을 추출한다. 그리고 정확한 눈과 입의 특징점들을 추출하기 위해 Principal component analysis를 사용하여 대칭축과 특징점들을 구한다. 제 1 절에서는 색상 움직임 정보 알고리즘에 대해 설명하고 제 2 절에서는 움직임 색상 정보 알고리즘을 이용한 얼굴 특징점 추출 알고리즘에 대해 설명한다. 제 3절에서는 제안된 알고리즘을 고성능 DSP 처리기를 이용한 실시간 구현에 관하여 설명하고 마지막으로 결론을 제 4절에서 논한다.

1절. 색상 움직임을 이용한 얼굴 영역 검출

가. 얼굴 색상 정보와 색상 변환

a. 색상 정보의 정규화

영상처리 분야에서 색상은 영상에서 관심의 대상이 되는 물체의 영역을 추출해 낼 수 있는 정보로 중요한 역할을 한다. 일반적으로 색상은 약 400nm에서 700nm까지의 파장을 지닌 가시광선 영역의 전자기파를 의미하는 것이다. 모든 색상은 소위 주요 컬러(RGB)라는 서로 다른 파장의 길이를 가지는 세 가지 성분의 조합과 빛의 휘도 성분으로 표현된다. 인간의 망막으로 받아들이는 위와 같은 세 가지의 주요 컬러 성분은 서로 독립적인 컬러 매칭 함수와 적절한 스펙트럼 함수로서 표현된다.

$$\begin{aligned}
R &= \int_{v_1}^{v_2} r(v)U(v)dv \\
G &= \int_{v_1}^{v_2} g(v)U(v)dv \\
B &= \int_{v_1}^{v_2} b(v)U(v)dv
\end{aligned}
\tag{4.1}$$

여기서 v 는 빛의 주파수이다.

대부분 사용되는 하드웨어의 종류에 따라 여러 가지 컬러 모델을 사용한다. 일반적으로 비디오 카메라는 RGB모델을 가장 많이 사용하고 있으며 이 모델은 다른 컬러 모델로 쉽게 변환이 가능하다. 비디오 카메라나 프레임 그레버는 아날로그 신호를 각각의 화소에 대해 세 개의 $[0,1,\dots,255]$ 의 값으로 변환한다. 즉, 한 화소는 각각 $[0,1,\dots,255]$ 값을 가지는 RGB의 조합 $Q=(R, G, B)$ 으로 표현된다. 하지만 이 모델은 하나의 색상뿐만 아니라 빛의 휘도 까지 포함된 값이다. 정규화 색상모델은 전체 분포에서 빛의 휘도 성분을 제외시킨 값이다. 그림 4.1(a)는 정규화 색상모델의 개념을 설명한다.

RGB공간상에서의 한 점 $Q=(R, G, B)$ 는 색상뿐만 아니라 휘도를 지니고 있다. 만약, 공간의 두 점 $Q_1=[r_1, g_1, b_1]$ 와 $Q_2=[r_2, g_2, b_2]$ 이 식 4.2와 같은 관계를 지니고 있다면

$$\frac{r_1}{r_2} = \frac{g_1}{g_2} = \frac{b_1}{b_2}
\tag{4.2}$$

이 두 점은 같은 색상을 가지고 있으나 다른 휘도값을 가지고 있다는 것을 의미한다. 인간의 시각 체계는 여러 가지 다른 밝기와 환경에 의한 조명의 변화에 대해 색상을 구분할 수 능력을 지니고 있으며 순수한 하나의 색상을 표현할 때 휘도는 적은 영향을 미친다. 그리고, 조명의 변화에 색상은 영향을 받지 않으나 휘도는 큰 변화를 가지게

된다. 따라서, RGB 공간상의 한 점에 대해 정확한 색상을 유지하면서 조명의 변화를 제거할 수 있는 정규화 과정을 거칠 필요가 있다.

$$\begin{aligned} r &= \frac{R}{R+G+B} \\ g &= \frac{G}{R+G+B} \\ b &= \frac{B}{R+G+B} \end{aligned} \quad 4.3$$

RGB모델의 3차원 공간상의 점들이 위와 같은 정규화과정을 거치게 되면 순수한 컬러 정보만을 지니게 된다.

$$r + g + b = 1 \quad 4.4$$

그리고, 식 4.4와 같은 관계를 가지게 되어 세 개의 요소 중 두 개의 요소로 컬러를 표현할 수 있다. 즉 $R^3 \Rightarrow R^2$ 의 매핑 관계로 차원을 낮출 수 있어 계산상에서도 효율적이다. 정규화 과정을 거친 색상모델에서는 화소에 대한 색상정보를 $q = (r, g)$ 로 나타낼 수 있으며, 이는 각 색상 성분이 8비트로 나타내 진다면 기존 RGB모델에서 표현 가능한 256^3 가지의 색상 정보중 휘도의 변화에 덜 민감한 $255^2/2$ 개의 색상정보를 정규화 색상모델에서 표현할 수 있음을 의미한다. 또한 이 정규화 과정은 인간의 시각체계를 잘 표현한다는 HSI모델과 밀접한 관련을 가지고 있다. 그림 4.1(b)에서 삼각형 위의 한 점에 대해 Hue 정보는 삼각형의 중심에서 R을 이은 직선에 대해 이루는 각도를 의미하며 Saturation 정보는 삼각형의 중심과의 거리를 의미한다. 그리고 밝기는 그림 4.1(b)에서 중간에 위치한 삼각형에 대해 직교상의 거리를 의미한다.

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)]$$

$$I = \frac{1}{3}(R+G+B)$$

즉 정규화과정을 거친 RGB모델은 HSI모델에서 밝기성분을 제거하고 Hue와 Saturation성분을 구좌표계에서 직교좌표계로 나타냈다고 할 수 있다. 위의 식에서도 알 수 있듯이 HSI모델보다는 정규화를 거친 RGB모델이 더 수식적으로 간단하고 계산상에서도 효율적이다.

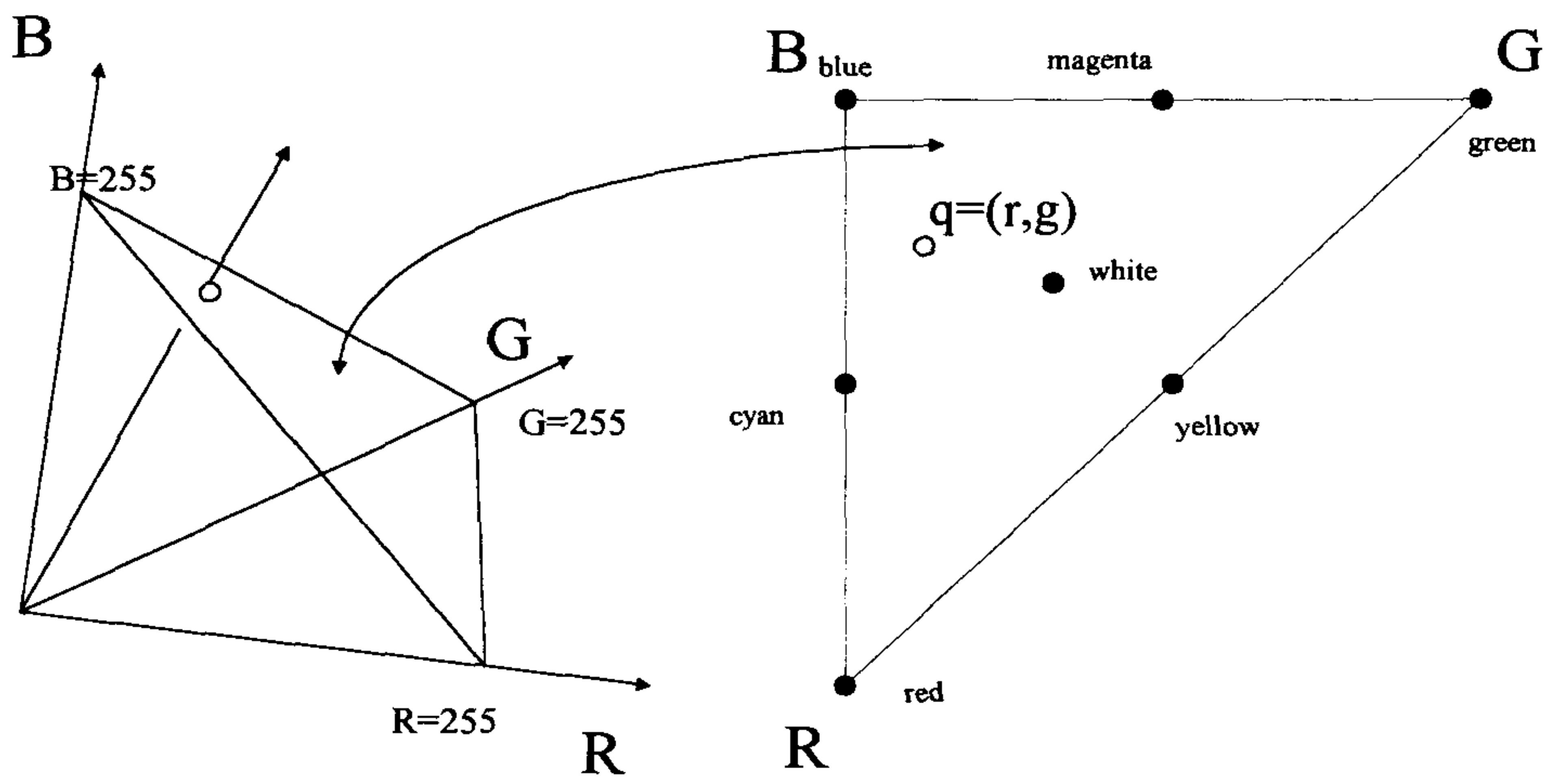
b. 얼굴 색상 정보의 정규화

색상 정보는 여러 영상에서 얼굴 영역을 추출하는데 중요한 정보로 사용될 수 있다. 하지만, 얼굴의 색을 특징으로 사용할 때에는 다음과 같은 문제점을 가질 수 있다.

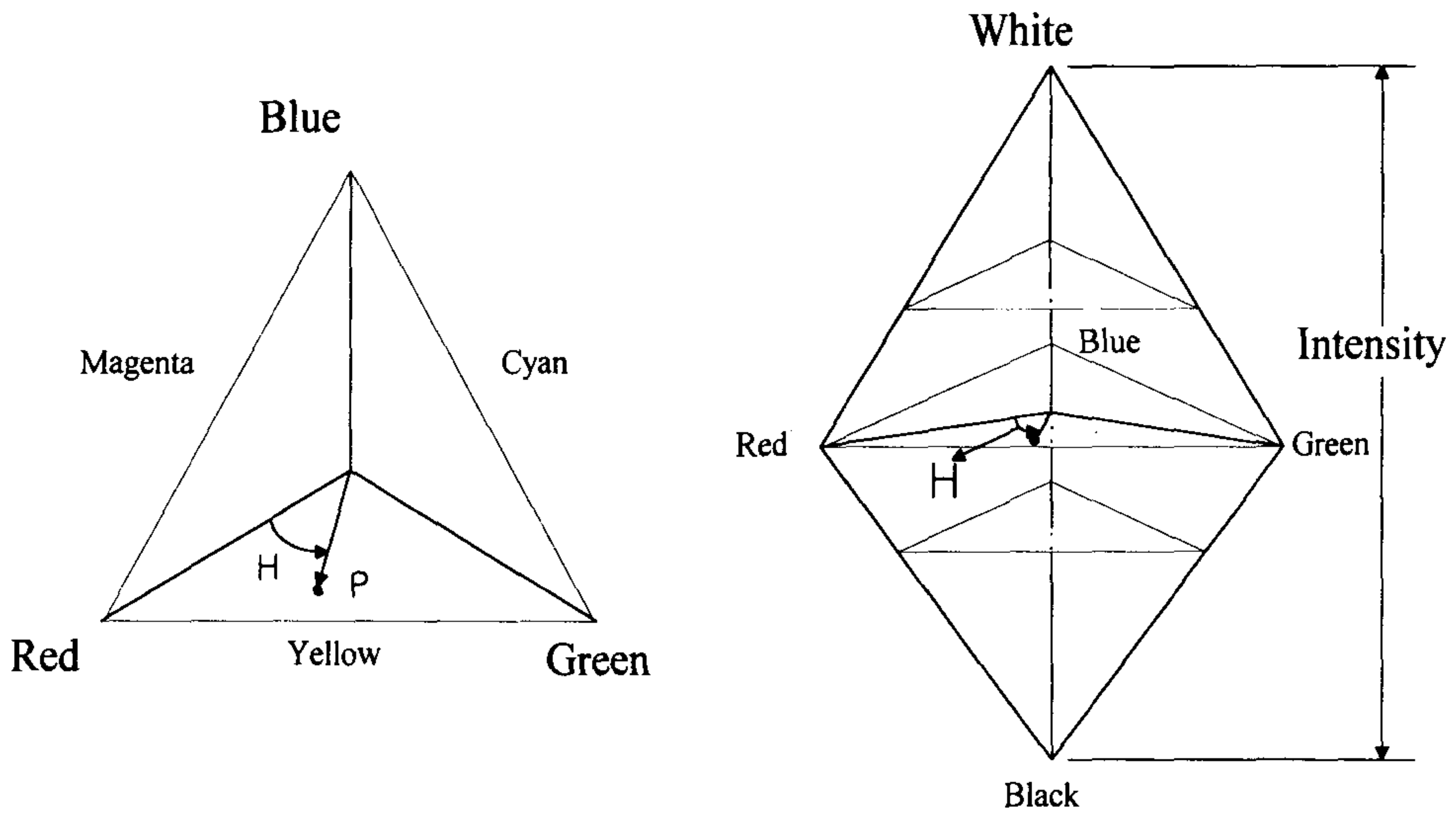
① 하드웨어의 차이에 따라 색의 분포가 달라질 수 있다. 같은 장소, 같은 조명, 같은 사람이라 할지라도 어떤 카메라를 사용했는가에 따라 다르게 나타나며 특히 조명에 따라 얼굴색 분포에 변화가 발생할 수 있다. 야외조명과 실내조명등과 같은 차이로 인해 얼굴의 색의 분포는 영향을 받는다.

② 인종, 성별, 나이 등에 따라 얼굴색의 분포는 개인마다 약간의 차이를 나타낸다. 그리고 같은 사람이라 할지라도 계절 등의 변화, 감정의 변화, 건강 상태 등의 변화로 인해 얼굴 색의 분포가 달리 나타날 수 있다.

이와 같은 문제점들을 해결하기 위해 정규화된 RGB모델을 사용한다.



(a)



(b)

그림 4.1 정규화 색상 모델의 개념 (a) 과 정규화 색상모델과 HSI모델과의 관계(b)

정규화된 RGB모델은 전체 분포에서 빛의 휘도 성분을 제외시킨 값이다. 이런 이유로

하드웨어의 차이로 인한 조명의 변화나 지역적으로 어두워지는 부분, 개인적인 얼굴 색의 변화에 대해 강한 특성을 가진다.

정규화 색상모델에서 (r, g) 성분에서 따른 얼굴 영역의 색상 분포 히스토그램은 좁은 영역에 집중하게 되며, 개인이나 환경에 상관없이 그 분포가 규칙적인 모양을 가지게 된다[8]. 그림 4.2 (a)~(f)는 서로 다른 시간과 환경에서 취득된 영상에서 얼굴색을 갖는 영역만을 추출한 영상이다. 이러한 영역에 해당하는 화소들의 색상 분포 특성은 정규화된 색상의 평균값 $m = (\bar{r}, \bar{g})$ 및 각 성분 방향으로의 분산값으로 나타낼 수 있으며 실제 측정된 값들을 요약하면 표 4.1과 같은 결과를 얻을 수 있다. 정규화 색상모델을 사용하는 경우 얼굴 영역의 색상 분포인 히스토그램은 2차원 Gaussian 분포와 같은 모양을 나타내며 개인적인 편차가 심하게 나타나지 않는다. 다만 같은 사람이라 할지라도 조명이나 하드웨어의 변화에 따라 r 과 g 의 평균값이 약간씩 다른 값을 갖게 된다. 예를 들어 형광등의 실내조명보다 햇빛의 실외조명은 좀더 높은 파란색 값을 가지고 있어 히스토그램의 분포가 파란색 쪽으로 약간 이동함을 알 수 있다.

표 4.1 그림 4.2 얼굴 영역에서의 색상 분포

	\bar{r}	\bar{g}	σ_r	σ_g
(a)	101.43	92.85	20.43	14.2
(b)	99.6	96.64	23.23	17.2
(c)	100.79	93.31	18.45	11.36
(d)	98.53	94.07	22.72	14.53
(e)	100.2	92.9	17.37	11.77
(f)	100.1	93.95	20.41	14.2

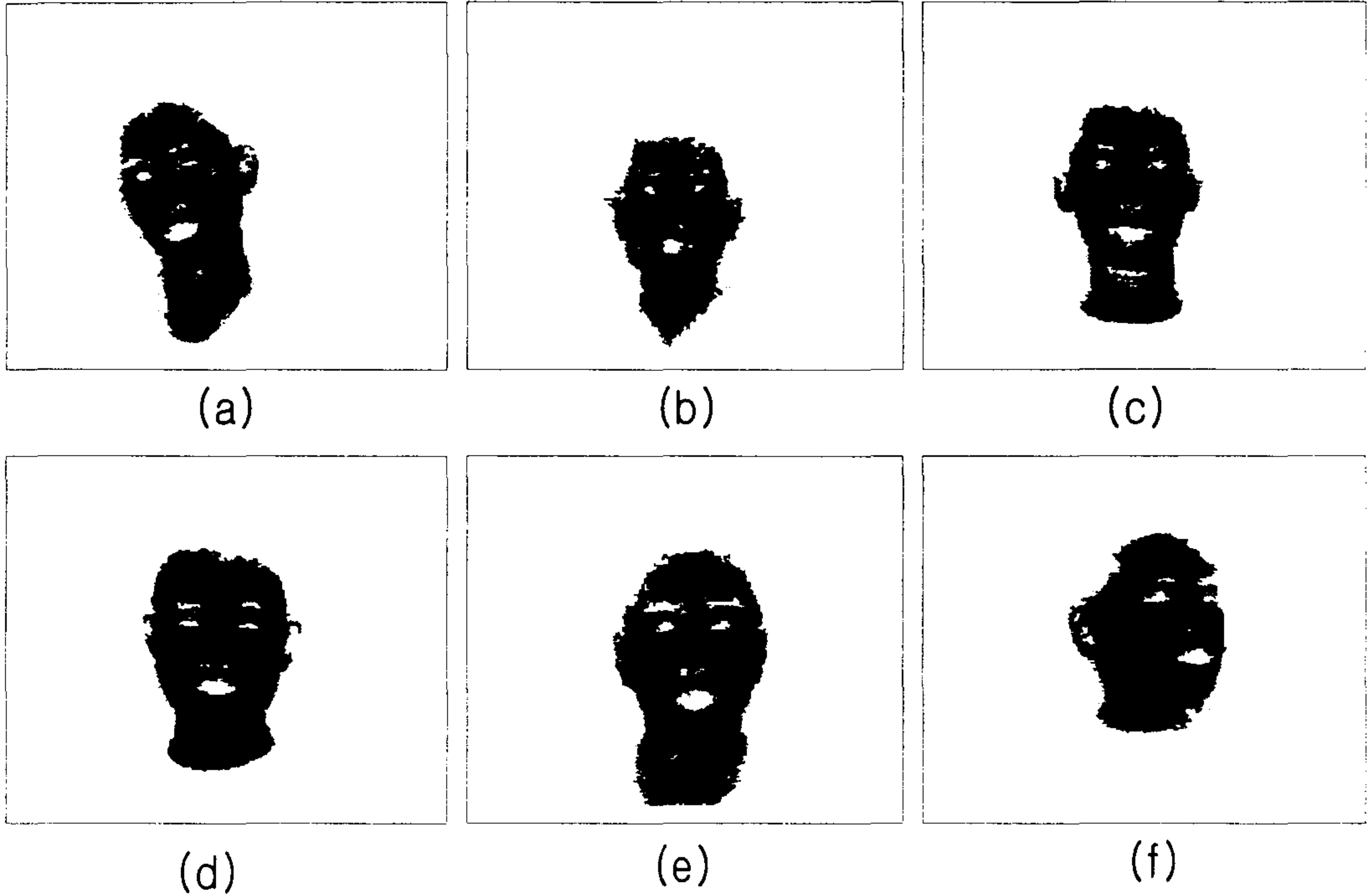


그림 4.2 얼굴 색상 분포의 모델링을 위한 얼굴 영역

c. 얼굴 색상 분포의 2D Gaussian 모델링

위와 같은 개인적인 색상 편차나 조명에 대한 영향을 받지 않기 위해 조명이나 카메라의 변화를 포함하는 넓은 범위의 분산값을 가지는 2차원 Gaussian 분포로 얼굴 영역의 색상 분포를 정의하며 이를 일반적 얼굴색 분포 GFCD (Generalized Face Color Distribution)라 정의한다. 2차원 Gaussian 분포는 $G(m, \Sigma^2)$ 로 나타낼 수 있다. 여기서 $m = (\bar{r}, \bar{g})$ 은 2차원 Gaussian 함수의 중앙을 나타내며 \bar{r} 와 \bar{g} 는 다음 식으로 주어진다.

$$\bar{r} = \frac{1}{N} \sum_{i=1}^N r_i, \quad \bar{g} = \frac{1}{N} \sum_{i=1}^N g_i$$

여기서 r_i 및 g_i 는 각각 얼굴 영역내에 존재하는 화소들의 정규화 색상성분 값들을 나타낸다. 또한 Σ^2 은 각 정규화 색상 성분의 분산을 나타내는 covariance matrix를 의미하며 다음의 식으로 주어진다.

$$\Sigma = \begin{bmatrix} \sigma_r & 0 \\ 0 & \sigma_g \end{bmatrix}$$

여기서 σ_r 및 σ_g 는 정규화 색상 성분의 표준편차를 나타낸다. 본 보고서에서는 여러 가지 실험결과 $m = (105, 95)$, $\sigma_r = 20$, $\sigma_g = 15$ 을 사용하여 GFCD를 정의하였다. 이와 같은 얼굴색상 분포의 초기 설정값은 여러 가지 환경의 변화를 극복하면서 동시에 개인적인 얼굴 색상의 변화에 잘 적응하기 위한 것이다.

얼굴 영역내의 눈과 입과 같은 얼굴 요소들은 얼굴색과는 다른 색상 분포를 나타낸다. 그림 4.3은 옆에 나타난 얼굴 영역과 눈, 눈썹 에 해당하는 영역에 대한 색상 분포 히스토그램을 정규화된 색상 공간에 나타낸 그림이다. 동양인의 눈은 일반적으로 검은 색과 흰색으로 구성되며, 정규화 과정을 거친 색상 영역에서 검은 색과 흰색은 같은 r과 g의 값을 갖는다. 즉, 검은 색은 R, G, B가 모두 0 에 가까운 값에 해당하고 흰색은 R, G, B가 모두 255에 가까운 영역에 분포하므로 정규화 과정을 거친 r과 g 는 모두 1/3 에 해당하는 값에 분포하는 것이다. 이와 같이 얼굴영역 내부의 얼굴요소중 흰색과 검은 색을 갖는 영역의 색상 분포는 살색과 마찬가지로 2차원 Gaussian 함수로 모델링이 가능하며 이를 BWCD(Black & White Color Distribution)이라 정의한다. 본 보고서에서는 여러 가지 실험결과 $m = (85, 85)$, $\sigma_r = 6$, $\sigma_g = 6$ 을 사용하여 BWCD를 정의하였다. 그림 4.4는 정의된 GFCD 및 BWCD를 정규화된 색상공간에서 나타낸 그림이다. 이와 같이 정규화 과정을 거친 얼굴영역의 색상 분포는 개인에 따라 약간씩의 변화는 생기지만 2차원 Gaussian분포로 나타냈을 때 큰 변화를 나타내지 않아 개인 및 환경에 따른 변화를 극복할 수 있다.

c. 색상분포를 이용한 색상변환(Color Transform)

앞에서 정의된 일반적 얼굴색 분포를 사용하여 입력 컬러 영상 I 를 얼굴색 화소가 강조된 명암 영상 Z 로 변환 할 수 있다. 이 변환은 각 화소의 색상정보를 정규화 색상으로 변환한 후 GFCD 및 BWCD 를 사용하여 얼굴영역 및 얼굴요소들의 위치를 강조하는 영상으로 변환한다. 색상변환은 입력되어진 화소의 정규화 색상공간 위치에서의 원하는 색상의 Gaussian 분포의 중앙에 가까운 정도를 출력 값으로 결정하는 것으로 다음과 같이 정의 될 수 있다.

$$f: R^2 \rightarrow R^1 \quad Z(x, y) = G(g(x, y), r(x, y)) \quad (x, y) \in I \quad 45$$

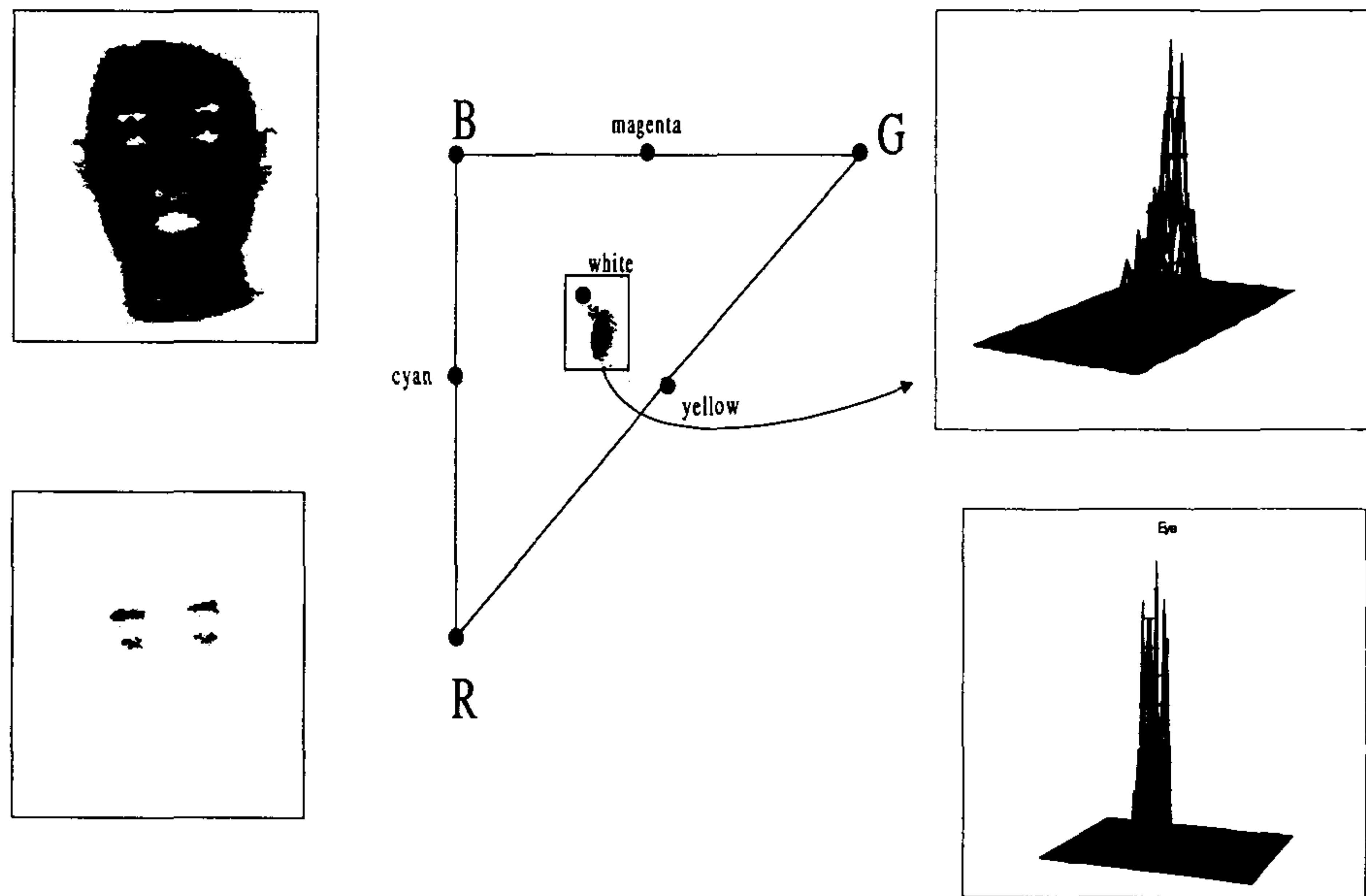


그림 4.3 정규화된 색상공간에서의 얼굴 색상 분포

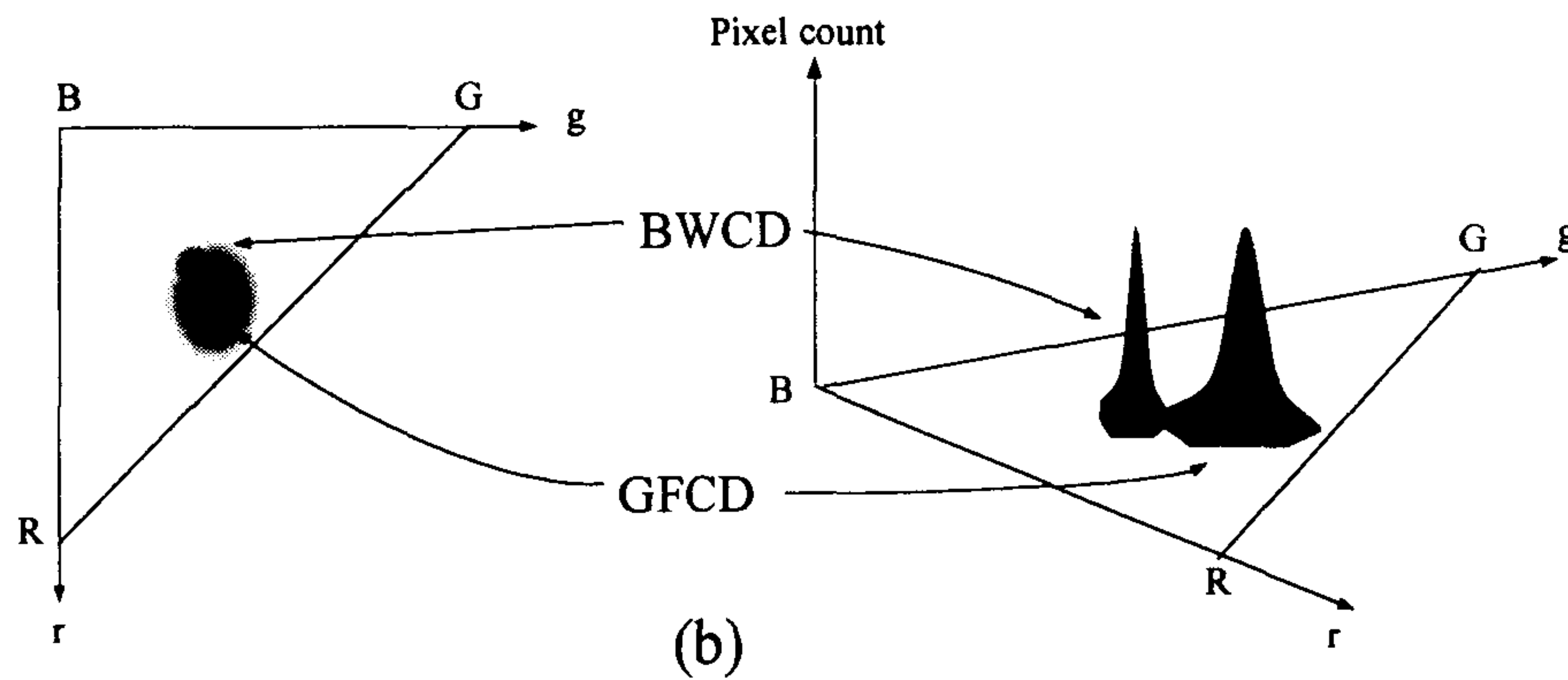


그림 4.4 GFCD와 BWCD의 모델링

여기서의 (x, y) 는 입력 컬러 영상의 좌표이며 $g(x, y)$ 및 $r(x, y)$ 는 해당 좌표에 있는 화소의 정규화 색상성분을 나타낸다. GFCD를 이용한 색상변환 영상은 각 화소의 색상정보를 살색에 가까운 정도를 나타내는 확률 값에 비례한 값을 해당 화소의 밝기 값으로 출력한다. 임의의 컬러 영상을 GFCD로 색상변환 하면 살색에 대한 가능성 정도가 0에서 255로 표현되는 명암 영상으로 표현된다. 그림 4.5 (a),(b),(c)는 얼굴영상들을 GFCD 색상 변환한 결과들로써 살색 영역이 강조된 명암영상 결과를 잘 나타낸다.

GFCD 색상변환을 하여 얻은 영역 내에서 앞절에서 정의된 BWCD를 이용한 색상변환을 적용하면 눈이나 눈썹과 같은 얼굴 요소 영역의 검출이 가능하다. 이 경우 살색 영역내의 검은 머리 영역, 눈과 눈썹, 그밖에 검은 색과 흰색에 해당하는 부분이 높은 밝기 값을 가지는 영상을 얻을 수 있다. 그림 4.5 (d),(e),(f)는 얼굴영상에서 BWCD 색상변환 하여 얻은 결과들을 나타낸다.

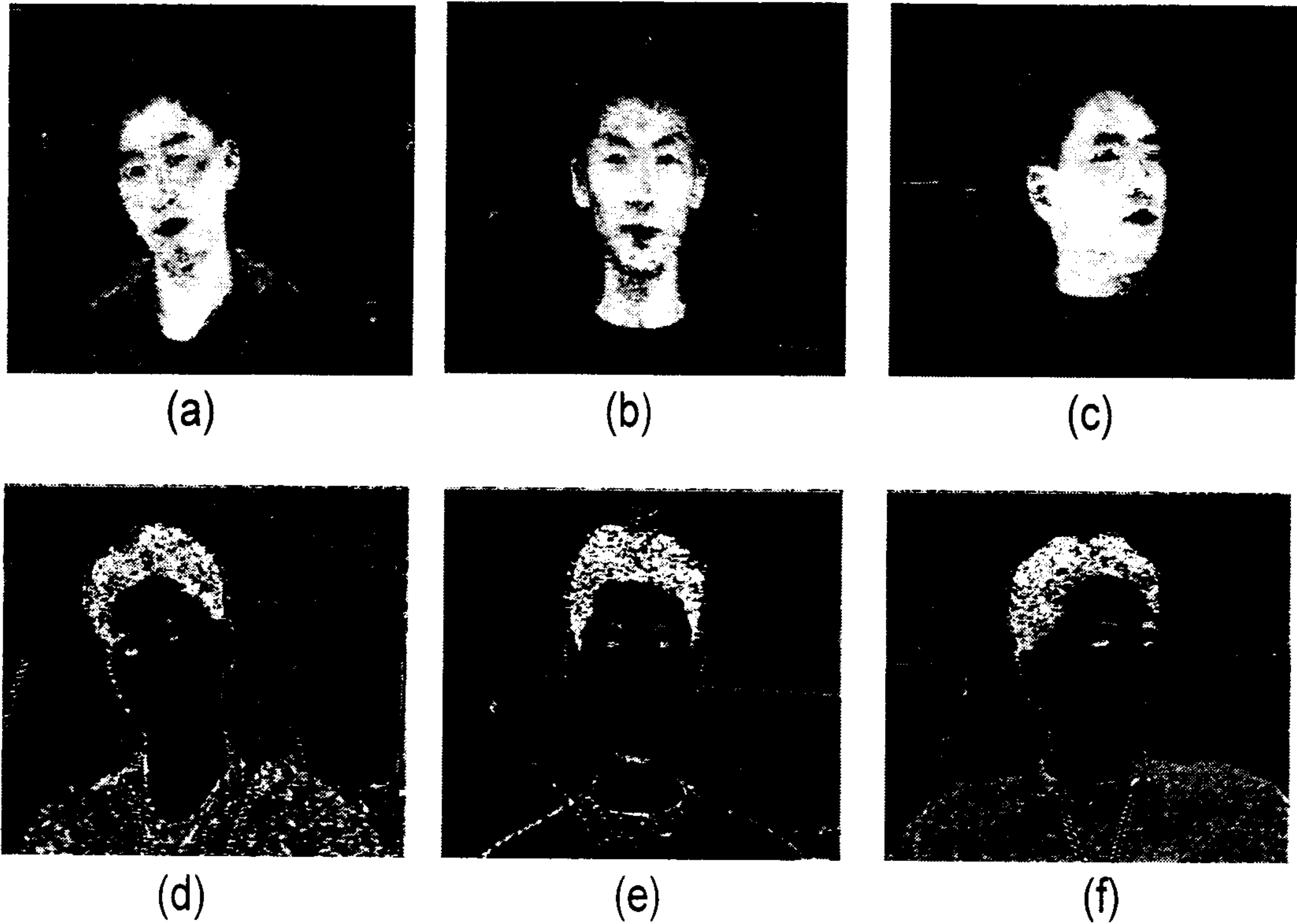


그림 4.5 GFCD 색상 변환(a),(b),(c)와 BWCD 색상 변환 (d),(e),(f) 결과

나. 색상 움직임을 이용한 얼굴 영역 검출

색상분포를 이용한 색상 변환(Color Transform)을 사용하는 경우 배경에 검출하고자 하는 물체와 비슷한 색상을 포함하는 경우 구별이 어려워지는 단점이 있다. 이 장에서는 사람얼굴의 움직임 특성을 반영하여 이러한 단점을 제거하는 알고리즘을 설명한다. 사람의 얼굴은 항상 미세한 움직임을 포함하고 있으므로 색상 변환된 영상에 움직임 정보를 효율적으로 적용하는 방법을 제안하며 이를 색상 움직임 정보라 정의한다.

a. UPC (Unmatched Pixel Count)를 이용한 움직임 검출.

연속적인 두 영상에서 물체의 움직임으로 인한 변화정도를 측정하는 방법에는 영역 기반 방법과 특징점 기반 방법으로 대별된다. 특징점 기반 방법은 비교적 빠른 처리를 보장하나 정밀한 움직임 정보를 얻기 위하여 보간 기법을 이용한 후처리 과정이 필요하게 되어 일반적으로 영역기반 움직임 검출 방법이 선호되고 있다. 영역기반 움직임 검출 기법에서는 일정한 크기의 창을 이용하여 중앙에 위치한 화소의 변화정도를 측정하는데 이웃에 위치한 화소의 밝기 값 차를 누적하여 이용하는 차-누적 측정 (Accumulated Difference Measure) 방법이 있다. 현재 영상의 (x, y) 위치에서의 화소 값을 $I(x, y, t)$ 라하고 이전 화면의 같은 위치의 화소 값을 $I(x, y, t-1)$ 이라 한다면 차-누적 측정 $ADM(x, y, t)$ 는 다음과 같은 식으로 표시된다[14].

$$ADM(x, y, t) = \sum_{i=x-N}^{x+N} \sum_{j=y-N}^{y+N} |I(i, j, t) - I(i, j, t-1)| \quad 4.6$$

여기서 $(2N+1)(2N+1)$ 은 한 화소의 유사도를 측정하기 위해 고려해야 하는 주변 영역을 나타내는 창의 크기이다. 차-누적 측정을 이용하여 한 화소의 움직임 여부를 결정하는 경우 주변 화소들의 차이를 전부 누적하여 최종적으로 움직임 여부를 결정하게 된다. 주위 화소의 변화량도 함께 고려함으로써 변화량을 좀 더 안정적으로 구할 수 있으며, 카메라의 잡음 등에 대해서도 덜 민감한 특성을 가진다. 하지만 화소의 밝기 값 차이를 누적하여 이용하기 때문에 화소 각각의 차이를 평균하는 결과를 나타내게 되어 겹쳐진 영역이나 배경과 밝기 차가 적은 화소에 대해서는 작은 변화량을 가지게 되고 움직임 영역을 결정하기 위한 문턱값 결정에 대해 큰 영향을 받게 된다.

주위 화소 값들의 차이를 누적하여 이용하지 않고 일정한 크기의 창내에서 밝기 차가 어느 정도 이상 되는 화소의 개수를 이용하여 움직임을 검출할 수 있다[12]. 이러한 비정합 화소 개수 (Unmatched Pixel Count)를 이용한 움직임 검출방식을 식으로 나타내면 다음과 같이 표현된다.

$$UPC(x, y, t) = \sum_{i=x-N}^{i=x+N} \sum_{j=y-N}^{j=y+N} U(i, j, t) \quad 4.7$$

$$U(x, y, t) = \begin{cases} 1, & \text{if } |I(i, j, t) - I(i, j, t-1)| > Th \\ 0, & \text{otherwise} \end{cases}$$

이때 Th 는 화소의 정합여부를 결정하는 문턱값이다. 비정합 화소 개수 (Unmatched Pixel Count)를 이용한 움직임 검출방식을 사용하는 경우 정합여부를 화소 단위로 반영하므로 화소개개의 특성을 잘 나타내게 되어, 물체와 물체가 겹치는 영역에서도 높은 변화량을 나타낸다. 이것은 차-누적을 이용하는 경우 창안의 화소중 밝기 값의 차가 큰 화소가 계산에 미치는 영향이 커지는 반면, $UPC(x, y, t)$ 의 경우 창안에 있는 모든 화소가 일정하게 결과에 영향을 미치기 때문에 나타나는 현상이다.

비정합 화소 개수 UPC 를 이용한 움직임 검출방식은 색상 변환한 영상의 움직임 정보 검출을 위해서도 적용 가능하다. 그림 4.6은 이 두 가지 방법의 성능을 비교하기 위한 실험 결과이다. 그림 4.6 (a)는 상대적으로 낮은 0에서 30까지의 무작위 밝기 값을 가지는 배경영역에 200에서 230까지의 높은 무작위 밝기 값을 가지는 직사각형의 물체를 나타내었다. 그림 4.6 (b)는 직사각형의 물체 가로, 세로 방향으로 50 화소만큼 일정하게 이동한 결과를 나타낸다. 이 영상은 다음절에서 설명될 색상 변환한 두 영상에서 얼굴색과 다른 배경에서 얼굴이 이동했을 때의 상황을 나타낸다. 그림 4.6 (c)는 차-누적 측정 $ADM(x, y, t)$ 을 이용한 움직임 영역 검출 결과이다. 결과에서 보듯이 움직임으로 인해 배경과 물체가 대응되는 영역은 높은 움직임 영역으로 잘 표시되지만 실제로 같은 움직임이 있어도 움직인 물체가 서로 겹치는 영역은 낮은 변화량을 나타내게 된다. 그림 4.6 (d)는 비정합 화소 개수 $UPC(x, y, t)$ 를 이용해 움직임 영역을 검출한 결과이다. 물체와 물체가 겹치는 영역에서도 높은 변화량을 나타낸다는 것을 알 수 있다. 비정합 화소 개수를 이용하는 경우 움직이는 물체가 겹치는 부분도 정확히 감지를 하기 때문에 앞의 두 방법보다 움직임 영역을 구하는 데 더 유리하다.

b. 색상 변환 영상의 움직임 정보

본 절에서는 색상 변환한 영상의 움직임 정보를 이용해 얼굴 영역을 효율적으로 찾는 방법을 설명한다. 기존의 움직임 영역 측정 방법은 화소의 밝기 값의 절대 차만을 고려함으로써 화소의 밝기 값에는 자체에는 어떠한 의미도 부여하지 않는다. 다만 움직임이 있다면 밝기 값의 상대적인 차가 클 것이라는 가정으로 움직임 정도를 계산한 것이다. 하지만 위에서 정의한 색상 분포 함수를 이용해 색상 변환된 영상은 화소의 밝기 값이 원하는 물체의 확률값을 나타내기 때문에 기존의 움직임 추정 방법과 비교하여 부가적인 정보로 화소의 밝기 값을 반영할 수 있다. 즉, 색상 변환한 명암영상에서는 밝은 부분이 원하는 색상정보가 많이 포함된 영역을 나타내며 원하는 색상정보가 움직임이 있을 때에는 앞에서 구한 움직임 정보에 그 위치의 밝기 값으로 가중치를 줌으로써 새로운 움직임 정보를 정의할 수 있다. 이러한 새로운 움직임을 변환색상 가중 움직임 정보라 정의한다. 앞 절에서 설명한 비정합 화소 개수 UPC를 이용한 움직임 검출방식을 색상 변환된 명암영상에 적용하여 변환색상 가중 움직임 측정치 WUPC를 식으로 나타내면 다음과 같이 표현된다.

$$WUPC(x, y, t) = Z(x, y, t) \otimes \sum_{i=x-N}^{i=x+N} \sum_{j=y-N}^{j=y+N} U(i, j, t) \quad 4.8$$

$$U(i, j, t) = \begin{cases} 1, & \text{if } |Z(i, j, t) - Z(i, j, t-1)| > Th \\ 0, & \text{otherwise} \end{cases}$$

또한 \otimes 연산은 퍼지-AND 연산을 나타낸다. 변환색상 가중 움직임 측정치 $WUPC(x, y, t)$ 는 식 4.5로 표현되는 색상분포 모델을 이용하여 두 화면에 대한 색상 변환 결과인 $Z(x, y, t)$ 와 $Z(x, y, t-1)$ 를 식 4.7에서 설명된 움직임 검출에 적용한 후 다시 $Z(x, y, t)$ 와 퍼지-AND 한 결과이다. 이러한 결과는 색상분포 모델로 표시되는 원하는 색상의 움직임만 강조하고 그렇지 못한 색상의 움직임 정보는 그 의미를

감소시키는 기능을 제공한다. 퍼지-AND 연산에는 대수적 곱셈(algebraic product) 연산을 사용하였다. 그림 4.6 (e)는 변환색상 가중 움직임 측정치 $WUPC(x, y, t)$ 를 이용해 움직임 영역을 검출한 결과이다. 원하는 색상을 갖는 물체 영역에서 높은 변화량을 나타낸다는 것을 알 수 있다. 변환색상 가중 움직임 검출방식은 움직이는 얼굴영역 뿐만 아니라 움직이는 얼굴 요소 추출을 위해서도 사용 가능하다.

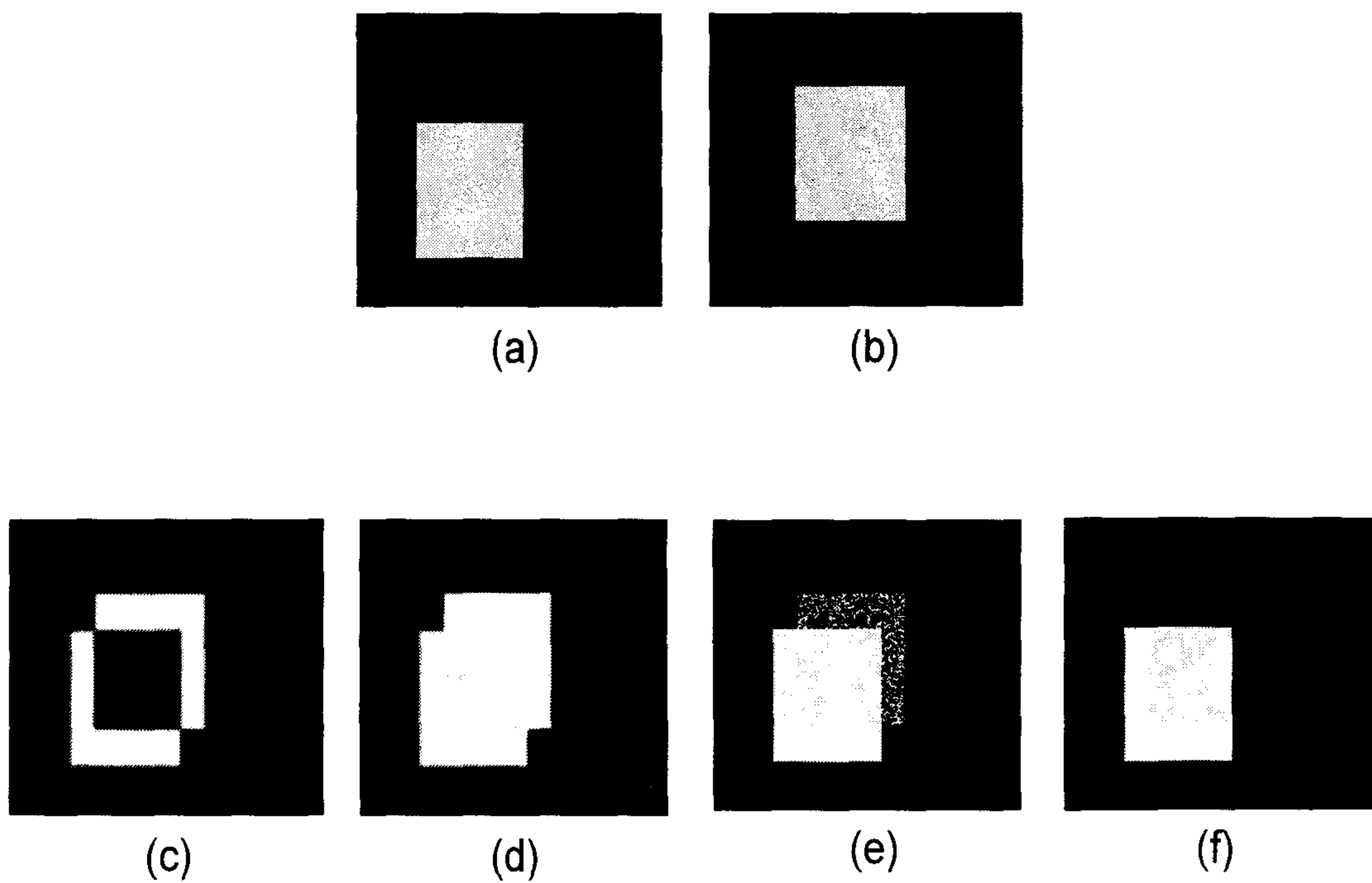


그림 4.6 모의영상에 대한 움직임 측정 방법의 성능비교. (a),(b) 모의 영상 1, 2 (c) ADM 결과 (d) UPC 결과 (e) WUPC 결과 (f)AWUPC 결과

c. 색상 변환 적응형 움직임 정보

이 절에서는 색상 변환된 영상의 움직임 정보를 이용해 얼굴 영역을 효율적으로 찾는 방법으로 변환색상 가중 움직임 측정치 WUPC를 구할 때 사용되는 문턱값을 색상 변환된 화소의 값에 따라 적응적으로 사용하는 AWUPC(Adaptive Weighted UPC) 방법을 제시한다. 식 4.8에서 나타낸바와 같이 WUPC 방식에서 각 화소의 정합 여부를 나타내는 $U(x, y, t)$ 를 결정하기 위한 문턱값 Th 를 해당 화소의 변환된 밝기 값 $Z(x, y, z)$ 를 이용하여 결정한다. 즉, 관심이 있는 물체의 화소에 대해서는 작은 움직임에도 민감하게 반응할 수 있도록 문턱값을 낮출 필요가 있고 배경이나 관심대상이 아닌 화소에 대해서는 큰 움직임에도 반응하지 않도록 해야한다. 이러한 문턱값을 정해주는 함수로는 주로 신경망의 반응함수로 사용되는 Sigmoid함수의 특성을 이용하여 다음과 같이 정의한다.

$$Th(Z) = \frac{255}{1 + e^{\frac{Z(x, y, t) - 255/2}{Q}}} \quad 4.9$$

여기서의 Q 값은 신경망 분야에서 Neuron의 온도라고 지칭하는 값으로 크기에 따라 함수의 기울기가 정해진다. Q 의 값이 낮아질수록 전체함수는 급한 경사를 갖게 되어 계단함수에 가까워지는 특성을 가진다. 이 함수는 비선형적 특성을 가지고 있어 최대, 최소 입력값 근처의 값에 대해 부드럽게 반응하기 때문에 본 응용에 필요한 밝기 정도에 따른 문턱값을 정하는 함수로 적합하다. 이 Sigmoid 함수를 본 응용에 맞는 함수로 나타내기 위해 입력 값이 큰 경우는 결과 값은 낮은 값이 나올 수 있도록 하고 입력 값이 작은 경우에는 결과 값이 큰 값이 나오도록 한다. 즉 밝기 값이 높은 화소의 움직임을 계산할 때는 낮은 문턱값을 가지게 하고 밝기 값이 낮은 화소에 대해서는 높은 문턱값을 가지게 한다. 그리고 결과 값이 영상의 밝기 값에 맞도록 $[0, 1, \dots, 255]$ 의 값들로

나타내 줄 수 있는 함수로 변환한다.

그림 4.6 (f)는 적응적 변환색상 가중 움직임 측정치 $AWUPC(x, y, t)$ 를 이용해 움직임 영역을 검출한 결과이다. WUPC 방식을 사용한 그림 4.6 (e) 에 비해 살색으로 강조된 부분의 움직인 물체의 모양을 잘 나타냄을 알 수 있다. AWUPC 변환은 t 번째 영상의 화소의 밝기 값이 클수록 움직임의 문턱값이 낮아지기 때문에 색상변환으로 명암이 강조된 살색영역은 작은 움직임에도 민감하게 높은 움직임 값을 나타내게 된다. 반면에 화소의 밝기 값이 낮은 영역에 대해서는 움직임의 문턱값이 높아져 비교적 큰 움직임이 있다하더라도 낮은 움직임 값을 나타내게된다. 예를 들어 살색이 아닌 옷의 움직임 영역을 고려해보면 얼굴과 같은 신체의 움직임과 비슷한 변화량이라고 할 수 있겠지만 본 과정을 거치게 되면 작은 움직임 영역으로 나타나게 되는 것이다. 앞에서 열거한 움직임 정보들을 실제 영상에 대해 적용한 결과를 그림 4.7의 (a)~(f)에 보인다. (a) 와 (h)는 연속 영상을 GFCD색상 변환한 영상이다. 그림 (c)는 차-누적 측정, 그림 (d)는 비정합 화소개수를 이용한 측정치, 그림 (e)는 카메라 잡음을 고려하여 문턱 값을 10으로 한 경우의 변환색상 가중 움직임 측정치이다. 그림 (f)는 적응적 변환색상 가중 움직임 측정 결과이다. 실제 영상에 대해서도 모의 환경 실험 영상과 유사한 결과를 나타내며, 움직임이는 색상 영역을 구하는데 효과적인 방법임을 알 수 있다.



(a)



(b)



(c)



(d)



(e)



(f)

그림 4.7 실제 얼굴 영상에 대한 움직임 측정 방법의 성능비교. (a),(b) GFCD 얼굴 영상 1, 2
(c) ADM 결과 (d) UPC 결과 (e) WUPC 결과 (f) AWUPC 결과

2절. 색상 움직임을 이용한 얼굴 특징점 추출

제 1절에서 설명한 색상 움직임을 이용한 얼굴 영역 검출후 눈과 입의 얼굴 요소에 대한 영역 및 특징점들은 다음의 방법으로 구하여 진다.

가. 눈 영역 추출 알고리즘

앞 장에서 설명한 색상 움직임 정보와 BWCD 색상 변환을 사용하여 눈 영역을 추출한다. 위에서 추출한 살색 움직임 영역에 대해 형태소 연산자를 이용한 Top-Hat 변환을 하면 명암에서의 Valley에 해당하는 부분을 추출할 수 있다[11]. Top-Hat 변환은 다음과 같다.

$$h = f - (f \cdot b)$$

여기서 h 는 결과영상을 의미하고 f 는 입력영상, b 는 Structuring element function을 의미한다. 눈 영역은 살색 컬러 분포에서 낮은 확률값을 가지며 급격한 밝기 값의 변화를 보이는 Valley에 해당한다. 그리고, 살색 영역내에서 눈 영역뿐만 아니라 눈썹이나 입, 흘러내린 머리칼, 그리고 움직임 영역에서 얼굴이 겹쳐져 나온 영역도 Valley에 해당된다. 눈과 눈썹이외의 영역을 제외한 다른 부분들을 억제하기 위해 Valley를 추출한 영상과 BWCD 색상 변환 영상을 퍼지-AND를 한다. 이 결과 검은 색이나 흰색이면서 살색 영역 중 Valley에 해당하는 부분만 높은 값을 가지게 된다. 그림 4.9 (a)는 이러한 과정을 거쳐 눈과 눈썹 영역만이 추출되는 것을 보인다. 하지만 검은 턱수염이나 흘러내린 머리칼이 있는 경우는 잘못된 결과를 나타낼 수 있으므로 올바른 눈과 눈썹의 영역을 결정하기 위해 눈과 눈썹의 지정학적 특성을 이용한다. 즉, 그림 4.8 에 나타낸바와 같이, 머리가 어느 방향으로 회전하더라도 눈과 눈썹의 중심점들을 서로 연결하면 직사각형에 가까운 형태를 가지게 된다. 이러한 관계를 이용하여 눈과 눈썹에 해

당하는 영역만 검출하고 기타 영역들은 제거한다. 그림 4.9 (a)의 결과영상을 일정 문턱값을 적용하여 이진 영상을 만든 다음 연결된 각 영역들을 labeling하여 각각의 영역들로 정의한다. 그리고 각 영역들을 4개의 쌍으로 하여 직사각형에 얼마나 가까운지를 판단하여 눈과 눈썹의 영역을 결정한다. 그림 4.8은 4개의 쌍에 대해 각 선분의 길이 $E_{11} - E_{14}$ 와 수평에 대해 각 선분이 이루는 각도 $E_{a1} - E_{a4}$ 를 나타내며, 이와 같은 4개의 쌍이 얼마나 직사각형에 가까운지를 판단값 D 는 다음과 같다.

$$D = R_l(E_{11}, E_{12}) R_l(E_{13}, E_{14}) R_a(E_{11}, E_{13}) R_a(E_{12}, E_{14}) \quad 4.10$$

여기서 $R_l(E_{11}, E_{12})$ 는 눈 거리와 눈썹의 거리의 비를 나타내고, $R_l(E_{13}, E_{14})$ 는 눈과 눈썹사이의 거리의 비로써, 1 보다 큰 양의 값으로 나타낸다. 또한 $R_a(E_{11}, E_{13})$ 는 왼쪽 눈과 눈썹이 이루는 각의 직각정도를 나타내고 $R_a(E_{12}, E_{14})$ 는 오른쪽 눈과 눈썹이 이루는 각의 직각 정도를 나타낸다. 직각의 정도는 다음과 같이 정의된다.

$R_a(E_{11}, E_{13})$ 은 $E_{a1} - E_{a3} > 90$ 도 이면 $(E_{a1} - E_{a3})/90$ 이고 그렇지 않는경우는 $90 / (E_{a1} - E_{a3})$ 의 값을 갖는다. 이 값은 직사각형에 가까울수록 결과값은 1보다 크고 1에 가까운 값을 가지게 된다. 이와 같은 판단식을 거치게 되면 각각의 쌍 중 직사각형에 가장 가까운 4개의 영역이 결정된다. 이 4개의 영역 중 왼쪽과 오른쪽에 위치하는 두쌍의 영역 중 아래에 위치하는 두 개의 영역을 왼쪽 눈과 오른쪽 눈으로 결정한다. 그림 4.9 (b)는 턱수염과 콧사이의 머리카락으로 인해 추출된 부분과 눈, 눈썹 영역이 나타났을 경우 위의 관계식을 이용해 눈과 눈썹의 영역을 결정한 영상이다.

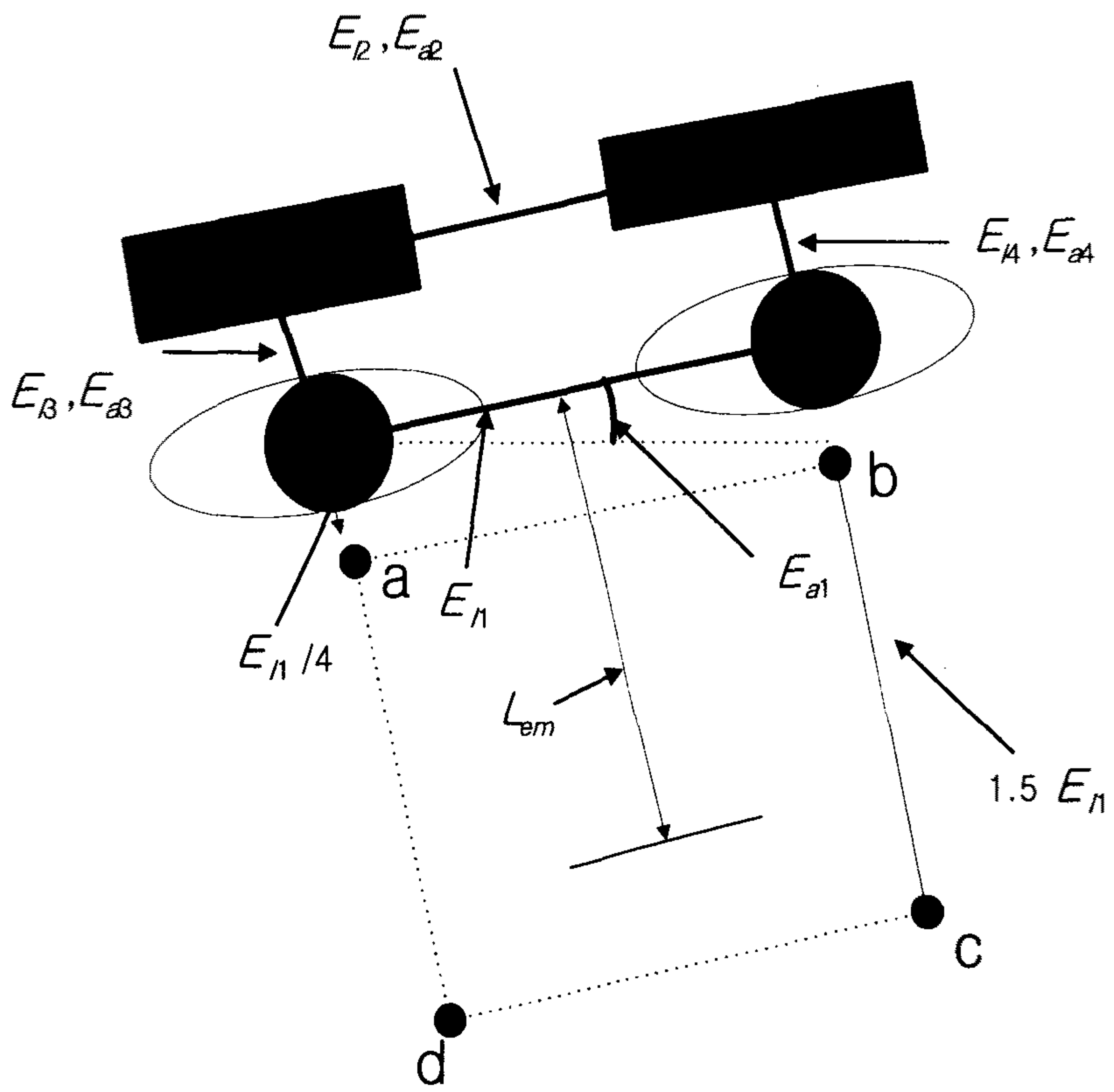


그림 4.8 눈과 입 영역 결정을 위한 얼굴의 지정학적 모델

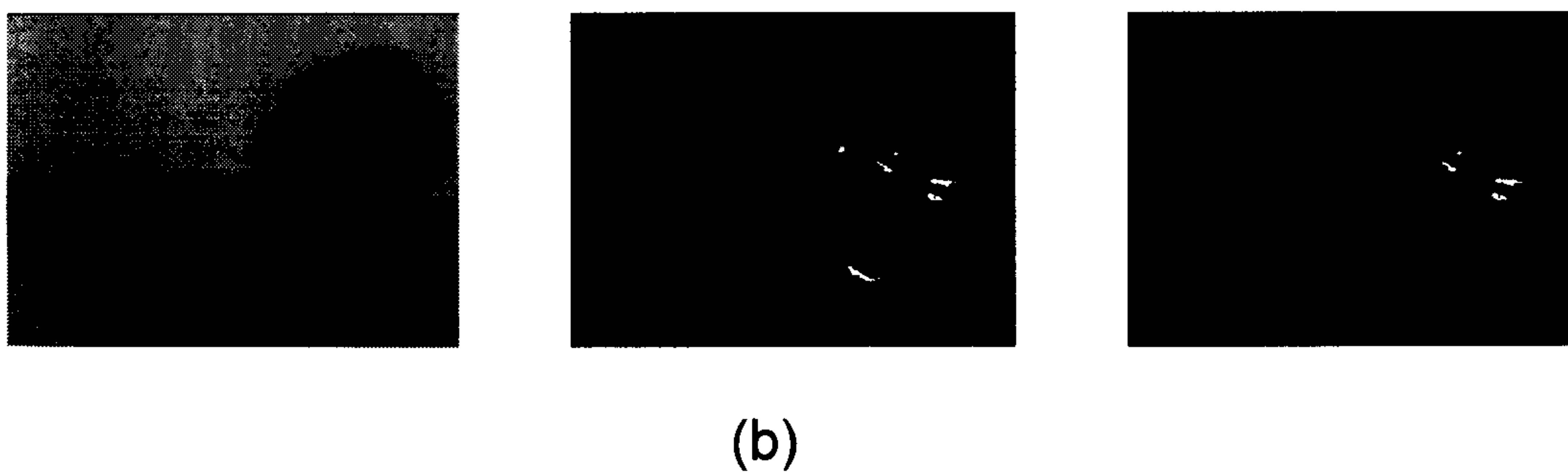
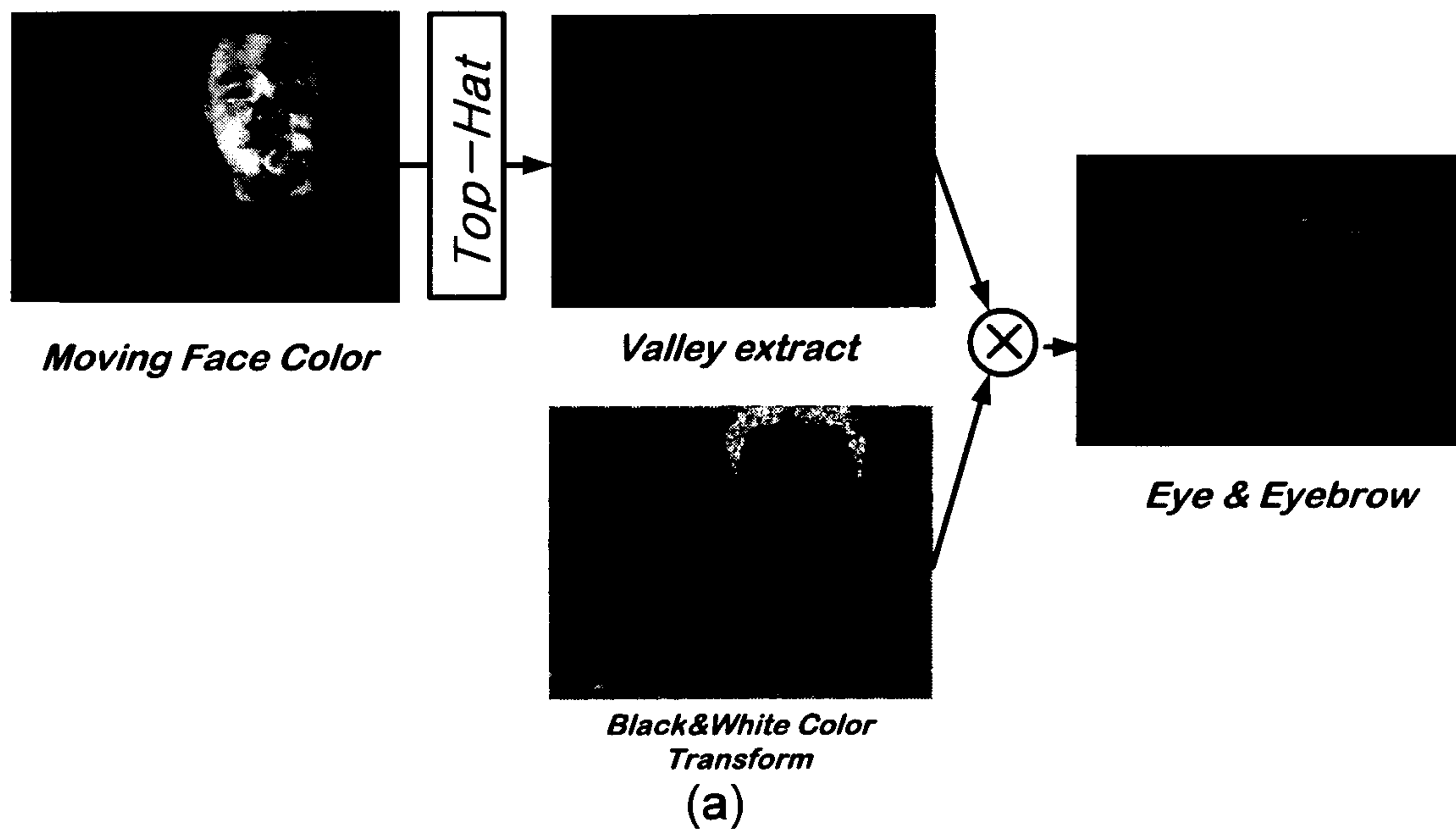


그림 4.9 눈 영역 검출 과정

(a) 눈 후보영역 검출과정 (b) 지정학적 관계를 이용한 눈 영역의 결정

나. 입 영역 추출과정

눈의 영역이 결정이 되면 얼굴 영역의 일반적인 사전지식으로 대강의 입의 영역을 예측할 수 있다. 한국인의 일반적인 얼굴은 그림 4.8에 나타낸 바와 같이 양 눈 사이의 거리 E_{11} 과 양 눈의 중심에서 입까지의 거리 L_{em} 이 거의 1대1의 비를 가진다.[9] 이와 같은 특성을 이용하여 E_{11} 을 $1/4 E_{11}$ 에 해당하는 거리만큼 평행 이동하여 한 번으로 하는 사각형 영역을 입 후보 영역으로 정의한다. 사각형의 다른 변의 길이는 얼굴이 정면이 아닐 경우 실제보다 작게 측정될 것이므로 E_{11} 의 1.5배에 해당하는 거리를 결정하였다. 결정된 입 후보영역에 대해 위에서 추출한 살색영역의 히스토그램을 이용하여 입의 영역을 결정한다. 입의 컬러분포는 눈이나 눈썹의 경우와는 달리 얼굴색과 흡사한 분포를 가지고 개인간의 편차가 심하기 때문에 얼굴 영역에서 얼굴색과 구분되는 입의 컬러 분포를 직접 구하기는 상당히 어렵지만 얼굴색의 분포와는 차이가 있기 때문에 GFCD로 색상 변환된 영상에서 입에 해당하는 영역은 살색 영역에 비해 낮은 밝기 값을 가지게 된다. 색상 변환된 영상에서 살색과 입술색 영역의 차이를 강조하기 위해 다음과 같은 연산을 수행한다.

$$L_R(x, y) = Z(x, y) \sum_{i=x-N}^{i=x+N} \sum_{j=y-N}^{j=y+N} Z(i, j) \quad 4.11$$

이 연산은 주위 화소의 값들을 더한 후 해당 화소에 대한 값으로 가중치를 주기 때문에 주위에 높은 밝기 값을 갖는 화소는 상대적으로 더 높은 값을 갖게 되고 낮은 밝기 값을 갖는 화소는 상대적으로 더 낮은 밝기 값을 가지게 되어 입영역을 강조한다.

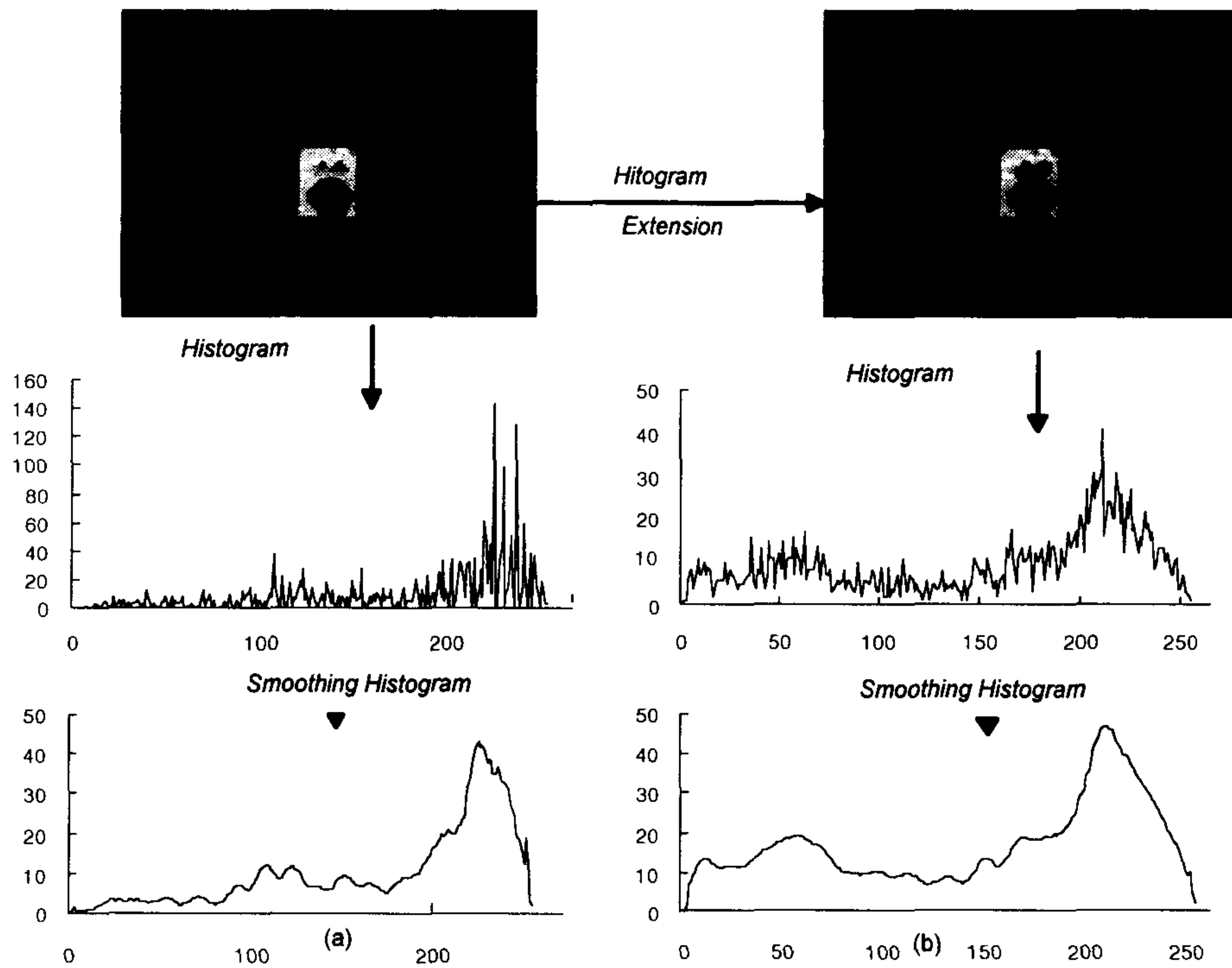


그림 4.10 입영역 결정을 위한 후보영역의 처리

그림 4.10은 이러한 변환을 입의 후보 영역에 적용한 결과와 각각의 히스토그램을 나타낸다. 그림 4.10(a) 영상에서 살색 영역과 입의 영역에 대한 밝기 차가 크지 않기 때문에 영역을 구분하기 위한 문턱값을 정하기가 어려우나 그림 4.10(b) 영상을 보면 살색 영역과 입의 영역이 좌측 영상에 비해 밝기 차가 크게 나타남을 확인할 수 있다. 또한 급격하게 변하는 지역 최소치를 없애기 위해 Gaussian 필터를 이용하여 히스토그램 분포를 고르게 하였다[14]. 히스토그램 분포를 이용한 입의 영역과 살색 영역을 분리하기 위한 문턱값 결정을 위해 Mode method[13]를 사용한다. 이 방법은 거의 균일한 밝기 값을 가진 배경에 역시 균일한 밝기 값을 가지는 하나의 물체를 분리해내는 방법으

로 입 후보영역 상에서 살색과 입의 영역은 이와 같은 특성을 나타내기 때문에 본 응용에 적합한 방법이다.

Mode method를 이용하여 선택된 최저값을 이용하여 입의 후보영역을 이진화하고, 연결된 화소의 개수를 이용하여 입의 영역을 구분하는 기준으로 삼는다. 입 이외의 다른 영역은 상식적으로 입의 크기보다는 클 수 없기 때문에 위의 방법은 타당하다고 할 수 있다. 구체적인 알고리즘은 다음과 같다.

< Mode method >

① 히스토그램 상에서 일정 크기 이상의 거리를 두고 있는 2개의 지역 최대치를 찾는다. 이를 $g_{\max 1}$, $g_{\max 2}$ 라 한다. 그림 2.2에서 $g_{\max 1}$ 은 입술의 영역을 대표하는 밝기값이라 할 수 있고, $g_{\max 2}$ 는 살색의 영역을 대표하는 밝기값이라 할 수 있다.

② $g_{\max 1}$, $g_{\max 2}$ 사이에서 가장 낮은 최소치 g_{\min} 을 찾는다. 여기에서의 g_{\min} 값은 $g_{\max 1}$, $g_{\max 2}$ 로 대표되는 두 영역을 구분할 수 있는 값이다. 그리고, Gaussian필터 과정을 통해 급격한 기울기를 가지는 Valley를 제거했기 때문에 갑자기 0값을 가지는 밝기값이 선택되지는 않는다.

③ 여기서 선택된 g_{\min} 은 입술과 살색영역을 분리할 수 있는 밝기값이라 할 수 있다. g_{\min} 를 기준으로 입의 후보영역을 이진화한다.

Mode method를 이용하여 이진화한 입의 후보 영역은 입의 영역이외에 콧구멍에 해당하는 부분, 그리고 목 아래 부근의 어두운 부분이 포함될 수 있다. 이들 입이외의 영역들과 입의 영역을 구분하는 기준으로 연결된 화소의 개수를 이용한다. 입이외의 다른 영역들은 상식적으로 입의 크기보다는 클 수 없기 때문에 위의 방법은 타당하다고 할 수 있다.

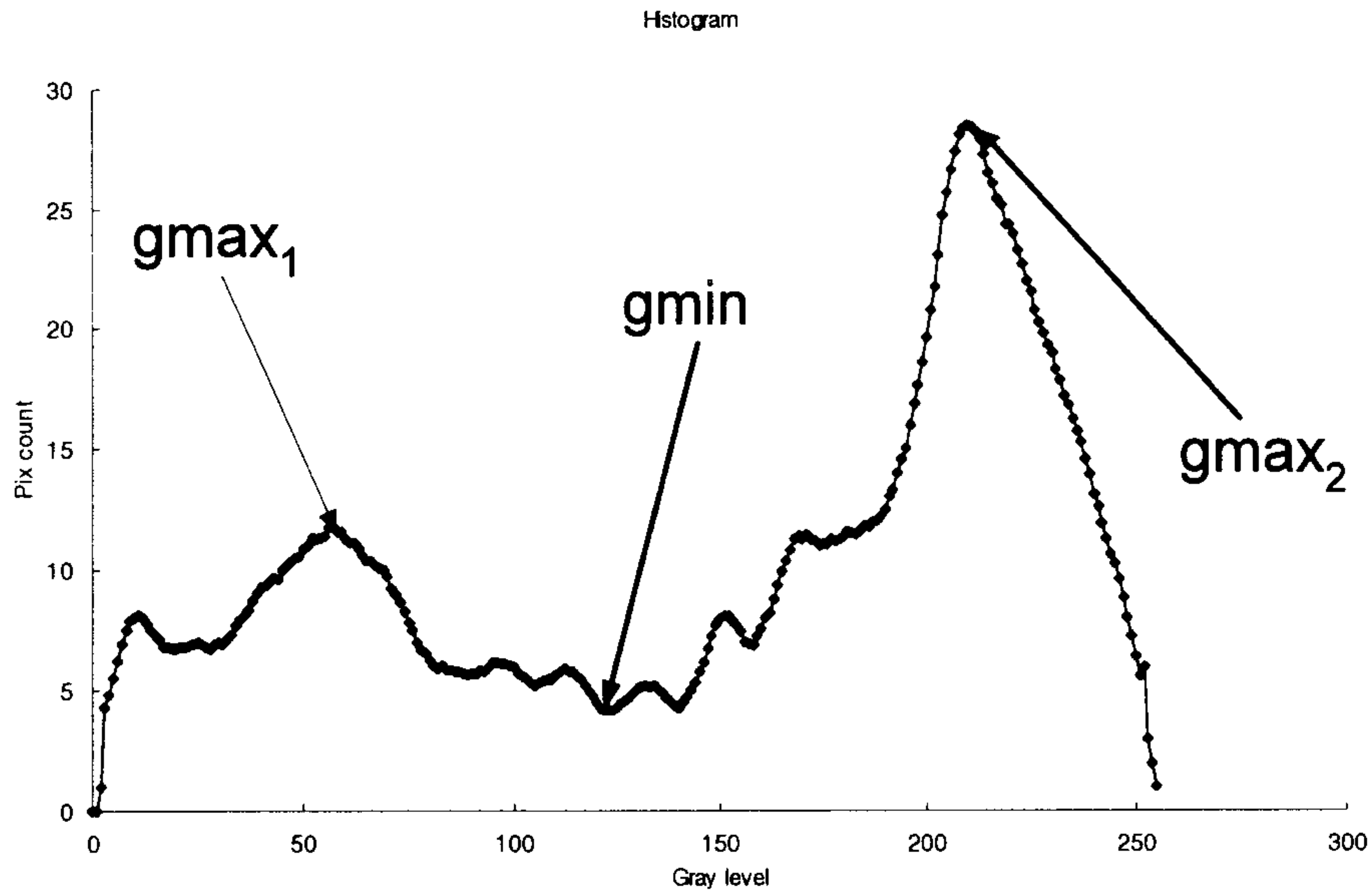


그림 4.11 입 후보영역의 히스토그램 분포

위에서 추출된 얼굴 요소 영역들에 대해 정확한 방향을 나타내는 특징점들을 추출한다. 이는 2차원 동영상에서 3차원 움직임을 추적하는 알고리즘중 특징점을 기반으로 하는 방법이나, 모델기반 부호화등에 사용되는 모델 정합을 위한 특징점 추출등에 사용될 수 있기 위함이다. 추출된 눈과 입 영역의 좌표에 대한 2×2 의 Covariance matrix(Σ)의 eigen vector(Φ)와 eigen value(λ)를 계산하여 각 영역에서 장축과 단축이 만나는 4개의 화소를 특징점으로 결정한다. 여기에 사용된 Covariance matrix는 식(14)와 같이 나타나고 eigen vector와 eigen value의 계산은 다음과 같다.

$$\mu = \frac{1}{N} \sum_{i=1}^N X_i$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^N X_i \cdot X_i^T - \mu \mu^T \quad 4.12$$

$$\Sigma \phi = \lambda \phi$$

여기서 $X_i = (x_i, y_i)$ 는 연결된 물체의 화소의 좌표값이고 N 은 연결된 화소의 개수, μ 는 x, y 좌표의 평균값을 의미한다. 인간의 눈이나 입같은 경우 방향성이 뚜렷한 타원형이기 때문에 이와 같은 계산이 가능하다. 이 특징점들은 2차원 동영상에서 3차원 움직임을 추적하는 알고리즘중 특징점을 기반으로 하는 방법[10]이나, 모델기반 부호화 등에 사용되는 모델 정합을 위한 특징점 추출[16]등에 사용될 수 있다.

다. 실험 결과 및 향후 연구 방향

a. 실험 방법 및 환경

전체적인 알고리즘의 흐름도는 그림 4.12에 나타내었다. 제안한 방법의 성능을 검증하기 위한 실험으로 Solaris 2.5 운영 체제하의 Sun Ultra Spark에서 C언어를 이용하여 실험 구현하였다. 컬러 비디오 시퀀스 영상은 Sony digital camcorder (DCR-VX1000)로 취득한 512×480 해상도의 영상을 320×240 해상도의 영상으로 샘플링하여 사용하였다. 복잡한 배경과 여러 인물에 대한 실험을 위해 총 10명의 인물에 대해 두 장을 쌍으로 하는 50장의 영상을 취득하였다. 배경이나 옷이 살색과 가까운 색상을 가지는 경우, 개인적인 얼굴색의 변화, 표정의 변화 등을 포함하였다. 여러 날에 걸쳐 영상을 취득함으로써 시간의 변화로 인한 영상의 변화도 포함토록 하였다. 그리고 실시간 처리에서 필요한 조건이라 할 수 있는 작은 움직임을 갖는 영상에 대한 실험을 위해 동일한 인물에 대한 총 130장의 연속 영상을 취득하였다.

b. 실험 및 결과

320 × 240 크기의 영상을 전 과정에 걸쳐 원 영상크기 그대로 이용하였다. 색상 움직임 이용을 이용한 얼굴 영역 검출에 사용되는 창의 크기는 5 × 5이며 변형 Sigmoid함수의 Q값은 20을 이용하였다. Top-Hat변환에 사용되는 Structuring element 의 크기는 9이다. 입 영역추출을 위한 히스토그램 평활화에 사용되는 Gaussian 필터는 커널의 크기가 15, 분산이 2인 필터를 사용하였다. 여러 인물과 여러 가지 배경을 위한 총 50개의 영상 중 47개의 영상에서 성공하여 94%의 성공률을 나타내었다. 그리고 처리시간은 약 2초 정도가 소요되어 실시간 구현에 효과적임을 알 수 있었다. 전체 실험 영상 중 구분되는 특징을 가지는 결과 영상에 대해 설명한다. 그림 4.13(a)는 얼굴 영역이 차지하는 비중이 작은 영상에서의 결과이다. 두 번째 프레임에서 눈의 영역이 너무 작아 눈의 끝점을 부정확하게 추출되었다. 그림 (b)는 표정의 변화와 살색에 가까운 옷을 입고 있는 경우이다. 표정의 변화로 눈의 영역이 작아졌으나 정확한 눈의 특징점을 추출했으며 움직이는 살색 옷 영역 안에서 BWCD에 해당하는 영역이 추출되더라도 눈과 눈썹의 지정학적 관계로 올바른 결과를 얻었다. 그림 (c)는 정면 상에서 기울어진 얼굴에 대한 실험이다. 이런 경우도 눈, 입 영역의 기울기를 정확하게 계산하기 때문에 정확한 특징점들을 추출하였다. 그림 (d)는 얼굴 영역이 큰 경우이다. 색상 정보의 특성상 영역의 크기에 상관없기 때문에 역시 올바른 결과를 얻을 수 있었다. 그림 (e)는 머리의 상하 좌우 회전과 입을 벌린 경우에 대한 실험이다. 모든 경우에 올바른 결과를 얻었다. 그림 (f),(g)는 실패한 경우의 영상을 보인다. 그림 (f)는 눈썹이 열린 여성의 경우이다. 남성에 비해 상대적으로 눈썹이 열고 화장을 하여 BWCD로 표현된 영상에서 눈썹의 영역이 제대로 나타나지 않은 결과를 보인다. 그림 (g)는 안경을 쓴 경우이다. 검은 색 안경을 쓰고 있기 때문에 눈 영역과 연결이 되어 정확한 특징점이 추출되지 않았다. 그림 4.13의 결과를 정리하여 표4.2에 나타내었다. 작은 움직임을 가진 130개의 영상에서도 높은 성공률을 보였다. 그림 4.14은 정면 얼굴에서 작은 움직임을 가진 때의 결과

영상들이다. 불과 몇 화소정도의 움직임에도 정확한 특징점들을 추출함을 알 수 있다. 그림 4.15는 시선이 위를 바라보는 얼굴에서의 실험이다. 조명이 머리 위쪽에서 내려오기 때문에 이마나 광대뼈와 같은 얼굴의 일부분이 반사되어 GFCD나 BWCD에서 약간의 오차가 생김을 알 수 있다. 그림 4.16은 시선이 아래를 바라보는 얼굴에서의 실험이다. 역시 조명의 방향에 의해 GFCD나 BWCD에서의 얼굴색 분포가 정면에 비해 부정확함을 알 수 있다. 그림 4.17은 왼쪽으로 얼굴이 회전할 때의 실험이다. 이와 같은 경우 정면에서의 경우와 비슷한 조명의 환경이기 때문에 얼굴색 분포는 안정적이거나 심한 회전으로 인해 눈이나 눈썹 등의 영역이 잘 나타나지 않으면 실패를 초래할 수 있다. 그림 4.18은 기울어지는 움직임에서의 실험이다. 이 역시 정면상의 경우와 비슷한 조명 환경으로 얼굴색 분포가 안정적이며 기울기에 상관없이 얼굴 특징점들이 정확히 구해짐을 알 수 있다. 움직임이 작은 경우에 대해서도 특수한 경우를 제외하고는 이와 같이 높은 성공률을 보여 실시간 구현을 위한 작은 움직임을 나타내는 연속 영상에서도 적합한 방법임을 알 수 있다.

표 4.2 그림 4.13 의 결과 요약

	특징	결과	성공 여부 (○/×)
(a)	작은 얼굴	부정확한 눈 특징점	○
(b)	표정 변화, 살색 옷	눈의 모양과 움직이는 살색에 상관없는 특징점	○
(c)	정면 기울어진 얼굴	기울기에 상관없는 특징점	○
(d)	큰 얼굴	얼굴 크기에 상관없는 특징점	○
(e)	머리의 상하좌우 회전과 입을 벌린 경우	머리의 방향과 입의 모양에 상관없는 특징점	○
(f)	눈썹 화장 여성	눈썹 영역의 부정확한 정의	×
(g)	안경착용	눈 영역과 안경테의 연결	×

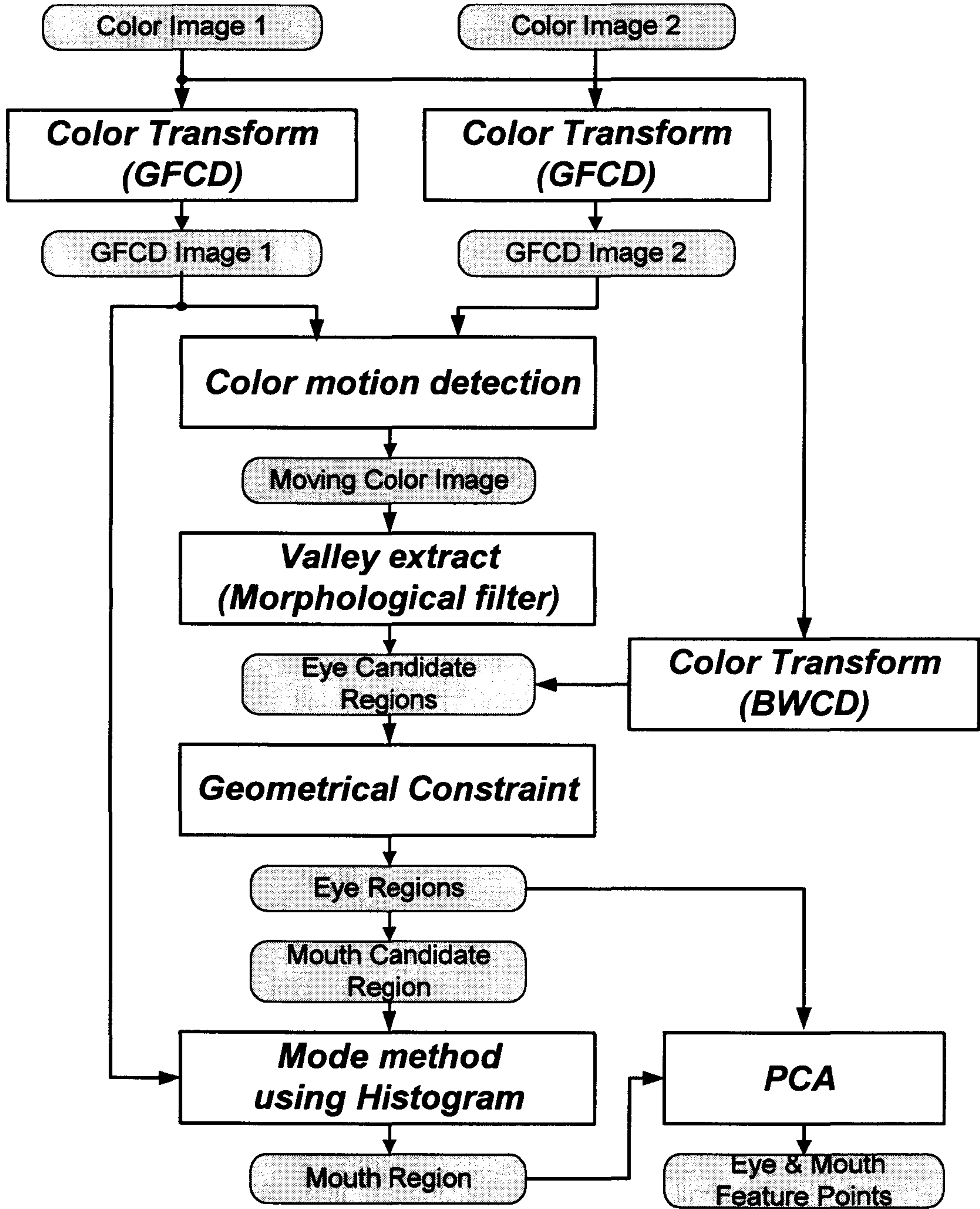


그림 4.12 제안된 알고리즘의 흐름도

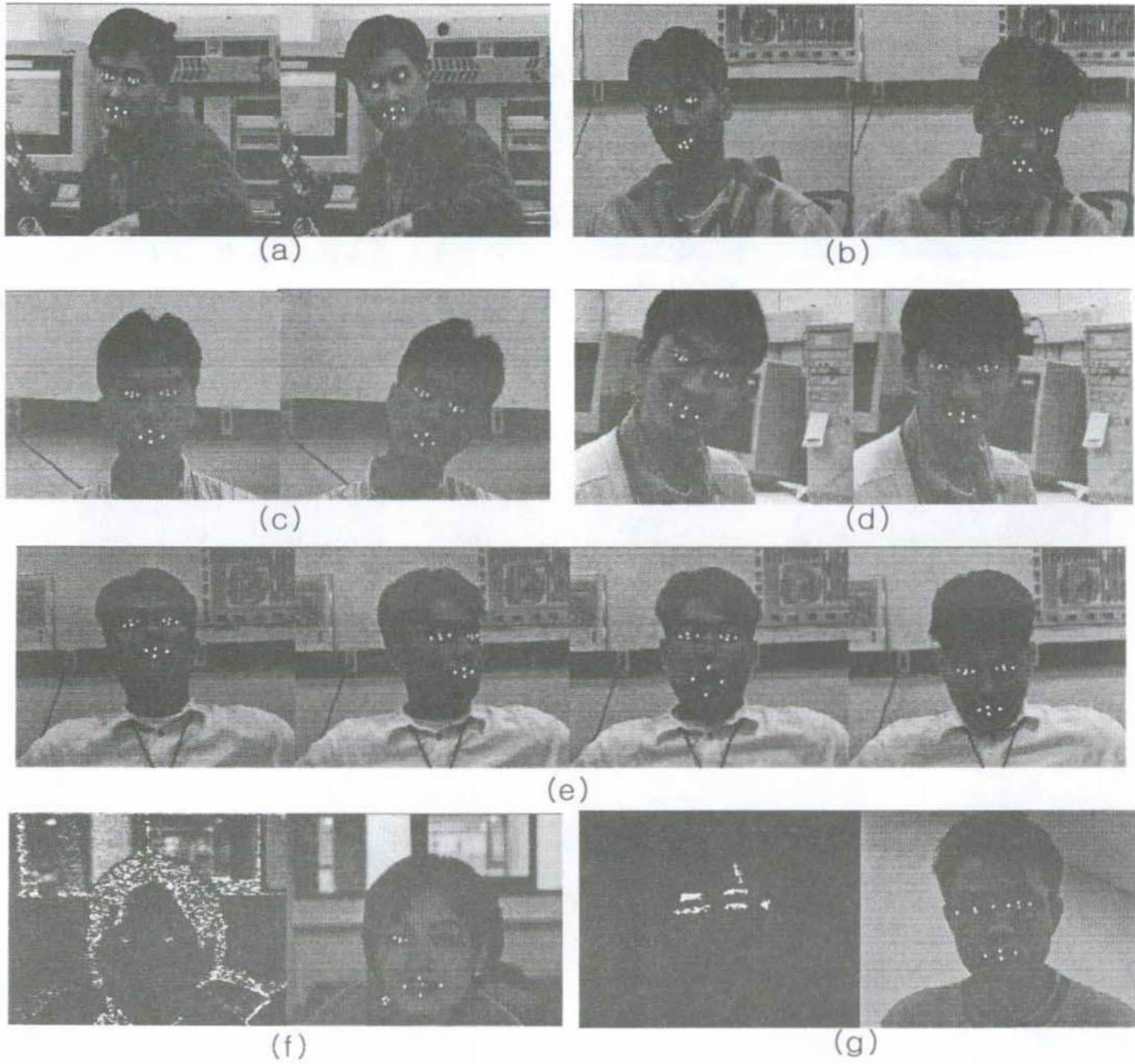


그림 4.13 얼굴요소 특징점 검출 결과

(a) 작은 얼굴 (b) 표정의 변화와 살색 옷 (c) 정면상에서 기울어진 얼굴 (d) 큰 얼굴 (e) 상하좌우의 머리의 움직임과 입을 벌린 경우 (f) 화장한 여성 (g) 안경 착용

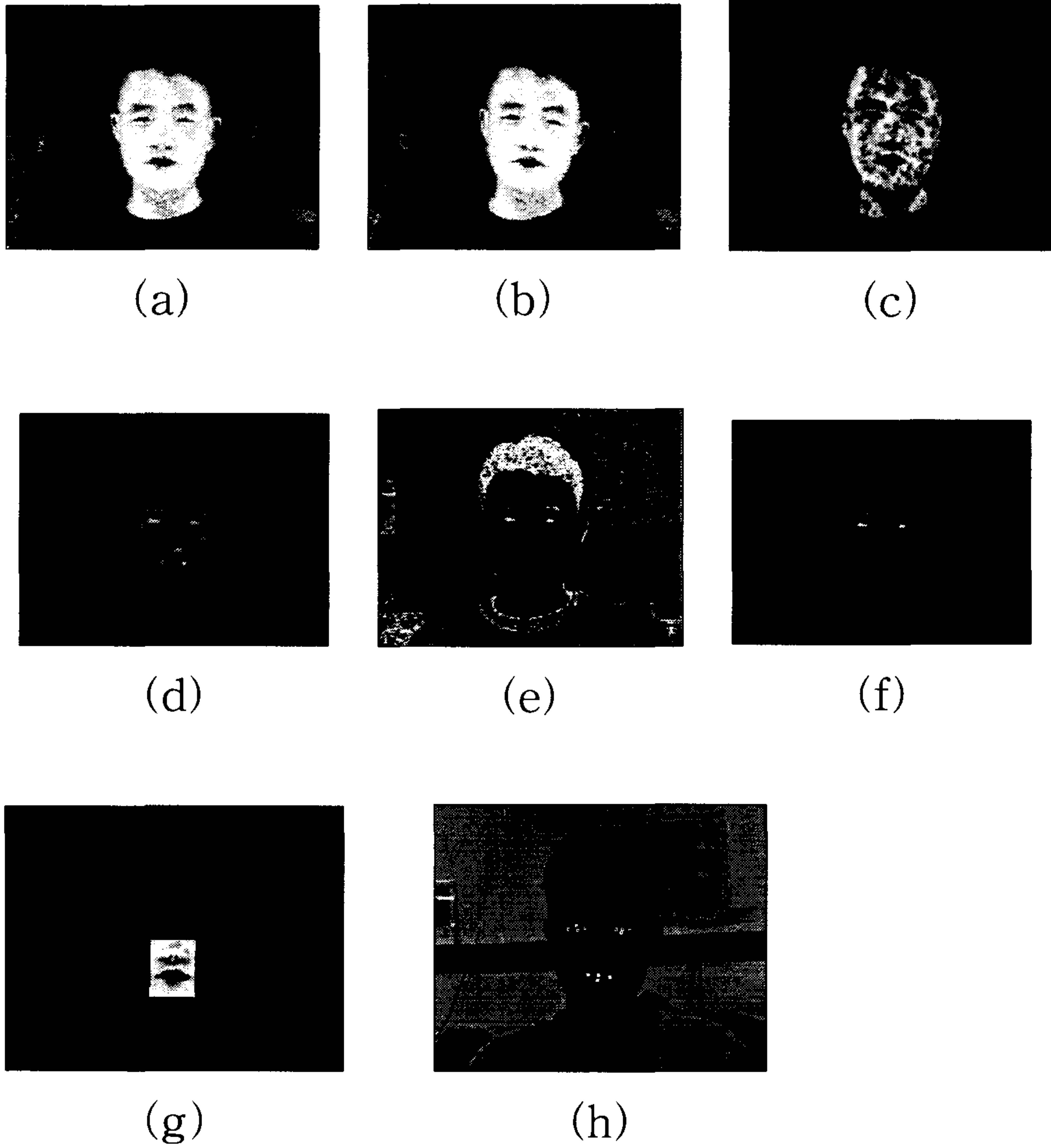
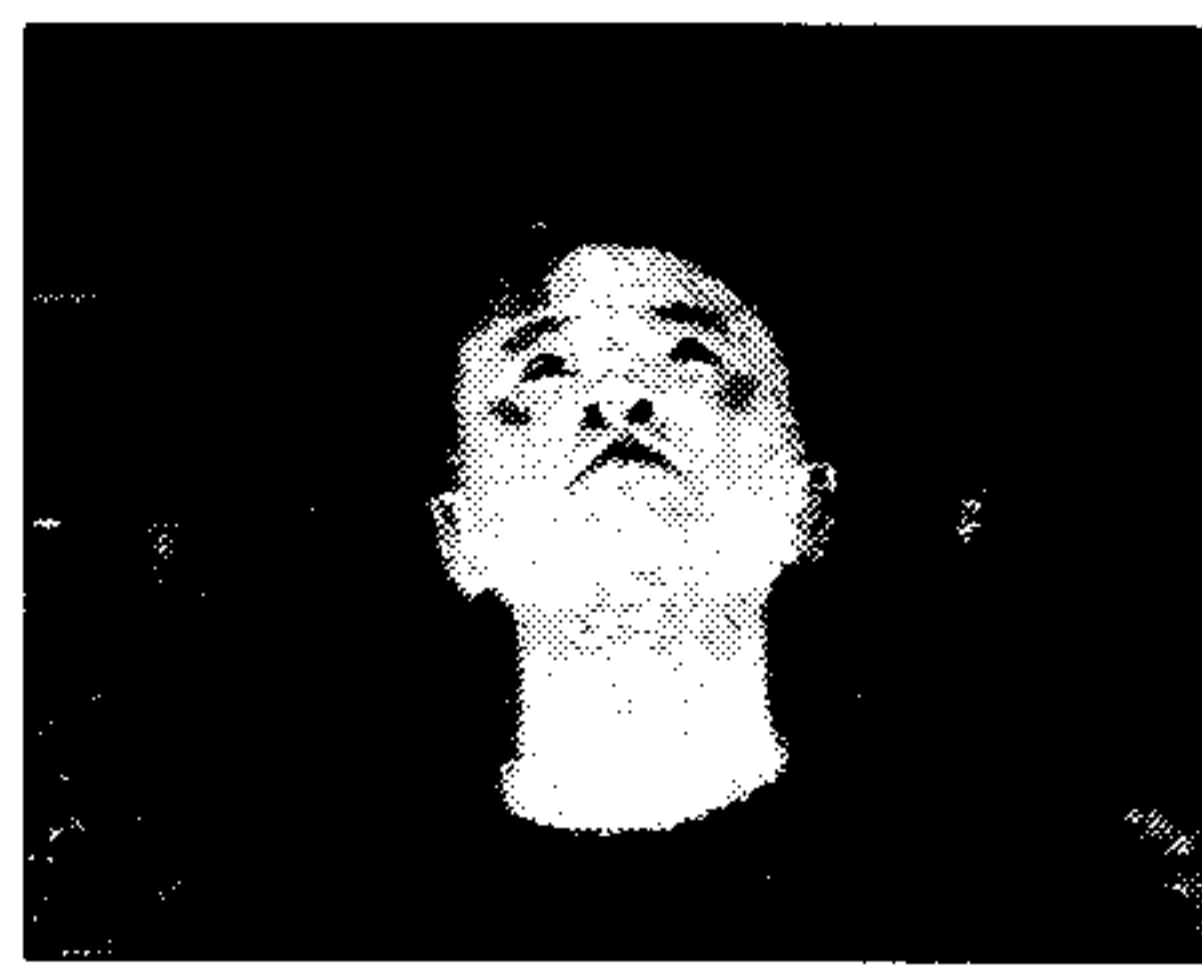


그림 4.14 정면을 바라보는 얼굴에서의 작은 움직임

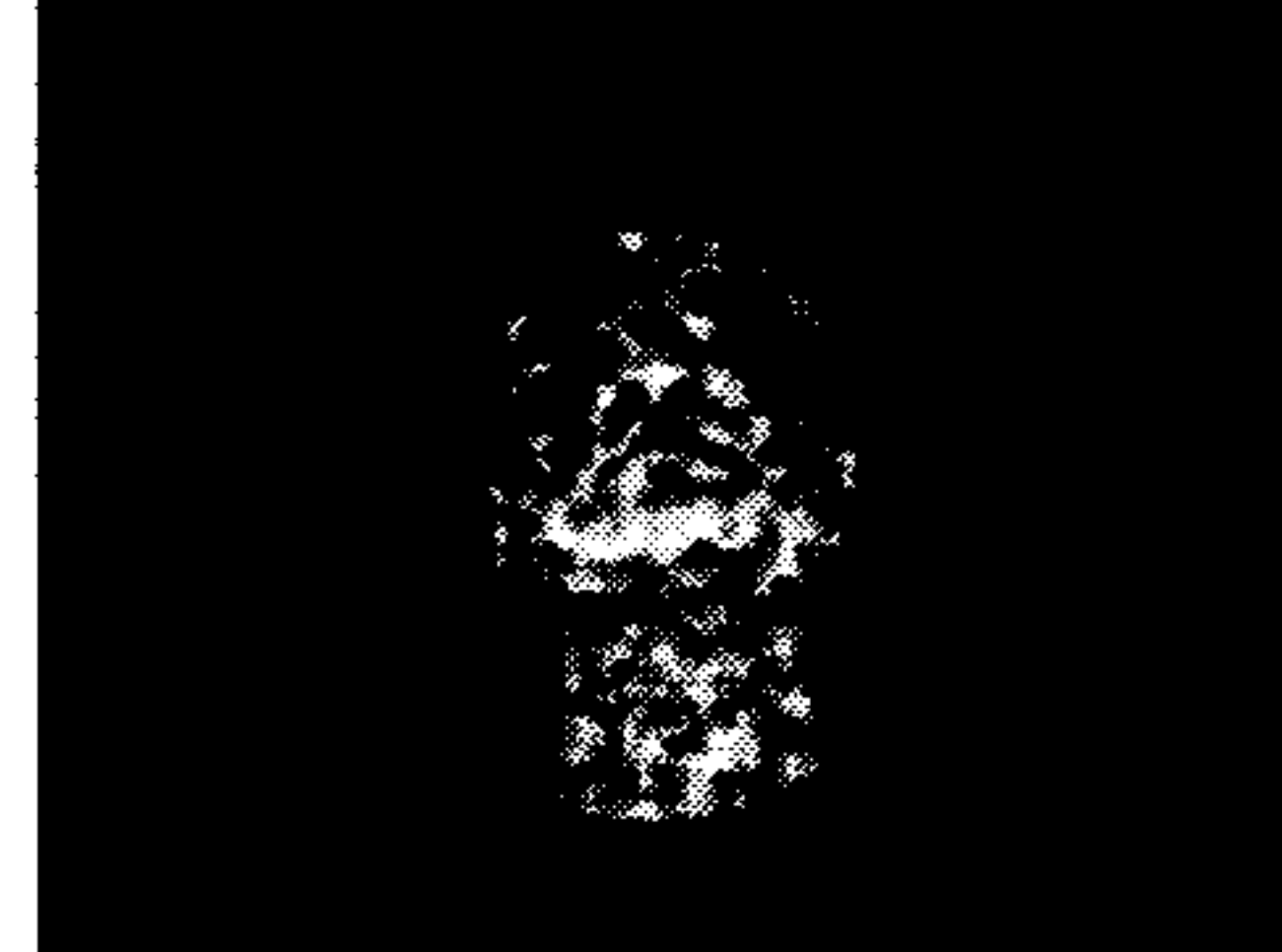
(a) GFCD 영상 1 (b) GFCD 영상 2 (c) 움직임 색상 영상 (d) Valley 추출 영상
 (e) BWCD 영상 (f) 눈 후보영역 (g) 입 후보 영역 (h) 얼굴 특징점 추출 영상



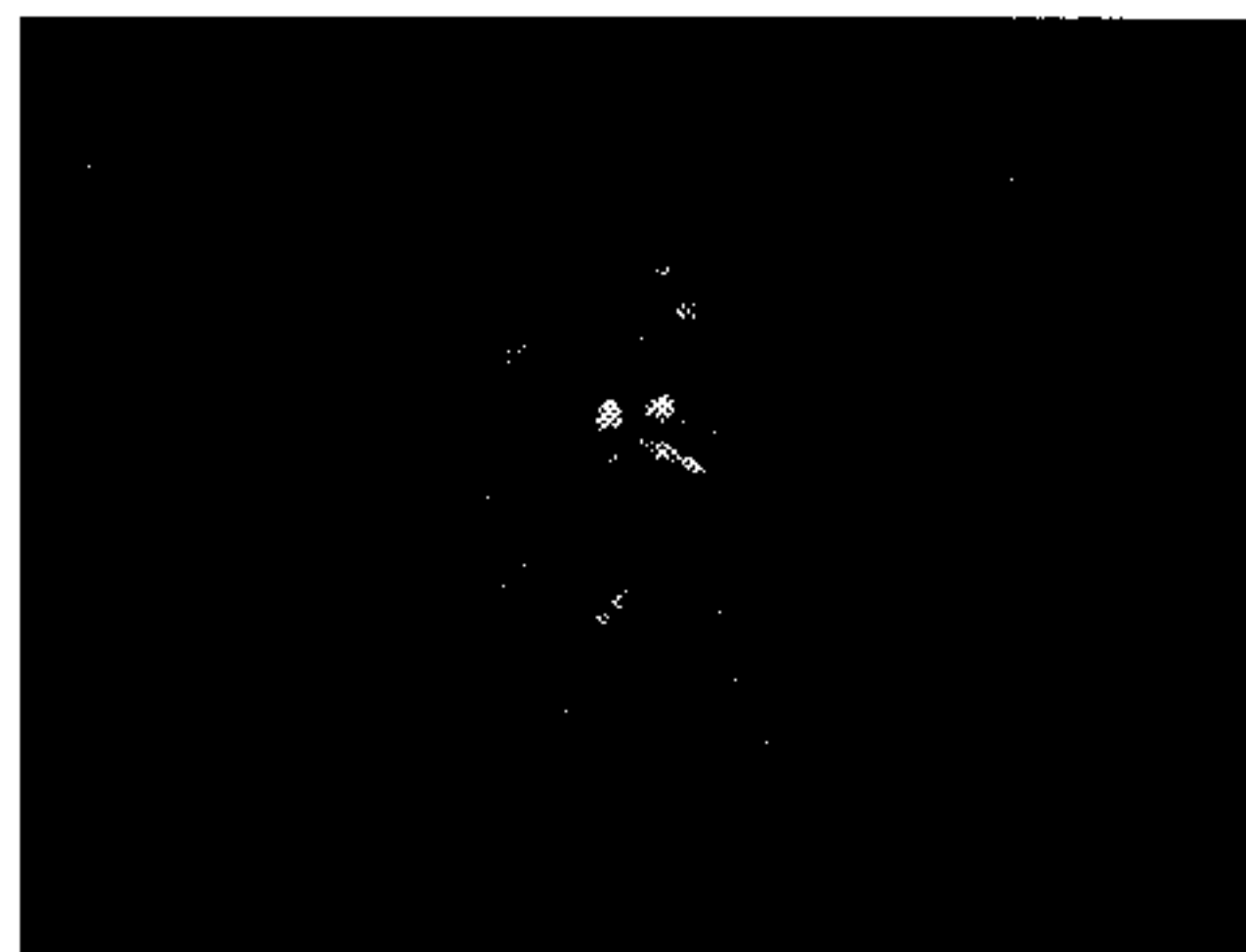
(a)



(b)



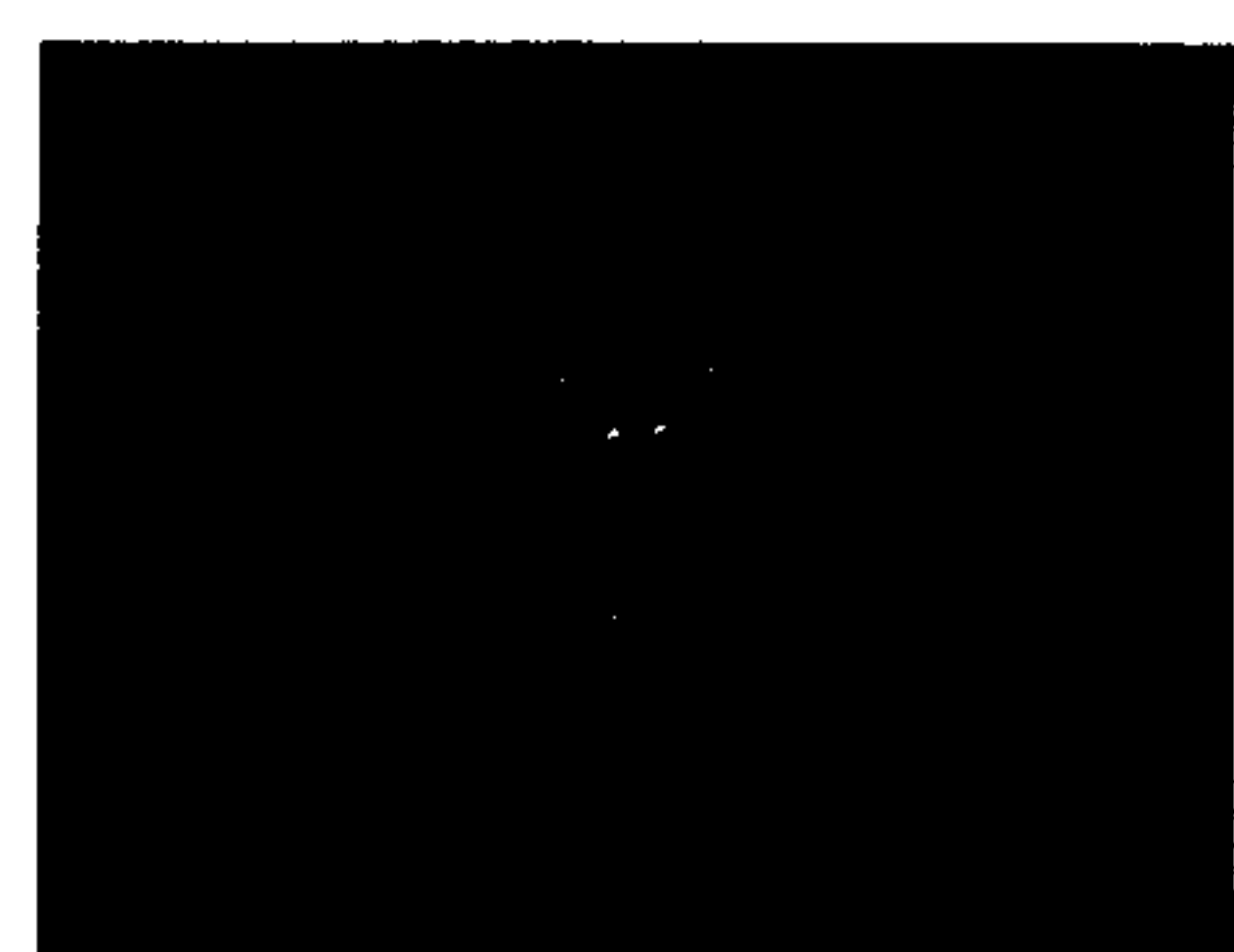
(c)



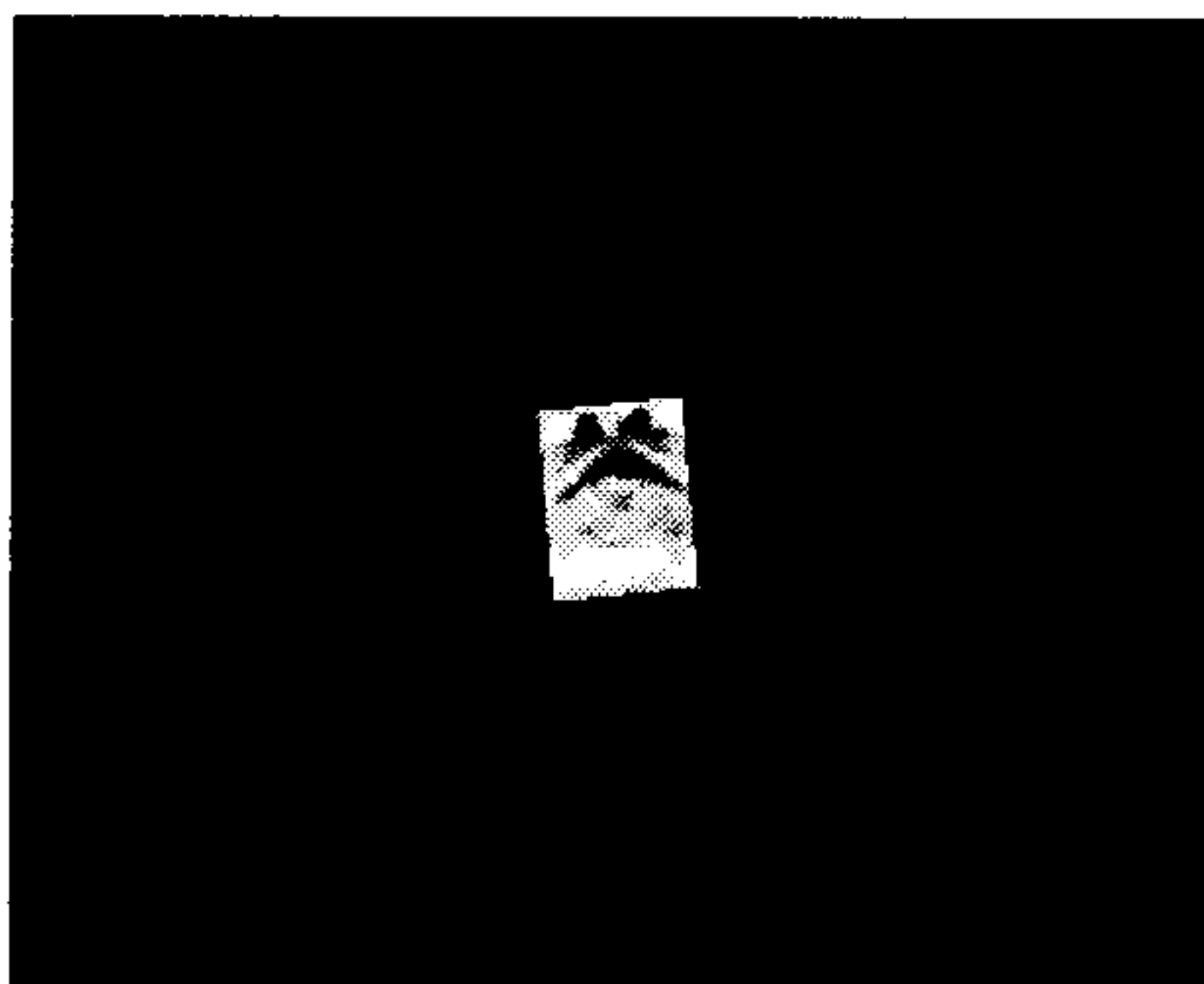
(d)



(e)



(f)



(g)



(h)

그림 4.15 위를 바라보는 얼굴에서의 작은 움직임

(a) GFCD 영상 1 (b) GFCD 영상 2 (c) 움직임 색상 영상 (d) Valley 추출 영상
(e) BWCD 영상 (f) 눈 후보영역 (g) 입 후보 영역 (h) 얼굴 특징점 추출 영상



(a)



(b)



(c)



(d)



(e)



(f)



(g)



(h)

그림 4.16 아래를 바라보는 얼굴에서의 작은 움직임

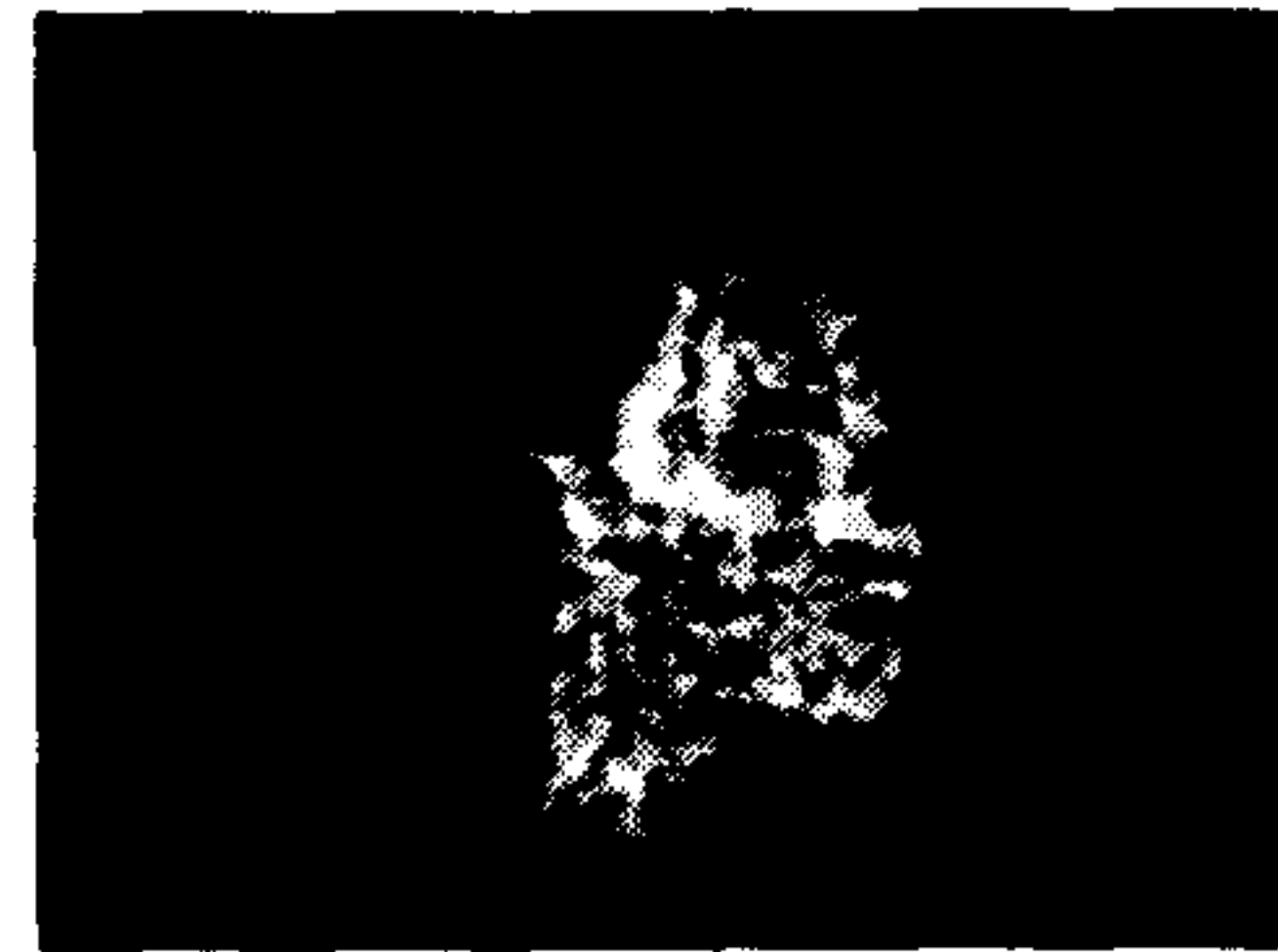
(a) GFCD 영상 1 (b) GFCD 영상 2 (c) 움직임 색상 영상 (d) Valley 추출 영상
(e) BWCD 영상 (f) 눈 후보영역 (g) 입 후보 영역 (h) 얼굴 특징점 추출 영상



(a)



(b)



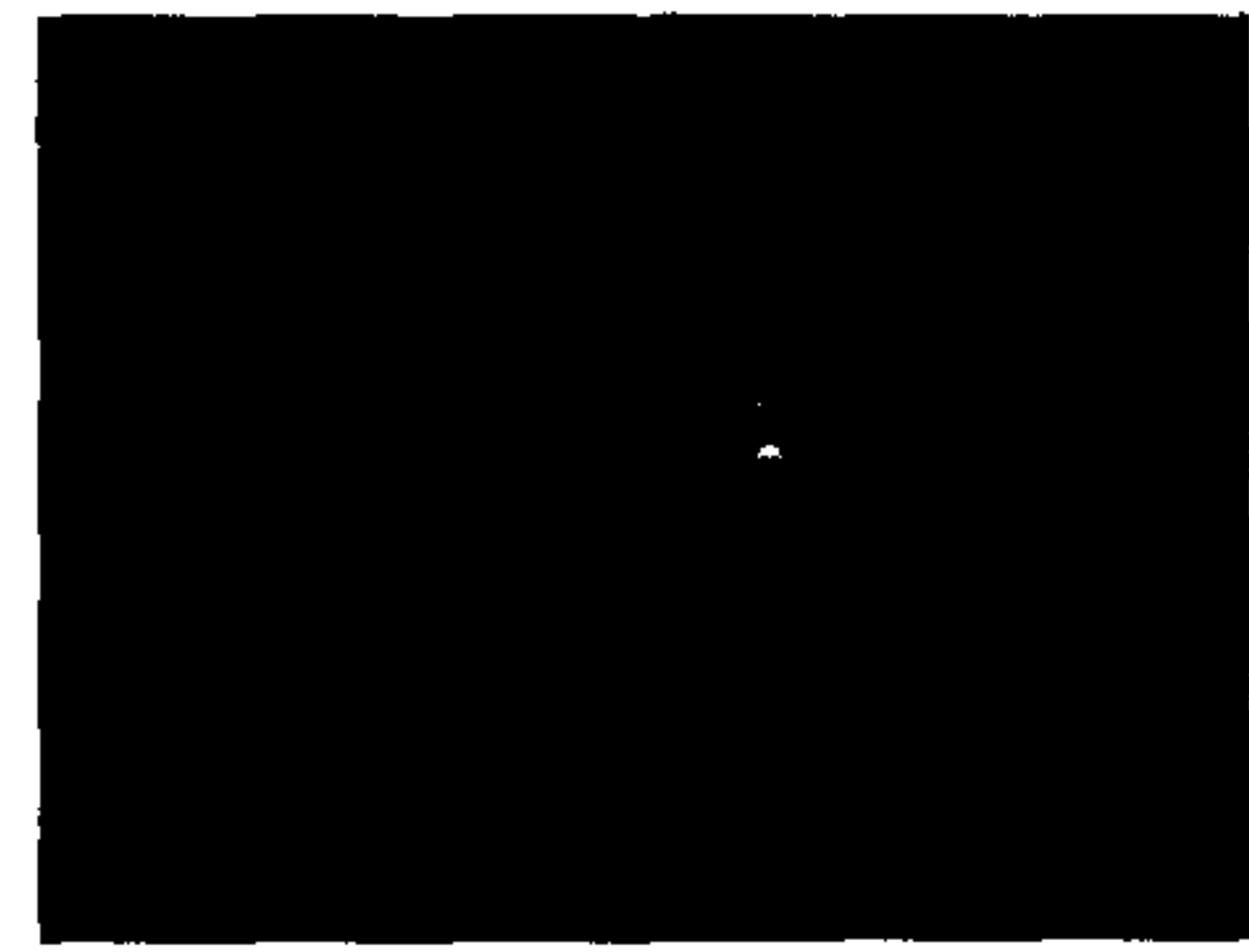
(c)



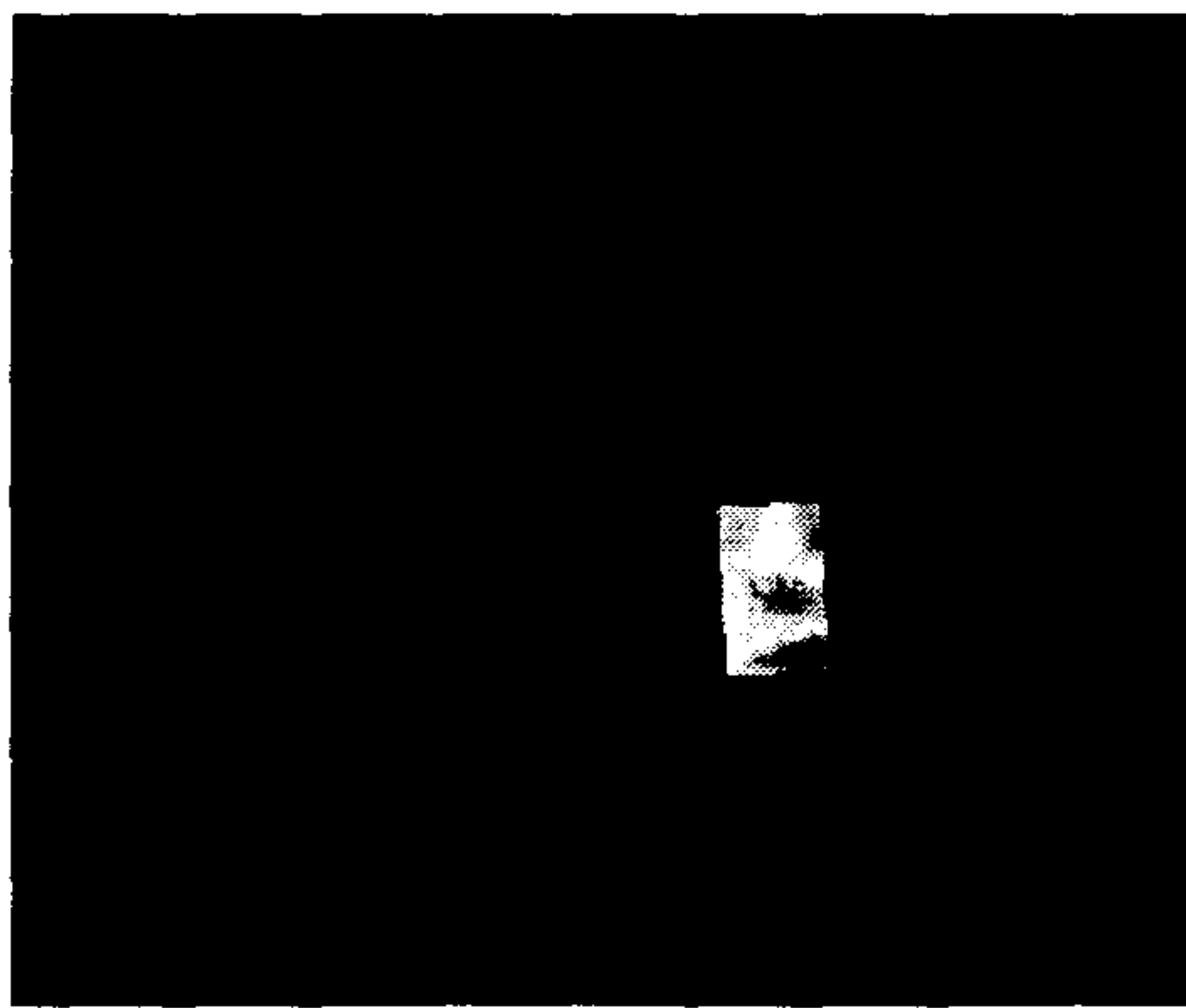
(d)



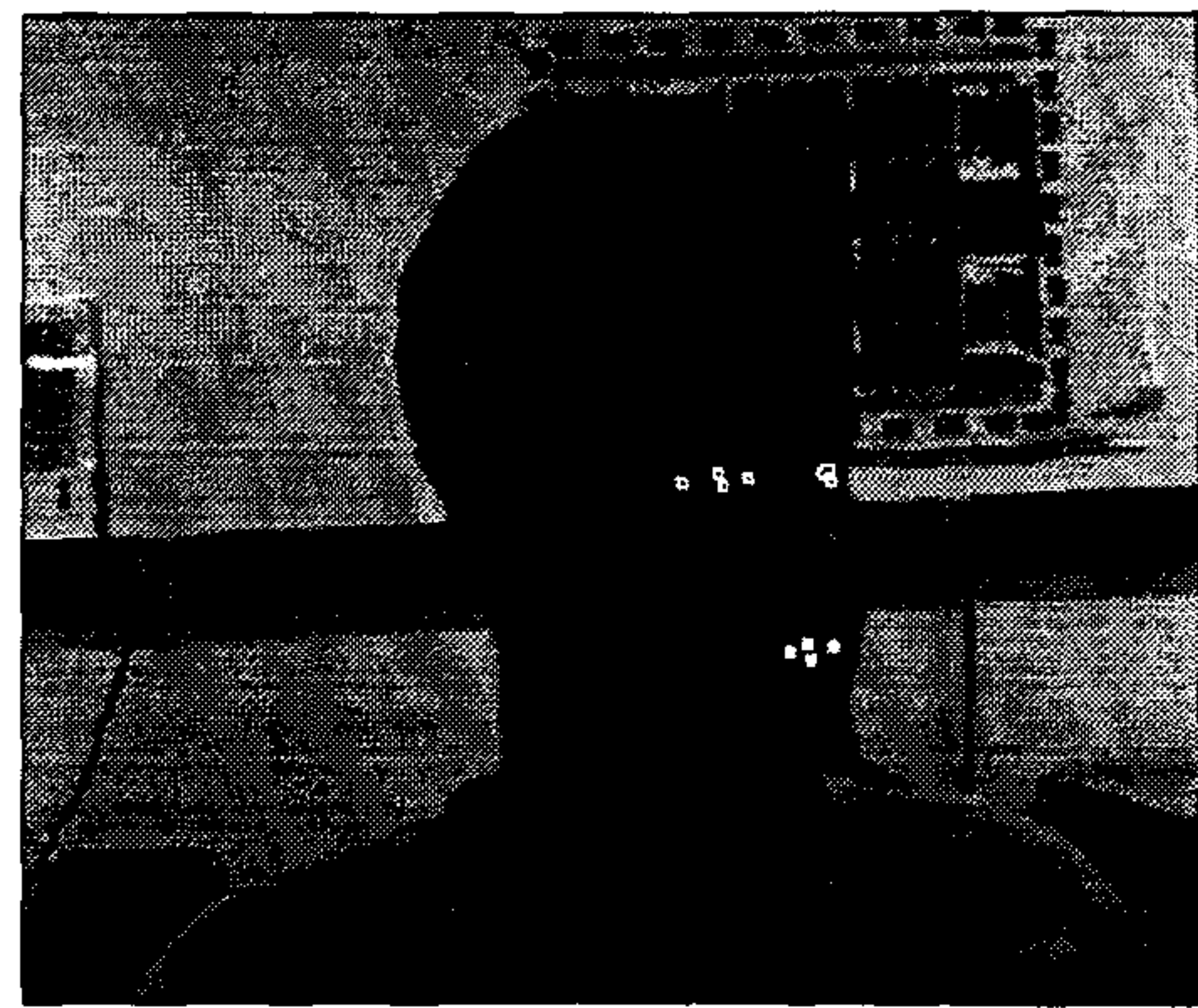
(e)



(f)



(g)



(h)

그림 4.17 왼쪽을 바라보는 얼굴에서의 작은 움직임

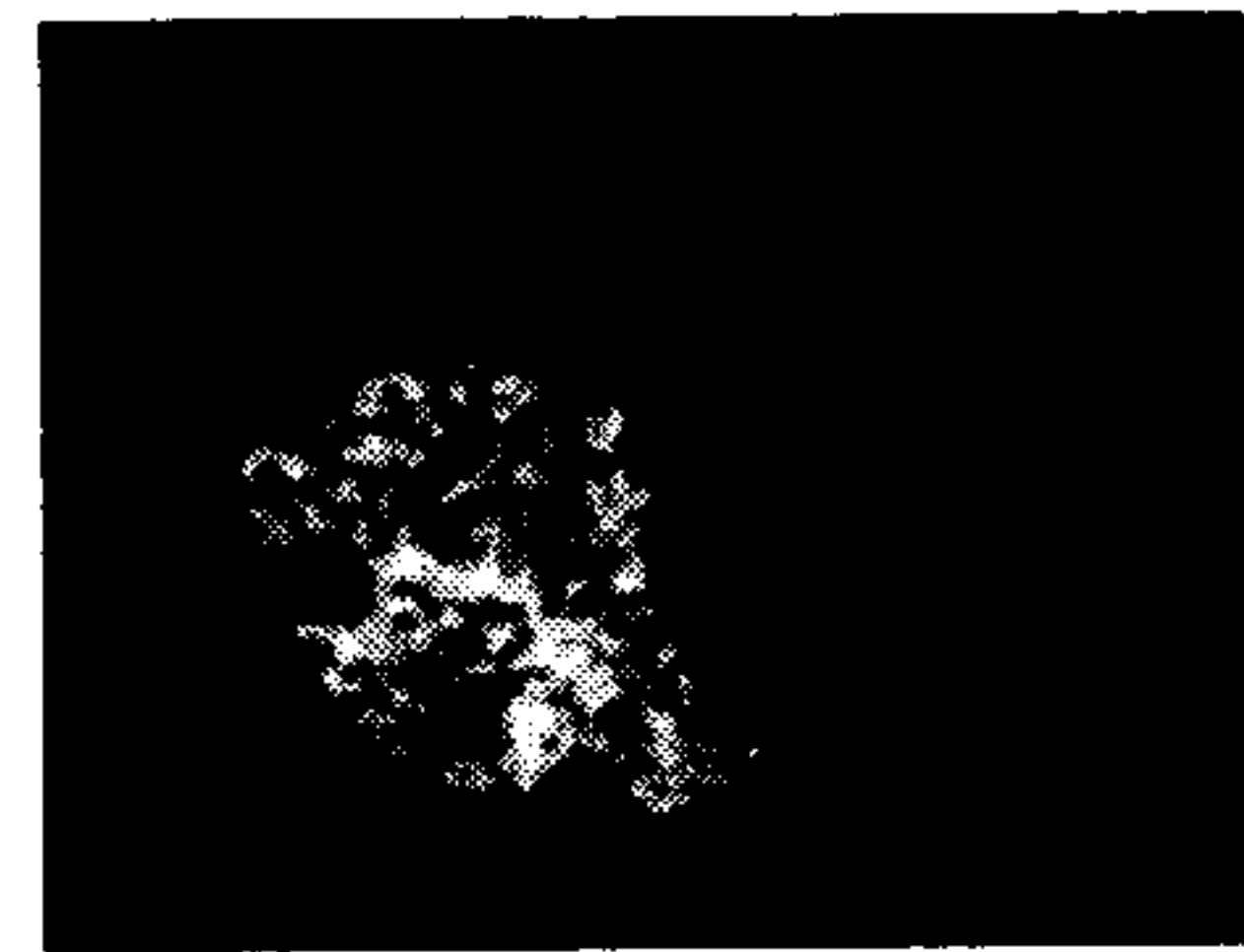
(a) GFCD 영상 1 (b) GFCD 영상 2 (c) 움직임 색상 영상 (d) Valley 추출 영상
(e) BWCD 영상 (f) 눈 후보영역 (g) 입 후보 영역 (h) 얼굴 특징점 추출 영상



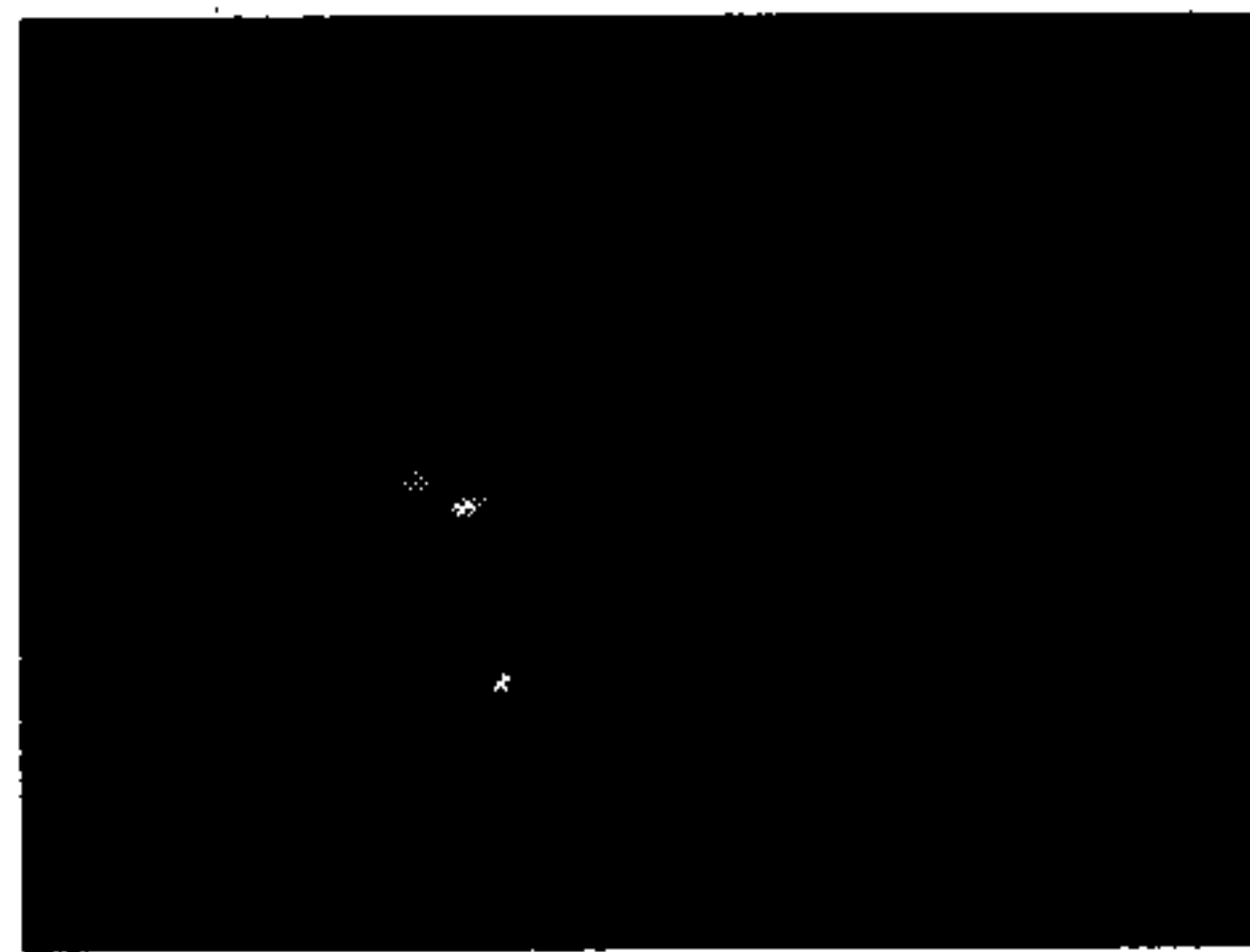
(a)



(b)



(c)



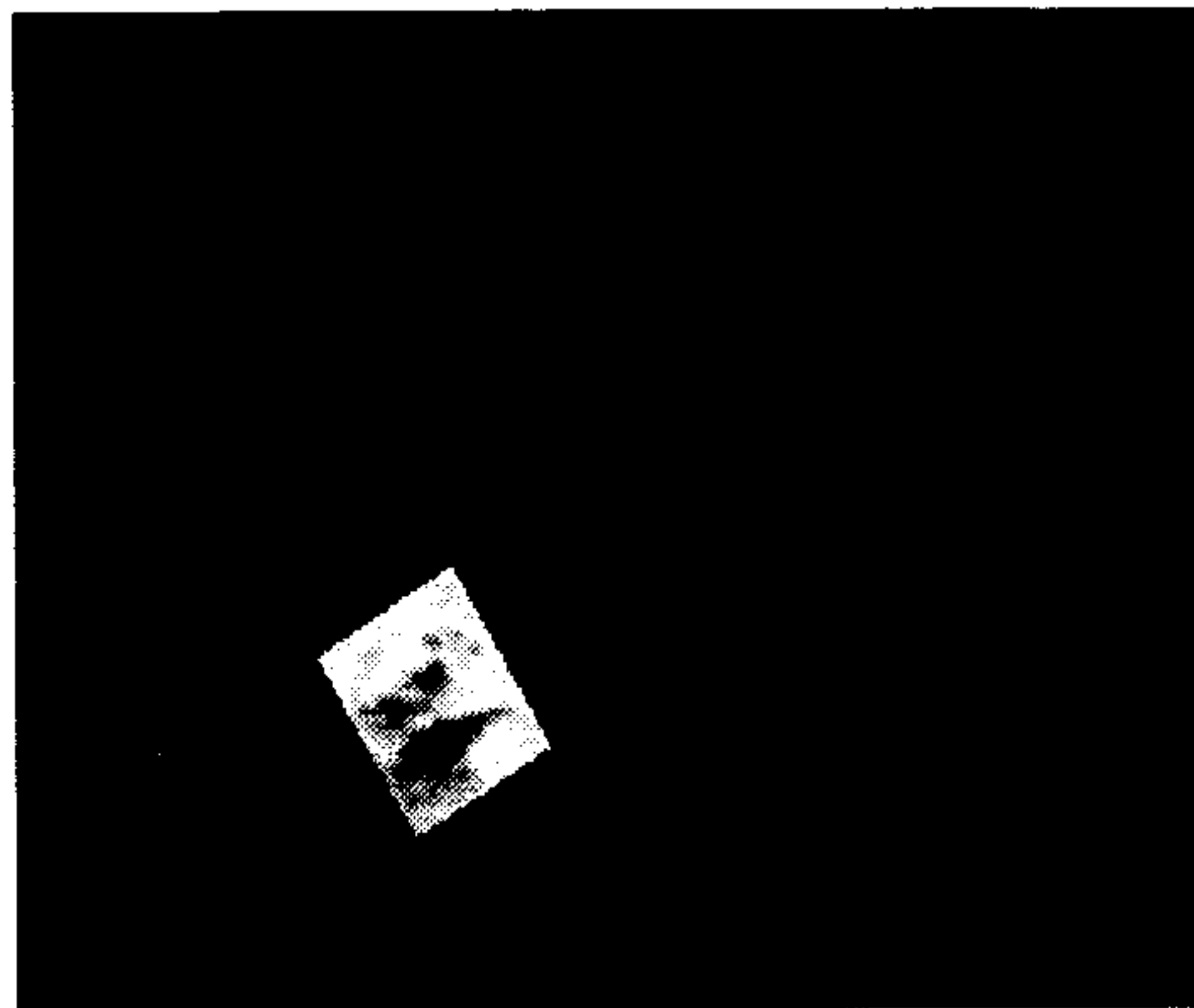
(d)



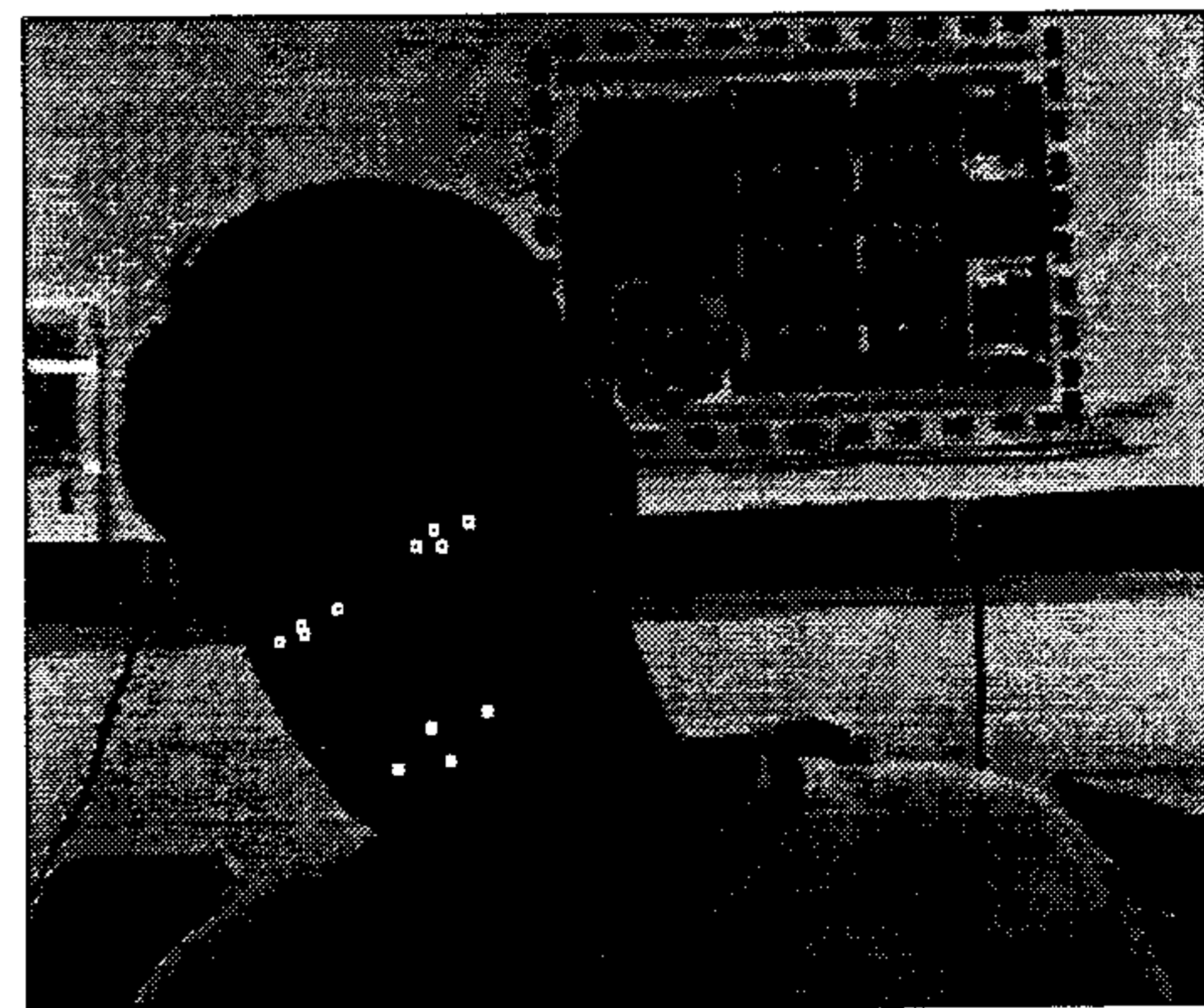
(e)



(f)



(g)



(h)

그림 4.18 기울어진 얼굴에서의 작은 움직임

(a) GFCD 영상 1 (b) GFCD 영상 2 (c) 움직임 색상 영상 (d) Valley 추출 영상
(e) BWCD 영상 (f) 눈 후보영역 (g) 입 후보 영역 (h) 얼굴 특징점 추출 영상

3절. 실시간 얼굴 추적 시스템

본 절에서는 4장 1절에서 제안한 '움직임 색상 정보 알고리즘'을 실시간으로 구현하는 시스템에 대해 설명하고자 한다. 실시간 영상 처리를 위해서는 제안한 알고리즘이 고속으로 수행될 수 있는 DSP전용 칩을 필요로 하는데 본 연구에서는 TI사의 TMS320C80 MVP (multimedia video processor)를 사용한다.

가. 시스템 개관

C80은 4개의 PP(Parallel Processor)라고 하는 정수 연산용 DSP와 1개의 MP(Master Processor)라는 부동 소수점 연산용 프로세서로 이루어져 있다. 또한 내부에는 데이터 전송을 제어하는 Transfer Controller (TC)와 비디오 신호를 다루는

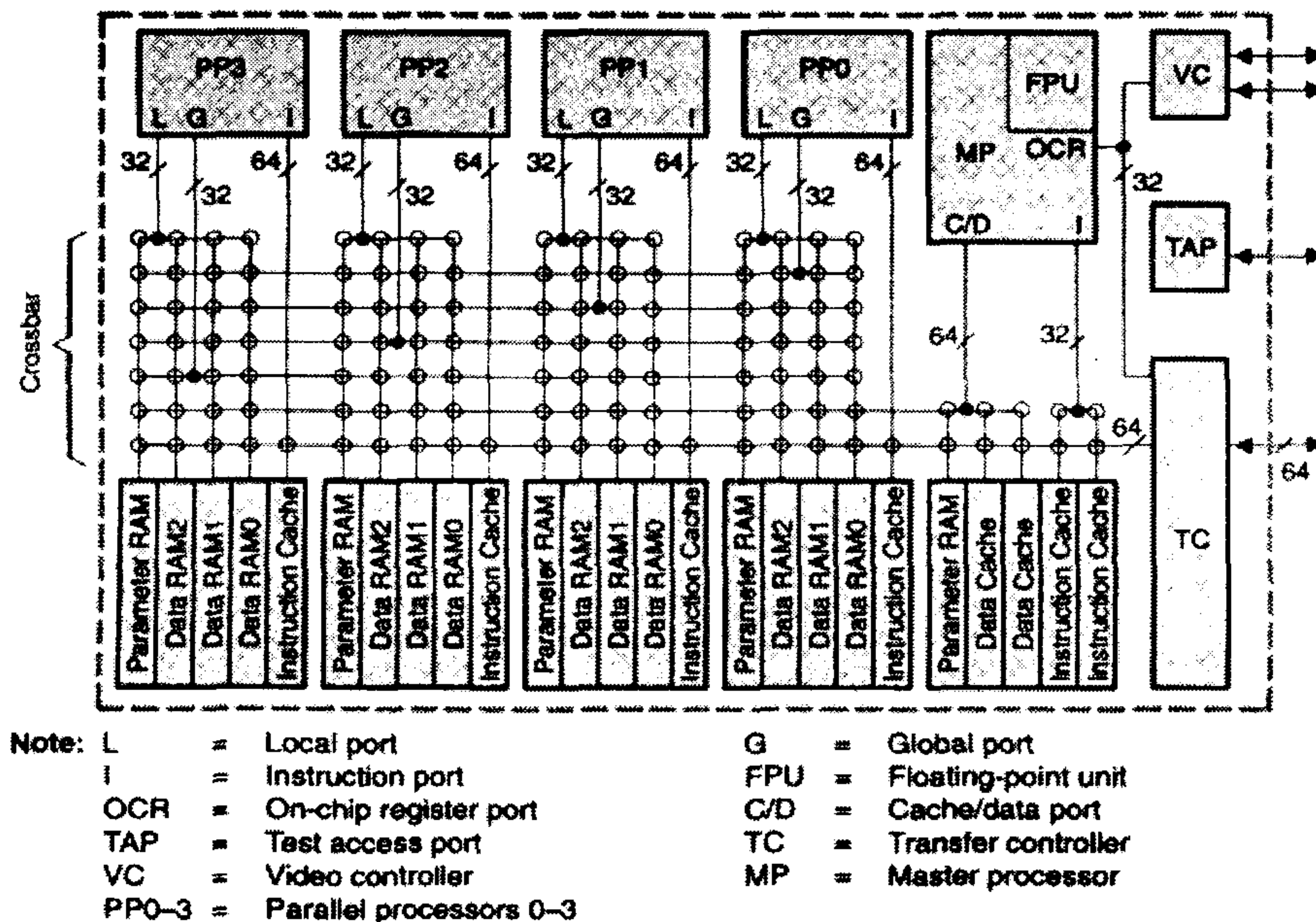


그림 4.19 C80 DSP 처리기의 내부 블록도.

Video Controller (VC)가 있으며, 50 KB 크기의 내부 SRAM도 포함하고 있다. 자세한 구조는 그림 4.3.1의 내부 블록도를 참고로 한다. C80은 영상 데이터의 빠른 연산을 위해 몇가지 특징을 가지는데, 그 중 대표적인 것은 블록 단위의 데이터 전송 즉 패킷 전송 (packet transfer)을 지원한다는 것이다. 블록의 단위는 사용자 정의가 가능하며, 특히 컨벌루션 (convolution) 이나 윤곽선 추출과 같은 $N \times N$ 윈도우 연산의 경우에 많은 처리 속도의 향상을 가져온다. TC는 이 패킷 전송의 중추로서 자체 주기를 사용해 타 장치의 간섭을 받지 않으며, 각 전송 요구의 우선순위에 따라 원활하게 칩 내부 혹은 외부 간의 데이터 전송을 가능하게 한다.

C80이 입/출력 장치와 함께 완전한 시스템으로 동작하게 하기 위해서는 여러 모듈이 필요하게 되는데 이는 Ariel사의 Griffin 보드에서 제공한다. 카메라를 통해 입력된

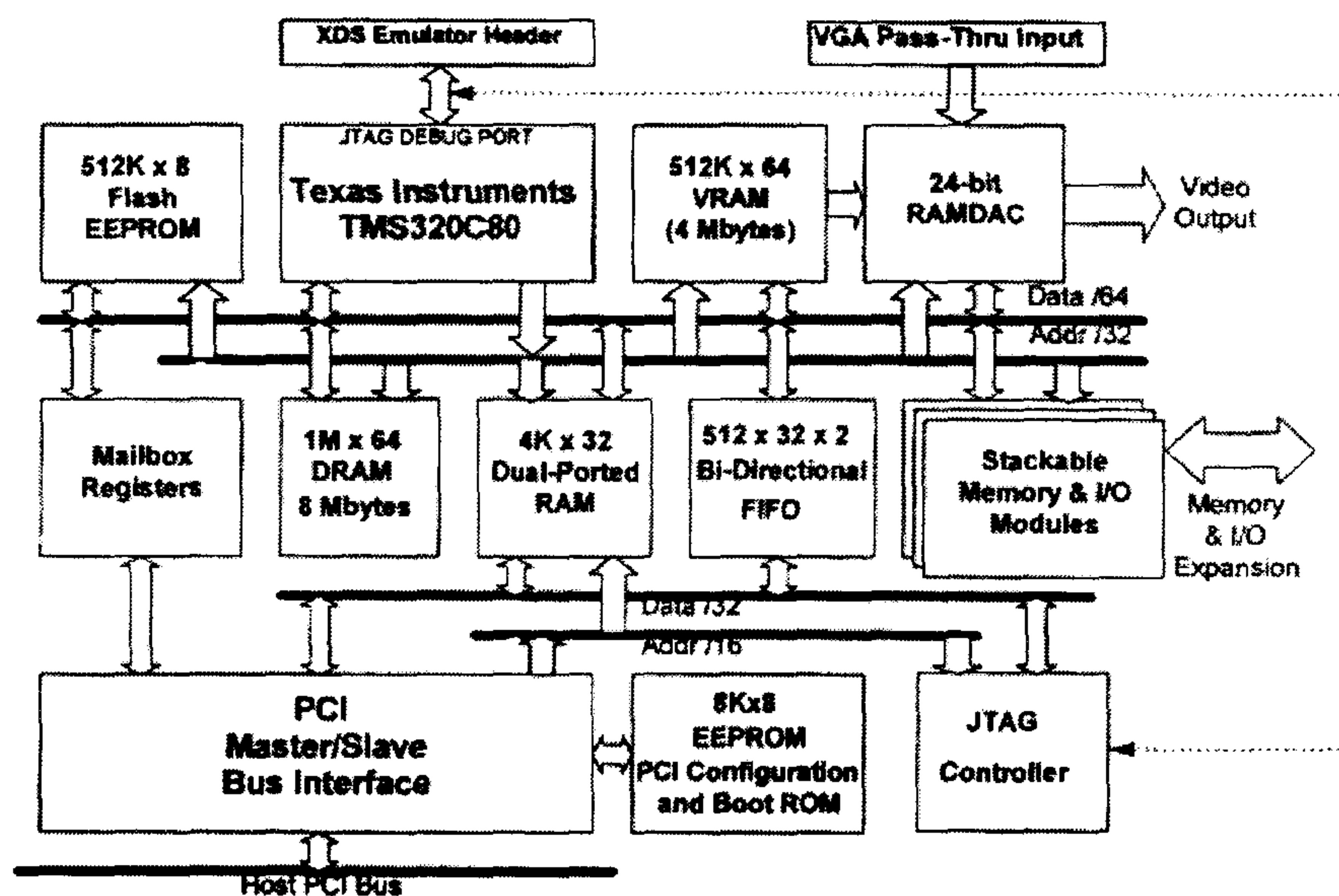


그림 4.20 Griffin 보드의 블록도.

영상은 메모리 및 입/출력 모듈을 통해 DRAM에 저장되며, C80은 이곳을 Buffer로 사용하여 각각의 연산을 수행하고 그 결과를 저장한다. 모든 연산이 끝난 데이터는 VRAM에 전송되며 RAMDAC를 거쳐 모니터에 출력된다. Griffin 보드의 전체적인 구조는 그림 4.3.2에 나타나 있다.

나. C80 환경에서의 움직임 색상 정보 알고리즘 구현

위의 4장 1절에서 제안하는 알고리즘을 C80을 이용해 실시간으로 구현하는데에는 몇가지 제약 사항이 따른다. 앞에서 언급하였듯이 C80내에서 실제의 연산을 수행하는 4개의 PP는 정수 연산용 프로세서이므로 부동 소수점의 계산이 불가능하며, 또한 지수 함수의 사용도 용이하지 않다 (실제 실험에서 색상 변환(Color transform)시에 Gaussian kernel을 그대로 사용했을 때 시스템의 처리 속도는 1/5 정도로 저하되었다). 또한 C80 내부의 메모리는 50 KB로 매우 적으므로 한번에 연산 가능한 데이터의 크기도 제한된다. 따라서 실시간 얼굴 추적 시스템으로 구현하는 알고리즘은 이러한 제한 조건을 고려하여 실제로 적용이 가능하도록 변경하여 사용해야 하며 일부는 다른 것으로 대체한다. 그림 4.3.3은 사용하는 알고리즘에 대한 블록도이다. 연속적인 두 프레임간의 차 영상은 배경이나 고정된 물체에 살색과 같은 색상이 포함된 경우, 이를 제거하기 위한 루틴이다. 현재 얼굴의 움직임이 문턱값 이하인 경우에는 이전에 찾아낸 얼굴 영역을 그대로 유지하며 그 이상이 되면 다시 현재 프레임간의 차영상을 구해 새로운 영역을 추출해 낸다. 머리카락을 검색하는 루틴에서는 손과 같은 기타 움직임이 있는 살색영역을 제거한다. 사람의 머리카락은 대부분 낮은 휘도(Luminance)를 가진다는 가정하에서 색상 변환으로 찾아낸 살색 부분의 상단에 그러한 특성을 가진 영역을 찾아낸다.

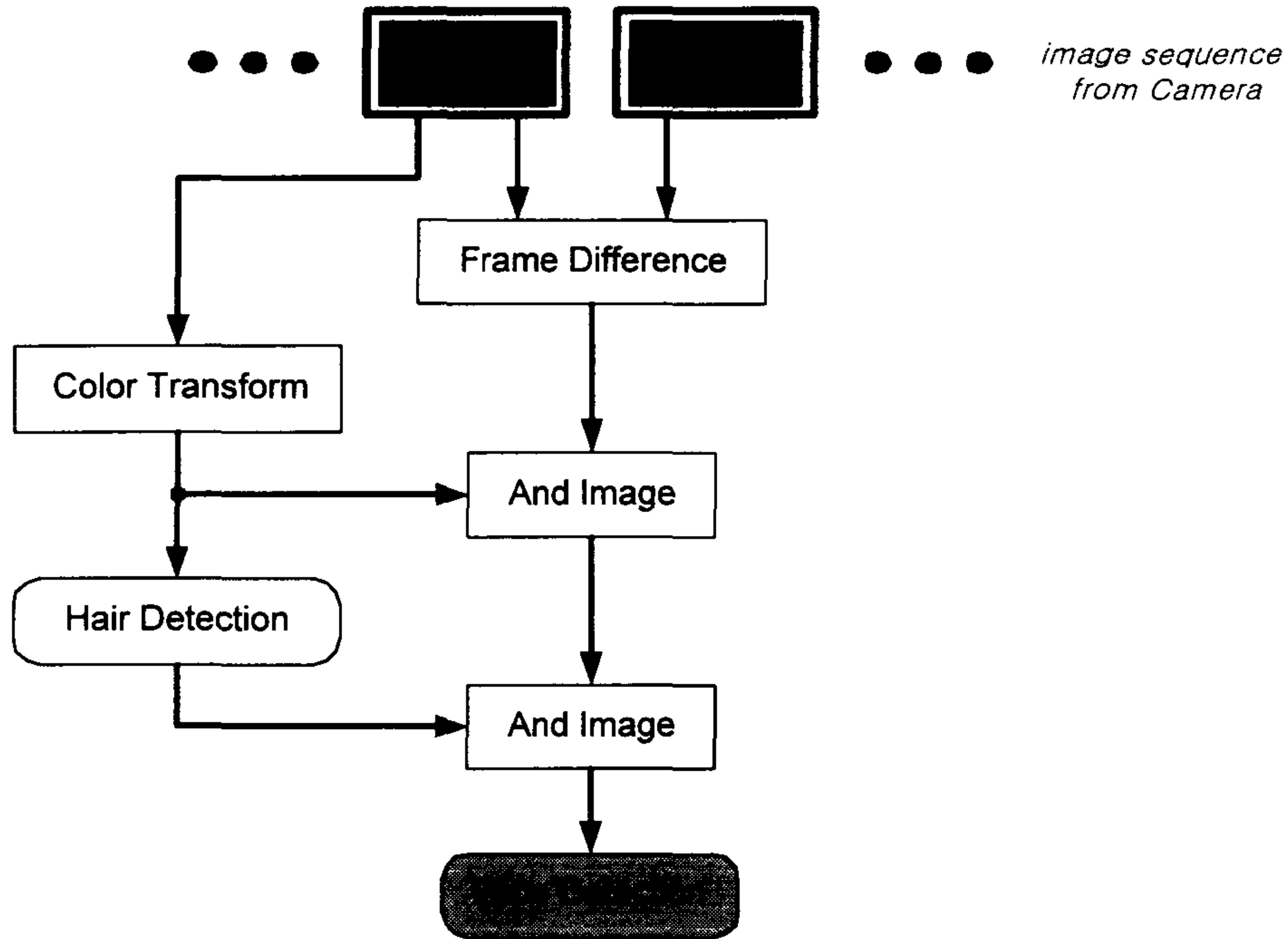


그림 4.21 실제 시스템에 적용되는 알고리즘의 블록도.

실제 시스템을 구현하는 프로그램은 MP에 사용되는 것과 PP에 사용되는 것이 독립적으로 필요하다. MP 프로그램은 C80 및 보드의 모든 모듈에 대한 초기화 및 설정을 포함하며 4개의 PP가 병렬 처리할 작업을 동일하게 할당하고 이를 제어한다. PP 프로그램은 MP가 명령한 작업들 즉 색상 변환, 차영상을 구하는 등의 실제 연산을 수행하는 부분이다. 본 실험에서는 영상을 4부분으로 균일하게 분할하여 4개의 모든 PP가 동일한 연산을 수행하게 하는 병렬 기법을 사용하였다. 또한 연산들은 앞에서 언급한 내부 메모리 및 데이터 종류를 잘 고려하고, 패킷 전송 등을 사용해 빠른 속도로 연산이 가능하도록 코딩하였다.

다. 실험 결과 및 결론

FULNIX CCD 카메라를 통해 초당 30 프레임의 속도로 얻어지는 입력 영상은 640*480의 크기로 저장되는데 실험에서는 이것을 약 4분의 1크기인 256*240으로 줄여서 사용하였다. 입력된 영상이 모든 처리를 거친 후 그 결과 즉, 추적된 얼굴 영역에 특정한 표시가 된 영상이 모니터로 출력되기까지 걸리는 시간은 약 10분의 1초 정도이다. 실제로 사람이 실시간이라고 느끼기 위해서는 초당 15 프레임 이상이 필요하게 되는데 본 연구에서 구현한 시스템은 10에서 12 프레임 정도의 성능을 보여주므로 거의 실시간에 근접했다고 볼 수 있다. 본 실험에서는 인위적으로 입력 영상에 손이 포함되게 하거나 사람과 유사한 인형을 나란히 두는 방식으로 실험 환경을 설정하였다. 앞서 언급했듯이 실시간 구현 및 MP와 PP의 특징을 고려하여 1절에서 제시한 알고리즘에 어느정도 수정을 가하였으나 얼굴 영역 추출의 정확도에 있어서 거의 근접된 성능을 보였다.

본 실험에서 얼굴이 아닌 살색을 제거하기 위해 사용한 머리카락의 검색 루틴은 머리카락이 짙은 갈색이나 검은색으로 낮은 휘도값을 가진다는 가정을 전제로 하기 때문에 금발이나 은발 등 보다 일반적인 적용을 위해서는 더 발전된 알고리즘이 필요하다. 또한 4개의 PP의 효율적인 사용 및 MP와의 역할 분담을 최적화 함으로써 처리 속도에서 상당한 향상을 가져올 것이다.

4절. 결론 및 향후 연구 방향

본 보고서에서는 복잡한 배경을 지닌 연속 컬러 영상에서 얼굴의 특징점을 추출하는 방법을 제안하였다. 조명의 영향을 줄이기 위해 정규화 색상 모델을 사용하였으며 개인의 얼굴 색상 변화를 흡수할 수 있는 일반적 얼굴색 분포를 정의하였다. 또한, 동영상에서 배경을 제거하고 얼굴 이외의 신체에 대한 움직임 배제하기 위해 얼굴색

분포를 이용한 색상변환 영상의 움직임 정보 개념을 도입하여 색상 움직임 개념을 제시하였다. 색상 움직임 정보는 요구되는 색상의 유사성 정도에 따라 움직임 정보에 가중치가 주어지며, 해당 화소의 움직임 민감도 역시 적응적으로 변화하는 움직임 측정 방법을 가능케 한다. 이 방법은 단순히 살색 영역이 아닌 임의의 색상분포에도 사용할 수 있으므로 관심 대상이 되는 색상을 가진 물체의 움직임을 알아내는데 이용할 수 있는 유용한 방법이다. 그리고 눈 영역 추출을 위해 눈의 색상분포와 눈과 눈썹에 대한 지정학적 관계를 이용하였다. 입의 영역은 얼굴 요소중 가장 색상의 변화가 심하기 때문에 눈의 위치에 기초한 입 후보 영역내의 히스토그램을 이용하였다. 색상이나 움직임 같은 연산이 간단한 방법만을 사용하고 명암 영상과 같이 얼굴의 크기 변화 등에 대비하여 여러 가지 연산을 수행하지 않으므로 고속 구현이 용이하며, 실시간 구현을 위한 지속적인 연구가 필요하다. 그리고 다양한 얼굴 표정 및 손을 이용한 제스처 인식에 대해서도 제안된 알고리즘을 적용하는 연구가 계속될 것이다.

제 4장 참고 문헌

- [1] Rama Chellappa, C. L. Wilson , and S. Sirohey, "Human and Machine Recognition of Faces: A Survey," *Proc. IEEE*, Vol. 83, No. 5, pp. 705-740, May, 1995.
- [2] Matthew Turk and Alex Pentland. "Face recognition using eigenfaces," in *Proc. Int. Conf on Patt. Recog.*, pp.586-591.1991.
- [3] A. L. Yuille., "Deformable Templates for Face Recognition:", *Journal of Cognitive Neuroscience*, 3, (1),1991.
- [4] D. Reifeld and H. Wolfson and Y. Yeshurun, "Detection of Interest Points Using Symmetry",*Proceedings of the 3rd ICCV*, p.62-65,Dec.,1990.
- [5] G. Yang, T. Huang, "Human Face Detection in a Complex Background," *Patt. Recog.*, 27/1 (1994), pp. 53-63.
- [6] R. Brunelli and T. Poggio, "Face Recognition: Features Versus Templates", *IEEE PAMI*, 15, 1042-1052, 1993.
- [7] K. Sung and T. Poggio., "Example-based learning for view-based human face detection", *TR 1521, M.I.T. AI Laboratory*, Vol. 3, Number 1 1991.
- [8] H. Martin Hunke, "Locating and Tracking of Human Faces with Neural Networks", *CMU-CS-94-155*, August 1994.
- [9] 유태웅, 오일석, "색채 분포 정보에 기반한 얼굴 영역 추출", 정보과학회 논문지(B) 제 24권 제 2호, pp. 180-192, 1997년 2월.
- [10] 정병오, 이준호, 김형곤, 고한석, "동영상으로부터 3차원 물체의 모양과 움직임 복원", 제9회 영상처리 및 이해에 관한 워크샵 발표논문집, 제주, pp.258-263, 1997년 1월.

- [11] Gloria Chow, Xiaobo Li, "Towards a System for Automatic Facial Feature Detection", *Pattern Recognition*, Vol. 26, No. 12, pp 1739-1755, 1993.
- [12] H. Gharavi, Mike Mills, "Blockmatching Motion Estimation Algorithm - New Results", *IEEE Transactions on Circuits and Systems*, Vol. 37, No. 5, May 1990.
- [13] P. K. Sahoo, S. Soltani, A. K. C. Wong, " A Suvey of Thresholing Techniques", *Computer Vision, Graphic, And Images Processing* 41, pp. 233-260, 1988.
- [14] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, "Machine Vision", pp 134-136, McGraw-Hill International Edtions.
- [15] Baback Moghaddam, Alex Pentland, "Maximum Likelihood Detection of Faces and Hands", *International Workshop on AFGR*, pp. 122-128, Zurich, 1995.
- [16] C.S.Choi, K. Aizawa, "Analysis and Synthesis for Facial Image Sequences in Model-Based Image Coding", *IEEE Transaction on CSVT*. Vol. 4, No. 3, June 1994.
- [17] L. Wiskott, J. M. Fellous, Christoph von der Malsburg, " Face Recognition and Gender Determination", *International Workshop on AFGR*, pp. 92-97, Zurich, 1995.

제 5 장 얼굴 및 제스처 인식 시스템

1절. 얼굴 인식 알고리즘

가. 얼굴 인식의 중요성

개개인의 구분과 인식 방법에 대한 연구는 보안이 중요시되는 현대에 있어서 요청되는 기술이며 따라서 많은 연구가 진행중인 연구 분야이다. 컴퓨터를 이용하여 자동적으로 개개인을 구분하고 인식하는 기술은 지문 인식, 손금 인식, 손 모양을 이용한 인식, 목소리를 이용한 인식, 서명 구분을 이용한 인식, 얼굴 인식 등과 같이 여러 가지 방향으로 전개되고 있다. 그리고 현재의 인식 기술들은 사람이 인식하는 방식과 가장 가까운 방식으로 접근해 가는 추세이다. 그런데, 위에서 열거한 여러 가지 인식 기술의 방향 중에서 사람이 인식하는 방식에 가장 가까운 방식은 개개인의 목소리 특징에 의한 음성 정보를 이용한 인식과 개개인의 다른 얼굴에 의한 시각 정보를 이용한 얼굴 인식이고, 이 두 가지 중에서 사람을 인식하는데 절대적인 정보를 제공하는 것이 시각 정보임을 경험적으로 알 수 있다. 따라서, 얼굴 영상은 컴퓨터를 이용한 사람 인식을 위한 알고리즘에 중요한 정보를 제공하고 있음을 알 수 있다. 이러한 영상을 이용한 인식 시스템은 개인용 또는 공용 컴퓨터 접근 제어, 범죄자 검색 시스템, 감시 카메라를 이용한 보안 시스템 등 여러 분야에 걸쳐 이용될 수 기술이라 할 수 있다.

나. 기존의 연구 방법

개개인의 얼굴 영상을 이용하여 그 개개인이 누구인지를 인식하는 연구는 여러

가지 방식으로 전개되고 있다. 많은 연구가 진행중이지만 지금까지 소개된 얼굴 영상을 이용한 사람 인식 방법의 분류는 크게 두 가지로 나뉘는데 신경망을 이용한 얼굴 인식과 얼굴 영상에서 얼굴의 특징점을 추출하여 데이터베이스내의 데이터와 비교하여 인식하는 방식이다. 또한 후자는 특징점을 추출하는 방식에 따라 PCA(Principle Component Analysis), Deformable Templet을 이용한 방법, Gabor wavelet을 이용한 graph matching방법 등으로 나뉜다[1].

a. PCA를 이용한 얼굴 인식

MIT에서 제안하는 PCA(Principle Component Analysis)를 이용한 얼굴 인식 알고리즘[2]은 K-L(Karhunen-Loeve) 변환[3]을 기본으로 한다. 인식하고자 하는 얼굴 데이터베이스의 영상들 모두를 K-L 변환을 하여 Eigenvector를 구하고, 이 데이터베이스에서 구한 Eigenvector들의 Eigenvalue의 가중치 조합으로 데이터베이스를 재구성한다. 그리고 데이터베이스 크기 감축과 인식과정에서의 속도 향상을 위하여 Eigenvalue 중 가장 큰 값부터 가장 데이터베이스를 잘 표현하는 몇 개만을 취하여 얼굴 영상 데이터베이스 구축한다. 인식 과정에서도 위의 같은 과정을 거쳐 PCA 특징점 비교를 통하여 수행한다 . 이 방법의 특징은 얼굴에서의 비교를 위한 특징점 추출이 빠르고 계산이 복잡하지 않다는 점이며, 대량의 데이터베이스를 압축하는 효과를 볼 수 있다는 장점이 있다. 단점으로는 얼굴 영상 데이터베이스를 만드는 데 있어서 얼굴 영상의 정규화를 정확히 하여야 한다는 점과 얼굴 영상의 조명에 의해 영향을 많이 받는다는 점이다. 이 방법이 본 보고서에서 제안하는 얼굴 인식의 기본이 되는 방법이다. 자세한 알고리즘 설명을 2절에서 설명한다.

b. Templet Matching 이용한 얼굴 인식

Deformable Templet을 이용한 방법[4][5]은 사람 얼굴의 특징점을 얼굴의 성분 요소 즉 얼굴 윤곽선, 눈, 입 등의 크기와 모양을 변수 화하여 인식하는 방법이다. 얼굴의 윤곽선은 머리 형태와 얼굴 영상의 보이는 면에 따라서 변화가 심하여 잘 쓰이지 않고, 얼굴 성분 중에서 코는 정면상을 위주로 하는 인식에서는 특징점으로 사용되기 힘들다. 따라서 얼굴의 성분 요소 중에서 가장 잘 많이 이용하고 있으며, 변화가 적은 특징점으로 이용할 수 있는 얼굴 요소가 눈과 입이다. 눈의 특징점으로서의 변수로 사용되는 것은 눈 모양인 타원의 아래, 위 타원의 곡률, 눈 내의 홍채의 중심 위치와 반지름, 눈의 기울어짐 정도의 각도 등이다. 인식 과정은 templet을 얼굴 요소(눈과 입 등)의 모양과 유사하게 초기화하고 이 초기화된 templet을 이동, 회전, 확장, 축소 등의 변경을 통하여 개개인의 얼굴 성분 요소의 모양에 맞추어 나간다. 얼굴 요소 성분에 가장 잘 맞는 위치에서의 변수값으로 특징값을 정하고, 이 특징값의 비교를 통하여 인식을 한다.

C. Gabor Wavelet을 이용한 얼굴 인식

Gabor wavelet을 이용한 얼굴 인식 알고리즘[6]에서의 특징점 추출은 2-D Gabor 함수식 5.1과 같고

$$g(x, y : u_0, v_0) = \exp(-[x^2/2\sigma_x^2 + y^2/2\sigma_y^2] + 2\pi i [u_0x + v_0y]) \quad (5.1)$$

그 함수에 에 방향성과 Scaling 성분을 추가한 식은 각각 식 5.2와 식 5.3과 같은 함수가 되고 이 두 함수로 이루어진 2-D 필터를 이용하여 얼굴 영상내의 곡률이 높은 곳(눈가, 코 끝 등)을 추출해낸다.

$$\begin{aligned} \Phi_{\lambda}(x, y, \theta) &= \exp[(-\lambda^2(x'^2 + y'^2)) + ipix'] \\ x &= x\cos\theta + y\sin\theta \\ y &= -x\sin\theta + y\cos\theta \end{aligned} \tag{5.2}$$

(θ : Spatial Orientation , λ : aspect ratio of the Gaussian)

$$\begin{aligned} \Phi(\alpha^j x, \alpha^j y, \theta_k), \alpha \in R, j = \{0, -1, -2, \dots\} \\ \theta_k = k\pi/4, k = \{0, 1, 2, 3\}, \alpha^j, j \in Z \end{aligned} \tag{5.3}$$

이렇게 추출한 얼굴 영역내의 점들을 노드로 하고 이 노드들로 연결한 것을 메쉬라 정의하여 개개인마다의 메쉬들의 상호 연결관계를 개개인 구분과 인식을 위한 특징점으로 이용한다. 그리고 인식 과정에서 데이터베이스와 입력 영상과의 비교는 Graph Matching 방법을 제시하고 이 방법을 이용하여 얼굴 인식을 수행한다.

2절. PCA 특징값을 이용한 얼굴 인식

본 보고서에서 제시하는 얼굴 인식 알고리즘은 앞 절에서 소개한 얼굴 인식 알고리즘중에서 PCA를 이용한 얼굴 인식 알고리즘을 기본으로 한다. 그리고 영상 장비들의 발전과 영상 정보의 고급화에 맞추어 영상 데이터들이 칼라화를 지향하는 현재의 추세에 맞추어 기존의 PCA 방식이 256의 값을 가질 수 있는 흑백 영상을 대상으로 제한한 알고리즘이나 이를 본 보고서에서는 확장하여 칼라 영상에서 적용하였다. 그리고 기존의 PCA 알고리즘의 단점인 얼굴 영상 데이터베이스의 정규화 문제를 다음과 같이 해결하였다. 기존의 PCA를 이용한 얼굴 인식에 대한 연

구에서는 이 알고리즘을 적용하기 위해서 얼굴 영상 데이터베이스 구축 과정에서의 얼굴 영상의 정규화 과정을 수작업으로 대부분 수행하였으나 본 보고서에서는 얼굴 영상의 칼라 정보를 이용하여 정규화를 자동적으로 수행함으로써 문제시되었던 부정확한 정규화 문제도 동시에 해결하였다.

가. 연구 대상

본 연구의 목표인 얼굴 인식 대상은 복잡하지 않는 배경이 존재하고 얼굴 영역을 포함하는 영상을 대상으로 한다. 즉 보통의 감시 칼라 카메라(CCD Color Camera)의 환경을 대상으로 한 연구이다.

나. 얼굴 영역 검출

입력 영상을 그림 5.2.1과 같은 보통의 감시 카메라 환경을 기준으로 함으로서 얼굴 인식을 수행하려면 2가지의 전처리 과정이 필요하다. 첫 번째는 입력 영상에서 얼굴 영역을 검출하는 과정이 필요하며 두 번째는 이렇게 검출된 얼굴 영역을 인식과정의 입력으로 사용하기 위한 정규화 과정이 필요하다. 정규화는 인식하고자 하는 얼굴을 데이터베이스화한 조건과 같은 조건으로 변경을 하는 과정을 의미한다. 이 과정은 영상을 이용한 인식의 어떠한 알고리즘이든지 필요한 과정이다.

입력 영상을 본 연구에서는 앞에서 제시한 바와 같이 칼라 영상으로 하고 있으며 얼굴 영역 검출에서도 그림 5.2.2와 같이 얼굴 색상의 칼라 정보를 활용하는데, 본 보고서의 4장의 1절 색상 정보 알고리즘에서 제시한 컬러 변환을 이용하여 그림 5.2.2 와 같이 얼굴 색상 정보가 많은 곳을 높은 명암 값을 가지게 하고 얼굴 색상과 다른 색상을 가질수록 어두운 명암 정보를 가지게 한다. 이렇게 하여 얼굴 영역을 검출하여 낸다. 자세한 알고리즘은 4장 1절을 참조한다.



그림 5.1 입력 영상

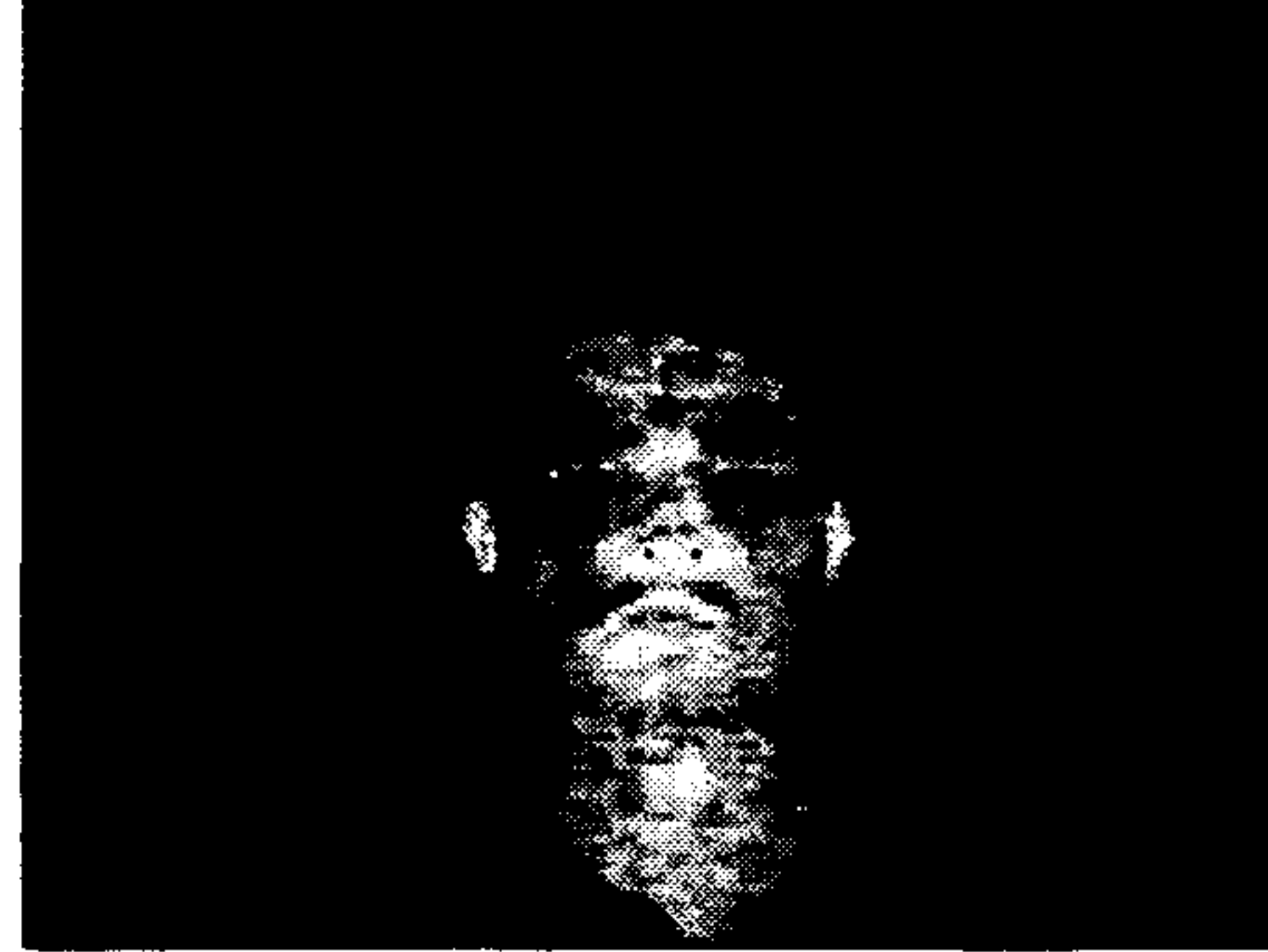


그림 5.2 컬러 변화된 입력 영상



그림 5.3 추출된 얼굴 영역



그림 5.4 칼라 변환된 얼굴영역

얼굴의 정규화 과정은 그림 5.3 과 같은 검출된 얼굴 영역을 기반으로 그림 5.4와 같은 배경에서 분리해 낸 영상을 미리 정해 놓은 데이터베이스의 영상 크기와 조건에 맞게 영상을 변경시키는 과정이다. 데이터베이스내의 인식 대상의 수로 인한 데이터베이스의 크기를 고려하여 얼굴 영상을 60*60 의 비로 데이터베이스화하였다. 정규화 방법은 검출된 얼굴 영상은 얼굴 색상 정보만을 가지고 검출하였으므로 얼굴 색을 포함하지만 얼굴 영역이 아닌 얼굴 이외의 영역을 분리해내는 과정이 필요하다. 왜냐하면 그림 5.4와 같이 얼굴 이외의 영역인 목부분이 포함 될 수 있기 때문이다. 그리고 기존의 사람의 얼굴에 대한 연구를 참조하면 평균적으로 사람

의 얼굴은 가로 세로의 비가 1:1.5를 가진다고 알려져 있다. 그런데 여기서 1:1.5의 비는 얼굴 영역 중에서 머리카락으로 덮여 있는 부분까지 포함한 가로와 세로의 비이고 본 연구에서는 얼굴 영역 검출 시에 얼굴 색상 정보만으로 얼굴 영역을 검출하므로 머리카락으로 덮여 있는 머리 영역을 제외되게 된다. 따라서 이 영역을 감안하여 정규화를 위한 얼굴 모델을 그림 5.5와 같이 만들어 가로 길이를 기본으로 하여, 가로와 세로의 비를 1:1 (영상 크기 : 60*60)로 정규화 하면 그림 5.7과 같은 얼굴 인식에 필요한 정확한 얼굴 영역만을 구할 수 있다.

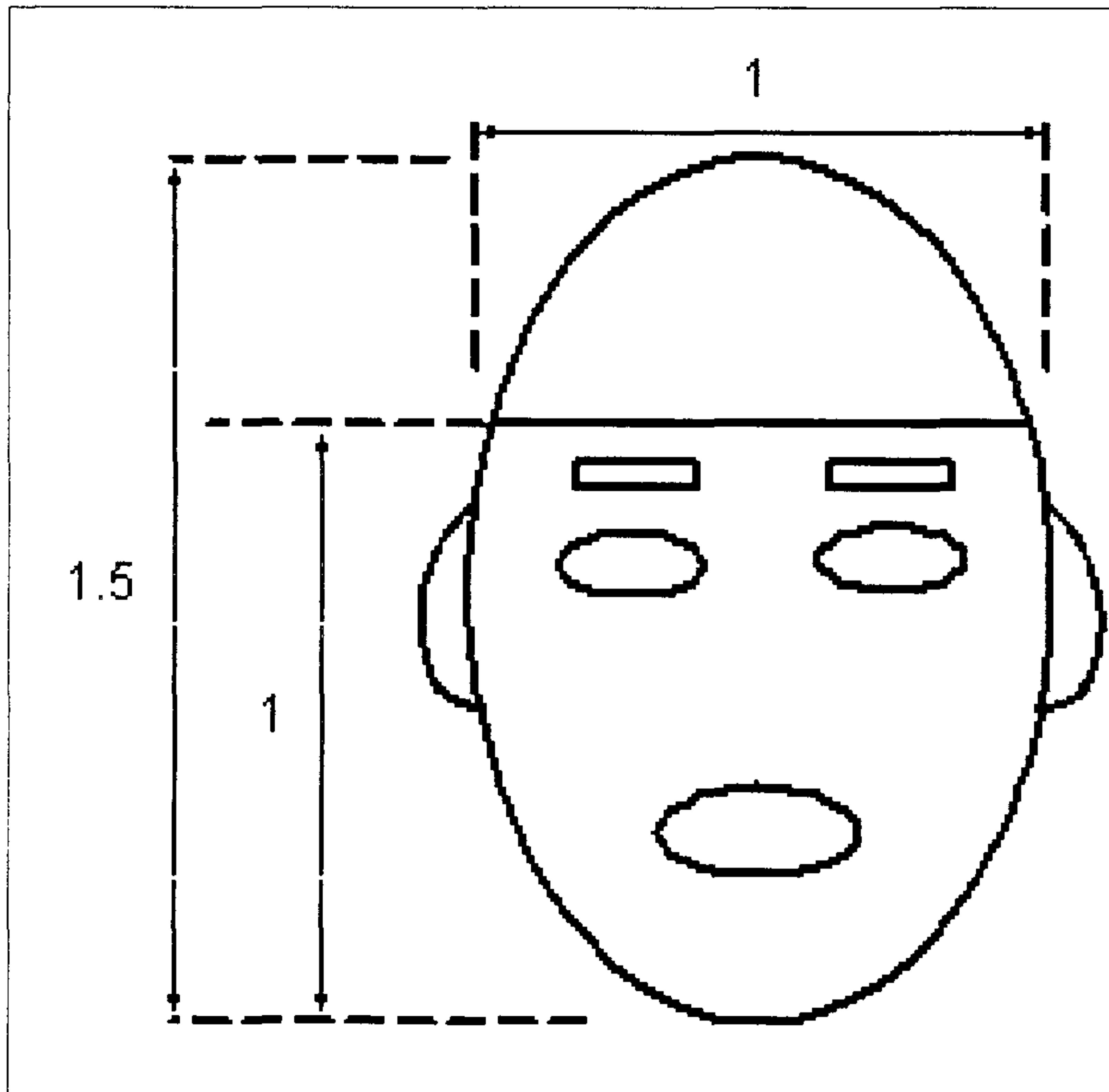


그림 5.5 얼굴 정규화 모델

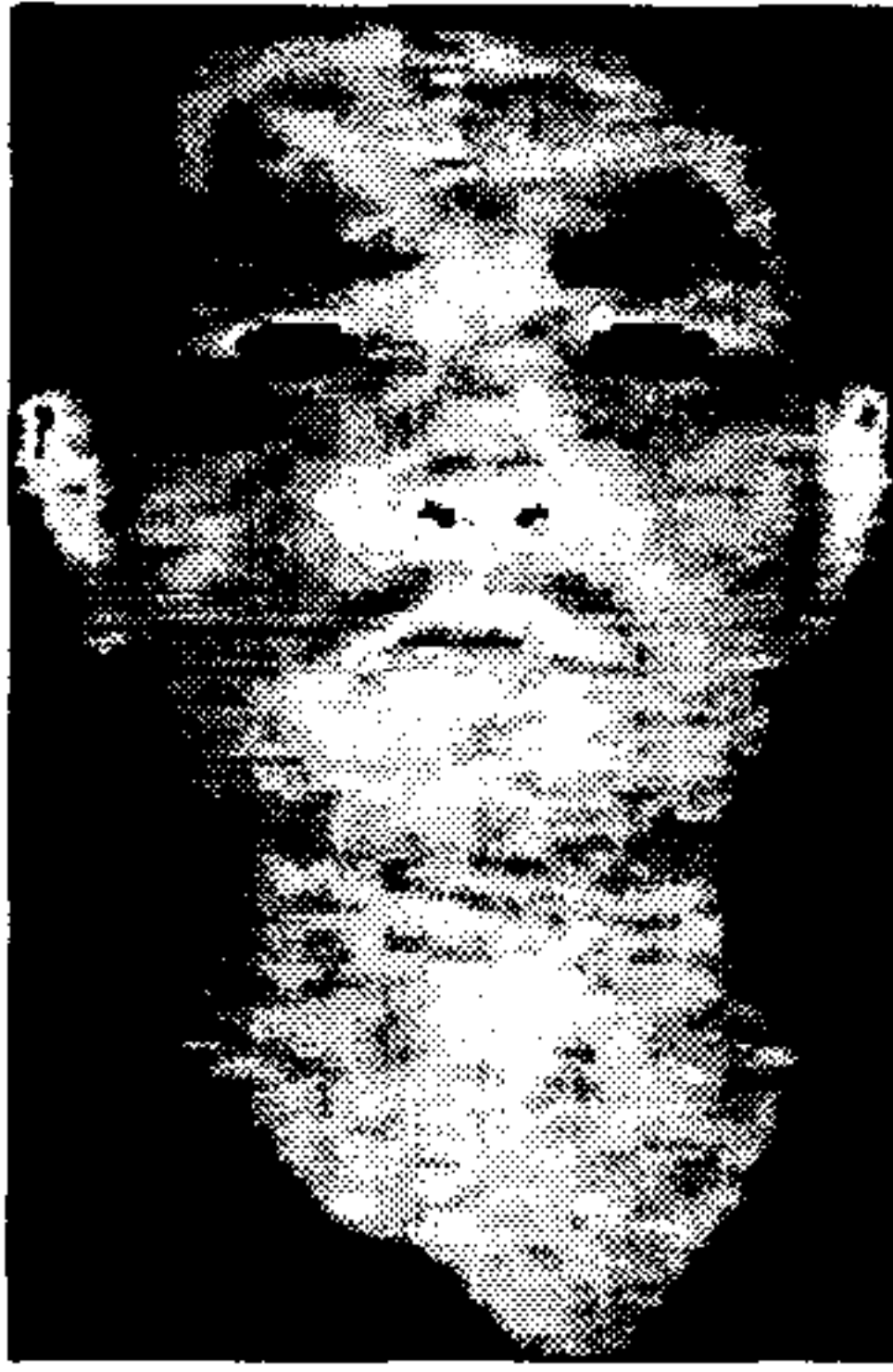


그림5.6 정규화된 입력영상

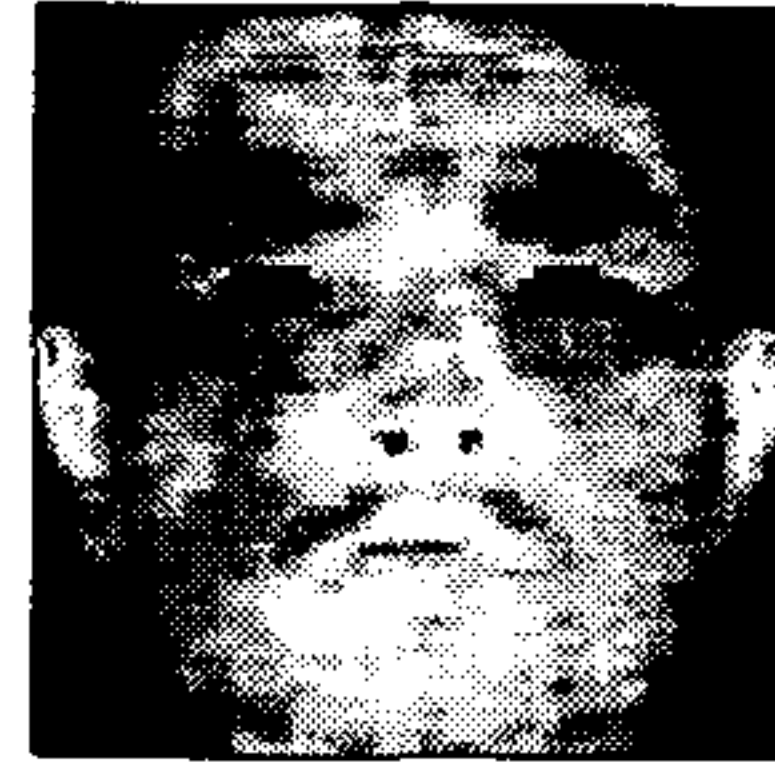


그림5.7 정규화된 60 x 60 영상

다. 얼굴 인식 알고리즘(Principal Component Analysis)

우리가 인식하고자 하는 사람의 수를 M 이라 하고 데이터베이스내의 영상을 $n \times n$ 이라 하면 이 영상을 1차원 벡터 ($1 \times n^2$)인 T 로 나타낼 수 있다. 이것은 데이터베이스를 만드는 Training set 이라 한다. 즉, 식 5.4와 같이 표현을 한다.

$$\text{Training set} : \Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M \quad (5.4)$$

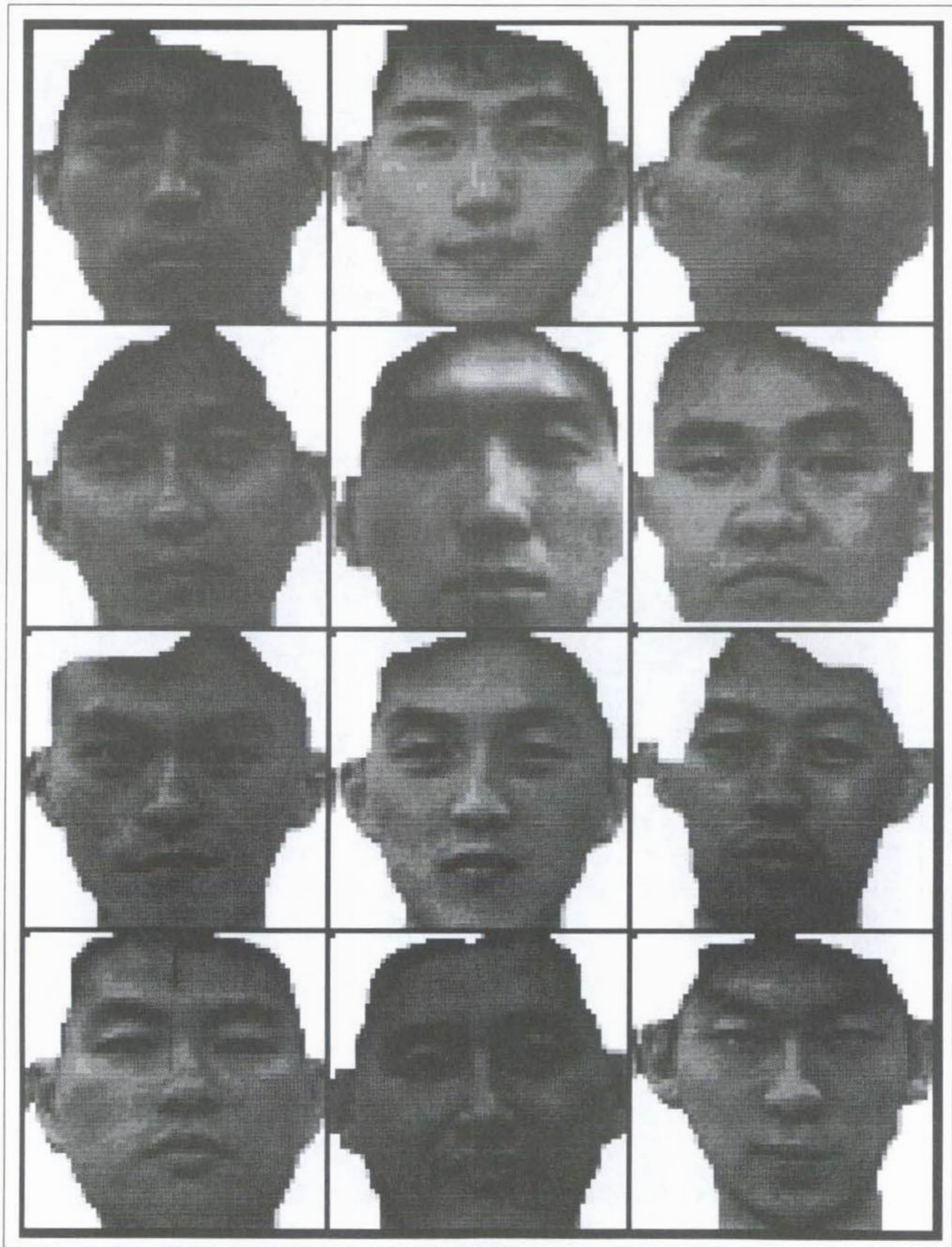


그림 5.8 Training 영상 Set (칼라 영상)



그림 5.9 색상 변화된 Training Set Image

그리고 이 Training set으로 식 5.5와 같이 평균 영상을 구한다. 그 평균 영상을 Ψ 라하고 그림 5.2.10과 그림 5.2.11과 같은 영상이 된다.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (5.5)$$

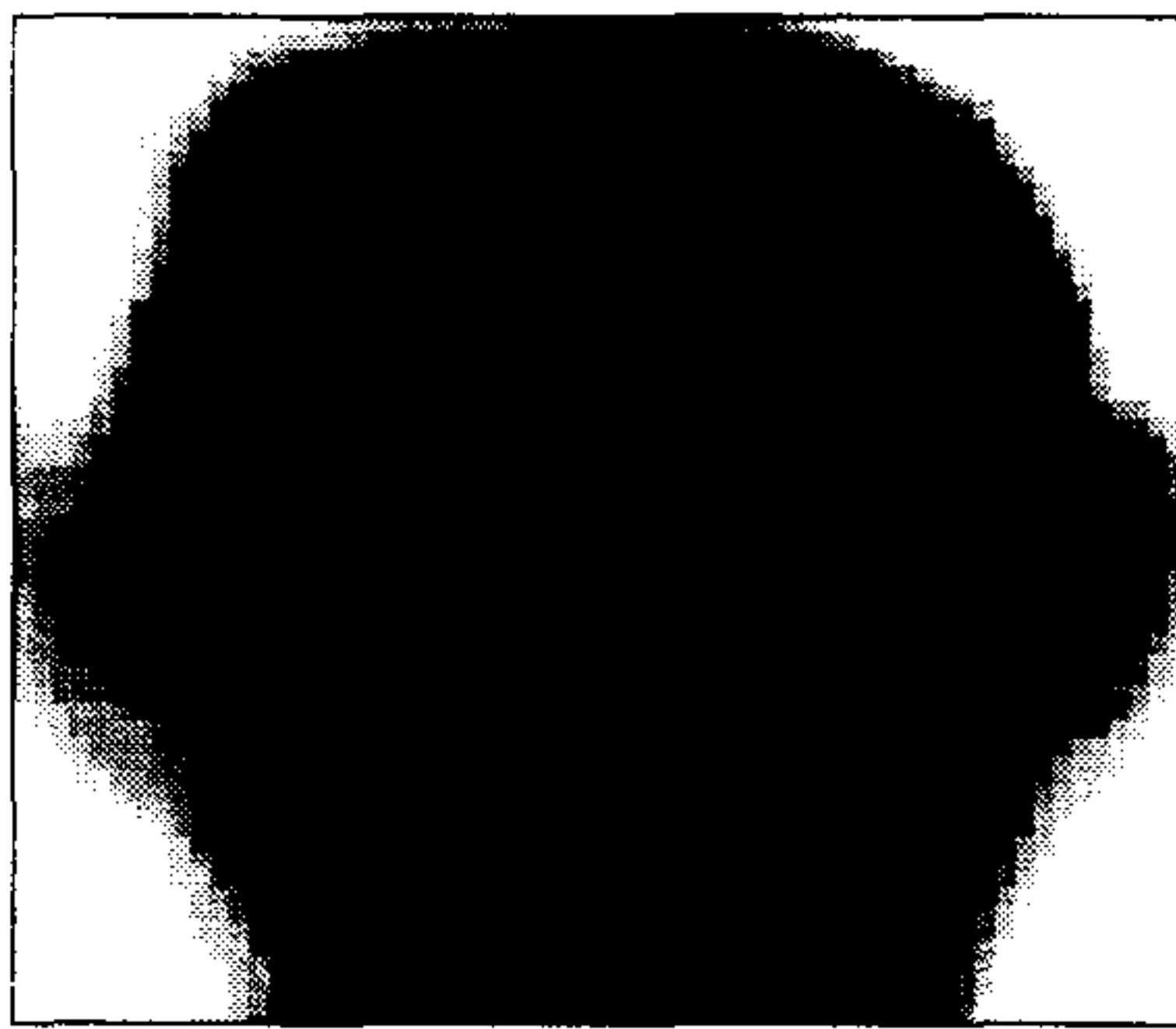


그림 5.10 평균 얼굴 영상 (칼라)

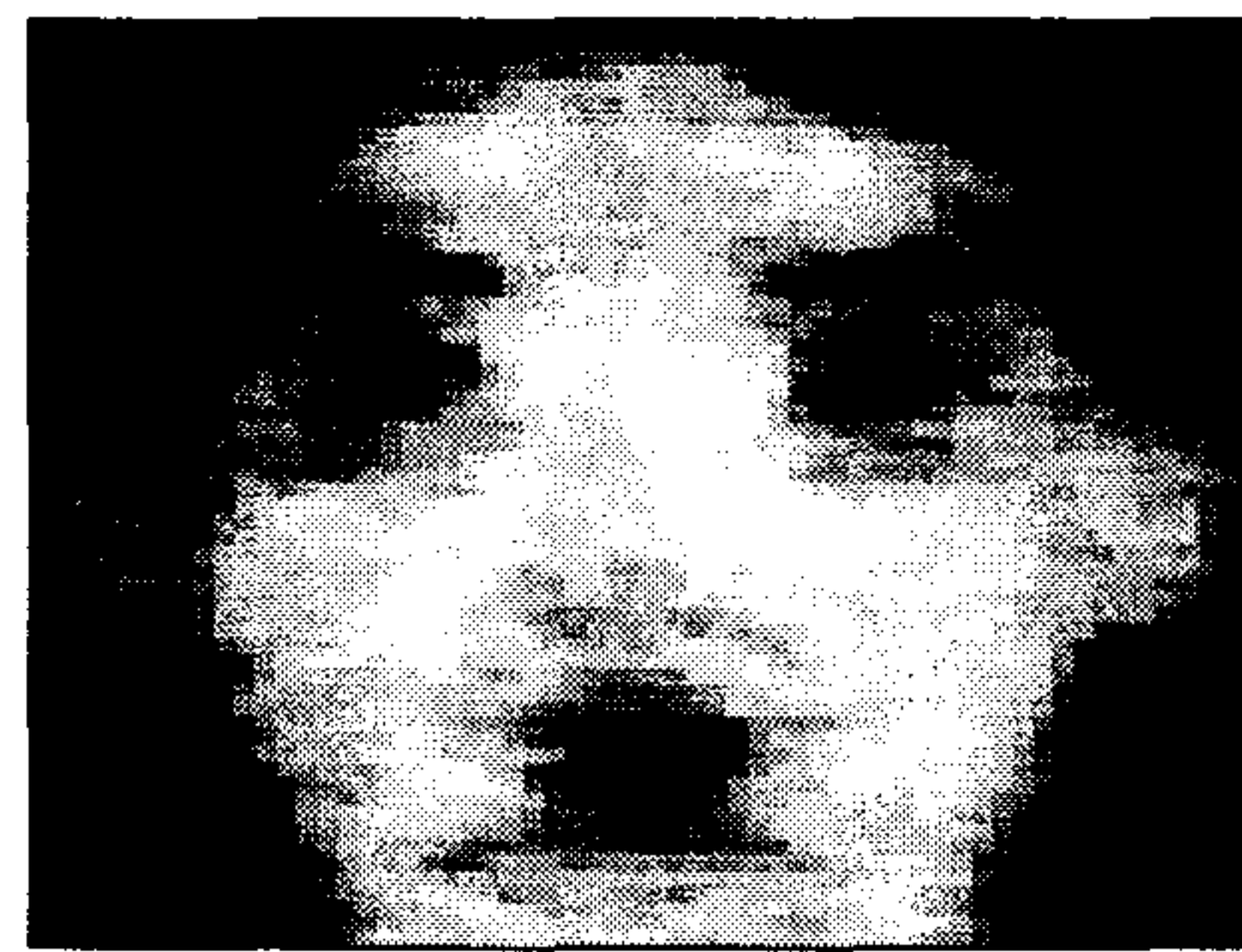


그림 5.11. 평균 얼굴영상 (색상변환후)

각각의 Training 영상에서 이 평균 영상을 뺀 차 벡터식 5.6을 구한 후,

$$\Phi_i = \Gamma_i - \Psi \quad (5.6)$$

이러한 차 벡터(Φ_i)들로 이루어진 공간을 표현할 수 있는 벡터 집합의 Eigenvector와 r와 Eigenvalue를 구하는데, 연산량을 줄이기 위하여 다음과 같은 방법을 사용한다. 얼굴 영상 공간을 표현 할 수 있는 공간의 축을 이루게 될, 구하려고 하는

Eigenvector를 \mathbf{u}_n ($n = 1, 2, 3, \dots, M$)이라 하고, \mathbf{u}_n 을 구하기 위하여 먼저 식 5.7과 같이 Covariance 행렬(\mathbf{C})을 구한다.

$$\begin{aligned} \mathbf{C} &= \frac{1}{M} \sum_{n=1}^M \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \\ &= \mathbf{A} \mathbf{A}^T \end{aligned} \quad (5.7)$$

여기서 $\mathbf{A} = [\boldsymbol{\phi}_1 \boldsymbol{\phi}_2 \dots \boldsymbol{\phi}_M]$ 이므로 \mathbf{A} 는 $n^2 * M$ 인 행렬이 되고 \mathbf{C} 는 $n^2 * n^2$ 인 행렬이 된다. 따라서 \mathbf{u}_n 를 구하려면 n^2 번의 연산이 필요하게 된다. 그런데 여기서 $\mathbf{A}^T \mathbf{A}$ 는 $M * M$ 행렬이 되므로 $\mathbf{A}^T \mathbf{A}$ 의 Eigenvector로 $\mathbf{A} \mathbf{A}^T$ 즉 \mathbf{C} 의 Eigenvector를 구함으로써 연산량을 감소할 수 있다. $\mathbf{A} \mathbf{A}^T$ 의 Eigenvector를 $\mathbf{A}^T \mathbf{A}$ 의 Eigenvector로 구하는 방법은 다음과 같다.

$\mathbf{A}^T \mathbf{A}$ 의 Eigenvector를 구하는 식은 식 5.8 같은데 이 식 양변에 \mathbf{A} 를 곱하면 식 5.9와 같이 되고 이 식은 구하고자 하는 $\mathbf{A} \mathbf{A}^T$ 의 Eigenvector를 구하는 식이 되어 된다.

$$\mathbf{A}^T \mathbf{A} \mathbf{v}_i = \mu_i \mathbf{v}_i \quad (5.8)$$

$$\mathbf{A} \mathbf{A}^T \mathbf{A} \mathbf{v}_i = \mu_i \mathbf{A} \mathbf{v}_i \quad (5.9)$$

즉, $\mathbf{A} \mathbf{A}^T$ Eigenvector \mathbf{u}_n 는 $\mathbf{A} \mathbf{v}_i$ 가 되고 연산량도 감소하게 된다. 여기서 구해진 M 개의 Eigenvector는 Covariance의 Eigenvector의 성질에 의해 식 5.10과

식 5.11 과 같은 두 개의 orthonomal 한 성질을 가지게 된다.

$$\mathbf{u}_l^T \mathbf{u}_k = 1, \text{ if } l=k \quad (5.10)$$

$$\mathbf{u}_l^T \mathbf{u}_k = 0, \text{ otherwise} \quad (5.11)$$

따라서 이 성질로 Eigenvector들이 구하려고 했던 얼굴 영상 데이터베이스의 공간을 표현할 수 있는 Basis 축을 이룰 수 있게 된다. 그리고 M 개의 모든 축을 모두 사용할 수도 있으나 Eigenvector들의 Eigenvalue가 작은 값을 가지는 Eigenvector는 인식에 많은 영향을 미치지 못하므로 전체 영상을 표현할 수 있는 M' 개를 취하여 \mathbf{u}_k ($k = 1, \dots, M'$)을 구한다. 이러한 과정은 인식률을 떨어뜨리지 않으면서도 데이터베이스의 데이터 량을 감축하는 효과를 볼 수 있다. 이렇게 구한 M' 개의 축으로 이루어진 공간으로 다음의 식 5.12와 같이 데이터베이스내의 모든 얼굴 영상들을 transform 한다.

$$\omega_k = \mathbf{u}_k^T (\Gamma - \Psi) \quad (5.12)$$

$$k = 1, \dots, M$$

따라서 데이터베이스의 모든 영상들은 변환된 공간의 축으로의 가중치인 weight 벡터 $\boldsymbol{\omega}^T = [\omega_1, \omega_2, \dots, \omega_M]$ 가 되고, 이 벡터가 인식을 위한 얼굴 인식의 특징점으로 사용된다.

라. 얼굴 인식 과정

얼굴 입력 영상의 인식 과정은 위의 인식하고자 하는 개개의 얼굴 영상 데이터베이스를 만드는 과정과 동일하다. 그리고 입력 영상에 대한 weight 벡터 (\mathbf{Q})가 구해지면 이 벡터와 데이터베이스의 weight 벡터와의 다음 식 5.13과 같이

$$\varepsilon_k = \|(\mathbf{Q} - \mathbf{Q}_k)\|^2 \quad (5.13)$$

euclidean distance를 구하여 나온 오차 값이 가장 작은 데이터베이스내의 얼굴로, 입력 얼굴이 누구인지를 인식하게 된다.

마. 실험

실험은 본 연구의 최종 목표인 실시간 동영상에서의 얼굴 인식을 위한 시스템 구현에 앞서 어느 정도 배경이 있는 사람 얼굴의 정면상이 있는 정지 영상으로 실험을 하였다. 그림 5.12와 같이 입력 영상은 Sony DCR-VX1000 디지털 캠코더를 이용하여 320*240 의 크기로 입력을 받아서 실험 환경으로 Pentium-133을 이용하여 실험을 하였다. 실험에 사용된 인식을 위한 얼굴 영상 데이터베이스는 위에서 그림 5.9와 같이 60*60의 크기의 영상으로 구축하였으면 실험에 사용된 구분하고자 하는 인물의 수를 14명으로 정하여 얼굴 영상 데이터베이스의 수는 14개로 구축하였다.

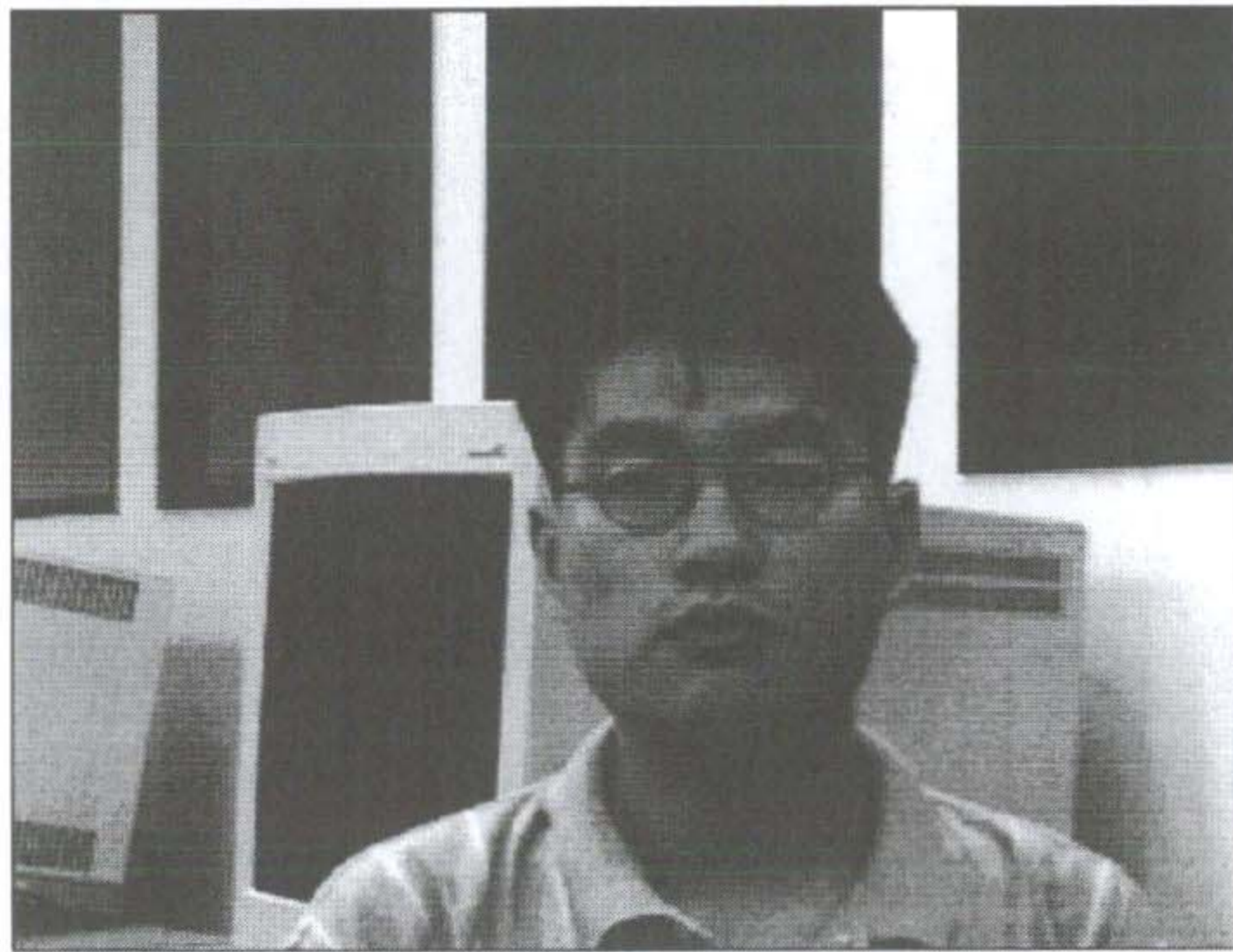


그림 5.12 실험 입력 영상 (320* 240)

그림 5.13은 데이터베이스 얼굴 영상 공간의 기본 축이 되는 Eigenvector 영상 중에서 Eigenvalue 값이 큰 것부터 작은 순으로 5개를 나타낸 것이다.



그림 5.13 Eigenvector 영상

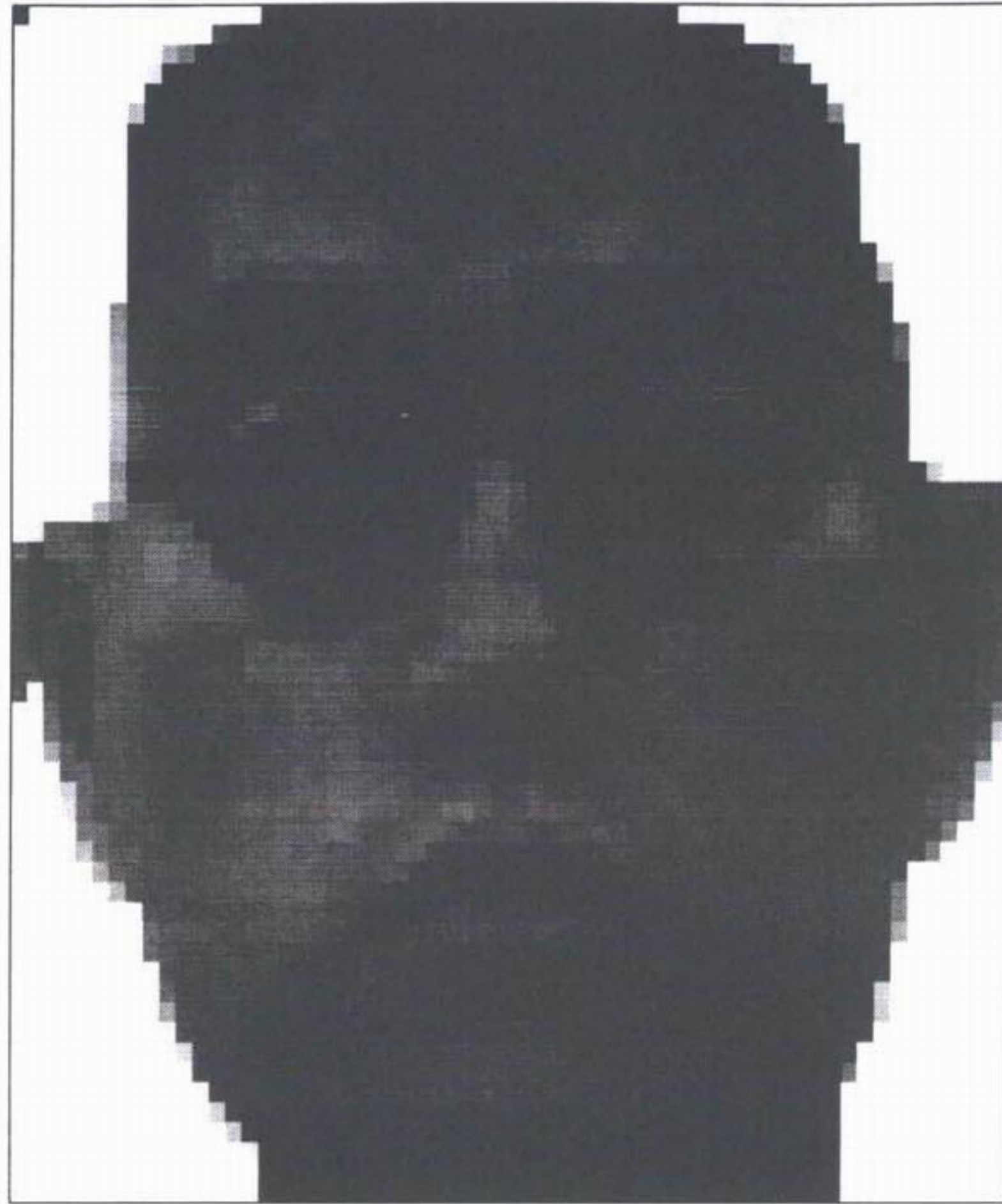


그림 5.14 인식 test 영상

인식에서의 얼굴 입력 영상은 동일인이라 하더라도 데이터베이스에 있는 영상과는 조금 변형된 영상을 입력하여 인식하는지를 실험하였다. 그림 5.14 과 같이 변형된 영상은 주로 안경을 이용하였다.

실험은 데이터베이스내의 14명중 어느 정도 데이터베이스내의 영상과는 다른 변형된 부분을 가지고 있는 영상 7개에 대하여 실험을 하였다. 얼굴 인식 실험은 두 가지 방향으로 수행했는데 얼굴 인식률에 대한 실험과 얼굴 인식률과 Eigenvector의 수에 대한 실험을 하였다. Eigenvector의 개수는 얼굴 영상 데이터베이스가 광

범위 할 경우에는 위의 알고리즘 설명으로 유추하였을 때 데이터베이스의 양과 인식 속도에 크게 좌우할 수 있는 요소가 됨을 알 수 있다.

실험 결과는 Eigenvector를 14개 다 사용하였을 경우 7개중 하나의 인식 실패를 보였다. 이러한 인식률은 Eigenvector를 14개에서 7까지 사용하였을 경우 같은 결과를 나타내었다. 따라서 이 결과로부터 제안하는 알고리즘의 얼굴 인식 시스템을 구성 시에 미리 정한 인식률 한계를 유지하려면 데이터베이스에서 만들어지는 Eigenvector 모두를 사용할 필요 없이 인식률을 유지하는 Eigenvector 개수만큼 사용하면 데이터베이스의 양도 감소되고 인식 속도의 향상과 인식 과정에서 사용하는 시스템의 메모리도 절약된다는 것을 알 수 있다.

3절. RBF 신경망 알고리즘

가. 신경망(Neural Network)

인간의 신경계는 뉴런(neuron)이라고 불리는 신경 세포들로 구성되어 있다. 이 뉴런은 시냅스를 통해 입력되는 다른 뉴런의 신호를 합하여 전송로의 역할을 하는 축색으로 응답 신호를 보낸다. 이 때 각 뉴런들은 다른 것들로부터 입력된 신호의 합이 문턱값 보다 클 때에만 반응하여 그 출력을 다음 뉴런으로 전달한다.

인공 신경망은 이러한 인간의 신경계를 수학적으로 근사화한 모델로서 그림 5.15와 같이 나타낼수 있다. 이를 이용해 목적으로 하는 작업의 표본을 학습시키고 그 결과를 저장하여 이를 일반적인 데이터에 적용한다. 이 근사화 모델에 사용된 일반적인 가정은 다음과 같다.

1) 정보 혹은 신호의 처리는 뉴런이라고 불리는 간단하고 수가 많은 구성 요소들에서 발생한다.

- 2) 신호는 전송로(connection link)를 통해서 각 뉴런들 사이로 전달된다.
- 3) 전송로는 각각 연관된 가중치(weight)를 가지고 있으며 이는 전달되는 신호에 곱해진다.
- 4) 뉴런은 자신의 출력을 결정하기 위해 가중치가 곱해진 입력 신호들의 합에 활성 함수(activation function)를 적용한다.

또한 인공 신경망은 다음에 의해 특징 지어진다.

- 1) 뉴런들 사이의 연결 형태 : 구조적 측면
- 2) 연결에서의 가중치를 결정하는 방법 : 학습 알고리즘 측면
- 3) 출력을 결정하는 활성 함수의 종류

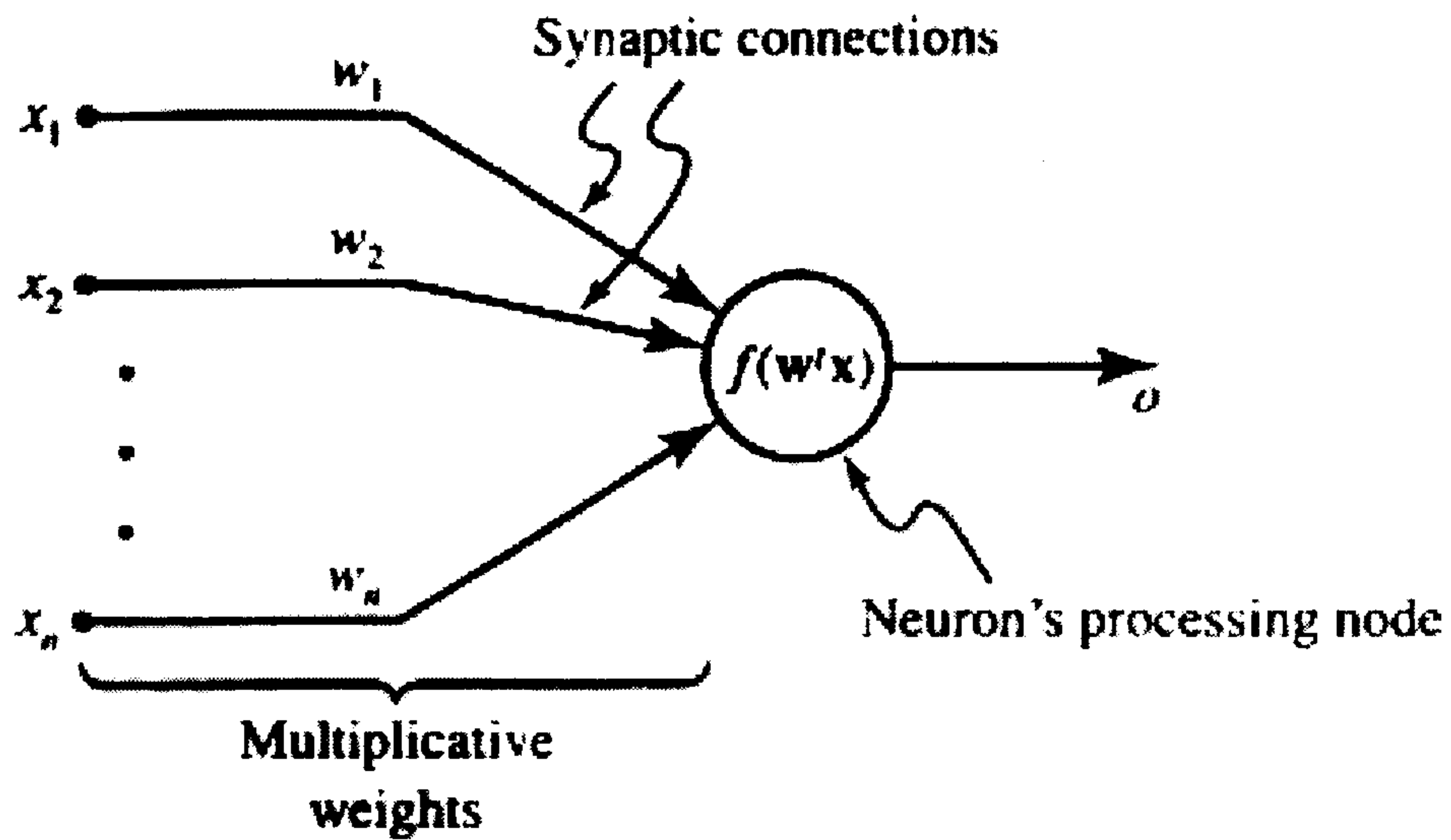


그림 5.15 일반적인 신경망 구조의 수학적 모델 구조

나. 기본적인 신경망 모델과 알고리즘

같은 방식으로 동작하는 뉴런들은 입력층, 은닉층(hidden layer), 출력층 등의 층(layer)을 이루어 동작한다. 일반적으로 각 층의 뉴런들은 같은 활성화 함수를 가지고 다른 층의 뉴런들과 동일한 형태로 연결되어 있으며 또한 인접한 두 층의 각 뉴런들은 모두 연결을 가진다.

a. 다중 층 퍼셉트론(Multilayer Perceptron) 구조

현재 가장 기본적으로 이용되는 신경망 모델은 그림 5.16에 나타낸 다중층 퍼셉트론 구조이며 학습 알고리즘으로는 BP(Backpropagation)을 사용한다. 퍼셉트론은 입력층, 출력층 그리고 두 층의 뉴런들을 서로 연결하는 가중치를 가지는 링크(connection link)로만 구성된 가장 간단한 신경망 구조인데, 이 때 입력층과 출력층 사이에 하나 이상의 은닉층을 가지는 것이 다중 층 퍼셉트론 구조이다. 하나의 은닉층과 비선형 활성화 함수를 가지는 다중 층 퍼셉트론 망은 가장 보편적인 분류기이다. 망의 각 뉴런은 가중치가 곱해진 입력들의 합을 취하여 비선형 활성화 함수를 통과 시켜 출력 값을 낸다. 일반적으로 사용되는 비선형 활성화 함수는 sigmoid 함수이며 그 식은 다음과 같다.

$$f(net) = \frac{1}{1 + \exp(-net)} \quad (5.14)$$

다중 층 퍼셉트론 구조는 단일 층 구조 혹은 퍼셉트론에 비하여 보다 복잡한 문제를 해결할 수 있으나 학습의 과정에서 어려움이 있다. 그러나 대부분의 경우 학습은 성공적으로 이루어지며 따라서 다중 층 퍼셉트론 구조에서 불가능했던 작업을 처리할 수 있다.

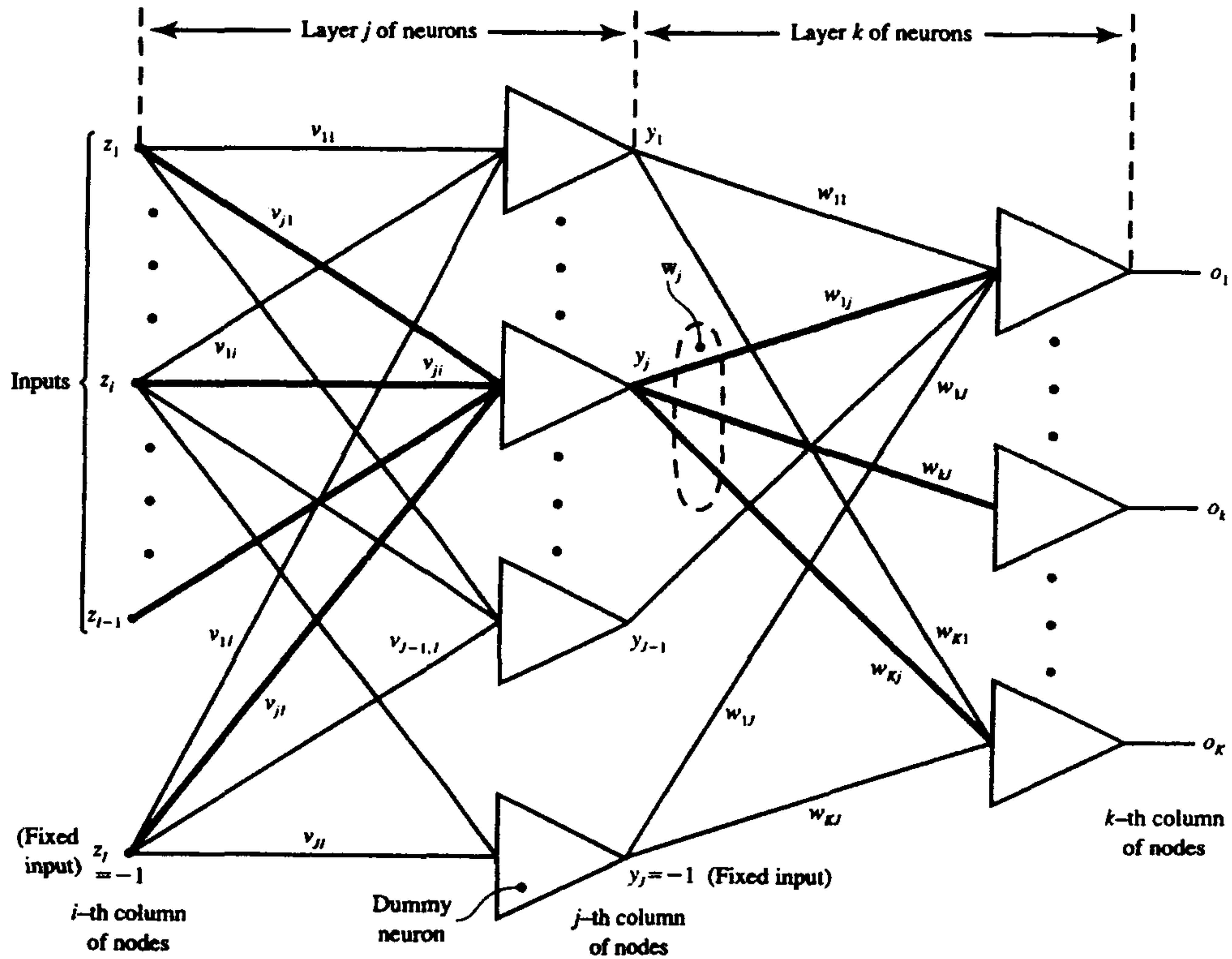


그림 5.16 다중 퍼셉트론 모델의 구조

b. BP(Backpropagation) Algorithm

BP는 가장 간단한 학습 알고리즘 중의 하나로서 뉴런들 사이의 가중치를 조절함으로써 망의 전체 오차를 최소화하는 것을 기본 개념으로 한다. 말하자면 이 알고리즘은 일련의 학습 자료에 대해 관련된 원하는 값(desired value)을 가지고 있는 supervised training이므로 학습과정에서 생긴 출력 값이 원하는 값으로 수렴할 때까지 그 오차를 최소화하는 방향으로 가중치를 조정해 나가게 된다. 오차의 계산은 일반적으로 Mean square error를 사용하지만 “연속적”이고 “미분가능”한, 실

제 망의 출력과 원하는 값과의 “거리”를 의미 있게 표현할 수 있는 어떠한 함수도 사용 가능하다. Mean square error는 다음과 같이 정의한다.

$$E_p = \frac{1}{2} \sum_{k=1}^K (t_{pk} - o_{pk})^2 \quad (5.15)$$

여기서, E_p 는 특정한 패턴 p 에 대한 오차이고 o_{pk} 는 출력 단에서 k 번째 뉴런의 실제 출력 값, t_{pk} 는 해당하는 원하는 값이다. 가중치에 대한 오차 함수의 음의 기울기(gradient)는 이 방향으로 진행함에 따라 오차가 급격히 줄어들어 결국 기울기가 0인 최소 지점에 도달할 것임을 말한다. 이 개념을 수학적으로 표현하면 다음과 같다.

$$\Delta w_{kj} = -\eta \frac{\partial E_p}{\partial w_{kj}} \quad (5.16)$$

Δw_{kj} 항은 개개의 가중치의 조정치(source 뉴런 j 와 destination 뉴런 k 를 연결하는 가중치의 변화량)를 의미하며, η 는 양수인 상수로 학습률(learning rate)이라고 한다. 신경망이 3개의 층을 가진 구조로 되어있다고 가정하면 식 5.16은 다음과 같이 쓸 수 있다.

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}} \quad (5.17)$$

$$\Delta v_{ji} = -\eta \frac{\partial E}{\partial v_{ji}} \quad (5.18)$$

출력 층에서, 오차 E 는 출력 값 o_k 의 함수이며 또한 o_k 는 입력값 net_k 의 함수이다. chain rule을 이용해 E 의 편미분은 다음과 같이 전개할 수 있다.

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial net_k} \frac{\partial net_k}{\partial w_{kj}} \quad (5.19)$$

여기서 net_k 는 앞 층으로부터의 출력에 가중치를 곱해 모두 합한 것이다.

$$net_k = \sum_j w_{kj} o_j \quad (5.20)$$

식 5.17과 식 5.19로부터 다음식을 얻을 수 있다.

$$\Delta w_{kj} = \eta \delta o_j \quad (5.21)$$

$$\delta_k = - \frac{\partial E}{\partial net_k} \quad (5.22)$$

$\partial E / \partial net_k$ 항을 계산하기 위해 다시 한 번 chain rule을 사용한다.

$$\begin{aligned} \delta_k &= - \frac{\partial E_p}{\partial net_k} = - \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial net_k} \\ &= -(t_k - o_k) \frac{\partial o_k}{\partial net_k} \\ &= -(t_k - o_k) f'(net_k) \end{aligned} \quad (5.23)$$

여기서 $f(\cdot)$ 은 출력층의 뉴런에 있는 활성화 함수로서 다음과 같은 관계를 가진다.

$$f(net_k) = o_k \quad (5.24)$$

BP 알고리즘에서 활성화 함수는 연속적이고, 미분가능하며, 단조 증가하는 특성을 가지고 있어야 한다. 식 5.24를 식 5.23에 대입하면 다음을 얻을 수 있다.

$$\begin{aligned} \Delta w_{kj} &= \eta (t_k - o_k) f'(net_k) o_k \\ &= \eta \delta_k o_k \end{aligned} \quad (5.25)$$

은닉층에 대해서도 같은 방법을 적용하면,

$$\Delta v_{ji} = \eta \delta_j o_j \quad (5.26)$$

여기서,

$$\delta_j = f'(net_j) \sum_k \delta_k w_{kj} \quad (5.27)$$

요약하면, BP를 이용한 신경망의 학습은 다음의 세 단계를 포함한다. 첫 번째 단계는 입력되는 학습 패턴들이 전방향으로 진행되는 것이다. 각 입력 뉴런들은 입력값을 받아들이고 은닉층의 뉴런과 출력층의 뉴런들이 차례로 활성화 함수를 이용해 출력값을 계산한다. 두 번째 단계는 관련된 오차를 역방향으로 전달한다. 이 단계에서 출력 뉴런은 원하는 값과 첫 번째 단계에서 얻은 실제의 출력값과의 오차를 구함으로써 관련된 오차를 결정하고 δ_k 를 계산한다. 그런 다음 같은 방법으로 은닉층의 각 뉴런에서 δ_j 를 계산한다. 세 번째 단계에서 다음의 식에 의해서 가중치를 조정한다.

$$\begin{aligned} w_{kj}(t+1) &= w_{kj}(t) + \Delta w_{kj}(t) \\ v_{ji}(t+1) &= v_{ji}(t) + \Delta v_{ji}(t) \end{aligned} \tag{5.28}$$

위의 과정은 망의 전체 오차가 최소화 될 때까지 반복된다.

다. Radial Basis Function Network

그림 5.17에 나타낸 RBF 망은 폭넓은 범위의 연구에 의해 가치 있는 모델로 인정되어 오고 있다. 그 가장 큰 특징은 우선 계산이 간단하고 단지 은닉 층과 출력 층 사이에만 supervised training이 필요하므로 수렴 속도가 빠르다. 두 번째는 잘 개발된 수학적 이론에 의한 기술로 안정성을 제공한다. 이것은 영상에서 공통적인 빈약하고, 높은 차원의 데이터를 다루는데 유리하므로 실제적인 비전 응용에 이상적으로 보인다. RBF 망은 충분히 높은 차원 차원의 은닉층의 뉴런을 가지는데 이는 입력되는 데이터의 비선형적 변환을 제공하며 출력층은 은닉층으로부터의 데이터를 다시 선형적으로 변환한다.

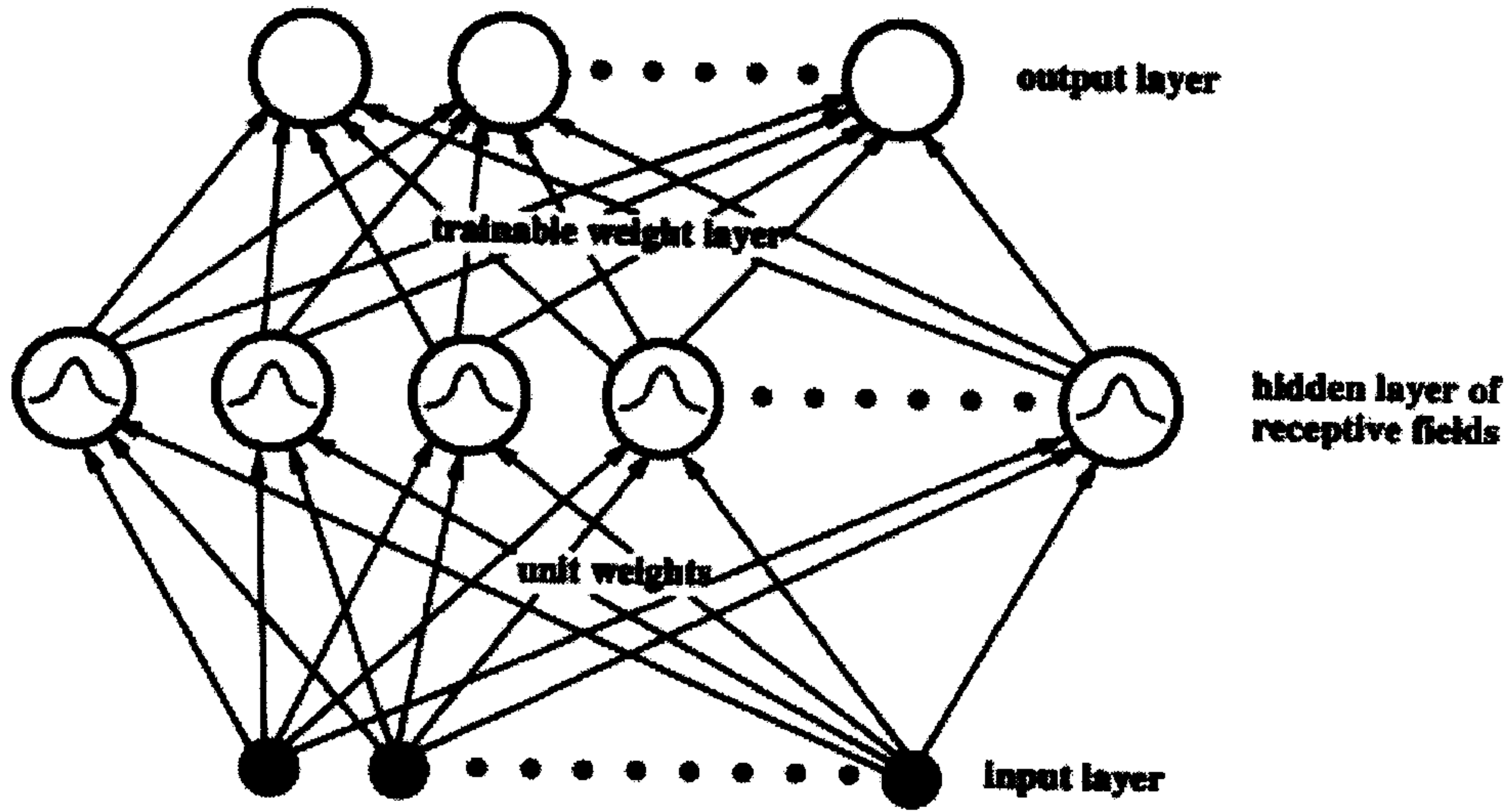


그림 5.17 RBF 망 구조 모델

a. RBF 망 모델

RBF 망은 다른 형태의 두 층이 혼합된 구조로 이루어져 있다[]. 출력층에는 MLP와 유사한 활성 함수가 은닉층으로부터의 출력을 선형적으로 변환하며, 두 층 사이의 연결에서 가중치를 조정하는 것도 유사한 학습 방법을 취한다. 그러나 은닉층의 뉴런(receptive field)들은 개개의 고정된 중심값을 가지며 RBF라고 부르는 활성 함수가 입력값이 얼마나 중심값에 가까운지를 계산한다.

$$out_j = g(\| \mathbf{x} - \mathbf{c}_j \|) = g\left(\sum_{i=1}^N (x_i - c_{ij})^2\right) \quad (5.29)$$

여기서 out_j 는 은닉층의 j 번째 뉴런의 출력을 말하며, c_{ij} 는 그 뉴런의 중심값이다. 활성 함수 $g(\cdot)$ 는 보통 Gaussian 함수를 취한다.

$$g(r^2) = Ae^{-r^2/\sigma^2} \quad (5.30)$$

은닉층의 각 뉴런은 연관된 표준 편차 σ 를 가지며 이는 중심값과 더불어 각 뉴런의 성질 및 허용 범위를 결정 지운다. 상대적으로 큰 값의 σ 을 가지는 뉴런은 많은 데이터에 대해서 반응할 것이며, σ 가 작으면 중심값과 상당히 유사한 데이터에 대해서만 반응하게 될 것이다. 출력층에서 k 번째 뉴런의 출력값 다음과 같다.

$$o_k = \sum_j w_{jk} o_j \quad (5.31)$$

비선형적인 입력과 출력의 맵핑(mapping)에 대한 전체적인 근사화를 취하는 MLP 망과는 달리 RBF 망은 Gaussian과 같은 함수를 이용해 지역적인 맵핑에 의존한다. 따라서 각 뉴런의 중심값과 표준 편차의 결정이 실제적인 망의 성능에 중요한 역할을 하게 된다.

b. RBF 망 학습 알고리즘

RBF 망에서는 은닉층의 각 뉴런의 중심값과 표준 편차 그리고 은닉층과 출력층 사이의 가중치 등 세 가지 형태의 학습이 존재한다. 중심값은 학습 데이터 중에서 임의로 뉴런의 개수만큼 선택해 사용하기도 하였으나, 일반적으로 클러스터링 알고리즘으로 구한 각 클러스터의 중심값(평균값)을 취한다. 대표적인 클러스터링 알고리즘으로는 K-nearest-neighbor 알고리즘이 있으며, Singular value decomposition(SDV), Orthogonal least square (OLS) 알고리즘 등이 사용되어 왔다. Moody와 Darken은 입력 학습 데이터에 관계없는 표준 편차의 학습 방법을 적용했다. 또한 한 뉴런의 중심값과 다른 뉴런의 중심값과의 차이 (Euclidean distance)를 이용해서 표준편차를 계산할 수도 있다.

$$\sigma_j = \frac{1}{\sqrt{2}} \{ (c_j - c_l)^2 \}^{1/2} \quad , j \neq l \quad (5.32)$$

은닉층과 출력층 사이의 가중치를 학습하는 방법은 앞장의 BP에서 언급했던 식()를 최소화하는 LMS(least mean square)법칙을 사용한다. BP와는 달리 가중치는 은닉층과 출력층 사이에만 존재하므로 수렴속도가 빨라 학습이 더 용이하다.

라. 실험 결과

본 실험은 RBF 망이 대표적인 인식 분야인 얼굴에 대해서 어느 정도의 성능을 보여주는지를 절대적 수치로 살펴 보는 것을 목적으로 하였다. 다른 방법들과의 상대적인 비교를 위해서는 사용하는 얼굴 데이터 베이스가 동일 해야하는 문제가 있고, 실제로 얼굴 영상에 대해 반드시 필요로 하는 부분이 다를 수 있으므로, 다시 말하면 각 방법에서 사용하는 데이터에 대한 표준형이 다르므로, 이에 대한 고려가 제시되지 않는 비교는 일단은 어렵다고 보여진다.



그림 5.18 학습용 데이터



그림 5.19 테스트용 데이터

본 실험에서 7명의 사람들에 대해 각각 다른 3장씩의 흑백 영상 즉, 총 21장의 영상을 기본 데이터로 사용한다. 이들 영상은 40*30의 크기이며 원 영상으로부터 양 눈끝의 중간과, 눈썹 상단과 코끝의 중간을 중심으로 해서 얻어진 것이다. 표 5.1부터 결과로부터 RBF 망에서도 다른 신경망과 마찬가지로 많은 데이터를 학습할수록 좋은 인식률을 보여준다는 것을 알 수 있다. 평균 오차는 테스트 시 실제의 출력과 기대치(해당 클래스이면 0.9 이상, 아니면 0.1 이하)와의 차이를 평균한 것이다. 표 5.1.2에서는 은닉층의 뉴런에 있는 평균값을 구하는 방법에 따른 인식률의 차이를 보여준다. 여기에서는 각 사람에 대해서 두 장씩의 영상을 학습 데이터로, 나머지 한 장씩을 테스트용으로 사용하였다. A는 K-Means 알고리즘을 이용해서 7개의 평균값을 구하며, B에서는 각 사람에 대한 두 장의 영상의 평균을 구해 이것을 사용한다. C는 앞의 두 가지를 혼합한 것으로서 A 방법에 K-Means 알고리즘으로 구한 3개의 평균값을 추가 한 것이다. 구조적 측면에서 RBF의 인식 능력은 평균값에 의해 크게 좌우되므로, 입력 데이터로부터 각 클래스의 특징을 가장 잘 표현할 수 있는 을 사용하도록 더 나은 방법이 요구된다.

학습/테스트 영상	인식률 %	평균 오차	중심값 계산 방법
7/14	92.9(13/14)	0.35	C
14/7	100(7/7)	0.24	C

표 5.1 학습 데이터의 양에 대한 실험 결과

중심값의 계산 방법	인식률 %	평균 오차
A	85.7(6/7)	0.36
B	85.7(6/7)	0.28
C	100(7/7)	0.24

표 5.2 중심값 계산 방법에 대한 실험 결과

다른 방법들과 마찬가지로 인식의 문제에 사용되는 RBF 망에서 가장 우선적인 문제는 어떤 데이터를 입력으로 사용하느냐 하는 것이다. 본 실험에서는 사람의 얼굴 영상을 그대로 사용하였으나, 이것은 얼굴의 특징점들 즉, 눈썹, 눈, 코, 입 등의 위치와 그 휘도값에 너무 의존적이므로, 회전이나 표정의 변화에 아주 민감한 단점을 가지고 있다. 사람의 특징을 가장 잘 나타내며 변화가 적은 요소들을 추출해 사용하는 것이 필요하지만 RBF 망의 가장 큰 장점이 입력 데이터의 차원이 클수록 더 좋은 결과를 내는 것이라는 사실 또한 고려해야 할 것이다. 그리고 기존의 다중 층 퍼셉트론 구조에서 가중치를 학습시키는 방법은 최소한 입력층과 은닉층, 은닉층과 출력층 사이의 두 개 이상이 존재할 때 유용한 방법이므로, RBF 망에서는 다른 적절한 방법을 사용하거나 혹은 기본적인 구조인 평균값이 존재하는 은닉층과 출력층 사이에 하나 이상의 층을 추가함으로써 가중치로 인한 인식 능력의 비중을 좀 더 높이는 것도 좋은 방법이 될 수 있다.

5장 참고문헌

- [1] R.Chellappa, C.L.Wilson and S.Sirohey, "Human and Machine Recognition of Face : A survey" in Proc. IEEE Computer Vision and Patt. Recog., vol. 83, no 5, 1995, pp 704.
- [2] A.Pentland, B.Moghaddam, T.Starner, and M.turk, "View based and modular eigenspaces for face recognition," in Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt. Recog., 1994, pp 84-91.
- [3] M.kirby and L.sirovich, "Application of Karhunen-Loeve procedure for the characterization of human faces," IEEE Trans. Patt. Anal. and Mach. Intell., vol 12, pp. 103-108, 1990.
- [4] A.Yuille, D.Cohen, and P.Hallinan, "Feature extraction from faces using deformable templete," in Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt. Recog., 1989, pp.104-109.
- [5] I.Craw, H.Ellis, and J.Lishman, "Automatic extraction of face feature," Patt. Recog. Lett. ,vol. 5, pp. 183-187, 1987.
- [6] B.S.Manjunath, R.Chellappa, and C. v. d. Malsburg, "A feature based approach to face recognition," in Proc. IEEE Computer Soc. Conf. on Computer Vision and Patt. Recog., 1992, pp. 373-378.
- [7] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representation by Error Propagation," in Parallel Distributed Processing, Vol. 1:Foundations, MIT Press, Cambridge, MA, pp. 318-362, 1986.
- [8] D.J. Burr, "Experiments on neural net recognition of spoken and written

text," *IEEE Trans, Acoust., Speech, Signal Processing*, vol. ASSP-36, pp. 1162-1168, July 1988

[9] J. Moody and C.J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation* 1, pp. 281-294, 1989.

[10] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science* 247, pp. 978-982, 1990.

[11] F. Girosi, "Some extensions of radial basis functions and their applications in artificial intelligence," *Computers Math. Applic.* 24, pp. 61-80, 1992.

[12] J. Moody and C.J. Darken, "Learning with localized receptive fields," in *Proc. 1988 Connectionist Models Summer School*. San Mateo, CA: Morgan Kaufmann, 1988.

[13] K. Stokbro, D.K. Umberger, and J.A. Hertz, "Exploiting neurons with localized receptive fields to learn chaos," *Science* 247, pp. 978-982, 1990.

제 6 장 결론

디지털 동영상 기술은 멀티미디어 기술의 도래와 함께 그 중요성이 크게 부각된 분야이다. 특히 문자, 소리, 컴퓨터 그래픽스 등의 기술이 혼합화 하여 새로운 미디어 기술을 이루면서 다른 미디어와 비교하여 상대적으로 큰 연산량과 컴퓨터 하드웨어를 요구하는 동영상 기술은 저장 및 전송의 효율성을 위하여 영상 압축 및 복원기술이 핵심으로 연구되어 왔다. JPEG 및 MPEG 연구는 이러한 연구의 중요성이 잘 인식된 결과이다. 표준화 기술의 정착으로 압축 및 복원기술이 어느정도 정착된 상황에서 이들의 효율적인 구현을 목적으로하는 대규모 집적회로 시스템 기술이 활발한 연구를 이루었으며 MPEG-2 부호화/복호화 처리기, 움직임 검출기 등 영상통신을 가능케 하는 기반기술들이 표준화 작업의 결과로 활발한 연구가 최근 수년간 진행 되었다. 최근들어 부상되는 차세대 디지털 동영상 기술에는 스테레오 영상의 효율적인 표시와 전송 및 저장을 위한 스테레오 영상기술, 디지털 영상을 이용하여 다양한 영상효과 및 편집 시스템을 가능케 하는 디지털 영상 저작 및 생성기술, 가상현실과 컴퓨터 그래픽스 등을 실제 영상과 혼합하여 새로운 가상의 공간을 생성하고 이러한 가상의 공간과 상호 반응하는 대화형 가상공간 기술 등이 새로운 분야로 대두되고 있다. 특히 미국의 디지털 텔레비전 방송 개시와 함께 방송 기술의 디지털화는 영상 시스템의 응용분야가 영화산업, 광고산업, 텔레비전 방송산업, 케이블 텔레비전 등 모든 분야가 영상의 형태 및 제작의 상호관계 등 새로운 대대적인 표준화 작업이 밀물처럼 홍수를 이룰 것으로 예상된다.

차세대 디지털 동영상관련 핵심 알고리즘 연구와 이들을 효율적으로 실시간 환경에서 구현하여 실제 사용 가능한 응용예를 보이는 것을 목적으로 하였다.

본 연구를 마감하는 3차 연도의 연구는 1차 및 2차 연도에서 연구 되어진 알고

리듬과 기법들을 효율적으로 실시간 환경에서 구현하여 실제 사용 가능한 응용예를 보이는 것을 목적으로 하였다. 또한 향후 그 중요성이 크게 부각될 것으로 사료되는 지능형 인간-기계 접속 기술의 핵심인 대화형 가상공간에서 효율적으로 사용 가능한 기반기술을 구축 하고자 하여 3차원 공간의 물체 입력기술과 사람을 중심으로하는 상호 대화형 기술 개발을 목적으로 1) 스테레오 영상 입력 시스템, 2) 고성능 스테레오 정합 알고리즘, 3) 실시간 얼굴 검출 및 추적 시스템, 4) 얼굴 및 제스처 인식 시스템 등과 같은 핵심기술을 다루었다.

고성능 스테레오 영상획득기술, 스테레오 정합 알고리즘 및 이의 실시간 구현기술, 얼굴영역 추출기술 및 추적기술, 얼굴 식별 및 제스처 인식기술 등은 본연구에서 기본적인 연구를 수행하였을 따름이지 결코 이와 관련된 연구가 마무리 되었다고 말할수 없다. 하지만 본 연구를 통하여 개발되고 축적된 기술들은 향후 이 분야의 연구에 기반이 되어서 많은 응용분야가 개발되길 바란다. 특히 본 연구를 통해 개발된 실시간 처리 능력을 갖는 스테레오 정합 처리기 기술과 병렬 DSP를 이용한 얼굴 영역 추적기술등은 산업화가 즉시 가능한 분야로 기업화를 위한 노력이 필요하며 이를위한 지원을 기대한다. 또한 개발된 관련 핵심 기술들이 향후 연구되어지는 대화형 가상환경기술 등에서 효율적으로 사용될수 있도록 기대해 보며 이를위해 지속적인 연구가 진행 되어야 할 것이다. 또한 최근 디지털 영상 미디어에 대해 고조되는 관심에 실제적으로 부응할 수 있는 연구 결과의 활용에 기대를 걸어본다. 특히 얼굴영상 추적기술은 방송사의 얼굴 마스크 기술 등에 응용이 가능하므로 방송사와의 공동연구를 통해 실제적인 응용분야 개발이 필요할 것으로 사료된다.