

제 3 차년도
최종 보고서

43 / 111

시각 정보 처리 기반 기술의 연구

Research for
Advanced Visual Information Processing Technology

연구기관
한국과학기술원

과 학 기 술 처

제 출 문

과학기술처장관 귀하

본 보고서를 “시각 정보 처리에 관한 연구” 사업의 세부과제 “시각 정보 처리 기반 기술의 연구” 사업의 최종 보고서로 제출합니다.

1992. 7.

주관연구기관 : 한국과학기술원

연구 기관 : 한국과학기술원

총괄 책임자 : 김 진형 (한국과학기술원 전산학과 부교수)

연구 원 : 조 창제 (한국과학기술원 전산학과 박사과정)

: 하 진영 (한국과학기술원 전산학과 박사과정)

: 조 성배 (한국과학기술원 전산학과 박사과정)

: 권 재욱 (한국과학기술원 전산학과 박사과정)

연구 조 원 : 김 선만 (한국과학기술원 전산학과 박사과정)

: 김 정현 (한국과학기술원 전산학과 석사과정)

: 김 진우 (한국과학기술원 전산학과 석사과정)

여 백

요 약 문

I. 제목

시각 정보 처리 기반 기술의 연구

II. 연구 개발의 목적 및 중요성

사람들은 언어나 글자 또는 그림으로 의사를 전달하고 있다. 그러나 현재의 컴퓨터는 주로 키보드를 통하여 미리 약속된 기호를 입력시킴으로써 대화하고 있다. 이 점이 보통 사람이 컴퓨터를 사용하는데 가장 불편하게 느끼는 점이다. 만약 컴퓨터가 말소리와 같은 음성 정보나 글씨와 같은 시각 정보를 입력으로 받아들일 수 있다면 쉽게 컴퓨터를 사용할 수 있어서 정보화 사회의 구현을 앞당길 수 있게 될 것이다.

본 연구에서는 시각 정보 처리를 위한 핵심 기술의 연구와 기본 환경의 조성이 그 목적인데, 온라인 필기 문자 인식 시스템(한글,영문,숫자,부호)을 개발하여 궁극적으로 컴퓨터 사용자에게 자연스럽게 쉬운 입력 방법을 제공해 주고 영상 처리 및 분석 기술과 모델 기반형 컴퓨터 시각 기술 등의 핵심 기술을 개발한다.

본 연구는 컴퓨터 활용의 극대화와 컴퓨터 보급의 확대에 기여하고, 사무 자동화와 산업 자동화 및 관련 산업 기술 개발을 촉진할 것이다.

III. 연구 개발의 내용 및 범위

1. 펜을 사용한 산수 교사 시스템의 개발

- 크기 및 위치에 무관한 온라인 숫자 및 기호의 인식 방법 개발
- 실시간 숫자 학습 방법의 개발
- 문제 풀이의 중간 과정 관찰에 의한 교사 시스템의 개발
- 학생 개인의 모델링에 의한 교수 전략의 수립

2. GUI 기법을 이용한 형식 문서의 구조 표현 방법에 관한 연구

- 형식 문서 처리의 연구 동향 정리 및 분석
- 그래픽 사용자 인터페이스 기법을 이용한 구조 표현 방법의 개발
- 그래픽 요소에 의한 문서 구조의 표현

3. 삼차원 물체를 인식하기 위한 모델 기반 시스템의 개발

- 매칭의 효율 개선을 위한 선형 탐색 제한 트리의 개발
- 표면 정보에 의한 삼차원 물체 표현 방법의 개발
- 여러 관점에서 본 부분에 의한 물체의 모델 구성

IV. 연구 개발 결과 및 활용에 대한 건의

- 연구 개발 결과

- 펜을 사용한 산수 교사 시스템의 개발
- GUI 기법을 이용한 형식 문서의 구조 표현 방법의 개발
- 삼차원 물체를 인식하기 위한 모델 기반 시스템의 개발

손으로 쓴 숫자 및 산술 부호, 그리고 제한된 수의 문자를 인식하는 필기 입력 산수 교사 시스템을 개발하였으며, 전표나 신상명세서 등과 같이 많은 양이 반복되어 나타나는 형식 문서(form)를 자동 입력하는 시스템에 필요한 지식 표현 방법을 개발하였다. 또한 보다 빠른 시간내에 매우 유사한 형태를 갖는 3차원 물체들을 인식할 수 있는 선행 탐색 트리 알고리즘을 개발하였다. 본 연구의 내용은 한글, 영문, 제스처를 사용하여 문서를 편집하는 시스템이나, 형식 문서로부터 필요로하는 내용을 자동 추출하여 데이터베이스로 만들 수 있는 문서 자동 입력 시스템, 또는 로봇트 시각 등 실제 사무실이나 공장에서 필요한 자동화 기술 연구를 기업체에서 수행하는데 많은 기술적 기반을 제공할 수 있다.

여 백

SUMMARY

This project is aimed at studying the essential technology and constructing fundamental environments for **Visual Information Processing**. For this purpose, we build pen-based arithmetic tutoring system(PAT), form structure representation system and 3-D object recognition system.

PAT utilizes pen as the input device to provide an easy-to-use natural interface to the student. It monitors the student's actions step by step to understand how the student solves the problem and what he does not understand. The system provides a very easy interface and it has been demonstrated that such a pen-centric mode of interaction is effective for teaching arithmetics to elementary school children.

In order to overcome the difficulty of usage of already proposed methods, we develop a form structure description method based on the graphical user interface (GUI) technique. In this system, all structure components for representing documents can be obtained by the combination of the three primitive GUI operations.

In 3-D object recognition system, objects are represented by surface patches obtained by segmenting image at depth or orientation discontinuity. Sequential constrained tree search method is usually used for recognition. But because this is less effective for object occluded by themselves and other objects, we develop a forward checking mechanism to this search method. The experimental results demonstrate significant gains in computation.

This project will contribute to the wide spread of computers, to promote office automation and the development of industrial automation technology.

여 백

CONTENTS

Chapter 1	Introduction	13
Chapter 2	Development of Pen-based Arithmetic Tutoring System	15
	Section 1 Background	16
	Section 2 Overview of PAT	24
	Section 3 Design and Implementation of PAT	35
	Section 4 Discussion and Analysis	61
	Section 5 Conclusion	64
	References	66
Chapter 3	A Study on Form Structure Representation	
	Using GUI Approach	70
	Section 1 Form Analysis and Recognition	71
	Section 2 Form Representation Using GUI Approach	77
	Section 3 Experiments and Analysis	84
	Section 4 Conclusion	90
	References	91
Chapter 4	Development of Model-based	
	3-D Object Recognition System	92
	Section 1 Recognizing 3-D Objects	
	by Forward Checking Constrained Tree Search	93
	Section 2 Generation of Attributed Graph	98
	Section 3 Matching	103
	Section 4 Experimental Results	120
	Section 5 Conclusion	135
	References	136
Chapter 5	Conclusion	140

여 백

목 차

제 1 장	서론	13
제 2 장	펜을 사용한 산수 교사 시스템의 개발	15
	제 1 절 연구 배경	16
	제 2 절 PAT의 개요	24
	제 3 절 PAT의 설계와 구현	35
	제 4 절 평가 및 분석	61
	제 5 절 결론	64
	참고 문헌	66
제 3 장	GUI 기법을 이용한 형식 문서의 구조 표현 방법에 관한 연구	70
	제 1 절 형식 문서의 분석 및 인식	71
	제 2 절 GUI를 이용한 형식 문서의 표현	77
	제 3 절 실험 및 분석	84
	제 4 절 결론	90
	참고 문헌	91
제 4 장	삼차원 물체를 인식하기 위한 모델 기반 시스템의 개발	92
	제 1 절 선형 탐색에 의한 삼차원 물체 인식	93
	제 2 절 속성 그래프의 생성	98
	제 3 절 정 합	103
	제 4 절 실험 결과	120
	제 5 절 결론	135
	참고 문헌	136
제 5 장	결론	140

여 백

제 1 장 서 론

사람들은 서로 간의 의사를 전달하기 위하여 말이나 글, 또는 그림 등을 사용한다. 따라서 이들이 가장 자연스런 대화 방법이다. 그러나 사람이 컴퓨터를 사용하려면 주로 키보드를 통하여 정해진 기호를 입력시켜야 하므로 사용하기 어렵고 불편함을 느끼게 된다. 따라서 만일 컴퓨터가 사람의 말과 같은 음성 정보나 문자, 그림 등의 시각 정보를 이해할 수 있게 한다면 누구라도 쉽게 컴퓨터를 사용할 수 있을 것이다.

이러한 사용하기 쉬운 컴퓨터를 만들기 위하여 이미 선진국에서는 문자 등의 시각 정보를 처리할 수 있는 개인용 컴퓨터 개발에 관한 연구가 활발히 진행되어 최근에는 키보드 없이 펜으로 사람의 필기를 입력받아 처리하는 키보드 없는 컴퓨터가 상품으로 개발되어 발표되고 있다. 이 컴퓨터들은 공책 정도의 크기를 가지며 배터리로 동작하고, LCD 화면에 펜으로 글씨를 쓰면 이것을 인식하여 컴퓨터가 이해할 수 있는 기호로 변환시켜 준다. Grid 시스템, SONY, CANON, 그리고 GO 등에서 이러한 컴퓨터를 시판하고 있으며, MicroSoft에서는 기존의 Windows에 펜을 사용할 수 있도록 하는 기능을 추가하였다. IBM에서도 이를 위하여 온라인 필기 문자 인식에 관한 집중적인 연구를 수행하고 있으며 국내의 여러 회사 및 연구소에서도 최근들어 활발한 연구가 수행되고 있다.

또한 컴퓨터 하드웨어 기술이 발전함에 따라 컴퓨터의 응용 분야는 단순한 자료 처리에서 벗어나 음성이나 영상 등 여러 형태의 정보를 복합적으로 처리하는 소위 멀티미디어 분야로 나아가고 있다. 따라서 시각 정보 처리 기술에 관한 연구가 필연적인데, 시각 정보 처리 및 인식에 관한 기술은 최첨단의 소프트웨어 기술로서 고도의 기술 수준을 요구하므로 선진국에서는 국가적인 차원의 집중 투자로서 연구를 수행하여 문자 및 문서 인식, 문서의 자동 검색, 로봇트 제어, 군사적인 응용 분야 등에서 이미 실용화 단계에 도달하였다.

국내에서도 이러한 시각 정보 처리의 요구가 증가함에 따라 시각 정보 처리 소프트웨

어 및 하드웨어의 수요가 급증하고 있으나 기술의 미비로 인하여 대부분 외국에서 개발된 제품이 사용되고 있다. 따라서 공통적으로 사용되는 소프트웨어를 개발하여 공급함으로써 중복 개발의 낭비를 막고 호환성의 효과를 얻을 수 있을 것이다. 또한 이를 바탕으로 시각 정보 처리 컴퓨터를 개발하여 국내 수요를 충족함은 물론 수출로 인한 외화 획득도 기대할 수 있을 것이다.

이 분야에서는 특히 한글 인식에 관한 연구가 필수적인데, 많은 대학이나 연구소의 연구에도 불구하고 한글 시각 정보 처리에 관한 수학적이고 체계적인 연구는 미약한 실정이다. 한글 처리에 관한 기술이 외국에 뒤짐으로써 발생할 수 있는 문화 종속을 방지하기 위해서라도 보다 과학적이고 효율적인 한글 인식에 관한 기초 연구가 필요하다.

제 2 장 펜을 사용한 산수 교사 시스템의 개발

본 연구에서는 펜 장치를 사용하여 국민학교 학생을 대상으로 분수식을 가르치는 PAT(Pen-based Arithmetic Tutor) 산수교사 시스템을 개발하였다. 이 PAT 시스템은 키보드를 기본 입력 장치로 하는 기존의 학습 시스템과는 달리 학생이 직접 펜으로 수식을 써 가면서 문제를 풀어 나가게 된다. 학생은 직접 스크린 상에 글씨를 쓰게 되는데, 시스템은 학생이 쓴 글씨를 바로 인식하여 스크린상에 다시 보여 줌으로써 마치 학생이 종이와 연필로 문제를 풀어나가는 것과 같은 느낌을 갖게 해 준다. 이 시스템은 학생이 쓴 수식을 인식하고 해석하여 그 문제 풀이의 중간과정을 상세히 관찰하면서 학생을 지도하게 되는데, 학생이 틀리게 문제를 풀었을 경우 그 틀린 부분과 학생이 잘못 이해하고 있는 부분을 지적해 준다. 또한 학생이 알고 있는 정도를 참고하여 학생이 잘 모른다고 판단되는 사항을 집중적으로 학습시키는 기능을 가지고 있다. 학생은 수식을 입력시키는데 있어 시스템이 자기 자신의 글씨를 잘 인식할 수 있도록 언제든지 자신의 글씨를 학습시키는 기능을 가지고 있다. 본 PAT 시스템은 학생이 보다 사용하기 쉬운 인터페이스를 제공함으로써, 기존 시스템의 문제점인 수식 입력의 불편함을 해결하였고, 문제 풀이의 중간과정까지도 상세히 지도할 수 있다는 장점을 가지고 있다.

제 1 절 연구 배경

1 산수 교사 시스템

BUGGY[Bro78]는 단순히 학생의 오류를 찾아내는 것 뿐만 아니라 학생이 왜 그와 같은 오류를 범했는가를 설명해 줌으로써 선생이 학생의 오류를 진단할 수 있도록 도와 준다. 이것은 자연수의 감산과 같은 산수의 매우 기본적인 영역을 다루고 있다. 또한 학생들이 계산을 하면서 범하는 오류를 모두 수집하여 이를 학생의 오류 모형으로 이용하였다. 그리고 수집한 오류를 분석하여 이러한 잘못을 발생시킨 계산 절차를 찾아내고 이를 오류형태에 따라 연결시킨 절차 네트워크(procedual network)으로 표시하였다. 오류를 찾아내는 방법은 답이 틀린 문제를 모든 절차 네트워크에 입력하여 결과를 먼저 계산한 후에 학생이 계산한 결과와 같은 답이 나온 네트워크로 학생의 오류를 설명해 준다.

DEBUGGY[Bur82]는 BUGGY를 기반으로 한 시스템으로서 두개의 진단 시스템을 제공한다. 하나는 표준테스트(standard test)를 다루는 것이고, 다른 하나는 학생과 시스템이 정보를 주고 받음으로써 학생의 상태를 진단하는 것이다. 전자의 목표는 가급적 많은 학생의 오류를 설명하기 위하여 최적 오류 집합 (optimum buggy set)을 구축하는 것이다. 후자는 진단을 위하여 AI 패러다임중 하나인 생성과 시험 (generate-and-test)을 사용한다. 즉, 시스템은 문제뿐만 아니라, 가정 자체도 자동적으로 생성하고 경우에 따라 그것을 시험한다.

LMS(Leeds Modelling System)[Sle81, Sle82a, Sle85, Sle87]은 정규칙(rule) 또는 오류규칙(mal-rule)을 사용하여 학생의 행동을 추론하거나 예측한다. Burton의 연구에 의하면[Bur82], 어린이가 산수문제 풀이과정에서 저지르는 오류의 대부분은 학생이 선생이 가지고 있는 지식을 다 갖추지 못했기 때문이 아니라, 자기 나름대로의 잘못된

지식을 가지고 있기 때문이라고 한다. 이 잘못된 지식에 해당하는 것을 규칙의 형태로 표현한 것을 오류규칙(mal-rule)이라고 하고, 올바른 지식에 대한 규칙을 정규칙(correct rule) 또는 그냥 규칙(rule)이라고 한다. LMS에서는 주어진 문제의 답과 학생이 제시한 해답을 미리 표현된 정규칙 또는 오류 규칙이 생성한 답과 비교함으로써 학생의 행동과 의도를 파악했다. 그러나 LMS는 학생이 오류를 범했을 때에, 그에 해당하는 잘못된 지식을 깨우쳐주는 기능을 갖지 못했다. PIXIE 학생 모델링 시스템[Sle87]은 LMS를 기반으로 하여 대수 문제의 영역을 다루는 ITS로서, 학생이 제시한 수식을 파싱하고 해석하며 그 결과를 미리 정의된 정규칙 또는 오류규칙과 비교하여 평가한다. LMS와 마찬가지로 학생의 답이 정규칙이 생성한 답과 맞지 않으면, 시스템은 오류모델을 찾기 위하여 오류규칙이 생성한 답과 비교하여 학생의 잘못된 지식을 찾아 낸다.

GEOMETRY[And85]는 기하의 증명문제를 다루는데, 앞에서 설명한 ITS들과는 달리 Anderson의 ACT 학습이론(ACT theory of learning)을 기본으로 한다. 학습 시스템은 학생의 행동을 자세히 살펴 보면서, 학생이 오류를 범했을 때에 학생의 행동을 중단시킨다. 학생은 연속적으로 기하학 정리를 적용함으로써 증명 문제를 풀어간다. 이때 학습 시스템은 학생이 제시한 각 단계의 입력을 올바른 적용의 경우와 틀린 적용의 경우의 예가 포함된 데이터 베이스와 비교하며, 학생이 잘못된 규칙을 적용하였을 때에는 풀이를 중단시키고 그에 맞는 처방을 한다.

본 연구에서 다루려고하는 분수식에 관련된 ITS에 대해 선행된 연구를 살펴보면 다음과 같다. Sleeman[Sle87]은 그의 PIXIE 프로젝트에서 가산과 감산을 포함한 다양한 분수 연산에 관한 오류규칙들을 조사하였으며, De Gery 팀[DeG85]에서는 가산, 승산, 제산 및 분수의 약분문제에서 학생들이 사용한 잘못된 방법에 대한 진단을 집중적으로 다루었다. 잘못 이해된 개념들을 오류규칙의 형태로 코드화 하고, 이것을 이용하여 학생의 지식 상태를 평가하였다. 이 연구에서 De Gery는 대부분의 학생들이 분수 자체의 의미를 생각하지 않고, 그것이 맞든 틀리든 간에 단지 표시기호로서 의미를 생각한다고 결론 지었다. 이 연구는 Hasemann[Has81]에 의해서도 뒷받침되는데, 그는 "Concepts in

Secondary Mathematics and Science”라는 연구 프로그램에서 대부분의 학생들은 분수의 가산과 감산을 할때 어떤 규칙을 왜 사용해야 하는가에 대한 이해없이 단지 임의의 규칙을(그것이 옳은 경우든, 혹은 틀린 경우든) 적용하여 풀려고 한다고 결론지었다.

Nwana는 분수의 가산과 감산을 다루는 ITS의 설계를 위한 선행 연구로서 학생이 종이와 연필을 사용하여 문제를 풀도록하는 실험을 하였다[Nwa88]. 실험은 우선 간단한 예비시험(pre-test)을 거친후 분수의 가산과 감산을 하도록 했는데, 그 실험에서 체계적인 오류(systematic error)가 전체 오류의 상당한 부분을 차지함을 보였다. 학생의 오류가 임의의 절차로서 생성될 수 있을때 그 오류는 체계적(systematic)이라고 한다. 이러한 체계적인 오류는 오류의 형태를 일정한 법칙으로 나타내는 것이 가능함을 의미하고, 결국 정규칙과 오류규칙은 학생의 모델링에 좋은 도구로 사용할 수 있음을 의미한다. 그러나 분수식과 같이 여러가지 개념이 복합된 문제 영역의 경우는 전통적인 오류규칙의 방법을 그대로 따르는데 문제점이 있다. 앞에서 언급한 대부분의 ITS들, 예를 들어 LMS[Sle81], BUGGY[Bro78], DEBUGGY[Bur82]등은, 학생이 제시한 마지막 답만을 사용하여 그들의 오류를 진단한다. 그들이 다루는 문제 영역은 어느정도의 간단한 문제이므로 가능했지만, 분수와 같이 복합된 영역의 경우에는 필요한 규칙의 수가 기하급수적으로 증가하게되어 이러한 규칙을 이용한 진단은 사실상 불가능하게 된다. Nwana는 결론적으로 복잡한 문제의 경우에는 여러 단계의 중간 과정을 이용해야함을 주장하였다.

Nwana는 이와 같은 문제점을 해결하기 위해 실제로 구현된 FITS(Fraction Intelligent Tutoring System)이라는 ITS를 통해서 가능한 연산을 미리 열거하고 학생이 그중 하나를 선택한 후, 그에 해당하는 연산을 하여 답하도록하는 상호 작용 방식을 사용하여 여러 단계에 걸쳐 복잡한 개념의 문제를 해결할 수 있도록 하였다[Nwa91]. 그러나 이와 같은 메뉴 방식의 문제 제시 형태는 학생의 행동을 미리 메뉴에 정의된 임의의 연산으로 제한한다는 점에서 학생의 지식 상태를 정확히 파악하는데 매우 좋은 방법이라고 볼수 없다. 그보다는 학생이 자유롭게 연산과정을 기술할 수 있고 그것을 ITS에서 감시하여 오류를 파악하는 것이 보다 적합한 형태라고 생각된다.

2 펜을 기반으로하는 인터페이스

1970년대 초반에 Alan Kay는 사람들이 동적으로 정보를 다룰 수 있는 혁신적인 컴퓨터를 제안하였다. Dynabook이라고 하는 이 컴퓨터는 노트북 정도의 작은 크기로서 휴대가능하고, 높은 해상도의 그래픽 스크린을 가지며, 그 스크린 위에 직접 펜장치로서 입력이 가능한 것이 특징이다. 그러나 당시의 기술 수준으로서는 이러한 하드웨어의 설계가 불가능하여 실제적인 구현이 어려웠다. 이와 병행하여 1960년대 후반부터는 손으로 쓴 글씨를 인식하는 기술에 관한 연구가 AI의 한 주제로서 계속 연구되어 왔으나 제반 기술의 부족으로 실효를 거두지 못하다가 최근에 와서야 비로소 전자 타블렛과 컴퓨터 계산 능력의 발전으로 실생활에 응용될 수 있을 만큼 많은 발전을 이루었다. 결국 Dynabook의 개념과 필기체 문자 인식의 연구를 결합하여 PLI(Paper Like Interface)라는 새로운 인터페이스가 등장하게 되었다[Wol89, Cho89]. 문자 그대로 이 인터페이스는 펜과 종이의 개념을 닮은 것으로서, 사용자가 기존의 키보드 기반의 컴퓨터를 써서 힘들거나 부자연스러웠던 여러가지 일을 자연스럽게 할 수 있도록 만들었다.

펜 장치를 사용함으로써 얻을 수 있는 장점은 여러가지가 있다. 우선 입력하는 방법에 있어서 자연스럽다는 것이다. 이것은 사람들이 어릴적부터 펜과 종이에 익숙해져 있어 직업이나 연령에 구분없이 누구나 별도의 교육없이도 사용이 가능함을 의미한다. 두 번째로, 펜은 특정 위치에서 직접 조작하는 것이 가능하다는 것이다. 키보드나 마우스의 경우에는 스크린 상의 원하는 부분에 어떤 조작을 가하기 위해서는 그 위치까지 커서를 옮겨야 하는 불편함이 따랐으나 펜의 경우에는 직접 그 위치에 가서 원하는 조작을 하면 된다. 세 번째로, 펜은 손으로 지적하는 곳과 스크린 상에 작용하는 곳이 일치하기 때문에 다른 입력 장치보다 정밀도가 높으며, 한번의 동작으로 여러가지 일을 가능하게 한다. 또한 펜은 포인터의 움직임을 알아내기 위하여 마우스처럼 여러가지 중간 단계의 기계적 또는 전기적인 과정을 거치지 않기 때문에 글씨, 그림, 제스처(gesture)등 연속적인 입력을 하는데 적합하다.

이러한 PLI를 구현하는데는 다음과 같은 핵심기술이 필요하다[Cho89]. 우선, 하드웨어의 관점에서 보면, 휴대가 가능하도록 가볍고 작게 만들기 위하여 높은 성능을 발휘하는 단일칩의 설계가 가능해야 하며, 고도로 집적화된 메모리칩이 필요하다. 또한 종이의 같은 효과를 나타내기 위해서 입력과 출력을 하나로 결합할 수 있는 LCD 타블렛의 개발이 필요하다. 그리고 펜으로 그은 흔적을 그대로 스크린 상에 나타내기 위한 전자 잉크의 개발도 필요하다. 소프트웨어의 관점에서 보면, 기본적으로 필기체 문자 인식이 가능해야 하며, 스크린 상의 데이터를 편집하기위한 제스처의 인식도 필요하다. 이러한 핵심 요소기술들이 결합되어야만 진정한 의미의 PLI를 구현할 수 있다.

펜을 기반으로 하는 인터페이스는 응용 분야가 무척 다양하나 그중 교육에 이용된 예를 몇가지 살펴보면 다음과 같다.

- 펜멘십(penmanship, learning to write) 그림 2.1과 그림 2.2에서 보듯이 학생이 글씨를 처음 배울때 점선으로 그려진 글자 자형을 따라 펜으로 그리는 연습을 하거나, 표본을 보고 그대로 옮겨 쓰는 연습을 하는 것이다. 컴퓨터는 학생의 펜이 글자 자형에서 벗어나면 부저를 울림으로써 학생이 올바르게 글씨를 쓰도록 유도한다.
- 크로스워드 퍼즐(crossword puzzle) 그림 2.3에서 보듯이 서로 연결된 사각형 안에 글자를 써 넣음으로써 단어를 알아 맞추는 게임이다. 각 사각형 안에 펜으로 직접 글씨를 쓰면 시스템이 그 문자를 인식하고, 인식된 결과를 이용하여 게임을 진행시킨다.
- 악보 그리기 (music note editing) 그림 2.4에서 보듯이 화면에 나타난 오선지위에 학생이 음표를 그리는 것이다. 컴퓨터는 음표를 인식하여 그 위치에 올바르게 그려준다. 학생은 자기가 그린 악보의 선율을 직접 컴퓨터를 통하여 연주해 볼 수도 있다.



그림 2.1: 펜멘십(penmanship) : 점선 따라 쓰기

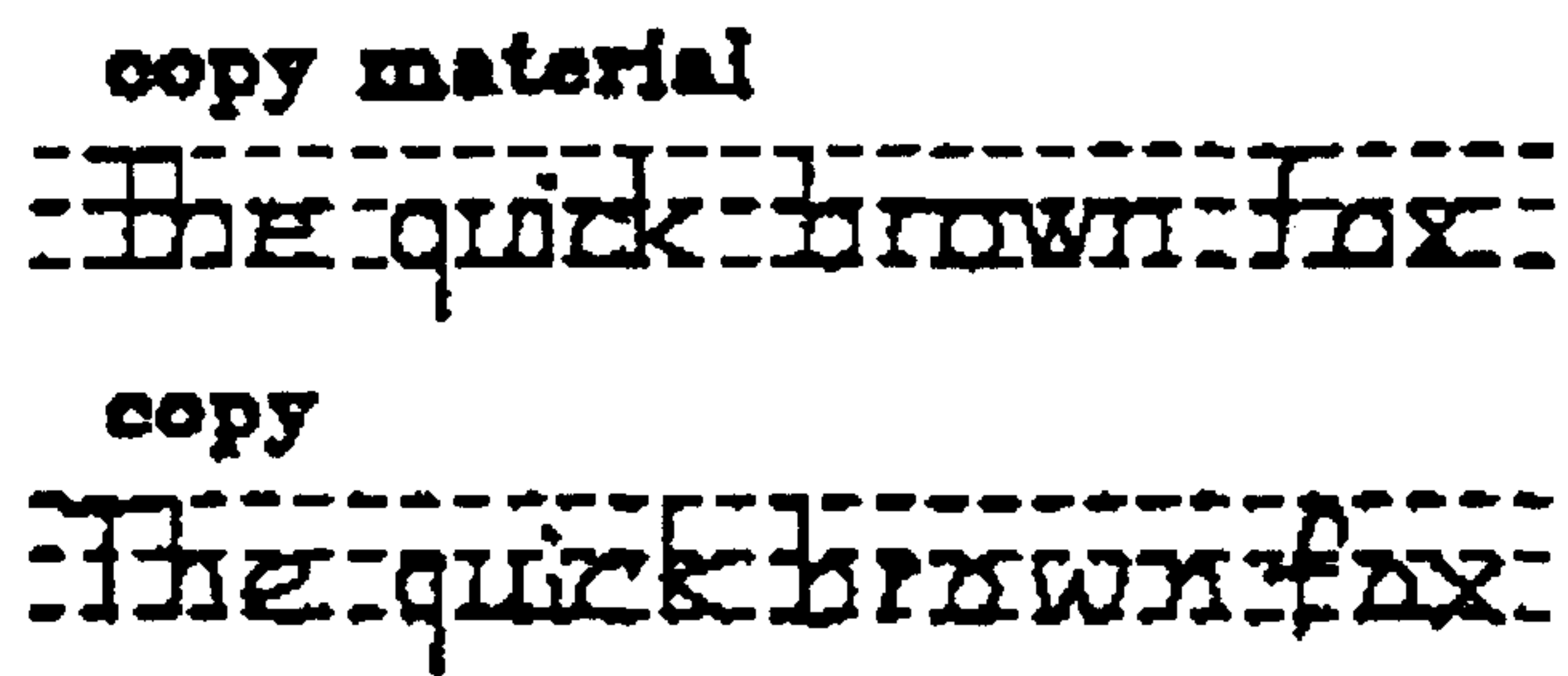


그림 2.2: 펜멘십 : 옮겨 쓰기

4 - Across
Baby cat

1 - Down
Baby duck

Quit Prototypes Refresh Answer Enter

그림 2.3: 크로스워드 퍼즐



그림 2.4: 악보 그리기

제 2 절 PAT의 개요

PAT(Pen-based Arithmetic Tutor)는 국민학교 산수 과정에서 분수식의 계산을 교육 목표로 하며, 사칙연산을 포함한 모든 연산의 전 과정을 관리하는 ITS이다. PAT는 펜을 기본 입력 장치로 하여 학생이 자연스럽게 사용할 수 있도록 하며, 학생이 문제를 풀어가는데 최소한의 제한을 두도록 설계되었고, 이를 통하여 기존의 산수 교사 시스템이 가지고 있는 단점을 개선했다. PAT는 UNIX Workstation상에서 X11/MOTIF를 이용한 윈도우 시스템을 기반으로 하여 구현되었으며, 그래픽을 사용한 인터페이스를 제공함으로써 학생이 친근감을 가지고 사용할 수 있도록 하였다. 그림 2.5은 PAT의 예를 보여준다. 전체 시스템은 C++ 객체 지향 언어를 사용하여 구현되었다.

1 PAT의 구조

그림 2.6은 PAT의 전체 구조를 보여 준다.

- 사용자 인터페이스 모듈(User Interface Module) 학생과 PAT 사이의 상호 작용을 관리한다. 이 모듈에서는 학생이 쓴 수식을 인식하기 위한 필기체 문자 인식 시스템과 PAT가 생성하는 도움말 또는 메시지를 출력하기 위한 도움말 관리기(Message Manager)를 포함한다. 메시지 관리기는 문제 해결 모듈이 생성하는 메시지를 텍스트, 그래픽, 애니메이션, 음성등의 형태로 재구성하여 학생에게 제공한다.
- 문제 해결 모듈(Problem Solving Module) 학생의 행동을 감시하는 모듈이다. 학생이 제시한 수식을 전 단계의 수식과 비교하여 학생의 행동을 알아낸다.

PAT(Pen-based Arithmetic Tutor)

File Option Font Train Lesson

복잡한 분수식에 관한 공부

학생이름

다음 식을 계산하세요.

$$\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3} = ?$$

			MOVE	
=	과정	힌트	?	설명

중간 계산 과정

$$\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3} = \frac{5}{2} + \frac{1}{6} - \frac{1}{3} =$$

수식은 여기에 쓰세요

$$\frac{6}{8} - \frac{1}{3}$$

분수의 덧셈에서 분자와 분모를 각각 더했군요. $\frac{5}{2} + \frac{1}{6} \rightarrow \frac{6}{8}$
 분수의 덧셈에 어려움을 느끼는 것 같군요.
 학생은 그것을 다시 공부해야겠어요. O.K.

그림 2.5: PAT의 예

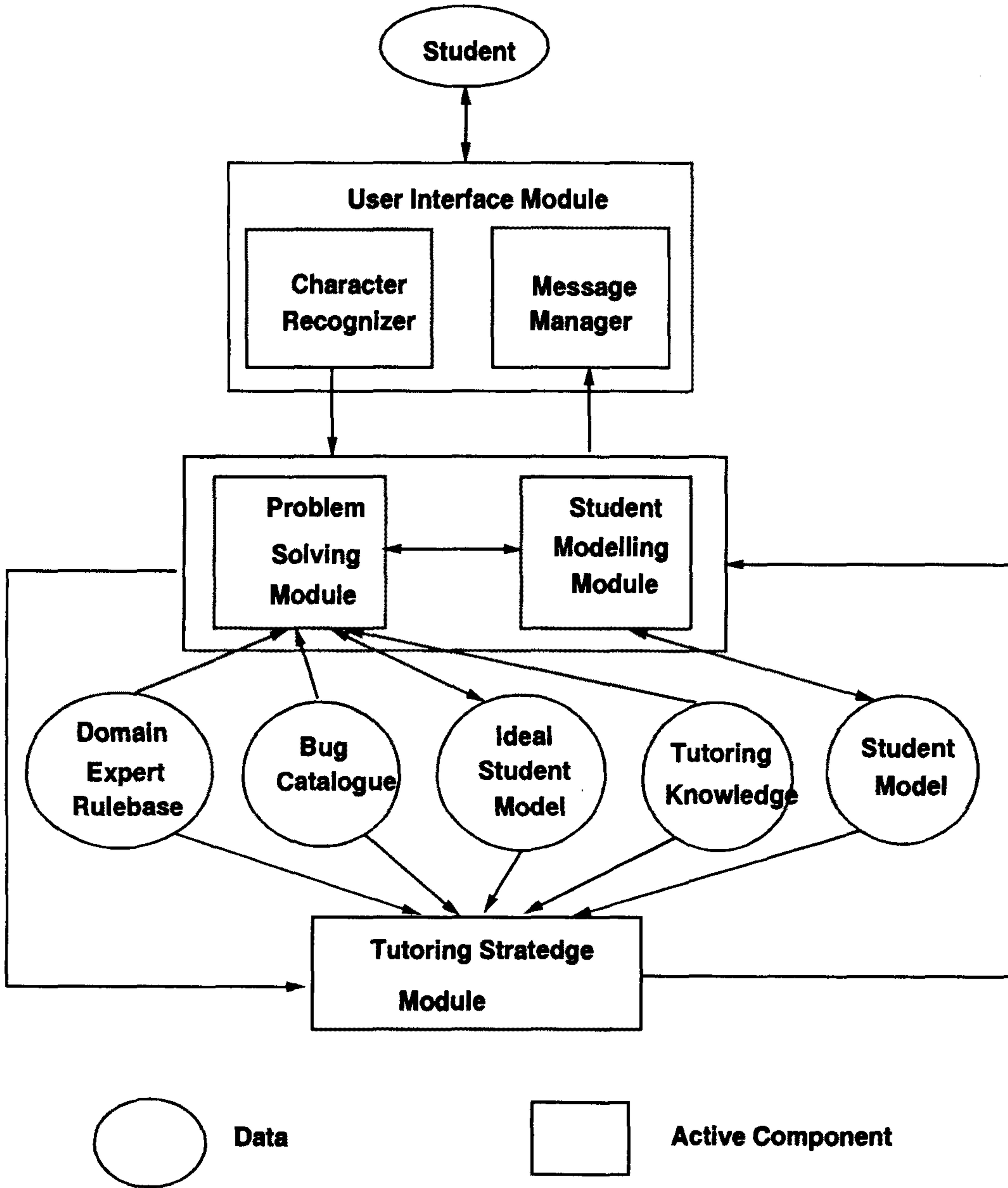


그림 2.6: PAT의 전체 구조

- 학생 모델링 모듈(Student Modelling Module) 문제 해결 모듈에서 찾아낸 학생의 행동을 영역 지식베이스(Domain Expert Rulebase) 또는 오류 목록(Bug Catalogue)의 규칙에 적용하여 학생의 의도를 파악하고 잘못된 지식을 찾아낸다. 그리고 그 결과를 이용하여 학생의 지식 상태를 동적으로 유지 관리한다.
- 교육 전략 모듈(Tutoring Strategy Module) 학생의 지식 상태를 참조하여 다음에 취할 교육 내용을 결정한다. 학생의 상태에 따라서 문제를 선별하며, 필요한 경우에 문제 풀이 도중에도 도움말을 생성한다.
- 영역 지식베이스(Domain Expert Rulebase) 분수식을 푸는데 필요한 모든 지식을 정규칙의 형태로 모아 놓은 데이터베이스이다.
- 오류목록(Bug Catalogue) 분수식을 푸는데 흔히 발견되는 오류패턴들을 오류규칙의 형태로 모아 놓은 데이터베이스이다.
- 모범 학생 모델(Ideal Student Model) 특정 문제에 대한 올바른 풀이 과정을 나열해 놓은 것으로서, 각 단계의 수식은 전 단계의 수식에 대해서 어느 부분을 어떻게 계산했는가가 표시된다. 모범 학생 모델은 하나의 문제에 대해서 문제 해결 모듈에 의해서 필요한 경우에 자동적으로 생성되고 제거된다. 이것은 학생 모델링과 교육 전략에 이용된다.
- 학생 모델(Student Model) 현재 학생의 지식 상태를 개략적으로 나타낸다.
- 교육 내용(Tutoring Knowledge) 학생에게 제시할 문제의 데이터베이스이다. 각 문제는 단원별로 구성되어 있고, 각 단원은 하나의 개념과 관련된다.

다음 네모안에 알맞은 수를 넣으세요.	끝
$\frac{2}{7} + \frac{3}{5} = \frac{\boxed{10}}{35} + \frac{\boxed{27}}{35}$ $= \frac{\boxed{}}{35}$	
답을 한 후 <끝>을 누르세요	

그림 2.7: 네모안에 수를 써넣는 문제 제시형태

2 PAT의 교육 방법과 문제 제시 형태

산수 과목의 교육에 있어서 학생의 독창적인 사고를 위해서 선생이 문제 풀이 도중에 자주 간섭하는 것은 바람직하지 않은 방법이다. 학생에게는 자신이 직접 수식을 계산해 나감으로써 스스로 문제를 해결하는 방법을 터득할 수 있는 기회를 주는 것이 좋다. 이점을 감안하여, PAT에서 제안하는 교육 방법은 가급적 시스템의 도움없이 학생이 스스로 문제를 해결하도록 하는 방법이다. 시스템은 학생이 오류를 범했거나 도움을 요청한 경우에만 학생에게 도움을 준다.

PAT에서 학생에게 문제를 내주는 형태는 크게 두가지 종류가 있다. 첫번째는 특정한 수 또는 기호만을 응답으로 요구하는 형태이며, 시스템은 그 입력이 시스템이 원하는

<p>다음 식을 계산하세요.</p> $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3} = ?$	<table border="1" style="margin: auto; border-collapse: collapse;"> <tr> <td style="width: 20px; height: 20px; text-align: center;">⌨</td> <td style="width: 20px; height: 20px; text-align: center;">☞</td> <td style="width: 20px; height: 20px; text-align: center;">☠</td> <td style="width: 20px; height: 20px; text-align: center;">MOVE</td> <td style="width: 20px; height: 20px; text-align: center;">24</td> </tr> <tr> <td style="width: 20px; height: 20px; text-align: center;">=</td> <td style="width: 20px; height: 20px; text-align: center;">과정</td> <td style="width: 20px; height: 20px; text-align: center;">힌트</td> <td style="width: 20px; height: 20px; text-align: center;">?</td> <td style="width: 20px; height: 20px; text-align: center;">설명</td> </tr> </table>	⌨	☞	☠	MOVE	24	=	과정	힌트	?	설명
⌨	☞	☠	MOVE	24							
=	과정	힌트	?	설명							
<p>중간 계산 과정</p> $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3} = \frac{5}{2} + \frac{1}{6} - \frac{1}{3} = \frac{5 \times 3}{2 \times 3} + \frac{1}{6} - \frac{1}{3} =$											
<p>수식은 여기에 쓰세요</p> <div style="text-align: center; font-size: 2em;"> $\frac{15+1}{6} - \frac{1}{3}$ </div>											

그림 2.8: 수식을 한꺼번에 입력하는 문제 제시형태

답인가 만을 비교한다. 따라서 PAT는 학생의 입력에 대해서 별도의 처리를 거칠 필요가 없다. PAT에서는 학생의 입력을 사각형 형태의 작은 윈도우를 이용하여 받아들인다. 이러한 문제 제시 방법은 여러 단계의 중간 과정이 필요 없는 문제, 가령 두 수중 어느 것이 큰가에 대해 부등호를 입력으로 받는 문제, 또는 16과 12의 최소 공배수는 무엇인가를 묻는 문제 등에 주된 입력방식으로 이용될 수 있다. 또한 수식의 어느 한 부분을 감추어 놓고, 그 부분에 들어가야 하는 수를 답으로 묻는 경우에 이용될 수 있다. 이와 같은 경우는 학생에게 새로운 개념을 소개하려고 할때 문제 풀이의 전과정을 보여 주되, 그중 일부만 감추어 놓고 그 곳에 알맞은 수를 쓰도록하여 학생에게 보다 쉬운 방법으로 개념을 가르치는데에 이용된다. 또한 학생의 지식을 좀더 정확히 파악하기 위한 방법으로, 여러 개의 개념이 복합되지 않은 하나의 개념만을 시험하는 목적으로도 이용된다(그림 2.7).

또다른 형태의 문제 제시 방법은 여러 단계의 중간 계산 과정이 필요한 수식을 다루

는데 이용되는 형태로서, 각 중간 단계마다 학생이 제시한 수식 전체를 입력으로 받는다. 학생은 수식을 임의의 순서로 표기할 수 있으므로 순서에 영향을 받지 않도록 입력을 처리할 수 있어야 하며, 받아들인 수식은 문제 해결 모듈에서 이용할 수 있는 형태로 변환된다(그림 2.8).

3 PAT의 사용자 인터페이스

PAT의 사용자 인터페이스는 크게 세가지의 윈도우로서 이루어진다. 하나는 문제를 나타내는 윈도우로서 학생이 풀어야 할 문제와 알고 있어야 할 사항을 제공한다. 다른 하나는 학생이 직접 답을 쓰는 윈도우이다. 중간 과정이 필요한 문제의 경우에는 그 학생이 계산한 중간 과정을 보여주는 윈도우가 별도로 첨가되며, 학생은 그 윈도우를 이용하여 필요한 계산을 할 수 있다. PAT는 이 중간과정 윈도우에 쓰여진 기호는 인식하지 않는다. 또한 PAT가 생성한 메시지를 보여주는 윈도우가 있다. PAT는 이 윈도우를 이용하여 학생이 알아야 할 사항을 지시하게 된다.

그밖에도 PAT는 보다 효율적이고 자세한 설명을 위해서 별도의 윈도우를 사용한다. 그림 2.9은 학생에게 특정 개념의 문제를 해결하는 방법을 적당한 예와 함께 그래픽을 사용하여 설명해 주는 것을 나타내고 있고, 그림 2.10은 현단계에서 학생이 풀어나가야 할 계산 과정을 단계별로 설명과 함께 보여준다.

그림 2.11의 윈도우는 학생이 현단계에서 취한 행동을 분석한 결과를 종합하여 보여주고 있다. 그림에서 보듯이 수식의 변화된 부분과 그것의 의미를 표시하고 그 변화된 부분을 분석하여 학생이 알고 있다고 판단되는 개념들을 표시해 준다. 또한 학생이 여러 단계의 중간 과정을 생략한 경우에는 그 생략된 과정의 수식을 찾아 보여준다. 이 윈도우는 학생에게 직접 도움을 주는 것은 아니지만 학생의 행동을 간결하게 모니터링하는데도

Help Concept

공통분모분수의 덧셈을 하는 방법

$$\frac{\triangle}{\square} + \frac{\circ}{\square} = \frac{\triangle + \circ}{\square}$$

(1) 분모는 그대로 놔두고 분자끼리만 더합니다.
 (2) 분수가 가분수일 경우에는 가분수를 대분수로 고칩니다.

예

$$\frac{2}{7} + \frac{3}{7} = \frac{2+3}{7} = \frac{5}{7}$$

Now, I know it!

그림 2.9: 특정 개념의 설명을 위한 윈도우

Help

$$\frac{4}{5} \times \frac{1}{3} \div \frac{2}{15}$$

↓ $\frac{4}{5} \times \frac{1}{3}$ 에서 두 분수를 곱하세요

$$\frac{4}{15} \div \frac{2}{15}$$

Previous
Next
Done

그림 2.10: 계산 과정의 설명을 위한 윈도우

Monitoring Student Status

Action

Change $\frac{2}{5} + \frac{1}{3}$ → $\frac{11}{15}$

Message

<i>Concepts</i>	<i>Intermediate</i>
<input type="text" value="Fraction Addition"/> <input type="text" value="Fraction Reduction"/> <input type="text" value="Get LCM"/> <input type="text" value="Fraction Scaling"/> <input type="text" value="Integer Multiplication"/> <input type="text" value="Equiv Fraction Addition"/>	$\frac{2 \times 3}{5 \times 3} + \frac{1 \times 5}{3 \times 5}$ $\frac{6}{15} + \frac{5}{15}$ $\frac{6+5}{15}$ $\frac{11}{15}$

Diagnosis

NO. of Action

그림 2.11: 학생의 행동을 모니터링하는 윈도우

움을 준다.

4 PAT의 예

이절에서는 학생이 분수식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 을 풀어가면서 PAT와 나누는 대화를 이용하여 시스템의 동작을 설명한다. 실제 시스템에서 학생은 모든 수식을 펜장치로 직접 윈도우에 쓰면서 입력한다. PAT가 생성하는 도움말은 특정 윈도우에 표시되거나 별도의 다이얼로그로서 텍스트, 그래픽 또는 애니메이션을 이용하여 설명한다. 기호 // 뒤에 나오는 말은 특정 예를 설명하기 위한 코멘트이다.

PAT 분수식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 을 계산하세요.

학생 $\frac{4}{8} \times \frac{1}{4} - \frac{1}{3}$

PAT 학생은 불가능한 연산을 계산한 것 같군요. 다시 하세요.

학생 HINT

PAT 가산, 감산, 승산, 제산이 합쳐진 분수식은 우선 승산과 제산을 먼저 계산하되, 왼쪽에 있는 것을 먼저 계산합니다. 계속하세요.

학생 $\frac{5}{2} + \frac{1}{6} - \frac{1}{3}$

PAT 잘 했어요. 계속 하세요.

학생 $\frac{6}{8} - \frac{1}{3}$

PAT 틀렸어요. 학생은 두 분수를 더할때, 분자는 분자끼리 분모는 분모끼리 더하는 것이라 생각하는 것 같군요. 다시 하세요.

학생 $\frac{5}{12} - \frac{1}{3}$

PAT 역시 틀렸어요. 이번에는 분자는 분자끼리 분모는 분모끼리 서로 곱했군요. 학생이 두 분수를 더하는 문제를 잘 못 푸는것 같군요. 그 부분을 다시 공부해야겠어요.

// PAT는 현재의 문제를 포기하고, 두 분수의 가산에 관한 문제를 내준다.

PAT $\frac{2}{3} + \frac{1}{4}$ 를 계산하세요.

학생 $\frac{3}{7}$

PAT 틀렸어요. 분수의 가산에서 분자는 분자끼리 분모는 분모끼리 더했군요. 분수의 가산에서는 우선 두 분수의 분모를 같게 통분한 후에 분모는 그대로 나두고, 분자만 더하는 것입니다. 다시 하세요.

// 이 부분에서 PAT는 그래픽을 이용한 문제풀이 과정과 적당한 예를 보여줌으로써 학생이 잘 이해할 수 있도록 설명한다.

학생 $\frac{11}{12}$

PAT 잘 했어요. $\frac{2}{3} + \frac{1}{4}$ 를 올바르게 풀었어요.

∴

// PAT는 학생의 지식 상태에 따라서 두 분수의 가산에 관한 문제를 만족한 상태가 될 때까지 반복해서 내준다.

∴

// PAT는 학생이 분수식을 푸는데 필요한 여러 개념들을 올바르게 이해하고 있다고 판단하면 원래 문제로 되돌아 와서 학생이 풀던 문제를 다시 내준다.

∴

PAT 분수식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 를 다시 풀어 봅시다.

제 3 절 PAT의 설계와 구현

1 산수 교과과정의 기술

하나의 연산을 하는데 기본적으로 필요한 개념을 연산 개념(operation concept) 또는 간단히 개념(concept)이라고 한다. 학습 시스템은 학생이 시스템이 목표로하는 연산을 완전히 알고 있다고 판단될 때 까지 반복해서 그 개념을 교육한다. PAT에서 목표로 삼고 있는 연산 개념은 사칙 연산을 포함한 분수식의 계산이다.

하나의 연산개념은 여러개의 하부 연산개념, 즉 하부개념(subconcept)으로 세분화할 수 있다. 특히 ITS가 다루는 문제의 영역이 복합적인 개념으로 이루어진 경우에는 이와같이 개념을 세분화시키는 것이 자연스럽다. 기존의 ITS중에서 DEBUGGY[Bur82], FITS[Nwa91]등은 이와같은 개념/하부개념의 구조를 학습자 모델링에 이용하였다. PAT에서 하나의 개념은 보통 하나이상의 하부개념과 기술(skill)로서 이루어진다. 기술은 새로운 개념을 위해서 하부개념을 이용하는 방법을 말한다. 예를 들어 그림 2.12에서 보는 바와 같이 두 분수의 가산을 위한 연산개념은 두 분수를 공통 분모로 통분하는 하부개념과 통분된 두 분수를 더하는 하부개념으로 세분화할 수 있고, 이 두개의 하부개념을 이용할 수 있는 능력으로서 기술이 필요하다. 또다시 통분의 개념은 두 분모의 최소공배수를 구하는 하부개념과, 분수의 분자와 분모에 같은 수를 곱해서 분수를 변환시키는 하부개념이 필요하며, 이 하부개념들은 제각기 또다른 하부개념을 가질 수 있다. 하나의 개념은 더 이상의 하부개념을 갖지 않을 수도 있는데, 그 경우에 우리는 특별히 원자개념(atomic concept)이라고 부른다. 결국 하나의 개념은 트리 형태의 계층적 구조를 하고 있음을 알 수 있다. 하나의 하부개념은 여러 개념들에 의해서 공유될 수 있기 때문에, 좀더 정확히 나타내자면 격자(lattice)의 구조를 이룬다고 할 수 있다. PAT에서

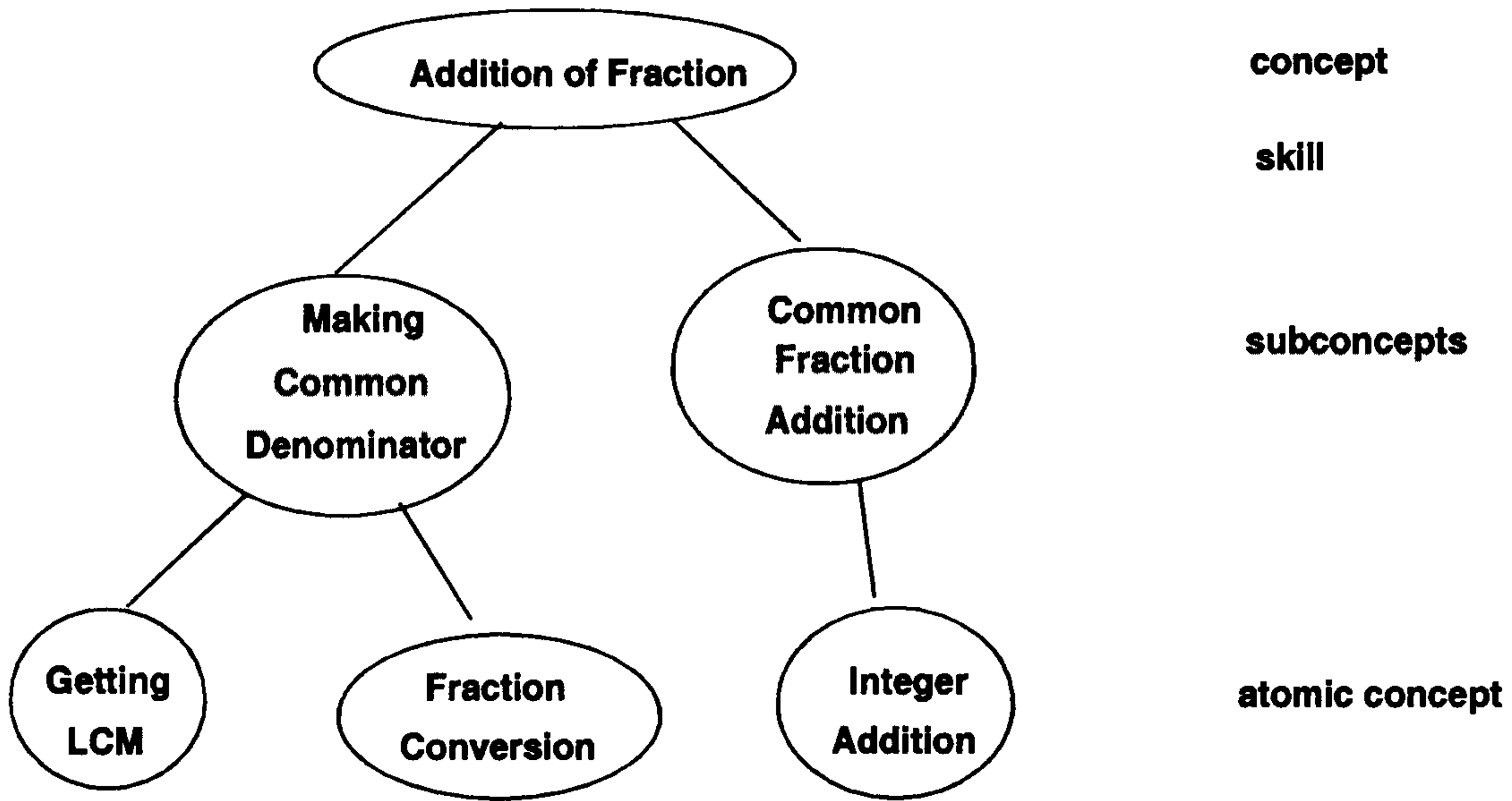


그림 2.12: 두 분수의 가산에 대한 연산 개념

분수식의 계산에 필요한 개념은 총 21개의 개념으로 이루어져 있다. 표 2.1는 이들 개념의 계층적 구조를 표로 나타낸 것이다. 표에서 보면, 다섯개의 개념만이 더 이상의 하부 개념을 갖지 않는 원자개념이고, 그 외의 개념은 모두 하나 이상의 하부개념을 가지고 있음을 알 수 있다. 표에 표현된 개념의 계층적 구조는 완전히 정확한 표현이라고 보기는 어렵다. 이는 하나의 개념을 이루는 하부개념을 정확히 정의하기가 힘들며, 하나의 개념을 이해한다는 것은 본질적으로 많은 배경과 기본 지식이 필요하기 때문이다. 그러나 산수 교과과정을 살펴 보면, 하나의 새로운 개념을 설명하기 위해서 이미 알고 있는 기존의 개념을 이용하는 것을 볼 수 있는데, 이것은 개념의 계층적 구조가 충분히 합당한 방법임을 시사해 준다. 이와같은 개념의 계층적 구조는 각각의 개념을 학생이 잘못 이해할 가능성이 있는 지식 단위로 표현함으로써, 표현 방식에 있어서 자연스러울 뿐만 아니라, 보다 훌륭한 추상화(abstraction)를 제공할 수 있다.

이와 같은 연산개념의 계층적 구조는 학생의 지식 상태를 모델링하는데 중요한 역할을 한다. 하나의 개념을 알고 있다는 것은 그 개념이 포함된 하부개념과 기술을 모두 이해하고 있다는 것을 의미한다. 또한 학생이 하나의 개념을 잘 모른다고 판단될 때, 학생은 그 개념의 하부개념중 최소한 하나를 모르거나 그 하부개념을 어떻게 이용하는 지를 모른다고 판단할 수 있다. 예를 들어, 두 분수를 더하는 문제에서 학생이 어려움을 느낀다고 판단될 때, 학습시스템은 그것의 하부개념인 통분을 하는 것과 공통분모의 두 분수를 더하는 개념을 학생이 제대로 알고 있는지를 조사해 볼 필요가 있다. 만일, 그 두 하부개념에 대해서 학생이 만족할 만한 수준의 지식을 가지고 있다고 판단되면, 학습 시스템은 학생이 두 개념을 이용하여 분수를 더하는 방법을 모른다고 생각할 수 있는 것이다. 결국 학습 시스템은 학생의 취약한 개념을 집중적으로 지도하는 교육 전략을 세움으로써, 보다 효과적인 교육이 가능하다.

2 펜을 사용한 사용자 인터페이스

수식을 키보드로 입력하는 것은 앞에서 언급한 바와 같이 자연스럽지 못한 방법이다. CAI를 위해서 수식을 키보드로 입력하는 방법이 [Nak89]에 의해서 연구된 바 있으나, 수식을 쓰기 위해서 임의의 위치에 커서를 옮기는데 많은 키조작이 필요하며, 어린 학생이 그와 같은 방법을 이용하기는 매우 힘들다. PAT는 학생의 모든 입력을 펜 장치를 사용하여 받는다. 학생은 크게 두가지 방법으로 입력하게 되는데, 하나는 특정 윈도우나 아이콘을 지적하거나 클릭함으로써 원하는 목록을 선택하는 것이고, 다른 하나는 온라인 문자 인식을 이용하여 원하는 문자를 입력하는 것이다. 본 절에서는 PAT에서 사용되는 문자 인식과 인식된 결과를 문제 해결 모듈에서 사용 가능한 형태로 바꾸는 방법에 대해서 간단히 알아본다.

연산 개념	하부 연산 개념
분수식	분수 가산, 분수 감산, 분수 승산, 분수 제산, 자연수식
분수 제산	제산을 승산으로 변환, 분수 승산
제산을 승산으로 변환	-
분수 승산	자연수 승산, 약분하기
분수 감산	분수 통분, 공통분모 분수의 감산
분수 가산	분수 통분, 공통분모 분수의 가산
분수 통분	최소공배수 구하기, 분수 변환
분수 약분	최대공약수 구하기, 자연수 제산
분수 변환	자연수 승산
공통분모 분수의 감산	자연수 감산
공통분모 분수의 가산	자연수 가산
가분수를 대분수로 변환	자연수 가산, 자연수 승산
대분수를 가분수로 변환	자연수 제산, 자연수 가산
최대공약수 구하기	자연수 제산
최소공배수 구하기	자연수 승산
자연수 식	연산자 우선 순위
연산자 우선 순위	-
자연수 제산	자연수 승산
자연수 승산	-
자연수 감산	-
자연수 가산	-

표 2.1: 분수식 계산을 위한 연산 개념의 계층적 구조

산수 교사 시스템에서 펜을 이용하여 수식을 받아들일 경우에 문자 인식에 있어서 다음과 같은 사항이 고려되어야 한다.

- 문자 인식은 가능한 빨리 이루어져야 한다. 학생이 쓴 글자는 곧바로 인식되어 글씨를 쓴 위치에 그 결과를 보여줌으로써 학생이 수식을 올바르게 입력하고 있는지 알 수 있게 해야 한다.
- 오인식을 하는 것보다는 인식을 못하는 것이 낫다. 현재까지 개발된 인식 알고리즘으로는 사람이 쓴 문자를 100퍼센트 인식 할 수는 없다. 따라서 당연히 인식에 실패하는 경우가 생기는데, 그 경우 다른 문자로 잘못 인식하는 것보다는 인식을 포기하는 것이 더 바람직하다. 산수 교사 시스템에서 오인식을 하는 경우에 학생의 의도와는 달리 다른 수식이 입력될 수 있으며, 결국 학생 모델링을 정확히 할 수 없다. 또한 오인식으로 인하여 그 문자를 다시 지우고 다시 입력하는 불편함을 지니게 되어, 어린 학생들에게 심리적으로 좋지 못한 영향을 끼칠 수 있다.
- 문자 인식과 제스처인식을 결합하는 것이 바람직하다. 수식을 입력하는 동안에 입력된 문자 중에서 한 부분을 지우거나 다른 위치로 옮기고 싶은 경우에 제스처가 사용된다. 현재 개발된 대부분의 문자 인식 방법은 근본적으로 제스처의 인식방법과 차이가 있기 때문에 실제 응용 시스템에서는 문자 인식과 제스처 인식을 별개의 모드로 두고 사용하는 경우가 대부분이다. 하지만 산수교사 시스템에서는 사용자가 어린 학생들이라는 점을 감안하여 볼때, 입력 방법에 있어서 두개의 모드가 존재한다는 것은 그만큼 불편함이 따르게 되고, 그 결과로 많은 실수를 유도할 가능성이 있어 좋지 못한 방법이다. 따라서 가능하면, 문자 인식과 제스처 인식을 결합하여 하나의 모드로 가능하게 하는 것이 좋다.

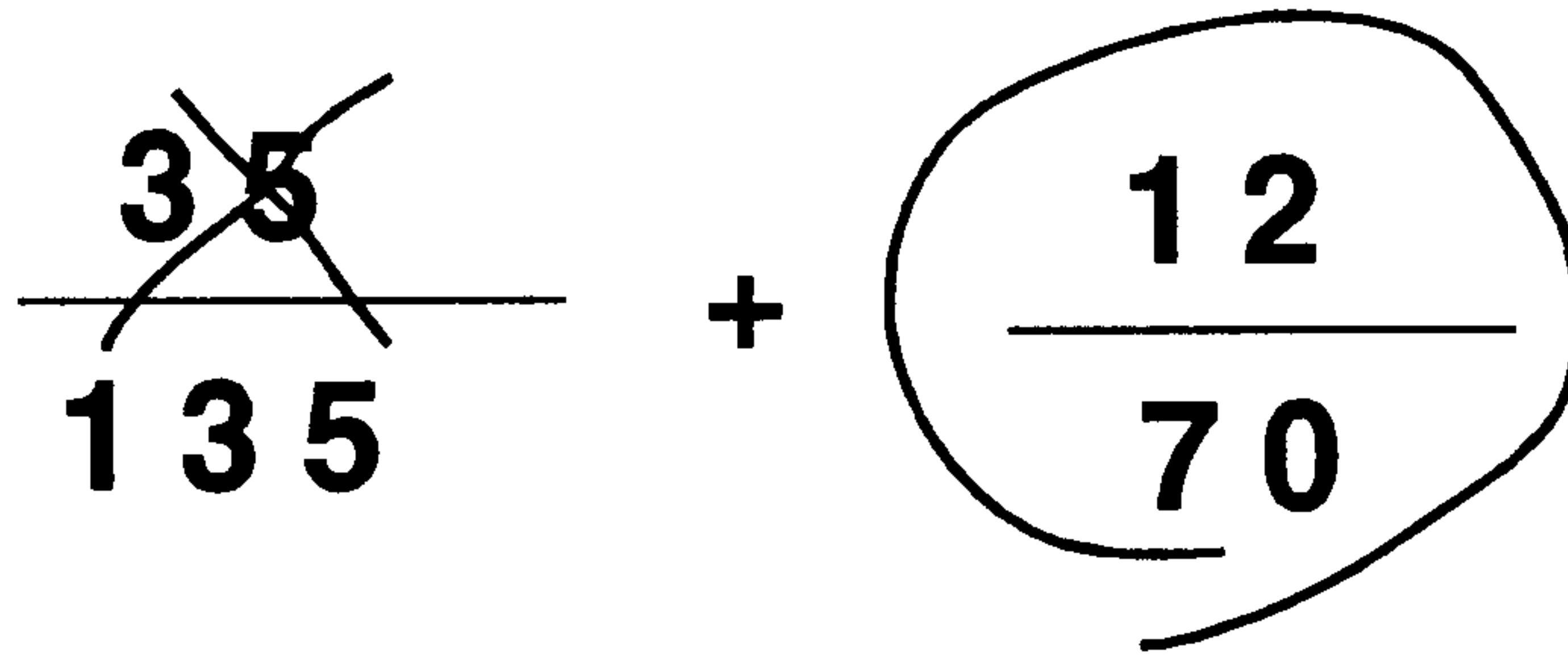


그림 2.13: PAT에서 사용되는 제스처

2.1 온라인 문자 인식

PAT에서는 앞에서 설명한 조건들을 충족시키는 온라인 문자 인식기를 이용한다. 이 인식 시스템은 수식에 쓰이는 문자나 기호뿐만 아니라, 인식된 문자를 편집할 수 있는 제스처 기호를 동시에 인식한다. PAT에서 사용되는 제스처 기호는 삭제 기호와 그룹핑(grouping) 기호로서 그림 2.13에 나타난 바와 같다. 그룹핑 기호는 여러개의 인식된 문자나 기호를 하나로 묶어서 한꺼번에 지우거나 다른 곳으로 옮기는데 이용된다. 그림에서 보듯이 삭제기호는 승산기호(\times)와, 그룹핑기호는 숫자 0과 그 모양이 같기 때문에 그것들을 각각 구분하는 것이 필요하다. 문자와 제스처의 구분은 문맥정보를 이용한다. 즉, 이미 인식되어 있는 문자위에 \times 또는 0 모양의 글씨가 쓰여지면, 시스템은 그것을 승산기호 또는 숫자 0으로 인식하지 않고 삭제기호와 그룹핑기호로 인식한다. 전체 구조는 그림 2.14에 나타나 있다. 인식기는 우선 입력 획 별로 분류 과정을 거친 후에 그것들의 조합으로 모델을 생성해서 이미 저장되어 있는 모델과 비교하여 인식한다. 각 모델은 획(stroke) 식별자의 오토마타로 구성되어 있어 학생이 글씨를 쓰는 즉시 바로 인식하도록 되어 있다. 세부적인 인식 방법은 여기서 다룰 문제가 아니므로 생략한다.

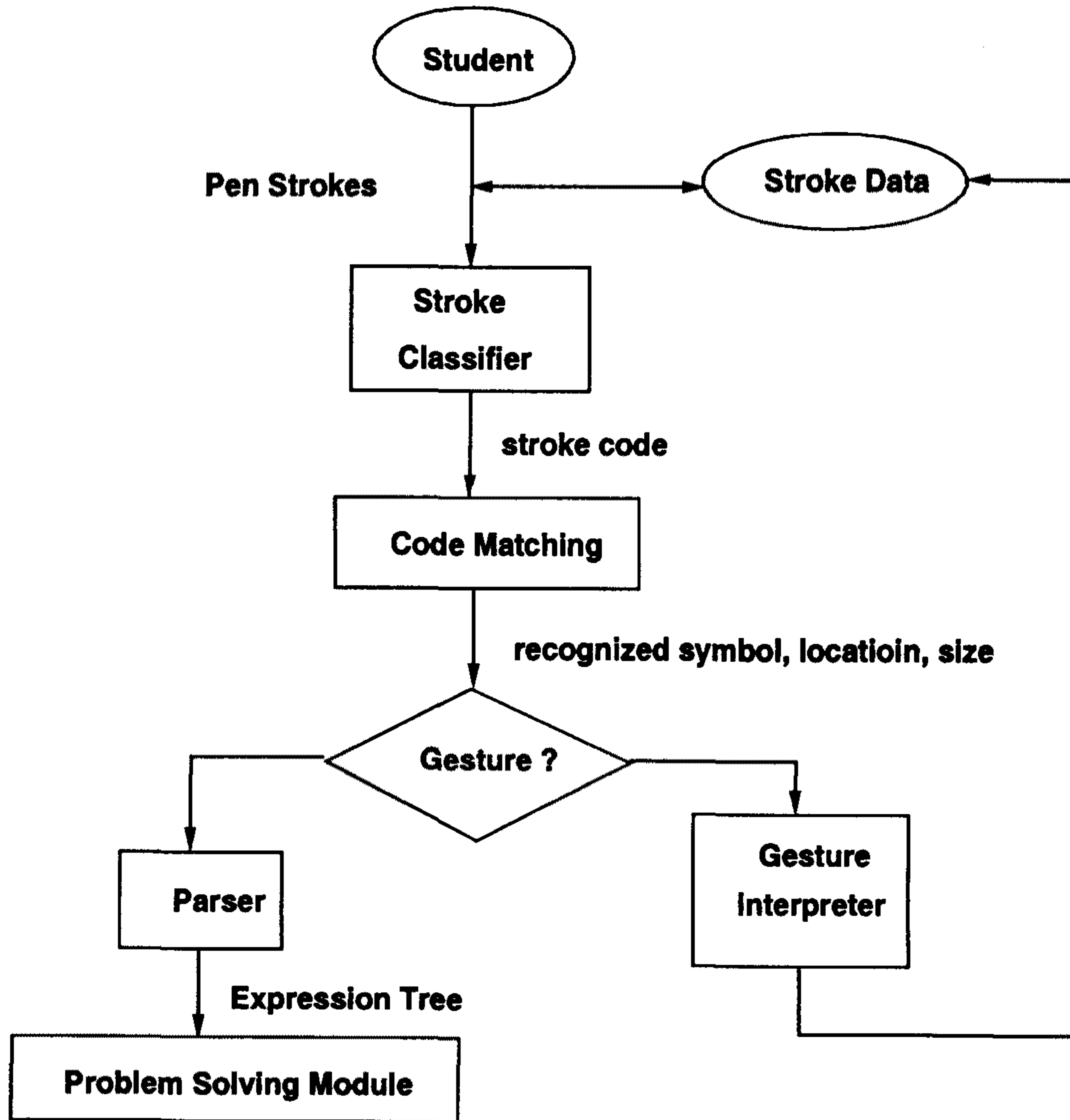


그림 2.14: 인식기의 구조

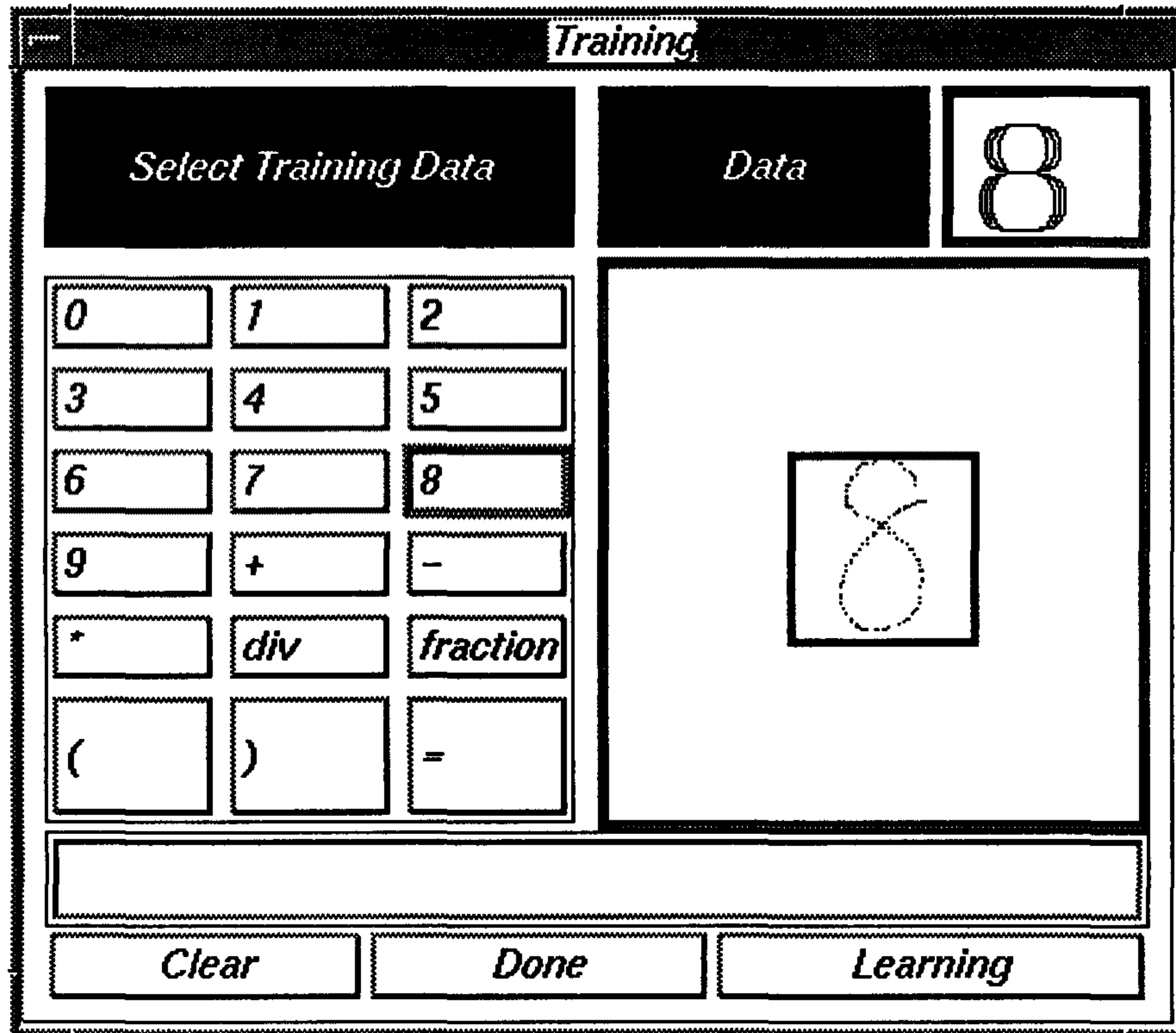


그림 2.15: PAT의 문자 학습기

2.2 실시간 문자 학습기

PAT에서는 학생이 자신의 글씨 모양을 시스템에게 학습시킬 수 있다. 이 기능은 시스템이 사용자의 글씨에 적용할 수 있는 능력을 의미하는 것으로 온라인 문자인식을 이용한 응용 프로그램에서는 반드시 갖추어야 할 사항이라고 할 수 있다. 문자 학습기에서는 기존 모델에 없는 새로운 모양의 글씨가 입력되면, 그 문자에 대한 코드를 새로 생성하고, 그 코드를 포함한 오토마타를 새로 구성한다. 그림 2.15는 PAT에서 제공하는 문자 학습기를 보여준다.

2.3 파서 (Parser)

파서는 학생이 입력한 수식의 인식된 각 문자에 대해서 그들의 크기와 위치 관계를 이용하여 문제 해결 모듈이 이용할 수 있는 자료 구조의 형태로 바꾸는 역할을 한다. 학생은 수식을 임의의 순서로 표기할 수 있기 때문에 쓰여진 순서에 상관없이 일정한 모양으로 변환시키는 기능이 필요하다. 파서는 우선 수식이 쓰여진 순서에 영향을 받지 않도록 하기 위해서 인식된 문자들의 위치관계를 이용하여 문자의 순서를 재조정한다. 이것은 Clustering 알고리즘에 의해서 간단히 이루어지며, 그 결과로 수식은 1차원 배열의 형태로 표시된다. 손으로 쓰여진 수식을 파싱하는 방법은 [Fau90]에 의해서도 연구된 바 있으나, 그 연구에서는 수학에 나오는 모든 기호를 다루기 때문에 보다 복잡한 방법이 필요했다.

Clustering 알고리즘은 그림 2.16에 나타난 바와 같다. Clustering 알고리즘은 우선 세로 방향으로 Cluster를 구성한 후에, 구성된 Cluster를 X 축 방향에 따라 정렬시킨다. Cluster는 임의의 방향에 대해서 서로 겹쳐지는 부분이 있는 원소들을 하나로 합쳐놓은 것을 말한다. 정렬된 각 Cluster에 대해서 가로방향으로 다시 Cluster를 구성하고 이번에는 Y 축 방향으로 정렬한다. 이와같은 Cluster 구성과 정렬 과정을 각 Cluster의 원소의 갯수가 하나가 될 때까지 계속 반복한다. 이러한 방법으로 수식이 쓰여진 순서에 상관없이 문자들을 일관성있게 재배열 할 수 있다. 그림 2.17은 수식 $\frac{5}{32} + \frac{12}{47}$ 에 대하여 Clustering 알고리즘을 적용한 결과를 나타낸 것이다.

Clustering 알고리즘에 의해서 1차원으로 재배열된 수식은 이제 분수식을 위한 파서에 의해서 파싱되고 그 결과로 문제 해결 모듈이 이용할 수 있는 자료 구조 형태로 변환된다. 그 자료 구조에 대해서는 3절에서 자세히 소개하기로 한다. 파서에 쓰이는 문법은 부록에 소개되어 있다.

```

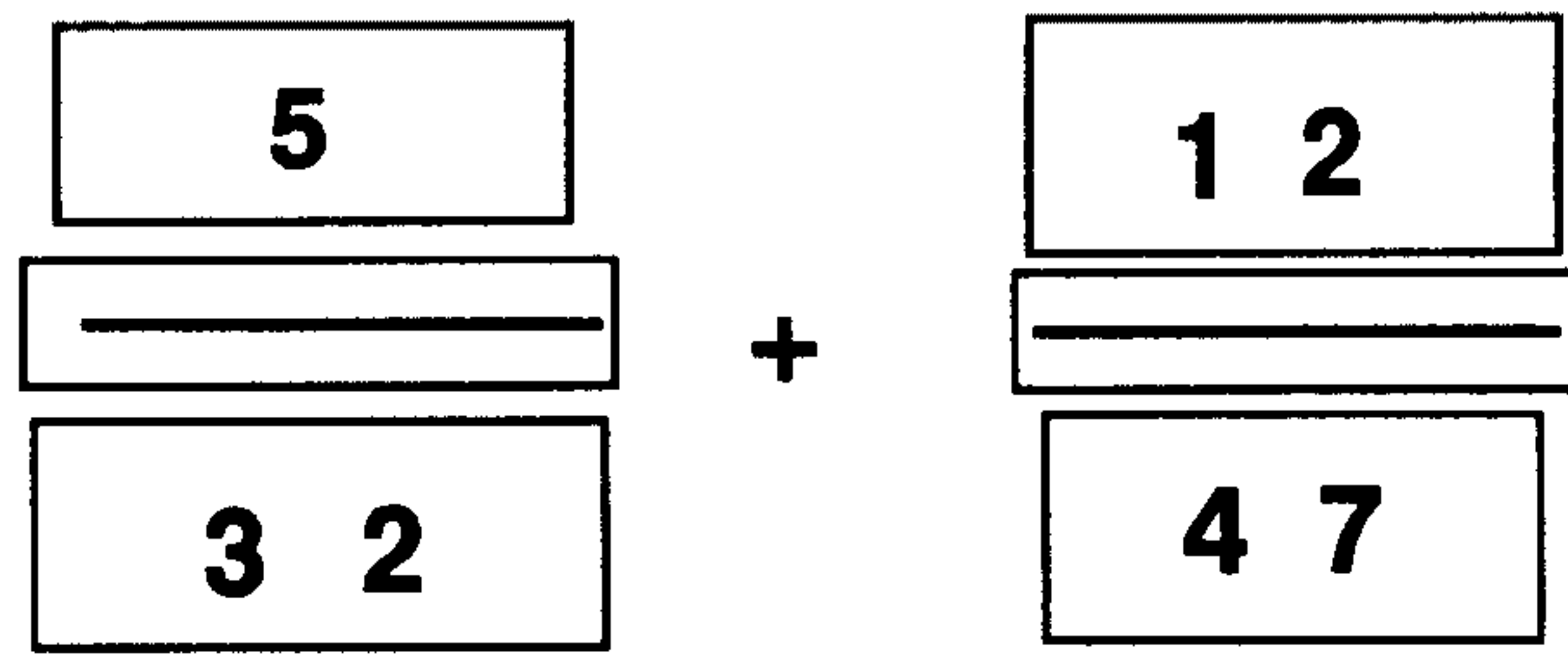
procedure Clustering(recognized symbols)
    make cluster for all recognized symbols;
    VerticalClustering(cluster);
end;

procedure VerticalClustering(cluster)
    if the number of symbols in the cluster is 1
        return;
    else
        make subclusters of the cluster for vertical direction;
        sort subclusters according to their xcoordinates;
        foreach subcluster begin
            HorizontalClustering(subcluster);
        end
    end;
end;

procedure HorizontalClustering(cluster)
    if the number of symbols in the cluster is 1
        return;
    else
        make subcluster of the cluster for horizontal direction;
        sort subclusters according to their ycoordinates;
        foreach subcluster begin
            VerticalClustering(subcluster);
        end
    end;
end;

```

그림 2.16: Clustering 알고리즘



_____ after vertical-clustering
 _____ after horizontal-clustering

Result : { (5) (—) (3) (2) } (+) { (1) (2) (—) (4) (7) }

그림 2.17: 수식 $\frac{5}{32} + \frac{12}{47}$ 를 Clustering 알고리즘에 적용한 결과

3 지식 표현

3.1 수식의 표현

학생이 제시한 수식은 파서에 의해서 문제 해결 모듈이 이용할 수 있는 트리 형태의 자료 구조로 변환된다. PAT에서 수식은 두가지 형태의 트리로 표현되는데, 하나는 한 노드의 자식 노드의 갯수가 임의의 수로 될 수 있는 일반 트리이고, 하나는 자식 노드의 갯수가 2인 이진 트리이다. 일반 트리와 이진 트리는 서로 변환이 가능하며, 시스템은 이 중 하나만을 유지한다. 그림 2.18과 그림 2.19은 수식 $\frac{5}{2} + \frac{2}{3} * \frac{1}{4} - \frac{1}{3}$ 을 각각 일반 트리 와 이진 트리로 나타낸 것을 보여준다.

(1) 일반 트리[Hor87]

수식의 일반 트리 표현은 두개의 수식을 비교하여 서로 다른 부분을 찾아 내는데 이용된다. 일반 트리는 한 노드의 자식노드의 갯수가 정해져 있지 않은 트리를 말한다. PAT에서 사용된 일반 트리의 한 노드는 그림에 나타난 바와 같이 값(value), 수식의 기호(symbol), 전위 연산자(preceding operator)의 3개의 필드로 구성되어 있다.

- 값(value) 그 노드 또는 그 노드가 포함한 서브트리 전체의 값을 나타낸다.
- 기호(symbol) 노드가 종단 노드(terminal node)인 경우에는 수의 기호(symbol)를 표현하며, 종단 노드가 아닌 경우에는 자식 노드의 포인터를 가진다. 자식 노드의 갯수는 하나 이상으로 미리 정해져 있지 않다.
- 전위 연산자(preceding operator) 노드가 영향을 받는 연산자를 나타낸다. 노드는 이 연산자에 의해서 왼쪽에 위치한 피연산자에 연산을 가하게 된다. 주의할 것은 한 서브 트리에 있어서 가장 왼쪽에 위치한 자식 노드는 전위 연산자로 널(null)을 가지게 되며, 그 노드 앞에 있는 연산자는 그 서브트리 전체에 영향을 주는 연산자로, 그 서브트리의 루트 노드의 전위 연산자에 해당한다.

수식을 이와같은 구조의 일반 트리로 나타내는 것은 다음과 같은 특징이 있다. 우선, 트리의 같은 레벨에 위치한 노드들은 같은 우선 순위를 갖는다. 같은 우선 순위의 두 노드는 서로 연산이 가능하며, 경우에 따라서는 서로의 위치가 뒤바뀔 수도 있다. 우선 순위가 다른 노드보다 높은 노드는 그 레벨에서 서브트리의 형태로 표현되어 다른 레벨에 위치하게 된다. 서로 다른 레벨에 위치한 두 노드끼리는 계산할 수 없다. 결국 이와같은 표현은 전체적으로 계층적인 모듈화된 구조를 이루게 한다. 이와 같은 일반 트리의 표현은 두개의 수식을 서로 비교하여 변화된 부분을 찾아내는데 효과적으로 이용될 수 있다.

(2) 이진 트리[Hor87]

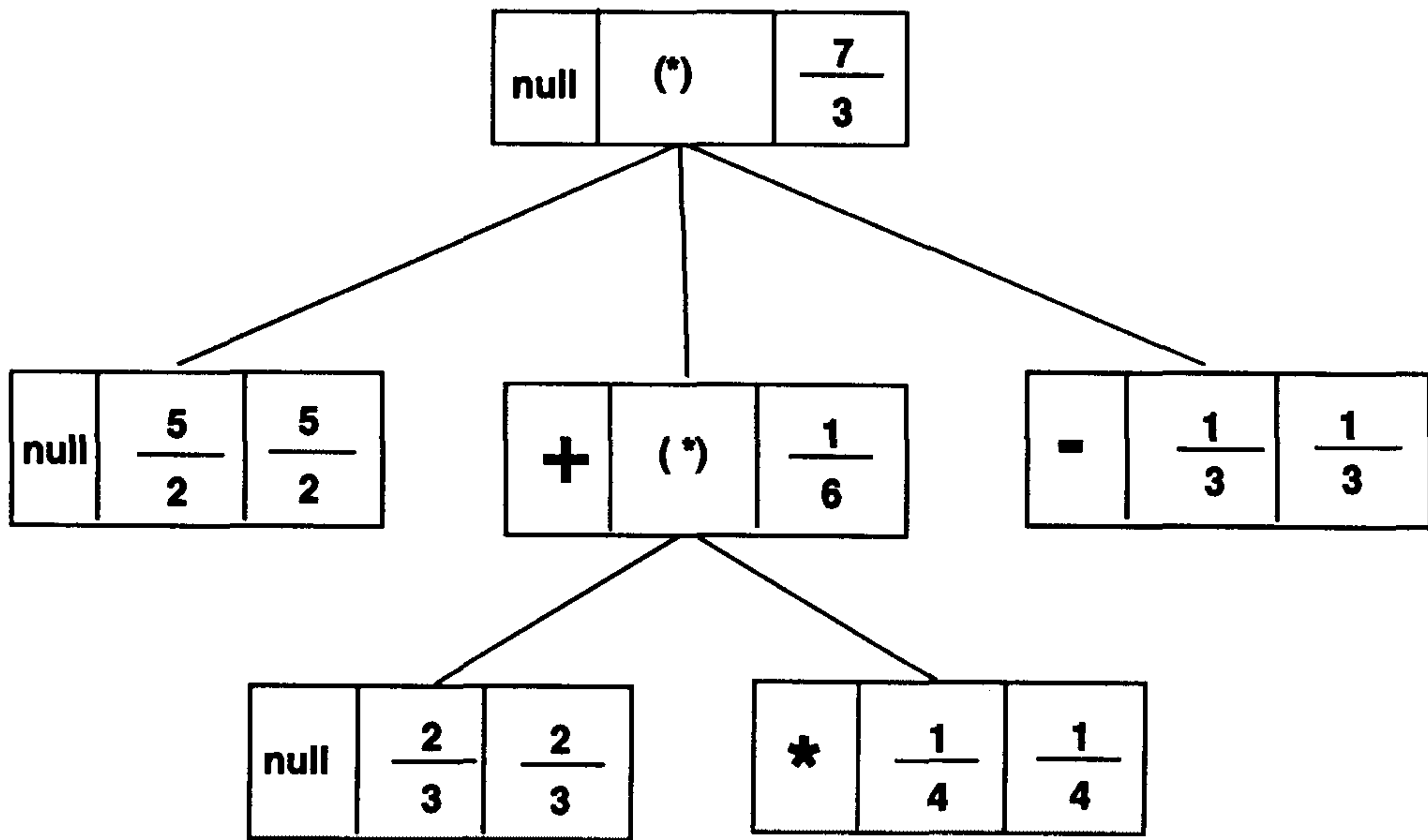


그림 2.18: 수식 $\frac{5}{2} + \frac{2}{3} * \frac{1}{4} - \frac{1}{3}$ 의 일반 트리 표현

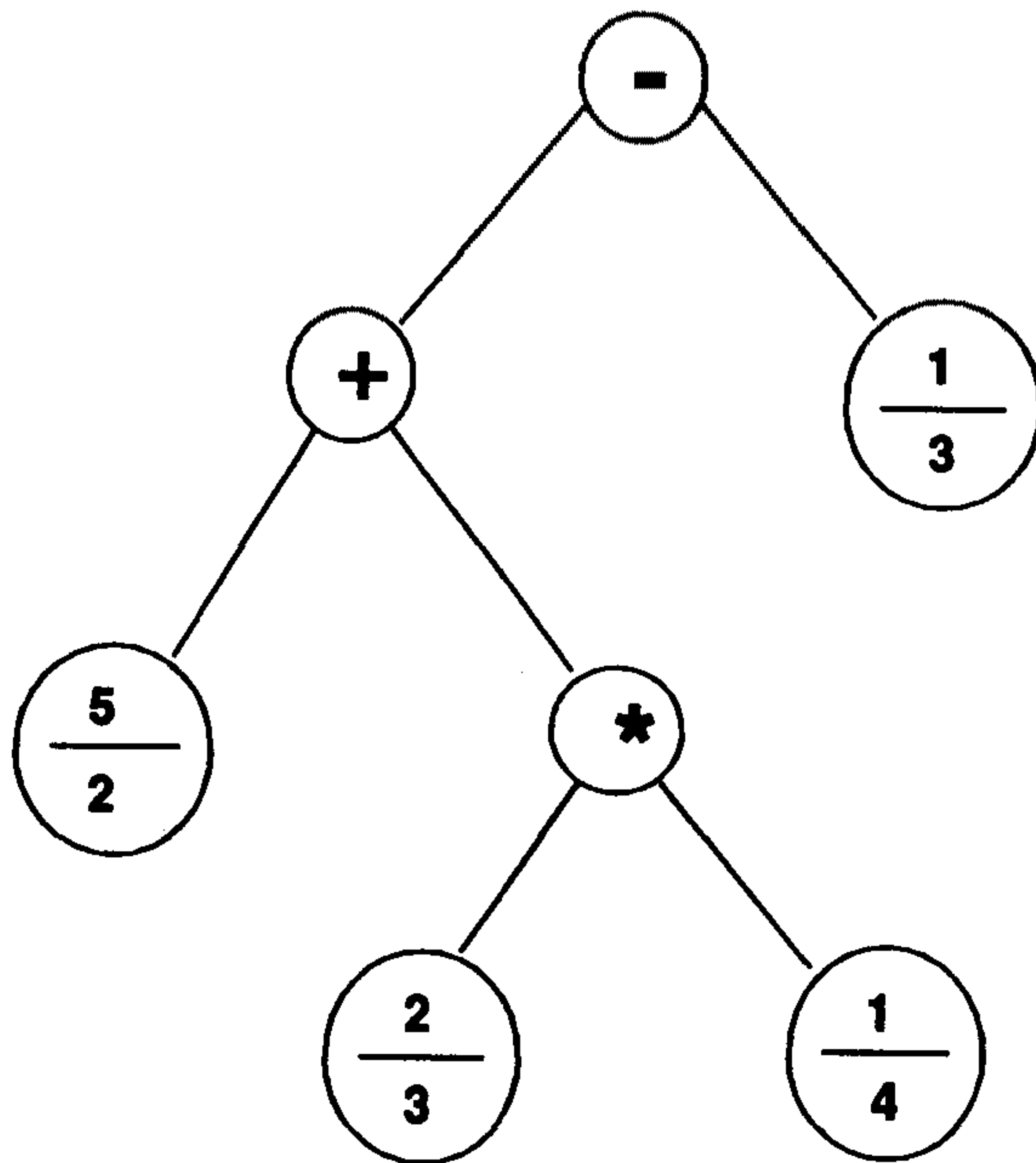


그림 2.19: 수식 $\frac{5}{2} + \frac{2}{3} * \frac{1}{4} - \frac{1}{3}$ 의 이진 트리 표현

수식의 이진 트리 표현은 두개의 트리를 완전 매칭하는데 이용된다. 그림에서 보는 바와 같이 수식에서 연산자는 모두 비종단노드(non-terminal node)로 표현되며, 피연산자는 종단 노드가 된다. 각 연산자는 우선 순위에 따라 트리상의 위치가 결정되며, 높은 우선 순위의 연산자가 트리의 아래쪽에 위치하게 된다. 같은 우선 순위의 연산자는 왼쪽에 있는 연산자 일수록 높은 우선 순위를 갖게 된다. 이와같은 수식의 표현은 문제 해결 모듈이 학생이 제시한 수식의 변화된 부분을 찾은 후에 그것의 의도를 알기 위하여 정규칙 또는 오류 규칙에 적용할 때에 사용된다.

3.2 영역 지식 모듈 (Domain Knowledge Module)

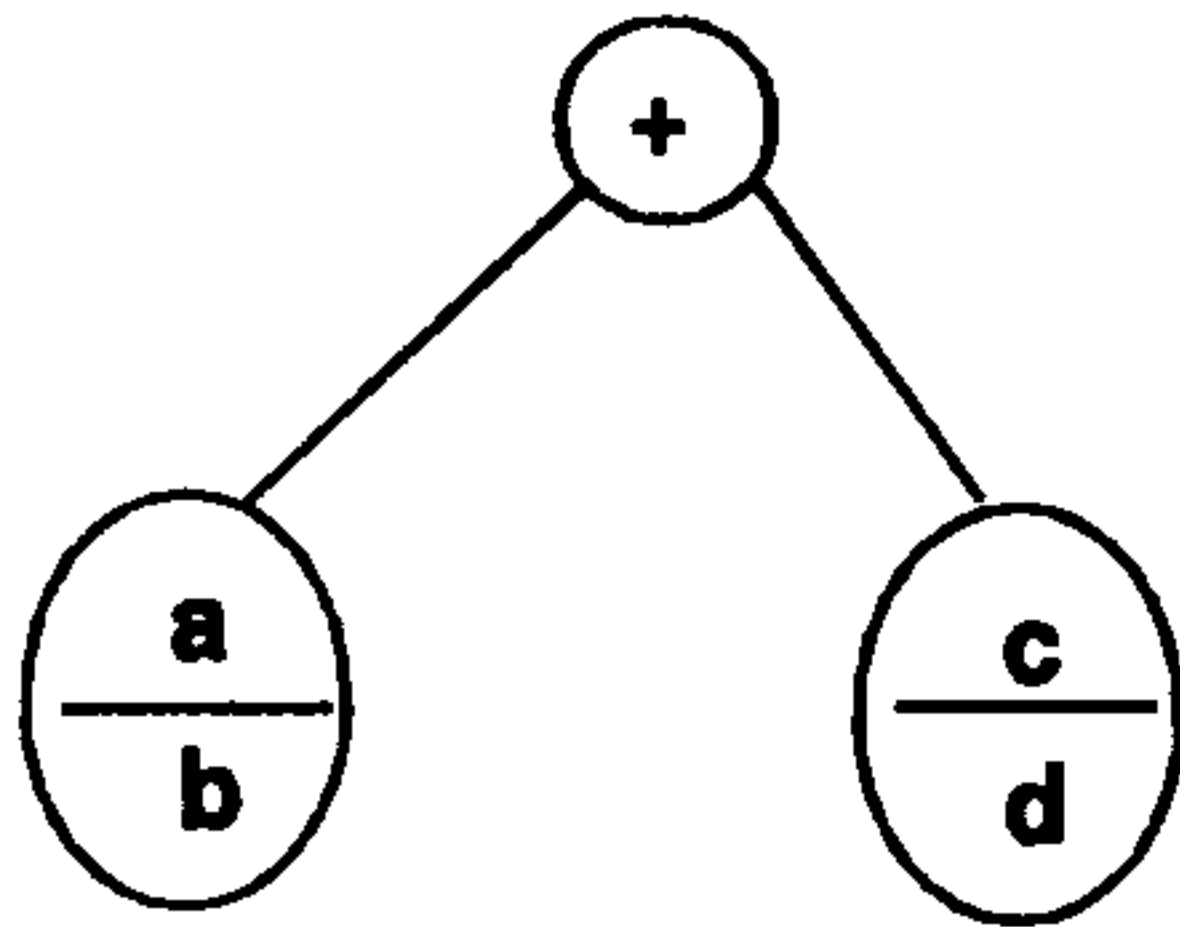
여기에서는 PAT가 사용하는 여러 종류의 지식에 대해서 알아본다.

(1) 영역 지식 베이스 (Domain Expert Rulebase)

분수식을 계산하는데 필요한 모든 정규칙들을 모아놓은 데이터베이스이다. 정규칙은 식의 한 부분이 올바르게 전개된 형태를 나타내며, 변환에 필요한 조건과 변환이 나타내는 의미, 그리고 그 변환에 기본적으로 필요한 연산 개념으로 이루어진다. 변환되기 전의 수식의 일부분과 그것의 변환된 부분이 특정 정규칙의 조건을 만족하면, 학생은 그 정규칙이 내포하는 의미대로 식을 전개했다고 생각할 수 있으며, 학생은 그 규칙이 포함한 연산 개념을 이해하고 있다고 판단할 수 있다. 보통 학생이 하나의 연산자에 대해서 연산을 할 경우에, 그 연산자에 대해서 나타날 수 있는 변환된 형태는 충분히 예측 가능하며, 규칙으로서 표현이 가능하다. 그러나 두개 이상의 연산자를 한꺼번에 계산한 경우에는, 다시 말해서 학생이 여러 단계의 중간 과정을 생략했을 경우에는 그에 해당하는 규칙을 찾아낸다는 것이 거의 불가능하다. 그 과정에 해당하는 규칙은 각 연산자에 대해 가능한 모든 규칙의 조합으로 이루어지기 때문이다. 이 경우에 시스템은 단순히 "A를 B로 계산

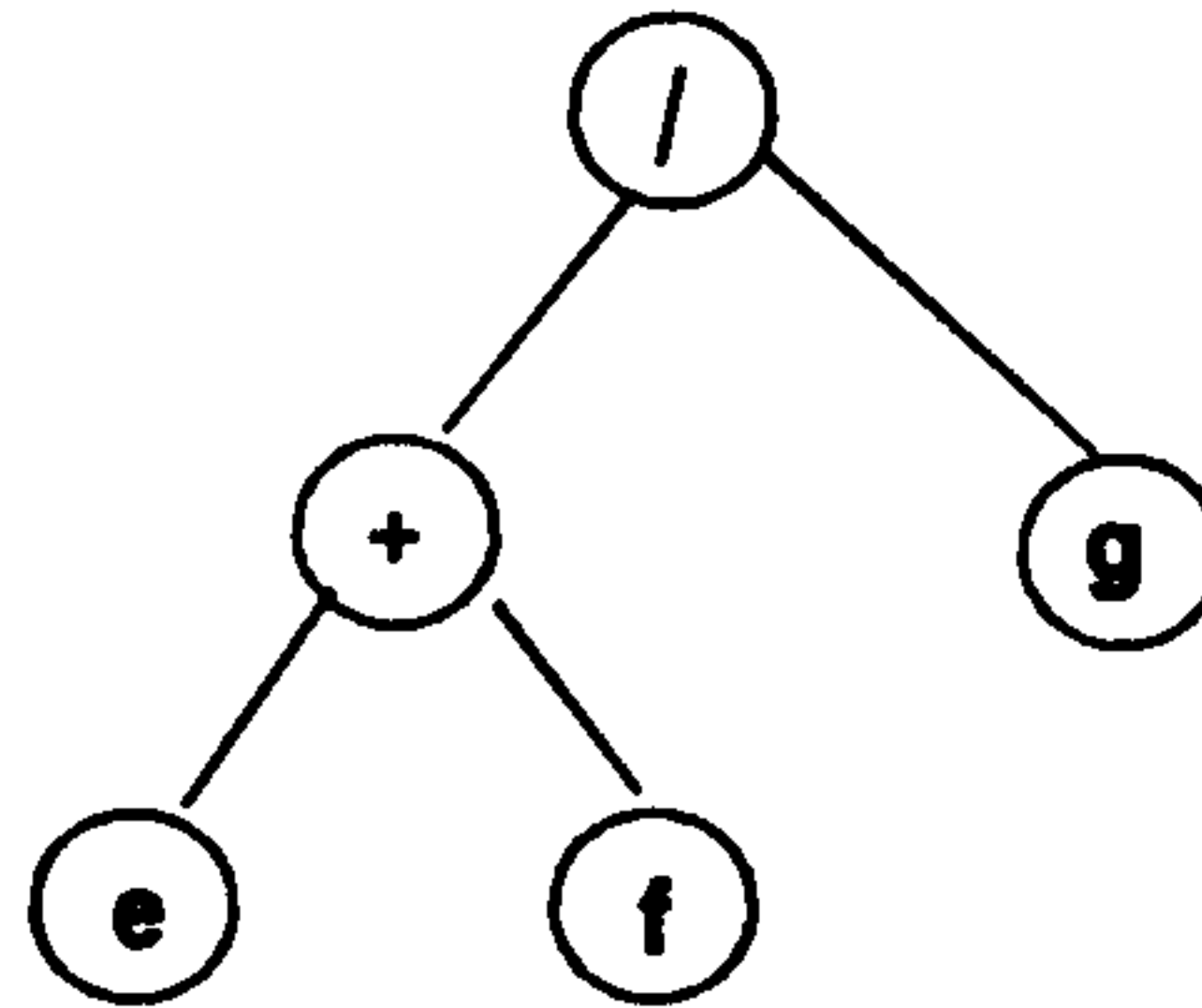
Previous Subexpression

$$a/b + c/d$$



Current Subexpression

$$(e+f) / g$$



Condition: $a = e, c = f, b = d = g$

Meaning: added two proper fractions having same denoms

Concept: common fraction addition

그림 2.20: 공통 분모 분수의 가산에 관한 정규칙의 한 예

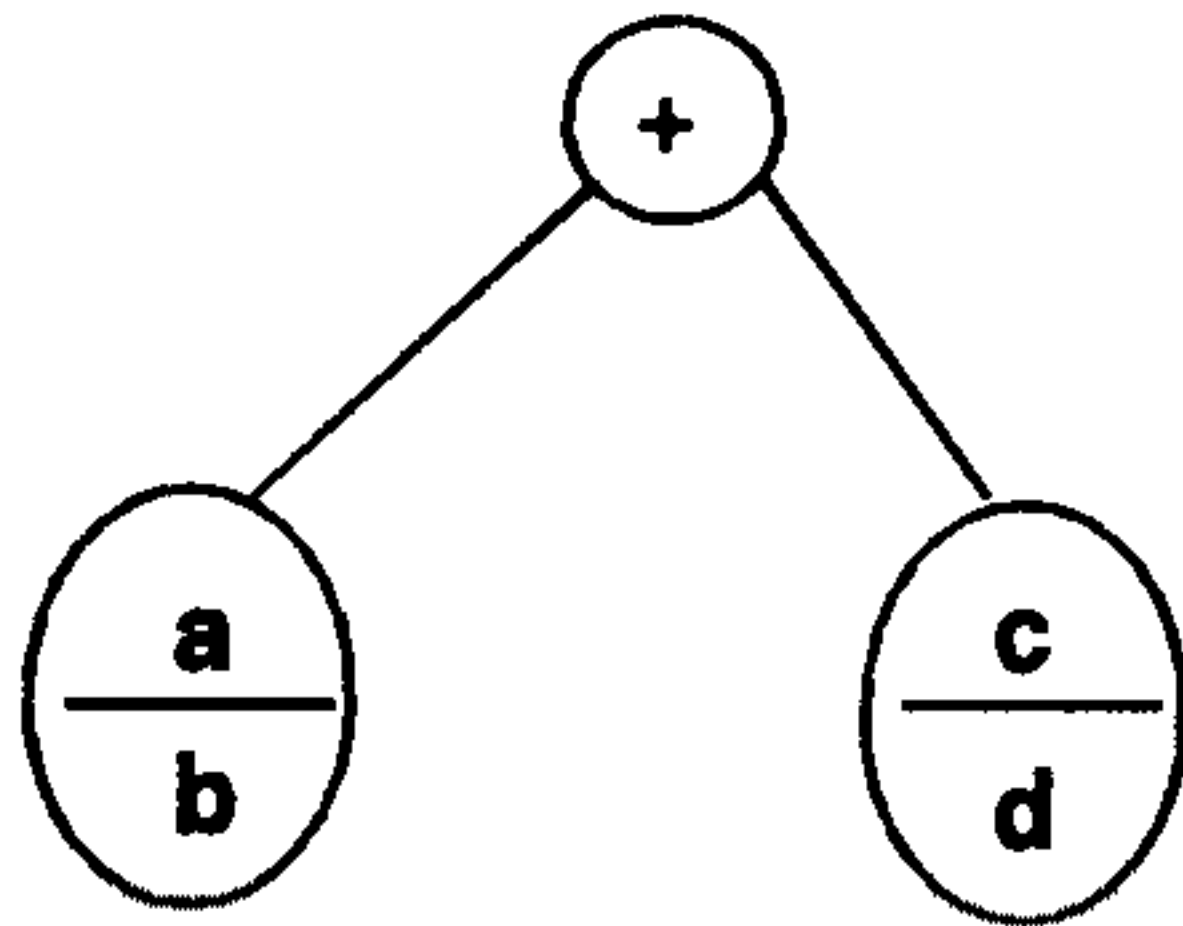
하였다” 정도로만 의미를 부여하게 된다. 뒤에서 설명할 모범 학생 모델을 이용하여 자세한 계산 경위를 찾아내고 그 과정에 사용된 연산 개념들을 추측하게 된다. 그림 2.20는 공통 분모 분수의 가산에 관한 하나의 정규칙을 보여준다.

(2) 오류 목록 (Bug Catalogue)

오류 목록은 분수식을 계산하는데 일어날 수 있는 오류 형태를 오류규칙의 형태로 모아놓은 데이터베이스이다. 오류규칙은 충분히 예측가능한 잘못된 수식 전개 형태를 나타내며, 앞서 설명한 정규칙과 마찬가지로 변환에 필요한 조건들과 변환이 나타내는 의미, 그리고 그 변환과 관련된 연산 개념으로 이루어져 있다. 변환의 의미는 학생이 어떠한 사항을 어떠한 형태로 잘못 이해하고 있음을 나타내며, 연산개념은 그와 관련된 개념을 나타낸다.

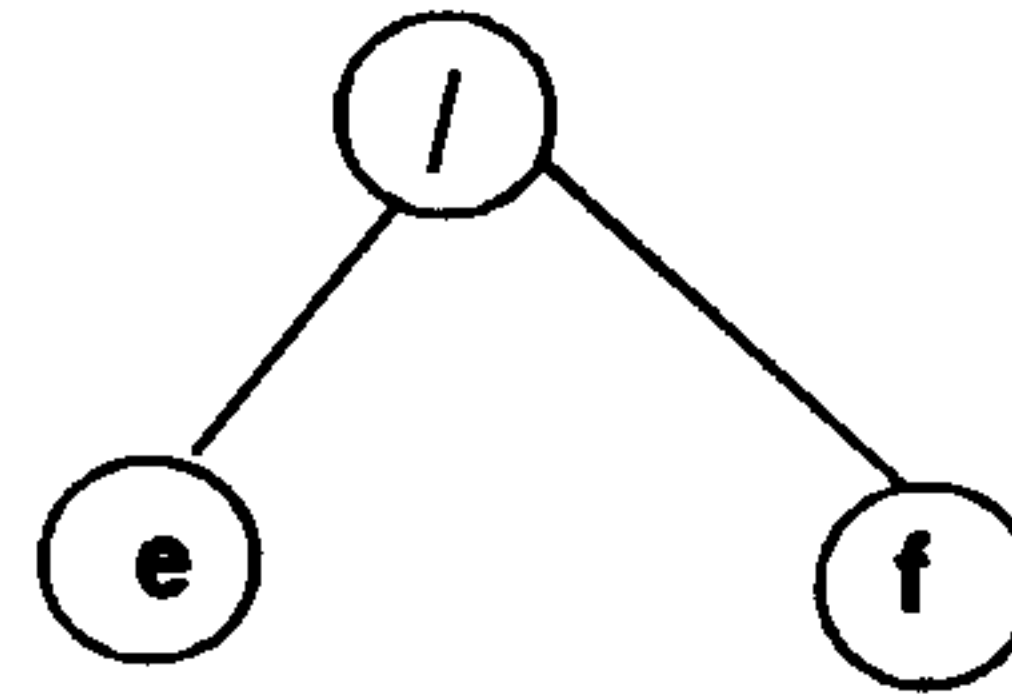
Previous Subexpression

$$a/b + c/d$$



Current Subexpression

$$e/f$$



Condition: $b = d, f = b + d, e = a + c$

Meaning: added both numer. and denom. respectively

Concept: common fraction addition

그림 2.21: 공통 분모 분수의 가산에 관한 오류규칙의 한 예

오류 규칙은 정규칙과는 달리, 학생이 계산한 부분의 연산자가 하나인 경우에도 잘못된 전개된 형태를 모두 나열한다는 것이 불가능하다. 즉, 학생의 오류는 예측 가능한 것뿐만 아니라 예측할 수 없는 경우도 있다. 예측 불가능한 경우에 PAT는 단지 학생이 그 연산을 하는데 어려움을 느끼고 있다고만 판단한다. 학생이 연산자가 둘 이상으로 이루어진 수식을 잘못 계산한 경우에는 PAT가 학생이 어느 부분을 어떻게 잘못 계산했는지 알아내는 것은 거의 불가능하다. 그림 2.21은 공통 분모의 두 분수를 더하는 경우에 있어서, 가능한 하나의 오류 규칙을 보여준다.

(3) 모범 학생 모델 (Ideal Student Model)

모범 학생 모델은 임의의 문제에 대한 모든 올바른 풀이 과정을 나열해 놓은 것이다. 각 단계의 수식은 전 단계의 수식에 대하여 어느 부분을 어떻게 계산했는가와 그 계산에 사용된 개념을 포함한다. 한 문제에 대한 모든 문제풀이 과정은 국민학교 산수 교과서에 소개하는 풀이 방법에 의하여 다음의 알고리즘으로 PAT가 자동적으로 생성한다. 분

수식을 계산할 때 임의의 중간 계산 단계에서 가능한 연산이 둘 이상 있을 수 있으므로 전체 식을 계산하는 과정은 여러 개의 방법이 존재할 수 있다.

1. 분수식에 대분수가 있으면 그것을 가분수로 고친다.
2. 분수식에서 가능한 모든 연산을 구해서 그것의 우선순위에 따라 순서를 정한다. 괄호를 포함한 모든 연산자의 우선순위는 다음과 같다. 여기서 괄호가 두개 이상인 경우는 생략하였다.

연산자	우선순위
(×), (÷)	4
(+), (-)	3
×, ÷	2
+, -	1

3. 모든 가능한 연산에 대해서 우선순위가 높은 연산부터 계산을 한다.
 - (a) 연산자와 피연산자를 고려하여 그에 맞는 전개 법칙을 적용한다. 전개 법칙은 여러단계의 수식을 생성할 수도 있다. 예를 들어, 두 분수의 가산은 다음과 같이 4단계의 중간 수식을 생성한다.

$$\begin{aligned}
 \frac{b}{a} + \frac{d}{c} \quad (a \neq c) &= \frac{b \times e}{a \times e} + \frac{d \times f}{c \times f} \quad (e = LCM(a, c)/a, f = LCM(a, c)/c) \\
 &= \frac{h}{g} + \frac{i}{g} \quad (h = b \times e, i = d \times f) \\
 &= \frac{h+i}{g} \\
 &= \frac{j}{g} \quad (j = h + i)
 \end{aligned}$$

전개 법칙은 항상 두개의 피연산자와 하나의 연산자를 몇 단계의 중간 과정을 통해서 하나의 피연산자로 만들게 된다.

(b) 새로 만들어진 수식이 더이상 연산자를 포함하지 않을 때까지 (a)의 결과를 2에 적용한다.

4. 답이 가분수이면 대분수로 고친다.

이러한 모범 학생 모델은 학생이 여러 중간 단계를 생략하고 답을 올바르게 계산한 경우에, 그 중간에 생략된 단계를 알아내는데에 이용된다. 즉, PAT는 학생이 생략된 단계에 해당하는 모든 개념을 잘 이해하고 있다고 모델링 할 수 있다. 그밖에 모범 학생 모델은 학생이 현 단계에서 어려움을 느끼고 있다고 판단될때, 도움을 주는 기능으로서도 이용된다. 즉, 다음 단계의 수식을 보여주거나 그 계산의 내용을 설명해 줌으로써 학생에게 도움을 줄 수 있다. 다음 표 2.2는 식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 에 대해서 PAT가 생성한 모범 학생 모델을 보여준다. 표에서 보듯이, 식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 을 푸는 방법은 두가지 이상이 존재함을 알 수 있다.

4 수식의 분석

문제 해결 모듈에서는 학생이 제시한 수식을 분석하여 학생의 행동과 의도를 파악한다. 문제의 제시 형태가 박스 형태의 경우에는 학생의 답이 시스템이 원하는 답인가만을 조사하면된다. 그리고 그 문제가 포함하는 개념에 대해서, 그 답의 옳고 그름에 따라 학생 모델링을 한다. 그러나 문제 제시 형태가 수식 전개 형태인 경우에는 보다 복잡한 절차가 필요하다. 이 경우에는 우선 학생이 제시한 수식과 전 단계의 수식을 비교하여, 수식의 변화된 부분을 찾아내어 학생의 행동을 알아내고, 그 변화된 부분을 규칙 또는 오류 규칙에 적용함으로써 학생의 의도를 파악한다. 학생의 행동은 두 수식을 일반 트리로 표현하여 그 트리를 매칭시킴으로써 알아 낼 수 있다. 트리의 매칭은 다음에 설명되는 행동 찾기 알고리즘(action-finding algorithm)을 이용한다. 행동 찾기 알고리즘은 다음

단계	수식	연산 개념
0	$\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$	-
1	$\frac{5}{2} + \frac{2}{12} - \frac{1}{3}$	분수의 승산
2	$\frac{5}{2} + \frac{1}{6} - \frac{1}{3}$	분수의 약분
3	$\frac{53}{23} + \frac{1}{6} - \frac{1}{3}$	통분
4	$\frac{15}{6} + \frac{1}{6} - \frac{1}{3}$	자연수 승산
5	$\frac{15+1}{6} - \frac{1}{3}$	공통 분모 분수의 가산
6	$\frac{16}{6} - \frac{1}{3}$	자연수 가산
7	$\frac{8}{3} - \frac{1}{3}$	약분
8	$\frac{81}{3}$	공통 분모 분수의 감산
9	$\frac{7}{3}$	자연수 감산
10	$2\frac{1}{3}$	대분수를 가분수로 고치기
3	$\frac{5}{2} + \frac{1}{6} - \frac{12}{32}$	통분
4	$\frac{5}{2} + \frac{1}{6} - \frac{2}{6}$	자연수 승산
5	$\frac{5}{2} + \frac{12}{6}$	공통 분모 분수의 감산
6	$\frac{5}{2} - \frac{1}{6}$	자연수 감산
7	$\frac{53}{23} - \frac{1}{6}$	통분
8	$\frac{15}{6} - \frac{1}{6}$	자연수 승산
9	$\frac{151}{6}$	공통 분모 분수의 감산
10	$\frac{14}{6}$	자연수 감산
11	$\frac{7}{3}$	약분
12	$2\frac{1}{3}$	대분수를 가분수로 고치기

표 2.2: 식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 에 대한 모범 학생 모델

의 두 휴리스틱을 기반으로 한다.

- 트리에서 루트노드의 레벨을 0으로 정의할 때, 전 단계의 수식은 다음단계에서 레벨 1의 두 노드가 하나로 합쳐지는 형태로 계산된다. 즉, 학생은 적어도 하나의 연산자에 대하여 계산을 하게되고, 그 결과 두 피연산자가 하나의 서브트리로 묶이는 형태가 된다. 따라서, 전단계의 수식에서 레벨 1에 있는 노드의 갯수는 학생이 문제를 풀어감에 따라 감소된다. 그러나, 레벨 2 이상의 노드들은 경우에 따라 줄 수도 있고 늘어날 수도 있다.
- 수식에 있어 각 수들은 특별히 의도된 경우가 아닌 한, 상대적인 위치 관계가 유지된다. 예를 들어, 수식 왼쪽에 있었던 수는 다음 단계에도 왼쪽에 있을 확률이 크다.

알고리즘의 입력은 전단계 수식과 현단계 수식의 두 일반 트리이며, 출력은 수식이 변환된 부분들의 리스트가 된다. 편의상 전단계의 수식의 트리를 A, 현단계 수식의 트리를 B로 하자.

1. 두 수식 트리간의 노드들을 서로 연결한다.

- (a) 트리 B의 레벨 1에 있는 각 노드에 대해서, 트리 A의 레벨 1에 있는 노드 중에서 값(value)과 선행 연산자(preceding operator)가 같은 노드를 연결한다. 이때, 각 노드의 상대적 위치가 서로 비슷한 것을 먼저 시도한다.
- (b) 레벨 1의 노드 중에서 1a에서 연결되지 않은 나머지 노드끼리 서로 연결한다.
- (c) 1a에서 서로 연결된 두 노드에 대해서 두 노드가 모두 비종단 노드일 경우에, 즉, 하나 이상의 자식 노드를 가질 경우에 그 비종단 노드가 이루는 서브트리에 대하여 1의 알고리즘을 재귀적으로 적용시킨다.

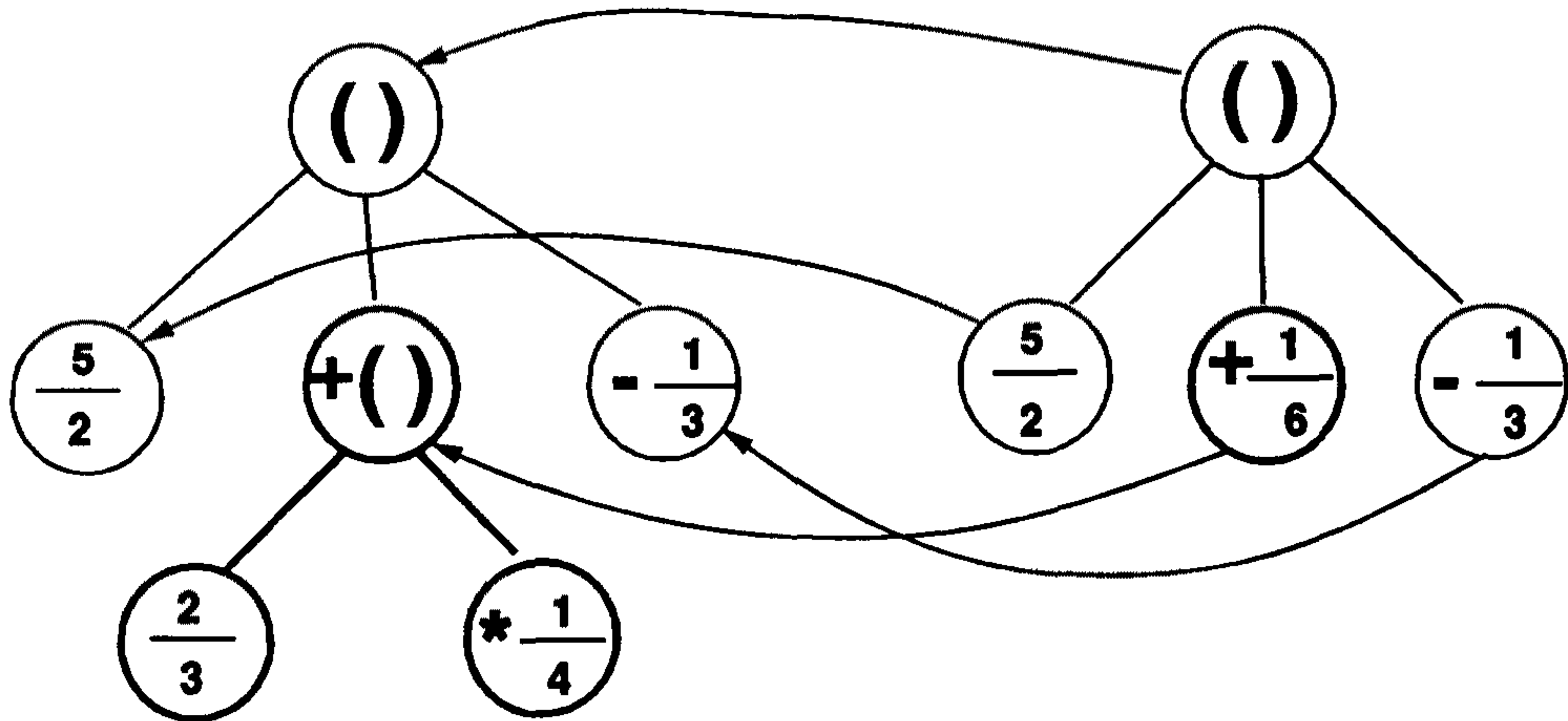


그림 2.22: 전단계의 식과 현단계의 식을 행동 찾기 알고리즘에 적용한 결과

2. 서로 연결된 두 노드중에서 변화된 부분만을 찾아 출력한다.

- B의 종단 노드와 A의 비종단노드가 연결된 경우, 또는 A의 종단 노드와 B의 비종단노드가 연결된 경우에는 당연히 변화가 있는 부분이므로 출력한다.
- B의 종단노드와 A의 종단노드가 연결된 경우에는 두 노드의 값, 선행 연산자, 기호 중에서 최소한 하나 이상 변화가 생긴 경우에 출력한다.
- B의 비종단노드와 A의 비종단노드가 연결된 경우에 있어서, B에 해당하는 노드의 모든 자식 노드가 A의 임의의 노드에 연결된 경우에는 각 자식 노드의 서브트리에 대해서 2의 알고리즘을 재귀적으로 적용시킨다. 그렇지 않은 경우에는 그 연결된 두 노드의 서브트리를 출력한다.

그림 2.22은 이 알고리즘이 사용된 한 예를 보여 준다. 그림에서 전단계의 식 $\frac{5}{2} + \frac{2}{3} \times \frac{1}{4} - \frac{1}{3}$ 과 현단계의 식 $\frac{5}{2} + \frac{1}{6} - \frac{1}{3}$ 을 행동 찾기 알고리즘 1에 적용한 결과를 볼 수 있다. 그림에서 알고리즘은 식 $\frac{2}{3} \times \frac{1}{4}$ 이 식 $\frac{1}{6}$ 로 변화된 것을 찾는다. 행동 찾기 알고리즘은 학생이 전 단계의 수식에서 하나의 연산자만을 계산했을 경우에는 정확하게 찾아낼 수 있지만, 두개 이상의 연산자를 동시에 계산하여 각각 다른 부분으로 변화시킨 경우에는 정확하게 찾지 못하는 경우가 있을 수 있다. 예를 들어, $1 + 2 + 3 + 4$ 의 수식에 대

해서, 학생이 $3 + 7$ 이라고 입력했을 때, 이 알고리즘은 3을 전 수식의 세번째 수인 3과 연결하고 7을 $1 + 2 + 4$ 로 연결하게 된다. 그러나 상식적으로는 3을 $1 + 2$ 로 연결하고 7을 $3 + 4$ 로 연결하는 것이 더 바람직할 것이다.

행동 찾기 알고리즘에 의하여 전 단계의 수식과 현 단계의 수식의 변화를 찾아낸 후에, 그 변화된 부분을 정규칙 또는 오류규칙에 적용시킴으로써 학생의 의도를 파악한다. 서로 연결된 두 서브트리의 값과 선행 연산자가 모두 같은 경우에는 학생이 옳게 식을 전개한 경우이므로 정규칙에 적용시키고, 그렇지 않은 경우에는 오류규칙을 적용시킨다. 각 서브트리를 정규칙 (또는 오류규칙)에 적용시켜서 학생의 의도를 파악하며, 각 정규칙(또는 오류규칙)이 포함한 연산 개념에 따라 학생의 지식 상태를 모델링한다. 필요한 경우에는 학생에게 메시지를 내준다.

5 학습자 모델링 (Student Modelling)

학습자 모델링은 학생의 지식 상태를 동적으로 유지 관리하는 부분이다. 문제 해결 모듈에서 학생이 제시한 수식을 분석한 결과와 연산개념의 계층적 구조를 이용하여, 학생이 문제를 풀어가는데 사용된 연산개념에 해당하는 학생의 지식점수(knowledge score)를 변경시킨다. 학생의 지식은 앞서 연산개념에서 설명한 바와 같이 21개의 개념으로 이루어진다. 학습자 모델링은 이 21개의 개념을 0과 1사이의 실수로 점수를 매겨가면서 학생의 지식 상태를 표현한다. 표 2.3에 나타난 바와 같이 한 학생에 대해서 각 개념의 점수는 3가지 레벨의 숙달 정도로 다시 표현된다. 한 개념의 점수가 0.9 이상이면 학생은 충분히 그 개념을 이해하고 있다고 판단하며, 시스템은 학생이 그 개념에 대해서 전문가라고 간주한다. 이 경우에 학생은 다음 개념의 문제로 넘어갈 수 있다. 점수가 0.5에서 0.9이하인 경우에 학생은 그 개념의 문제에 대하여 좀더 공부할 필요가 있다. 따라서 시스템은 그 개념에 해당하는 문제를 계속해서 내준다. 점수가 0.5 이하인 경우에,

레벨	점수	PAT가 취하는 행동
전문가	0.9 이상	다음 단계의 개념으로 넘어감
보통	0.5 ~ 0.9	현단계의 문제를 계속 내줌
초보자	0.5 이하	하부개념에 대한 학생의 지식 상태 재점검

표 2.3: 특정 개념에 대한 학생의 숙달 정도

학생은 그 개념에 대하여 어려움을 느끼는 초보자로 판단하고, 보다 자세한 설명을 하거나, 그 개념의 기본이 되는 하부개념의 문제를 내줌으로써 특별한 조치를 하게 된다.

한 학생에 대한 각 지식 단위의 점수의 변경은 문제 해결 모듈에서 학생이 제시한 수식을 분석한 결과를 이용한다. 우선 학생의 행동이 수식을 올바르게 전개한 경우이면, 전개에 사용된 모든 개념에 대해서 그 지식의 점수를 어느 정도 올려 준다. 첫번째로 수식의 변환이 특정 정규칙에 해당하는 경우에는 그 정규칙의 개념과 그것의 하부개념에 대하여 각각 점수를 올려 준다. 학생이 특정 정규칙을 올바르게 사용했음은 학생이 그 정규칙이 포함하는 연산개념을 이해하고 있음을 나타내고 있는 것이고, 또한 한 개념을 이해하고 있다는 것은 그 개념의 모든 하부개념을 이해하고 있음을 의미하기 때문이다. 두번째로, PAT가 학생의 수식 전개에 해당하는 정규칙을 찾을 수 없을 경우에는 학생이 몇단계의 중간 계산 과정을 생략하고 올바른 답을 낸 경우이다. 이 경우에는 모범 학생 모델을 사용한다. 이때 시스템은 모범 학생 모델로서 전 단계의 수식에 해당하는 모든 올바른 수식의 전개 과정을 생성한다. 시스템은 모범 학생 모델이 생성한 수식들 중에서 학생이 제시한 수식과 가장 가까운 수식을 찾아 낸다. 학생은 전 단계의 수식과 찾아진 수식 사이에 존재하는 모든 전개 과정의 수식을 생략했다고 할 수 있다. 따라서 시스템은 그 사이의 모든 수식에 이용된 개념과 그 개념의 하부개념에 대해서 학생의 지식의 점수를 어느 정도 올려 준다. 세번째로, 학생이 현재 주어진 문제를 올바르게 풀어서 시스템이 원하는 답을 제시한 경우에는 그 문제의 개념에 대해서 지식 점수를 올려 준다. PAT에서는 식 2.1을 이용하여 학생의 지식 점수를 변경시킨다. 식에서 α 는 학생이 올바른 답을 낸 경

우 그것이 기존의 점수에 반영되는 정도를 나타내며, PAT의 실제 구현에서는 0.4이다.

$$NewScore = (1 - \alpha) * OldScore + \alpha \quad (2.1)$$

학생의 행동이 수식을 잘못 전개한 경우, 즉 오류가 발생한 경우에는 그 오류에 포함된 연산개념의 지식의 점수를 어느 정도 내린다. 우선, 학생의 오류가 특정 오류규칙에 해당되었을 경우에는 그 오류규칙이 포함하고 있는 연산개념의 지식점수를 변경시킨다. 이때, 그 연산개념의 하부개념에 대해서는 지식점수를 변경하지 않는다. 그것은 학생이 한 개념을 이해하지 못한다는 것만으로는 그 개념의 하부개념중 어느 한 부분을 모른다고 판단할 수 없기 때문이다. 그 대신에 그 개념을 하부개념으로 하는 모든 상부개념에 대해서 지식점수를 깎는다. 하나의 하부개념에 어려움을 느끼면, 그것을 기반으로 하는 상부개념에도 역시 어려움을 느낄 가능성이 있기 때문이다. 두번째로 학생의 오류가 특정 오류규칙에 해당하지 않았을 경우이다. 이 경우는 다시 두가지 경우로 나뉘는데, 우선 학생이 계산한 부분이 하나의 연산자 만을 포함한 경우이다. 이 경우에 시스템은 그 연산자와 피연산자가 이루는 개념과 그것의 상부개념에 대해서 학생의 지식 점수를 변경한다. 즉, 시스템은 학생이 그 피연산자와 연산자로 이루어진 부분제(subproblem)의 개념을 잘 모른다고 판단할 수 있다. 그러나, 학생이 계산한 부분이 두개 이상의 연산자를 포함한 경우에는 시스템이 학생의 지식상태를 모델링하는 것이 매우 어렵다. 예를 들어, 학생이 식 $\frac{1}{2} + \frac{2}{3} \times \frac{1}{5}$ 을 $\frac{4}{25}$ 로 답했다고 하자. 이 경우에 시스템은 학생이 가산을 잘못해서 이와 같은 오류를 범했는지, 아니면 승산을 잘못해서 오류를 범했는지 직접적으로 판단할 수 없다. 따라서 이 경우에는 학생의 지식점수를 변경할 수가 없다. 하지만, 시스템은 이 수식을 포함하고 있는 각 연산자에 대해서 그 연산에 해당하는 문제의 예를 내주고 학생에게 풀도록 함으로써 각 연산자의 개념에 대한 학생의 지식을 세부적으로 파악해 볼 수 있다. 이와 같이 학생이 오류를 범한 경우에는 다음 식 2.2를 이용하여 학생의 지식 점

수를 어느 정도 내린다. 식에서 β 는 0.7 이다.

$$NewScore = \beta * OldScore \quad (2.2)$$

6 교육 전략 (Tutoring Strategy)

6.1 문제의 선택

PAT에서 모든 문제는 단원별로 나누어져 있고, 각 단원은 하나의 개념과 연관되어 있다. 따라서 하나의 문제를 선택하는 것은 학생에게 시험할 하나의 개념을 선택하는 것과 같다. 문제의 선택은 그 시기에 있어서 다음 두가지로 나뉜다. 우선 학생이 현재 제시된 문제를 올바르게 풀고 다음 문제로 넘어갈 때이다. 이 경우에 시스템은 21개의 지식 점수 중에서 전문가 레벨이 안되는 개념들 중에서 가장 하위 개념의 문제를 선택한다. 이것은 학생이 숙달되지 않은 개념중에서 가장 기초적이고 다른 개념에 기본이 되는 개념을 선택함을 의미한다. 따라서 PAT는 항상 학생에게 기본 개념을 확실히 하도록 유도한다.

문제의 선택은 학생이 문제를 풀고 있는 동안에 이루어질 수도 있다. 학생이 그 문제를 풀고 있는 동안에 계속 반복되는 오류로 인하여 특정 하부개념의 지식점수가 계속 내려가서 그 점수가 초보자 레벨로 바뀔 수 있다. 그 경우에 시스템은 현재의 문제를 포기하고 그 하부개념의 문제로 전환하여 내준다. 결국 현재 학생의 단계는 그 하부개념의 단계로 전이된다. 학생이 그 하부개념을 충분히 반복 학습한 후에 그것의 지식점수가 전문가 레벨로 높아지면, 시스템은 자동적으로 원래의 문제로 복귀한다.

6.2 도움 기능

학생이 문제를 풀고 있는 동안에 어려움을 느낀다고 판단될때 시스템은 자동적으로 학생의 지식 상태를 이용하여 도움말을 생성할 수 있다. 현재 문제의 개념에 대해서 그 개념의 하부 개념은 만족할 정도의 지식 점수를 가지고 있음에도 불구하고, 그 개념의 점수가 현저히 낮을 경우에는 학생이 하부개념은 잘 알고 있지만 그 하부개념을 이용하는 방법, 즉 기술을 잘 모른다고 판단할 수 있다. 이 경우에 시스템은 그 기술에 대한 상세한 설명을 제공해 준다.

학생이 맞는 답을 낸 경우라도 그것이 바람직한 방법이 아닐 경우에 시스템은 정규칙과 모범 학생 모델을 이용하여 그 내용을 지적하고 보다 바람직한 방법에 대해 설명해 줄 수 있다. 예를 들어, 학생이 두 분수의 가산에서 통분을 할 경우에 최소 공배수로 통분하지 않은 경우를 생각해 보자. 식 $\frac{1}{4} + \frac{1}{6}$ 이 식 $\frac{6+4}{24}$ 로 변환된 경우와 같이 두 분수가 다음 단계에서 하나의 서브트리로 결합된 경우에는 정규칙으로 충분히 찾아낼 수 있다. 그러나 식이 $\frac{6}{24} + \frac{4}{24}$ 와 같이 변환된 경우에는 변환된 부분이 두개로 나뉘어지기 때문에 정규칙으로는 파악이 불가능하다. 이때에는 모범 학생 모델을 이용한다. 모범 학생 모델이 생성한 중간 과정에는 식 $\frac{3}{12} + \frac{2}{12}$ 가 존재하기 때문에 식 $\frac{6}{24} + \frac{4}{24}$ 가 이 식에 매칭됨을 알수 있고, 따라서 학생이 최소공배수로 통분하지 않았음을 찾아낼 수 있다.

모범 학생 모델은 학생에게 바람직한 계산 과정을 추천해 주는 도움 기능으로서도 사용될 수 있다. 모범 학생 모델에 의해서 현 단계에서 취할 수 있는 가장 바람직한 연산을 먼저 하도록 설명해 줄 수 있고, 학생에게 도움말과 함께 계산 과정을 단계별로 차례로 보여줌으로써 문제를 해결해 나가는 방법을 설명해 줄 수 있다.

제 4 절 평가 및 분석

이 장에서는 PAT 산수 교사 시스템을 평가하고 분석한다. 불행하게도 이제까지 ITS의 성능을 객관적으로 평가하는 방법이 표준화되지 못했으며 또한 어려움이 많다. 그 이유는 ITS 자체가 매우 다양한 요소를 포함하고 있어 이를 종합하여 평가하기가 힘들기 때문이다. Littman은 ITS를 평가하는 두가지 방법을 제시했다[Lit88]. 그중 하나는 외부 평가(external evaluation)로서 학생에게 그 시스템을 직접 사용하도록 하여 ITS가 학생에게 얼마나 많은 영향을 주었으며 학생의 지식과 문제 풀이 능력이 얼마나 향상되었는가를 조사하는 것이다. 이와같은 외부 평가는 객관적인 실험 뿐만 아니라 실험하는 동안 주변의 환경과 학생의 심리 상태에도 영향을 받기 때문에 치밀한 계획하에 이루어져야 한다. ITS를 평가하는 또하나의 방법은 내부 평가 (internal evaluation)로서 ITS 내부의 구조와 각 요소들의 성능을 여러 가지 관점에서 평가하는 것이다. 즉 ITS 설계시에 유념해야 할 중요한 요소들에 대하여 객관적인 평가를 한다.

현재 구현된 PAT에서는 펜으로 글씨를 쓰는 타블렛과 그 흔적을 보여주는 스크린이 서로 분리되어 있어 학생이 수식을 올바르게 입력하기에 많은 어려움이 있으며 따라서 시스템 사용전에 약간의 연습과정이 필요하다. 그러므로 펜의 장점인 자연스럽고 쓰기 쉽다는 특징을 충분히 살릴 수 없다는 문제점이 있다. 따라서 학생을 대상으로 하는 외부 평가는 완전한 시스템의 모습을 갖출 때까지 차후로 미루기로 한다. 여기에서는 PAT를 설계하는데 사용된 방법들을 나열하고 그것들의 장점과 문제점을 분석하며, PAT가 아직 갖추지 못한 부족한 점들을 앞으로의 연구를 위해서 지적해 보기로 한다.

PAT에서 도입한 학습 방법은 학생이 어떠한 제약없이 생각한 것을 직접 펜으로 쓰는 것이며, 시스템은 가능한 한 학생의 행동에 간섭을 하지 않는 것이다. 이와같은 교육 방식은 학습 성취도 면에서도 가장 바람직한 방법으로 알려져 있고, 시스템의 입장에서도 학생이 어느 부분을 알고 있고 어느 부분을 잘 모르는지를 판단하는데 가장 좋은 방법이다.

PAT는 크게 두가지 사항에 의해서 학생을 모델링하며 그것에 의해서 학생에게 주어질 지식의 정도가 정해진다. 우선 PAT는 학생의 입력에 아무런 제약도 가하지 않기 때문에 학생의 행동을 정확하게 찾아 내는 것이 필요하다. 현재 구현된 시스템에서는 학생이 수식에 대해서 하나의 연산만을 했을 경우에는 학생의 행동을 모두 찾아낼 수 있다. 그러나 학생이 두개 이상의 연산을 동시에 수행하였을 경우에는 학생의 행동을 못찾을 가능성이 있다. 그 경우에는 학생의 행동에 대한 정확한 지도가 불가능하다. 이러한 문제를 해결하기 위해서는 학생이 문제를 풀어나가는 습관과 방식을 조사하고 분석하여 이를 행동 찾기 알고리즘에 반영하여 이를 개선시킬 필요가 있다.

또다른 사항은 학생이 취한 행동을 기존의 지식에 반영하여 학생의 의도를 유추하는 것이다. 이것은 학생이 틀린 답을 냈을 경우에 학생의 잘못된 지식을 바로 잡아주기 위하여 반드시 필요한 사항이다. 현재 구현된 시스템에서는 오류 모델을 이용하여 학생의 오류를 파악한다. 오류 모델은 충분히 예측 가능한 경우에 대해서 그것을 하나의 지식으로 코드화한 것을 말한다. 따라서 학생이 정형화되지 않은 패턴의 오류를 범한 경우에는 실질적으로 학생의 잘못을 바로 잡아 줄 수가 없다. 또한 학생이 여러 단계의 중간 과정을 생략하고 오류를 범했을 경우에도 시스템은 학생이 수식의 어느 부분을 잘 모르는지 파악하기 힘들다. 이 경우에는 학생이 자주 범하는 오류 패턴과 그 빈도수를 이용한 통계적인 방법으로 학생의 오류를 예측할 수 있을 것이다. 오류규칙에 의한 학생의 오류 파악은 기본적으로 그 패턴을 일일이 열거하기 어렵다는 단점이 있다. 따라서 그 오류 패턴을 보다 정확히 얻기 위해서는 종이와 연필을 사용하는 예비 실험을 하여 학생들이 자주 범하는 오류를 조사하고 그것을 분석하는 단계가 필요하다.

모범 학생 모델은 학생 모델링과 학생에게 도움을 주는 기능에 효율적으로 사용된다. 이것은 학생이 여러 단계의 중간 과정을 생략하고 올바른 답을 제시한 경우에 그 중간 과정을 추측하는데에 이용될 수 있고, 그 결과로 시스템이 보다 정확한 학생 모델링을 할 수 있게 된다. 또, 현단계에서 학생이 해야 할 사항을 힌트로 제공해 주거나, 올바른 문제 풀이 과정을 단계별로 설명과 함께 제공할 수 있다. 결국 PAT에서 모범 학생 모델은 시스

템이 분수식에 관한 문제 해결 능력을 가지고 있음을 나타내는 것이라고 생각할 수 있다.

PAT에서 분수식과 관련된 여러 개념을 계층적으로 구성하였으며, 각각의 지식은 그것의 기본이 되는 지식으로 여러개의 하부 개념으로 이루어 진다. 이와같은 지식의 계층적 구조는 학생 모델링을 보다 효율적으로 할 수 있으며, 그 구조를 이용하여 학생의 지식 정도를 참고하여 문제를 선택하는데 좋은 도구로서 이용될 수 있었다. 그러나, 지식을 단순한 격자(lattice)의 형태로 표현하는데는 문제점이 있음을 발견할 수 있다. 예를 들어 통분을 하는 경우에 있어서 PAT에서는 하부 개념으로 최소공배수를 구하는 것과 분수 변환을 반드시 필요한 요소로 규정하였는데, 실제로 학생은 최소공배수 구하는 방법을 모르고 있더라도 다른 공배수로 통분을 하고 나중에 약분을 할 수 있다. 따라서 지식을 계층적으로 구성하는 방법에 있어서 여러가지 선택적인 방법이 존재함을 반영하여야 한다.

PAT에서는 학생의 지식정도에 상관없이 학생에게 하나의 개념을 차등을 두지 않고 설명한다. 그러나 ITS에서 보다 효율적으로 학생에게 설명을 해주기 위해서는 학생의 지식 정도를 반영하여 상세한 정도에 차등을 두어 설명하는 것이 필요하다. 즉, 어느 한 개념을 거의 모르는 학생에게는 자세한 설명을 해주고 이미 어느 정도 알고 있다고 판단 되는 학생에게는 간단한 설명을 해주며, 학생이 더욱 자세한 설명이 필요하다고 요구하면 그때 보여주는 것이다. 이것은 학생들이 지루함을 느끼지 않도록 하는데 중요한 요소가 될 수 있다.

제 5 절 결 론

본 논문에서는 펜 장치를 사용자 인터페이스의 기본 입력장치로 사용하는 산수 교사 시스템을 설계하고 구현하였다. 입력 수단으로서 펜을 사용하는 것은 학생에게 편하고 자연스러운 입력 방법을 제공해 줄 뿐만 아니라 키보드를 기반으로 한 시스템에서는 이용이 불가능했던 계산의 중간 과정까지도 학생 모델링에 이용할 수 있다는 장점을 가질 수 있었다. 본 논문에서는 PAT에 알맞은 문자 인식 시스템을 자체적으로 개발하였으며 인식된 결과를 학습 시스템이 이용할 수 있는 형태로 바꾸어주는 알고리즘을 개발하였다.

PAT에서 제안하는 학습 방법은 가능한 한 학생이 문제를 풀어가는데 간섭을 줄이고 어떠한 제약도 가하지 않음으로써 학생이 자기가 생각한대로 수식을 입력하도록 하는 것이었다. 이와같은 방식은 학생의 지식 상태를 보다 정확히 파악하는데 도움을 주었다. 제한되지 않은 학생의 입력을 해석하기 위해서 시스템은 전단계의 수식과 비교하여 수식의 변화된 부분을 찾았으며, 이때 일반 트리 구조가 효율적으로 사용되었다. 수식의 변화된 부분은 정규칙 또는 오류 규칙에 적용하여 학생의 의도를 파악하였으며, 오류가 발생한 경우에는 해당하는 오류 규칙으로서 학생의 잘못된 지식을 찾아낼 수 있었다.

PAT에서는 분수식과 관련된 지식을 연산 개념의 형태로 표현하고 이를 계층적으로 분류하여 학생 모델링과 교육 전략에 이용하였다. 개념의 계층적 구조는 지식을 추상화하기 때문에 학생 모델링을 보다 효율적으로 하는데 도움을 주었으며, 교육 전략 모듈은 그 개념들의 지식 점수를 참고하여 학생의 취약한 부분을 집중적으로 교육할 수 있었다.

PAT는 자체적으로 문제를 해결하는 능력을 갖기 위해서, 국민학교 교과서에 소개되는 문제풀이 방법을 알고리즘화 하였으며, 이것을 이용하여 자동적으로 모범 학생 모델을 생성할 수 있었다. 모범 학생 모델은 학생이 중간 계산 과정을 생략한 경우에 그 생략된 부분을 찾아내는데 이용될 수 있었고 학생에게 도움을 주는 기능으로서도 응용될 수 있었다. 또한 학생이 바람직 하지 않은 방법으로 문제를 풀었을 경우에 그것을 바로 잡아

주는 역할을 할 수 있었다.

PAT의 사용자 인터페이스는 그래픽을 기반으로하는 윈도우 시스템을 이용하여 구현되었으며, 메시지를 텍스트 뿐만 아니라, 그래픽, 애니메이션, 음성등으로 다양하게 내줌으로써 학생이 흥미를 가지고 학습에 임할 수 있도록 유도하였다.

앞으로 더 연구해야할 과제는 다음 두가지로 요약할 수 있다. PAT가 학생의 수식을 이해하고 그에 따른 지도를 할 수 있는 능력은, 행동 찾기 알고리즘에 의한 수식의 변화된 부분을 찾아내는 것과 그 변화된 부분을 규칙에 적용하여 학생의 의도를 파악하는 것에 영향을 받는다. 현재 구현된 PAT에서는 이 두 부분이 학생 입력의 대부분을 처리할 수 있다고 판단되나, 앞에서 지적한 바와 같이 그것의 한계점이 존재하므로 그 부분을 각각 개선시킬 필요가 있다. 또한 PAT에서 구성한 지식의 계층적 구조는 하나의 개념을 이해하기 위해서는 반드시 하부 개념을 이해하여야 한다는 조건을 두고 있어서, 다른 우회적인 해결 방법을 허용하지 않았다. 그러나 보다 다양한 학생의 문제풀이 과정을 처리하기 위해서는 하나의 개념을 여러개의 선택적인 방법으로 해결하는 것도 가능하도록 복합된 지식의 계층적 구조를 설계하여야 한다.

참고 문헌

- [And85] J.R.Anderson, D.F.Boyle and B.J.Reiser, "Intelligent Tutoring System," *Science*, Vol. 288, pp. 456-462, 1985.
- [Bro78] J.S.Brown and R.R.Burton, "Diagnostic Models for Procedural Bugs in Basic Mathematical Skill," *Cognitive Science*, Vol. 2, pp. 155-192, 1978.
- [Bur82] R.Burton, "Diagnosing Bugs in a Simple Procedural Skill," In D.Sleeman and J.S.Brown(eds.), *Intelligent Tutoring System*, Academic Press, pp 157-183, 1982.
- [Cho89] D.Chow and J.Kim, "Paper Like Interface for Educational Application," *Proc. National Educational Computing Conf.*, pp. 337-344, June, 1989.
- [DeG85] C.J.DeGery, J.Drouard, B.Dumount, S.Hocquenghem, D.Lacombe and G.Sol, "Computer Assisted Testing by Questions and Answers," In K.Duncan and D.Harris(eds.), *Computers in Education*, North Holland: Elsevier Science, pp. 311-316, 1985.
- [Fau90] C.Faure and Z.X.Wang, "Automatic Perception of the Structure of Handwritten Mathematical Expressions," In R.Plamondon and C.G.Leedham(eds.), *Computer Processing of Handwriting*, World Scientific Publishing Co., pp. 337-361, 1990.

- [Has81] K.Hasemann, "On Difficulties with Fractions," *Educational Studies in Mathematics*, Vol. 12, pp. 71-87, 1981.
- [Hor87] E.Horowitz and S.Sahni, *Fundamentals of Data Structure in Pascal*, 2ed., Computer Science Press, 1987.
- [Lit88] D.Littman and E.Soloway, "Evaluating ITSs: The Cognitive Science Perspective," In M.C.Polson and J.J.Richardson(eds.), *Foundation of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates Inc. Publishers, pp. 209-242, 1988.
- [Nak89] Y.Nakayama, "Mathematical Formula Editor for CAI," *ACM CHI'89 Proceedings*, pp. 387-392, May 1989.
- [Nwa88] H.S.Nwana and P.Coxhead, "Towards an Intelligent Tutoring System for a Complex Mathematical Domain," *Expert System*, Vol. 5, No. 4, pp. 290-299, 1988.
- [Nwa91] H.S.Nwana, "User Modelling and User Adapted Interaction in an Intelligent Tutoring System," In *User Modelling and User Adapted Interaction*, Vol. 1, No. 1, pp. 1-32, 1991.
- [Ric89] J.W.Rickel, "Intelligent Computer-Aided Instruction: A Survey Organized Around System Components," *IEEE Trans. SMC*, Vol. 19, No. 1,

pp. 40-57, 1989.

- [Sle81] D.H.Sleeman and M.J.Smith, "Modelling Student's Problem Solving," *Artificial Intelligence*, Vol. 16, pp. 171-188, 1981.
- [Sle82a] D.Sleeman, "Assessing Concepts of Competence in Basic Algebra," In D.Sleeman and J.S.Brown(eds.), *Intelligent Tutoring System*, Academic Press, pp. 185-199, 1982.
- [Sle82b] D.Sleeman and J.S.Brown, "Introduction: Intelligent Tutoring Systems", In D.Sleeman and J.S.Brown(eds.), *Intelligent Tutoring System*, Academic Press, pp. 1-9, 1982.
- [Sle85] D.Sleeman, "Basic Algebra Revisited: a Study with 14-year-olds," *International Journal of Man-Machine Studies*, Vol. 22, pp. 127-149, 1985.
- [Sle87] D.Sleeman, "A Shell for Developing Intelligent Tutoring Systems," In R.Lawler and M.Yazdani(eds.), *Artificial Intelligence and Education, Learning and Tutoring System*, Vol. 1, Ablex Publishing, pp. 239-265, 1987.
- [Tap90] C.C.Tappert, C.Y.Suen, T.Wakahara, "The State of the Art in On-Line Handwriting Recognition," *IEEE Tr. on PAMI*, Vol 12. No. 8, pp. 787-808, August 1990.

- [Wol89] C.G.Wolf, J.R.Rhyne and H.A.Ellozy, "The Paper Like Interface," *Proc. Third International Conf. on Human-Computer Interaction*, Sep., 1989.
- [Woo84] B.Woolf and D.D.McDonald, "Building a Computer Tutor: Design Issues," *IEEE Computer*, pp. 61-73, Sep. 1984.
- [Woo86] B.Woolf, D.Blegen, J.Jansen and A.Verlog, "Teaching a Complex Industrial Process," *Proc. of 5th National Conference on Artificial Intelligence (AAAI-86)*, 1986.
- [Yaz86] M.Yazdani, "Intelligent Tutoring System Survey," *Artificial Intelligence Review*, Vol. 1, pp. 43-52, 1986.
- [Yaz87] M.Yazdani, "Intelligent Tutoring Systems: An Overview." In R.W.Lawler and M.Yazdani(eds.), *AI and Education*, Vol. 1, pp. 183-201, 1987.

제 3 장 GUI 기법을 이용한 형식 문서의 구조 표현 방법에 관한 연구

최근에 문서를 분석하고 인식하고자 하는 연구가 활발히 진행되고 있으며 특히 그 구성 요소가 비교적 단순하고 형태가 자주 바뀌지 않는 형식 문서의 인식에 대한 연구가 활발하게 진행되고 있다. 본 연구에서는 이러한 연구들을 실제적인 기법을 중심으로 정리하고 이의 분석을 통하여 얻어진 문제점을 개선하는 방법으로 그래픽 사용자 인터페이스에 기반을 둔 방법을 이용하여 형식 문서의 구조를 표현하는 방법을 제시한다. 또한, 이를 시스템으로 구현하여 실제 문서 데이터의 구조를 효과적으로 표현할 수 있음을 보인다.

제 1 절 형식 문서의 분석 및 인식

문서처리에 관한 연구는 크게 문서의 분석 (document analysis)과 문서의 인식 (document understanding) 분야로 나눌 수 있다. 이러한 관점은 문서의 구조가 기하학적인 구조와 논리적 구조로 나뉠 수 있다는데 기인하는 분류방법이다.

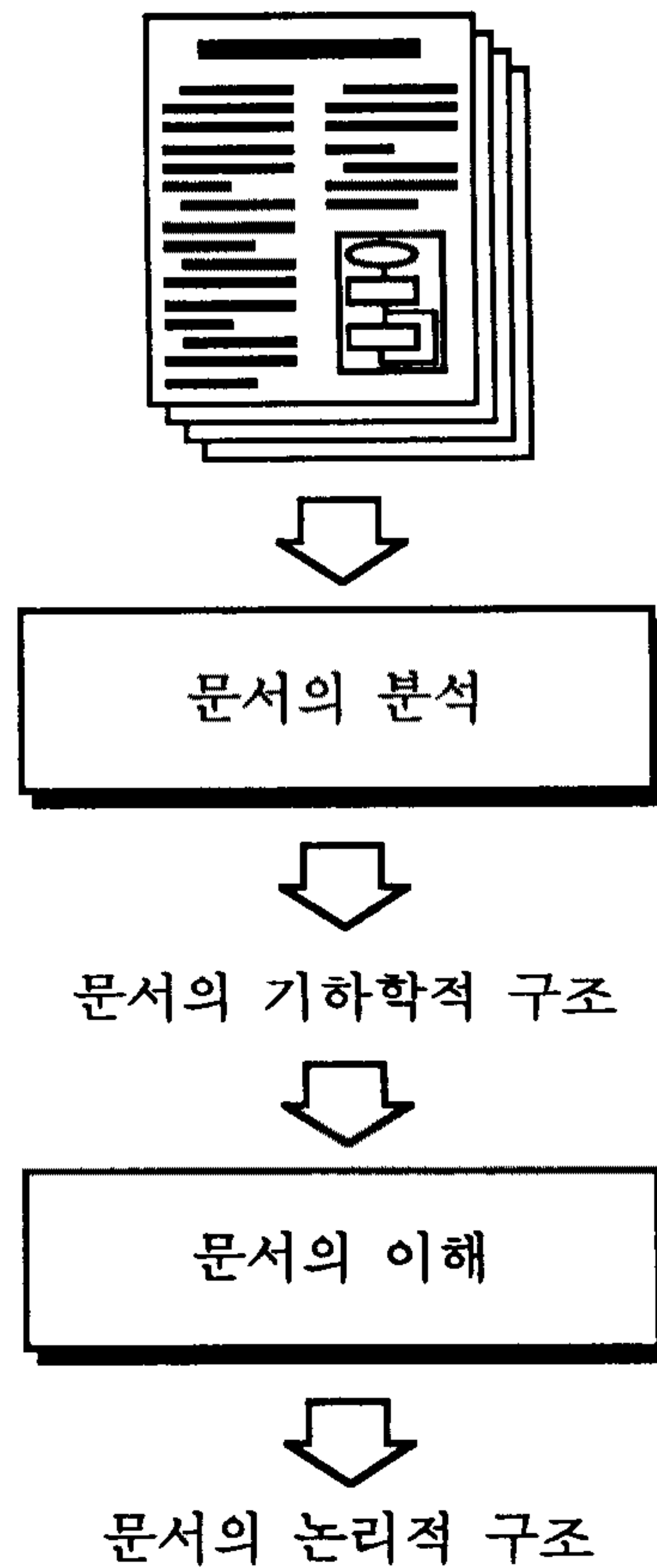


그림 3.1: 문서 처리의 과정

즉, 주어진 문서의 기하학적인 구조를 알아내는 과정은 문서의 분석이라고 볼 수 있으며, 이러한 기하학적 구조로부터 논리적 구조를 유추해 내는 것은 문서의 인식이라 할 수 있다. 이들 간의 관계는 그림 3.1에 도시된 바와 같다. 여기에서는 간단한 정리만을 목적으로 하며 보다 자세한 설명은 참고 문헌[4]을 참고하기 바란다.

1 문서의 형태 분석

문서의 형태 분석이란 문서 영상으로부터 문서의 기하학적인 구조를 추출하는 것이다. 즉 문서의 기울어진 정도를 알아내어 교정하고 문서를 단위 블록의 집합으로 분할한 후, 각 블록의 특성을 조사하여 문자 영역과 비문자 영역으로 분리하는 작업이다. 본 절에서는 이러한 문서 형태의 분석 작업을 위하여 기울어진 문서를 교정하는 방법과 이를 통해 교정된 문서를 실제로 분석하는 방법에 대하여 알아본다.

문서의 기울어짐은 그 정도가 적을 경우에도 문서 인식 시스템의 성능을 급격히 저하시킨다. 또한 문서의 형태 분석 방법 중 비교적 속도가 빠르기 때문에 널리 사용되고 있는 투영 윤곽 분할 방법(projection profile cut)이나 RLSA(Run Length Smoothing Algorithm) 등은 기울어진 문서에는 적용하기 어렵다. 따라서 문서의 기울어짐을 교정하는 것은 문서의 형태 분석을 위한 필수 과정이라 할 수 있다. 문서의 기울어진 정도를 측정하기 위하여 가장 일반적으로 사용되는 방법은 Hough 변환을 이용하는 방법과 투영 윤곽 분석 방법이 있다.

문서로부터 기하학적인 구조를 추출하는 방법은 상향식 방법과 하향식 방법으로 나눌 수 있다. 하향식 방법은 문서의 일반적인 특성에 근거하여 문서를 점점 작은 영역으로 분할하는 방법이고, 상향식 방법은 가장 기본이 되는 작은 부분에서 출발하여 유사한 특성을 갖는 부분을 단계적으로 병합함으로써 원하는 형태를 생성하는 방법이다. 하향식

방법은 정해진 형태의 문서를 빠르고 효율적으로 처리할 수 있으며 미리 알고 있는 지식을 문서의 처리에 사용할 수 있다는 장점을 가지고 있는 반면에 복잡한 형태의 문서나 기울어진 문서의 처리에 부적합하다는 문제를 가지고 있다. 반면에 상향식 방법은 다양한 형태의 문서를 처리할 수는 있으나 많은 처리 시간을 필요로 하므로 비효율적인 방법이라 하겠다. 하향식 방법으로는 smearing 방법(RLSA), 투영 윤곽 분할 방법, Fourier 변환 검출 방법, 원형 정합 방법, 교차 횡수를 이용한 방법등이 사용되고 있으며, 상향식 방법으로는 인접 선분 밀도 방법과 연결된 요소 분석 방법 등이 사용되고 있다.

2 형식 문서의 인식

형식 문서를 인식하는 방법은 그 사전 지식의 표현과 이용 방법에 따라서 크게 모델기반 접근방법(model-based approach)과 규칙기반 접근방법(rule-based approach) 및 구문론적 접근방법 (syntactic approach)으로 나눌 수 있다. 모델기반 접근방법은 문서의 형태 분석 및 이해를 위하여 이에 도움을 주는 지식을 각 모델에 대한 프레임(frame)이나 언어의 형태로 표현하는 방법이다. 이 경우 지식은 문제에 대한 지식과 이 지식을 다루기 위한 메타지식 (metaknowledge)으로 구성된다.

Kreich 등[5]은 이러한 지식을 (document - letter - business_letter) 와 같이 계층적으로 구성하고, 각 지식을 프레임 형태로 구성하여 문서에 대한 사전 지식을 표현하였다. 프레임 형태로 저장되어 있는 지식을 사용하여 형식 문서를 인식하기 위해서 모델기반 접근방법에서는 가설을 생성하고 선택한 후, 검증하고 평가하는 과정을 반복한다. 이러한 과정을 제어하기 위해서는 메타지식 뿐만 아니라 여러가지 종류의 경험적 지식이 사용되는데, 이러한 복잡한 지식의 제어에 적합한 블랙보드 모델이 사용되기도 한다.

모델기반 접근방법 중 근래에 가장 많이 연구되고 있는 것은 DFDL (Document

Form Definition Language)을 이용하는 방법이다. 이 방법은 문서를 사각형 형태의 영역의 집합이라 가정하고, 이를 언어의 형태로 표현함으로써 사용자가 좀 더 편리한 방법으로 문서에 대한 지식을 표현할 수 있도록 하여 준다. Higashino 등[3]은 문서의 크기 및 위치에 관한 정보와 제어에 필요한 정보를 포함하고 있는 언어를 이용하여 형식 문서의 구조를 표현하고, 이를 이용하여 입력된 형식 문서를 분석하고 인식하였다. 또한 Suen 등[7]은 특별히 형식 문서의 특성을 최대한 활용할 수 있는 저수준의 언어를 설계하고 이를 이용하여 여러가지 형태의 형식 문서를 인식하는 시스템을 개발하였다.

규칙기반 접근방법은 문서에 대한 지식을 각 구성 요소에 대하여 if <condition> then <action> 의 형태[1, 6]로 표현한 후 이를 이용하여 문서를 인식하는 방법이다. Tsujimoto와 Asada[6]는 각 영역별로 나뉘어진 문서를 기하학적 트리로 변환하고 제목과 내용의 계층적 관계를 나타내는 4가지의 변환 규칙을 사용하여 이를 논리적 트리로 변환하였으며, Dengle[2]은 업무용으로 사용되는 많은 양의 우편물을 통계학적으로 분석하여 얻은 일반적인 특성을 이용하여 SDB(statistical database)를 구성하고 이를 이용하여 문서를 인식하고자 하였다.

구문론적 접근방법은 문서의 공간적 구조를 분석하고 각 영역의 논리적 의미를 부여하기 위하여 CFG(context free grammer)를 사용하는 방법이다. CFG의 각 터미널 노드는 그 위치나 크기 등이 별도의 배열에 저장되어 있으므로 분할된 문서의 영역이 어느 문법과 일치하는가를 알면 즉시 문서를 이해할 수 있게 된다. 그러나 이 방법은 문법의 생성이 수동적으로 이루어진다는 것과 backtracking이 불가능하게 된다는 단점을 가지고 있다.

3 평가 및 토의

본 절에서는 현재까지의 문서 형태 분석 및 인식에 관한 연구들을 정리하고 이를 분석한다. 먼저 기존에 연구된 시스템 중에서 대표적인 것을 몇가지 정리해 보면 표 3.1과 같다.

표 3.1를 분석해 보면 대부분의 연구에서 문서의 형태 분석을 위하여 투영 윤곽 분할 방법이나 연결 요소 분석 방법, 또는 RLSA 방법을 사용하였다는 것을 알 수 있다. 이러한 방법을 사용하면 많은 경우에 입력 문서 영상으로부터 문자부와 그림부를 분리할 수 있다. 그러나 투영 윤곽 분할 방법은 복잡한 문서의 처리에는 적용할 수 없고, 연결 요소 분석 방법은 처리 시간이 너무 많이 소요되며 연결 요소를 결합시키기 위해서 많은 경험적 지식이 필요하다는 단점을 가지고 있다. 또한 RLSA 방법은 문서 영상의 평활화시에 필요한 임계값의 결정이 어렵고 이 임계값을 어떻게 결정하는가에 따라서 전체 시스템의 성능이 크게 좌우된다는 단점을 가지고 있다. 결국 한가지 방법만을 사용하는 것보다는 여러가지 방법을 병합하여 각각의 장점을 살리는 방향으로 연구가 진행되어야 할 것이다.

저자	기울기 교정	문서분석 방법	문서인식 방법	대상 문서	비고
Akiyama	투영윤곽분석	투영윤곽분할 교차 횟수 최소 인접사각형		신문 paperbacks 학술지	
Dengel	좌측공백분석	투영윤곽분할	SDB rule-based	business letter	(a)
Esposito	RLSA	투영윤곽분할	model-based	IEEE trans.	(b)
Fisher	Hough 변환	RLSA 연결요소분석	rule-based	US patent 형식	
Fujisawa		RLSA	SFDL model-based	patent 문서	
Kreich		연결요소분석	model-based		(c)
Lam		연결요소분석	model-based	business letter mail envelope	
Tsujimoto		투영윤곽분할	트리 변환 model-based	잡지 학술지	(d)
Viswanathan		투영윤곽 분할	syntactic	학술지	
Yamashita		RLSA	model-based	patent 문서	

- (a) expectation-driven
- (b) 학습 가능
- (c) ODA(Office Document Architecture)
- (d) 일반적인 문서 인식

표 3.1: 문서 인식 시스템의 예

제 2 절 GUI를 이용한 형식 문서의 표현

형식 문서를 인식하기 위한 구조 표현의 기본 요소는 선분과 그림 요소로 정의될 수 있으며, 문자 영역은 이러한 그림 요소와의 관계를 이용하여 표현되고 추출될 수 있다. 즉, 형식 문서를 인식하기 위해서는 먼저 앞에서 설명한 모델기반 접근방법이나 규칙기반 접근방법, 또는 구문론적 접근방법 등의 방법을 사용하여 주어진 문서의 형태를 표현하여야 한다. 그러나 기존의 방법을 이용하여 사용자가 문서의 형태를 표현하고자 하면 그 나름대로의 복잡한 문법을 잘 알고 있어야 하는 문제가 있다.

본 절에서는 이러한 문제를 해결하기 위하여 GUI 기법을 사용하여 문서의 구조를 표현하는 방법을 제시한다. 즉, 형식 문서의 구조를 표현하기 위하여 필요한 기본 요소들이 GUI 기법에서 제공되는 그래픽 연산자의 집합으로 정의될 수 있음을 보이고자 한다. 그림 먼저 사용한 그래픽 연산자에 대해서 알아보자.

정의 1 : (선택 연산자)

선택 (메뉴) : 메뉴 \in { 직선 , 사선 , 그림 , 복사 , 문자 }

선택 연산은 사용자가 메뉴 버튼을 클릭(click)하는 것으로 구현된다.

정의 2 : (위치지정 연산자)

위치지정 (0 | 1 , x, y)

1: button press

0: button release

x, y: 연산이 행해지는 좌표

위치지정 연산은 마우스의 움직임과 마우스 버튼의 클릭으로 구현된다. 이때, 드래깅 (dragging)과 같은 연산은 위치지정(1, x_1, y_1), 위치지정(0, x_2, y_2)와 같은 연산의 집합으로 표현할 수 있다.

정의 3 : (명칭부여 연산자)

명칭부여 (문자열)

이 연산은 위치지정 연산자를 선택한 후 나타나는 윈도우에 이름을 입력함으로써 수행된다.

1 구조 표현의 기본 정의

형식 문서의 구조는 다음과 같이 정의되는 구조의 기본요소에 의하여 표현된다.

정의 4 : (구조의 기본 요소)

구조의 기본요소는 형식 문서 공간 $\Omega_F = (\Sigma, \sigma_D, \sigma_P, \Delta)$ 의 부분 집합이다. 즉,

$$\text{구조의 기본 요소} \subset \Sigma \times \sigma_D \times \begin{pmatrix} P_{sx} \\ P_{sy} \\ P_{ex} \\ P_{ey} \end{pmatrix} \times \Delta$$

- $\Sigma = \Sigma_L \oplus \Sigma_D$
 - $\Sigma_L = \{L_1, \dots, L_l\}$: 선분의 집합,
 - $\Sigma_D = \{D_1, \dots, D_d\}$: 그림 요소의 집합
- $\sigma_D = \{D_v, D_h, D_s\}$: Σ 의 방향
 - D_v : 수직, D_h : 수평, D_s : 경사
- $\sigma_P = (P_{sx}, P_{sy}, P_{ex}, P_{ey})^T$ 는 Σ 의 위치를 나타내는 벡터
- Δ 는 인덱스 번호

위에서 정의한 구조의 기본요소를 얻기 위하여 본 시스템에서는 다음과 같은 그래픽 연산 과정을 거친다.

1. 선택 (직선 | 사선 | 그림)
2. 위치지정 (1, x_1 , y_1)
3. 위치지정 (0, x_1 , y_2) 또는 위치지정 (0, x_2 , y_1) : 직선의 경우
 위치지정 (0, x_2 , y_2) : 사선이나 그림의 경우

즉, 사용자는 그림 요소의 종류를 선택한 후 선분의 경우 양 끝점을, 그리고 그림 요소의 경우 이를 포함하는 인접 사각형을 마우스를 이용하여 표시함으로써 그 위치를 표현한다. 시스템은 위의 정보로부터 Σ 의 종류 및 위치(σ_P)를 얻을 수 있고 x, y 의 길이 비율을 이용하여 Σ 의 방향(σ_D)을 계산할 수 있다. Δ 는 사용자가 선택한 순서에 따라 주어진다.

2 같은 형태의 표현

같은 크기의 선분이 반복적으로 존재하는 경우는 사용자의 편의 및 내부 표현의 효율성을 위하여 다음과 같이 정의되는 구조를 사용한다.

정의 5 : (같은 형태의 표현)

그림 요소의 집합 $\Sigma = \{\Sigma_1, \dots, \Sigma_n\}$ 이 아래와 같은 성질을 갖는다면 해당 문서의 구조는 내부적으로 다음과 같은 형태로 표현될 수 있다.

$$(\Sigma, \sigma_D, \sigma_R, \Delta) =$$

$$(\Sigma, (D_H \oplus D_V \oplus D_S), (R_A \oplus R_B \oplus R_R \oplus R_L), \Delta_1, \dots, \Delta_n)$$

\in 같은 형태의 표현

- \forall_i, \forall_j (Σ_i 와 Σ_j 는 같은 종류의 서로 다른 그림 요소)
- \forall_i, \forall_j ((($P_{sx}^i = P_{sx}^j$) \cap ($P_{ex}^i = P_{ex}^j$)) \oplus (($P_{sy}^i = P_{sy}^j$) \cap ($P_{ey}^i = P_{ey}^j$)))

형식 문서의 경우는 길이와 방향이 같은 수평선이나 수직선 등 유사한 성질의 그림 성분이 다수 존재한다. 이러한 특성은 문서의 형태 분석에 매우 중요한 역할을 하며, 구조의 기본 요소만을 사용할 경우 사용자는 많은 수의 선분을 일일이 그려야 한다. 본 시스템에서는 다음과 같은 과정을 거쳐 유사한 성질의 그림 요소를 표현한다.

1. 선택 (복사)
2. 위치지정 ($1, x_1, y_1$), 위치지정 ($0, x_1, y_1$)
3. 위치지정 ($1, x_2, y_2$), 위치지정 ($0, x_2, y_2$)

먼저 메뉴에서 복사를 선택한다. 그리고 이미 지정한 그림 요소를 선택한 후 이 그림 요소가 위치할 곳을 지적한다. 이 경우 사용자는 그림 요소가 위치할 축 상의 임의의 점을 선택하면 되므로 정확한 지점을 표시하는 것보다 효율적일 수 있다.

3 문자 영역의 위치 표현

문자 영역의 위치는 그림 요소와의 상대적 위치 관계로 표현될 수 있으며. 하나의 문자 영역은 여러개의 문자 요소로 구성될 수 있다. 많은 경우에 사용자는 문자 영역 전체가 아닌 일부만을 처리하기를 원한다. 따라서 문자 영역의 일부만을 표현할 수 있는 방법을 제공하여야 하는데 이러한 목적으로 다음과 같은 연산자를 정의한다.

정의 6 : (문자 영역의 위치)

문자 영역의 집합 $\alpha = \{\alpha_1, \dots, \alpha_m\}$ 과 그림 요소의 집합 $\Sigma = \{\Sigma_1, \dots, \Sigma_n\}$ 간의

관계를 나타내는 집합 $\Gamma = \{\Gamma_1, \dots, \Gamma_k\}$ 이 존재하면, Γ_i 는 다음에 정의되는 행렬 M 에 의하여 표현될 수 있다.

$$M = \begin{cases} \Gamma_l & \text{if } (\alpha_i, \Sigma_j) \in \Gamma \\ 0 & \text{if } (\alpha_i, \Sigma_j) \notin \Gamma \end{cases} \quad \begin{matrix} \forall_l(\Gamma_l = (\alpha \mathfrak{R} \Sigma)), \\ \mathfrak{R} = \{R, L, A, B\} \end{matrix}$$

정의 7 : (문자 요소의 선택)

입의 문자 영역 α_i 에 대하여 요소선택연산자(jk, rs)는 다음과 같이 동작한다.

요소선택연산자(jk, rs) $\alpha_i =$

$$\text{요소선택연산자(jk, rs)} \begin{bmatrix} w_{11}^i w_{12}^i \dots w_{1m}^i \\ w_{21}^i w_{22}^i \dots w_{2m}^i \\ \vdots w_{n1}^i w_{n2}^i \dots w_{nm}^i \end{bmatrix} \Rightarrow w_{jk}^i \dots w_{rs}^i$$

이를 위한 그래픽 연산은 다음과 같다.

1. 선택 (문자)
2. (a) 위치지정 (1, x_1 , y_1), 위치지정 (0, x_1 , y_1) 또는
(b) 위치지정 (1, x_1 , y_1), 위치지정 (0, x_2 , y_2)
3. 명칭부여 (문자열 명칭)

(a)의 연산자는 상하좌우가 그래픽 요소로 둘러싸인 내부의 모든 문자 영역을 선택하는 연산을 위하여 사용되며 (b)의 연산자는 지정된 영역 내의 문자 만을 선택할 경우 사

용된다. 본 시스템에서는 사용자가 문자 영역의 위치를 지적하면 상하좌우 방향으로 가장 가까운 거리의 그림 요소를 자동으로 찾아내어 행렬 M 을 생성한다. 또한 문자 영역의 의미를 표현하기 위하여 문자 영역의 명칭을 부여할 수 있다. 또한 처리하기를 원하는 영역만을 마우스로 지정하여 원하는 정보를 얻을 수 있다.

제 3 절 실험 및 분석

본 시스템은 Sony News 워크스테이션에서 C 언어를 사용하여 구현되었다. 실험에 사용된 문서는 신상명세서로서 News Image Reader를 통하여 400 DPI의 해상도로 입력되었으며 GUI 부분의 구현을 위하여 Motif를 사용하였다. 실제 시스템의 동작을 살펴보면 다음과 같다.

먼저 스캐너를 이용하여 문서를 입력하면 그림 3.2와 같이 입력된 문서가 화면에 출력된다. 화면에 출력된 문서 영상을 이용하여 사용자가 마우스로 직선, 사선, 문자 영역 그리고 그림 영역 등의 문서 구조를 표현하면 그림 3.3과 같이 화면 상에 굵은 라인과 검은 영역으로 사용자가 표현한 영역이 나타난다. 사용자가 문서의 구조를 모두 표현하면 그림 3.4와 같은 자료 구조가 생성되어 화일에 저장되며, 이를 표로 나타내 보면 표 3.2와 같다.

여기서 〈 구조의 기본 요소 〉는 앞 절에서 정의한 문서의 기본 요소인 각 그림 요소의 인덱스와 형태, 방향, 그리고 시작점의 위치와 그 크기를 나타내고 있으며, 〈 같은 형태의 표현 〉은 같은 형태의 그래픽 요소들을 묶어서 표현하고 있다. 여기서 형태란과 원소란의 숫자들은 문서의 기본 요소를 표현한 표의 인덱스이다. 또한 〈 문자 영역의 위치 〉는 문자 영역의 명칭, 위치, 크기, 그리고 상하좌우의 그림 요소가 무엇인가를 나타내고 있다. 따라서 위의 결과를 자세히 살펴보면 GUI 기법을 이용하여 문서의 구조 분석에 필요한 모든 정보를 정확히 표현하였음을 알 수 있다.

만일 사용자가 사용한 문서의 구조를 FDL을 이용하여 작성한다고 하면 직접 측정자를 사용하여 각 요소의 길이를 측정하고, 서로 간의 상대 위치를 조사하여 그 구조를 표현하여야 한다. 한가지 예를 들어보면 위의 실험에서 사용한 문서의 구조를 FDL로 표현한 결과가 표 3.3이다. 즉 이와같이 복잡한 과정을 거쳐 표현해야 하는 문서의 구조를 GUI 기법을 사용하여 보다 쉽게 같은 내용의 정보를 표현할 수 있었다.

직선

사선

그림

복사

문자

취소

종도

Personel Profile

Yeon Joo Inc. 1991 / 03 / 15


ID Number	580312 - 001	Name	Kim YongHun	
Current Position	PRESIDENT	Current Income	W 523,500	
Entrance Date	1905 / 03 / 02	Entrance Class	A01 - 1234	
Entrance Position	Representative	Military	Navy	
Current Address				
Domiciled Address				
Education	Graduate	School	Major	
	High School	GyungGi		
	College			
	University	Korea	Economics	
	Graduate School			
Promotion Date	1910 / 11 / 24			

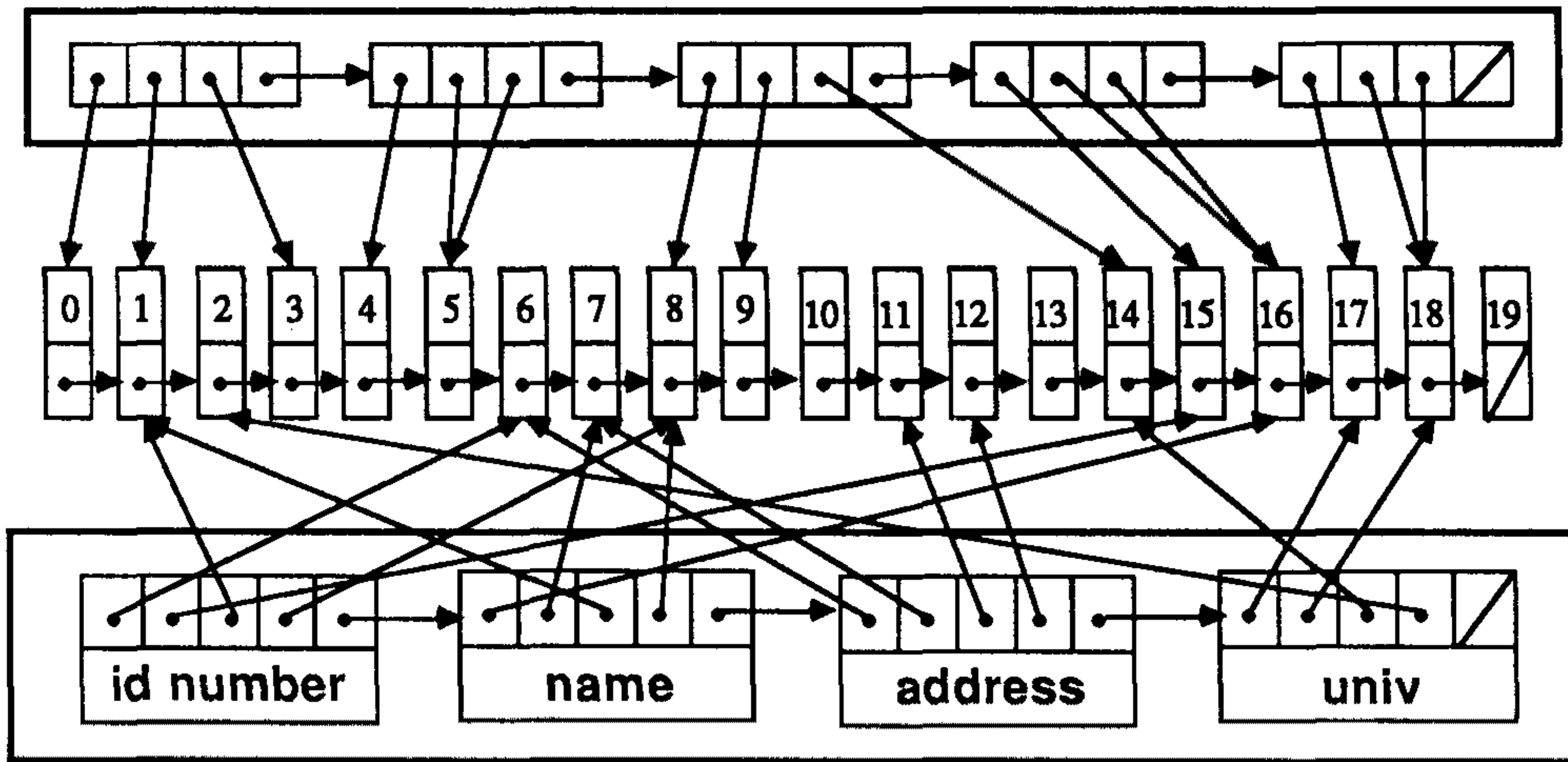
그림 3.2: 화면에 출력된 입력 문서

- 직선
- 사선
- 그림
- 복사
- 문자
- 취소
- 종료

Personel Profile			
Yeon Joo Inc.		1991 / 03 / 15	
ID Number		Name	
Current Position	PRESIDENT	Current Income	W 523,500
Entrance Date	1905 / 03 / 02	Entrance Class	A01 - 1234
Entrance Position	Representative	Military	Navy
Current Address			
Domiciled Address			
Education	Graduate	School	Major
	High School College University Graduate School	GyungGi [Redacted]	Economics
Promotion Date	1910 / 11 / 24		

그림 3.3: 사용자에게 의한 문서의 구조 표현 결과

같은 형태의 표현



문자 영역의 표현

그림 3.4: 문서 구조의 내부 표현

〈 구조의 기본 요소 〉

index	type	direction	x	y	x size	y size
0	line	horiz	75	234	1812	0
1	line	horiz	75	264	1812	0
2	line	horiz	75	1551	1812	0
3	line	horiz	75	1665	1812	0
4	line	vert	78	234	0	1428
5	line	vert	1887	234	0	1428
6	line	vert	435	270	0	1401
7	line	vert	1512	270	0	1281
8	line	horiz	84	375	1425	0
9	line	horiz	84	495	1425	0
10	line	horiz	84	615	1425	0
11	line	horiz	84	729	1425	0
12	line	horiz	84	849	1425	0
13	line	horiz	84	963	1425	0
14	line	horiz	84	1080	1425	0
15	line	vert	795	273	0	450
16	line	vert	1149	273	0	450
17	line	vert	792	963	0	594
18	line	vert	1149	963	0	594
19	graphic	vert	1527	291	333	444

〈 같은 형태의 표현 〉

type	direction	relation	size	elements
0	horiz	below	3	1 2 3
4	vert	right	1	5
8	horiz	below	6	9 10 11 12 13 14
15	vert	right	1	16
17	vert	right	1	18

〈 문자 영역의 위치 〉

label	x	y	x size	y size	left	right	up	down
id number	435	264	360	111	6	15	1	8
name	1149	264	363	111	16	7	1	8
address	435	729	1077	120	6	7	11	12
univ	855	1335	240	78	17	18	14	2

표 3.2: 표로 나타낸 문서의 구조

구조 표현의 기본 요소					같은 형태의 표현							
LHTM	75,	234,	1812,	1	LHB	1,	2,	4				
LHTM	75,	264,	1812,	2	LVR	5,	6,	6				
LHTM	75,	1551,	1812,	3	LHB	9,	10,	15				
LHTM	75,	1665,	1812,	4	LVR	16,	17,	17				
LVTM	78,	234,	1428,	5	LVR	18,	19,	19				
LVTM	1887,	234,	1428,	6	문자 영역의 위치 표현							
LVTM	435,	270,	1401,	7	IA	1 ;	IB	8 ;	IL	6 ;	IR	15
LVTM	1512,	270,	1281,	8	/* id number */							
LHTM	84,	375,	1425,	9	IA	1 ;	IB	8 ;	IL	16 ;	IR	7
LHTM	84,	495,	1425,	10	/* name */							
LHTM	84,	615,	1425,	11	IA	11 ;	IB	12 ;	IL	6 ;	IR	7
LHTM	84,	729,	1425,	12	/* address */							
LHTM	84,	849,	1425,	13	IA	14L2 ;	IB	2 ;	IL	17 ;	IR	18
LHTM	84,	963,	1425,	14	/* univ */							
LHTM	84,	1080,	1425,	15								
LVTM	795,	273,	450,	16								
LVTM	1149,	273,	450,	17								
LVTM	792,	963,	594,	18								
LVTM	1149,	963,	594,	19								
GVTM	1529,	291,	444,	20								

표 3.3: FDL에 의한 신상 명세서의 구조 표현

제 4 절 결 론

문서의 구조 분석은 여러가지 처리 기술의 개발로 많은 진전이 있었으나 문서의 인식에 관한 연구는 현재 그 수준이 매우 미약한 실정이다. 따라서 현재의 추세는 특정한 형식의 문서를 대상으로 미리 사용자가 문서에 대한 지식을 입력하고 이를 이용하여 형식 문서를 인식하려는 방향으로 나아가고 있다. 이 경우 중요한 것은 지식의 표현, 저장, 활용 수단이므로 문서 처리에 관한 연구와 병행하여 이러한 지식 처리 방법에 관한 연구도 병행하여 진행되어야 한다. 본 논문에서는 이러한 추세에 따라 그래픽 사용자 인터페이스를 이용하여 사용자가 문서의 구조에 대한 지식을 표현할 수 있는 형식 문서 인식 시스템을 제시하였다.

이러한 연구들을 기반으로 문자부와 그림부가 효과적으로 추출되고 그 역할이나 의미가 인식된 후, 문자부는 문자 인식 시스템에 의하여 처리되고 그림부는 그래픽 처리 시스템에서 인식되거나 압축 또는 부호화되는 문서 처리 시스템을 구축할 수 있을 것이다.

참고 문헌

- [Aki90] T. Akiyama and N. Hagita, "Automated entry system for printed documents," *Pattern Recognition*, vol. 23, no. 11, pp. 1141-1153, 1990.
- [Den90] A. Dengel, "Document image analysis - expectation-driven text recognition," *Proc. SSPR90*, pp. 78-87, 1990.
- [Hig86] J. Higashino, H. Fujisawa, Y. Nakano and M. Ejiri, "A knowledge-based segmentation method for document understanding," *Proc. 8th Int. Conf. on Pattern Recognition*, pp. 745-748, 1986.
- [Kre91] J. Kreich, A. Luhn and G. Maderlechner, "An experimental environment for model based document analysis," *Proc. of ICDAR91*, vol. 1, pp. 50-58, 1991.
- [Tsu90] S. Tsujimoto and H. Asada, "Understanding multi-articled documents," *Proc. 10th Int. Conf. on Pattern Recognition*, vol. 1, pp. 551-556, 1990.
- [Yan91] C. D. Yan, Y. Y. Tang and C. Y. Suen, "Form understanding system based on form description language," *Proc. of ICDAR91*, vol. 1, pp. 283-293, 1991.
- [Jok92] 권 재욱, 조 성배, 김 진형, "문서 분석 및 인식의 연구 동향," *인공지능연구실 메모*, 한국과학기술원, 1992.

제 4 장 삼차원 물체를 인식하기 위한 모델 기반 시스템의 개발

본 연구에서는 삼차원 물체(3-D objects)를 인식하기 위한 모델 기반 시스템(model based system)을 제안한다. 점이나 모서리에 의한 표현보다 표현력이 풍부하고 입력영상으로부터의 표면 표현의 추출이 Volume에 의한 표현 방법보다 훨씬 더 용이한 surface 표현 방법을 사용하였으며, 순차적 탐색의 단점을 극복하기 위해 A* 최적 탐색의 방법을 도입하였다.

최적 탐색을 위한 비용함수는 현재까지의 탐색 비용과 아직 처리되지 않은 부분의 추정 비용으로 구성된다. 최적 탐색이 효율적이 되기 위해서는 이 추정 비용이 실제의 비용에 근사해야 한다. 본 과제에서는 최적 탐색을 위한 비용 함수의 추정 비용으로 선행 탐색에서 구한 매칭 가능 쌍의 수를 사용하였으며 이 추정비용이 적당하다는 것을 실험적으로 증명하였다.

제 1 절 선행탐색에 의한 삼차원 물체 인식

본 과제에서는 삼차원 물체(3-D objects)를 인식하기 위한 모델 기반 시스템(model based system)을 제안한다. 삼차원 물체의 인식은 시각 시스템의 근본적인 문제로 공장 자동화, 군사적 응용 등 다양한 분야의 필수적인 부분이다. 물체의 인식은 카메라에 잡힌 장면(scene)에서 그 장면을 구성하는 물체(object)들을 찾아내고 각각의 물체의 삼차원 공간상에서의 방향, 위치를 찾아내는 문제이다[Bes85, Chi86].

과거에는 대부분의 컴퓨터 시각 시스템이 명암 화상(intensity image)을 처리하여 물체를 인식하였다. 그러나 명암 화상은 조명 상태, 그림자, 반사 등에 민감하게 영향을 받기 때문에 그것을 처리하기 매우 힘들고, 삼차원 정보를 추출하기 위해 매우 복잡한 과정이 필요하다. 최근에는 카메라에서 물체의 표면의 각 점까지의 거리를 표시하는 거리 영상(range image)을 사용하여 명암 화상 처리의 어려움을 극복하고 있다. 거리 영상은 stereo image 처리 등 간접적인 방법으로 구할 수도 있지만, 레이저를 사용하여 삼각형 법칙을 사용하여 직접적으로 구할 수 있다. 이러한 거리 영상은 카메라에서 물체의 표면의 각 점까지의 거리를 표시하는 수치의 이차원 배열로 표현된다. 거리는 물체의 기하학적 정보에만 좌우되고, 조명 상태 등에 무관하기 때문에 삼차원 물체 인식을 위해 많이 사용된다. 또한 물체의 다양한 표면 정보를 추출하기 쉽고, 그것을 사용하여 삼차원 물체의 인식이 용이하다[Bha84, Bes86, Bol86, Bra85, Che89, Fan89, Fau86, Gri84, Gri87, Jai88, Osh83]. 본 과제는 이러한 거리 영상을 이용한 삼차원 물체 인식 시스템을 개발한다.

일반적으로 물체 인식 시스템은 모델 구성(modelling), 특징 추출(feature extraction), 매칭 전략(matching method)으로 구성된다. 모델 구성은 인식 시스템이 인식할 물체들을 구성하는 방법에 관한 연구이고, 특징 추출은 점, 모서리, 표면 등 물체를 구성하는 특징점을 추출하는 과정으로 그것에 좌우하여 매칭 방법도 달라진다. 매칭 방법

은 장면과 모델이 특징점들로 구성된 후 장면의 각각의 특징점과 대응되는 모델의 특징점을 찾고 그 모델의 삼차원 상의 위치 변형 (transformation)을 찾는 방법이다. 이러한 각 과정에 대한 연구가 전 세계적으로 많이 추진되었지만 모든 경우에는 아직 적용할 수 없고 부분적인 경우에만 만족할만한 결과를 보이고 있다.

물체는 point, edge, contour, surface, volume 등 다양한 계층으로 표현할 수 있다. 낮은 계층으로 표현하면 추출하기는 쉽지만 표현할 수 있는 물체의 종류가 제한되고, 높은 계층은 표현 가능한 물체의 종류는 다양하지만 영상에서 그 표현을 추출하기 매우 힘들다. volume으로 물체를 표현하는 방법은 각 물체의 고유 좌표계상에서 물체를 표현하는 방법으로 가장 이상적인 방법이지만, 명암 영상이나 거리 영상에서 그 특징을 추출하기는 몇몇 규칙적인 물체를 제외하고는 거의 불가능하다. point로 물체를 표현하면 그 추출은 매우 쉽지만 한 물체의 표현이 다른 물체의 표현과 혼돈될 수 있다. 따라서 물체의 표현은 추출하기 쉽고 표현력이 충분한 것으로 선택해 한다. 이러한 관점에서 surface 표현 방법은 점이나 모서리 표현보다 표현력이 풍부하고 입력 영상에서 표면 표현의 추출은 Volume 표현보다 훨씬 더 용이하다. 즉 표면 표현 방법은 현재의 기술로는 일반적인 물체의 표현에 가장 적합한 방법이다. 표면 표현 방법은 크게 세가지 방법으로 세분된다. 첫번째 방법은 EGI나 곡률 히스토그램 [Yan86]등과 같이 입력 영상 전체의 특징을 추출하는 방법인데 한 영상에 여러 물체가 겹쳐 있는 경우에는 처리하기 매우 힘들다. 둘째는 물체 표면의 종류에 관계없이 물체의 표면을 평면의 표면 조각으로 근사시키는 방법[Bha84, Fau86, Gri84, Osh83]으로 다면체의 인식에는 적합하지만 곡면이 포함된 영상의 처리에는 부적합하다. 세번째 방법은 물체 고유의 모서리 부분을 추출하여 각 표면 조각을 평면이나 곡면(일반적으로 2차 곡면으로 근사)으로 표현하는 방법으로[Bes86, Bes88, Bra85, Fan87] 물체를 보는 관점의 변화에 덜 민감하고 물체의 실제 경계선을 따라 영상을 분할하기 때문에 두번째 방법보다 적은 수의 표면 조각으로 입력 영상을 표현할 수 있다. 본 과제에서는 거리 영상에서 추출하기 쉽고 표현력이 우수한 표면으로 물체를 표현한다. 즉, 물체의 실제 모서리를 추출하여 표면을 여러 개의 작은 표면 조각 (surface patch)으로 구성한다. 하나의 물체는 그의 표면 조각들과 연결 관계에

의해 표현된다. 이러한 연결 관계는 속성 그래프(attributed graph)로 표현된다. 각각의 표면 조각은 노드(node)로 표현하고 표면 조각들간의 관계는 링크(link)로 표현된다.

인식 시스템이 인식할 모델을 구성하는 방법에는 한 관점 (viewpoint)에서 본 장면, 여러 관점에서 본 장면들, 완전한 삼차원 표현 등 다양한 방법이 있다. 한 관점에서 본 장면으로 물체를 표현하면 혼돈되는 경우가 생길 수 있고, 완전한 삼차원 표현은 영상으로부터 그 추출이 어렵다. 이러한 문제점들을 해결하기 위해 본 과제에서는 다른 많은 인식 시스템처럼 하나의 삼차원 물체를 여러 관점에서 본 장면들로 표현하였다. 규칙적인 물체는 적은 수의 장면으로 표현될 수 있지만 불규칙적인 물체는 많은 장면들이 필요하다.

모델 물체와 인식하려는 장면이 표면 조각들의 속성 그래프로 표현된 상태에서, 매칭 과정은 장면 표면 조각(scene surface patch: SSP)과 모델 표면 조각(model surface patch: MSP)과의 대응 관계를 찾아 내고 삼차원 정보를 추출한다. 고려하는 물체가 rigid하기 때문에 입력 물체를 모델에 정합한다는 것은 각 SSP가 그 모델의 어느 MSP에 대응되고 대응된 SSP의 구조가 MSP의 구조와 같아야 한다. 일반적으로 구조가 같은지의 검사는 부분적인 기하학적인 제약들을 검사함으로써 수행할 수 있다. 부분적 기하학적 제약은 1차 제약(unary constraint), 2차 제약(binary constraint), 3차 제약(ternary constraint)을 사용한다. 1차 제약은 하나의 SSP와 하나의 MSP와의 대응 가능성을 검사하는 것으로 표면 조각의 면적, 표면 형태, 곡률(curvature) 등을 사용하여 두 표면 조각의 대응 가능성을 검사한다. 2차 제약은 두개의 SSP와 대응되는 두개의 MSP와의 대응 가능성을 검사하는 제약이다. 두 표면 조각의 거리, 각도, 접속 형태 등을 사용하여 두 쌍의 표면 조각의 매칭 상태를 검사한다. 3차 제약은 3개의 SSP와 대응되는 세개의 MSP가 구조적으로 비슷한가를 검사하는 제약으로 normal vector의 triple product의 부호를 조사하거나 삼차원 기하학적인 조사로 검사한다. 이러한 부분적인 구조 검사는 전체적으로는 옳지 않을 수 있지만 부분적으로는 옳다고 결론을 내릴 수 있다. 많은 부분적인 정합으로부터 전체적으로 옳은 정합을 유도해가는 방법은 일반적으로 가정 설정 및 검증(hypothesize and test)[Bol86, Che89, Fan89, Fau86,

Gas84, Gri84, Gri87, Gri90, Osh83], relaxation labelling[Bal82, Bha84] 및 pose clustering[Bal82, Sto87] 등의 방법이 있다.

가정 설정 및 검증 방법은 부분적인 정합에서 어떤 해를 제시하고 나머지 쌍을 가지고 그 가정의 옳고 그름을 판가름하는 방법이다. 이 방법은 enumerative search 방식이므로 항상 해를 제공하지만 많은 탐색 시간을 필요로 하므로 이를 줄이려는 휴리스틱 한 방법들이 연구되고 있다. Relaxation labelling 방법은 각각의 SSP에 대응될 수 있는 MSP들을 구한 다음, 근처의 SSP와의 관계를 고려하여 그 후보의 수를 줄여나가는 방법이다. 이 방법은 전체 비용함수의 최소값을 구하기 위해 Gradient descent 방식을 사용하기 때문에 부분적인 최적 해에 도달하여 애매한 해를 얻을 수 있다. pose clustering은 한 SSP와 한 MSP의 쌍에서 발생할 수 있는 모든 해를 구한 다음 그 해들에 한 표를 추가하고 이러한 방법으로 모든 가능한 MSP와 SSP의 쌍을 고려하여 가장 많은 표를 얻은 것을 최종 정답으로 인정하는 방법이다. 이 방법은 해의 차원이 n 차이면 n 차원의 배열을 유지해야 하고, 기하학적인 변형을 확실히 구할 수 있는 다면체의 인식에는 적합하지만 곡면을 포함하고 여러 물체가 겹쳐 있는 경우에는 매우 복잡한 과정을 거쳐야 한다.

본 과제에서는 표면 조각을 매칭의 기본 단위로 사용하여 부분적인 기하적인 제약(geometric constraint)을 사용하여 한 쌍의 SSP와 MSP에서 출발하여 점진적으로 쌍을 확장해 나간다. 한 쌍의 매칭 쌍으로 부터 점진적 확장을 위해 순서적 제약 트리 탐색(sequential constrained tree search)의 방법을 사용하였다. 순서적 제약트리탐색 방법은 enumerative search 방식인 트리 탐색 방법에 인식하려는 물체의 기하학적인 특성을 반영하여 표면 조각들간의 제약을 고려하여 탐색 시간을 줄이려는 방법으로 단일 물체만 존재 하는 영상을 인식하기에는 매우 효율적인 방법이다. n 개의 표면 조각으로 구성된 장면 물체와 m 개로 구성된 모델 물체와의 매칭을 위해 생성되는 단말 노드는 nm 개이지만 제약들을 이용하여 다항 갯수로 줄일 수 있다. 그러나 여러 물체가 겹쳐져 있는 일반적인 상황에서는 여전히 exponential 갯수의 단말 노드가 생성된다[Gri90]. 왜냐하

면 하나의 모델을 고려할 때 그 모델의 MSP에 해당되지 않는 SSP가 생긴다. 이는 그 모델과 다른 물체의 표면 조각일 수도 있고, 입력 영상에서 표면 조각을 생성할 때의 오류일 가능성도 있다. 이러한 상황을 해결하기 위해 탐색 트리에 wild match 노드를 두어 이를 해결 해야 하는 데 이러한 상황에서의 제약 트리 탐색은 exponential 시간이 소요된다.

트리를 탐색하는 과정에서 생성된 부분 매칭 쌍(partial matching pair)은 나머지 아직 처리하지 않은 대응가능 쌍의 수를 많이 줄일 수 있다. 그 부분 매칭 쌍이 해의 경로의 일부분이면 나머지 처리하지 않은 MSP에 해당되는 SSP의 수는 극히 적을 것이고 트리의 깊이가 깊어지면 정확하게 하나만 남을 것이다. 해의 경로의 일부분이 아니면 해당하는 SSP는 없을 것이다. 이러한 관찰을 이용하면 제약트리 탐색의 비효율성을 개선할 수 있다. 본 과제에서는 현재까지의 해의 경로를 이용하여 아직 처리하지 않은 SSP의 대응가능 MSP의 수를 줄이는 방법을 제안하고 이를 선행 탐색 제약 트리라 명명하였다. 이러한 선행 탐색을 이용하여 탐색 공간을 줄일 뿐 아니라 현재의 탐색 지점에서 탐색 종료를 검사할 수 있는 강력한 criteria를 생성할 수 있다.

본 과제에서는 순차적 탐색의 단점을 극복하기 위해 A* 최적 탐색의 방법을 도입하였다. 최적 탐색을 위한 비용함수는 현재까지의 탐색 비용과 아직 처리되지 않은 부분의 추정 비용으로 구성된다. 최적 탐색이 효율적이 되기 위해서는 이 추정 비용이 실제의 비용에 근사해야 한다. 본 과제에서의 최적 탐색을 위한 비용 함수의 추정 비용은 선행 탐색에서 구한 매칭 가능 쌍의 수를 사용하였으며, 이 추정비용이 적당하다는 것은 실험적으로 증명하였다.

제 2 절 속성 그래프의 생성

본 장에서는 실제 거리 영상에서 속성 그래프(attributed graph)를 생성하는 과정을 설명한다. 속성 그래프는 $G = (N, L)$ 로 표현되는데 N 은 노드의 집합으로 각 노드는 영상에서 추출된 표면 조각을 표현하고 각종 속성(attribute)을 포함하고 있다. L 은 링크의 집합으로 각 링크는 연결된 두 표면 조각의 관계 정보를 표현하고 각종 속성이 포함된다.

1 노드와 링크의 속성

노드 속성은 한 표면 조각의 기하학적인 성질을 표현한다. 예를 들어 표면의 'type'은 가능한 하나의 노드 속성이 될 수 있다. 이론적으로는 노드 속성은 변형에 무관한 특성을 가져야 한다. 즉, 관점이 달라져도 그 표면 조각의 속성이 변하지 않는 것을 의미한다. 그러나 표면의 면적과 같은 속성은 물체를 보는 관점에 따라 그 값이 달라진다. 따라서 노드 속성은 변형에 무관한 것과 그렇지 않은 것으로 분류해야 한다. 변형에 좌우되는 속성, 예를 들어 면적 같은 것은 기하학적인 제약을 많이 가하지 못하지만 그 후보의 수를 많이 줄일 수 있기 때문에 탐색 과정에서 능동적으로 사용해야 한다.

링크 속성은 연결된 두 노드(두 표면 조각)의 관계 정보를 표현한다. 예를 들어 'type of adjacency'는 가능한 하나의 링크 속성 중의 하나이다. 링크 속성도 원칙적으로는 변형에 무관한 성질을 가져야 한다. 링크 속성도 노드 속성과 마찬가지로 변형에 무관한 것과 그렇지 않은 것으로 나눌 수 있다.

이러한 속성들은 최소한 한 영상이나 모델에서 두 개의 표면 조각이 똑같은 속성과 값을 가지지 말아야 한다. 표 4.1에서는 지금까지의 연구가들이 사용한 속성들을 나열

하고 있다. 표면을 물체의 특성점으로 사용할 경우에는 수치적인 속성(면적, 반지름, perimeter, compactness 등), 기호적인 속성(종류, 곡률의 부호 등) 등이 주로 노드 속성으로 사용되고 링크 속성으로는 수치적인 속성(거리, 각도, 연결된 표면 조각의 수, 연결 강도)와 기호적인 속성(연결 종류) 등이 주로 사용되었다.

본 과제에서 실험한 결과, 많은 표면 조각들은 거의 비슷한 수치 또는 기호 속성을 가지므로 1차 제약만으로는 SSP들과 MSP들과의 대응 관계를 찾기는 불가능하고 큰 임계값을 사용하여 1차 제약에 의한 후보들을 선정하고 2차, 3차 그리고 변형의 오차등을 사용하여 가정을 검증하는 방식을 사용하는 것이 바람직하다.

2 영상 분할

본 절에서는 거리 영상에서 표면 조각들로 분할하는 방법을 설명한다. 이 방법은 Fan 등[Fan87]이 개발한 방법으로 비슷한 방법들이 많이 발표되었다[Bes88, Bra85]. 표면 곡률의 성질을 표면 경계선을 추출하거나 region-based 방식으로 영상 분할하는 데 사용할 수 있다. 표면의 한 점에서의 표면 곡률은 측정되는 방향에 따라 그 값이 달라지고, 그 중에서 최대, 최소값의 곡률을 최대 곡률, 최소 곡률이라하고 이 두 값으로부터 모든 방향의 곡률값을 계산할 수 있다.

거리의 차이로 발생하는 경계선을 점프 경계선(jump boundary)이라 하고 거리의 차는 거의 없지만 방향이 달라서 생기는 경계선을 크리지 경계선(crease boundary)이라 한다. 물체의 경계선은 대부분 이 두 가지의 경계선이다. 표면 곡률의 성질에 의해, 점프 경계선은 경계선의 직교 방향으로의 곡률값이 그 경계선에서 0을 통과하는 현상(zero crossing)을 발견할 수 있다. 크리지 경계선은 그 경계선에서 부분적인 극값을 가진다. 이러한 성질을 이용한 점프, 크리지 경계선을 추출하기 위한 전체적인 블록 다이어그램은

다음과 같다.

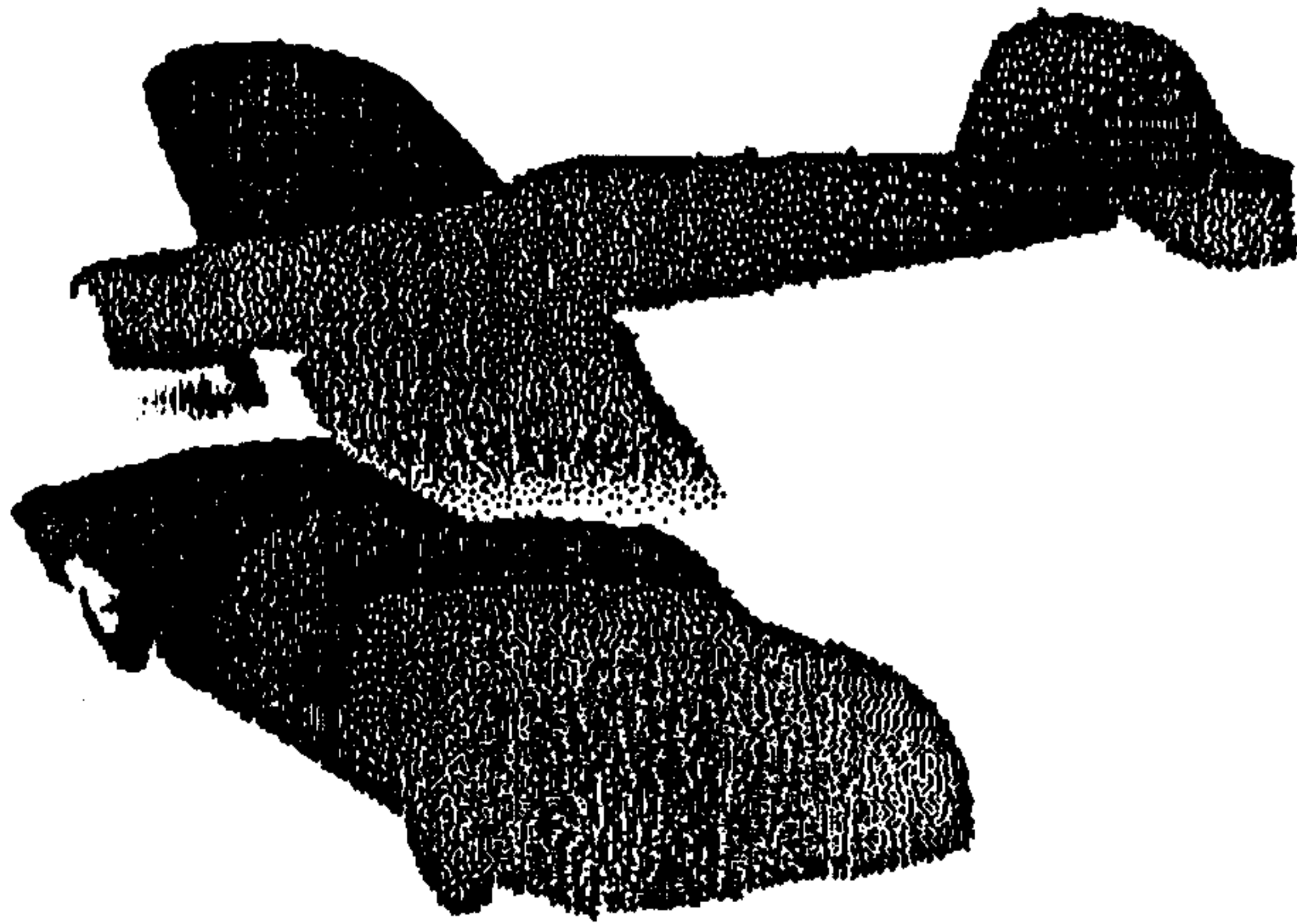
곡률의 계산은 이차 미분의 계산을 요하기 때문에 노이즈의 증폭으로 노이즈에 매우 민감하다. 노이즈의 영향을 최소화하기 위한 평활화 방법들 median filter, nearest neighborhood smoothing, maximum likelihood smoothing, edge preserving smoothing 그리고 adaptive smoothing 등이 개발되었다. 본 과제에서는 경계선이 무디어 지는 단점은 있지만 가장 널리 사용되는 가우지언 평활화 방법을 사용하였다.

모든 점에서 표면곡률을 계산한 다음, 그 값이 부분적으로 극값에 해당하거나 0을 통하는 경우에는 경계선 영상에 1을 표시하고 그 외의 경우에는 0을 표시한다. 연속성의 원칙을 지키면서 연결된 경계선을 연결하고 주위에 선분이 있는 경우에는 연장하여 폐곡선을 만든다. 이렇게 하여 입력 영상을 작은 조각들로 분할한다.

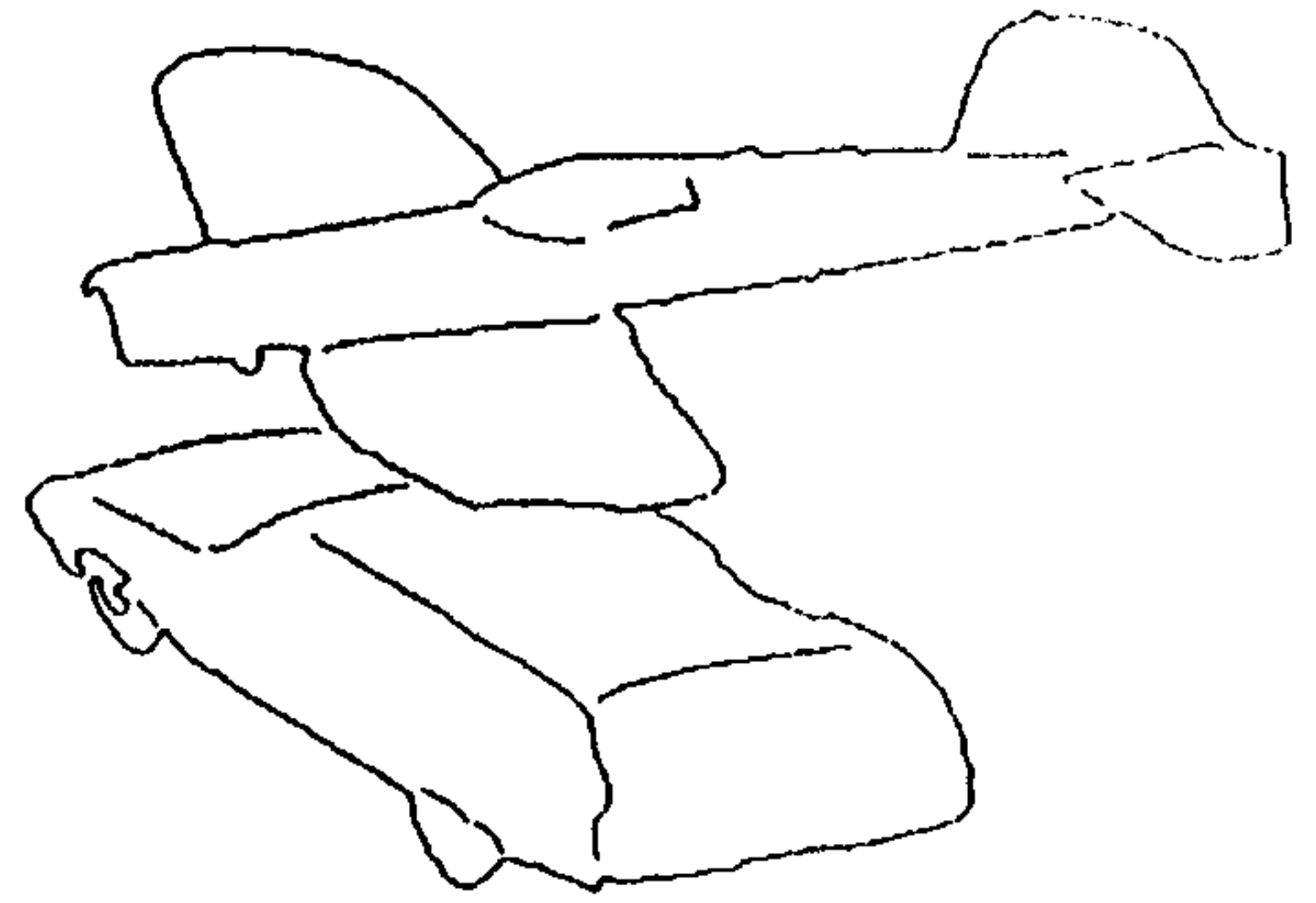
각각의 분할된 표면 조각은 이차 표면식에 의해 최소 자승 오차 방식으로 fitting하여 그 곡면의 종류를 찾는다. 그림 4.1은 거리 영상에서 영상을 분할하는 과정을 간단히 보인 것이다. (b)에서는 (a)의 거리 영상을 곡률의 극값과 0 교차점에 의한 경계선의 선분을 표시한 것이고, (c)에서는 영상 분할 과정이 완료된 후에 구해진 표면 조각들을 표시한 것이며 이러한 표면 조각들과 그들의 관계를 그래프로 표시한 것이 (d)이다.

	Feature	Shape(unary)	Relation (binary)
Polyhedral Object Grimson	point	position normal	distance angle direction * triple-product
Curved Object Grimson	linear line circular line	length radius swept-angle pointing	angle, distance component, circular center circle swept angle cross distance, cross component cross angle
General Object Faugeras Hebert	point line plane		distance (point) angle(plane) angle, distance(line)
Specific Object Bolles Horaud	straight dihedral circular dihedral straight tangential	length radius	angle between two surfaces widths of two surfaces
General Object Hoffman		surface type (planar, convex,concave) boundary type (jump, normal, crease)	
General Object Blanu	plane	area, perimeter max radius, min radius avg radius, # of vertices angle between max & min radius compactness, centrod neighbors	
Simple Object Oshima Shirai	planar surface quadric surface	surface type, area # of adjacent regions perimeter, compactness mean radius, SD of radius minimum radius maximum radius, occlusion	type of intersection (convex, concave, mixed, none) angle distance
Specific Object Chen Kak	planar surface cylindrical surface conic surface straight curve ellipsoidal curve point	surface type, # of pixels radius, normal, axis, area moment direction point -on-axis	edges types adjacent regions
General Object Fan Medioni Nevatia	region	3D area orientation avg principal curvatures ratio of occlusion centroid	type of adjacency connection possibility angle distance * 3D geometry

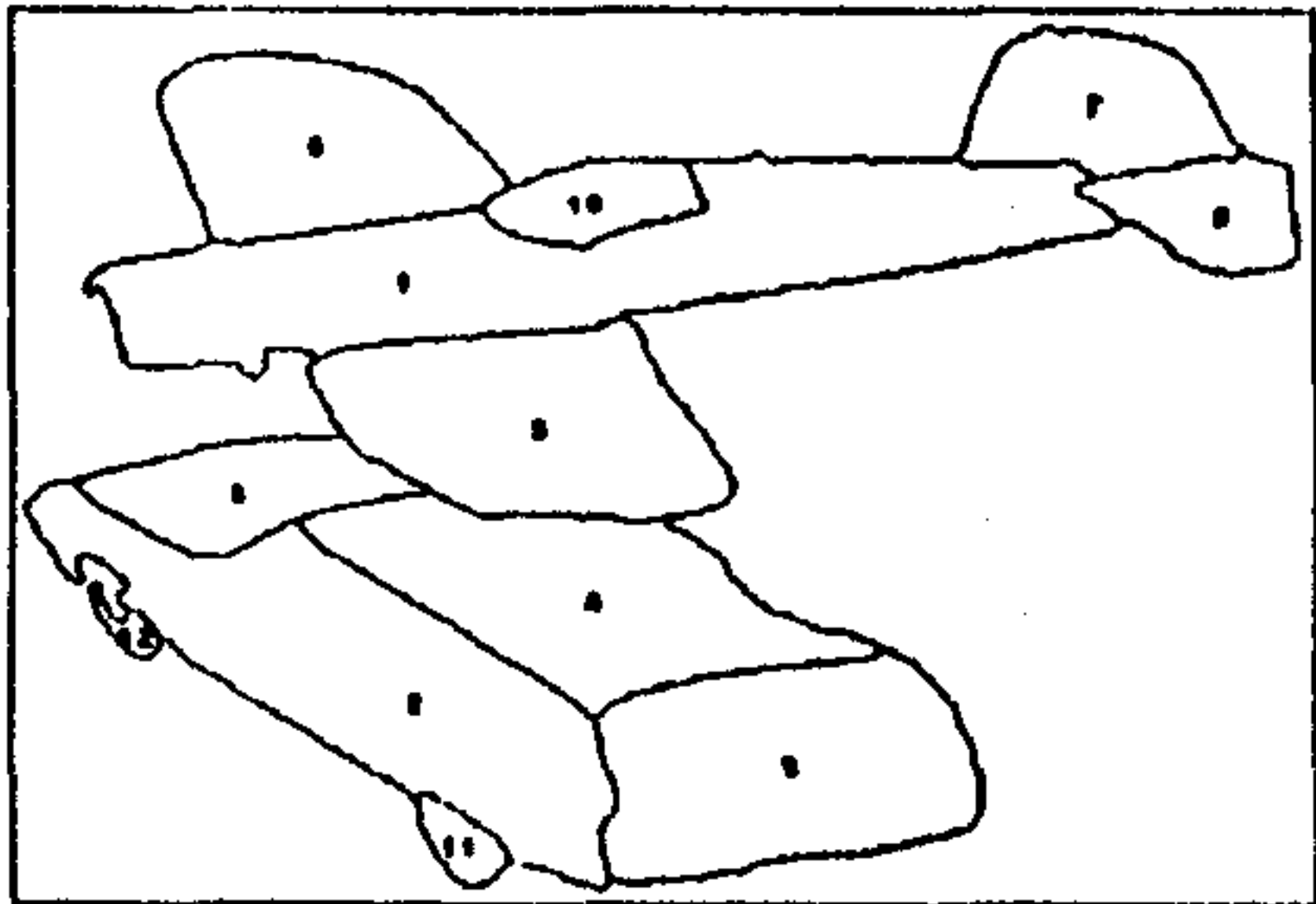
표 4.1: 모양과 관계 속성의 문헌 조사



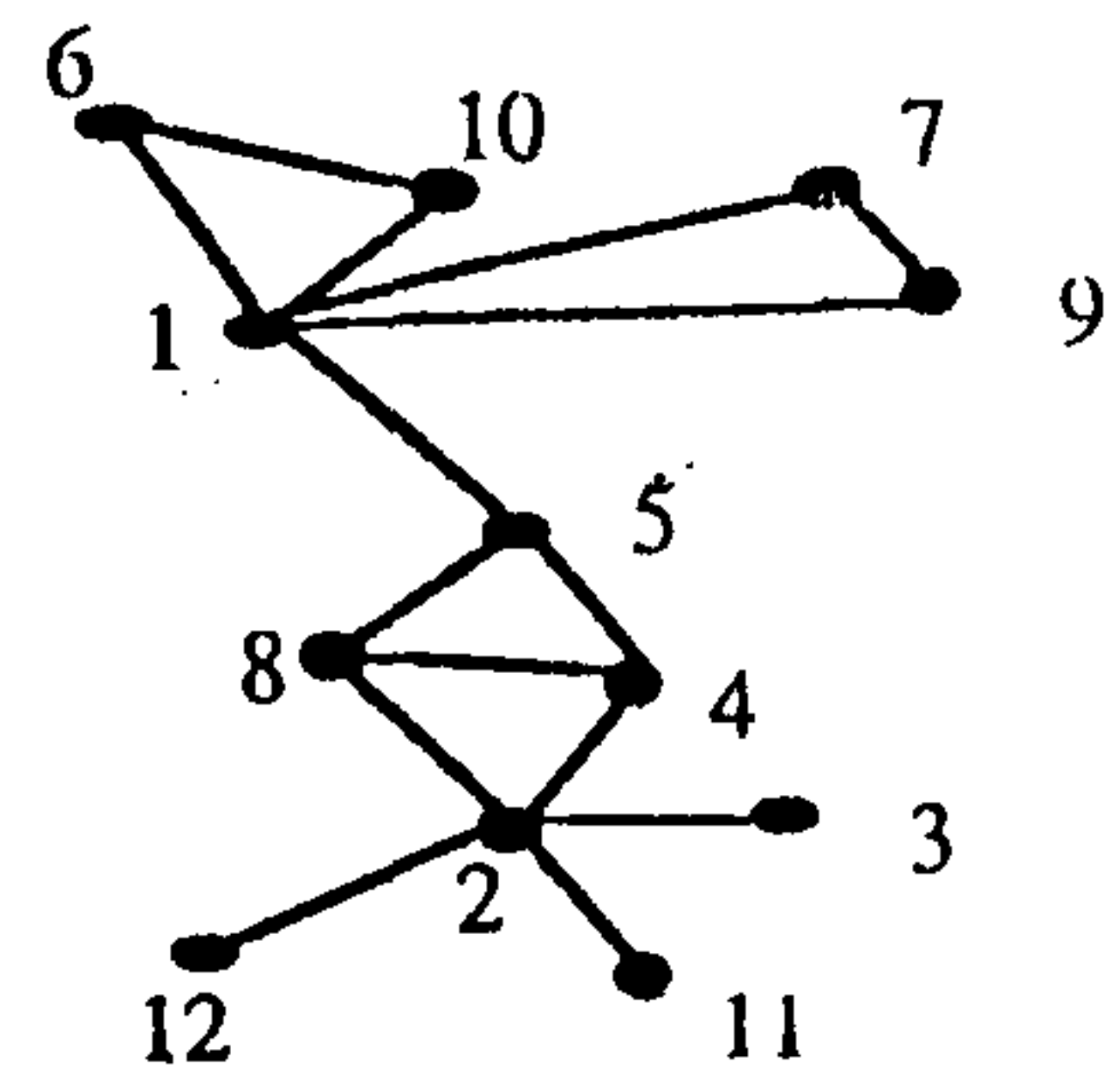
(a) A scene image



(b) Discontinuities



(c) Surface patches



(d) Attributed graph

그림 4.1: Surface Segmentation

제 3 절 정 합

모델 속성 그래프 G_m 과 입력 영상 속성 그래프 G_s 가 주어졌을 때, 인식의 문제는 대응되는 최대 정합, 즉 최대의 부분 집합을 구하는 과정이다. 정합은 G_m 의 한 부분 집합과 G_s 의 부분 집합의 대응 관계이다. 이론적인 subgraph isomorphism 알고리즘은 그 복잡성 때문에 이 문제에 그대로 적용할 수 없다. 이 장에서는 최대 정합 문제를 좀 더 정규적으로 정의하고 순서적 제약 트리 탐색 방법을 이용한 인식 방법을 설명한다. 또한 선행 탐색에 추가한 방법들 - 순서적 제약 트리 탐색 방법과 최적 탐색 방법을 설명한다.

1 문제 정의

본 보고서에서는 SSP는 S 나 S_i 로 표시하고 i 번째 SSP는 S_i 로 표시한다. 마찬가지로 MSP는 M 또는 M_j 로 표시한다. 입력 영상에서 추출한 표면 조각의 수는 n 으로 표시하고, 모델 물체에서 추출한 표면 조각의 수는 m 으로 표시한다. 따라서 입력 영상은 $O_s = \{S_i | i=1, \dots, n\}$ 로 표시되고 모델 물체는 $O_m = \{M_j | j=1, \dots, m\}$ 로 표시된다.

고려하는 물체는 모두 rigid하기 때문에 각각의 SSP는 어떤 모델의 한 MSP가 되어야 한다. 입력 영상의 표면 조각의 부분집합과 한 모델 물체의 표면조각의 부분집합이 서로 대응된다면 모든 SSP를 그에 대응되는 MSP에 옮겨 주는 변환 T 가 존재해야 한다. 최대 정합 문제는 입력 영상에 가장 큰 정합값을주는 모델을 선택하고, 그 SSP와 대응되는 MSP간의 대응관계를 구하고 두 물체사이의 변환 T 를 구하는 문제이다. 일반적으로 입력 영상이 여러 물체로 구성된 경우에는 입력 영상의 부분에 가장 잘 정합하는 부분을 구하고 그 부분을 제외한 나머지 부분을 순환적으로 반복한다. 따라서 정합 문제는 다음의 부분제들을 순환적으로 해결하여야 한다.

1. 모델 라이브러리에서 하나의 모델을 선정한다.
2. 입력 SSP와 MSP와의 대응 관계를 구한다.
3. 변환 T를 구한다.
4. 변환 T를 이용하여 개선된 대응 관계를 구한다.
5. 정합값을 구한다.
6. (1) - (5)를 반복하여 최대의 정합값을 주는 모델을 구한다.
7. 입력 SSP에서 (6)에서 구한 정합의 SSP를 제외하고 (1) - (6)을 입력 SSP가 NULL이 아닌 경우 계속 반복한다.

부문제(1)의 복잡도는 라이브러리의 모델의 수에 비례하지만 노드의 수, 평면 노드의 수, 곡면 노드의 수, 특정 곡면의 보유, 면적 등을 고려하여 이 수를 줄일 수 있다. 본 과제의 실험에서는 이러한 휴리스틱 정보를 포함하지 않고 구현하였다.

부문제(2)의 해결이 핵심적인 부분으로, 일반적으로 입력 SSP와 하나의 MSP가 주어졌을 때, 최대 정합을 찾는 문제는 그림 4.2에 보인 것처럼 깊이 우선 트리 탐색으로 고려할 수 있다. 트리를 traversal하는 것을 정합의 과정으로 간주하여 탐색 트리의 각 arc는 그 레벨에 해당하는 SSP에 한 MSP가 대응될 수 있는 가를 시도해 보는 과정이고, 각 node는 그 노드까지의 부분적인 정합 결과와 변형 함수 (c^* , T^*)를 표현한다. 트리 탐색의 초기에 해당하는 root 노드에서는 ($c^* = \text{nil}$, $T^* = \text{nil}$)의 정보를 가진다. 트리 탐색 과정에서 깊이가 깊어질수록 대응 쌍 c^* 가 증가하고, 변환 함수 T^* 가 더욱 더 구체화 된다. 이러한 트리 탐색이 정한 임계값보다 큰 값을 가지고, 단말 노드까지 도달한다면 그 정합쌍은 해로 간주되고 해당하는 모델은 그 SSP에 해당하는 모델로 간주된다.

일반적으로 한 입력 영상은 여러 물체가 겹쳐져 있거나 영상 분할의 오류로 인해 대응되는 MSP가 없는 SSP가 있을 수 있다. 이러한 경우를 해결하기 위해 그 SSP에 wild

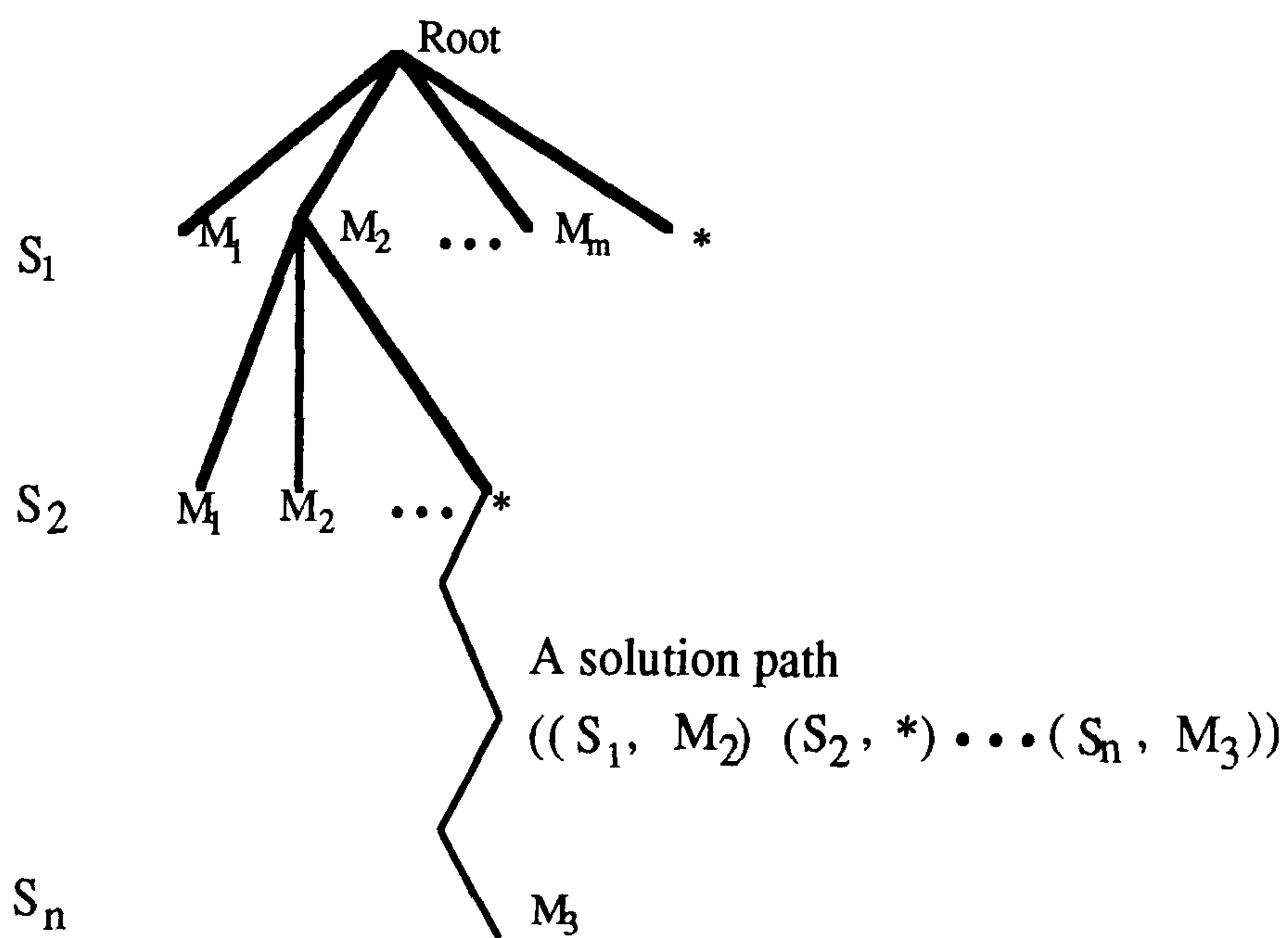


그림 4.2: Search Tree

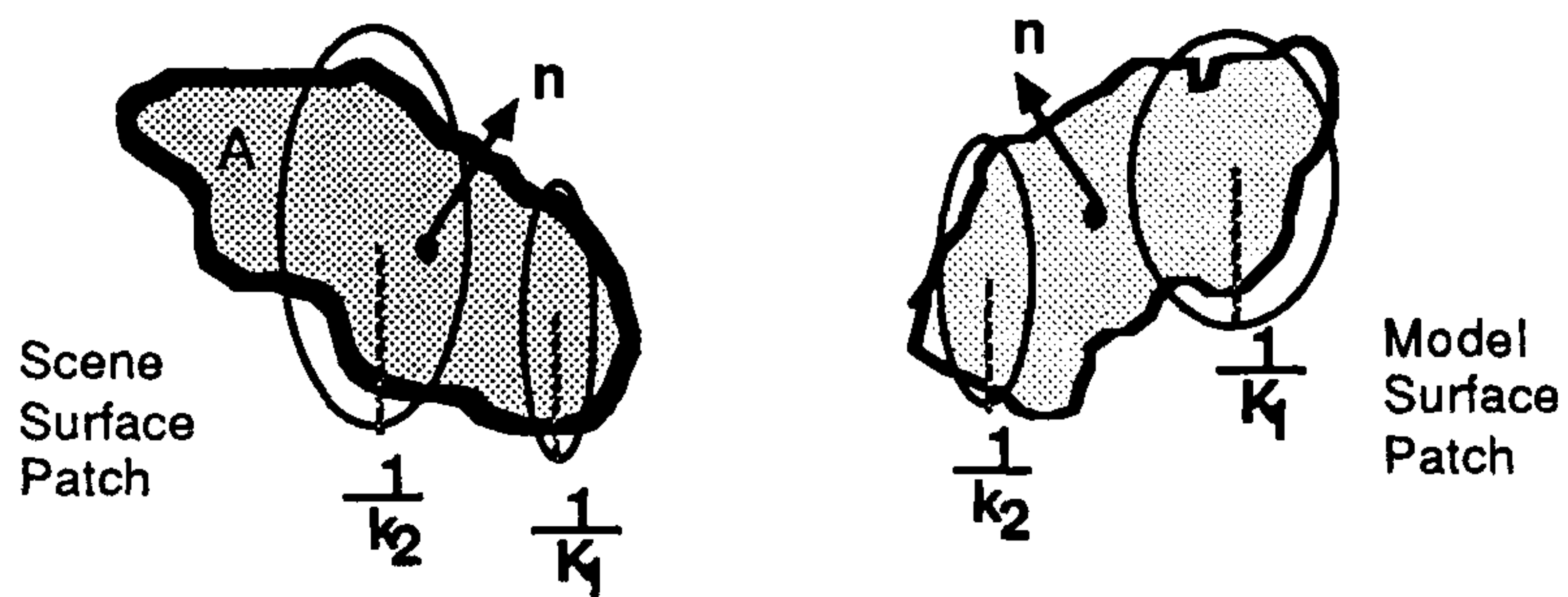
match (*)를 대응시켜 탐색 트리를 확장해 나간다. 그림 4.2에 이 wild match 노드를 * 로 표시하여 해당하는 해의 경로에서 그 SSP를 제외시켜 최대 정합을 구하였다.

2 순서적 제약 트리 탐색

이러한 트리 탐색 과정은 본질적으로 exponential 시간이 소요되므로 물체 인식을 위한 효율적으로 해를 찾기 위해서는 사전에 탐색 트리에서 고려할 필요가 없는 부분을 제거해야 한다. 이러한 사전 제거를 위해서 물체의 모양에 근거한 기하학적인 제약을 사용한다. 본 과제에서는 물체가 표면 조각들로 표현되기 때문에 표면 조각간의 유사성과 표면 조각의 쌍들간의 관계 유사성을 제약으로 사용하였다. 물체간의 겹침 정도와 계산상의 복잡도 등을 고려하여 본 과제에서는 2차, 3차 관계 제약을 사용하였다.

2.1 1 차 제약 (unary constraint)

1차 제약은 하나의 장면 표면 조각(SSP)과 하나의 모델 표면 조각(MSP)의 대응 가능성을 검사하는 것으로 두 표면 조각의 유사성으로 검사한다 (그림 4.3). 두 표면 조각의 유사성은 표면 조각의 속성들을 비교함으로써 검사한다. 본 과제에서는 표면 조각의 3차원 면적 (3-D visible area), 평균 최대 곡률(average maximal curvature), 평균 최소 곡률 (average minimal curvature)을 구한 다음 SSP와 MSP의 값의 차이가 정해진 임계값보다 작은 경우에 1차 제약을 만족한 것으로 간주한다. 3차원 면적은 물체를 정면에서 본 경우의 그 표면 조각의 면적을 의미하는 것으로 현재의 관점과 그 표면조각의 normal vector를 고려하여 계산한다. 이 값은 단순히 영상에서 구한 표면조각의 2차원적 면적보다 관점에 무관한 정보를 제공한다. 모든 점에서의 곡률을 측정하는 방향에



- A : 3-D area
- K_1 : maximum curvature
- k_2 : minimum curvature
- n : normal at center point
- P : perimeter

그림 4.3: Unary Constraint

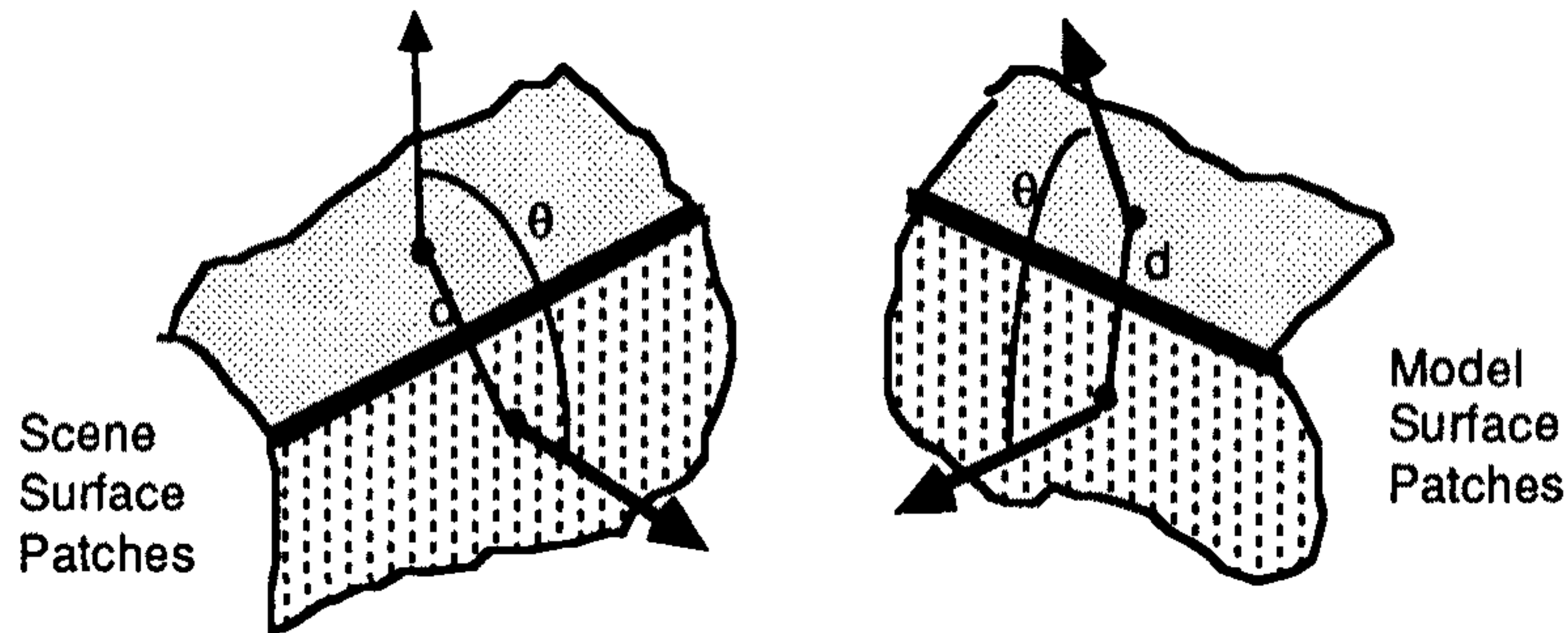


그림 4.4: Binary Constraint

따라 그 값이 변하는 데 가장 큰 값과 가장 작은 값을 최대 혹은 최소 곡률이라한다. 모든 화점(pixel)에서 이 두 값을 구하여 평균한 값을 각각 평균 최대 곡률, 평균 최소 곡률이라 한다. 곡률값은 표면의 특성을 반영하는 주요한 특징점으로 많이 사용되었다.

1차 제약은 표면 조각의 속성들에 의한 제약이므로 다른 물체와의 겹침이나 영상 분할의 오류 등에 영향을 많이 받기 때문에 강한 제약성을 갖지 못한다. 예를 들어 평면인 표면 조각들의 곡률은 모두 0에 가깝고 면적은 비슷한 경우가 많다. 또 심하게 다른 물체에 겹쳐져 있는 경우에는 어떤 SSP 에 정확하게 해당하는 MSP가 그 후보에서 빠질 수 있다.

2.2 2 차 제약 (binary constraint)

1차 제약에 의해 대응 가능한 한 쌍의 SSP와 MSP $\langle S_i, M_i \rangle$ 와 또 하나의 쌍 $\langle S_j, M_j \rangle$ 사이의 관계의 유사성을 검사하는 것이 2차 제약이다 (그림 4.4). 두 SSP

S_i 와 S_j 의 구조적 특성이 두 MSP S_i 와 S_j 의 구조적 특성과 같은지를 검사한다. 본 과제에서는 사용한 구조적 특성은 접속 형태(점프, 오목 크리지, 블록 크리지, 연결되지 않음), 두 normal vector 사이의 각도, 두 중심점 사이의 거리 등을 사용하였다.

- 접속 형태의 일치성: $l(M_i, M_j) = l(S_i, S_j) \quad l(S_1, S_2)$ 는 두 표면 조각 S_1 and S_2 의 접속 형태
- 각도 일치성: $|q_1 - q_2| < q_{tol} \quad q_1(q_2)$ 는 두 표면 조각 M_i and M_j (S_i and S_j)의 normal vector 사이의 각도
- 거리 일치성: $|d_1 - d_2| < d_{tol} \quad d_1(d_2)$ 는 두 표면 조각 M_i and M_j (S_i and S_j)의 중심점 사이의 거리

2.3 3 차 제약 (ternary constraint)

1차 제약과 2차 제약에 의해 대응 가능한 두 쌍 $\langle S_j, M_j \rangle$ 와 $\langle S_k, M_k \rangle$ 이 주어졌을 때 새로운 한 쌍 $\langle S_i, M_i \rangle$ 과의 좀 더 복잡한 형태의 구조적 유사성을 검사하는 과정이다 (그림 4.5). 세 개 SSP S_j, S_k, S_i 구조적 특성과 세 개의 MSP M_j, M_k, M_i 의 구조적 특성의 유사성을 검사하기 위해 본 과제에서는 삼차원 기하학적 특성과 normal vector의 triple product의 부호 등을 사용하였다.

- 삼차원 기하학적 특성 : $|f_1 - f_2| < f_{tol} \quad f_1(f_2)$ 는 $M_i(S_i)$ 의 중심점에서 $M_j(S_j)$ 의 중심점까지의 한 벡터와 $M_j(S_j)$ 의 중심점에서 $M_k(S_k)$ 의 중심점까지의 또 하나의 벡터 사이의 방향성을 고려한 각도이다.
- sign of triple-product : $s_1 = s_2 \quad s_1(s_2)$ 는 M_i, M_j and M_k (S_i, S_j and S_k)의 세 normal vector의 triple-product의 부호이다.

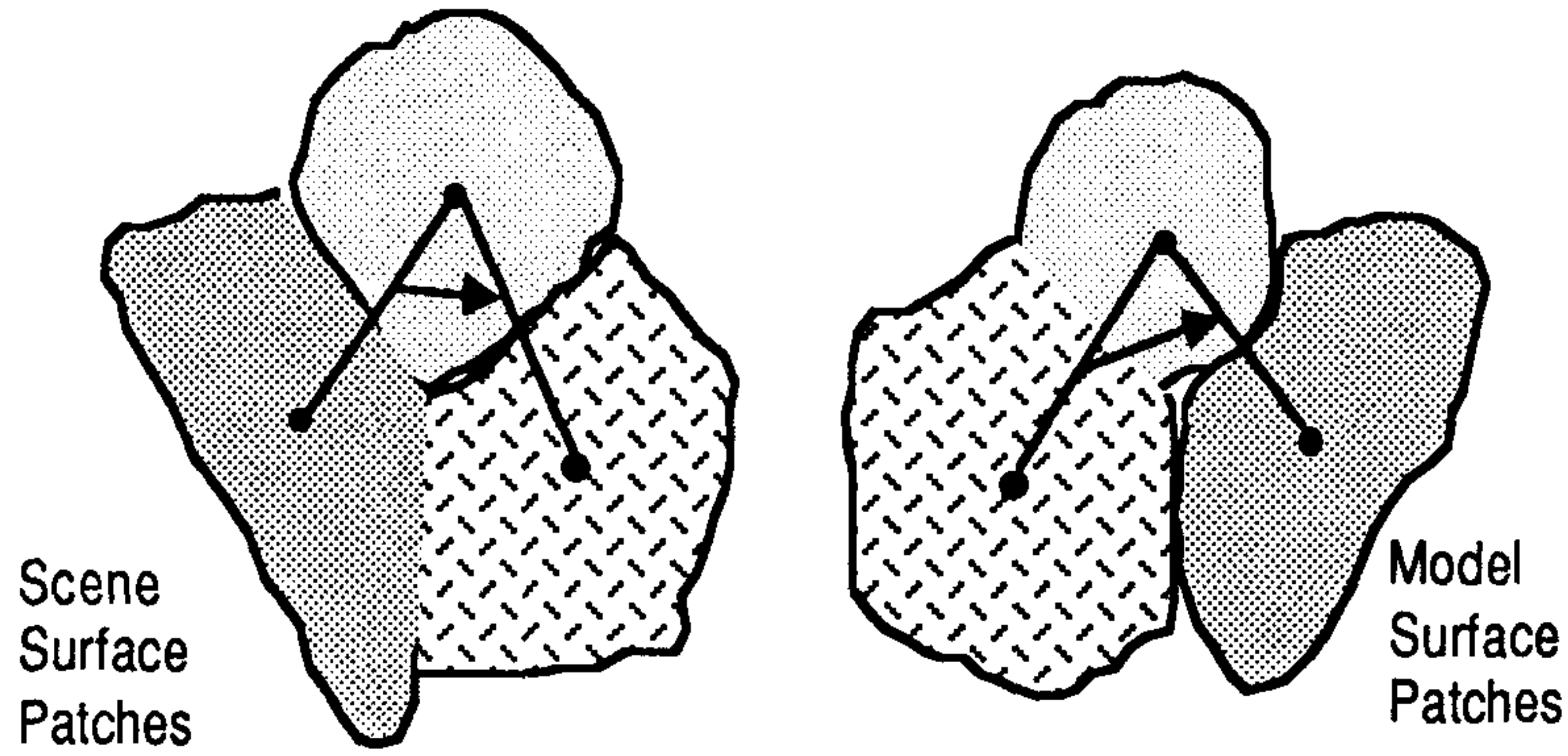


그림 4.5: Ternary Constraint

2.4 총괄적 제약 (global constraint)

1차, 2차 그리고 3차 제약에 의한 정합 쌍은 부분적으로는 옳지만 전체적으로 옳지 않는 경우가 발생할 수 있다. 이러한 부분적인 제약에 의한 애매한 정합 쌍을 전체적으로 옳은가를 검사해야 하는데 이런 목적으로 SSP들과 MSP들 간의 변형 함수의 에러를 사용한다[Bol86, Che89, Fan89, Fau86, Gri84]. 부분적인 제약에 의한 정합쌍이 주어졌을 때, 즉 $((S_1, M_{i_1})(S_2, M_{i_2})(S_3, M_{i_3}) \cdots (S_k, M_{i_k}))$ 의 형태로 SSP와 MSP 간의 대응이 주어졌을 때 각 표면 조각의 방향과 위치 벡터를 구하여 모델 물체를 장면 물체로의 삼차원상의 변형을 해주는 변형 함수 T 를 계산한다. 이 때 발생한 변형 함수의 오차를 총괄적 제약으로 사용한다.

$$\text{총괄적 제약} : S \|V_i - T(V_i)\|^2 < e_{tol}$$

$V_i(V_i)$ 는 MSP(SSP)의 위치나 방향 벡터를 나타내고 T 는 계

산된 변형 함수이다.

1차, 2차, 3차 제약들을 이용한 순서적 제약 트리 탐색 알고리즘(sequential constrained tree search: SCTS)은 다음과 같다. SCTS는 깊이 우선 트리 탐색의 방법을 택하고 새로운 노드의 생성시에는 지금까지 구한 부분적인 정합 쌍과 새로운 한 쌍과의 제약이 일치하는 경우에만 새로운 노드를 생성하는 방법이다. 효율적인 제약들을 사용하면 exponential 시간의 비효율적인 트리 탐색의 단점을 극복할 수 있다.

현재까지의 레벨 k 에서의 부분적인 해를 $P_k = ((S_1, M_{j_1})(S_2, M_{j_2})(S_3, M_{j_3}) \cdots (S_k, M_{j_k}))$ 라 하자. S_k 는 k 레벨에 해당하는 SSP이고, M_{j_k} 는 그 SSP에 대응되는 MSP이고 이 값은 실제 한 모델의 MSP일 수 있고, wild match인 경우도 있다. 레벨 $k+1$ 에서 새로운 노드를 생성하는 경우 그에 해당하는 SSP가 S_{k+1} 라고 할 때 그에 대응되는 MSP $M_{j_{k+1}}$ (j_{k+1} 은 1에서 $m+1$ 까지 변할 수 있다.)를 이용하여 구한 새로운 해는 다음과 같다.

$$P_{k+1} = ((S_1, M_{j_1})(S_2, M_{j_2})(S_3, M_{j_3}) \cdots (S_k, M_{j_k})(S_{k+1}, M_{j_{k+1}}))$$

이 새로운 해는 다음의 제약들을 만족하는 경우에만 새로운 노드를 생성하고 그 노드의 후손들(descendant node)을 순환적으로 생성한다.

- $M_{j_{k+1}}$ 가 wild match이면, 새로운 해 P_{k+1} 는 허용할 수 있는 것으로 간주하고 탐색을 계속한다.
- $M_{j_{k+1}}$ 가 실제의 한 MSP이면, S_{k+1} 와 $M_{j_{k+1}}$ 과의 1차 제약을 만족하는지를 검사한다. 지금까지 구한 부분적인 해의 각 쌍과의 2차 제약을 검사한다. 그리고 지금까지 구한 해의 모든 가능한 두 쌍에 대해 3차 제약이 만족되는지를 검사한다.
- 모든 제약들이 모두 만족되면 새로운 해 P_{k+1} 는 확장된 해로 인정되고 탐색을 계속한다. 적어도 하나의 제약이 만족되지 않으면 그 해를 버리고 새로운 MSP를 고

려한다. 순환적으로 탐색을 계속한 결과 단말 노드(terminal node)에 도달하면 그해의 정합값(matching value)을 계산하고 백트랙한다. 현재의 노드에서 모든 자손 노드 (child node)를 다 탐색하여 그 중에서 가장 좋은 정합값을 가지는 자손과 그 정합값을 부모 노드(parent node)에 전달한다. 순환적 탐색(recursive search)이 종료하면 근원 노드 (root node)는 최종 정합값과 해의 경로를 가진다.

3 순서적 선행 제약 트리 탐색 (sequential forward checking constrained tree search)

순서적 제약 트리 탐색은 근본적으로 모든 가능한 경로를 탐색하고 그 중에서 가장 좋은 경로를 고르는 방법으로 단일 물체만 있는 영상을 인식하는 데는 효율적이지만 여러 물체가 겹쳐 있는 일반적인 상황에서는 비효율적이다. 본 절에서는 효율적인 탐색 방법의 하나로 선행 검사 기능을 가진 순서적 제약 트리 탐색 방법을 설명한다(SFCCTS). 선행 검사란 탐색 중간에서 그때까지 생성된 부분적인 해를 가지고 아직 처리하지 않은 쌍과의 기하학적인 제약등을 검사하여 후보의 수를 줄이는 방법이다. 이러한 기하학적인 제약의 검사는 1차, 2차, 3차, 총체적 제약을 차례로 적용하여 불일치하는 쌍을 미리 제거한다.

탐색이 레벨 k 에 해당하는 노드에 도달하여 그때까지의 부분적인 해가

$$P_k = ((S_1, M_{j_1})(S_2, M_{j_2})(S_3, M_{j_3}) \cdots (S_k, M_{j_k}))$$

일때, 그 노드보다 하부의 레벨에 해당하는 SSP 인 $S_l, l = k+1, \dots, n$ 들에 대해 1차 제약과 k 보다 낮은 레벨에서 적용한 제약 심사에서 살아 남은 후보 들을 k 번째 추가된 쌍

을 사용하여 심사하여 불일치하는 후보들을 제거한다.

1 번째 레벨의 그 전과정에서 살아남은 후보 MSP M_{j_l} 에 대해 첫째 S_l 와 M_{j_l} 가 1차 제약을 만족하는 지를 검사한다. 그리고 k 번째 추가된 쌍 (S_k, M_{j_k}) 과 현재 고려하는 쌍 (S_l, M_{j_l}) 사이의 2차 제약을 만족하는가를 검사한다. k 번째 추가된 쌍과 경로상의 모든 쌍 $[(S_l, M_{j_l})(S_k, M_{j_k})(S_i, M_{j_i})]$ for $1 \leq i \leq k - 1$ 과 현재 고려하는 쌍 (S_l, M_{j_l}) 사이의 3차 제약을 만족하는가를 검사한다. 이 제약 중에 하나라도 불만족하면 SSP S_l 에 대한 후보를 제외시킨다. 이런 과정을 S_l 의 모든 후보에 대해 적용시켜 만족하는 후보만을 남긴다.

처음 탐색을 시작할 때는 구한 해가 NULL 이므로 모든 SSP $S_l, l = 1, \dots, n$ 는 1차 제약에 의해서만 후보의 수가 줄 수 있지만 그 깊이가 증가할수록 제약하는 힘이 커져 살아 남는 후보의 수는 매우 적어진다. 레벨 1에 해당하는 SSP S_1 는 wild match node까지 포함하여 $(m+1)$ 개의 형제 노드(sibling)을 가지지만 깊이가 깊어질수록 형제 노드의 수는 심하게 줄어든다. 시작 노드 (root node)에서 현재 고려하는 노드까지의 해의 경로에서 적어도 하나가 wild match node가 아니면 그 하부 노드에서는 wild match node가 생성될 수 없다. 물체의 인식에서 가장 정합이 가능한 SSP와 MSP의 한 쌍은 휴리스틱하게 찾을 수 있기 때문에 이 선행 기능을 가진 제약 트리 탐색 방법은 wild match node가 없이 탐색 트리를 생성할 수 있다.

그림 4.6에 SFCCTS을 기술한다. 탐색 시작에 모든 SSP에 1차 제약에 의한 후보 MSP들을 구하고 배열 D에 저장한다. 함수 SFCCTS(match, level, candidates)는 첫 번째 인자가 현재까지의 구해진 해를, 두 번째 인자가 현재 고려하는 노드의 레벨, 세 번째 인자가 모든 SSP에 대한 후보 MSP를 나타내고 함수값은 그 레벨이하의 부트리에서의 최종 정합값과 그에 대응되는 해의 부분을 넘겨 준다. 탐색 시작에는 SFCCTS(NULL, 0, D)를 호출하면 순환적으로 모든 부트리를 생성하여 최종적으로 정합값과 해를 넘겨 준다.

```

MAIN ()
1   for i:= 1 to n do
2      $D_i :=$  candidate MSP's matchable to ith SSP by unary constraints
3   end for
4   SFCCTS(nil, I, D)

SFCCTS (old-match, level, candidates)
1   i := level; best-value :=0
2   if (i ≤ n) then
3     begin
4       while there are elements in  $D_i$  then
5         begin
6            $M_i :=$  pop( $D_i$ )
7           new-match := old-match U ( $S_i, M_i$ )
8           if transform-consistent (new-match) then
9             begin
10              next-level := ∞
11              #-alive := 0
12              for k :=(i+1) to n do                                /* forward check */
13                begin
14                  push ( $D_k$ , stack)
15                  remove elements from  $D_k$  which are inconsistent with new-match
16                  if not null( $D_k$ ) then
17                    begin
18                      #-alive := #-alive +1
19                      if #-alive == 1 then next-level = k
20                    end if
21                  end if
22                end for
23              if |new-match| + #-alive > (max |best-match| ,  $\alpha * n$ ) then
24                begin
25                  (value, match) :=SCTSWFC(new-match, next-level, D)
26                end if
27              for k :=n downto (i+1) do
28                begin
29                   $D_k :=$  pop(stack)
30                end for
31              if best-value < value then
32                begin
33                  best-value := value
34                  best-match := new-match
35                end if
36              end if
37            end while
38          end if
39        if best-match is assigned then
40          return ( best-match, best-value)
41        else return (old-match, calculate-match-value(old-match))

```

그림 4.6: SFCCTS

현재의 노드에서 생성된 해가 1차, 2차, 3차 제약과 총괄적 제약을 검사하는 transform-consistent 함수를 만족하면 (line 8), 선행 검사가 아직 처리 하지 않은 SSP들의 후보 MSP들에 대해 적용된다 (line 10-22). 선행 검사는 단순히 후보 MSP의 수를 줄일 뿐만 아니라 적어도 하나의 후보를 가지는 아직 처리하지 않은 SSP의 수를 계산한다. 그림 4.6에서는 이 수를 #-alive 라고 정의하였고 이 수는 탐색을 종료하는 criterion을 산출하는 데 사용된다. 이 수는 모든 하부 트리를 다 생성하고 난 후의 실제 SSP의 수에 근사할 수록 좋은 탐색 종료 criterion을 제공할 수 있다. 또 선행 검사 과정에서 현재 보다 깊은 레벨에서 적어도 하나의 후보를 가지는 SSP에 해당하는 레벨의 값을 구한다. 이 값은 다음 레벨을 고려할 때 순서적으로 하나씩 증가하지 않고 중간에 wild match node에 해당하는 레벨을 통과하기 위해 사용된다. 즉 SFCCTS에서는 생성되는 레벨이 초기에는 1 부터 시작하여 능동적으로 그 값을 변화 시킨다.

현재의 노드에서 선행검사에서 구한 두 가지 정보를 사용하여 탐색 종료 criterion을 통과하면 순환적으로 SFCCTS(new-match, next level, modified D) 를 부른다. (line 25). new-match 는 새로운 SSP와 MSP의 쌍을 추가한 해이고, next level은 적어도 하나의 후보를 가지는 SSP에 해당하는 레벨값이다. modified D는 새로운 쌍이 추가됨으로써 제약 검사에 의해 남은 후보들을 저장하고 있다. 탐색 종료 검사 기준은 다음의 식을 사용한다(line23-26). 이 기준을 통과하는 경우에만 최적의 해를 구할 가능성이 이 있기 때문에 만족하지 않는 경우에는 탐색을 종료하는 것이 효율적이다. current-match는 현재 노드까지의 해를 의미하고 , best-match는 지금까지 구한 해 중에서 가장 좋은 해를 의미한다.

$$|current - match| + \# - alive > Max(|best - match|, a * n)$$

탐색이 단말 노드에 도달하면 그 해의 크기가 best-match보다 큰 경우에는 best-match 를 그 해로 대치하므로 이 조건은 항상 최적의 해를 제공한다. 초기에 변수 a 의

값을 주면 적어도 $a * n$ 의 크기를 가지는 해를 발견하는 것이다. 즉, 모델 물체의 얼마만큼 이상이 해에 포함되어야 하는가를 평가하는 임계값이다. 이 값이 크다면 탐색 시간을 줄일 수 있지만, 심하게 겹쳐 있는 경우는 해를 찾기 불가능해진다. 어떤 레벨에서 그 레벨의 SSP에 대응되는 후보 MSP가 다른 것으로 바뀌면 즉 형제노드가 생성되면 그 전 MSP에 의해 조정된 것을 복구해야 한다 (lien 27-30). 여러 모델 중에서 가장 잘 정합되는 것을 고르거나 현재 탐색 중인 트리에서 가장 정합값이 좋은 경로를 찾기 위해서 해를 주어진다면 즉 SSP들과 대응되는 MSP들이 주어진다면 그 정합값을 계산해주는 함수가 필요하다. 본 과제에서는 변형 함수의 오차, 각 표면조각의 속성의 차, 해의 크기 등을 고려한 함수를 정의하였다. 그림 4.6에서는 calculate-match-value ()가 그 값을 계산하는 함수이다 (line 41).

$$w_1x(\exists/!6cf) + w_2Sd(S_i, M_i) + w_3n/|matchingpairs|$$

여기에서 $d(S_i, M_i)$ 는 표면 조각의 속성의 차이를 구하는 함수.

각각의 노드에서 그 자손 노드들의 정합값을 계산하고 대응되는 해를 그 노드의 부모 노드에게 전달한다. 최종적으로 시작 노드는 최종 정합값과 해를 구할 수 있다.

4 최적 선행 제약 트리 탐색

SFCCTS는 순차적으로 경로를 생성해야 한다. 후보 경로의 수를 최대한 줄이기 위해서 물체 고유의 제약등을 사용한 제약 트리 탐색 방법을 앞 절에서 설명하였다. 이 절에서는 비용함수를 사용하여 최적 우선 탐색을 하는 A^* 휴리스틱 알고리즘을 도입한 최적 제약 트리 탐색 방법을 고안하였다. 비용함수 $f(x)$ 는 현재까지의 해에 대한 비용 $g(x)$ 와 아직 처리 하지 않은 부분에 대한 추정 비용 $h(x)$ 로 구성되는 되는 데 효율적인 탐색

이 되기 위해서는 이 추정 비용이 얼마나 실제 비용을 잘 추정하느냐에 달려있다.

$$f(x) = g(x) + h(x)$$

$$g(x) = \text{Terror}(x) + \text{Uerror}(x) + n / | \text{current match} |$$

$$h(x) = \text{ETerror}(x) + \text{EUerror}(x) + n / \# \text{-alive}$$

여기에서 $\text{Terror}(x)$ 는 현재까지 구한 해의 변형 함수의 오류,

$\text{Uerror}(x)$ 는 해의 SSP들과 MSP들간의 속성의 차이,

$\text{ETerror}(x)$ 는 아직 처리하지 않은 부분의 추정 변형 함수의 오류

$\text{EUerror}(x)$ 는 아직 처리하지 않은 부분의 SSP들과 MSP들간의 속성의 추정된 차이

$\# \text{-alive}$ 는 선행 검사에서 계산한 적어도 하나의 후보를 가지는 SSP의 수.

$\text{Terror}(x)$, $\text{Uerror}(x)$ 와 $\# \text{-alive}$ 는 현재의 노드 x 에서 계산할 수 있지만, $\text{ETerror}(x)$ 와 $\text{EUerror}(x)$ 는 계산하기 힘들거나 매우 계산하기에 비용이 많이 든다. 본 과제에서는 단순히 $\text{ETerror}(x) = \text{EUerror}(x) = 0.0$ 라 추정하였다. 따라서 추정 비용 $h(x)$ 는 $h(x) = n / \# \text{-alive}$. 아직 처리하지 않은 부분에서 발견할 수 있는 SSP의 수는 $\# \text{-alive}$ 보다 작기 때문에 $h(x)$ 는 최적 비용 $h^*(x)$ 보다 작거나 같다. 따라서 휴리스틱 함수 $h(x)$ 는 admissible하다. 즉 최적의 해가 있는 경우에 그것을 찾을 수 있다. 자세한 알고리즘의 기술은 그림 4.7에 있다. OFCCTS는 전형적인 A^* 알고리즘으로 OPEN, CLOSE의 자료 구조를 가지고 다음과 같은 기능을 반복한다.

1. OPEN에서 최소 비용을 가지는 노드 x 를 가져와서
2. 그 노드의 자손 노드 x' 들을 확장한다.
3. 모든 자손 노드들에 대해 비용함수 $f(x')$ 를 계산한다.
4. 모든 자손 노드들을 OPEN 자료구조에 삽입한다.

```

OFCCTS()
1  for i:= 1 to n do
2    Di := candidate MSP's matchable to ith SSP by unary constraints
3  end for
4  open := { (nil, 1, 0.0, D)}      /* (match, level, value, candidates) */
5  while not null(open) do
6  begin
7    nn := get-minimum-node (open)
8    insert-close (nn, close)
9    old-match := pop(nn); i := pop(nn); best-value := pop(nn)
10   if (i > n) then return(old-match, best-value)
11   D := pop (nn)
12   for every M in Di do
13   begin
14     TD := copy (D)
15     new-match := old-match U (Si, M)
16     if transform-consistent (new-match) then
17     begin
18       gvalue := calculate-gvalue (new-match)
19       next-level := ∞; #-alive := 0
20       for k :=(i+1) to n do          /* forward check */
21       begin
22         remove elements from TDk which are inconsistent with new-match
23         if not null(TDk) then
24         begin
25           #-alive := #-alive +1
26           if #-alive == 1 then next-level = k
27         end if
28       end for
29       hvalue := calculate-hvalue(#-alive)
30       insert-open ({new-match, next-level, gvalue + hvalue, TD})
31     end if
32   end for
33 end while

```

그림 4.7: OFCCTS

5. (1)-(4) 과정을 목적 노드를 발견할 때까지 반복한다. 목적 노드는 레벨이 n 인 노드이다.

OPEN 이나 CLOSE의 각 요소는 네 가지 성분으로 구성된다. 첫째는 현재까지의 해, 둘째는 next-level, 셋째는 비용함수값 넷째는변경된 후보들이다. 원칙적으로 선행 기능 검사는 SFCCTS와 동일하고 구하는 정보도 동일하다. 탐색 시작에는, 1차 제약을 사용하여 각 SSP에 대응되는 모든 후보 MSP 를 구하고 그것들을 배열 D에 저장한다. 시작 노드 s 를 (nil, 1, 0.0, D)로 정의하고 노드 s 를 OPEN 자료 구조에 삽입한다 (line 4). 모든 자손 노드들을 생성하고 각 자손 노드에서 그 이하의 SSP에 대응될 수 있는 후보 MSP의 수를 선행 검사를 통해 줄인다. 이때 적어도 하나의 후보를 가지는 MSP의 수와 적어도 하나의 후보를 가지는 가장 가까운 SSP에 해당하는 레벨을 구한다. 이 값을 이용하여 그 자손 노드의 비용함수 $f(x')$ 을 계산한다. 이 자손 노드들을 OPEN 자료구조에 추가한다. OPEN 자료 구조에 적어도 하나의 원소가 있는 동안에는 line 5-33을 목적 노드가 발견될 때까지 (line 10) 반복 수행한다.

제 3 절 실험 결과

본 절에서는 실제 거리 영상을 사용한 실험 결과와 정합 방법들을 비교 분석한다. 이 알고리즘은 LISP 언어로 구현하였고 초 단위로 인식 시간을 계산하였다.

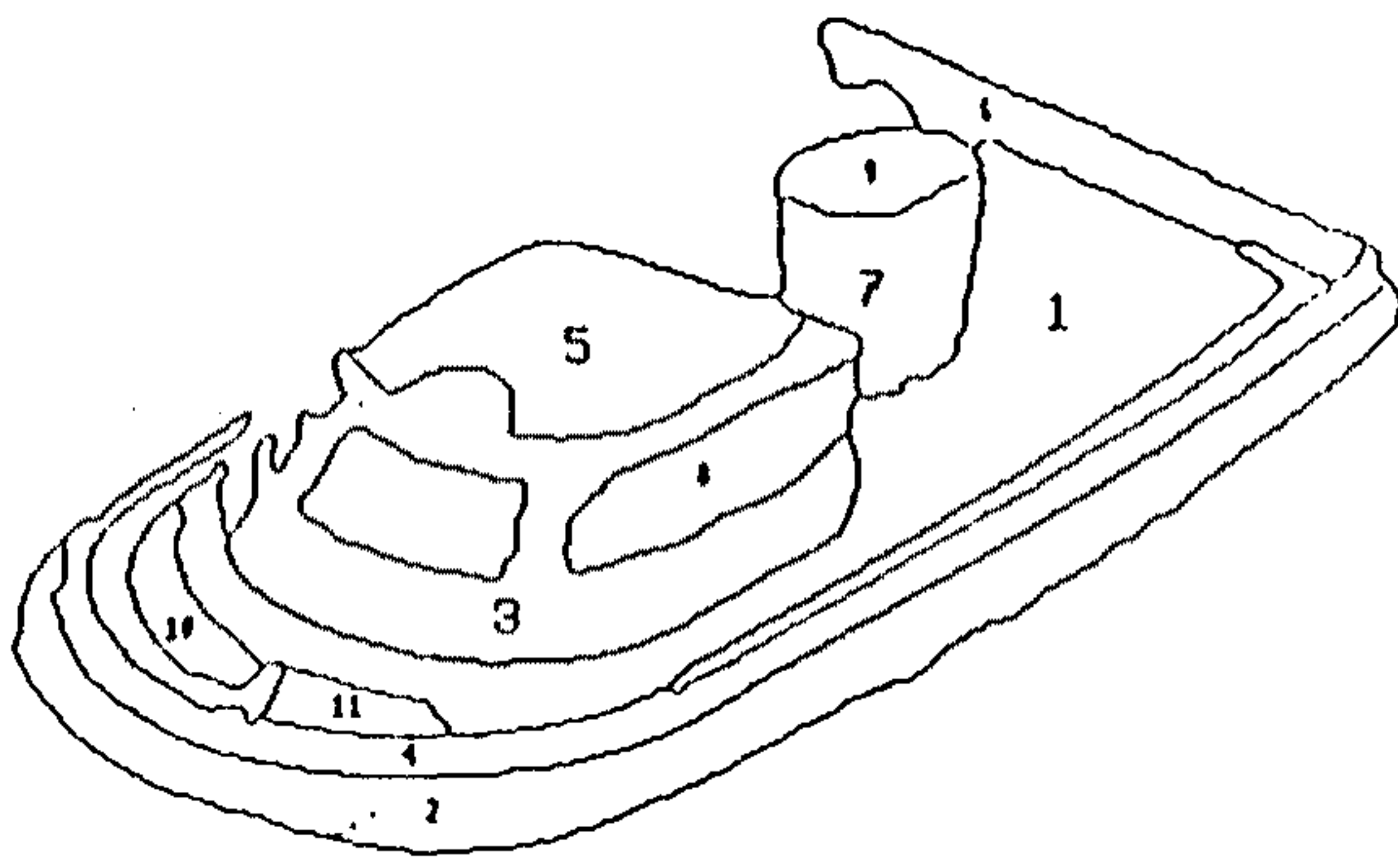
1 모델

본 실험에서는 10개의 물체를 사용하여 30개의 모델을 생성하였다. 10개의 물체를 생성된 장면의 수와 같이 표시하면 다음과 같다. - boat(2 views), car(4) , chair(4), telephone(4), table (2), face (6), hatchback car (2), wagon(2), ship(2) 그리고 airplane (2). 그림 4.8에 이러한 물체의 한 장면씩을 영상 분할된 형태로 보여준다. 서론에서 설명한 것처럼, 본 과제에서는 모델을 생성하기 위해 각 물체를 여러 특징 장면을 사용하는 방법을 사용하였다. 따라서, 복잡한 물체는 여러 개의 장면을 가지고 단순한 물체는 적은 수의 장면으로 표현할 수 있다.

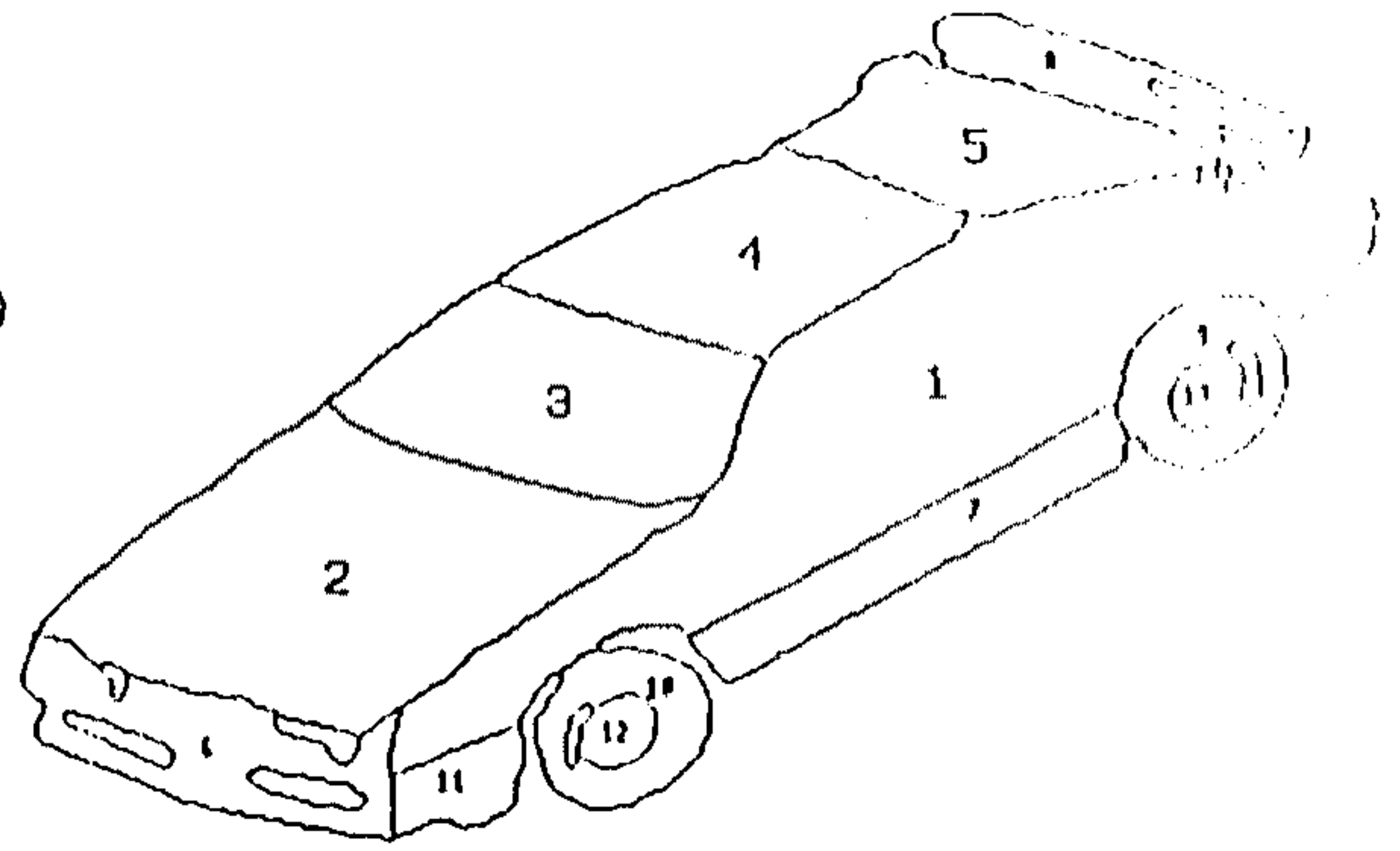
정합 과정에서의 시간을 줄이기 위해 모델 생성 과정에서 표면 조각과 표면 조각 간의 관계 정보를 미리 계산하여 저장하여 사용한다.

2 입력 영상

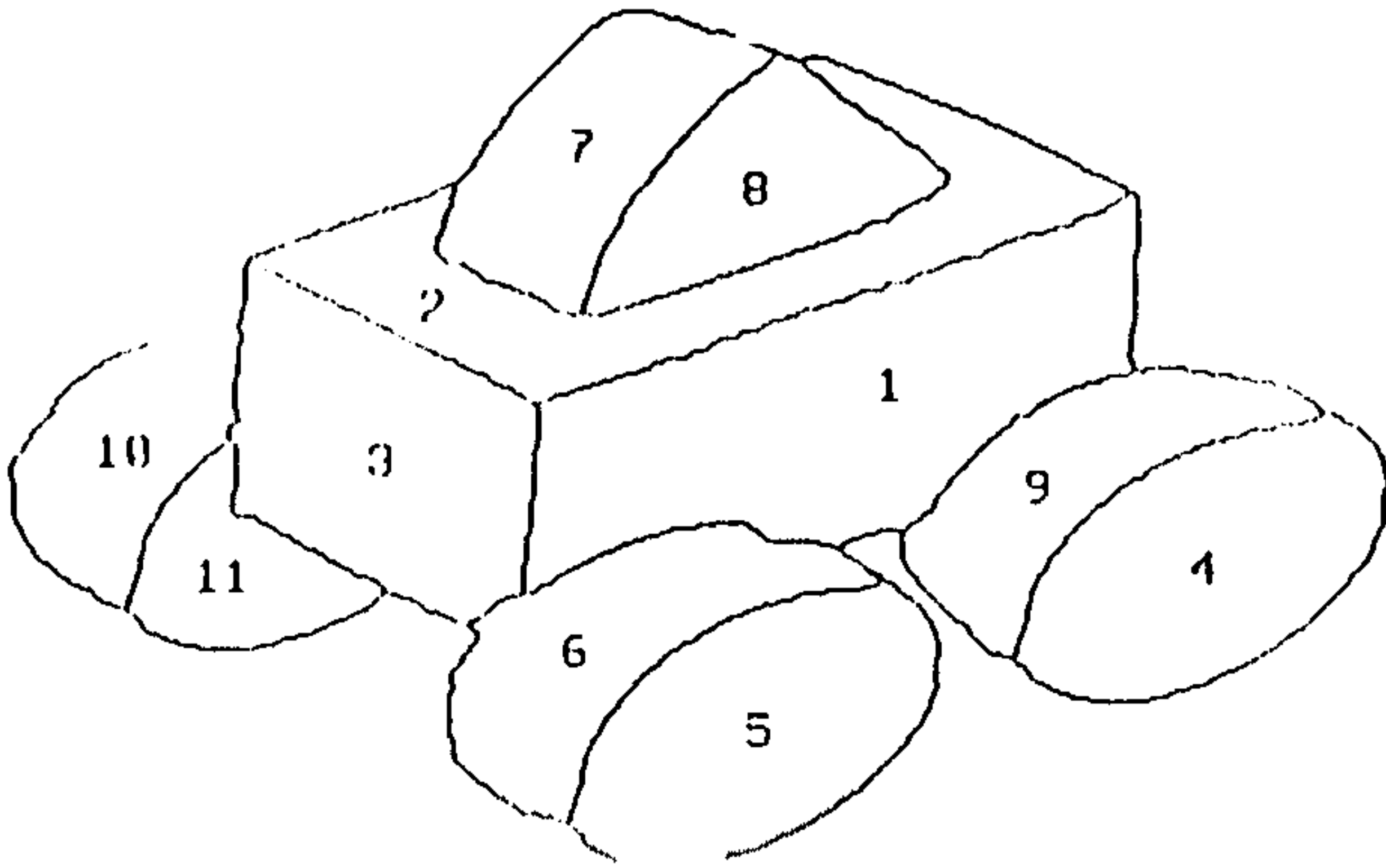
거리 영상은 레이저와 카메라를 이용하여 삼각형 법칙을 사용하여 구한 실제 거리 영상을 사용하였다. 이 영상은 Fan 등이 구한 영상으로 참고 문헌[Fan87]에 잘 설명되어



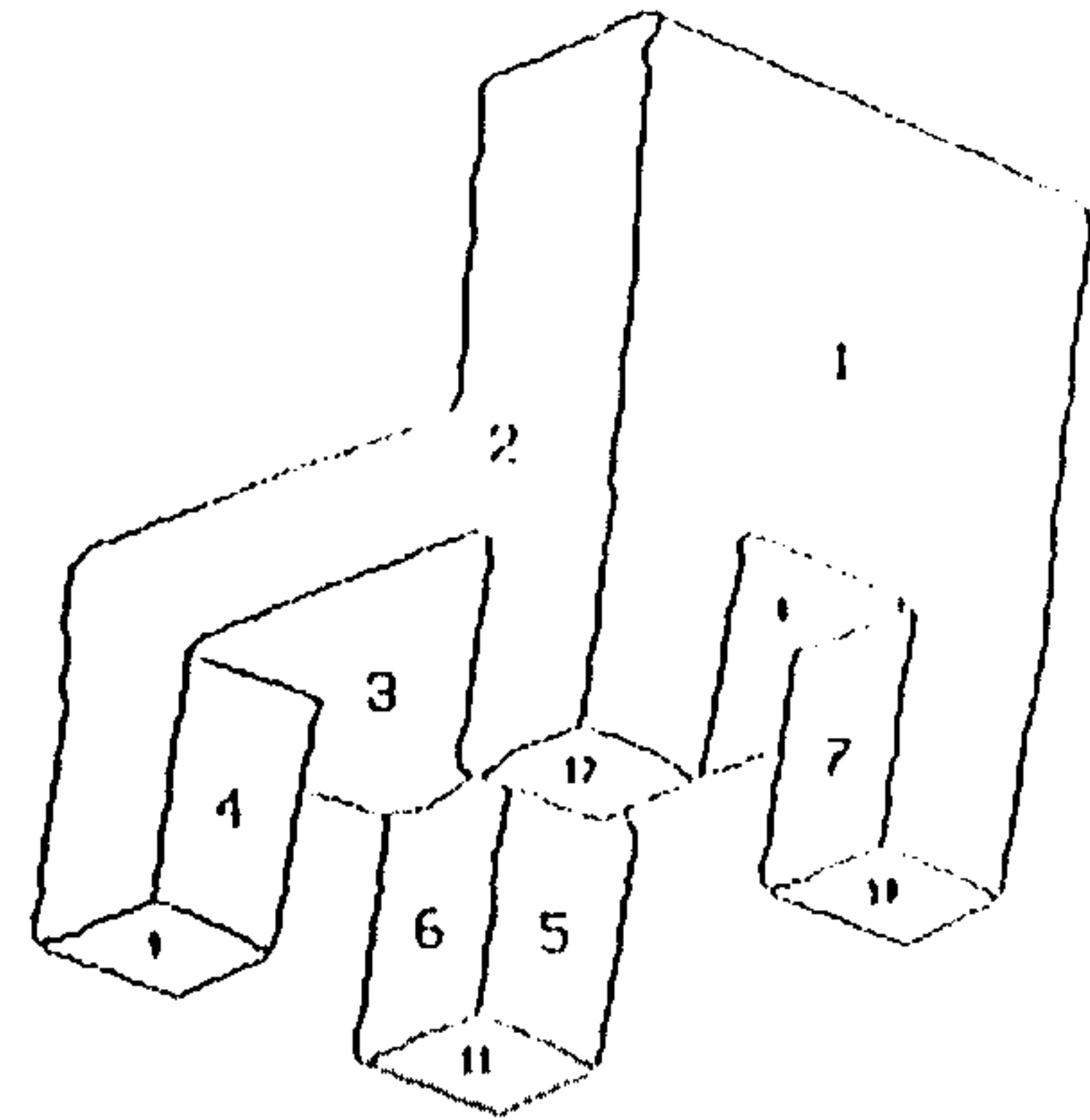
(a) a model boat



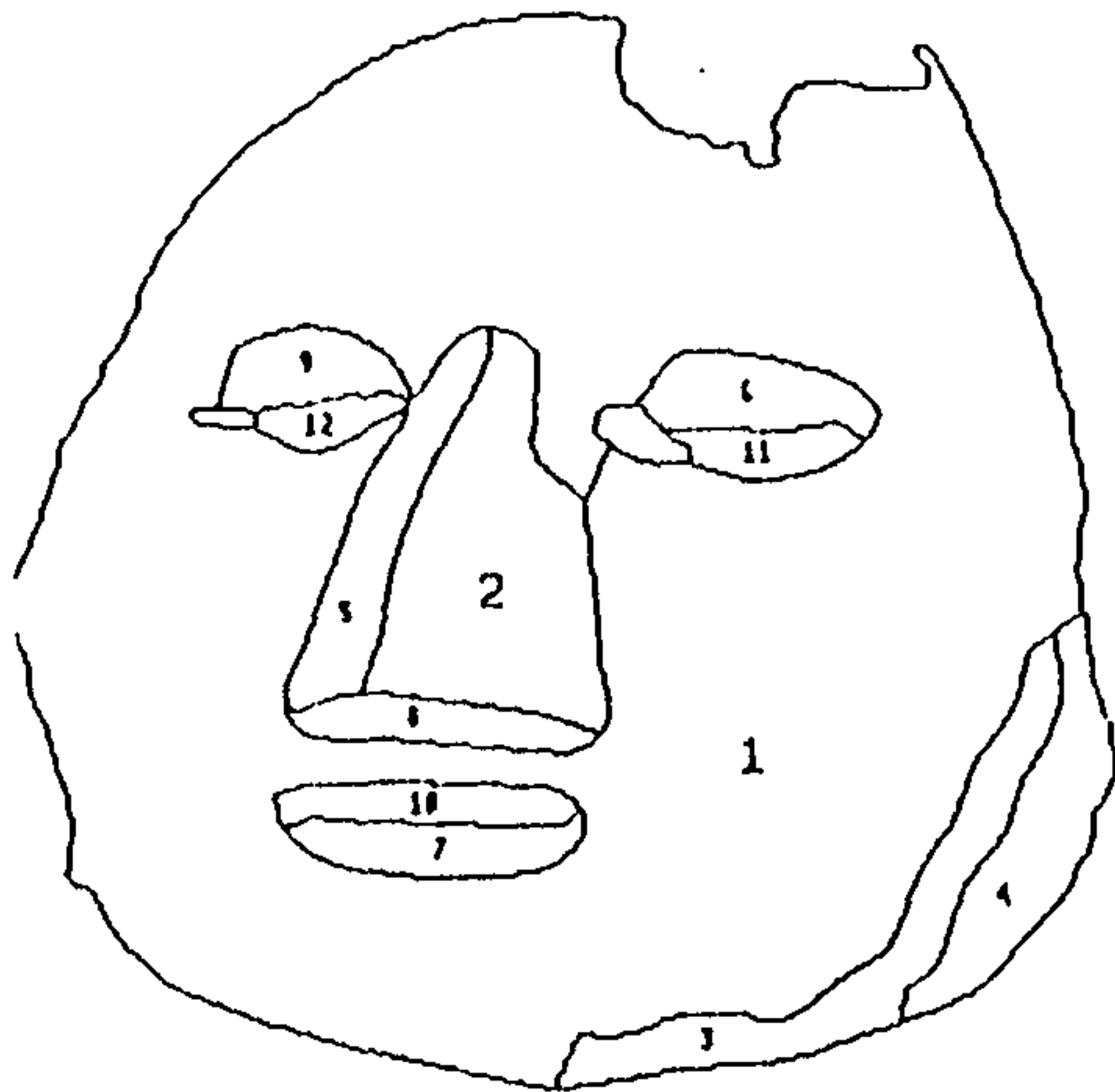
(b) a model cab



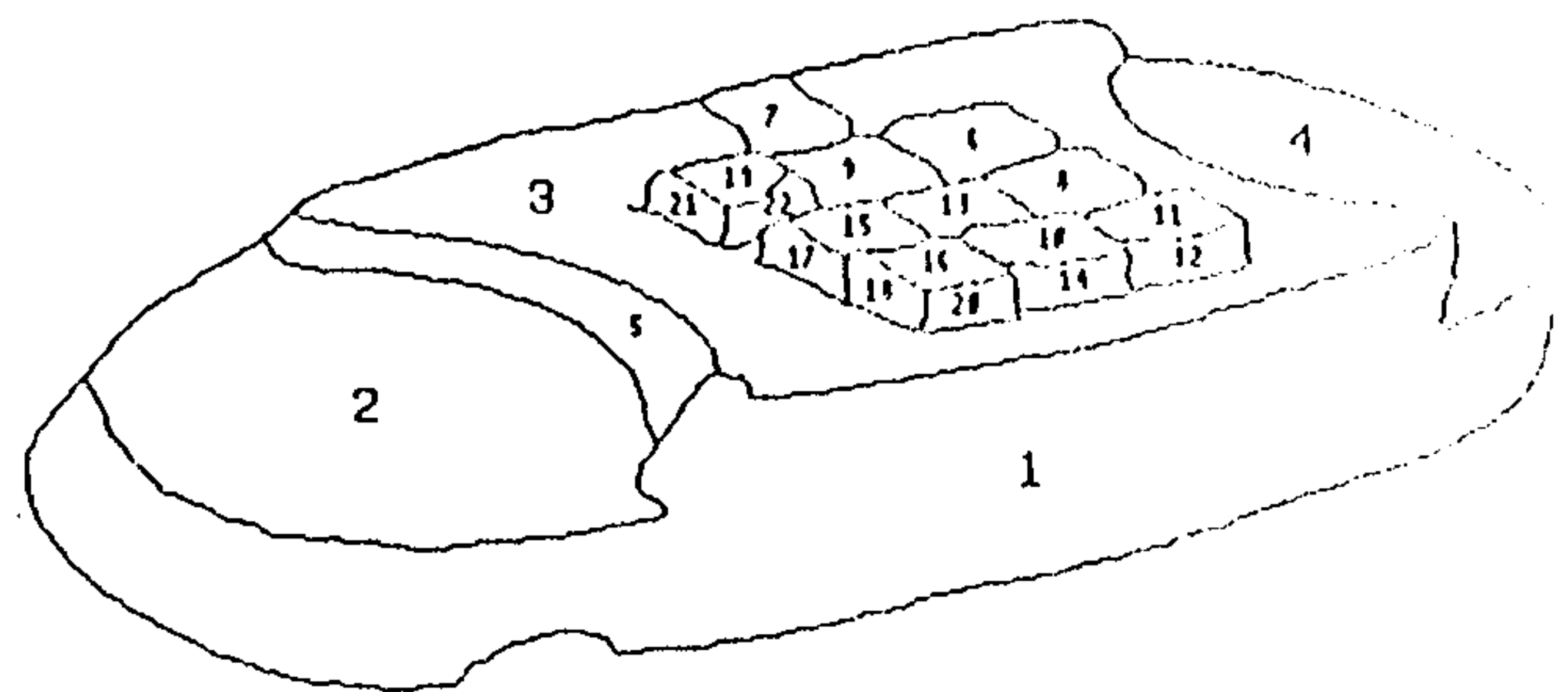
(c) a model car



(d) a model chair



(e) a model face



(f) a model phone

그림 4.8: Model Objects

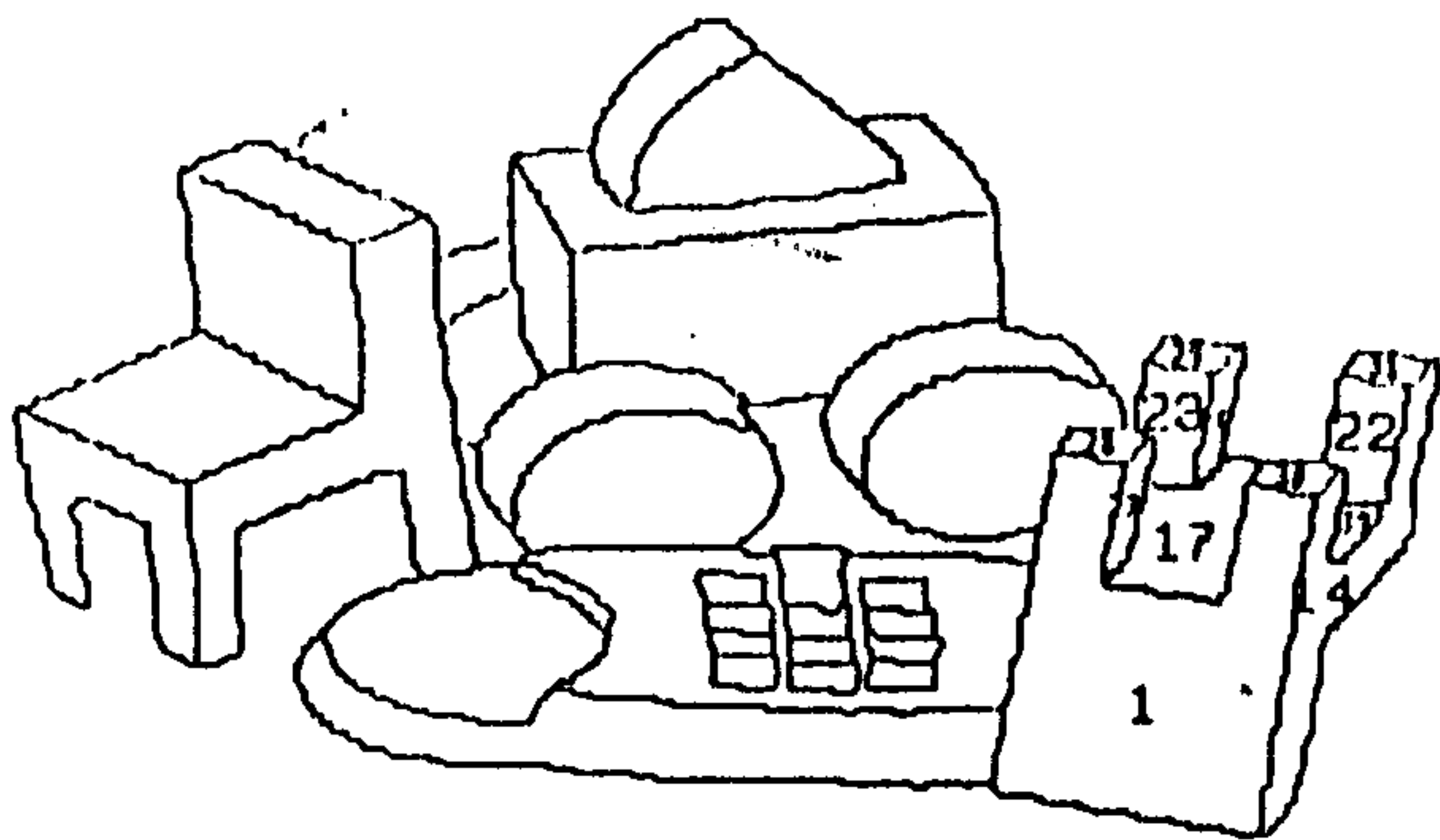
있다.

3 정 합

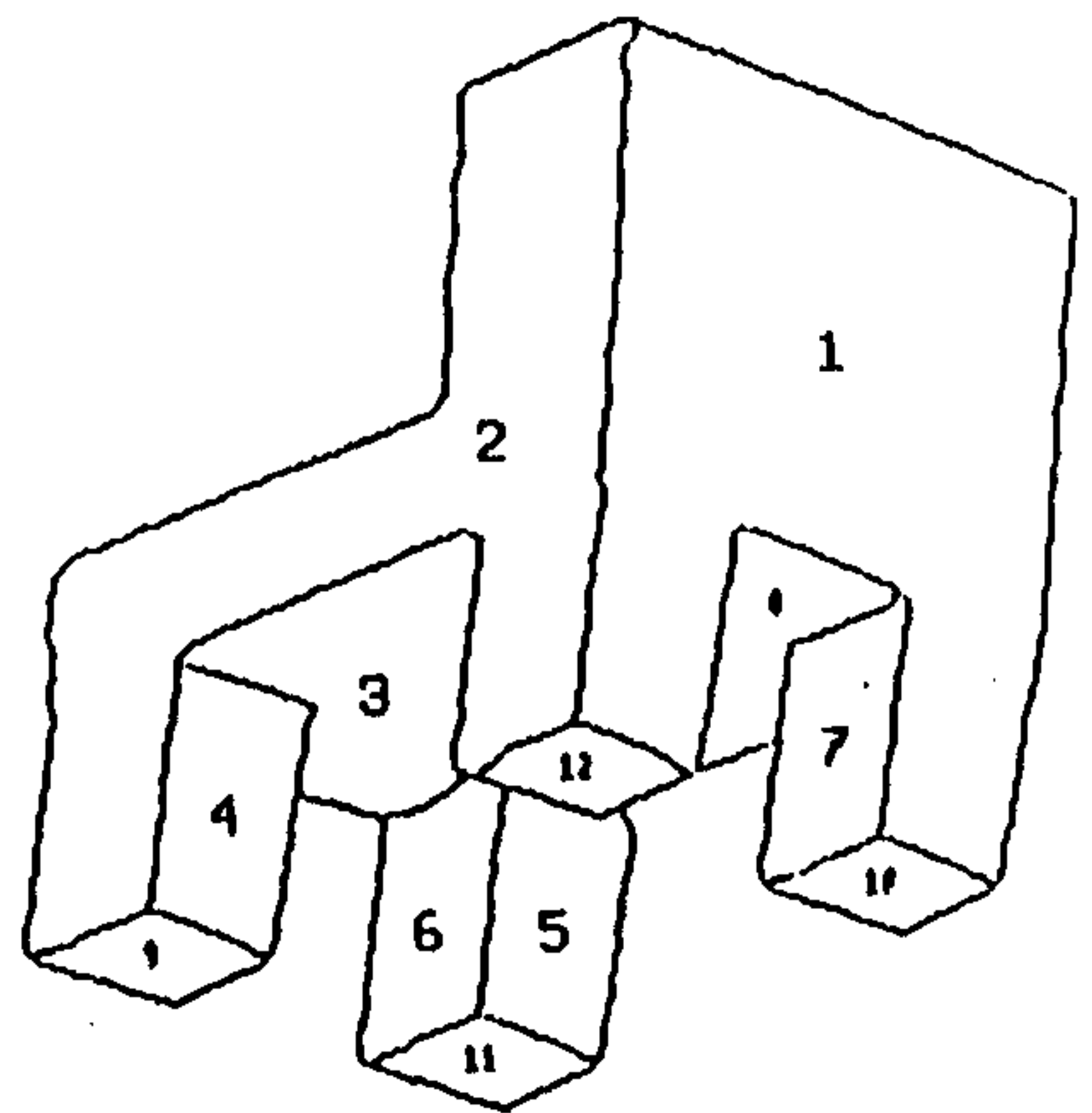
정합 알고리즘의 수행 과정을 보이기 위해 그림 4.9 (a)의 의자 물체와 그림 4.9 (b)의 모델 물체와의 정합 과정을 세 가지의 정합 방법을 사용하여 설명한다. 1 차 제약만을 사용하여 각 SSP에 대응가능한 후보 MSP는 표 4.2에 나열하였고 이 값들은 임계값을 0.15를 사용한 경우이다. 표에서 nil 표시는 후보가 없다는 것을 표시한다. 이 예에서는 후보 MSP의 수가 최대 4이지만 다른 복잡한 물체에서는 큰 값이 될 수 있다. SSP 17와 SSP 27는 심하게 다른 물체에 의해서 겹쳐 있기 때문에 후보 집합에 정확 MSP에 없다. 따라서 이러한 SSP는 정합 과정이 다 끝난 후에 다시 고려되어 져야 한다.

SCTS의 비효율성을 개선하기 위한 휴리스틱 방법으로 Fan 등[Fan89]은 두 단계 탐색 방법 (two-level tree search)을 사용하였다. 이 방법은 휴리스틱하게 가장 잘 정합되는 몇 개의 쌍을 이용하여 탐색 트리의 상부를 생성하고 그 중에서 가장 정합값이 좋은 하나의 경로를 구하고, 아직 고려하지 않은 나머지 쌍을 이용하여 경로를 확장해 가는 방법이다. 예제 문제를 이 두단계 탐색 방법을 이용하여 생성한 탐색 트리는 그림 4.10과 같다. 이 보기에서는 가장 큰 면적을 가지는 평면 표면 조각을 이용하여 상부 트리를 생성하고 그 중에서 가장 최적인 하나의 경로 $\{ (S_1, M_1) (S_{14}, M_2) \}$ 를 찾고, 나머지 부분을 확장한 것이다.

그림 4.11에는 선행 검사 기능을 추가한 SCTS 방법인 SFCCTS에 의해 생성한 탐색 트리를 보여주고 있다. 탐색 시작에는 레벨 1에 해당하는 SSP S_1 에 1차 제약에 의한 대응 가능한 후보 MSP는 M_1 와 M_2 이므로 시작 노드에 두 개의 자손 노드가 생성된다. 아직 처리 하지 않은 SSP는 $S_{14}, S_{22}, S_{23}, S_{26}, S_{27}, S_{28}, S_{29}, S_{30}, S_{31}, S_{39}$ 이고 그



(a) Scene object (numbered)



(b) Model object

그림 4.9: Scene Object and Model Object

SSP	candidate MSP
1	1 2
14	2 1
17	nil
22	8 5 4 7 6
23	8 5 4 7 6
26	6 7 4 3
27	3 6
28	12 11 10 9
29	12 11 10 9
30	12 11 10 9
31	12 11 10 9
39	12 11 10 9

표 4.2: 각 SSP에 대한 후보 MSP

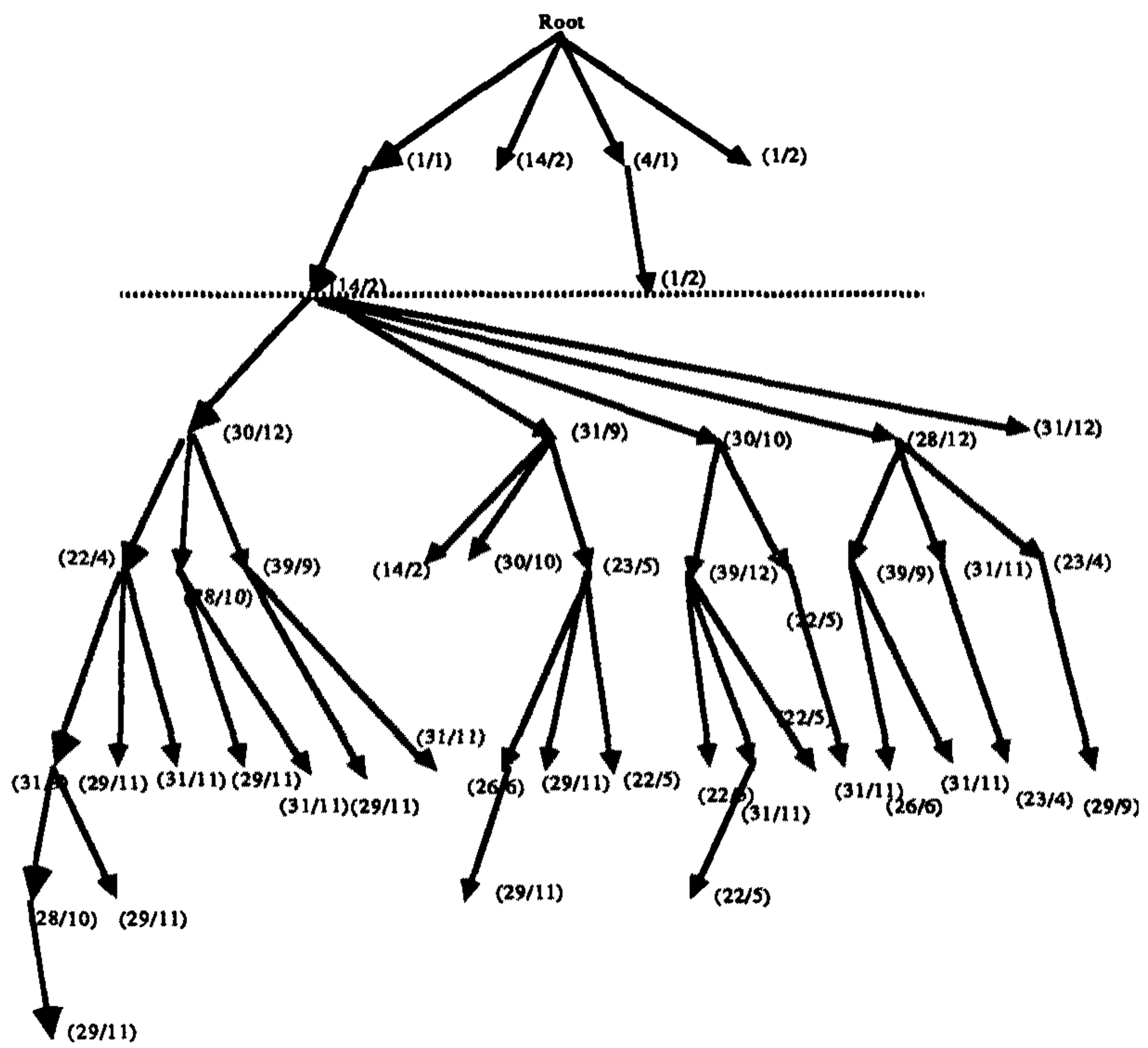


그림 4.10: Search Tree by Two Level SCTS

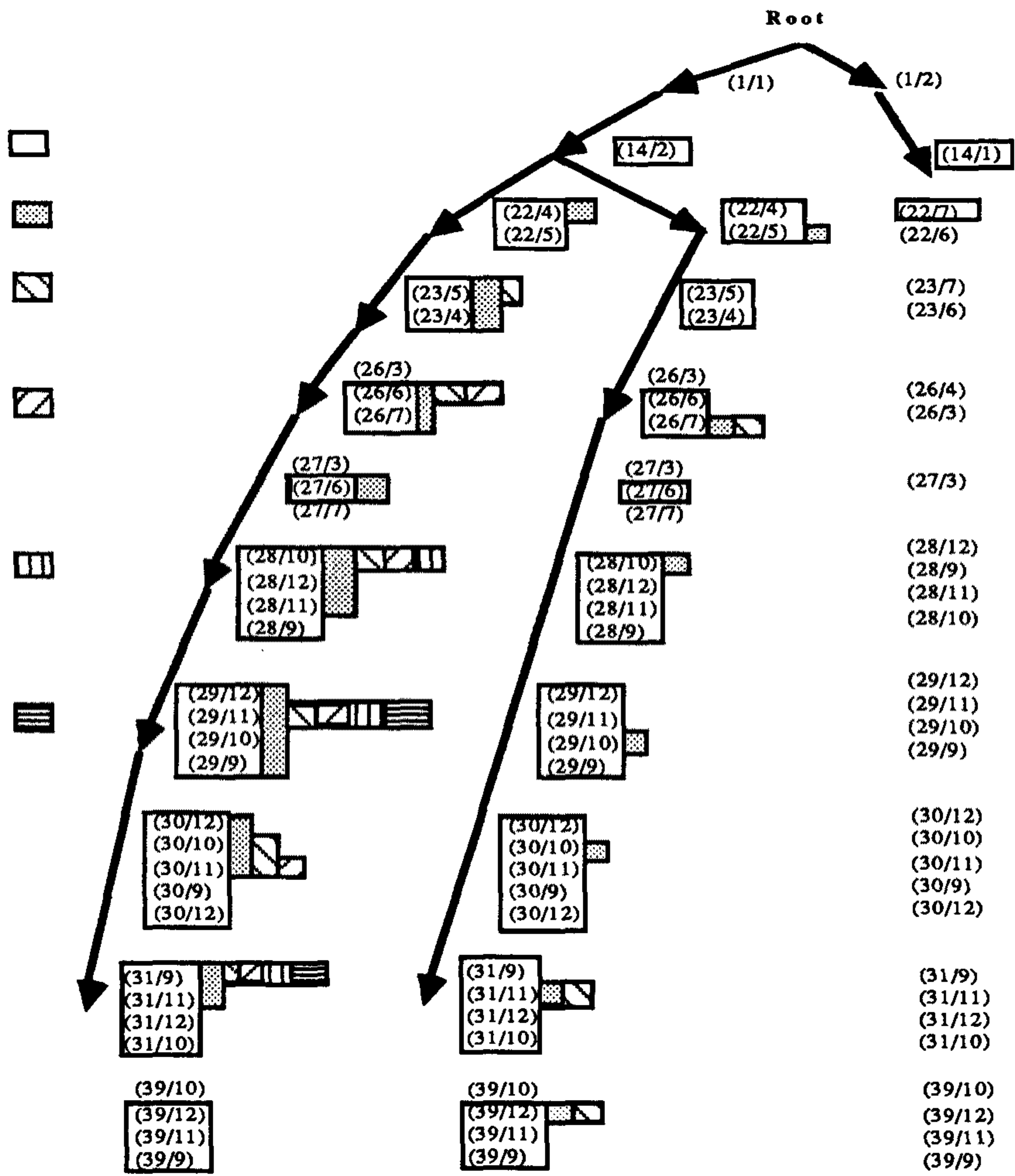


그림 4.11: Search Tree by SFCCTS

에 1차 제약에 의한 대응 가능한 후보 MSP는 SSP #/ MSP #의 형태로 시작 노드의 아래에 나열되어 있다. S_{14} 에 M_2 가 대응되면 (레벨 2) 선행 검사가 나머지 SSP에 적용되어 그 후보가 조정되어 사각형 안에 나열되어 있다. 탐색의 깊이가 깊어지면서 후보 MSP의 수가 줄어들고 살아 남은 후보는 무늬가 다른 사각형안에 나열되어 있다. 화살표는 중간 노드들을 생성하지 않고 적어도 하나의 후보를 가지는 SSP에 해당하는 레벨로 능동적으로 점프하는 것을 보인 것이다. 화살표사이의 중간 노드는 해당하는 SSP가 현재 고려하는 모델에서 대응되는 MSP를 찾을 수 없는 경우이다.

현재의 탐색 경로가 실제 해의 경로이면 몇 개의 노드만을 생성하고 나면 나머지 SSP에 대응되는 후보 MSP의 수는 매우 적어지고 깊어질수록 그 값은 정확하게 1이 된다. 만약 해의 경로가 아니면 몇 개의 노드를 생성하고 나면 살아남은 후보는 거의 없어 상위 레벨에서 탐색의 종료를 알 수 있다. 단순 탐색 방법에 의한 1212 개의 노드의 탐색 공간에서 SFCCTS의 방법에 의해 13개의 노드를 가지는 탐색 공간으로 줄일 수 있었다.

그림 4.12는 OFCCTS에 의한 탐색 트리를 보여준다. 트리상의 노드에는 생성되는 순서, 정합쌍 (SSP #/ MSP #), 비용함수 $g(x) + h(x)$ 를 표시하였다. 실제 해의 경로에 있는 노드는 비용 함수값이 매우 적어 미리 선택되어지므로 필요없는 노드의 생성을 가능한 억제한다.

여러 입력 장면들에 대해 실험한 결과 선행 기능이 추가된 탐색 방법이 매우 효율적으로 수행됨을 알 수 있었다. 본 보고서에서는 그림 4.13에 보여진 3개의 입력 장면을 인식한 결과를 발표한다. 각 장면은 4개의 다른 종류의 물체로 구성되어 있다. 그림 4.14 - 4.16에는 장면1, 장면2 그리고 장면 3에 있는 자동차 물체의 인식 시간, 생성된 노드의 수 그리고 정합 값을 모든 물체에 대해 구한 결과를 보여 준다. x 축은 물체의 모델 번호이고 y 축은 (a) 초단위 시간 (b) 갯수 그리고 (c) 정합율을 나타낸다. 세 가지의 정합 알고리즘 SCTS, SFCCTS 그리고 OFCCTS을 비교한 결과 (c)에서 보여주듯이 정합 값은 거의 비슷하지만 인식 시간과 생성된 노드의 수는 매우 적음을 알 수 있다. 특히 모

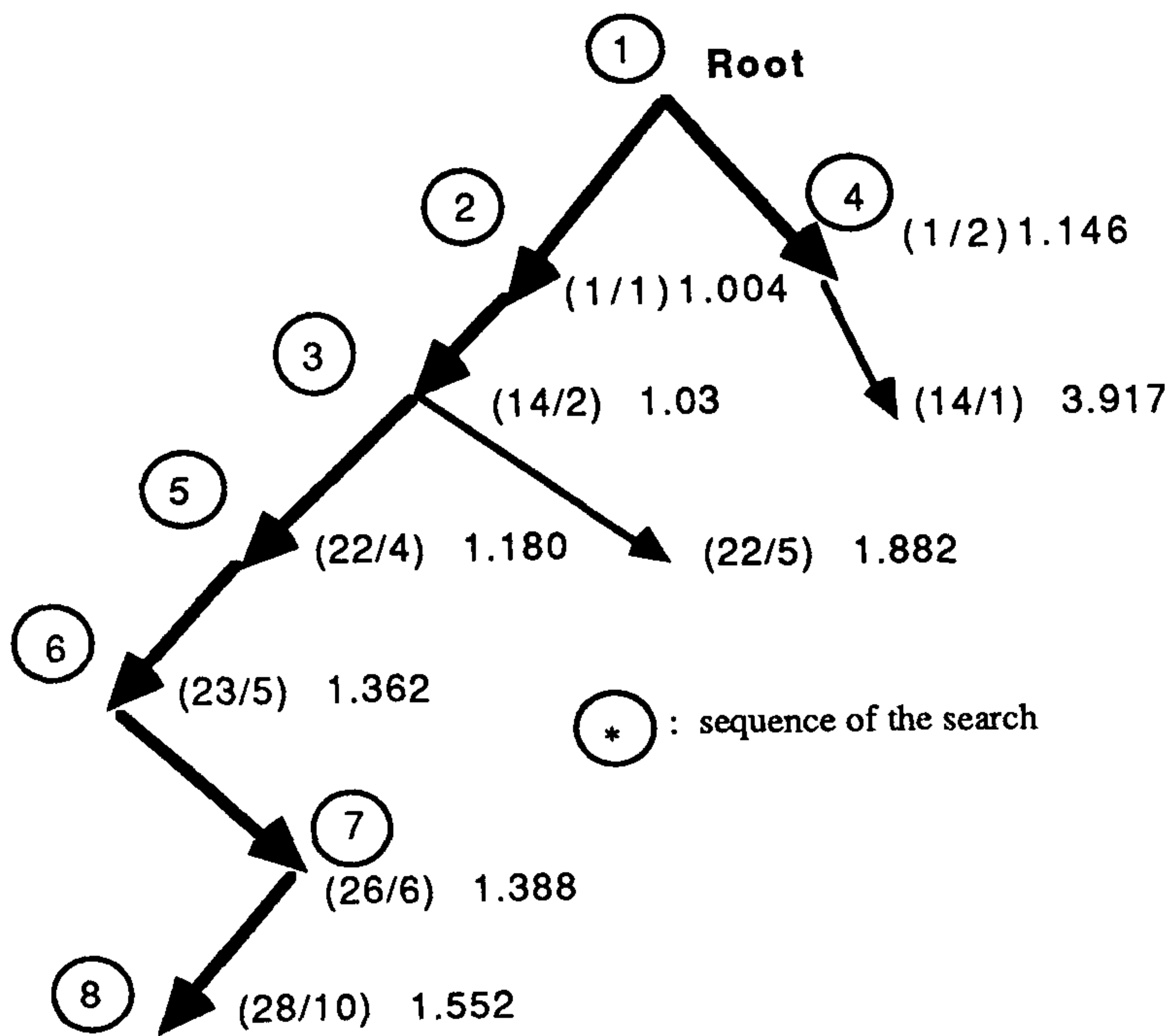


그림 4.12: Search Tree by OFCCTS

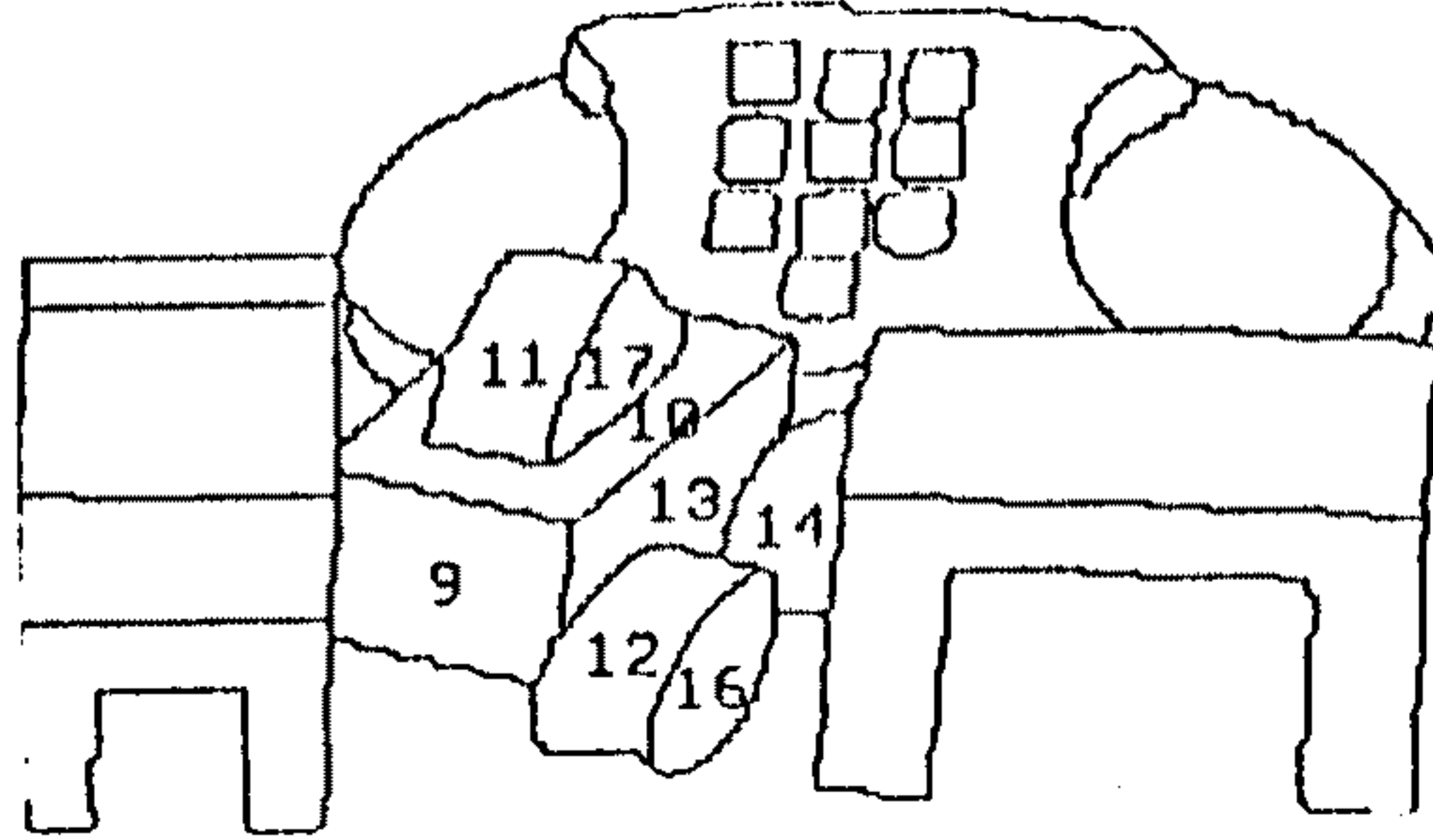
	degree (model id)			matching time			# of nodes		
	Two Level	SFCCTS	OFCCTS	Two Level	SFCCTS	OFCCTS	Two Level	SFCCTS	OFCCTS
object 1	20.63(19)	20.63(19)	20.67(19)	1570.21	275.48	514.38	625	87	83
object 2	15.25(5)	15.25(5)	15.25(5)	1765.32	77.81	121.68	692	69	74
object 3	9.08(9)	9.08(9)	9.08(9)	121.97	12.34	22.46	217	46	48
object 4	1.32(27)	1.32(27)	1.32(27)	3.58	2.01	2.16	77	25	25

표 4.3: Scene-1에 대한 인식 결과

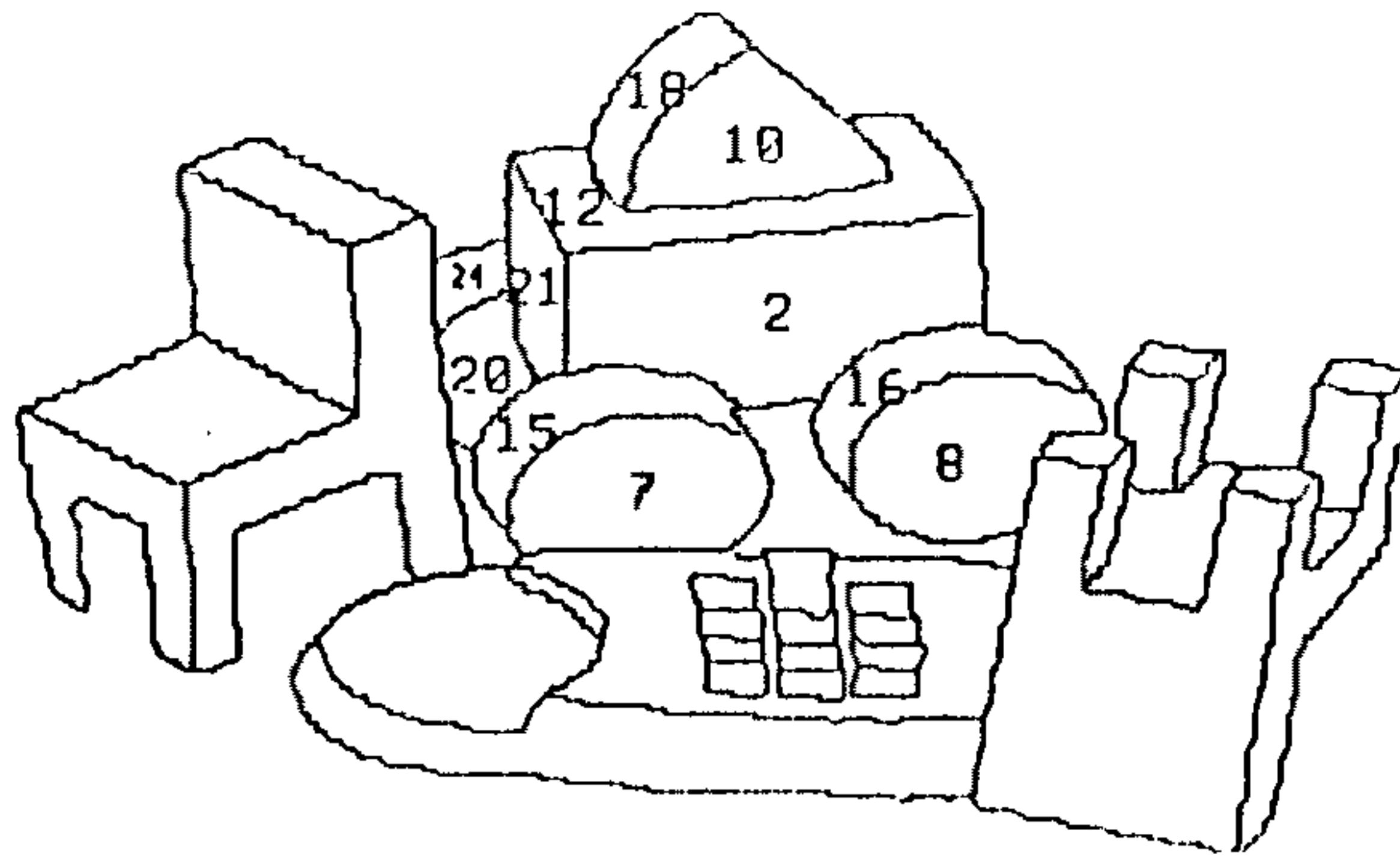
델의 모양이 매우 비슷한 경우 (모델 5-8은 같은 물체에서 생성된 것임) 기존의 탐색 방법은 그들 사이의 차이점을 찾기 위해 많은 노드를 생성해야 하지만 선행 검사 기능이 부가된 방법에서는 미리 검사하므로 훨씬 더 효율적임을 알 수 있다.

표 4.3- 4.5에서는 장면1, 장면2 그리고 장면 3의 모든 물체를 인식하는 과정을 세 가지 방법에 의해 구한 정합값, 시간 그리고 생성된 노드의 수를 표시한 것이다. 이 표에서도 선행 검사의 기능이 매우 효율적임을 알 수 있다.

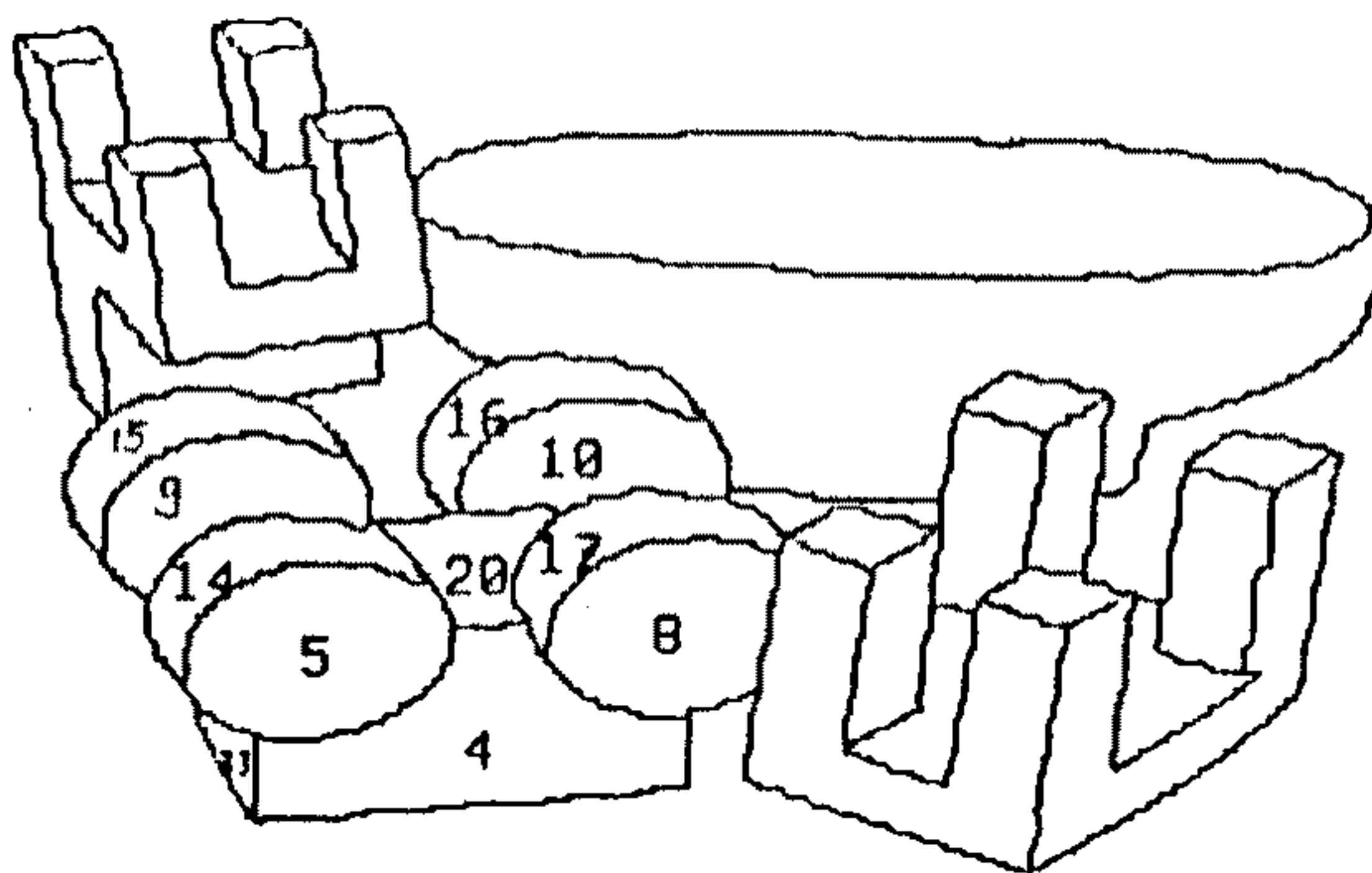
표면 조각들이 단순하고 표면 조각들이 복수 관계를 가지므로 어떤 노드에서 가능한 많은 표면 조각 간의 관계를 조사해야 한다. 이러한 취지에서 선행 기능은 아직 처리하지 않은 부분을 미리 검사함으로써 가능하지 않은 경로를 탐색하지 않는다. 비록 선행 기능이 더 많은 수의 제약 검사를 하지만 전체적으로 줄여진 노드 수에 비하면 그 시간이 크지 않아 선행 기능이 추가된 탐색은 매우 효율적임을 알 수 있다.



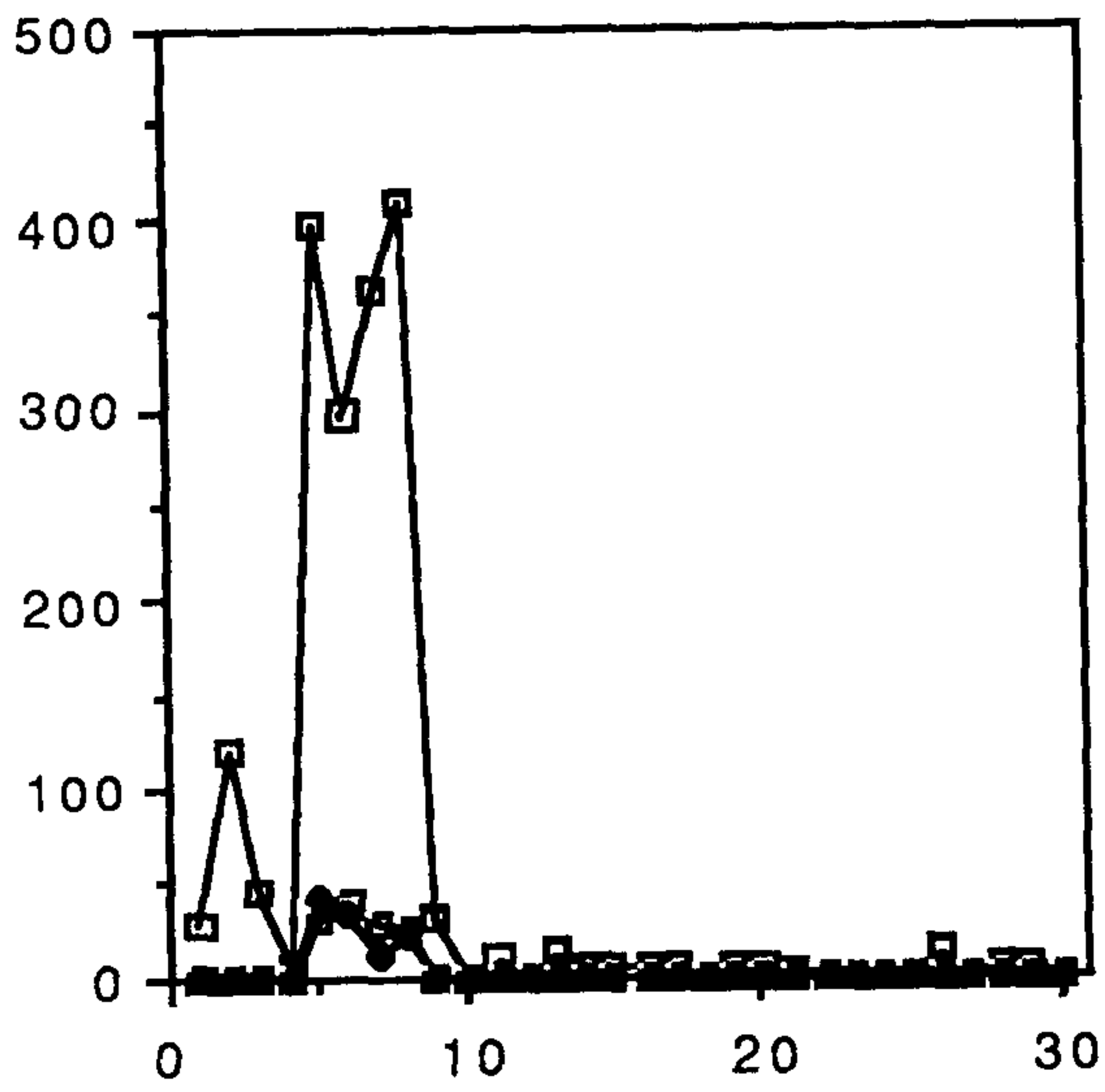
(a) Scene-1



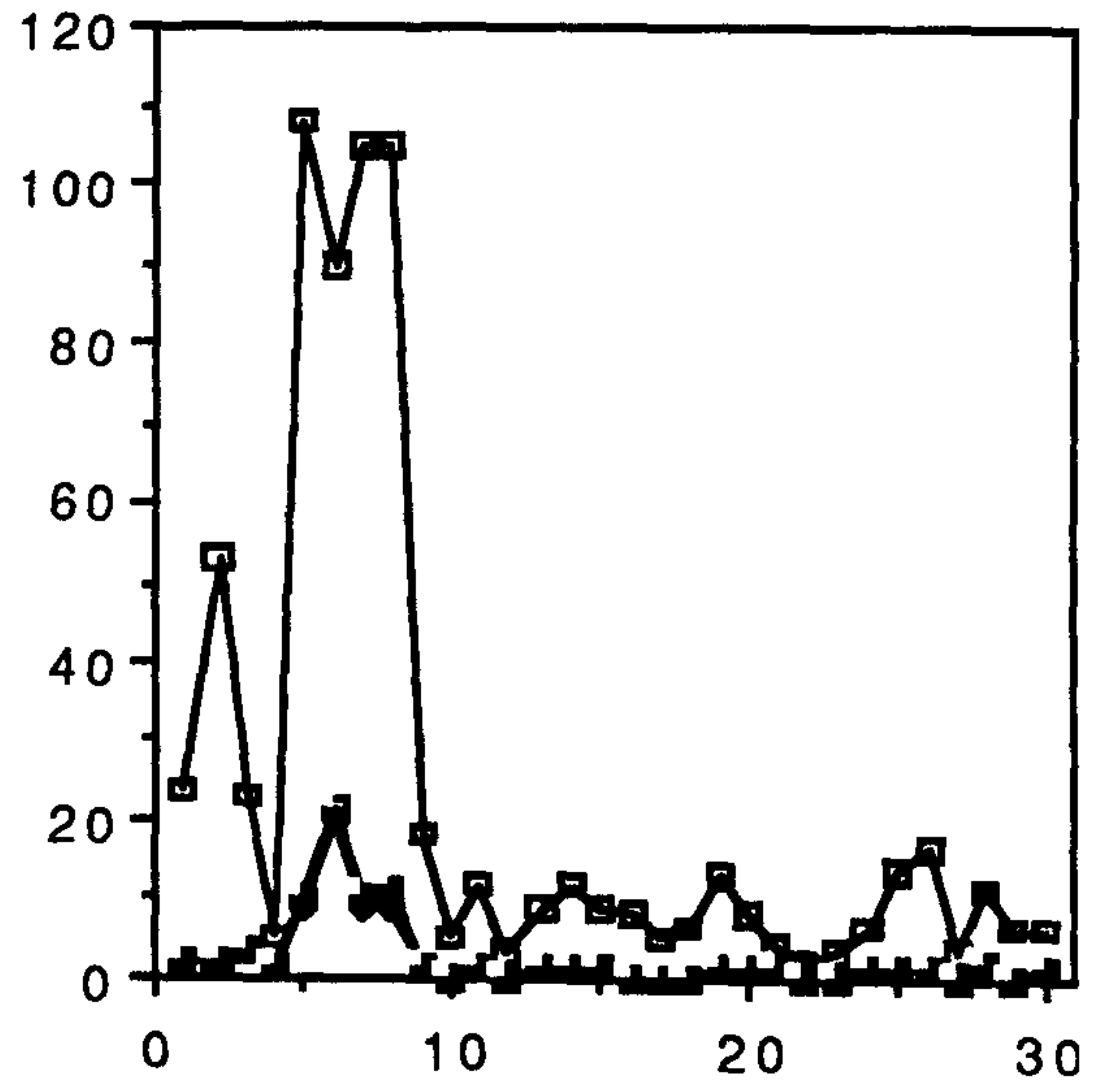
(b) Scene-2



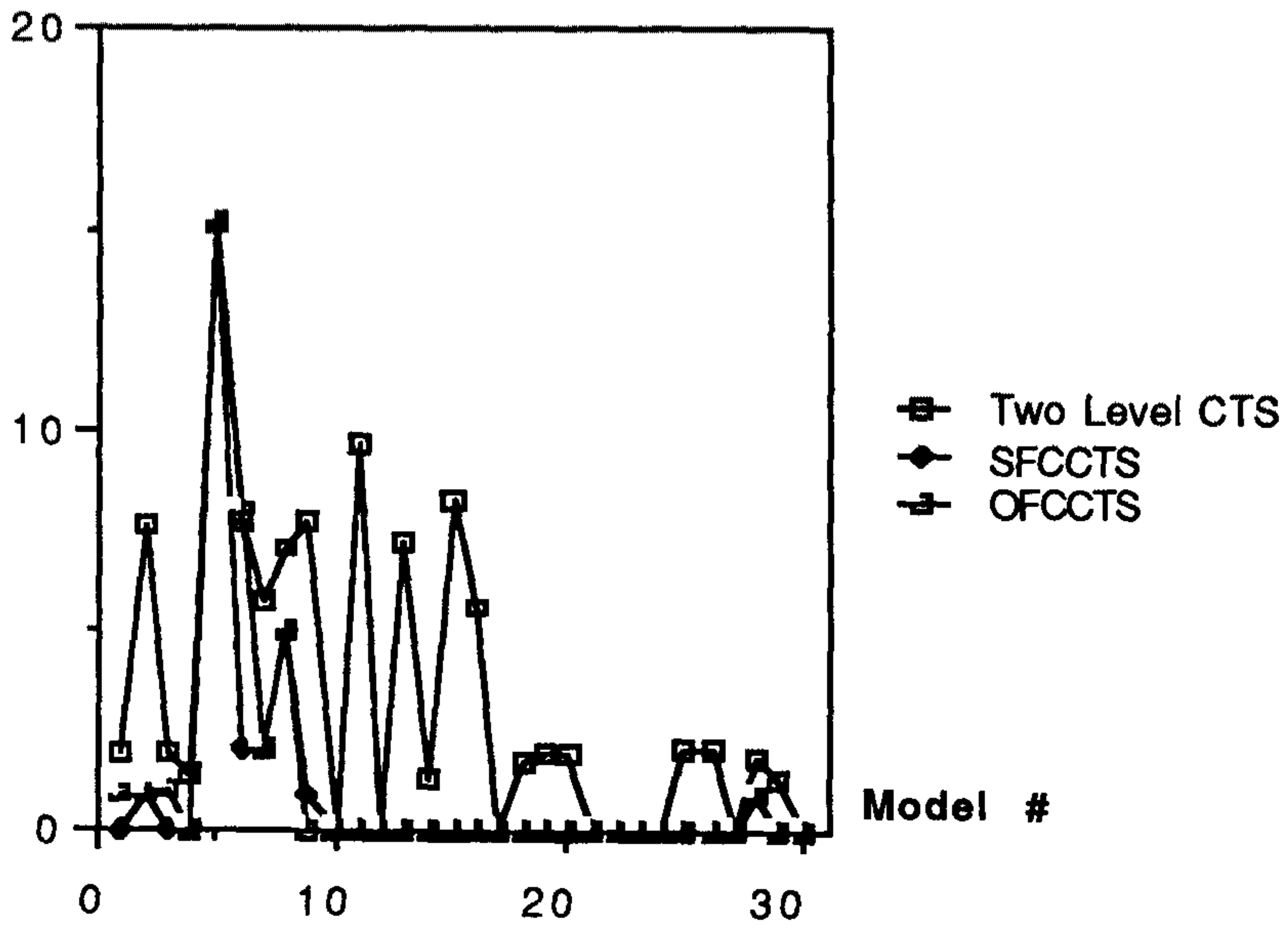
(c) Scene-3



(a) time

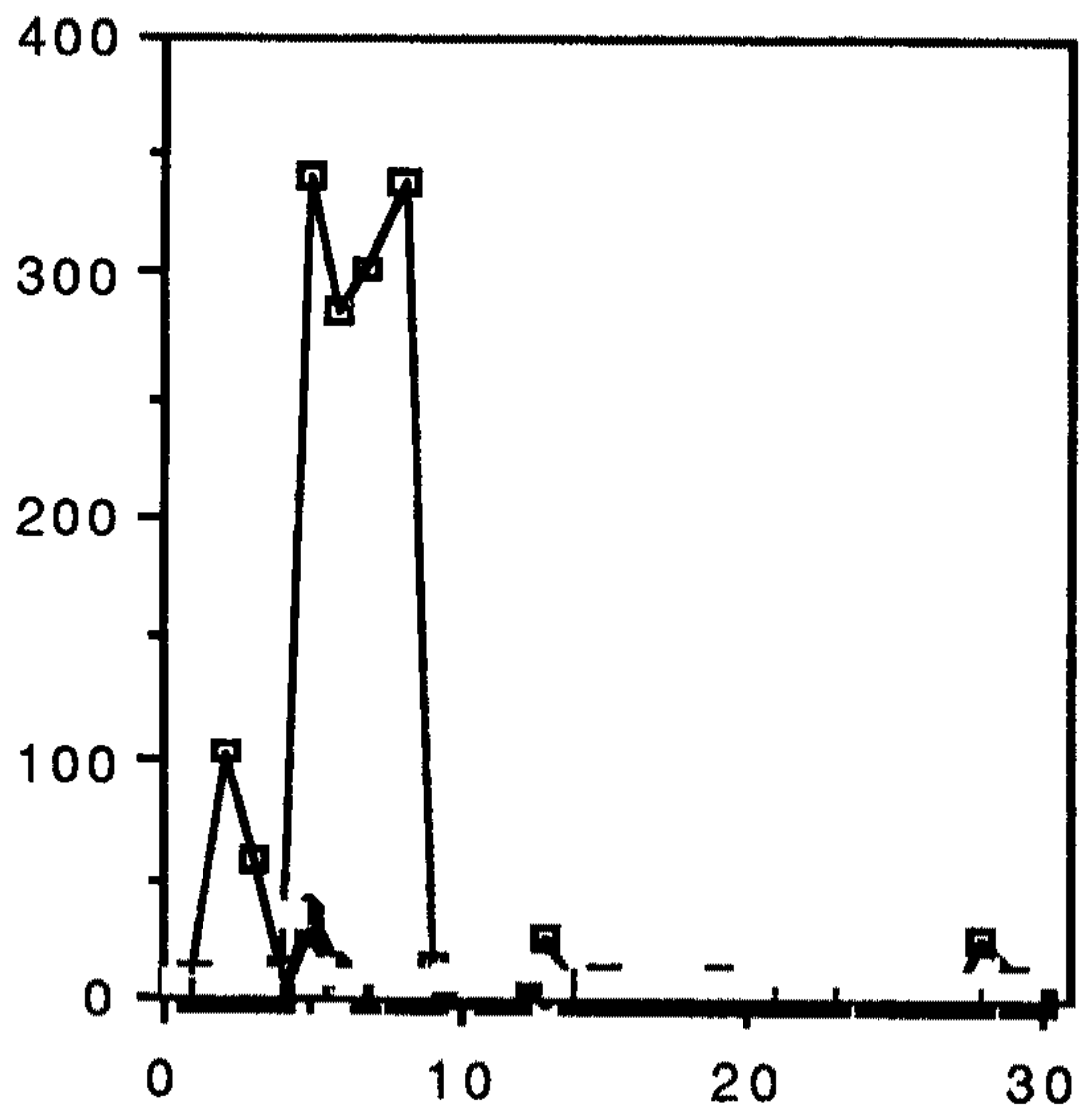


(b) # of generated nodes

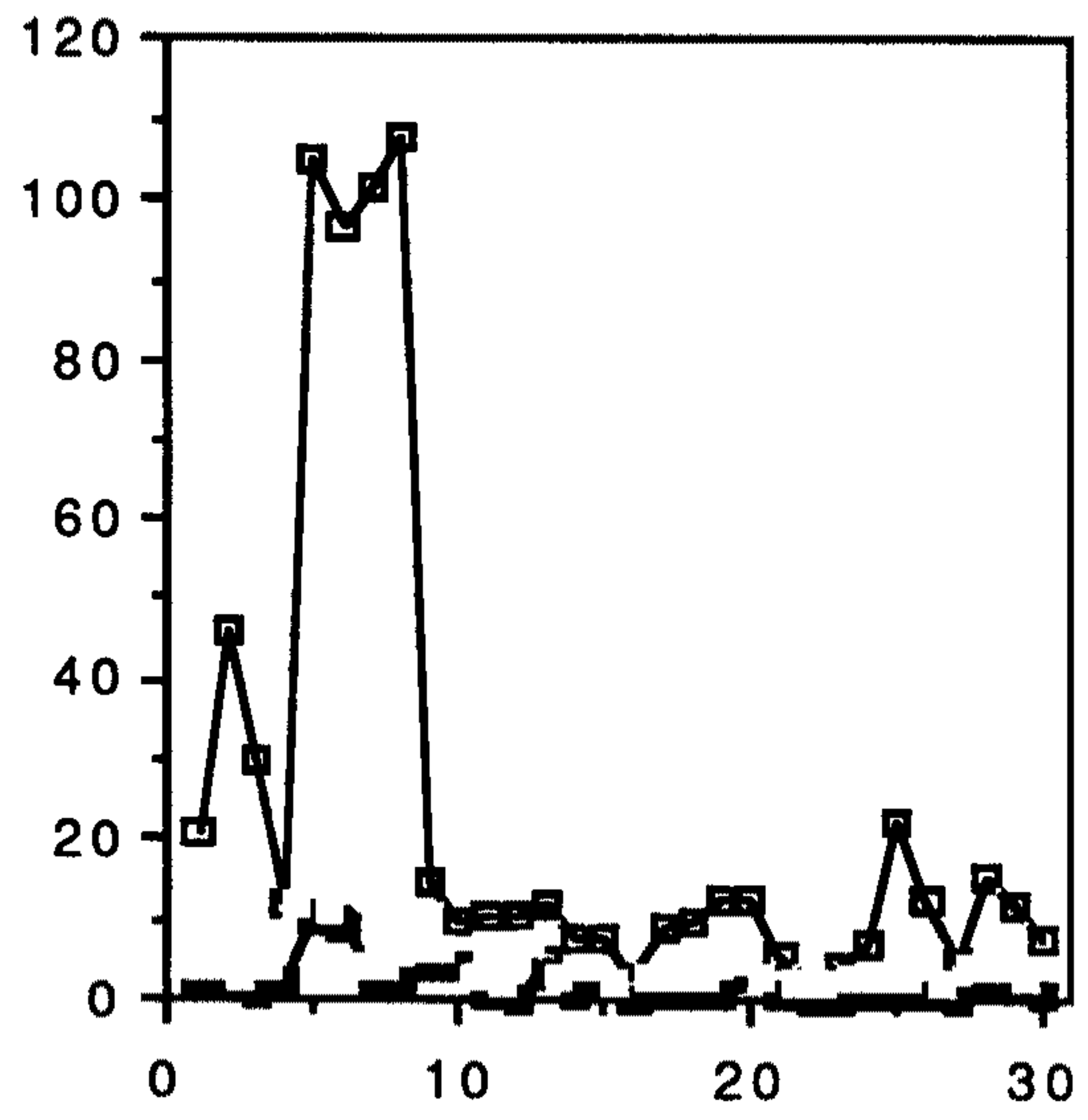


(c) Matching Degree

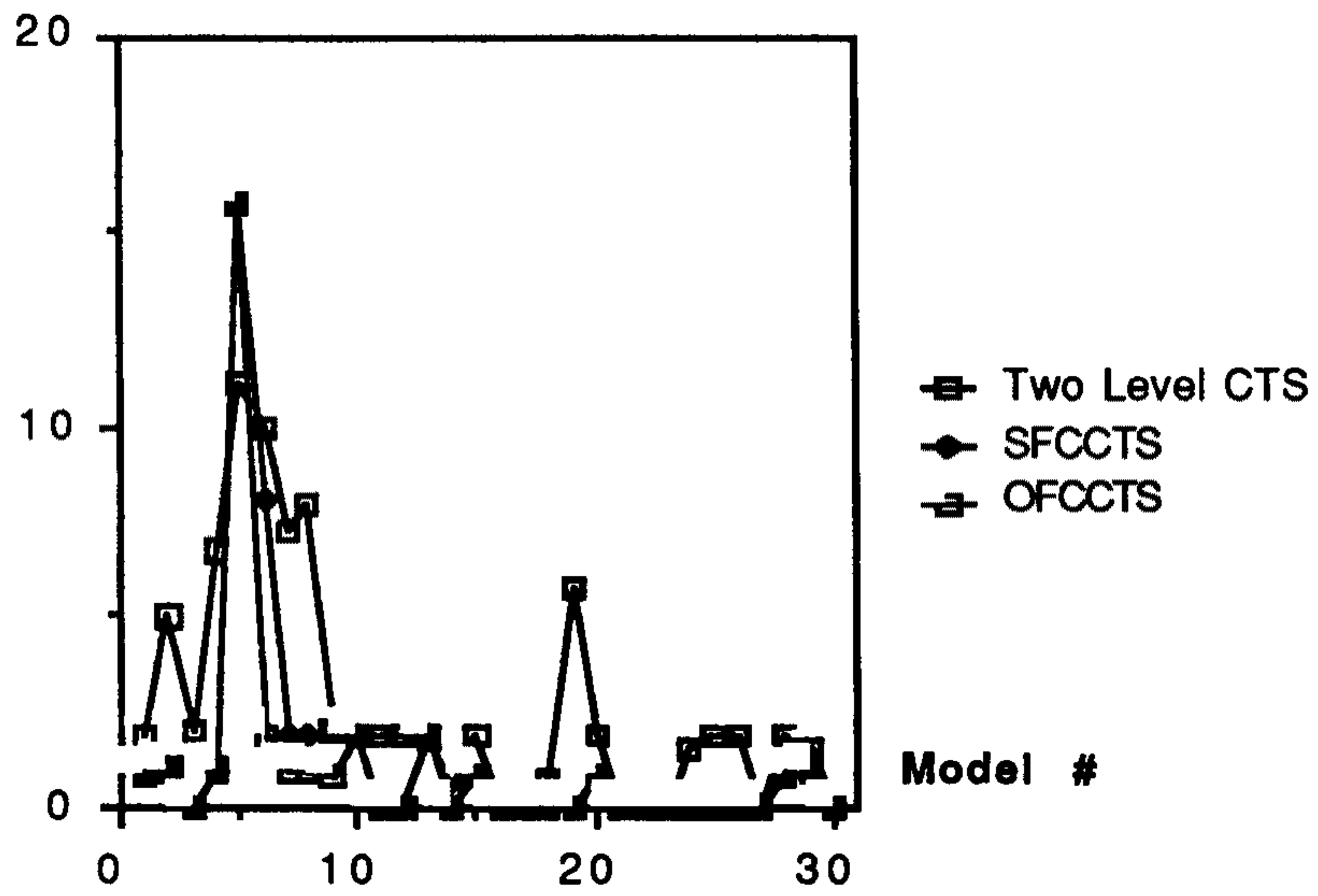
그림 4.14: Results vs model# for Scene-1



(a) time

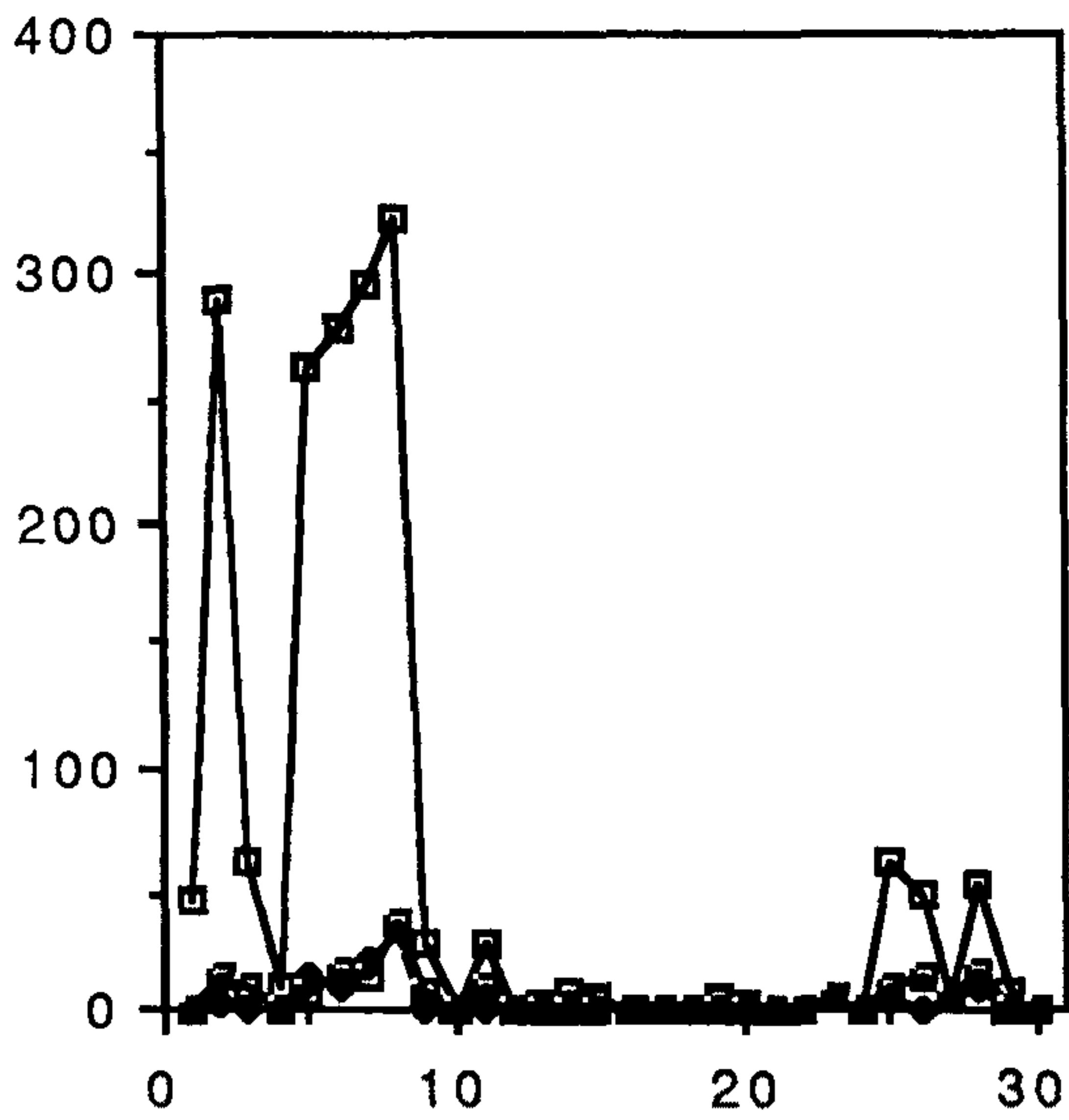


(b) # of generated nodes

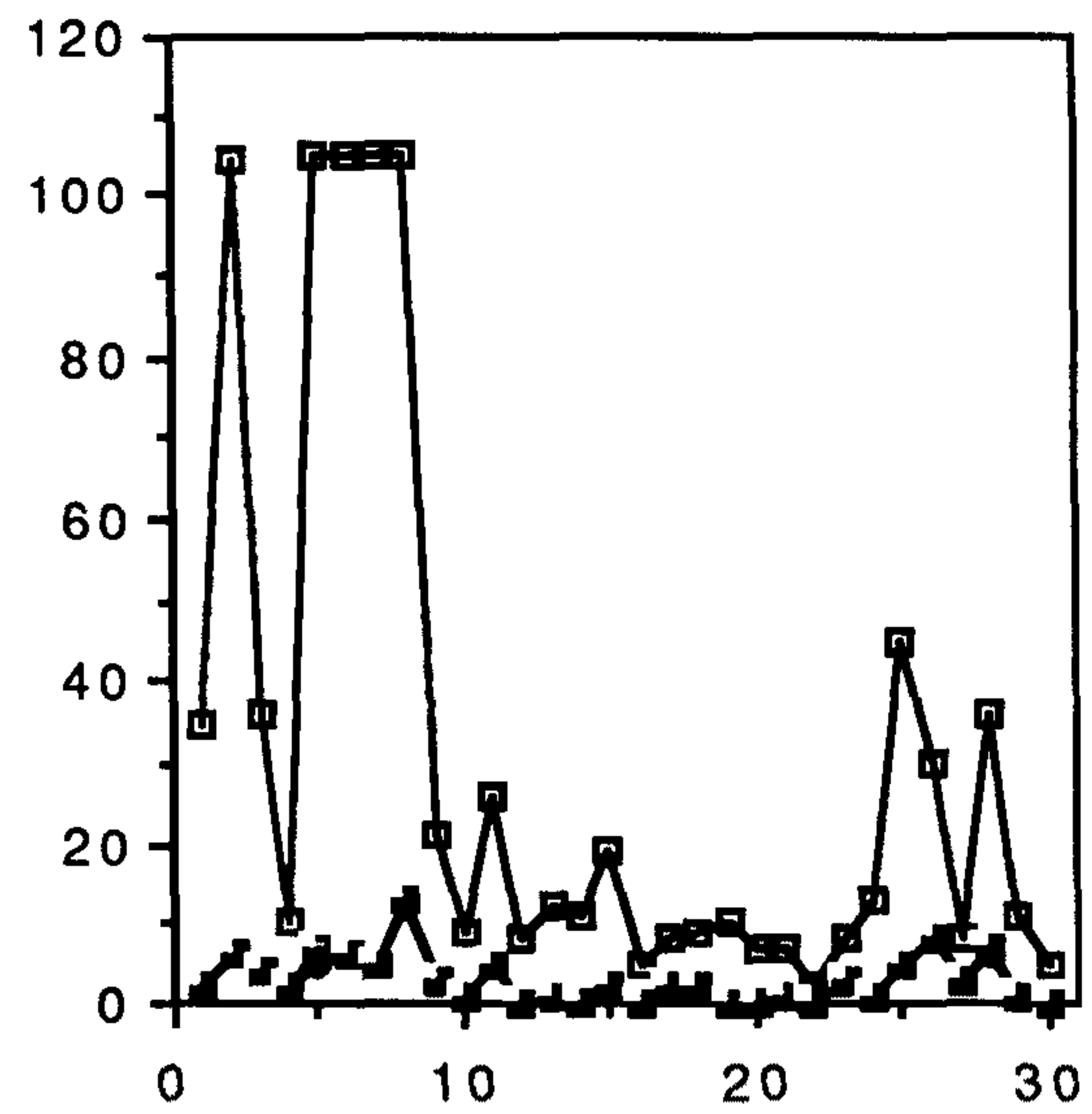


(c) Matching Degree

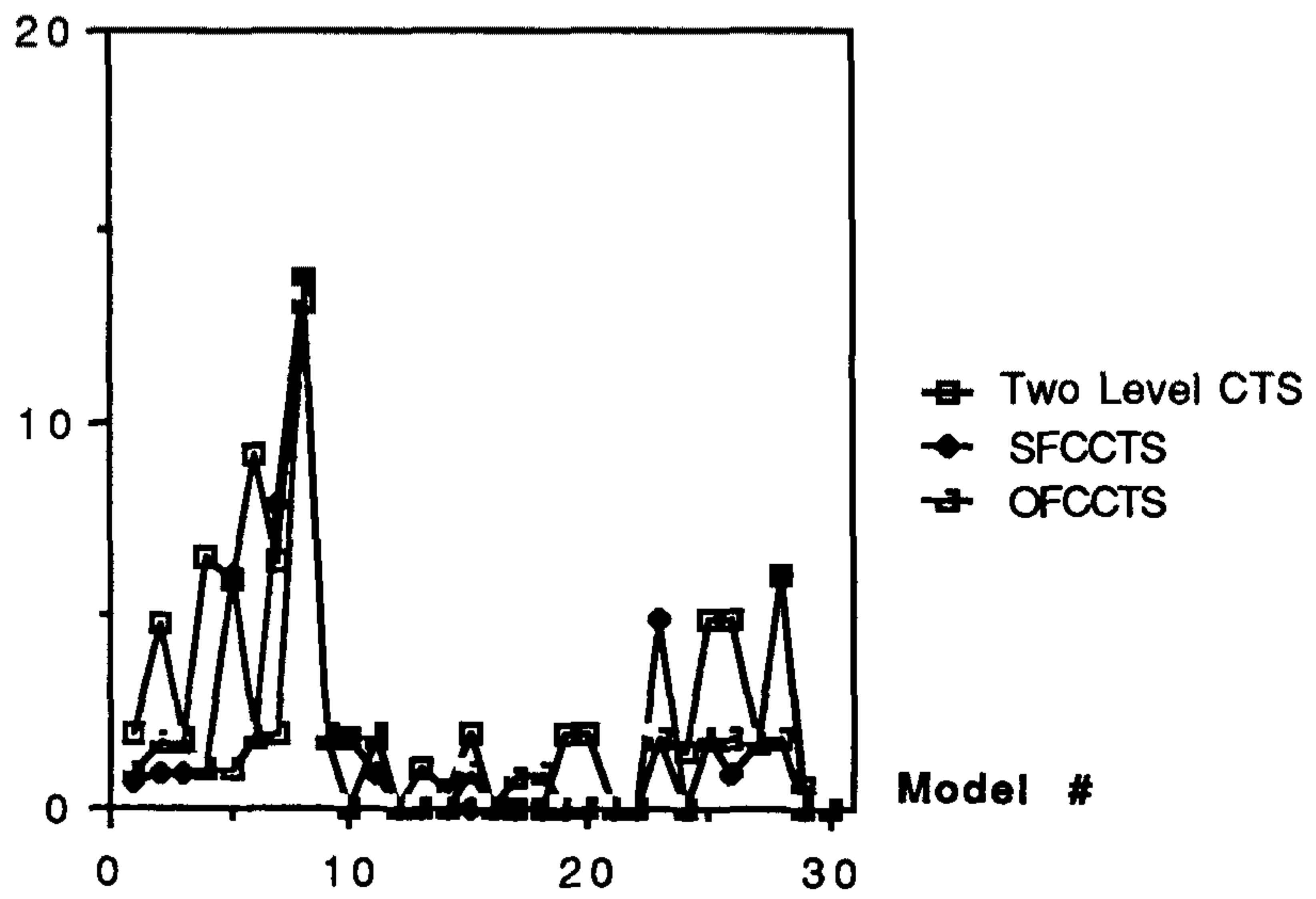
그림 4.15: Results vs model# for Scene-2



(a) time



(b) # of generated nodes



(c) Matching Degree

그림 4.16: Results vs model# for Scene-3

	degree (model id)			matching time			# of nodes		
	Two Level	SFCCTS	OFCCTS	Two Level	SFCCTS	OFCCTS	Two Level	SFCCTS	OFCCTS
object 1	19.23(20)	17.96(20)	19.25(20)	1145.54	168.54	231.46	548	48	55
object 2	17.30(12)	17.30(12)	17.30(12)	682.24	75.97	105.10	564	69	69
object 3	11.06(5)	15.79(5)	15.82(5)	1583.04	60.89	45.66	737	62	62
object 4	16.00(9)	16.00(9)	16.00(9)	173.87	15.73	18.83	25	29	28

표 4.4: Scene-2에 대한 인식 결과

	degree (model id)			matching time			# of nodes		
	Two Level	SFCCTS	OFCCTS	Two Level	SFCCTS	OFCCTS	Two Level	SFCCTS	OFCCTS
object 1	16.80(11)	16.80(11)	16.80(11)	707.12	62.06	92.76	603	69	68
object 2	19.35(28)	14.35(28)	14.35(28)	548.45	66.80	56.24	447	50	47
object 3	13.70(8)	13.70(8)	13.12(8)	1815.88	100.62	159.32	915	78	90
object 4	1.91(22)	1.91(22)	1.91(22)	3.88	3.01	2.87	68	13	13

표 4.5: Scene-3에 대한 인식 결과

제 5 절 결 론

본 과제에서는 거리 영상에서 삼차원 물체를 인식하는 한 방법을 제안하였다. 인식을 위해 거리 영상은 물체 고유의 경계선을 따라 영상을 분할하여 각각의 표면 조각은 그래프의 노드로 표면 조각 간의 관계는 그래프의 링크로 표시한 속성 그래프로 표현한다. 모델 속성 그래프와 장면 속성 그래프가 주어질때 표면 조각들간의 대응관계를 구하고 삼차원 공간상의 변형 함수를 계산하기 위해 선행 탐색 기능이 내재된 순서적 선행 제약 트리 탐색 방법과 최적 제약 트리 탐색 방법을 제안하였다. 실제의 거리 영상을 사용한 실험 결과 선행 기능을 가진 방법들이 여러 물체가 겹쳐 있는 상황에서도 매우 효율적임을 보였다. 특히 다른 물체 인식 방법과 비교하면 이 선행 기능을 가진 방법은 모델 물체들이 유사한 경우에 더욱 효율적임을 알 수있다. 왜냐하면 선행 기능이 유사한 모델 물체간의 차이점을 미리 발견하기 때문에 비슷한 물체를 비교하는 시간을 줄일 수 있기 때문이다. 제약의 능력을 향상시키기 위한 연구가 진행되어야 하는데, 한 방법으로 표면 조각의 경계선의 정보를 이용하는 방법이다. 표면 조각의 경계선의 정보는 단순한 속성보다 더 많은 정보를 제공한다. 이 경계선의 정합과 표면의 정합을 계층적으로 적용하는 연구가 앞으로 진행되어야 할 것이다.

참고 문헌

- [Bal82] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice-Hall, Inc., New Jersey, 1982.
- [Bha84] B. Bhanu, "Representation and shape matching of 3-D objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6(3), May 1984.
- [Bes85] P.J. Besl and R.C. Jain, "Three-dimensional object recognition," *ACM Comput. Surveys* 17(1), March 1985.
- [Bes86] P.J. Besl and R.C. Jain, "Invariant surface characteristics for 3-D object recognition in range images," *Computer Vision, Graphics, and Image Processing* 33, 1986.
- [Bes88] P.J. Besl and R.C. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-10(2), March 1988.
- [Bol86] R.C. Bolles and P. Horaud, "3DPO: A three dimensional part orientation system," *Internat. J. Robotics Research* 3, 1986.
- [Bra85] M. Brady, J. Ponce, A. Yuille and H. Asada, "Describing surfaces," *Computer Graphics and Image Processing* 32, 1985.

- [Che89] C.H. Chen and A.C. Kak, "A robot vision system for recognizing 3-D objects in low-order polynomial time," *IEEE Trans. System Man Cybern.* SMC-19 , 1989.
- [Chi86] R.T. Chin and C.R. Dyer, "Model-based recognition in robot vision," *ACM Comput. Surveys* 18(1), March 1986.
- [Fan87] T.J. Fan, G. Medioni and R. Nevatia, "Segmented descriptions of 3-D surfaces," *IEEE J. Robotics and Automation* 3(6), December 1987.
- [Fan89] T.J. Fan, G. Medioni and R. Nevatia, "Recognizing 3-D objects using surface descriptions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-11(11), November 1989.
- [Fau81] O.D. Faugeras and M. Berthod, "Improving consistency and reducing ambiguity in stochastic labeling: an optimization approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-3(4), 1981.
- [Fau86] O.D. Faugeras and M. Hebert, "The representation, recognition and locating of 3-D objects," *Internat. J. Robotics Research* 5(3), Fall 1986.
- [Gas84] P.C. Gaston and T. Lozano-Pe'rez, "Tactile recognition and localization using object models: The case of polyhedra on a plane," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-6, May 1984.

- [Gri84] W.E.L. Grimson and T. Lozano-Pe'rez, "Model-based recognition and localization from sparse range or tactile data," *Internat. J. Robotics Research* 3(3), 1984.
- [Gri87] W.E.L. Grimson and T. Lozano-Pe'rez, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9(4), July 1987.
- [Gri90] W.E.L. Grimson, "The combinatorics of object recognition in cluttered environments using constraint search," *Artificial Intelligence* 44, 1990.
- [Har80] R.H. Haralick and G. Elliott, "Increasing tree search efficiency for constraint satisfaction problems," *Artificial Intelligence* 14, 1980.
- [Hof87] R. Hoffman and A.K. Jain, "Segmentation and classification of range images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9(5), September 1987.
- [Jai88] A.K. Jain and R. Hoffman, "Evidence-based recognition of 3-D objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-10 (6), November 1988.
- [Nud83] B. Nudel, "Consistent-labeling problems and their algorithms: expected-complexities and theory-based heuristics," *Artificial Intelligence* 21, 1983.

- [Osh79] M. Oshima and Y. Shirai, "A scene description method using three-dimensional information," *Pattern Recognition* 11, 1979.
- [Osh83] M. Oshima and Y. Shirai, "Object recognition using three-dimensional information," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI- 5(4), 1983.
- [Sha87] L.G. Shapiro and R.M. Haralick, "Structured descriptions and inexact matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-3(5), 1987.
- [Sto87] G. Stockman, "Object recognition and localization via pose clustering," *Computer Vision, Graphics, and Image Processing* 40, 1987.
- [Yan86] H.S. Yang and A.C. Kak, "Determination of the identity, position and orientation of the topmost object in a pile," *Computer Vision, Graphics, and Image Processing* 36, 1986.

제 5 장 결 론

시각 정보 처리 및 영상 분석에 관한 기술은 사용자에게 보다 자연스러운 인터페이스를 제공하는 쓰기 쉬운 컴퓨터를 만들기 위해 꼭 필요한 기술이며, 컴퓨터의 응용 분야가 단순한 자료 처리에서 벗어나 음성이나 영상 등 여러 형태의 정보를 복합적으로 처리하는 소위 멀티미디어 분야로 나아가고 있는 시점에서 그 중요성이 더해가고 있다.

이러한 필요성에 따라 선진국에서는 국가적인 차원의 집중 투자로서 연구를 수행하여 이미 문자 및 문서 인식, 문서의 자동 검색, 로봇트 제어, 군사적인 응용 분야 등에서 실용화 단계에 도달하였다. 그러나 이러한 선진국들의 연구 성과와는 달리 국내에서는 아직까지 시각 정보 처리 및 영상 분석에 관한 연구가 미약한 실정이며, 기술의 미비로 인하여 대부분 외국에서 개발된 제품을 사용하고 있다.

이에따라 본 연구에서는 시각 정보 처리 기반 기술의 확보를 위한 연구를 수행하였다. 손으로 쓴 숫자 및 산술 부호, 그리고 제한된 수의 문자를 인식하는 필기 입력 산수 교사 시스템을 개발하였으며, 전표나 신상명세서 등과 같이 많은 양이 반복되어 나타나는 형식 문서(form)를 자동 입력하는 시스템에 필요한 지식 표현 방법을 개발하였다. 또한 3차원 시각 정보의 처리를 위하여 보다 빠른 시간내에 매우 유사한 형태를 갖는 3차원 물체들을 인식할 수 있는 선행 탐색 트리 알고리즘을 개발하였다.

위의 연구들을 통하여 일반인이 불편을 느끼는 키보드를 통한 입력의 단점을 보완하여 사용자에게 전자펜을 이용한 좀 더 친숙한 인터페이스 방법을 제공하고, 전표 등의 형식 문서를 수동 입력하는데 필요한 노력과 시간을 감소시켜 줌으로써 사무자동화 분야의 기술 향상에 이바지하였다. 또한 3차원 물체를 더 효율적으로 정확히 인식할 수 있는 방법을 제안함으로써 컴퓨터 시각 기술을 이용한 산업 자동화 분야의 발전에 기여하였다.

위에서 언급한 바와 같이 본 연구에서는 앞으로 s/w 연구의 핵심이 될 시각 정보 처

리에 관한 기반 기술을 제공함으로써 컴퓨터 활용의 극대화와 컴퓨터 보급의 확대에 기여하고, 사무 자동화와 산업 자동화 및 관련 산업 기술 개발을 촉진할 것이다.