IMAGE SEGMENTATION AND INTERPRETATION

USING A KNOWLEDGE DATABASE

by

© SAMIR IBRAHIM SHAHEEN

B.Sc.(Hons.) 1971, M.Sc. 1974, Electrical Engineering,

Cairo University, EGYPT.

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Electrical Engineering Department
McGill University
Montreal, CANADA.
1979.

ABSTRACT

This thesis proposes a new paradigm for a modular computer vision system for segmenting and interpreting real world pictures. The system is both data directed and knowledge based. This paradigm exhibits the properties of extensibility, modularity, and model/analysis separability. The system consists of two associative memories and several independent processes. The first, a SHORT TERM MEMORY (STM), maintains the current data and deductions about the scene under analysis. The second, a LONG TERM MEMORY (LTM), is the embodiment of the relevant information (syntactic, semantic and pragmatic) about the image domains to be analyzed. Both memories are designed as relational databases and are completely separate from the analysis processes. These processes have access to both memories using a special communication language which is designed as a relational algebraic sublanguage. A process can only interact with another through the STM. The concept of competition and cooperation is adopted as the system's basic processing strategy. It is implemented as a continuous relaxation labelling process which simultaneously segments and interprets the input picture. The system is a powerful experimental tool and the results of several tests on a natural scene, as well as recommendations for future development, are presented.

RÉSUMÉ

Cette thèse propose un nouveau paradigme pour un système modulaire de vision par ordinateur, lequel est à la fois dirigé par les données et basé sur des connaissances préalables, pour segmenter et interpréter des images réelles. Ce paradigme possède les propriétés d'extensibilité, de modularité, et de séparabilité modèle-analyse. Le système est constitué de deux mémoires associatives et de plusieurs processus indépendants. La première, une "MEMOIRE à COURT TERME" (MCT), retient les données courantes et les conclusions provenant de l'image analysée. La seconde, une "MEMOIRE à LONG TERME" (MLT) est l'incorporation de l'information (syntactique, sémantique, et pragmatique) ayant trait aux domaines d'images devant être analysées. Les deux mémoires sont conçues commes base de données relationnelle et sont complètement séparées des processus d'analyse. Ceux-ci ont accès aux deux mémoires utilisant un langage particulier de communication qui est conçu comme sous-langage de relation algébrique. Deux processus ne peuvent dialoguer entre eux que par l'entremise de la MCT. Le concept de la compétition et cooperation est adopté comme stratégie de traitement de base du système. Il est realisé en tant que procédé continu d'étiquetage par relaxation lequel segmente et interprète simultanément l'image d'entrée. Le système est un outil expérimental puissant et les résultats de plusieurs essais sur des paysages sont presentés, ainsi que des recommendations pour des développements ultérieurs.

ACKNOWLEDGEMENTS

I sincerely thank my thesis advisor Prof. M.D. Levine for his continual supervision, helpful suggestions, the hours of discussions and critical review of the manuscript.

I am also grateful to Dr. S. Zucker for his constructive comments and kind help.

I would like to acknowledge the efforts of D. Kashtan, R. Quik, J. Leemet and D. Ting for their assistance with the computer system (both hardware and software). My colleagues, the other graduate students in the Computer Vision and Graphics Laboratory, in the Electrical Engineering Department, are remembered for their concern and constant encouragement. In particular, I would like to mention A. Nazif for his friendly assistance and Y. Leclerc for translating the abstract.

Thanks are due to the Computing Centre, McGill University, for allowing me to use the MUSIC SCRIPT facility in preparing the manuscript.

Most of all, I wish to thank my wife whose support and patience throughout this study has been extraordinary. I

also acknowledge the long hours she spent in typing and editing the final draft.

This work was partially supported by the National Science and Engineering Research Council of Canada under Grant No.A4156, by the Department of Education, Province of Quebec under Grant No.EQ-633, and by the Fellowship program at McGill University.

Table of Contents

	page
ABSTRACT	i
RESUME	ii
ACKNOWLEDGEMENTS	iii
Table of Contents	v
Table of Figures	ix
CHAPTER I: INTRODUCTION	
1.1 RESEARCH PROBLEM	1
1.2 SYSTEM OVERVIEW	5
1.3 SURVEY OF PREVIOUS RESEARCH	16
1.4 ORIGINAL CONTRIBUTIONS	49
1.5 THESIS OVERVIEW	53
CHAPTER II: SHORT AND LONG TERM MEMORIES: A RELATIONAL DATABASE APPROACH	
2.1- INTRODUCTION	55

2.2- WHAT IS A DATABASE? 64

2.3- WHY A RELATIONAL DATABASE? 68

2.3.1 THE HIERARCHICAL DATA MODEL 70

2.3.2 NETWORK DATA MODEL 72

2.3.3 RELATIONAL DATA MODEL 74

2.4- DATABASE NORMALIZATION 81

2.5- RELATIONAL ALGEBRA:
A DATABASE SUBLANGUAGE 93

2.6- SHORT AND LONG TERM MEMORIES:
A RELATIONAL MODEL 115

CHAPTER III: SHORT TERM MEMORY

3.1- INTRODUCTION 120

3.2- REGION MAP RELATION 121

3.3- REGION FEATURES RELATION 123

3.4- SPATIAL RELATIONS 128

3.5- CHAIN CODE RELATION 136

3.6- REGION INTERPRETATION RELATION 140

3.7- OTHER STM RELATIONS 142

CHAPTER IV: LONG TERM MEMORY

4.1- INTRODUCTION 144

4.2- OBJECT FEATURES RELATION 147

4.3- DISTINGUISHING FEATURES RELATION 152

4.4- PART-OF RELATION 155

4.5- CONSTRAINTS RELATION 156

4.6- CONDITION-ACTION RELATION 161

CHAPTER V: SYSTEM PROCESSES AND CONTROL

5.1- INTRODUCTION 166

5.2- LOW LEVEL PROCESSOR 171

5.3- FEATURE ANALYZER 179

5.4- HYPOTHESIS INITIALIZER 181

5.5- COMPETITION AND COOPERATION 185

5.5.1 HISTORICAL BACKGROUND 185

5.5.2 HYPOTHESIS VERIFIER/INTERPRETER 193

5.6- FOCUS OF ATTENTION 208

5.7- SCHEDULER 219

CHAPTER VI: RESULTS AND CONCLUSIONS

6.1- INTRODUCTION 225

6.2- LOW LEVEL SEGMENTATION 227

6.3- HYPOTHESIS INITIALIZATION 232

6.4- OVERALL SYSTEM PERFORMANCE 250

6.5- EXPERIMENTS WITH DIFFERENT VERIFICATION STRATEGIES 263

6.6- CONCLUSIONS AND FUTURE RESEARCH 280

APPENDIX I: SHORT TERM MEMORY RELATIONS 286

APPENDIX II: LONG TERM MEMORY RELATIONS 303

REFERENCES 316

Table of Figures

		page
Figure 1.1	General system structure	8
Figure 1.2	System block diagram	12
Figure 2.1	System S1	56
Figure 2.2	System S2	56
Figure 2.3	Example of a segment of a database for a computer vision system	67
Figure 2.4	Simplified view of a database system	69
Figure 2.5	Example of the hierarchical model	71
Figure 2.6	Example of the network model	73
Figure 2.7	Example of the relational model	76
Figure 2.8	Example of a relational model	76
Figure 2.9	Relation normalization	78
Figure 2.10	Representation of functional dependence	83
Figure 2.11.a	The relation S	85
Figure 2.11.b	Functional dependence of relation S	85
Figure 2.12.a	Relations S1 and S2	89
Figure 2.12.b	Functional dependence of relations S1 and S2	89
Figure 2.13.a	A relation T in 1NF	91
Figure 2.13.b	Functional dependence of relation T in 1NF	91
Figure 2.13.c	Relations T1 and T2 in 3NF	91
Figure 2.13.d	Functional dependence of relations T1 and T2 in 3NF	91

Figure 2.14 Examples of relational algebraic operations 99

Figure 3.1 Binary image of a region and the position parameters calculated for it 125

Figure 3.2 Illustration of the fuzziness of a spatial relation LEFT-OF 130

Figure 3.3 Eight possible quantized directions for chain code representation 137

Figure 4.1 Example of PART-OF relation 155

Figure 4.2 Sample of part of the CONSTRAINTS relation 158

Figure 5.1 Adaptive threshold versus the ratio of standard deviation and average value of feature q over region R 177

Figure 5.2.a Regions for illustrating the updating process 198

Figure 5.2.b Relation that contains the constraints involving the objects SKY and ROOF 198

Figure 5.3.a Example of a REGION MAP 213

Figure 5.3.b The REGION STACK changes after each iteration of the region updating process 213

Figure 5.4.a The FOCUS OF ATTENTION without the use of the QUEUE 215

Figure 5.4.b The FOCUS OF ATTENTION with the QUEUE of length 4 215

Figure 5.5 Example of the FOCUS OF ATTENTION with a QUEUE of length 4 217

Figure 6.1 A typical real outdoor scene 226

Figure 6.2 REGION MAP for figure 6.1 with regions arbitrarily coloured 228

Figure 6.3 Illustration of the region boundaries with reference to the original picture 229

Figure 6.4 Region boundaries with different threshold values and clean up parameter equal to 8 231

Figure 6.5.a Initial first best interpretations and associated confidences for the REGION MAP in figure 6.2.b when colour and position are used as distinguishing features and position was given a veto power 235

Figure 6.5.b Initial second best interpretations and associated confidences for the REGION MAP in figure 6.2.b when colour and position are used as distinguishing features and position was given a veto power 236

Figure 6.5.c Initial third best interpretations and associated confidences for the REGION MAP in figure 6.2.b when colour and position are used as distinguishing features and position was given a veto power 237

Figure 6.6.a Initial first best interpretations and associated confidences when colour is used as the only distinguishing feature 239

Figure 6.6.b Initial second best interpretations and associated confidences when colour is used as the only distinguishing feature 240

Figure 6.6.c Initial third best interpretations and associated confidences when colour is used as the only distinguishing feature 241

Figure 6.7.a Initial first best interpretations and associated confidences when colour and position are used as distinguishing features with equal weight and no veto 243

Figure 6.7.b Initial second best interpretations and associated confidences when colour and position are used as distinguishing features with equal weight and no veto 244

Figure 6.7.c Initial third best interpretations and associated confidences when colour and position are used as distinguishing features with equal weight and no veto 245

Figure 6.8.a Initial first best interpretations and associated confidences with relaxed SKY features 247

Figure 6.8.b Initial second best interpretations and associated confidences with relaxed SKY features 248

Figure 6.8.c Initial third best interpretations and associated confidences with relaxed SKY features 249

Figure 6.9.a Final first best interpretations and associated confidences using the initialization of figure 6.5 254

Figure 6.9.b Final second best interpretations and associated confidences using the initialization of figure 6.5 255

Figure 6.9.c Final third best interpretations and associated confidences using the initialization of figure 6.5 256

Figure 6.10 Region boundaries after merging as a result of figure 6.9. The number of regions is 71. 260

Figure 6.11 Illustration of the PATH for region verifications 262

Figure 6.12.a Initial first best interpretations and associated confidences for the new REGION MAP of figure 6.10 264

Figure 6.12.b Initial second best interpretations and associated confidences for the new REGION MAP of figure 6.10 265

Figure 6.12.c Initial third best interpretations and associated confidences for the new REGION MAP of figure 6.10 266

Figure 6.13.a Final first best interpretations and associated confidences using the initialization of figure 6.12 267

Figure 6.13.b Final second best interpretations and associated confidences using the initialization of figure 6.12 268

Figure 6.13.c Final third best interpretations and associated confidences using the initialization of figure 6.12 269

Figure 6.14 The resulting region boundaries after the analysis. The number of regions is equal to 58 270

Figure 6.15 A manually created REGION MAP 272

Figure 6.16 Final interpretations and associated confidences for the example in figure 6.15 273

Figure 6.17.a Initial first best interpretations and associated confidences for the REGION MAP in figure 6.15 274

Figure 6.17.b Initial second best interpretations and associated confidences for the REGION MAP in figure 6.15 275

Figure 6.17.c Initial third best interpretations and associated confidences for the REGION MAP in figure 6.15 276

Figure 6.18 Variation of information measure I(T) for two different queue lengths 278

Figure 6.19 Variation of information measure I(T) by processing in parallel, sequentially and with queue length equal to 5 278

CHAPTER I

INTRODUCTION

1.1 RESEARCH PROBLEM

Computers are playing an increasingly important role in the development of our society. The last decade exhibited a pressing need for computers to analyze and process pictorial data. Applications of computer vision have included the analysis of satellite pictures in remote sensing, automated production lines in industry, quality control in paper production, cell analysis in medicine, domestic applications in traffic control, mail sorting, and robotics.

In the early fifties, researchers were primarily concerned with the problem of classifying two dimensional images (patterns) into a small number of categories (e.g. alphanumeric characters) [30]. The study of robots became an area of interest in the early sixties and in a few years several research laboratories had produced systems capable of performing simple tasks and intelligently executing plans of action based on an internal world model [81]. However, all of these were trained and designed to implement simple

tasks or to navigate in a simple environment of rooms, and corridors, and to deal with large uniformly coloured blocks. At the same time, research was active in developing programs to segment and interpret line drawings of the artificial world of blocks [93]. Several systems were designed to interpret line drawings of complex scenes of planar surface objects. Many researchers argue that this concentration of work on the blocks world has actually hindered progress in computer vision because "it has allowed workers to avoid facing certain key problems associated with domain semantics, distinguishing features of complex objects and new representational schemes" [81]. By the early seventies a new trend could be detected towards analyzing real world pictures. Several systems were developed to analyze images from specific domains such as human faces [56], city scenes [82], road scenes [127], office scenes [37], industrial parts [47, 128], and radiographic images [10, 45, 111]. However, most of these systems were designed to work with a specific world model and there was no means of modifying them to analyze other types of scenes, or even to test the effect of other heuristics in the analysis. Their use of semantic information was limited to the specific class of pictures to be analyzed and could not be used in other domains.

In the last few years, the need for a more general system has evolved in the research community [8, 12, 42, 43, 44, 54, and 66]. Researchers are attempting to develop new representation schemes and perceptual strategies that are not limited to a specific image domain. These systems can thus deal with a wide variety of real world pictures. They would also use knowledge (syntactic, semantic, and pragmatic) from all available sources in their processing strategy. The work of the vision group at McGill University is concerned with the development of such a computer vision system. In this dissertation, I will present my contribution to the development of the system. Our main objective is to segment and interpret real world pictures. This process involves the transformation of the input sensory data into a symbolic description of the scene. The input sensory data is usually represented by three two dimensional arrays of numbers that correspond to the RED, GREEN and BLUE colour scans of the picture. The picture is first segmented into regions containing pixels whose primary features are similar. This low level process reduces the number of individual entities to be analyzed at the higher levels. The resulting segmentation is partial in the sense that region boundaries do not necessarily correspond to object boundaries in the scene under analysis. The problem can now be defined as that of simultaneously merging and

labelling regions in that partial segmentation in order to obtain a complete segmentation. The latter is obtained when the system outlines the objects in the picture and labels them with the appropriate interpretation.

Our work represents the first version of the system, and hence it was primarily concerned with the construction of a framework within which different sources of knowledge could be incorporated. This vision system is built as an experimental tool to investigate different models and different processing strategies. The results can be evaluated by comparing the original picture in figure 6.1 with the final segmentation and labelling shown in figure 6.9. Since the system is intended to be an experimental tool, it is not restricted to a specific class of scenes. Although the current implementation was tested on a real outdoor scene, the model is accessible to the user and can be easily modified in order to analyze other picture domains. It is also intended to be used by different users at different times. In the next section, we will discuss the design considerations maintained during the development of the system and describe its structure and implementation.

1.2 SYSTEM OVERVIEW

Computer vision systems are very complex and require three basic properties for their successful implementation, all very much dependent on each other. The first one relates to the extensibility of the system, from the point of view of adding both model data and control information. Next, we observe that the development of such a system will necessarily require a large number of people working over an extended period of time. Therefore to provide for this, we focus on the second point which is the modularity of both the model knowledge and the different processes in the system. The interpretation strategy, whatever the level of complexity of the analysis, will involve a multitude of interacting, competing and communicating processes. These processes must be organized in a fashion which stresses both the independence of each process as well as the interaction of one with the other. One of the major challenges of the system engineering is how to achieve these seemingly conflicting requirements simultaneously. Finally, the third criterion for the design is the complete separability of the model or knowledge information from the actual analysis computer code. This will enforce a large measure of system generality by requiring the designer to create analysis processes which are independent of the class of pictures

being analyzed. We have concluded that these three aspects are of primary importance in the design of a computer vision system. Therefore, these ideas form the backbone to the design methodology of the system presented in this thesis. It has also become quite clear that all levels of analysis must be integrated initially within the same framework so that the necessary feedback paths can be included in a natural fashion. It is quite unlikely that the different levels of the analysis could be designed as independent and basically very different subprocesses, and that the feedback paths (communication links) could be added after all levels were completed. This has been the primary obstacle that prevented the integration of different processes developed by different people. Each process might have different data structures, processing strategies, and visual model. To integrate these processes, one was forced either to:

i) Rewrite the computer code for each subprocess, or

ii) Develop an interface between different processes, a difficult problem in itself.

For the above reasons, one of our main concerns was the system engineering.

In this thesis we propose a new paradigm for vision which is both data directed and knowledge based. From a philosophical point of view it was motivated by the

HEARSAY II understanding system [33,63] and the writing of Arbib [3]. As will be seen, the paradigm provides for the properties of extensibility, modularity and model analysis separability.

Figure 1.1 is a schematic of the computer vision system. It essentially consists of three different types of units, two of which are memories represented as standard relational databases [22,25]. These are the SHORT TERM MEMORY (STM), which contains the raw data and the most current interpretations and deductions about the picture under analysis, and the LONG TERM MEMORY (LTM), which contains the syntactic, semantic and pragmatic knowledge about the scene under consideration. A collection of independent analysis processes, each of which is specialized for a particular task, can communicate with both the STM and the LTM.

The STM is designed as an associative memory for the immense amount of data contained in the picture, the measurements made on the input picture, and the deductions made by the processes of the system. The primary role of the STM is to act as a buffer or communication channel between the processes of the system as they compete and cooperate to achieve their processing tasks. This memory

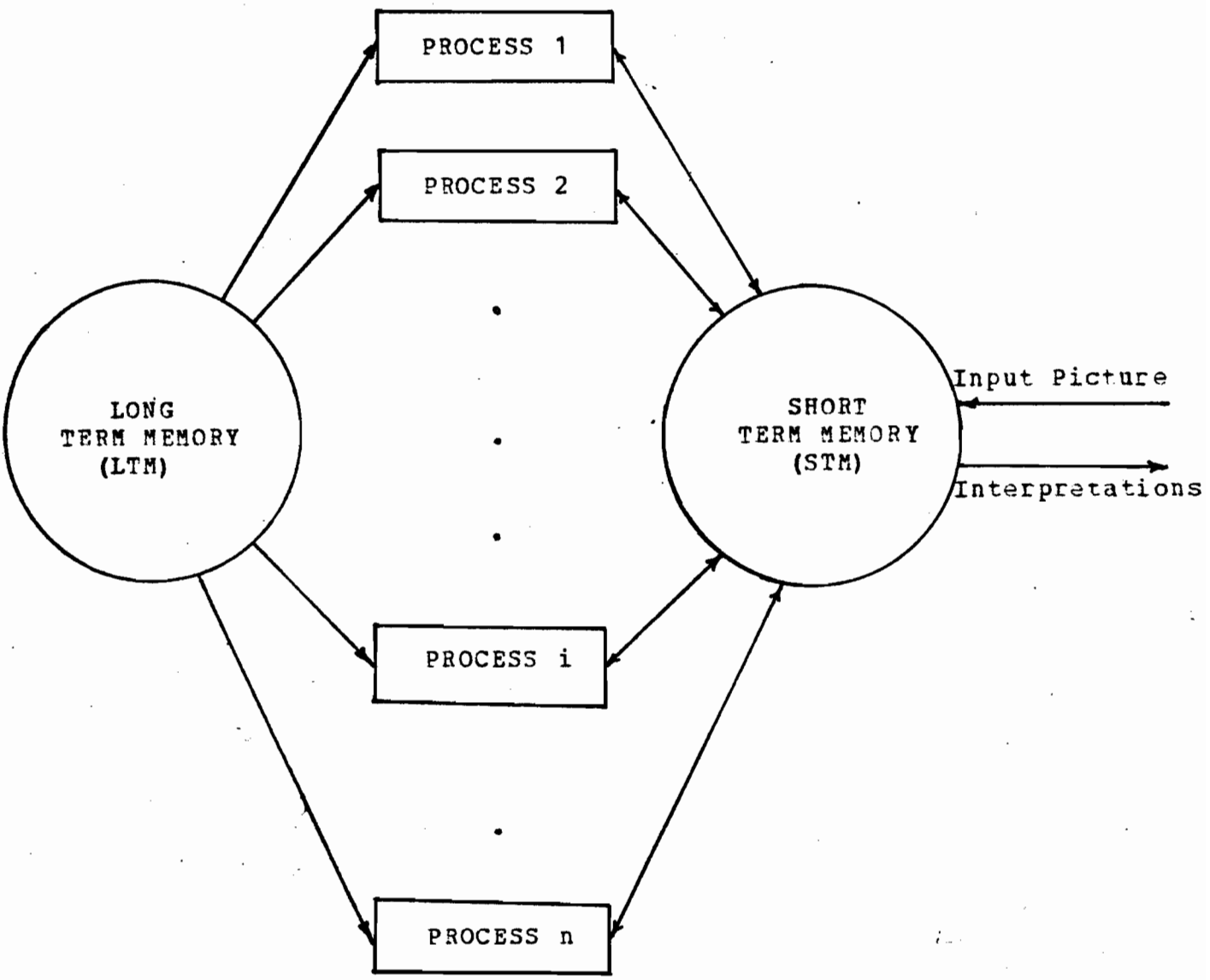


Figure 1.1 General system structure.

facilitates the desired level of interaction by allowing each process to transmit (broadcast) its own results, and to have access to both the information about the scene under study as well as all the deductions made by the other processes. To achieve this, each process need only be aware of the overall STM structure and the communication language to and from the STM. Because of the modularity property, the system is not affected by the previously installed processes or the removal or reformulation of some of them. The STM is designed and implemented as a relational database, and therefore the individual processes are relieved of the task of data access and management. They have to communicate with the database using the appropriate communication language. The communication language is designed and implemented as a relational algebraic sublanguage. The use of the relational database and the algebraic sublanguage eliminates data redundancy which reduces inconsistancies. It enforces the use of a standardized data model for all the processes, thereby enhancing their modularity and independence. The STM also provides an efficient and simple accessing mechanism for data updating and retrieval. Conceptually the data in the STM may be represented at several levels of abstraction. For example, the STM will contain the three colour image planes, edge and line information, atomic regions as well as

clusters of merged regions, and of course their descriptions and interpretations. Moreover, it maintains a trace of the activities and decisions made by the processes of the system during the analysis.

The LTM is the embodiment of the relevant information (syntactic, semantic and pragmatic) about the class of scenes to be analyzed. It is completely separate from the program code used to implement the analysis. It is also implemented as an associative memory, using the same relational database formulation. Thus it is simple to alter the model information, thereby providing an attractive and convenient method for experimenting with different models. Indeed this simplicity is absolutely necessary in order to design an appropriate model for a particular interpretation task. The ensuing modularity will allow us to study the effects of alternative models and system processes. In the current implementation, we have considered natural outdoor scenes that contain objects such as SKY, TREE, ROOF, WALL, GRASS, etc. We have also assumed that the objects in the scene are at relatively large distance from the viewer, and therefore they can be represented in terms of two dimensional models. In this sense, the scene domain is a subjective 2 1/2 D space.

Since the initial segmentation is assumed to be partial, we are always proceeding on the basis of a very fragmented picture. Hence model information that refers to the object's geometric shape or other such properties related to an appropriate complete segmentation are not applicable to the level of analysis under study in this thesis. Thus the visual model represented in the LTM at this stage is restricted to information which is meaningful to segmented parts of the objects. For example, spatial constraints that describe the relative position of two adjacent objects are extremely useful in the present level of analysis.

The system is designed so that processes may be added incrementally with ease. Currently, the system contains six processes, as shown in figure 1.2. The picture data are input to the STM and are subsequently analyzed by the LOW LEVEL PROCESSOR which organizes them into basic entities more suitable for further analysis. Generally, these entities might be edges, lines, closed boundaries, atomic regions and clusters of similar atomic regions. In the current implementation, this process is a very simple segmentation algorithm which groups pixels according to their values in three dimensional colour space, using an adaptive threshold technique to be described in section 5.2. The resulting region segmentation is transmitted in the form of a REGION MAP to the STM and these regions are the

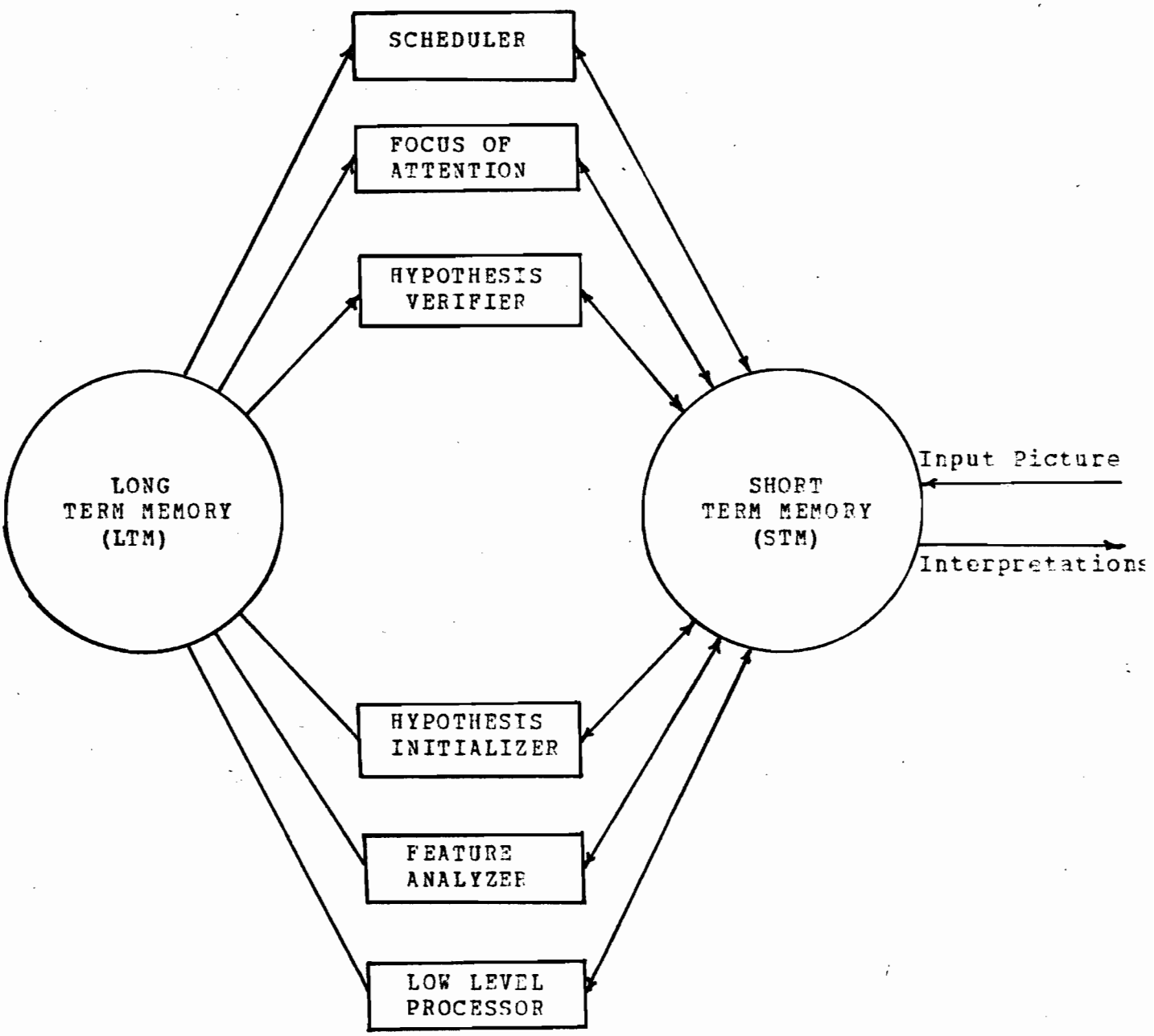


Figure 1.2 System block diagram.

entities which are manipulated in order to reach an interpretation for the scene. Because region splitting is not permitted, it was necessary to make this segmentation process as conservative as possible. We should stress that the system always assumes that the resulting region segmentation is partial. The output of the LOW LEVEL SEGMENTATION is a REGION MAP which resides in the SHORT TERM MEMORY.

The FEATURE ANALYZER computes a set of attributes for the segmented input picture and transmits the result to the STM, to be available to the other processes of the system. Obviously, it is activated to compute the features of the regions in the initial segmentation (i.e. REGION MAP). However, it is also activated whenever regions are merged and a new set of attributes needs be computed.

Three processes form the heart of the computer vision strategy presented in this thesis. One is the HYPOTHESIS INITIALIZER which takes as its input the region descriptions as they appear in the STM, and uses the model information in the LTM to generate all possible interpretations for the regions in the STM. Generally the LONG TERM MEMORY contains information about the intrinsic features of the objects, such as their possible colours and relative position. Thus

each region will be associated with a set of interpretations and their associated confidences. These confidences represent the certainty of labelling the regions with the corresponding interpretation. Again these results are transmitted to the STM and can be used by any of the other processes.

The process which is primarily concerned with region interpretation is the HYPOTHESIS VERIFIER/INTERPRETER. In addition to the scene descriptions in the LTM, this process uses the STM information in the form of the region features and the structural relationships between the regions, as well as the current set of hypotheses about each region and their associated confidences. Its basic function is to verify and update each interpretation by altering its confidence level. This process adopts competition and cooperation as a basic paradigm for its processing strategy. It is implemented as a continuous relaxation labelling process [131].

The FOCUS OF ATTENTION is intended to work in a cycle consisting of situation recognition followed by an action or set of actions. This process controls the order of verifications of the regions by bringing appropriate regions to the top of a REGION STACK stored in the STM. It is this

top region that will usually be examined by the HYPOTHESIS VERIFIER/INTERPRETER. The FOCUS OF ATTENTION might also be used to implement PLANS [66, 106] which are model dependent, as will be discussed in section 5.6.

The last process is called the SCHEDULER which monitors the current state of analysis of the system by monitoring the data in the STM. It is able to decide which process should be activated at any given time. We note that the design considerations require the system modules be independent and that they interact only by transmitting their results to the STM. No direct transfer of results is allowed between the different processes, a property which enforces the desirable level of independence. With regard to processes activation, two possibilities present themselves. One is to allow each process to monitor the STM for changes in the status of the analysis which may interest it and thereby allow each process to initiate itself based on its own measurements. This tends to be a very costly computation. An alternate approach is to create a special process, the SCHEDULER, which monitors the status of the STM for all the processes of the system, attempting to identify an activity appropriate to the initiation of a particular process. In this way, it acts as a measurement activation module for all the processes collectively. This allows for

a more efficient use of the computer resources while not negating the possibility of parallel processing.

We have briefly presented the overall system structure and the processes used in the analysis. In the next chapters, we will describe in detail each of these processes and the implementation details of both the STM and LTM. We will also analyze the results of the experiments performed on a real outdoor scene. First, we will present a brief history of the related research work in scene analysis.

1.3 HISTORY OF PREVIOUS WORK

Since the 1950s researchers have been concerned with the problem of interpreting visual data by computer. Early work in the fifties concentrated on designing systems that could classify two dimensional images into a small number of categories, for example, alphanumeric characters. In fact, most of the research during this period was concerned with the subject of Pattern Recognition and it still continues as an active research area today. Since the work of Roberts in 1963, research in vision has been centered on the problem of interpreting and describing complex three dimensional

scenes. In fact, Roberts' work set the tone for practically all the "blocks world" (world of prismatic solids) systems which followed it. Working on scenes that consist of planar surface objects, Guzman (1968), Falk(1970), Winston (1970), Waltz (1972), Shirai (1972), Grape (1973), and others have helped to establish a wide range of techniques to convert the input line drawings or raw image data into symbolic descriptions based on lines, regions and simple objects.

In parallel to the above, much of the early research on real pictures concentrated on the development of techniques for edge detection and region growing. These techniques are reviewed in Nazif [78], Zucker [136], and Riseman and Arbib [88]. Several systems were developed to analyze scenes in a specific image domain such as Kelley (1971), Bajcsy (1972), Tenenbaum (1973), Yakimovsky (1973), Liberman (1974), and Ohlander (1975). The work in this thesis is an outgrowth of this type of research.

In this section we present a brief review of earlier work pertinent to our research. We do not intend this to be an exhaustive survey, but hope that it will provide some insight to the interested reader into the basic research approaches and techniques. For a complete view of the field of computer picture processing the reader is directed to

Rosenfeld's surveys [94, 95, 97, 98, 99, and, 100].

Roberts 1963:

The history of computer vision begins with Roberts [92, 93] who was the first of many researchers to deal with the so-called blocks world. His work covered the complete spectrum, from camera input to three-dimensional description and display of planar surfaced objects. He developed a system that constructed a three dimensional model of simply connected polyhedra scenes under ideal conditions. Starting with the digitized input picture, he used some fairly simple preprocessing heuristics in an attempt to construct a connected line drawing. Because of the difficulty in achieving this goal, he wrote his program to use as input a perfect line drawing. The program then proceeds by finding all well shaped regions (polygons), and attempts to match these regions with a pre-stored three dimensional model. The matching is performed in a number of steps, with each one of the following steps performed only if the previous step fails:

1- using polygons around a vertex.

2- using polygons surrounding a line.

3- using a polygon and a third line from one vertex.

4- using a three line vertex.

From one of the previous steps, the program locates point pairs between the picture and a pre-stored finite set of models (parallelopiped, wedge and hexagon). Given the key match, Roberts applies a similarity test to find the best transformation and a mean square error of fit. If the error is less than some threshold, the transformation is accepted, the recognized object is deleted from the scene, and the program proceeds to test the remaining objects in the scene.

Roberts' treatment of composite objects is interesting in that he interprets complex objects as conglomerates of instances of his simple models. Then, if a single object is recognized, it is deleted from the scene. In this way complex objects were interpreted by abutting the three simple basic models. This fact is remembered when projecting the object back to any position and under any rotation. He assumed all objects to be upright and touching the ground plane. Therefore if the camera model is known, absolute depth can be determined.

Roberts' work constitutes an important initial effort for a complete vision system from preprocessing through recognition to display of the final output. Its weakness lies in the necessity of perfect line drawings for the matching stage. This in turn necessitates an almost perfect

camera input so that the simple heuristics used can obtain the perfect line drawings required. Another weakness of Roberts' technique is that the program seems dependent on the set of models used, whence, the incorporation of a new model would require considerable program changes. Much of the research that followed Roberts' sought to improve on these weaknesses.

Guzman 1968:

Guzman's work [40, 41] demonstrates that quite complex scenes, assuming perfect line drawings without shadows or other irregularities, may be interpreted as the projection of three dimensional objects, using a rather limited set of local heuristics based on the properties of vertices. His "SEE" program creates a network of linked regions on a graph, where the nodes correspond to the regions (a region is a surface bounded by simply closed curves) and the adjacency relations correspond to the links. Using a set of chiefly local heuristics that are based on the properties of the vertices, weak or strong links may be established between adjacent regions. Linkages may be inhibited by neighbours of certain types. Regions are subsequently grouped together into nuclei according to the nature of their associated links. The linked regions, known as nuclei

are further grouped and the final nuclei constitute the body interpretations.

The main weakness of Guzman's program was that it is sensitive to imperfections in the line drawings. If lines are missing or incomplete, the program will sometimes either reject the scene if it contains isolated line-ends, or will fail and clump together several objects due to missing exterior lines. SEE was able to successfully analyze very complex scenes, but these were abstract and did not use real data. The results are not in this sense immediately and explicitly relevent to the practical problem of computer vision.

Rattner, M.H. 1970 and Orban, R. 1970:

Rattner, Orban [83, 86] and others at MIT modified Guzman's SEE program by analyzing the consistency of the interpretations using a global graph structure. At the same time Orban [83] developed some heuristics that could be used in the preprocesing stages to eliminate shadows which divided regions into segments.

Winston, P.H. 1970:

Winston [124] used Guzman's SEE program to analyze scenes of perfect line drawings and to learn structural

relations between objects and groups of objects from presented examples. Simple scenes were used to create models of simple block configurations. These models were later used to identify other scenes by matching the scenes with a known model or series of models. He developed a structural description of a scene which enhances the crucial features composing it, but yet was sufficiently flexible for the addition of explicit details in any given scene. Flexibility was accomplished through the use of a network type data structure which enables the formation of conceptual models by confining simpler concepts or relations between certain types of objects. Explicit differences were found by attempting to transform the scene description into another model which preserves the essential structure. This was probably the first example of learning of scene structure in computer vision.

Huffman 1969, Clowes 1971:

Huffman [49] and Clowes [18] devised a labelling scheme which attaches a unique label to each vertex that may be found in a perfect shadowless line drawing of scenes of tetrahedral objects. The labelling scheme constrains the number of possible interpretations of a vertex. These constraints may be propagated to adjacent verticies, which limits the number of interpretations of a face. By

systematically analyzing a labelled picture, one is able to decide whether the object is legal and physically realizable in three dimensional space. Their work introduced a theoretical structure to the prior ad hoc techniques that were implemented.

Falk 1970:

Falk [34] at Stanford University utilized Huffman's labelling scheme to develop a program that interprets imperfect line drawings as a three-dimensional scene. He devised heuristics for body separation, which work for more general scenes than those of Guzman. The program uses the vertex labels to form links between lines, whence the bodies are defined in terms of such links. Following body segmentation, simple heuristics were used to determine occlusion relations over the scene, and the separated bodies are completed as far as possible based on collinearities and extension vertices.

Falk used the hypothesis of ground plane support introduced in Roberts' program and extended it to the hypothesis of objects supporting objects (flat on top). He also used nine fixed-size models (cube, right rhomboid, L-beam, four parallelopipeds with different length relations and two wedges) in the recognition part of the sytem. In

short, Falk's program is related to both Roberts' (the use of models) and Guzman's (the implementation of body segmentation). In comparison to Guzman's scheme, it is less sensitive to missing lines and imperfections in the drawings. This is due to the fact that his segmentation procedure is based on links among lines rather than between regions. However, it shares Guzman's sensitivity to shadows and noise, which is due to the fact that segmentation and recognition were performed as unrelated processes.

Waltz 1972:

Waltz [122] also used and extended the vertex labelling scheme introduced by Huffman [49] and by Clowes [18] to include and utilize shadows in the process of scene segmentation. First, the vertices in the scene are labelled by all their possible interpretations, given that the scene consists of convex trihedral objects. Lines are assigned labels based on the vertex labels. These line labels cover most of the possible edge interpretations in 3-D scenes, such as convex, concave, boundary, as well as the lighting conditions on the sides.

Waltz's filtering program is applied to check the inter-consistency of the two sets of vertex labels for each line. It deletes vertex labels whenever the line labels

cannot agree (inconsistent labels). In a surprisingly large number of cases, unique labels are assigned, and if the labelling is not unique, a full tree search for consistency is performed where inconsistent labels are deleted. The resulting labelling determines the segmentation. In the cases of non-unique labellings, the program gives more than one segmentation which in a sense is a strong point of the program. Waltz also provided some facilities for dealing with missing lines by including such a special case in his labelling scheme. His program represented a new approach, because it utilizes the entire context and the scene is interpreted as a whole. However, the specification of the labelling made it error sensitive to imperfections in line drawings. Furthermore, the computation time depends on the order in which the junctions are visted. The filtering program of Waltz spurred researchers to investigate whether the method could be generalized, ultimately leading to the introduction of continuous relaxation labelling in computer vision [131, 132, 133, 134].

Shirai 1972:

Systems discussed so far were based on the assumption of essentially perfect line drawings or almost perfect line drawings with or without shadows. These programs are

hierarchical in the sense that they have a pre-specified hierarchy for their processing strategies. Shirai, in contrast, [109] selected a heterarchical control structure. He constructed a program that attempts to find bodies in a scene, working directly on the digitized picture. His program was based upon the strategy of recognizing objects step by step, each time making use of previous results. The contour lines which separate the bodies from the background are first found, followed by other boundary lines which separate two bodies. In searching for internal lines (lines which lie at the intersection of two faces of the same body), the program may employ clues obtained from the contour and boundary lines. These clues are used to propose the existence of a line and to determine the search area. The process is heterarchical and is based on apriori knowledge of the domain (i.e. the general concept of body as a guide). This is in contrast to hierarchical schemes, where successively higher levels of abstraction are formed in a pre-specified sequence of a hierarchy of heuristic processes.

More specifically, the program attempts to find lines at concave junctions in the contour, or in other places. If a proper line segment is found, the end of the line is determined by tracking along the line, looking for vertices

or extensions, determining the implication of the findings as far as the concept of an object goes. The program does not work with shadows and is not sufficiently general for concave objects. As opposed to hierarchical schemes whose major deficiency is that a single obscure feature or noise might throw the whole process off, Shirai's system demonstrated that a computationally simple heterarchical process can achieve better results.

Grape 1973:

Grape [38] has developed a system that utilizes global information, in the form of two dimensional models, (prototypes) to interpret an image as the representation of three-dimensional scenes. His system consists of three stages, namely: preprocessing, parsing and object completion. In the preprocessing stage, he used the edge follower of Pingle and Tenenbaum [85] to obtain the edges in the picture, followed by a least square method that fits lines to edges. The resulting line drawing, which is not necessarily complete, is now parsed in a recursive process, each time finding the best possible match between elements of the scene and some prototypes, isolating that mapping, modifying the scene (by removing the lines), and iterating. The process is ended when there are no more possible links (mappings). The established mappings are then investigated

in decreasing order of complexity until a complete object is found or the links are exhausted. In the final stage, objects are completed by considering lines which were not accounted for, with partially matched objects. While the previously cited work relied upon perfect input data, Grape's sytem performs well in the presence of noise, shadows, glare and missing lines. Planer faced convex objects were used exclusively. His chief contribution is the use of global models to guide locally based decisions in parsing scenes.

Kuipers 1974:

Based on the idea of frames as proposed by Minsky [72], Kuipers [60] developed a system for recognizing objects in the blocks world. He used the concept of a global frame which, in this case, consists of detailed hypotheses about the line structures of the domain under analysis. The domain here is the block world. Basically, the system attempts to verifiy the set of assertions made about the scene. A procedural formalism was used and it was based on a transition network. Emphasis is justly placed on the programming of a control structure that is logically able to verify the scene based on the embedded knowledge.

Although the work in the blocks world lead to a variety of techniques and added to our knowledge about the visual world, many researchers contend that it has allowed them to avoid facing the problems associated with the everyday visual world. Tenenbaum [103] has suggested that the blocks world "may also have impeded progress by diverting investigators to difficult but less important problems attributable to its very sterility", because "the lack of variety and of meaningful contextual constraints in this artificial world limited the extent to which perceptual strategies could capitalize on task goals and expectations; exhaustive descriptions were invariably sought. Moreover, the unavailability of attributes like color and texture led to a preoccupation with articulated boundary descriptions (i.e. shape) as the principle bases for recognition". Realizing that, a gradual shift of research to the analysis of complex natural scenes can be noticed in the early seventies. A fundamental requirement for these systems is its ability to segment and identify the objects in a digitized real image. Thus a specific processing hierarchy was imposed. That is, systems start by segmenting the image into regions, then proceed to identify the objects in the picture. A variety of region growing algorithms were designed [136].

Kelley 1970:

Kelley's work [55, 56] is an interesting example of context dependent segmentation. His work was concerned with the visual identification of human faces, that is, finding the locations of features such as eyes, hair, nose or shoulders in the image and then classifying the image on the basis of the measurements of distances between pairs of such features. His work has two interesting aspects. The first is the use of planning to reduce the search space. He obtained a reduced picture by applying an averaging operator over the original image. Since the averaging process reduces the amount of noise in the original image, objects can now easily be identified in the reduced image. These objects can then be used as a plan to verify the existence of edges in the original picture. The idea of planning was later expanded in the work of Tanimoto and Pavlidis [110], Levine (PYRAMIDS) [66], and Riseman and Hanson (PROCESSING CONES) [43, 89].

The second interesting aspect of Kelley's work is his use of semantic information to guide the search for objects. Thus the program would search for a basic object that can easily be detected (e.g. head in face recognition). Once one part is found, the program uses this knowledge in its search for other parts that bear a certain relation to the

first part (e.g. the eyes). Hence, he was able to use a model of the desired objects to direct and drive the search process.

Bajcsy 1972:

Bajcsy's research could be considered as the only significant work that utilized texture analysis to segment natural scenes [7]. The local descriptors are defined over windows of different sizes and are obtained from the analysis of the power spectrum.

Yakimovsky and Feldman 1973:

Yakimovsky and Feldman [126, 127] developed a vision system that utilizes a probabilistic semantic base to segment natural scenes. First, sample points are selected from the input picture which are considered to be representative of the different regions. A non-semantic region grower melts two adjacent regions if the difference between their feature vectors is less than some conservative threshold. At this stage, boundary strength is based on the feature vectors of the two regions, differences between sample points, the length of the boundary and the size of the two regions. This first region grower stops when the weakest boundary strength surpasses a pre-specified threshold.

The next stage of the analysis uses a probabilistic world model. In a learning phase, the statistical model is input to the system. For each region a probability estimate is formed given the feature values of the region. This, of course, requires that the apriori probability distribution for the various objects be given in the model. The boundary strength is calculated as the Bayesian probability that the boundary separates sub-parts of images of different objects based on the properties of the boundary and the two regions defining it. The merging process continues by melting the weakest boundary until the weakest boundary surpasses some threshold or a good interpretation for the current segmentation is reached. The system represents the first unified attempt in analyzing outdoor scenes and was shown to work quite well on two pictures.

This work demonstrated the importance of using both syntactic and semantic information, with each aiding the other, to form the regions. It also illustrated how segmentation and interpretation can be performed simultaneously by using the probabilities of region interpretations to guide the merging process. However, these probabilities of the regions' interpretations were estimated based only on measurements done on each individual region. Also, the probability that a boundary between two

regions separates two different objects was estimated based purely on the local properties of the boundary, and did not include all of the adjacent regions. Thus the merging and interpretation process was exclusively based on information from a very limited neighbourhood around the boundary between the two adjacent regions. In comparison, we have utilized a more global approach because we merge regions and update their interpretation based on the evidence provided by all the regions in the local neighbourhood and not just on the local boundary information.

Liberman 1974:

Liberman [69], as a student of Bajcsy, developed a system that is able to identify a limited number of objects in a simple outdoor scene. His main contribution is the use of complex texture analysis to obtain depth cues. Thus, he was able to make some three-dimensional inferences about the objects in the picture. For example, he used the texture gradient to identify the ground with a large meadow. He also used semantic nets to model the objects and their relationships.

Ohlander 1975:

Ohlander [82] has made an extensive study of the use of multidimensional histograms as a mechanism for segmenting

natural scenes. The cutoff values are obtained from the histograms of multiple sensory parameters. He utilized a total of nine dimensions, namely the three colour systems (RGB, IHS and IQY). The thresholding operation is applied in a recursive control structure to isolate and refine segments of the picture. In each case, the histograms are derived for the largest unprocessed portion remaining in the image. Thus in each thresholding operation, nine histograms are examined to determine the most sharply defined feature. Thresholding on a limit provided by the minimum bounding the best peak will thus produce clusters of points which are uniform for the selected feature. These clusters (regions) are then isolated and eliminated from further consideration. This in turn will enhance the effectiveness of other features. That is, features that were previously obscured would now become more distinct. Textured areas were removed using a simple texture measure based on the density of edges. Ohlander's recursive analysis was quite effective in the segmentation of several scenes and was similar to the independent work of Tomita, Yashida and Tsuji [118] which appeared at approximately the same time.

Although histogram analysis provides a global view of the data in the feature space, it does not reflect the spatial relationships in the image space. Thus the

formation of clusters in the feature space does not take into consideration the spatial distribution of pixels in the image [43]. In addition, Ohlander made no attempt to label these regions with interpretations.

Tenenbaum et al 1973-1976:

Instead of exhaustively analyzing the scene, Tenebaum et al [112, 114] designed a system to find a specific object needed by a robot in the performance of a task. This approach was based "on the premise that in real scenes there is sufficient redundancy of perceptual clues, as well as contextual constraints among objects, so that an intelligent system can devise a relatively simple strategy for distinguishing the specific objects of interest from others likely to be present" [112]. Multidimensional data (colour and range) is used to increase the likelihood of finding a distinguishing set of features for a particular object.

The search for an object proceeds in two phases, called acquisition and validation. The objective of the acquisition phase is to rapidly disqualify obviously irrelevant areas of the scene so that subsequent search can concentrate on the most likely locations for the goal object. In this phase, the image is sampled for

characteristic attributes of the desired object based on a pre-defined model. If planning is good, the irrelevant areas of the scene will rapidly be disqualified, and thus the system will concentrate only on the most promising locations. The features are tested at each sample in an order of the most discriminatory features first. In this process, contextual clues are used to direct the search. For example, sampling may be localized in the vicinity of known objects such as tabletop which is normally found within a certain height from the ground floor. In other cases, it may be simpler to look for an easily recognizable object that maintians a known relationship to the desired object. For example, telephones are small and known to lie on desks.

The validation phase can be considered an extension of the acquisition testing sequence to more global features that are too costly to consider for each sample (e.g. global attributes such as shape and size). Generally, validation consists of using a more computationally expensive feature to verify whether or not a candidate obtained in the acquisition phase is the genuine article. It proceeds as a sequential decision process, after each feature is sought, a decision must be made whethter to accept the original acquisition hypothesis, to reject it, or to continue the

validation process.

In the course of this study, an Interactive Facility for Scene Analysis Research was constructed [114]. This system allows an experimenter to describe basic perceptual concepts to a computer in terms of examples. These examples are designated graphically by encircling areas of a displayed scene with a curser. The concepts are represented internally in terms of the values of primitive feature extraction operators that distinguish it from the previously defined objects. Thus, the operator can extract a variety of local attributes (e.g. light, hue, saturation, surface orientation and range) from input arrays of colour and range data. By outlining different objects in the scene, the operator can determine the information necessary to develop a perceptual strategy for distinguishing them. He can also validate the adequacy of a given set of attributes by displaying all points in a scene that satisfy the specified predicate. The interactive facility proved that most objects in an office environment can be distinguished by a small number of available features. It also showed the importance of semantic information in the segmentation and interpretation of real pictures where region interpretations are deduced from local attributes and contextual constraints imposed by previously interpreted regions. As will be

discussed in detail in Chapter V, Tenenbaum and Barrow used these ideas in the implementation of two systems, Interpretation Guided Segmentation (IGS) and MSYS: A System for Reasoning about Scenes [12, 113]. These two systems are directly relevant to the work of this thesis.

Currently, several research groups are building and developing computer vision systems that use different sources of knowledge to interpret real world scenes. These systems are at different levels of development. For example, at the University of Rochester, Ballard, Brown and Feldman [8], proposed a vision system which uses a semantic network model and a distributed control structure to accomplish the image analysis process. The system is structured in layers where the most abstract end of the structure is the semantic network representing the model. At the bottom, the natural image is stored in what is called an image data structure which consists of the original image at different magnifications, spectra, relationships, etc., together with various filtered versions of the image, edge images, texture images, etc. This multi-layered structure is similar to the one proposed by Levine [66], and Hanson and Riseman [43, 89]. Between these two data structures there is an intermediate data structure that interfaces the

raw data at one end to the model description at the other end. It is called a sketch map and it is synthesized during the image analysis and provides association between the model and the image.

The model is constructed of different kinds of knowledge about the image domain. It includes a relational network of nodes where the links can encode constraint relations between nodes, the probability that the relationship holds and a quantifier representing the expected value of the relationship. The system is being tested on two applications: the finding of ships in a dock scene, and the finding of ribs in a chest X-ray film [9].

At the University of Kyoto, Sakai, Kanade and Ohta [54, 104] have described a scene analysis system which can perform semantic segmentation of a scene using a hierarchical structure. Both top-down and bottom-up control are successfully combined in the processing strategy. A semantic network is used to represent the knowledge in the bottom-up process. Each node of the network is called a knowledge block. This knowledge block holds information about matter in the world, for example, object "SKY", material "concrete", property "blue", or relations "linear boundary". The information is stored in the block as a set

of declarative rules, indicating properties it must satisfy and relations it has with other blocks. First, the input picture is segmented using recursive thresholding techniques. The bottom-up process assigns different interpretations to each region in the scene, thus generating a plan that can be used by the top-down process. The knowledge in the top-down process is organized as a production system. Each production rule has two parts; situation and actions. Due to the existence of a large number of situations to be tested to find the rule with highest priority, they used the scene hierarchical structure to divide the rules into many groups. Thus, the number of rules to be examined in each phase is reduced. The work of Kanade et al, done in parallel with the research of this thesis, shares similar objectives, although implemented using a different formalism.

At the University of Massachusetts, Hanson and Riseman et al [43, 44, 77, 89, and 90] are designing and developing a general purpose computer vision system that will scan the whole spectrum of computer vision problems. The system is divided into two stages. The first stage would produce a description of regions and boundary segments as well as two dimensional shape attributes. This information is passed to the higher stage of analysis (i.e. the interpretation process).

The low level stage consists of two separate processes. The first utilizes local information to aggregate edges into boundaries using a parallel iterative relaxation procedure. Using a Kirsch operator [59], an edge picture is generated, and the relaxation process [131] is applied to organize edges into boundaries. The second process utilizes global histograms and local spatial analysis to form regions. A number of cluster centers are first defined in the colour feature space and each image point is given a set of probabilities depending on its position relative to the different cluster centres in the colour feature space. By applying a parallel iterative relaxation procedure, each pixel should be associated with one cluster. Although the cluster centres in the reported results were specified by the researcher, they have described several approaches for finding these cluster centres automatically using global histogram analysis [77, 82, and 103].

A hierarchical parallel array computer, "the processing cone", which is hierarchically organized into layers of decreasing spatial resolution in which information extracted from increasing sizes of receptive fields can be stored and processed. Its function is the reduction of the

immense amount of image data, while at the same time providing a structure in which information at the higher levels can direct more detailed processing at the lower levels [55, 89, 110, 119, and 120]. A method for integrating the results of both boundary and region formation is proposed. This method would make use of the spatial extent of the gradient around the boundary. For large gradient width, the acceptable disparity between region-induced boundaries and those obtained from the edge analysis can be increased. This combined segmentation represents the input to the higher stages of analysis.

The higher level stage is called VISIONS (Vision Integration by Semantic Interpretation of Natural Scenes). In this system they have proposed the following:

i) Multiple levels of representation for the image-specific knowledge (knowledge about the scene under analysis) as well as the general knowledge about the class of scenes under analysis.

ii) A set of modular knowledge sources for the transformation of data between the different levels of representation.

iii) A hierarchical control structure for the application of the knowledge sources.

iv) A tree representation for the history of search

through the space of possible models.

The highest level knowledge structure is constructed as schemas [2, 3, 72, 79, 84, and 105]. The knowledge sources proposed would include attribute matching, 3-D shape manipulation, shadow, prespective, and occlusion. Knowledge used by the different sources can be declarative or procedural. Many parts of the VISIONS system are still under continuing development and will have to be integrated into the overall system.

Results reproted in [44] focused on three knowledge sources:

a) region-object attribute matching,

b) size estimates from prespective analysis, and

c) 2-D shape fitting.

Other knowledge sources are being implemented, such as 3-D shape matching, for example. Currently, their main concern is the study of the "quality of the information produced by individual knowledge sources, the content and structure of the long-term memory, and the combined effect of knowledge sources in forming hypotheses at the object and schema levels" [44]. The proposed system is a very ambitious one and it remains to be seen how these processes will eventually be integrated and how the knowledge of each source will interact with the other sources.

All of the research groups reviewed so far have assumed static objects in a picture. An important aspect of vision is the analysis of a sequence of images [1] that represent the same scene with the position of objects changing due to motion (e.g. pictures of a traffic intersection).

In this respect, at the University of Hamburg, Jain, Militzer, and Nagel [52] have introduced similar considerations to the ones adopted by researchers working on static scenes. They have implemented an algorithm to analyze sequences of TV-frames from real world scenes, such as a street intersection with cars. They assumed that the systematic grey value differences, based on second order statistics, between consecutive frames are due to images of moving objects. First, the frame is divided into regions of 4X6 pixels each. Each region is called a geo-pixel. For each geo-pixel, the mean and standard deviation is computed and stored. When analyzing the n+1 frame, each geo-pixel is compared to data previously observed at the same geo-pixel coordinates in the n-frames by means of a likelihood ratio. If the likelihood ratio is smaller than some threshold t, then the two geo-pixels are considered to be drawn from the same grey level distribution; otherwise it is decided that

the two geo-pixels belong to different grey level distributions (mismatch). These observations are accumulated and stored. Thus if the failing match of a geo-pixel is due to genuine motion of the corresponding object then the next frame should not also match at the same geo-pixel position. In this work, an algorithm is decribed to cluster mismatched geo-pixels over a sequence of frames and to determine the speed and direction of the moving object.

Nagel [76] has proposed a system that uses the above algorithm to extract non-stationary components. Next, relations are established between these components in different frames. A partial 3-D description of the non-stationary as well as stationary components could then be obtained. In addition, the trajectory of, and the rotation of, the moving objects can be inferred. Nagel proposed that the system exploits the frame sequences by gradually building up a reliable description based on evidence from many frames without undue emphasis on the first one. He proposes the use of planning to determine which part of the next frame should be investigated. An interesting aspect of this system is its attempt to accumulate evidence incrementally using deductions over a sequence of frames. Another important point is its use of

planning in the time domain to guide the analysis.

During the seventies, speech understanding was a very active research area. In certain respects, it bears a relationship to image understanding. In both image and speech analysis, knowledge from different sources are required. However, knowledge sources are usually errorful and incomplete due to deficiencies in theory, implementation or data. Both areas involve a huge amount of data and a large search space. One of the speech understanding projects that has had a great impact in the research community is HEARSAY II [32, 33, 62, 63]. It has influenced many researchers and provided a new approach for complex system design. HEARSAY II was constructed of diverse and independent knowledge sources. The knowledge sources are kept separate, independent and ananomous. A global data structure, the "blackboard", was introduced as the only means of communincation and interaction of the knowledge sources. Thus each knowledge source responds to changes of the blackboard that are of interest to it and it applies its knowledge within the context of such a change. The activation of a knowledge source is decided by the pattern matching of its precondition with the blackboard. This implies data directed activation of the knowledge sources.

When a precondition of a knowledge source matches the blackboard, the knowledge source is added to the list of tasks pending execution.

The blackboard has many levels, one for each problem space decomposition. As a result, each level is used to hold a different representation of the problem. Examples of levels are, "phrase", "word", "syllable" and "segment". These levels form a loose hierarchy and imply a hierarchy of knowledge sources. In general, a knowledge source deals only with a few (usually two) levels in order to apply its knowledge. In computer vision the function of knowledge sources is the transformation of data from one level of abstraction to a higher level; for example, image to regions, regions to objects, image to edges and edges to boundaries. However, there are no generally accepted levels of abstraction for vision as there exists for speech.

The system adopts the hypothesis and test paradigm for its processing strategy and a knowledge source can create new hypotheses and examine or modify the attributes of these hypotheses. HEARSAY II, uses a scheduler to decide which of the waiting tasks would next be activated. The scheduler calculates a priority for each waiting task and activates the task with the highest priority. The priority

calculation is goal directed because it involves the estimation of the usefulness of the knowledge source activation in fulfiling the overall system goal of recognizing the utterance. The results obtained by the system came close to the original perfromance goals set out in 1971.

The HEARSAY II system organization is very important because of the potential for its application to other research domains. It provides an interesting example of how the final system performance is affected by the system's engineering. Experiments with simulation of this system have suggested that "parallel processing can be exploited effectively in this architecture" [62].

The concept of cooperating independent sources of knowledge which operate on a global data structure has affected the framework we have proposed for a computer vision system. The role of the STM in the vision system is very similar to that of the blackboard in HEARSAY II. In fact, HEARSAY II has had an impact on the work and ideas of many researchers in different fields, especially in computer vision.

1.4 ORIGINAL CONTRIBUTIONS

In this dissertation, we have presented a new paradigm for a computer vision system, which is both data directed and knowledge based, for segmenting and interpreting real pictures of natural scenes. The system is a new approach which supports the incorporation of both bottom-up and top-down control strategies within the same framework. In this section, we will briefly discuss our contributions in the following aspects:

i) Overall System Structure and Control

The system is constructed of several independent processes that interact only through the SHORT TERM MEMORY. The concept of using a global data structure (STM) was introduced in the HEARSAY II speech understanding system [33, 62]. However, in addition to introducing this concept to computer vision, we have extended it by implementing the STM as an associative memory using the relational database formalism. We have also designed and implemented a relational algebraic sublanguage to serve as a communication language between the SHORT and LONG TERM MEMORIES and the different processes of the system. Thus, we have achieved a high level of structure flexibility and process modularity. A new process need only be aware of the overall STM and LTM data model (the relational model) and the communication

language to and from this data model. This facilitates the addition of new processes as well as the reformulation of any of the system's processes.

Moreover, we have proposed a new framework within which a flexible control strategy can be implemented. This control strategy differs from the previous approaches in that it is data driven. There is no pre-specified control hierarchy, but the system monitors the current status of analysis and decides accordingly what to do next. The strategy can be explicitly expressed using both the SCHEDULER and the FOCUS OF ATTENTION processes. The SCHEDULER enables the system to specify when a process should be activated based on the status of analysis as exhibited in the STM. The FOCUS OF ATTENTION process facilitates the incorporation of different heuristic PLANS in the analysis.

ii) Model Representation

All model information is stored in a LTM. In distinction to previous designs, it is completely separate from the analysis process and is implemented as a relational database. Thus model information can be easily modified to experiment with the same model, or to use the system in the analysis of different picture domains. The use of the

relational data model provides an attractive and convenient method for experimenting with different image models.

iii) Hypothesis Verification and Interpretation

In the current implementation, we have adopted the concept of competition and cooperation as a basic paradigm for the system processing strategy. This paradigm is implemented as a continuous relaxation labelling process, which has been used by researchers in the low level stages of picture analysis [43, 131]. Their work concentrated on using relaxation labelling to find edges, boundaries and regions using a very limited number of constraints. These constraints were uniformally applicable to every entity in the computation. Although we have used the same updating equation, we have described and implemented relaxation labelling for region interpretation at a higher level of analysis. In comparison to low level analysis, there is a larger number of constraints. Also different sets of constraints must be considered for the different interpretation combinations; hence constraint representation and satisfaction require a more sophisticated mechanism. We have therefore explicitly expressed the constraints in the model representation, thereby providing a powerful tool to experiment with different constraint combinations and to test their effect on the results of the analysis.

iv) Low Level Segmentation

As a minor contribution, we have also proposed and implemented a LOW LEVEL SEGMENTATION process which is based on an adaptive thresholding technique. Although many techniques have been proposed, this algorithm is very fast, considering the results obtained on a real picture.

v) Results with Natural Scenes

The system is a new approach that incorporates the use of high level contextual clues in the analysis. We have tested it on a natural outdoor scene that contains objects such as SKY, TREE, ROOF, WALL, etc. Using a 2-D world model, we have formalized a model to describe the objects and the constraints that govern their coexistance. The resulting segmentation is excellent, as will be shown in section 6.3. We have also performed a set of experiments to investigate the effects of structuring relaxation labelling as a parallel iterative process as opposed to a sequential one. Although these experiments were performed on a simple scene, the results obtained suggest that sequential application of the continuous relaxation labelling would reduce the total time needed for the analysis by 20%-30%. The results and conclusions of these experiments are given in Chapter VI.

1.5 THESIS OVERVIEW

The work presented in this thesis deals with the problem of building a computer vision system for segmenting and interpreting real pictures. Initially, we have set certain guidlines and requirements as have been described in section 1.1. We have attempted to maintain these guidlines. In the next few chapters we will present the system's structure and analyze why this structure was selected. This is followed by a detailed description of the system and each of its components. Finally, the experimental results, along with suggestions for future development are presented.

In Chapter II, we will review and present several database models and discuss why the relational model was chosen for the design of the SHORT and LONG TERM MEMORIES. The communication language of both the STM and LTM is also described and a brief review of its use by the different processes is presented.

In Chapter III, the features computed by the system are introduced, and the STM relations are then defined. Next we discuss the computations performed to obtain the values of the features stored in these relations.

Chapter IV deals with the visual model of the scene used in the analysis. The LTM relations are defined and their use by the different processes of the system is discussed.

Having discussed the STM and LTM and their implementation as relational databases, we proceed to introduce in Chapter V each of the system processes that have been implemented todate. We will describe in detail the algorithm for each process.

Finally, experimental results obtained on real and test pictures are presented in Chapter VI. We will analyze the experiments to test the performance of the individual processes. We will then examine the performance of the system when the analysis was applied to a real outdoor scene. We will also consider the results of the experiments done to investigate the effect of different processing strategies on the speed of convergence. Lastly, we comment on the advantages of the current implementation, as well as the limitations encountered and suggest further directions for future research.

CHAPTER II

SHORT AND LONG TERM MEMORIES: A RELATIONAL DATABASE APPROACH

2.1 INTRODUCTION

Computer vision systems are characterized by the necessity to solve subproblems containing large search spaces, diverse and large sources of knowledge, and requiring nondeterministic decisions made at various levels of analysis. To distinguish the approach we have taken, let us consider and analyze two different vision systems, S1 and S2, both consisting of a number of modular processes. The first system S1, has a completely integrated structure in that the visual model information, input data, intermediate and final results are part of the storage structure of the processes (figure 2.1). On the other hand, all the processes in S2 have accesss to a common data structure that contains the complete visual model and processsing strategies (LTM), input data, as well as intermediate and final results (STM) (see figure 2.2).

System S1 suffers from certain difficulties, namely:

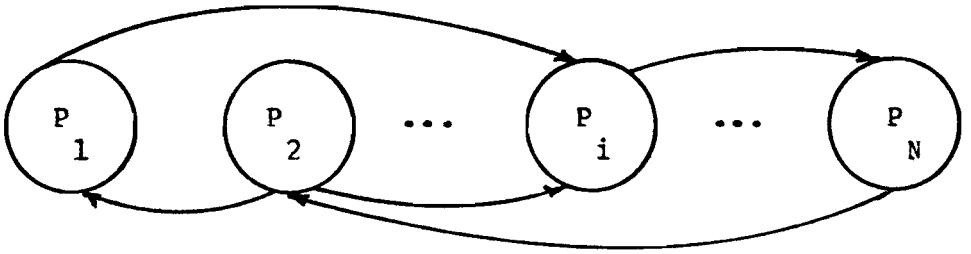


Figure 2.1 System S1

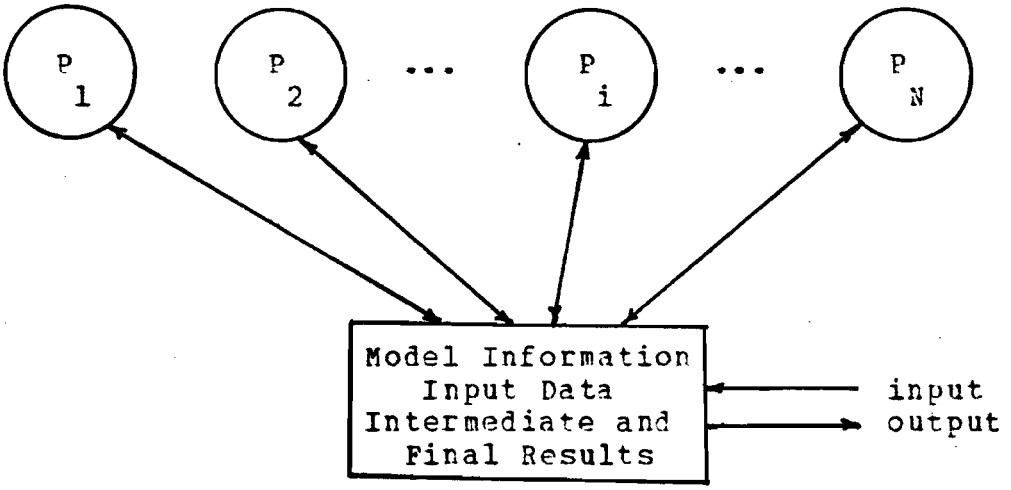


Figure 2.2 System S2

1. experimentation capabilities

It is difficult to introduce even minor changes to the world model or to adapt the processes to another world model. This arises from the fact that the model is buried within the internal storage structure of each process, and hence we are faced with the problem of knowing the storage structure of each process or maybe even rewriting the computer code that implements the same process. Furthermore, the fact that the model is not directly and easily accessible but is embedded in the computer code makes it extremely difficult to distinguish between the model and the algorithm. In contrast, the visual model in system S2 is explicitly available to the casual user, i.e., he need only know the common data structure used to store the model information. Thus it is easier to change the model or to use the system with other world models.

2. communication capabilities

In general, vision systems contain a number of processes that interact dynamically to achieve their global processing task(s). This interaction is usually implemented by allowing each process to transmit results to other interested processes. However,

processes in S1 do not have any global broadcasting medium or general communication links except through the internal storage structure of each process or through the direct transfer of information between modules as shown in figure 2.1. The direct transfer of information between processes seriously reduces the modularity of the system. It complicates the design and implementation of the overall system, especially, as is usually the case, if it is designed by several persons over a long span of time. In addition, it increases the risk that any changes in any process will affect all of the other processes that interact with it. On the other hand, the common data structure in system S2 serves as a communication channel for the system processes; an example of this is the "blackboard" in the speech understanding system HEARSAY II [33]. Each process in the system transmits its intermediate and final conclusions to the other processes through this common data structure, thus allowing each of them to have access to all of the information necessary for achieving its task. Since the processes are able to share data, they do not have to repeat calculations done by any one of them.

3. system expandability and structural flexibility

Incorporating a new process in the first system represents a very difficult, almost impossible, task because the designer must have an intimate knowledge of the operation and internal storage structure of each of the other processes. Also, in interfacing his process to the system, he must consider its effect on the operation of the others. Alternatively, system S2 can easily adapt itself to the incorporation of new processes or the deletion of some of the old ones. This is because the only data structure the new process must communicate with is the common data structure of the system (e.g. STM and LTM). The operation of an individual process does not directly affect the functioning of the rest. It must only transmit its conclusions to the common data structure which may or may not be used by the other processes, depending on their needs. Thus, system S2 will facilitate the introduction of any alterations to the system configuration or the overall control structure.

4. parallel operation

It is obvious that S1 requires some form of synchronization between the different processes. Each of the latter in S1 must ensure that the others are

ready to deliver the necessary conclusions in order to continue their own computation. The system design therefore needs a complex synchronization mechanism. In contrast, the processes of system S2 need only monitor the changes in the information in the common data structure, waiting for the apppropriate conditions in the data which are required for their processing task. Thus each process broadcasts its conclusions and does not have to be involved in the synchronization of the others. The system can therefore operate in an asynchronous mode, or it is even possible that different processes may run in synchronism (i.e. parallel) as long as they are able to obtain the data needed for their tasks from the common data structure.

Because of these arguments, we conclude that the S2 system configuration, with its associated common data structure, is favoured over S1. The framework for the computer vision system which we presented in section 1.2 is modelled after system S2. It contains a SHORT TERM MEMORY (STM), a LONG TERM MEMORY (LTM) and a number of independent processes. So far we have avoided the question of the form of the common data structure in system S2. The STM and LTM were designed and implemented as a relational database system with an associated accessing mechanism in the form of

a relational algebraic sublanguage [25]. Two questions come to mind, namely:

(i) Why do we need to use a database?

(ii) Why was the relational model adopted as a data model for the database?

According to Engles [31], "A database is a collection of stored operational data used by the application systems of some particular enterprise". The "enterprise" in the sense used here is simply a convenient term to describe any large-scale system (e.g. commercial or scientific). In the vision system to be described, a database is used as an associative memory for the large amount of data used in the processing, such as input data, model information, intermediate and final results, and so on. It provides an accessing mechanism to the information and relieves the individual user from the problems associated with either the administration of huge amounts of data or the search of large spaces. In addition, it acts as a communication channel between the system processes. In other words, each process obtains its input from the database and outputs its final conclusions as well as intermediate results to the same database. The system user need only know the appropriate data model and the accessing

language of the database. At this stage, it is appropriate to point out the distinction between the storage structure of the database and its data model. The former is concerned with the actual physical structure and layout of the data on the particular computer storage device used, and is of interest only to the database administration programs [25]. The latter is the user view of the database and this will be discussed further in the next section.

In the above, the advantages of having a common data structure for the computer vision system have been reviewed. Now, let us consider the other advantages that we may obtain by implementing the common data structure as a database system:

(i) The use of a database reduces the amount of redundancy in the stored data, which in turn will reduce, even remove to some extent, data inconsistency. This is, in general, achieved by normalization of the data in the database, as will be shown later in the discussions in section 2.4.

(ii) Using a database system enhances the modularity of the system processes. It also provides the independence of the system from the particular physical

hardware or software capabilities of the computer system, since each process or user employs a standard accessing mechanism (data sublanguage). Thus if the computer system changes, we need only change the database administration programs which are transparent to the application programs (users).

(iii) The use of a database system enforces standards, which in turn, enhances the modularity and flexibility of the overall system structure. Basically, the database system ensures the use of a particular communication language (data sublanguage) and data is only viewed through the database data model. This unifies the design procedures for the different processes and simplifies problems that may arise if we decide to add, remove or change any of the system's processes.

So far we have answered the first question, that is, why do we need a database. In the next few sections, we will define what we mean by a database system, as well as discuss different possible data models. The data sublanguage used with the relational model will also be described in the subsequent sections.

2.2 WHAT IS A DATABASE?

We start by defining what constitutes a database system by considering four aspects. First, we note that a database is a large collection of data stored on some computer storage device, probably disks. Second, there exists a set of application programs, processes in our context, that operate in conjunction with this data by performing the usual functions of retrieving, updating, adding or deleting. In addition, there may be a number of concurrent on-line users who interact with the database. Generally, both the application programs and the on-line users have a predefined view of the database which is called the data model. Again here we make a distinction between the storage structure of the database and the data model. The former represents the actual structure and organization of the data on the physical storage device. The latter describes only how the user views the data. Third, there is the database management system which is responsible for handling all of the access to and from the database. Any access request is intercepted by the database management system which inspects the database model and then performs the necessary operation on the physical stored data. The data model is the user's view of what is in the database, and the data sublanguage is the language used for

transferring data to and from the database. The data model and the associated data sublanguage are the basic means by which data independence is achieved. They actually relieve the user from all of the problems of data organization, management, and maintenance. Thus the user of the database need only know the database model and the database accessing mechanisms, the latter in the form of the data sublanguage. Both of these will be discussed in detail in sections 2.3 and 2.4.

A working definition[1] of a database system describes the database as a collection of stored operational data used by the application systems (specific application programs or on-line users) of a particular large system (e.g. vision system). As an illustration of the notions discussed in this section and to explain the concept of operational data, let us consider a small segment of the data that is used by computer vision programs. Usually the vision system finds it necessary to retain certain information about the current segmented regions of the image, such as their different feature values and the structural relationships between the regions in the image. It will also need to keep information about the set of interpretations attached to each region and the confidence level associated with each of the

---

[1] Based on one given by Engles [31].

interpretations. This is illustrated in figure 2.3 where each entity appears as a block and represents all the data concerned with the entity. An association between any number of entities will be represented as a contour enclosing these entities. For example, there is an association between the regions, their interpretations and associated confidences. That is, each region has a set of interpretations and each interpretation is associated with a set of regions. Each region interpretation is also labelled with a specific confidence. The significant point here is that these relationships are as much a part of the operational data as are the associated entities and must therefore be represented in the database. This may be done by means of pointers, by physical adjacency or by any other method that seems suitable. Although most relationships in the diagram associate two types of entities, in general, any number of entities may be involved. In the example of figure 2.3, there are contours enclosing three types of entities (regions, interpretations, and confidences). Association may also involve only one type of entity, such as regions for example. Thus a linkage between one region and another occurs when a particular region is adjacent to another. Note as well that the same entity may be associated in any number of relationships (e.g. regions).

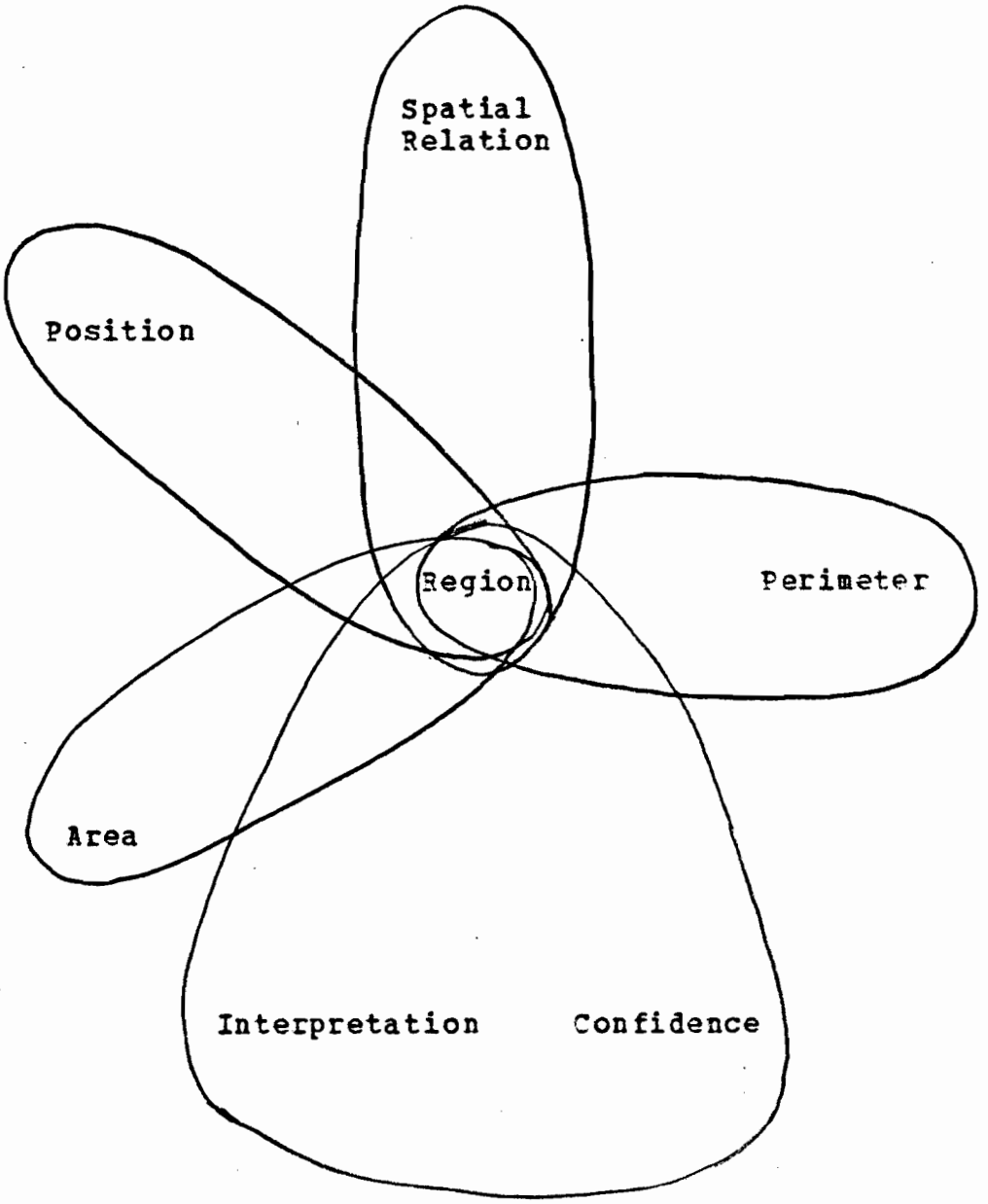


Figure 2.3 Example of a segment of a database for a computer vision system

The physical storage structure of a database is of importance only to the database management system, that is, the software which handles all accesses to the database. It is responsible for the data organization and structure on the secondary storage device. On the other hand, the user or application programs view the database through the data model and interact with the database using the appropriate database sublanguage (figure 2.4). Thus the database storage structure and its accessing software programs are invisible to the user which thereby ensures data independence.

2.3 WHY A RELATIONAL DATABASE?

As discussed in the previous section, the data model represents user view of the the database. He employs a database sublanguage to interact with it, performing the usual functions of retrieving, updating, adding or deleting. The question is: What form should the data model and the associated data sublanguage take? Before we answer this question, we shall briefly review the three most common data models presently in use, namely: the hierarchical, the network and the relational models. A detailed review of these models and their associated sublanguages can be found in [25].

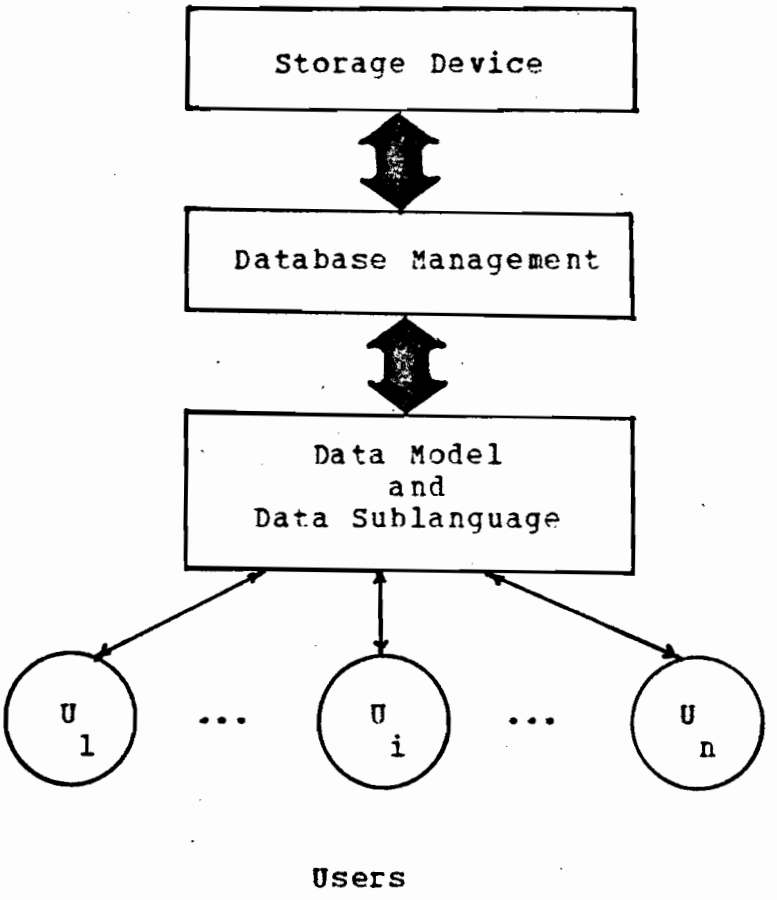


Figure 2.4 Simplified view of a database system

2.3.1 THE HIERARCHICAL DATA MODEL

For historical reasons the hierarchical approach has been very popular and it is used in many existing database systems (e.g. IBM's Information Management System IMS [51]). It has its origin in the storage structure which was prevalent when most data processing was performed with purely sequential media (magnetic tapes), and there was a minimal distinction between the data model and the storage structure. Figure 2.5 is an example of the hierarchical model, where each node has one and only one superior and can have any number of subordinates. It is fundamental to the hierarchical data model that any given node of the graph takes its full significance in the context defined by its superior node. For example, the fact stated in node 2 (REGION 18, LEFT) takes its full meaning when viewed in the context of its superior node 1 (REGION 7). Then we have REGION 7 is LEFT-OF REGION 18.

The hierarchical approach suffers from many disadvantages mostly due to the fact that hierarchies are very suitable for modelling one-to-many correspondences. However real-life data, in most cases, involve many-to-many situations which complicate the processes of accessing, updating, adding and deleting data. For example, if we delete the only region that has a particular object as one

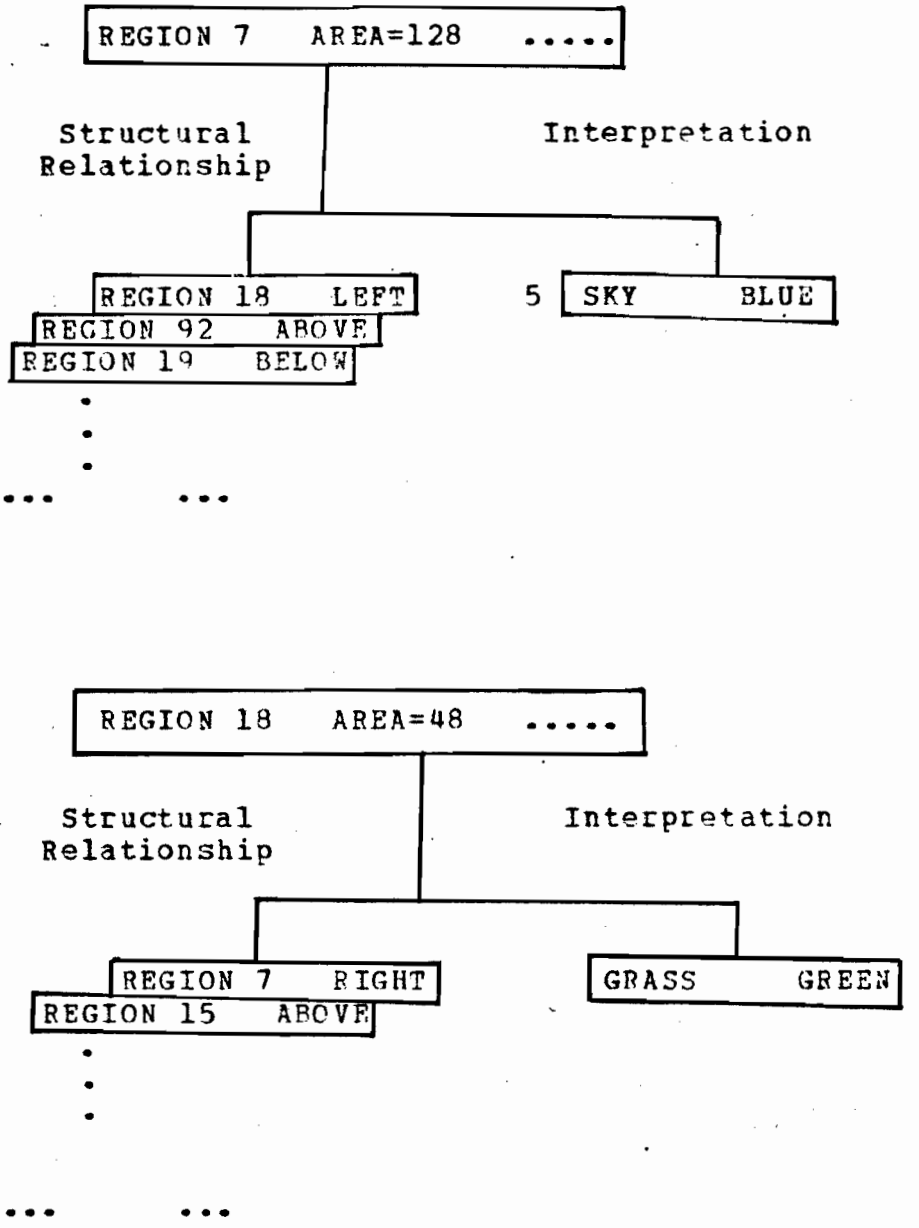


Figure 2.5 Example of the hierarchical model

of its interpretations, data concerning that object will be lost. If REGION 7 was the only region in the database that had SKY as an interpretation, all the information about the object SKY, for example its colour BLUE, would be lost by deleting REGION 7. Also, to make efficient use of the database, the data should be retrieved in the same sequence as they are stored, which means that the primary node should first be restored and then its subordinates. In general, this may not be the sequence that the user desires. Most of the disadvantages suffered by the hierarchical model may be avoided in a specific database implementation; however, we are here concerned with discussing the basic features of the hierarchical structure and not the specific features of a particular implementation. The major advantage of the hierarchical model is that it is obviously a very natural way of modelling real-world hierarchical structures.

2.3.2 NETWORK DATA MODEL

A network representation is a more general structure than a heirarchical representation since each node may have any number of immediate superior or subordinate links. Figure 2.6 is an example of the network data model. The network approach overcomes all of the difficulties encountered in the heirarchical model. For example,

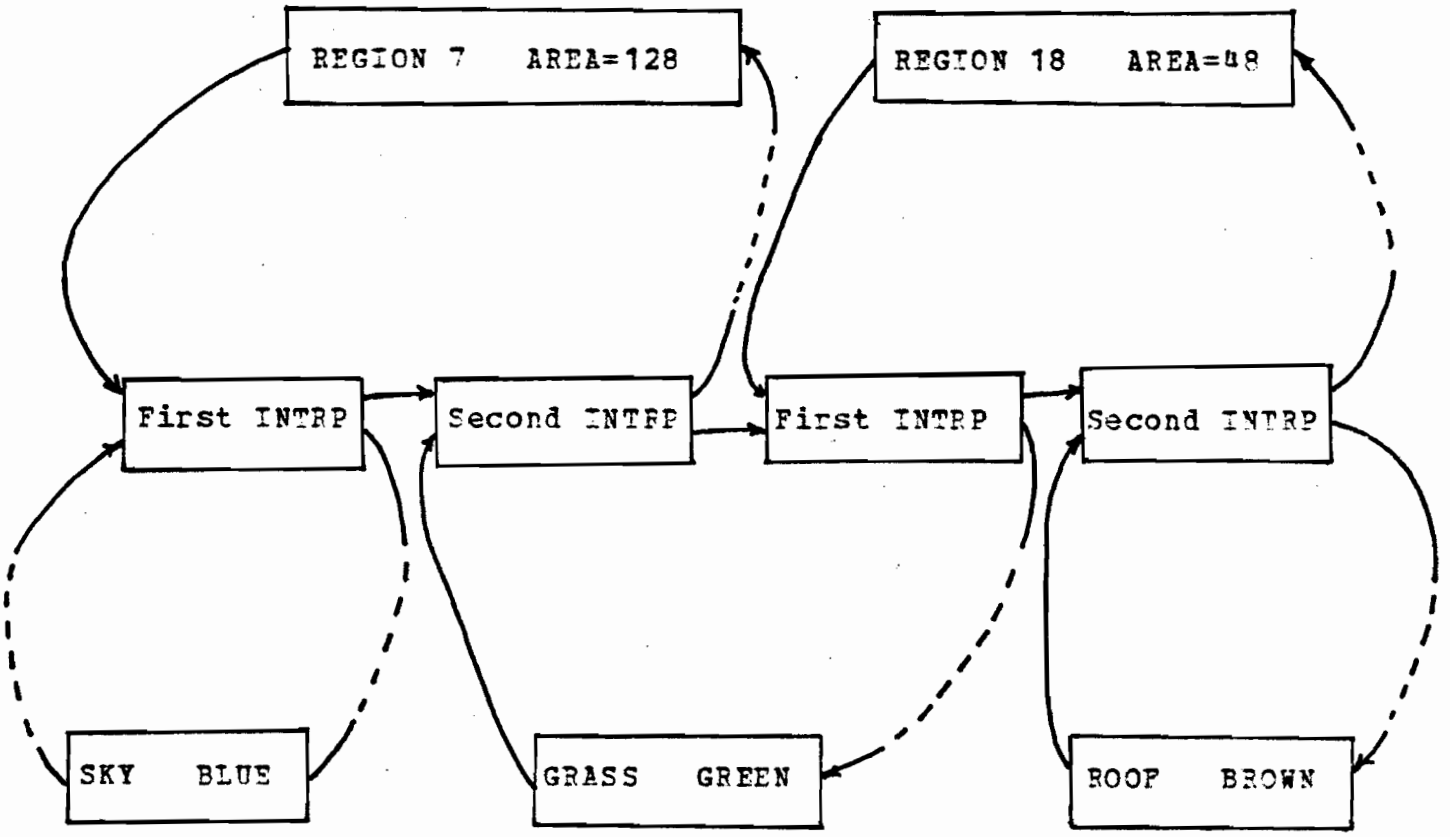


Figure 2.6 Example of the network model
(INTRP is used for Interpretation)

operations such as adding, deleting, updating and retrieving data are accomplished simply by manipulating pointers. However, it turns out that the difficulties do not disappear simply because of the nature of the network model, but rather because of the resulting data normalization [25]. The major disadvantage of the network model is simply that it is too close to the actual storage structure. The data model resembles to some extent the storage structure, which violates the requirement of data independence. The user must be thoroughly aware of which chains (links) do and do not exist. More important, these chains are directly visible to the user and hence must be directly represented in the storage structure in some way or another. Briefly, the existence of two types of entities, namely the nodes and links, complicates the user data model and results in a complex accessing data sublanguage [26].

2.3.3 RELATIONAL DATA MODEL

The relational approach was first introduced as a basis for a data model of database system by E. F. Codd [24]. This data model is based on the mathematical theory of relations which provides a sound theoretical foundation. The term RELATION is used throughout this text in its accepted mathematical sense.

Given sets D1, D2, ..., Dn (not necessarily distinct), R is a RELATION defined on these n sets if it is a set of n-tuples (d_1, d_2, ..., d_n) such that d_1 belongs to D1, d_2 belongs to D2, ... and d_n belongs to Dn. Sets D1, D2, ..., and Dn are called the DOMAINS of R. The value n is called the degree of R. Figure 2.7 illustrates a relation called REGION-DESCRIPTION which is defined on the domains REGION (region number), AREA (region area), PERIMETER (region perimeter) and ASPECT-RATIO (region aspect ratio). As the figure shows, it is convenient to represent a relation as a table with rows representing the n-tuples and each column representing one domain. It is important to observe the following properties of relations:

P1- All n-tuples of a relation must be distinct; that is, duplicate rows are not permitted.

P2- The ordering of the n-tuples within a relation is immaterial.

P3- Because we refer to the relation domains (columns) by name only, the ordering of the domains is insignificant. Thus when two or more of the relation domains have the same name, we add an appropriate suffix to distinguish between the domain roles, as illustrated in the relation LEFT-OF in figure 2.8.

REGION-DESCRIPTION

REGION	AREA	PERIMETER	ASPECT-RATIO
7	128	50	2
18	48	36	1.3
23	350	93	3.5
.	.	.	.
.	.	.	.
.	.	.	.
166	18	20	2.0

Figure 2.7 Example of the relational model

LEFT-OF

REGION-X	REGION-Y	V-LEFT
7	18	0.3
8	10	0.5
20	29	0.7
20	35	0.7
.	.	.
.	.	.
.	.	.

Figure 2.8 Example of a relational model

Here the domain REGION (region number) is used twice, but we have added the suffix X and Y to identify the role played by REGION in its two appearances (i.e. REGION-X is LEFT-OF REGION-Y by the amount V-LEFT).

P4- Each domain element in the n-tuple is an atomic data item; that is, none of the relation domains has an element which is itself a set. An atomic data item represents a number or character string depending on the domain type. Thus at every row/column position within the table (relation), there exists precisely one value which is a number or a string of characters.

All relations must satisfy the first three properties (P1, P2, and P3). However, if a relation also satisfies property P4, it is considered to be a normalized relation. Figure 2.9 shows an example of unnormalized and normalized relations. The normalization of the database relational model will be discussed later in section 2.4.

In section 2.2, we pointed out that a database system must be able to represent two types of objects, namely; "entities" and "relationships". Fundamentally, there is no real difference between the two; a relationship is merely a special type of entity. The three favoured data models (hierarchical, network and relational) differ in the way

LEFT-OF

REGION-X	REGION-Y	V-LEFT
7	[18 20 22 23]	[0.3 0.41 0.72 0.75]
19	[1 18 22]	[0.82 0.25 0.37]
23	[18 20 .]	[0.34 0.12 .]

Unnormalized Relation

LEFT-OF

REGION-X	REGION-Y	V-LEFT
7	18	0.3
7	20	0.41
7	22	0.72
7	23	0.75
19	1	0.82
19	18	0.25
19	22	0.37
23	18	0.34
23	220	0.12
.	.	.
.	.	.
.	.	.

Normalized Relation

Figure 2.9 Relation normalization

they permit the user to view and manipulate relationships. In the hierarchical approach, relationships are entirely implicit. That is, the relationship between two entities is represented in some way by the relative position of the node(s) concerned. For example, the relationship between REGION 7 and the interpretation SKY in figure 2.5 is represented by the fact that a node occurrence of the interpretation SKY is subordinate to the node occurrence of REGION 7. As a consequence, the corresponding data sublanguage relies heavily on the concept of position within the database. In the network approach, on the other hand, relationships are represented explicitly by means of pointers (links). However the very fact that pointers are used explicitly means that relationships and entities are assumed to be different. In this case, the database sublanguage must provide two types of operations, and retrieving information from relationships involves searching for the information dynamically [25]. This should not be interpreted as being an argument against using pointers in the storage structure or the data model, it is rather an argument against using pointers to carry information in the data model. These pointers are visible to the user or applications program in both the data model and the database sublanguage. Usually the user will lock himself into a particular storage structure by making his programs dependent on the knowledge of the links and nodes of the

model.

The relational model does not differentiate between the "relationships" (associations between the entities) and the "entities" themselves. Consider the two relations in figures 2.7 and 2.8., the relation REGION-DESCRIPTION has tuples each of which represents an association between a region number (REGION) and a set of values for its AREA, PERIMETER, and ASPECT-RATIO[1]. Another example is given in figure 2.8 where each tuple of the relation represents an association between the two regions REGION-X and REGION-Y and the value of their structural relationship (LEFT-OF). In this case, the primary keys of the relation are the combination REGION-X and REGION-Y. Thus, the tuple of a relation represents the association between the key elements and the non-key elements. This model therefore makes no distinction between the "entities" and the "relationships". This in turn, provides a uniform set of operations for manipulating both "entities" and "relationships", an obvious simplification. In fact, the relational model is a view of the data in terms of its natural structure only and not of its storage structure or accessing mechanisms. In addition, the user's model of the relation is very simple in that he is dealing exclusively with tables. Because of the above considerations, we have decided to use the relational model

---

[1] The primary key of this relation is the region number (REGION), and this concept will be defined in section 2.4.

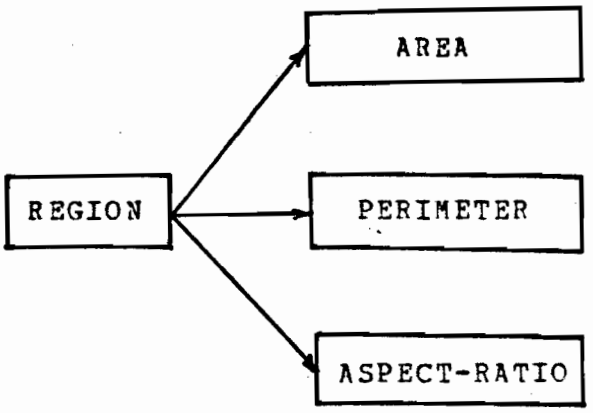
in the current implementation of the SHORT and LONG TERM MEMORIES.

2.4 DATABASE NORMALIZATION

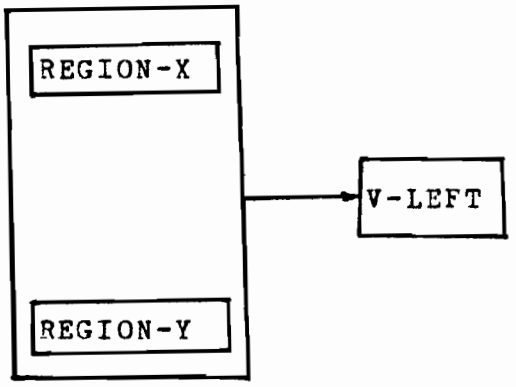
A normalized relation was defined in the previous section as a relation in which each of its underlying domains contains atomic (non-decomposable) items only, that is, none of its domains has elements which are themselves sets. Codd [21, 22] referred to a relation that satisfies the above-mentioned condition as being in the first normal form. An unnormalized relation may easily be transformed to an equivalent normalized relation in the first normal form as shown in figure 2.9. The general concept of normalization was introduced by Codd who defined three levels of normalization referred to as the first, second and third normal forms, respectively. The objective of this section is to briefly review the concept of normalization of the collection of relations that constitute the database. Detailed discussions and analysis of the normalization process are presented in Codd [21, 22], Date [25], Heath [46] and Rissanen et al [91].

The notion of functional dependence is of vital importance to the designer and administrator of the data

model. Given a relation R that has domains A and B, domain B of R is said to be functionally dependent on domain A of R, if and only if, at every instant of time, each value in A has associated with it precisely one value in B under R. This definition of functional dependence can be extended to collections of domains. If X and Y are distinct collections of domains of relation R, the Y collection of domains is functionally dependent on the X collection of domains, if and only if, at every instant of time, each value of X has associated with it precisely one value of Y under R. We write R.X ---> R.Y if Y is functionally dependent on X in R and R.X --/--> R.Y if Y is not functionally dependent on X in R. Also, if both R.X ---> R.Y and R.Y ---> R.X hold, then at all times R.X and R.Y are in one to one correspondence, and we may write R.X <---> R.Y. For example, in the relation REGION- DESCRIPTION shown in figure 2.7, the domain AREA is functionally dependent on the domain REGION (region number) since each region number has associated with it one and only one value of the domain AREA. Another example in figure 2.8 is the case where the domain V-LEFT is functionally dependent on the composite domain (REGION-X, REGION-Y). Given a particular combination of REGION-X and REGION-Y values, there is exactly one unique value for V-LEFT. We will use the scheme shown in figure 2.10 to express functional dependence.



(a) REGION-DESCRIPTION



(b) LEFT-OF relation

Figure 2.10 Representation of functional dependence

Next, we proceed to define the concept of full functional dependence. Domain Y is fully functionally dependent on the set of domains X if it is functionally dependent on X and not functionally dependent on any subset of X. Again in relation LEFT-OF (figure 2.8), domain V-LEFT is fully functionally dependent on the composite domains REGION-X and REGION-Y. Relation S in figure 2.11 contains domains REGION (region number), INTERPRETATION (object name), COLOUR (object colour), AREA (region area) and PERIMETER (region perimeter). It is obvious that domain COLOUR is functionally dependent on the composite domains REGION and INTERPRETATION. However, because the object colour is determined by the INTERPRETATION, the COLOUR domain is functionally dependent on the domain INTERPRETATION. Thus, the COLOUR is not fully functionally dependent on the composite domains REGION and INTERPRETATION because of its dependence on the subset INTERPRETATION.

Because all relations in the third normal form are also in the second normal form, and all relations in the second normal form are also in the first normal form, we will start with the definition of third normal form.

S

REGION	INTERPRETATION	COLOUR	AREA	PERIMETER
1	SKY	BLUE	150	82
2	ROOF	BROWN	257	130
5	SKY	BLUE	123	47
10	TREE	GREEN	109	79
18	WALL	YELLOW	48	259
25	SKY	BLUE	73	320
29	ROOF	BROWN	385	180

Figure 2.11.a The relation S

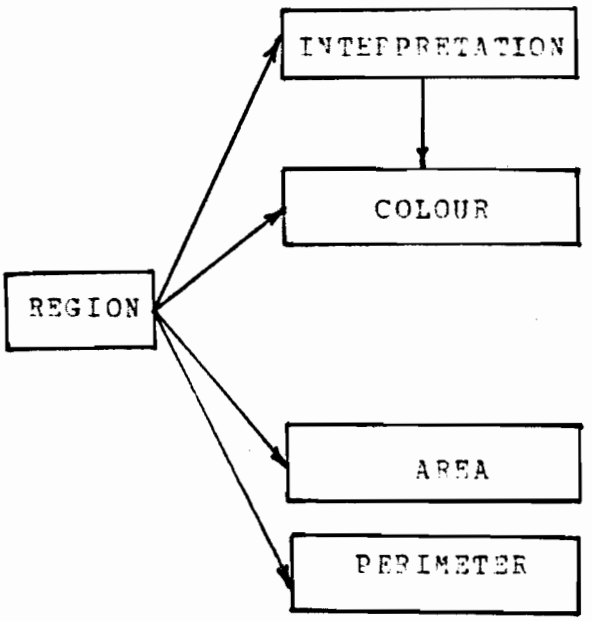


Figure 2.11.b Functional dependence of relation S

Definition [21]

"A normalized relation R is said to be in the third normal form (3NF) if and only if the non-key domains of R, if any, are (a) mutually independent and (b) fully functionally dependent on the primary key of R".

The term "primary key" refers to the combination of domains K (possibly a single domain) that has the following properties:

1- In each tuple of R, the values of the K domains, uniquely identify that tuple, that is, R.K ---> R.Ω where Ω denotes the collection of all the domains of the relation R.

2- No domain in K can be discarded without destroying property 1.

3- No tuple is allowed to have an undefined value for any of its K domains.

It is obvious that all of the domains of R can be considered in K since R.Ω ---> R.Ω , but property 2 limits this to a subset. Property 3 is imposed to reflect the vital part played by the key domains in search algorithms.

To simplify the discussion, we have adopted the notation used by Codd, Date and others, so that R(A, B, C, D, E) represents a relation R with the key domains A, B, C and non-key domains D and E. Thus, relation REGION-DESCRIPTION may be expressed as (figure 2.10 (a)):

REGION-DESCRIPTION(REGION, AREA, PERIMETER, ASPECT-RATIO)

and relation LEFT-OF may be represented as (figure 2.10 (b)):

LEFT-OF(REGION-X, REGION-Y, V-LEFT).

In figures 2.7 and 2.8 both relations REGION-DESCRIPTION and LEFT-OF are in the third normal form since their non-key domains are mutually independent and are fully functionally dependent on the key domains.

Now, if we remove the first requirement of the 3NF definition, namely, that of mutual independence among the non-key domains, we obtain the definition of a relation in the second normal form (2NF).

Definition [21]

"A normalized relation R is said to be in Second Normal Form 2NF, if and only if, the non-key domains, if

any, are fully functionally dependent on the primary key of R".

Relation S (figure 2.11) is an example of a relation in 2NF since its domains INTERPRETATION, COLOUR, AREA and PERIMETER are fully functionally dependent on the domain REGION. It is not in 3NF because domains COLOUR and INTERPRETATION are not mutually independent since the INTERPRETATION domain determines the COLOUR (COLOUR is functionally dependent on the INTERPRETATION). Due to the above mentioned functional dependence between non-key domains (INTERPRETATION ---> COLOUR), we are faced with lots of problems in adding, deleting and updating information. For example, if we wish to change the colour of the ROOF INTERPRETATION from BROWN to RED, we are faced with either searching the S relation for every occurrence of ROOF or allowing the possibility of having inconsistent results. Another problem arises in deleting REGION 18 from the relation since we will lose the fact that a COLOUR (YELLOW) is associated with the WALL interpretation. It is also impossible to add the fact that the COLOUR of the WINDOW might be VIOLET if there is no region having that interpretation. (It is not permissible to create a tuple with primary domains undefined). The solution is to transform the 2NF relation S into two 3NF relations (figure 2.12):

S1

REGION	INTERPRETATION	AREA	PERIMETER
1	SKY	150	82
2	ROOF	257	130
5	SKY	123	47
10	TREE	109	79
18	WALL	48	259
25	SKY	730	320
29	ROOF	385	180

S2

INTERPRETATION	COLOUR
SKY	BLUE
ROOF	BROWN
TREE	GREEN
WALL	YELLOW

Figure 2.12.a Relations S1 and S2

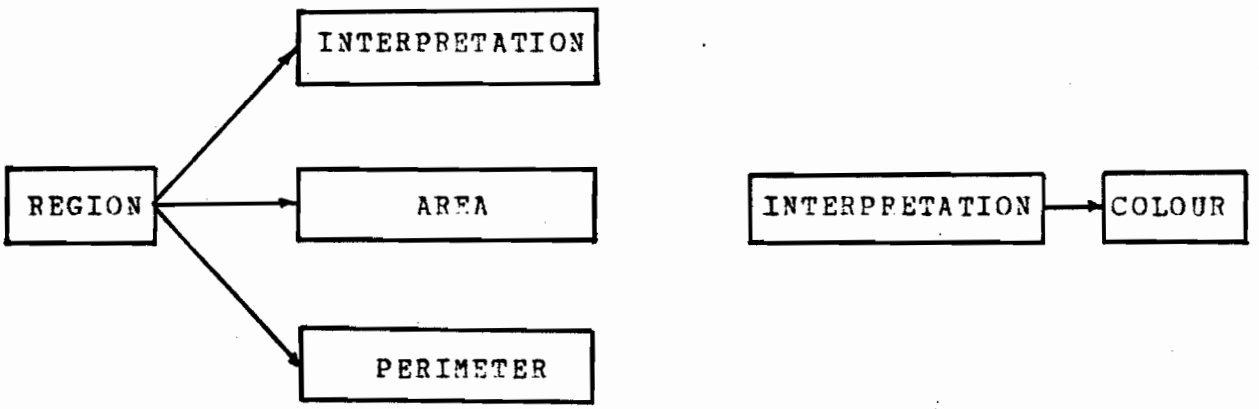


Figure 2.12.b Functional dependence of relations S1 and S2

S1(REGION, INTERPRETATION, AREA, PERIMETER)

and

S2(INTERPRETATION, COLOUR)

First relation S1 is a relation over the domains REGION, INTERPRETATION, AREA and PERIMETER. The second relation has domains INTERPRETATION and COLOUR. Thus we can now change the COLOUR of the ROOF to RED by a single access to the relation S2. We are now able to add the fact that the colour of the WINDOW may be VIOLET, even though there is no region interpreted as WINDOW. In addition, by deleting REGION 18 in relation S1, we do not lose the information that the colour of the WALL is YELLOW.

If we recall the definition of a normalized relation given in section 2.3, it turns out to be the same definition for a relation in the first normal form (1FN). That is, any normalized relation is in first normal form (1FN). For example, consider relation T which consists of domains REGION-X, REGION-Y, V-LEFT and AREA-X (figure 2.13). Obviously, it is a normalized relation since it satisfies the properties given in section 2.3. However, we note that domain V-LEFT is functionally dependent on the composite domains REGION-X and REGION-Y, and domain AREA-X is functionally dependent on REGION-X only. Thus AREA-X is not fully functionally dependent on the relation's key domains

T

REGION-X	REGION-Y	V-LEFT	AREA-X
1	3	0.17	75
2	5	0.32	239
2	7	0.46	239
2	23	0.15	239
23	8	0.26	250
23	19	0.92	250

Figure 2.13.a A relation T in 1NF

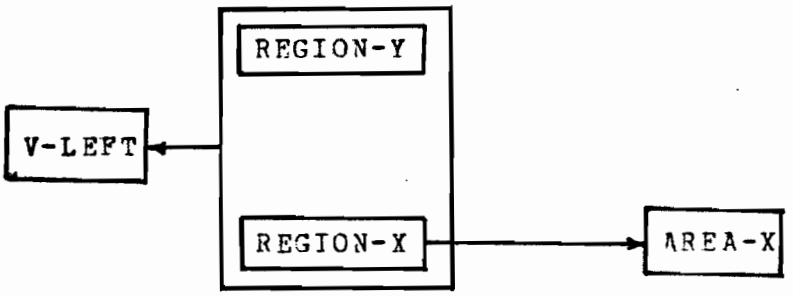


Figure 2.13.b Functional dependence of relation T in 1NF

T1

REGION-X	REGION-Y	V-LEFT
1	3	0.17
2	5	0.32
2	7	0.46
2	23	0.15
23	8	0.26
23	19	0.92

T2

REGION-X	AREA-X
1	75
2	239
23	250

Figure 2.13.c Relations T1 and T2 in 3NF

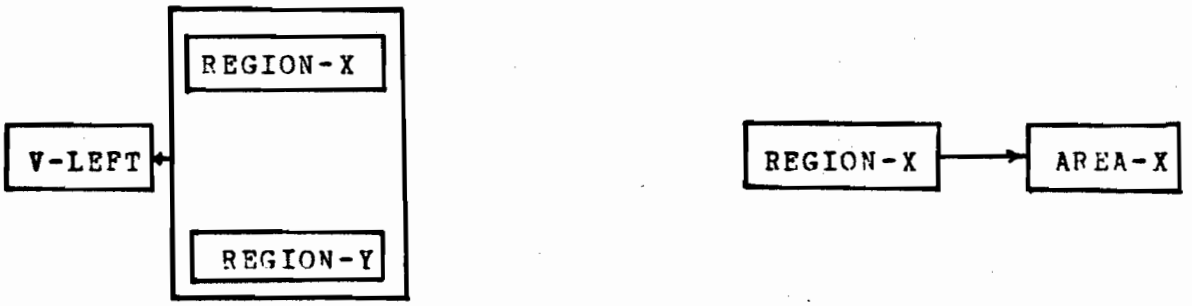


Figure 2.13.d Functional dependence of relations T1 and T2 in 3NF

and since it is not in the 2NF (nor the 3NF) it must be in the 1NF. Obviously we will encounter anomalies in attempting the operations of adding, deleting or updating the data in the relation. Again the solution is to replace the relation T in 1NF by two 3NF relations T1 and T2 (figure 2.13). In general the process of replacing a 1NF relation by a number of 2NF relations is achieved by projecting[1] the relations over different domains to eliminate nonfully functional dependencies. Projections of 2NF relations to eliminate any transitive dependencies (mutual dependence between non-key domains) will produce a collection of 3NF relations.

From the above discussion, it is obvious that casting the database in 3NF will remove (in most cases) the problems associated with the various storage and access operations. In addition, the 3NF ensures the consistency of the data in the database and maintains data integrity by eliminating redundancies. The user is also provided with a very clear definition of the data, which is an obvious benefit. Thus at all times he is sure that data consistency is maintained, irrespective of any storage operations that are performed.

The formal explanation and definition of the relational model, normal forms and other problems associated

---

[1] The term PROJECTION is defined in the next section.

with 3NF (e.g. inter relation dependencies), can be found in Codd [21], Rissanen and Delobel [91] and Date [25]. In the next section, we will review the relational algebraic operations as defined by Codd [20], Delobel and Date [25].

2.5 RELATIONAL ALGEBRA: A DATABASE SUBLANGUAGE

A database system may be classified according to the data model with which the user interacts and the language provided for the user to express this interaction. We have briefly reviewed the three most favoured data models with some emphasis on the relational model, a relatively new data model proposed by Codd [24]. Then we have introduced the concept of data normalization and the three normal forms as defined in [21, 22, and 25]. Now, we will describe the data sublanguage used for data retrieval from the Short and Long Term Memories. We use the term sublanguage (rather than language) because we are not concerned with the capability of computing all computable functions or general processing. Instead, we need to focus only on those language components which support storage and retrieval of data from large shared database systems.

Two principle types of sublanguages have emerged for

the relational data model; namely, relational calculus and relational algebra. The former is based on applied predicate calculus and was first proposed by Codd [23] as a data sublanguage for relational database systems. He defined a calculus based data sublanguage "ALPHA" which is founded on applied predicate calculus [23], and [25]. The relational calculus is simply a notation for expressing the definition of a particular relation to be derived from the data model. It is a very simple language to use because the user need only define the data to be retrieved in a predicate calculus format. He need not know the details of the retrieval operations, how they will be applied, or the order of their application. However the data to be retrieved must be described in the predicate calculus format provided by the sublanguage. On the other hand, data retrieval in the algebraic approach is viewed as the formation of a new relation from the database relations by means of algebraic operations. It should be stressed here that these act upon entire relations as their operands. The user is expected to define not only the algebraic operations to be performed and their operands but also the order of their execution. He is expected to be aware of the database relations, domains and the operations needed for data retrieval. Before deciding on a sublanguage to be employed in the computer vision system, we should examine closely the needs and qualifications of the users of vision

systems.

In general, the user of the computer vision system is either a human operator who will experiment interactively with the system or a computer programmer who will also be involved in the design and implementation of one or more of the system processes. The user of the computer vision system might be expected to be the same type as the casual user of commercial database systems, although admittedly this level of sophistication has not yet been needed. The programmer on the other hand is assumed to be aware of the details of the database relations and domains and is experienced enough to be able to write the necessary algebraic expressions needed for data retrieval. Since at this stage we are primarily concerned with the programmer, and it is not also obvious what specific form a vision calculus should take, we have chosen tc implement a relational algebra. Codd [20] has defined a relational algebra which possesses as much selective power as the relational calculus and is much simpler to implement. He has presented an algorithm for reducing an arbitrary relation-defining expression (based on the calculus) into a semantically equivalent expression of the relational algebra [20]. Thus the implementation of an algebraic-based sublanguage may be considered as the first step towards implementing a calculus-based sublanguage. Moreover, the

algebraic approach has the advantage of being much simpler to implement. In fact, the algebraic approach has been favoured by researchers working in the field of fact retrieval and deductive inference, as has been discussed in Codd [20], Levein and Maron [64] and Bell [15].

Because of the above arguments, the relational algebra as defined in [20] was adopted as a sublanguage for data retrieval in the current implementation of the relational database. The primary objective of this algebra is to provide a collection of operations on relations of all degrees (not necessarily binary) suitable for selecting data from a relational database. The relations to be operated upon are assumed to be normalized as defined in section 2.4. We will start by defining the relational algebraic operations. These definitions follow closely the ones given in Codd [20] and Date [25]. Later, we will discuss the use of the algebra in the computer vision system and the notation used in the query language.

First, we will introduce the notation that will be used in the definitions of the algebraic operations. Suppose r is a tuple of the n-ary relation R. The notation r(Di) designates the value of domain Di in the tuple r where Di is a domain in the relation R. We extend the notation to a list A=(D1, D2, ..., Dn) of domains (not necessarily

distinct) from the domains of the relation R as follows:

$r(A) = (r(D1), r(D2), \ldots, r(Dn))$.

Also, the relation R can be represented as:

$R = (r : r \in R)$.

A simple domain is a set, all of whose elements are numbers, or a set, all of whose elements are character strings. In the remainder of this study, whenever the term domain is used without further qualification, it means simple domain. Also, by the term relation we mean a simple normal relation, i.e. a relation defined only on simple domains. Two simple domains are union-compatible if both are domains of numbers or both are domains of character strings. We will now proceed to define the algebraic operations that are used in retrieving data from the relational model.

1- UNION

The UNION of two relations R1 and R2 is the new relation R3 that consists of all tuples in relation R1 and all tuples in relaticn R2. This operation can only be performed on relaticns that are union-compatible (i.e. relations R1 and R2 must have union-compatible domains). The UNION operation can be expressed as:

$R1 = (r_1 : r_1 \in R1)$

$R2 = (r_2 : r_2 \in R2)$

$R3 = R1 \cup R2$

$R3 = (r_1 \cup r_2 : r_1 \in R1, r_2 \in R2)$

Figures 2.14.c is an example of the UNION operation applied to the relations shown in figures 2.14.a and 2.14.b.

2- INTERSECTION

The INTERSECTION of two relations R1 and R2 is the new relation R3 that consists of the tuples that exist in both R1 and R2. Again this operation can only be performed on relations that are unicn-compatible. It can be expressed as:

R1

REGION	AREA	INTERPRETATION
2	200	SKY
3	179	TREE
18	87	TREE
27	137	WALL
38	81	DOOR
57	252	WALL

(a) Relation R1

R2

REGION	INTERPRETATION	AREA
1	TREE	352
3	TREE	179
22	WALL	107
27	WALL	137
70	DOOR	47

(b) Relation R2

Figure 2.14 Examples of relational algebraic operations

R3

REGION	AREA	INTERPRETATION
2	200	SKY
3	179	TREE
18	87	TREE
27	137	WALL
38	81	DOOR
57	252	WALL
1	352	TREE
22	107	WALL
70	47	DOOR

(c) UNION R3= R1 U R2

R4

REGION	AREA	INTERPRETATION
3	179	TREE
27	137	WALL

(d) INTERSECTION R4= R1 $\wedge$ R2

Figure 2.14 (cont.) Examples of relational algebraic operations

R5

REGION	AREA	INTERPRETATION
2	200	SKY
18	87	TREE
38	81	DOOR
57	252	WALL
1	352	TREE
22	107	WALL
70	47	DOOR

(e) SYMMETRIC-DIFFERENCE R5 = R1 $\triangleq$ R2

R6

REGION	INTERPRETATION
2	SKY
3	TREE
18	TREE
27	WALL
38	DOOR
57	WALL

(f) PROJECTION R6 = R1 $\not\%$ [REGION, INTERPRETATION]

Figure 2.14 (cont.) Examples of relational algebraic operations

R7

INTERPRETATION
SKY
TREE
WALL
DOOR

(g) PROJECTION R7 = R1 % [INTERPRETATION]

R8

INTERPRETATION
SKY
WALL

(h) Relation R8 is used to get the regions that are interpreted as either SKY or WALL

R9

REGION	AREA	INTERPRETATION
2	200	SKY
27	137	WALL
57	252	WALL

(i) JOIN R9 = R1 * R8

Figure 2.14 (cont.) Examples of relational algebraic operations

R10

REGION-X
19

(j) Relation R10 is used to get the information about REGION 19 from relation LEFT-OF

R11

REGION-X	REGION-Y	V-LEFT
19	1	0.82
19	8	0.25
19	22	0.37

(k) JOIN R11 = LEFT-OF * R10

Figure 2.14 (cont.) Examples of the relational algebraic operations

ADJACENT

REGION-X	REGION-Y
1	2
1	3
1	5
2	1
2	5
2	12
2	20
3	1
3	5
3	8
3	12
.	.
.	.
.	.

(l)

R12

REGION-Y
5
12

(m)

R13

REGION-X
2
3
.

(n)

DIVISION R13 = ADJACENT [REGION-Y ÷ REGION-Y] R12

Figure 2.14 (cont.) Examples of the relational algebraic operations

R14

REGION	AREA	INTERPRETATION
2	200	SKY
18	87	TREE

(o) RESTRICTION R14 = R1 : (INTERPRETATION = TREE & AREA < 100 ∂ INTERPRETATION = SKY)

R15

REGION	AREA	INTERPRETATION
2	200	SKY
3	179	TREE
27	137	WALL

(p) RESTRICTION R15 = R1 : (AREA ⩾ 100 & AREA ⩽ 200)

Figure 2.14 (cont.) Examples of relational algebraic operations

$R1 = (r_1 : r_1 \in R1)$

$R2 = (r_2 : r_2 \in R2)$

$R3 = R1 \cap R2$

$R3 = (r_1 \cap r_2 : r_1 \in R1, r_2 \in R2)$

Figure 2.14.d is an example of the INTERSECTION operation.

3- DIFFERENCE

The DIFFERENCE of two relations R1 and R2 is a new relation R3 that contains the tuples in relation R1 which are not tuples of relation R2. It is expressed as:

$R1 = (r_1 : r_1 \in R1)$

$R2 = (r_2 : r_2 \in R2)$

$R3 = R1 - R2$

$R3 = (r_1 : r_1 \in R2)$

A more popular operation is the SYMMETRIC DIFFERENCE which is defined as follows:

The SYMMETRIC DIFFERENCE of two relations R1 and R2 is a new relation R3 that contains the tuples in relation R1 which are not tuples of relation R2 and the tuples in relation R2 which are not tuples of relation R1. It is

expressed as:

$R1 = (r_1 : r_1 \in R1)$

$R2 = (r_2 : r_2 \in R2)$

$R3 = R1 \triangleq R2$

$R3 = (r_1 \cup r_2 : r_1 \notin R2, r_2 \notin R1)$

Figure 2.14.e is an example of the SYMMETRIC DIFFERENCE operation. Again, both DIFFERENCE and SYMMETRIC DIFFERENCE operations are applicable only to union-compatible normal relations.

4- PROJECTION

The PROJECTION of relation R1 over the list of domains L which is a subset of the domains of R1 is a new relation R3 which is defined over the set of domains L and retains the values of the tuples in R1 over the domains in L. Suppose r is a tuple of the n-ary relation P. For D1, D2, ..., Dn the notation r(Di) designates the tuple value of the domain Di. We extend the notation to a list A=(D1, D2, D3, ..., Dn) of domains not necessarily distinct from the set of domains in R as follows:

$r(A) = (r(D1), r(D2), ..., r(Dn))$

Then the PROJECTION of R1 on the list A is defined as follows:

$$R3 = R1\ (A)$$

$$R3 = (\ r_1\ (A)\ :\ r_1 \in R1\ )$$

Note that when A is a permutation of the domains of R, R(A) is a relation whose domains are the same as those of R except for change in order of appearance. In figure 2.14.f, we project relation R1 over domains REGION and INTERPRETATION. If we need to know all of the possible interpretations in the relation R1, it is only necessary to project it over the domain INTERPRETATION as shown in figure 2.14.g. We will refer to the PROJECTION by the symbol %.

5- NATURAL JOIN

The natural join of two relations R1 and R2 containing a common domain (or set of domains) D is a new relation R3 which consists of all concatenations of R1-tuples and R2-tuples containing the same D component (or components) [25]. Given relation R8 in figure 2.14.h, figure 2.14.i shows an example of the JOIN operation. It follows from the definition that if a tuple in one of the two relations contains a D-value (values) that does not appear in any of

the tuples of the other relation, that tuple does not participate in the JOIN. Thus the JOIN operation represents a powerful tool for the extraction of just those tuples containing a pre-specified value (or set of values). For example, if we need to obtain the information about REGION 19 in the relation LEFT-OF in figure 2.6, we need only JOIN it with a relation that contains REGION 19 as shown in figure 2.14.j and 2.14.k. We will use the symbol * to represent the JOIN operation.

6- DIVISION

Suppose R1, R2 are relations and A,B are union-compatible lists of domain occurences A in R1 and B in R2. R1 may be considered a binary relation on compound domains A and $\bar{A}$ where $\bar{A}$ is the complement of A in R1. For example, if R1 is defined over domains D1, D2, D3, and D4 and A is defined as domains D1, and D3, then $\bar{A}$ is considered to be D2 and D4. Let the image set $g_{R1}(r_1(\bar{A}))$ of the tuple $r_1(A)$ under R1, be the set of tuples $(r_1(A))$ such that R1 maps $r_1(A)$ into that set

$$g_{R1}(r_1(\bar{A})) = (r_1(A) : (r_1(A), r_1(\bar{A})) \in R1).$$

Then the DIVISION of R1 on A by R2 on B is defined as:

$$R1(A \div B)R2 = (r_1(\bar{A}) : r_1(R1 \cap R2(B) \subseteq q_{R1}(r_1(\bar{A}))$$

where R1(A) and R2(B) are union-compatible. We should point out that when R1 is empty, R1 divided by R2 is empty, even if R2 is also empty. Actually, DIVISION is definable in terms of the operations already introduced [20]:

$$R1(C \div D)R2 = R1(\bar{C}) - ((R1(\bar{C}) \otimes R2(D)) - R1)(\bar{C})^{1}$$

For example, if we have a relation ADJACENT which describes the adjacency of regions (REGION X is ADJACENT to REGION Y) and we would like to obtain all the regions that are adjacent to both regions 5 and 12, we divide relation ADJACENT over relation R12 that has two tuples 5 and 12. The result is shown in figures 2.14.1, 2.14.m and 2.14.n.

7- RESTRICTION

Suppose A, B are two domains in a relation R. Let θ denote any of the relations $=$, $\neq$, $<$, $\leq$, $>$, and $\geq$. The θ RESTRICTION of R on domains A, B is a new relation that contains the tuples of R that have $r(A)\ \theta\ r(B)$ true. Thus, the θ RESTRICTION of R on domains A, B is defined by:

---

[1] The symbol $\otimes$ refers to an EXCLUSIVE OR operation

$R(A\ \theta\ B) = (r : r \in R \cap (r(A)\ \theta\ r(B)))$

provided that every element of R(A) is θ-compatible with every element of R(B). The RESTRICTION operation is a very powerful operation for selecting specific data. In the current implementation, we have extended its use to include a list of θ-conditions. Thus the RESTRICTION operation on a relation is defined by :

$R : (EXPRESSION) = (r : r \in R \cap EXPRESSION)$

The EXPRESSION is a BOOLEAN expression of the form:

$(A_{11}\ \theta_{11}\ B_{11})\ \&\ (A_{12}\ \theta_{12}\ B_{12})\ \&\ \ldots\ (A_{1n}\ \theta_{1n}\ B_{1n})\ @\ (A_{21}\ \theta_{21}\ B_{21})\ \ldots\ (A_{2n}\ \theta_{2n}\ B_{2n})$ [1]

where A , and B are either domain names in the relation R or values of tuples. Thus if the user wishes to obtain all the regions in relation R1 of figure 2.14.d that have interpretation TREE and AREA less than 100 or interpretation SKY, he uses the following expression:

R1 : (INTERPRETATION = TREE & AREA < 100 @ INTERPRETATION = SKY)

The result is shown in figure 2.14.0. As another example

---

[1] The symbols & and @ refer to logical AND and OR operations, respectively.

(figure 2.14.p), suppose we need all regions in R1 that have their area in the range from 100 to 200; we use the expression:

R15 = R1 : (AREA $\geqslant$ 100 & AREA $\leqslant$ 200).

The symbol : has been used to represent the RESTRICTION operation.

The conventions for evaluating expressions are similar to those encountered in algebra. That is, the expression is evaluated left to right within the limits imposed by the priorities of the operators in the expressions. The available operators in decreasing priority are :

: and %	RESTRICTION and PROJECTION
* and $\div$	JOIN and DIVISION
$\cap$	INTERSECTION
$\cup$, - and $\triangle$	UNION, DIFFERENCE and SYMMETRIC DIFFERENCE

Moreover, parentheses may be used in the usual way to force a desired order of evaluations as will be illustrated in the following examples:

(i) X * Y % [D1, D2]

(ii) (X * Y) % [D1, D2]

(iii) X ÷ Y * W

In example (i), relation Y is a PROJECTION over domains D1 and D2, and the result is then a JOIN with relation X. In example (ii), relations X and Y first appear in a JOIN operation and we then obtain the PROJECTION result over domains D1 and D2. In example (i) the PROJECTION was performed first since it has a higher priority than JOIN. However in example (ii), the existence of parentheses enforces a certain order of execution. Example (iii) seems to be ambiguous as it is not obvious whether the DIVISION of X and Y is performed first followed by the JOIN operation, or the JOIN of Y and W is performed first followed by the DIVISION operation. The rule is that the expression is evaluated from left to right, unless the priority of the operators dictates otherwise. Thus X divided by Y is performed first followed by the JOIN operation.

We have implemented the UNION, INTERSECTION, PROJECTION, JOIN, SYMMETRIC DIFFERENCE and RESTRICTION operations to be used in the data sublanguage for both the SHORT and LONG TERM MEMORIES. The current implementation allows the user to input, delete, update and retrieve data in both interactive and noninteractive modes. In the

interactive mode, the human operator (the user) is offered the different functions as a menu. The database system carries out the selected functions and requests the user to input the necessary information needed for the completion of these functions. For example, if the user decides to create a new relation, the database system requests him to enter the name of the new relation, the number of domains, the maximum number of tuples, the names of the domains, etc. Throughout such an interactive session, the system checks on the current database information and displays error messages to the user when it detects any errors. For example, if the user attempts to create a relation with the same name of an already existing relation, an error message is displayed and he is asked to select another name. The user is offered the following interactive functions:

1- CREATE: Creates a relation.

2- EDIT: This is a relation editor that can be used to add, delete or update a relation.

3- DELETE: This is used to delete relations from the database.

4- DISPLAY: This is used to list the database relations.

5- ALGEBRA: This is used to retrieve information from the database using the relational algebra.

If the user selects the ALGEBRA function, the system asks him to enter an algebraic expression. After executing the algebraic expression, the result is displayed on the user's terminal. The user can then ask the system to save this result for later use. A detailed description of these functions can be found in Shaheen [108]. The above-mentioned functions can also be activated non-interactively from within a computer program as they must be by the computer vision processes. Throughout this thesis, the term "user" will refer to both the human who operates on the database relations interactively as well as the computer programs that use the database non-interactively.

2.6 SHORT AND LONG TERM MEMORIES: A RELATIONAL MODEL

As mentioned in Chapter I, the Short Term Memory (STM) contains the regions' intrinsic features, the structural relationships between the regions, shape and inferences deduced by the processes of the system. This information is stored in a set of relations that will be described in detail in the next chapter. The system processes retrieve the data from the STM using the relational algebra. For example, the HYPOTHESIS VERIFIER/INTERPRETER employs the RESTRICTION and PROJECTION operations to obtain all of the

regions adjacent to a specific region, using the expression:

(ADJACENT : (REGION-X = N) % (REGION-Y, V-ADJACENT))

where N is the region number and V-ADJACENT is the value of adjacency.

The LTM is also designed as a relational database that contains model information about the class of scenes under study as well as certain vision strategies used by the processes during the analysis as will be described in Chapter IV. Information here is retrieved using the relational algebra and currently the LTM is not changed during the processing. That is, the system processes are only able to interrogate the LTM for the information needed to achieve their processing task but do not change it. The LTM is generated using the interactive editor of the database. As described above, this editor is able to access the LTM adding, deleting and updating any of its tuples. Moreover, it may be used to interactively define new relations in the database.

We will now consider some examples illustrating the algebraic expressions for the retrieval of information from the database using the relational algebra.

(i) Obtain the regions with interpretation SKY that have an area greater than 500:

(REGION-DESCRIPTION * INTERPRETATION) : (AREA > 500, ℰ INTERPRETATION = SKY) % (REGION)

A JOIN on relations REGION-DESCRIPTION and INTERPRETATION will produce a relation that contains all of the regions and their interpretations. The RESTRICTION operation will select the regions that have AREA > 500 and INTERPRETATION = SKY. Alternatively, the same information could be retrieved using the expression:

[REGION-DESCRIPTION : (AREA > 500) * INTERPRETATION : (INTERPRETATION =SKY)] % (REGION)

Here, two RESTRICTION operations are applied to REGION-DESCRIPTION and INTERPRETATION to obtain the regions that have AREA > 500 or INTERPRETATION =SKY. Then, the JOIN operation on the result of the two RESTRICTION operations produces the required region numbers. Finally, a PROJECTION operation is applied to obtain a relation that contains the region numbers that satisfy the constraints of the example.

(ii) Obtain the regions that are adjacent to both regions 1 and 2.

ADJACENT : (REGION-X =1) % (REGION-Y) ∩ ADJACENT : (REGION-X=2) % (REGION-Y)

First we find the regions ADJACENT to REGION 1 by a RESTRICTION operation, and then regions ADJACENT to REGION 2 using another RESTRICTION operation. Each RESTRICTION operation is followed by a PROJECTION over REGION Y. Finally, we apply INTERSECTION to the results of the previous operations to get the required result.

(iii) Obtain the interpretations of the regions that have their area greater than their perimeter.

[REGION-DESCRIPTION : (AREA > PERIMETER) % (REGION) * INTERPRETATION] % (INTERPRETATION)

Here the RESTRICTION operation is followed by the PROJECTION operation to produce a relation that has regions that have their AREA greater than their PERIMETER. We obtain the required data when the result is JOINed with the INTERPRETATION relation and then projected only over domain INTERPRETATION.

The previous examples show how the application programs (system processes) or the human operator retrieves the information necessary for its processing using the relational algebra. He need only define the algebraic expression and the database software will automatically retrieve the correct information. Having described the relational database, its use as SHORT and LONG TERM MEMORIES, and the relational algebraic sublanguage implemented, we will proceed to describe the STM and LTM relations in detail in the next chapters. We should point out that the relations given in the previous examples of this chapter are simply for the purpose of illustration. Samples of actual relations used by our system are found in Appendices I and II.

CHAPTER III

SHORT TERM MEMORY

3.1 INTRODUCTION

The primary role of the SHORT TERM MEMORY is to act as a buffer or communication channel between the different processes of the system. It is implemented as a relational database [25] and may be considered as an associative memory for the immense amount of data required for analyzing real world pictures. The STM contains measurements made on the input data as well as the intermediate and final deductions of the different processes in the system. The data in the STM may be represented at several levels of abstraction. In this way, it has many similarities to the concept of the "blackboard" in "HEARSAY II" [62]. For example, the STM could contain the three colour image planes, edge and line information, atomic regions as well as clusters of merged regions. All the processes of the system have equal access to all of the data in the STM, therefore, they do not have to repeat calculations done by any of them.

As described before, the STM is used as a communication channel between the processes of the system. Each process gets its input from the STM and transmits its intermediate

and final results to it. Thus there is no direct communication between the processes, which enforces the required level of independence between them. Because the STM is implemented as a relational database, the processes of the system view the data in it through the relational data model. Hence, a process need only know the data model (i.e. the relational data model) and the communication language of the database. As discussed in Chapter II, the communication language is designed and implemented as a relational algebraic sublanguage. All the data in the STM are viewed in the form of relations as defined in Chapter II.

In what follows, we will assume that STM and STM relations are synonymous terms which will be used interchangeably. In the next few sections, we will define the existing STM relations and the measurements done to obtain the data stored in them. Samples of these relations are found in Appendix I.

3.2 REGION MAP RELATION

The REGION MAP relation is generated by the LOW LEVEL SEGMENTATION process as will be described in section 5.2. It represents the current segmentation of the picture under

analysis. This relation is essentially a labelling of the existing connected components of the picture at any one time. These components are initially generated by the LOW LEVEL SEGMENTATION process, but are also later updated by the FEATURE ANALYZER when regions are merged during the analysis.

The REGION MAP relation has n+1 domains and is defined as:

REGION MAP = (ROW-NO, COLUMN-1, COLUMN-2, ..., COLUMN-n)

where:

n represents the number of columns in the input picture.

Each tuple in the relation contains the row number in the first domain and n values that correspond to the n labels attached to the pixels of this row in the picture under analysis. All the pixels that belong to the same region are labelled with a unique label that identifies that region. Region labels are chosen to be positive integer numbers. Obviously, the relation will contain a number of tuples equal to the number of rows in the image. Figure 6.2 is a graphical display of the REGION MAP where all the pixels in the same region are given the same colour and adjacent regions are given different colours.

3.3 REGION FEATURES RELATION

Since regions are the basic entities in the system presented in this thesis, we must compute the intrinsic attributes of each region stored in the REGION MAP. These attributes are maintained in the REGION FEATURES relation which is defined as follows:

REGION FEATURES = (REG-NO, FLAG, FEAT-1, FEAT-2, FEAT-3,
VAR-1, VAR-2, VAR-3, INTEN, CENT-X,
CENT-Y, MIN-X, MIN-Y, MAX-X, MAX-Y,
AREA, PERIMTR, CIRCLRTY, ASPECT-R).

The domain REG-NO is the only key domain of the relation and represents the region label in the REGION MAP. It is a positive integer number that identifies this region throughout the analysis. FLAG is a domain that indicates whether a certain region is active or not. That is, if the region exists in the REGION MAP, it is set to 0. On the other hand, when the region is not in the REGION MAP, it is set to -1. In general, this happens when two or more of the regions are merged during the analysis generating a new region. In this case, the FEATURE ANALYZER deactivates the merged regions by setting their FLAG domain to -1. It also calculates the features of the newly generated region and inserts them in the REGION FEATURES relation. FEAT-1, FEAT-2 and FEAT-3 refer to the three colour features used to represent the input colour picture. These may be RED, GREEN

and BLUE, or INTENSITY, HUE and SATURATION [57], or any other colour system[1]. In particular, these domains represent the average of each of these features over the region, while the VAR-1, VAR-2, and VAR-3 domains contain the value of the standard deviation of the three colours over the region. The average intensity over the region is stored in the domain INTEN.

The centroid of the region is the normalized first order moment of the region. We used the cartesian system to represent position information in the picture, where the lower left corner of the picture corresponds to the point (1,1) in the cartesian space (figure 3.1). Thus the centroid of a region R is stored in the domains CENT-X and CENT-Y and is calculated as follows:

$$\text{CENT-X} = \frac{\sum_{\text{MIN-X}}^{\text{MAX-X}} \sum_{\text{MIN-Y}}^{\text{MAX-Y}} X.\text{PICT}(X,Y)}{\sum_{\text{MIN-X}}^{\text{MAX-X}} \sum_{\text{MIN-Y}}^{\text{MAX-Y}} \text{PICT}(X,Y)} \qquad (3.1)$$

$$\text{CENT-Y} = \frac{\sum_{\text{MIN-X}}^{\text{MAX-X}} \sum_{\text{MIN-Y}}^{\text{MAX-Y}} Y.\text{PICT}(X,Y)}{\sum_{\text{MIN-X}}^{\text{MAX-X}} \sum_{\text{MIN-Y}}^{\text{MAX-Y}} \text{PICT}(X,Y)} \qquad (3.2)$$

where:

$$\text{PICT}(X,Y) = \begin{cases} 1 & \text{if point } (X,Y) \text{ is labeled as region } R_i \\ 0 & \text{if point } (X,Y) \text{ is not labeled as region } R_i \end{cases} \qquad (3.3)$$

---

[1] RED, GREEN and BLUE are the colour features used throughout the subsequent analysis with 64 gray levels.

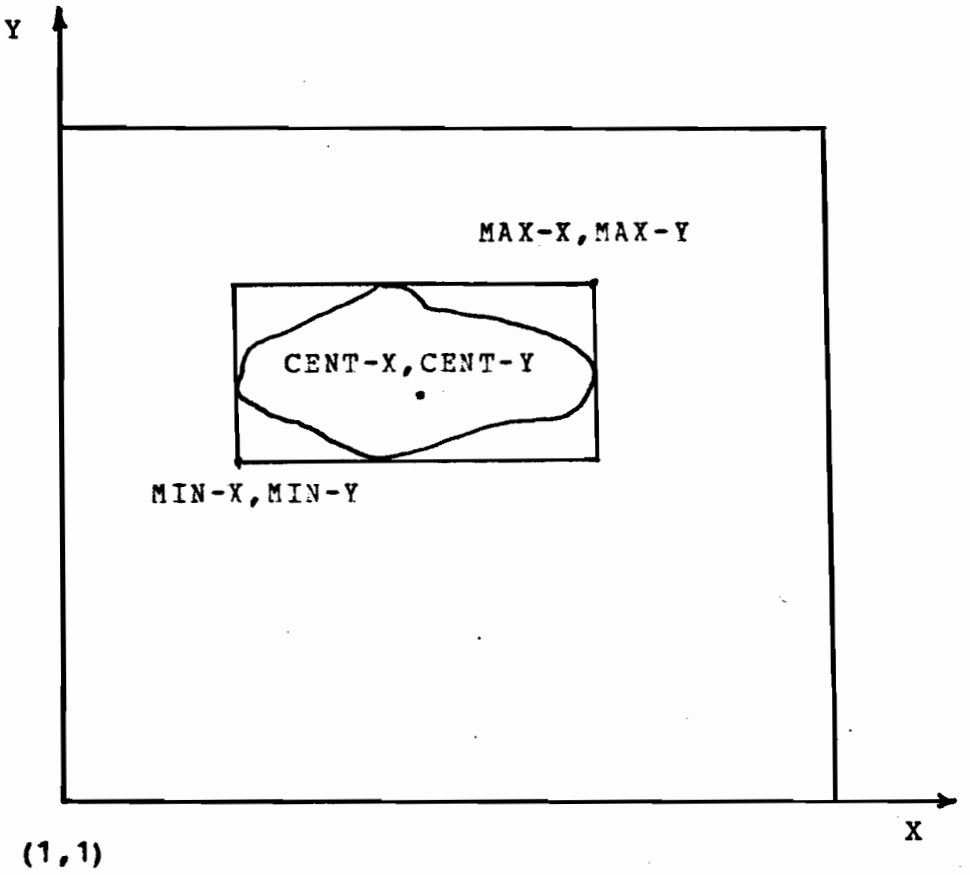


Figure 3.1 Binary image of a region and the position parameters calculated for it.

The minimum rectangle of a region is defined as the rectangle which would enclose the region in the REGION MAP. The orientation of the rectangle is parallel to the X and Y axis. This feature is very important for calculating other features such as the spatial relations, as will be shown later. It is clear from figure 3.1 that the minimum rectangle can be defined in terms of the position of its two vertices in the lower left and upper right hand corners. The values of the X and Y positions of these two vertices are stored in the domains MIN-X, MIN-Y, MAX-X, and MAX-Y respectively.

The area of a region R_i is here defined as the number of pixels labelled as region R_i in the REGION MAP and is stored in the domain AREA. The perimeter of the region is measured by following the region's contour and is stored, in terms of units, each of which is equal to one pixel in length, in the domain PERIMTR.

Finally, the aspect ratio [30] and circularity are stored in the domains ASPECT-R and CIRCLRTY, respectively. The aspect ratio is defined as the ratio of the length to width of a region. Its use is questionable since it can often be a misleading value in instances wheere the major

axis of the region is askew. It is calculated from the size of the minimum rectangle, defined above, as follows:

$$\text{ASPECT-R} = \frac{\text{MAX-Y} - \text{MIN-Y} + 1}{\text{MAX-X} - \text{MIN-X} + 1} \qquad (3.4)$$

The circularity of a region is a relative measure of the ratio between its area and the square of its perimeter. Its value varies between 0 and 1, the latter value obtained for a circle. Circularity [30] has often been used as a measure of "smoothness", or the "complexity" of the boundary since the more complex a boundary, the longer the perimeter. This ratio has also been used as a measure of non-compactness of figures; however, Rosenfeld [98] disputes the reliability of this measure for a digitized picture. It is computed as follows:

$$\text{CIRCLRTY} = \frac{4\,\pi\ \text{AREA}}{\text{PERIMTR}^{\,2}} \qquad (3.5)$$

Although we have computed many features for the regions, the features used to distinguish regions in the reported experiments were the cclour and vertical position features.

3.4 SPATIAL RELATIONS

Throughout the analysis the initial segmentation produced by the LOW LEVEL PROCESSOR is assumed to be partial. That is, the regions in the initial segmentation do not necessarily correspond to complete objects in the scene under study. Thus, the specific descriptions of the individual objects are useless and inapplicable; however, the relative position of the objects is useful since it is a more general description. This generality allows one structural description to define a variety of images within the class under study. This is the case in outdoor scenes where the relative position of SKY, TREES, ROOF and GRASS provides a description which is invariant from scene to scene. For example, the SKY is usually expected to be above everything and the ROOF is usually below the SKY or TREES, etc. Thus the relative position of the regions in the partial segmentation provides strong contextual clues to their interpretation.

Freeman [35] discusses how people encode the spatial relations between objects in a picture. He also describes some mathematical and computational formalisms which can be used to embody their semantic context. In this study, he concludes that some sort of relational data structure would be appropriate to represent these relations. He also showed

that a fuzzy relational structure would be particularly appropriate to model the continuum from applicability to non-applicability that characterizes most of these relations. Consider for example, the relation LEFT-OF in figure 3.2. We can easily see that LEFT-OF(A,B) is clearly applicable in figure 3.2.a and clearly inapplicable in figure 3.2.f. Briefly, an ordinary binary relation X over pairs of objects A and B from some object space S can be defined by its characteristic function:

$$q_X(A,B) = \begin{cases} 1 & \text{if } X(A,B) \text{ holds} \\ 0 & \text{otherwise} \end{cases} \qquad (3.6)$$

for all A,B in S.

This representation cannot encode the relation LEFT-OF(A,B) in figures 3.2.b and 3.2.c. Defining LEFT-OF as a fuzzy relation would allow the degree of relatedness for the cases in between those cases shown in figures 3.2.a and 3.2.f. A fuzzy relation would be one whose characteristic function q_X associates each pair of objects in the space S with some number in the interval (0,1). A value of $q_X(A,B)$ closer to 1 indicates a greater relatedness between A and B. Thus, the characteristic function of LEFT-OF would map the relation LEFT-OF(A,B) into 1 or some number closer to 1 for figure 3.2.a. Each successive figure is mapped into a smaller number, until LEFT-OF(A,B) of figure 3.2.f would be equal to 0.

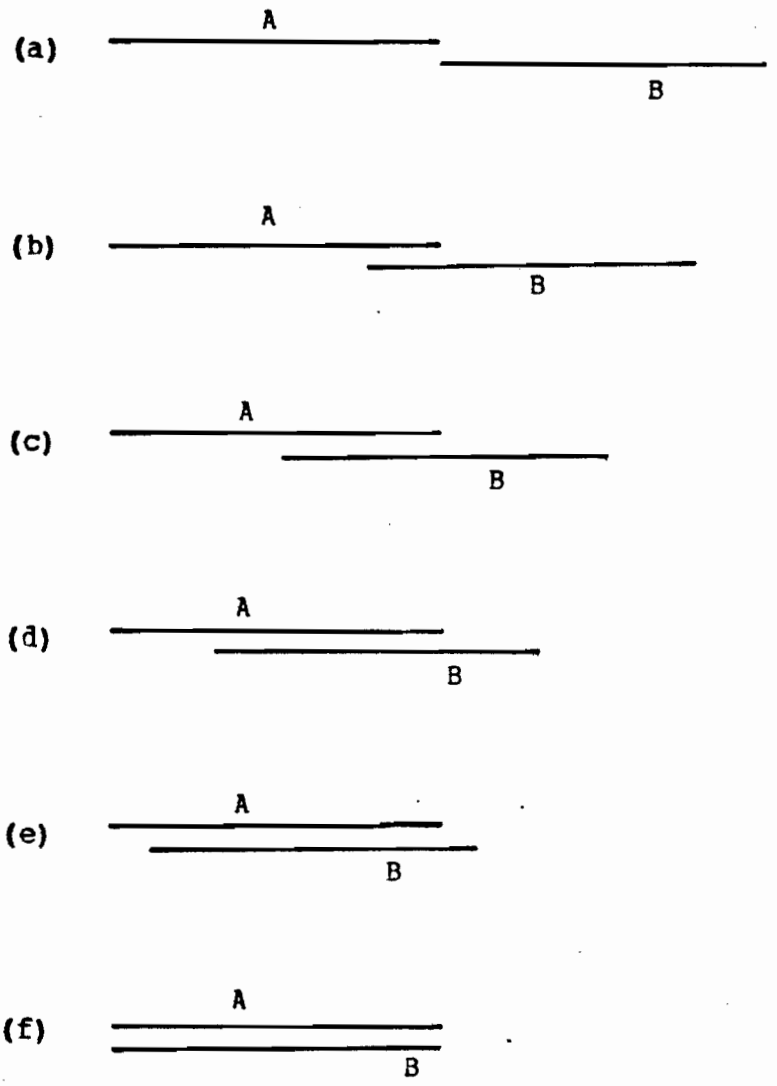


Figure 3.2 Illustration of the fuzziness of a spatial relation LEFT-OF.

The fuzzy representation of the spatial relation is more general than the binary representation. The binary representation can be obtained by introducing a threshold value $0<t<1$ such that if:

$g_X(A,B) < t$, the relation X does not hold,

$g_X(A,B) \geq t$, the relation X holds.

Freeman also suggested another mapping where two threshold values t_1 and t_2 are defined $0<t_1<t_2<1$ such that if:

$g_X(A,B) \leq t_1$, the relation X does not hold,

$g_X(A,B) \geq t_2$, the relation X holds,

$t_1 < g_X(A,B) < t_2$, the relatedness is intermediate.

In [35] Freeman argued that:

i) this approach gives a response that is more human-like, and

ii) the strictness or laxity of a relational definition is localized in the threshold selection.

The spatial relations are represented as fuzzy relations throughout the computer vision system presented here. Those actually calculated are:

1) LEFT-OF

2) RIGHT-OF

3) ABOVE

4) BELOW

5) ADJACENT (or TOUCHING)

6) CONTAINS

7) ENCLOSED-BY

In calculating the relations, the following feature data (figure 3.1) is required:

i) The location of the minimum enclosing rectangle around the region, which is specified by the minimum and maximum coordinates of the vertices of the rectangle denoted by (MIN-X, MIN-Y), and (MAX-X, MAX-Y), respectively.

ii) The location of the centroid (first geometric moment) of the region denoted by (CENT-X, CENT-Y).

Furthermore, to compute the ADJACENCY relation, a chain code description of the region boundary is also required. The REGION MAP is used in the computation of CONTAINS and ENCLOSED-BY. In the following description of the relations, the coordinates belonging to region A and region B are denoted by subscripts A and B respectively. The value of each relation is then computed as REL(A,B) where REL represents any of the spatial relations mentioned earlier. The following formulas are used to define relations LEFT-OF, RIGHT-OF, ABOVE and BELOW:

$$\text{LEFT-OF}(A,B) = \frac{CTX_B - CTX_A + 1}{MXX_A - CTX_A - CTX_B - MNX_B} - \frac{MXX_B - MNX_A}{MXX_A - MNX_A} \qquad (3.7)$$

$$\text{RIGHT-OF}(A,B) = \frac{CTX_A - CTX_B + 1}{MXX_B - CTX_B + CTX_A - MNX_A} - \frac{MXX_A - MNX_B}{MXX_A - MNX_A} \qquad (3.8)$$

$$\text{ABOVE}(A,B) = \frac{CTY_A - CTY_B + 1}{MXY_B - CTY_B + CTY_A - MNY_A} - \frac{MXY_B - MNY_A}{MXY_A - MNY_A} \qquad (3.9)$$

$$\text{BELOW}(A,B) = \frac{CTY_B - CTY_A + 1}{MXY_A - CTY_A + CTY_B - MNY_B} - \frac{MXY_A - MNY_B}{MXY_A - MNY_A} \qquad (3.10)$$

where (see figure 3.1):

CTX, CTY denote CENT-X and CENT-Y, and

MNX, MNY denote MIN-X and MIN-Y, and

MXX, MXY denote MAX-X and MAX-Y.

In all the relations, the resulting values are restricted to be between 0 and 1. In equations 3.7-3.10, use is made of both the centroid location and absolute location of the objects. This allows the relation to respond to the basic shape characteristics of the regions as well as their relative position. The above equations will result in a fuzzy representation of the spatial relation instead of the more restricted binary valued representation. A more detailed description concerning the calculation of the spatial relations is available in [116, 117].

Two regions are said to be adjacent neighbours, if they touch each other. Adjacency is defined as the ratio of the length of the common (shared) boundary between two regions to the perimeter of one of the regions. Thus, it is a measure of the amount by which one region touches another It is computed as:

$$\text{ADJACENT}(A,B) = \frac{\text{shared boundary between regions A and B}}{\text{perimeter of region A}} \qquad (3.11)$$

To determine the points where the boundaries are touching, the chain code descriptions for the two regions are used to determine those points of each chain code which are four connected neighbours. To obtain the regions adjacent to region R , we perform the following algebraic operation:

ADJACENT : (REG-X = R) % (REG-Y)

The containment and enclosure spatial relations are a measure of the amount by which one region engulfs another and vice versa. CONTAINS is defined as the ratio of the area of region B to the area of region A, which engulfs it. ENCLOSED-BY is the ratio of the area of region A to that of region B, which surrounds it. These relations are computed as:

$$\text{CONTAINS}(A,B) = \frac{\text{area of B}}{\text{area of A}} \qquad (3.12)$$

$$\text{ENCLOSED-BY}(A,B) = \frac{\text{area of A}}{\text{area of B}} \qquad (3.13)$$

In computing the CONTAINS and ENCLOSED-BY relations, the REGION MAP is used to ensure that region B actually lies

within region A (for CONTAINS) or that region A lies within region B for (ENCLOSED-BY).

Each of the spatial relations described above are calculated by the FEATURE ANALYZER and stored in a STM relation[1] which is named after it. Thus, LEFT-OF relation is defined by the relation:

LEFT-OF = (REG-X, REG-Y, VALUE)

where:

VALUE is a domain that contains the value of the relation LEFT-OF(REG-X,REG-Y).

As seen in Appendix I, the STM contains LEFT-OF, RIGHT-OF, ABOVE, BELOW, ADJACENT, CONTAINS, and ENCLOSED-BY relations. Since we are only interested in the spatial relationships between direct adjacent pairs of regions, we only compute and store the spatial relationships of these pairs. In addition, if the value of a relationship between two adjacent regions A and B was found to be zero (e.g. ABOVE(A,B) = 0), the value of the relationship is not stored in the corresponding STM relation. That is, relation ABOVE would not have a tuple that relates A and B. This assumption reduces the amount of data stored and does not reduce the ability of the system to find the value of the relationship ABOVE(A,B). As another example, the relation

---

[1] The term relation here refers to the relational data model of the STM.

ENCLOSED-BY contains only the pairs of regions where the value of ENCLOSED-BY spatial relations is greater than zero. Thus if two regions A and B do not appear in the relation, the relation ENCLOSED-BY(A,B) does not hold and is assumed to be zero (i.e. ENCLOSED-BY(A,B) is equal to 0).

Although only the above seven relationships were used, computed and stored in the STM, it is easy to expand these to include other relationships such as INSIDE-OF, OUTSIDE-OF, INFRONT-OF and BEHIND. These were not considered because some (e.g. INFRONT-OF and BEHIND) relate to three dimensional descriptions, while others (e.g. INSIDE-OF and OUTSIDE-OF) are difficult to compute and are not informative. However, they can be easily added to the STM when needed. In this case it is required to add to the FEATURE ANALYZER the procedure that computes them.

3.5 CHAIN CODE RELATION

The chain code is a sequence of contour directions obtained by tracing the boundary of the region. Since each pixel has eight possible neighbours (8-connected figures), the possible directions can be quantized to values between 0 and 7, as shown below in figure 3.3.

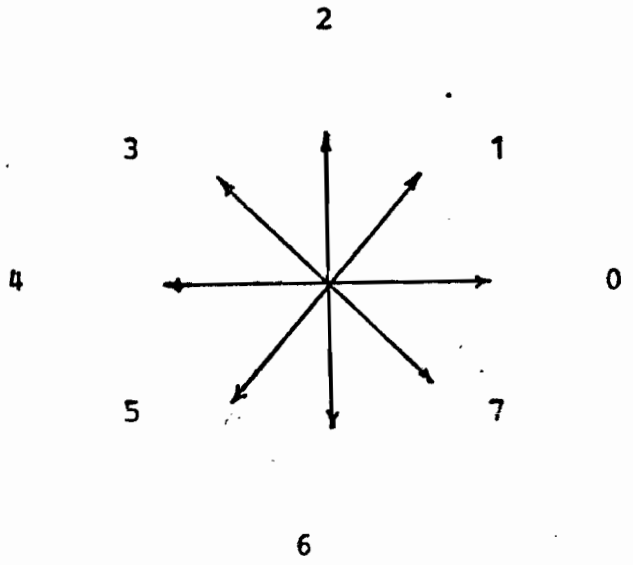


Figure 3.3 Eight possible quantized directions for chain code representation

The boundary is found by using a contour follower suggested by Ledley [61], which follows the path a bug would take if it always kept the figure to its left. This implies that the exterior contour of a region would be followed in a counter-clockwise direction. Once a trace is completed, the chain is fully specified by its starting absolute location and the sequence of contour directions. We only store the exterior boundaries of a region. The interior boundaries of a region R can be easily found by determining the boundaries of all the regions enclosed by the region R using the ENCLOSED-BY relation in the STM.

The algorithm for determining the chain code

representation of the boundary of a region R is as follows:

i) Scan the REGION MAP row by row until a horizontal contour point that belongs to region R is found.

ii) Starting from this point, trace the contour of region R using the bug, until the path returns to the initial starting point. During the contour tracing, obtain the chain code of the contour and store it using the eight directions shown in figure 3.3.

This algorithm is quite fast since the only operation necessary is a comparison of the present pixel value with the previous one. A more formal description of the algorithm is found in Montanari [73], who also discusses a means of generating a connectivity graph, relating the positions of the various contours in the region.

The region's chain code is stored in the STM relation CHAIN-CODE defined as follows:

CHAIN-CODE = (REG-NO, FLAG, LENGTH, X-START, Y-START,
CHAIN-1, CHAIN-2, ..., CHAIN-7).

where:

REG-NO is the region number,

X-START is the value of the X-coordinate of the starting point of the contour,

Y-START is the value of the Y-coordinate of the starting point of the contour,

LENGTH is the length of the chain code,

CHAIN-1, ..., CHAIN-7 are the chain code representation of the region.

The length of the chain code of a region is variable, in the sense that some regions may have a chain code of 100 links while others may have only 5 links depending on the length of the contour. Each domain in the relation is used to store eight chain codes. Each tuple can store 56 links. Thus if a region has a chain code of length less than or equal to 56, its chain code is put in one tuple (e.g. region number 1 in relation CHAIN-CODE in Appendix I). If a region's chain code has a length greater than 56, we store the first 56 in a tuple whose FLAG domain has the value 0, the next 56 chains are stored in another tuple with the FLAG set to 1, and so on until all the chains of the regions are stored (e.g region number 2 in CHAIN-CODE relation in Appendix I).

To obtain the chain code of a region R_i, we perform the following algebraic operation:

CHAIN-CODE : (REGION = R_i)

This will produce a new relation that contains only the tuples whose REGION dcmain equals R_i. We can also obtain the same result by a JCIN operation between the CHAIN-CODE relation and a temporary relation called TEMP, as follows:

CHAIN-CODE * TEMP

where TEMP contains one domain and one tuple whose value is

equal to R_i.

Region chain codes are used in the calculation of the perimeter and spatial relationships. It is also used for displaying the contours of regions.

2.6 REGION INTERPRETATION RELATION

The REGION INTERPRETATION relation contains the current set of hypotheses deduced by the system about the regions in the REGION MAP. These deductions are represented by the region interpretations and their associated confidences. During the analysis, each region has a set of interpretations and each interpretation is associated with a confidence level that represents the degree of certainty of the analysis in interpreting this region with that interpretation. Naturally the interpretations can only assume the object names which already exist in the LONG TERM MEMORY. The confidences are represented by numbers between 0 and 1. Thus the relation embodies the current set of hypotheses about the regions in the picture as determined by the system. The REGION INTERPRETATION relation is defined as follows:

REGION INTERPRETATIONS = (REG-NO, FLAG, NO-INTRP, UPDATES,
OBJ-NAM1, CONF-1, OBJ-NAM2, CONF-2,
..., OBJ-NAM8, CONF-8).

where

REG-NO is the region number,

FLAG is set to 0 when the region is in the REGION MAP; otherwise it is set to -1,

NO-INTRP is the current number of interpretations assigned to the region,

UPDATES represents the number of times the region interpretations and their associated confidences have been verified by the HYPOTHESIS VERIFIER/INTERPRETER,

OBJ-NAMi is the ith best interpretation of the region,

CONF-i is the confidence of the ith best interpretation of the region.

The user of the system is able to specify the maximum number of interpretations that will be associated with each region. Hence, the number of domains can be varied according to the user's needs.

As will be described later, the initial region interpretations are generated by the HYPOTHESIS INITIALIZER which transmits its results to the STM in the REGION INTERPRETATION relation. At that time, the domains NO-INTRP, FLAG and UPDATES are set to the appropriate values. During the subsequent analysis, the HYPOTHESIS VERIFIER/INTERPRETER uses this relation to:

i) Obtain the regions' interpretations and their associated confidences.

ii) Transmit the new set of interpretations and

confidences of the region under consideration.

When two regions are merged by the FEATURE ANALYZER, the merged regions are deactivated by setting the FLAG domain to -1.

3.7 OTHER STM RELATIONS

Other relations in the STM are used for archival purposes, such as the MERGE HISTORY relation which maintains a record of which regions have been merged and at which iteration of the analysis. It is defined as follows:

MERGE-HISTORY = (ITER-NO, REG-X, REG-Y)

Obviously, the ITER-NO represents the iteration number at which the merge was executed, REG-Y is merged with REG-X and the new region is arbitrarily labelled REG-X. As shown in Appendix I, region 182 is merged with region 188 after 1861 region verifications.

Another important relation is used to record the order of the region verifications. This relation is called PATH and is defined as follows:

PATH = (ITER-NO, REG-NO, CENT-X, CENT-Y).

The REGION, CENT-X and CENT-Y domains contain the region number, the value of the X-coordinate and Y-coordinate of the centroid of the region respectively. The domain ITER-NO

represents the total number of the region verifications that has been performed by the HYPOTHESIS VERIFIER/INTERPRETER. Thus the order in which regions were examined may be studied and reproduced. For example, figure 6.11.a shows the path of the analysis during the first 100 region verifications.

Finally, there are a number of relations which are used as directories in order to reduce the computation time. In this category there is a relation which stores the addresses of each region in the different STM relations and another relation that stores information about each relation in the STM. These directory relations are only used to reduce the access time to and from the STM.

CHAPTER IV

LONG TERM MEMORY

4.1 INTRODUCTION

In the development of computer vision systems todate, it has usually been the practice to embed the specific scene model within the program. This has made it very difficult to distinguish between the model and the analysis programs. Perhaps it was possible to analyze a picture similar to the class of pictures described by the model, but in general, it was necessary to rewrite a major part of the system in order to be able to analyze a new picture class. There existed no simple way to include a different model, to add, delete, or alter the existing model without editing the program code. In addition, a complete understanding of the system computer code was necessary to account for any of these operations. Another disadvantage of these systems is that experimentation with the model itself proved to be very difficult, in that the independent effects of the model and the analysis could not be segregated and therefore experimented with. Even changing the model parameters and values was not an insignificant task.

In case of the computer vision system described in

this thesis, the model is explicitly represented in the LONG TERM MEMORY. As with the STM, these descriptions are maintained as a set of relations in a relational database. We have attempted to fulfill our design commitment of having a declarative format for the model as much as is practically possible. However, we have also allowed the user of the system to include procedures in the model descriptions, although these were not used to obtain the experimental results in Chapter VI. This is particularly true when dealing with reasonably well defined picture models, such as for example, a FACE or a TABLE. Under these circumstances it is desirable to use pre-defined PLANS as will be described later. These PLANS are stored in the LTM as pointers to procedures stored in a particular process and can be activated when certain situations arise.

The LONG TERM MEMORY is the embodiment of the model representation or the description of the class of scenes to be analyzed. It is completely separate from the system processes which are responsible for the anlysis of the input picture. The information in this world model is stable and does not change at all during the period of analysis. Since the LTM is also implemented as an associative memory using the relational model described in Chapter II, it is simple to alter, thereby providing an attractive and convenient method for experimenting with different models. This

simplicity, in turn, enhances the experimentation capabilities of the system and is absolutely necessary in order to design an appropriate model for a particular task. It also facilitates the use of the system for analyzing different classes of pictures without having to rewrite the computer code of the programs. In the LTM, there are relations that specify object features, their distinguishing features, the complexity predicate and contextual information in terms of model constraints. In addition, there exists a relation, whose full power has not been used in the current experimentation, which contains a set of condition/action rules, and which will be briefly described. In the following sections we will describe these relations and the model information stored in them. Furthermore, we will briefly discuss how the processes of the system use this information during the analysis.

The model to be described is that of an outdoor scene, of which figure 6.1 is a typical. The objects considered in the model are: BUSHES, CAR, CLOUDS, FENCE, GRASS, ROAD, ROOF, SHADOW, SKY, TREE, WALL, WALL-SHADOW and WINDOW. Some objects were introduced to overcome the problems associated with shadows (e.g. WALL-SHADOW and SHADOW) as will be shown later. These might conceivably be removed if a successful shadow analyzer were designed for real outdoor pictures. Moreover, we have assumed only a two dimensional model for

the class of scenes under study. As a result, the type of pictures considered have been referred to as 2 1/2 D, instead of the more complex 3-D representation.

4.2 OBJECT-FEATURES RELATION

The OBJECT-FEATURES relation describes the average value for each object feature as well as the acceptable variation in each of them. The relation is defined as:

OBJECT-FEATURES = (OBJ-NAME, OBJ-FLAG, FEAT-1, FEAT-2, FEAT-3, VAR-1, VAR-2, VAR-3, INTEN, CENT-X, CENT-Y, MIN-X, MIN-Y, MAX-X, MAX-Y, AREA, PERIMTR, CIRCLRTY, ASPECT-R, DFEAT-1, DFEAT-2, DFEAT-3, DVAR-1, DVAR-2, DVAR-3, DINTEN, DCENT-X, DCENT-Y, DMAX-X, DMAX-Y, DAREA, DPERIMTR, DCRCLRTY, DASPCT-R).

Each tuple in the relation contains the average features of one object in the model. The OBJ-NAME domain contains the object name. The domains FEAT-1, FEAT-2, ..., CIRCLRTY refer to the average value of the object's features which are defined analogously to the domains of the REGION-FEATURES relation in the STM (see section 3.3). The domains DFEAT-1, DFEAT-2, ..., DCRCLRTY contain the

acceptable range of variation in the features of the objects. As shown in Appendix II, for example, the average value for the object SKY of domain FEAT-1[1] is 50 and the acceptable range of variation for the feature, DFEAT-1, is 10. Thus, the SKY objects are expected to have FEAT-1 values in the range 40-60. Another example is the value of the CENT-Y domain of the SKY which is set to 128[2], with the domain DCENT-Y set to 35. Because we assume the picture resolution to be 128X128 with 64 grey levels, we are in effect specifying that the SKY is expected to be in the range 93-128 (i.e. SKY normally is in the upper part of the picture). If other picture sizes are used, the model can be scaled up or down accordingly. Generally, relative restrictions of this type could be easily introduced.

The OBJECT-FEATURES relation is a very flexible tool for the model specification. The user is able to make his model as strict as possible by decreasing the acceptable range of variation of the object features. By increasing this range for different objects, he is able to relax the model parameters depending on the class of scenes under analysis. For example, in a controlled environment, as in industrial applications, the acceptable range of variation of the different features can be decreased. On the other hand, in analyzing outdoor scenes, the range of variation of

---

[1] FEAT-1 refers to the RED colour feature.
[2] The position values assume a 128X128 input picture.

different features would be increased.

The average values for the object features and their acceptable range of variation could be obtained interactively during a learning phase of the class of scenes to be analyzed. Using a set of representative pictures (training set) of the class of scenes under study, the operator would encircle the objects of the model and a program would compute the required values and accumulate the results over the pictures in the training set.

In the experiments to be described in Chapter VI, the average colour features were obtained by computing them over different areas of the scene under analysis. One may argue that this is an overly simple and biased approach for learning. However, in the results obtained in Chapter VI, we have not used these feature values in the same sense as in Pattern Recognition for classification. Generally, in Pattern Recognition, the input is classified according to its position in the feature space relative to the defined classes. In natural scene analysis, however, the classes in the feature space are usually overlapping and the concepts used in Pattern Recognition are thereby inapplicable. For example, a blue region can be classified as sky or water surface in a swimming pool, depending on the context surrounding it. The problem is further complicated because

the context itself (e.g. the neighbouring regions) is not known. In the system presented here, the absolute features of the objects can be used to:

i) obtain an initial set of hypotheses about the regions in the initial segmentation, and

ii) constrain the interpretations of the regions during the hypothesis verification.

The second aspect is very important, especially, if the object features are obtained by an organized learning stage. For example, it would be useful in certain applications where the features of the objects are very well defined, such as, in industrial applications or biomedical image processing. The use of these absolute features will generally reduce the total time needed to converge to an appropriate result.

Since no learning was used in this research, we have restricted the use of the absolute values of the features to the generation of an initial set of hypotheses about the regions in the initial segmentation. In addition, the acceptable range of variation in the features of each object provides a large area of overlapping values between the features of the objects. For example, a SKY object has an average[1] value for FEAT-1 equal to 50 and the acceptable range of variation in this feature is 10. Thus the range

---

[1] We are assuming a 64 gray level picture.

over which SKY objects can exist extends from 40 up to 60. On the other hand, a TREE object has a range that extends from 30 to 50. In effect, the value of FEAT-1 for the two objects overlaps in the range of 40-50. Generally, the range of variation of the object's features are large, so that the model can fit a larger number of outdoor scenes. In most cases, the acceptable range of variation was taken to be about 30%-40% (and even 50% in some cases) of the average value of the feature.

It is important to stress that both in the case of interpretation initialization and constraint verification the absolute feature values are not used to classify the regions in the picture. They are only used to filter out the unrealistic interpretations. For example, in the initialization stage, a region at the upper third of the image would only match the CENT-Y feature of SKY, TREE, ROOF, WALL, and WALL-SHADOW. On the other hand, a region in the bottom third of the image would have a CENT-Y feature that lies in the acceptable range of FENCE, GRASS, ROAD, SHADOW and CAR. Thus, by using the vertical position (i.e. the CENT-Y feature), we are able to avoid assigning unrealistic interpretations to the regions.

The OBJECT-FEATURES relation is used in the current implementation to obtain an initial set of interpretations

and associated confidences for the regions in the STM. It is possible to include many descriptions for the same object. That is, we can describe the same object with two sets of features. For example, the TREE is described using two sets of feature values; one describes a dark green TREE and the other describes the yellowish red TREE that appears in the fall. We can add as many descriptions of the object as desired by adding more tuples. Since no two tuples in a relation can have the same values for the key domains, the OBJ-FLAG domain for these descriptions of the same object must be different. For this reason, the two tuples that describe the TREE have flags equal to 1 and 2 respectively (Appendix II).

Although we have only used colour and vertical position features in the experimentation, we have included in the relation all the possible features for the regions. We set the values of the unused features to zero. Obviously it is very simple to add more features to the relation by adding more domains, if the application requires.

4.3 DISTINGUISHING FEATURES RELATION

As mentioned before, in the OBJECT FEATURES relation, each object is described in terms of the average values of

its features as well as the acceptable range of variation of these features. The DISTINGUISHING FEATURES relation defines which features are the object's distinguishing features and a weight is associated with each of these features. Some of the objects' distinguishing features are also designated as having a veto. The HYPOTHESIS INITIALIZER employs this information to generate an initial set of interpretations for each region in the STM. It matches the object's distinguishing features against the region features, and an object interpretation is added to the list of region interpretations if all of the distinguishing features having vetos are matched. It also assigns to each interpretation a confidence which is proportional to the summation of the weights of the successfully matched features. The matching process is described in more details in Chapter V.

The basic function of the DISTINGUISHING FEATURES relation is to specify for each object which features should be used in the matching process. This relation is defined as follows:

DISTINGUISHING FEATURES = (OBJ-NAME, OBJ-FLAG, FEAT-1,
VETO-1, WEIGHT-1, FEAT-2,
VETO-2, WEIGHT-2, ..., FEAT-5,
VETO-5, WEIGHT-5).

where

FEAT-i is the ith distinguishing feature name of the object,

WEIGHT-i is the weight associated with FEAT-i,

VETO-i is the flag which is set to TRUE or FALSE depending on whether FEAT-i has a veto or not,

As shown in Appendix II, each object's distinguishing features are described in a tuple, where the distinguishing feature is defined by the triplet (FEAT,VETO, WEIGHT). An object can be described by more than one set of distinguishing features. This can be done by having more than one tuple in the relation each with a different OBJ-FLAG value.

Any set of features can be used as distinguishing features; however we have found that the most useful features for outdoor scenes are the three colour features (RED, GREEN and BLUE) and the position of the object (CENTROID). The user can experiment with different distinguishing features in the initialization process and with different weights. He need only edit the DISTINGUISHING FEATURES relation to do this.

4.4 PART-OF RELATION

The complexity predicate or PART-OF relation allows us to limit the subsequent search after a particular object has been recognized. It describes the constituents of an object in terms of its parts. The relation consists of two domains which define the object and its superset, and thereby permits the creation of a simple graph structure. It is defined as follows:

PART-OF = (OBJ-X, OBJ-Y)

where OBJ-X is part of OBJ-Y.

The user can use this relation to specify a hierarchical structure such as:

DOOR-KNOB is part of DOOR.

DOOR is part of HOUSE.

Figure 4.1 is an example of the PART-OF relation.

OBJ-X	OBJ-Y
ROOF	HOUSE
WALL	HOUSE
WINDOW	HOUSE

Figure 4.1 Example of PART-OF relation.

We have not made any use of this relation in the work reported here, but we anticipate that it will be extremely useful in the analysis. Knowing what parts constitute the object, PLANS may be invoked to seek a match for regions in the STM to this subset of object parts. Also, when the system recognizes one part with high confidence, it can be used to direct the system to other parts of the object.

4.5 CONSTRAINTS RELATION

One of the most important relations in the LTM is the CONSTRAINTS relation. The work of Yakimovsky and Feldman [126], Waltz [122], Tenenbaum and Barrow [12, 113] and Zucker et al [131] have demonstrated the power of techniques which employ constraint satisfaction. The constraints defined in this relation represent the contextual information that the system uses during the analysis. They are given by rules that describe the world in terms of the conditions under which the objects may be realistically in coexistance. These rules are defined as Boolean AND/OR conditions. Each rule constrains the coexistance of two objects and is stored in one tuple. A rule consists of one or more conditions to be satisfied in order for the two objects to be compatibly adjacent. As shown in Appendix II, each rule consists of a set of domain triplets which are

ANDed together to define one constraint on two objects. The triplet includes the constraint name, a qualifier (operator), and a value. The domains of the relation are defined as:

CONSTRAINTS = (OBJ-X, OBJ-Y, FLAG, CONSTR-1, OPERAT-1,
VALUE-1, CONSTR-2, OPERAT-2, VALUE-2, ...,
CONSTR-7, OPERAT-7, VALUE-7).

Each tuple in the relation represents one rule that constrains the coexistance of OBJ-X and OBJ-Y as adjacent regions. An example of part of the CONSTRAINTS relation used in the analysis is shown in figure 4.2.

The computer vision system presented in this thesis always assumes that the initial segmentation is partial. Because of this assumption, the analysis is always proceeding on the basis of a very fragmented picture. Therefore, no use was made of the objects' geometric shape or other such properties that refer to a complete segmentation. Thus, the type of constraints that are used are restricted to those which are meaningful to segmented parts of objects. Spatial constraints that restrict the relative position of two adjacent objects proved to be extremely useful in eliminating or de-emphasizing certain arrangements of neighbouring regions. For example, consider the constraint that involves ROOF as one object. As shown in figure 4.2, a ROOF segment can be ADJACENT to SKY, TREE,

RELATION CONSTRAINTS (EXAMPLE)

DOMAINS: 1 ---> 12

	OBJ_X	OBJ_Y	NUMBER	CONSTR_1	OPERAT_1	VALUE_1	CONSTR_2	OPERAT_2	VALUE_2	CONSTR_3	OPERAT_3	VALUE_3
1-	ROOF	ROOF	1	FEAT-1	DIFF_LT	8.000000	FEAT-2	DIFF_LT	8.000000	FEAT-3	DIFF_LT	8.000000
2-	ROOF	SKY	1	BELOW	.GT.	0.300000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
3-	ROOF	SKY	2	BELOW	.GT.	0.300000	FEAT-2	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
4-	ROOF	SKY	3	BELOW	.GT.	0.300000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
5-	ROOF	TREE	1	ABOVE	.LE.	0.000000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
6-	ROOF	TREE	2	ABOVE	.LE.	0.000000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
7-	ROOF	TREE	3	ABOVE	.LE.	0.000000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
8-	ROOF	WALL	1	ABOVE	.GT.	0.100000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
9-	ROOF	WALL	2	ABOVE	.GT.	0.100000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
10-	ROOF	WALL	3	ABOVE	.GT.	0.100000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
11-	ROOF	WALL_SH.	1	ABOVE	.GT.	0.100000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
12-	ROOF	WALL_SH.	2	ABOVE	.GT.	0.100000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
13-	ROOF	WALL_SH.	3	ABOVE	.GT.	0.100000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000

RELATION CONSTRAINTS (EXAMPLE)

DOMAINS: 13 ---> 24

	CONSTR_4	OPERAT_4	VALUE_4	CONSTR_5	OPERAT_5	VALUE_5	CONSTR_6	OPERAT_6	VALUE_6	CONSTR_7	OPERAT_7	VALUE_7
1-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
2-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
3-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
4-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
5-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
6-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
7-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
8-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
9-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
10-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
11-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
12-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
13-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000

Figure 4.2 Sample of part of the CONSTRAINTS relation

WALL, WALL-SHADOW (WALL-SH.), or another ROOF segment. It must also be ABOVE the WALL, WALL-SHADOW, BELOW the SKY and not ABOVE the TREE. In addition, it must differ in at least one colour feature (FEAT-1, FEAT-2, or FEAT-3) by a specified threshold in order to be ADJACENT to any of the other objects. However, the difference in each of the three colour features must be below a threshold in order that it be ADJACENT to another ROOF segment. These are expressed in the thirteen tuples shown in figure 4.2. The first specifies that two adjacent regions can be interpreted as ROOF, if and only if the differences between the colour features of the two regions are all less than a threshold (8). The second tuple specifies that two adjacent regions R_i and R_j can be interpreted as ROOF and SKY respectively if:

i) $\text{BELOW}(R_i, R_j) > 0.3$

ii) The difference between the first colour feature of the two regions is greater than a threshold (15).

When we want to specify that objects A and B are not to coexist as adjacent regions in the class of scenes under analysis, no tuples are inserted in the relation CONSTRAINTS with A and B in the domains OBJ-X and OBJ-Y. On the other hand, it is possible to specify that two objects A and B are compatible no matter what the feature values are by inserting a tuple of the form:

A B 1 NULL NULL 0

in the CONSTRAINTS relation.

The CONSTR domain in the relation may take any of the feature names described in sections 3.2 and 3.4 (i.e. REGION CONSTRAINTS and SPATIAL relations). The qualifier (operator) can take any of the following values:

1) LT less than
2) GT greater than
3) LE less than or equal to
4) GE greater than or equal to
5) EQ equal to
6) NE not equal to
7) DIFF-LT difference less than
8) DIFF-GT difference greater than
9) DIFF-LE difference less than or equal to
10) DIFF-GE difference greater than or equal to
11) DIFF-EQ difference equal to
12) DIFF-NE difference not equal to

The VALUE domain contains the threshold value of the condition. Thus, the features of the two regions are compared to the threshold value using the operations defined above to check whether or not the condition is established. The condition is said to be satisfied if all the constraints in the tuple are satisfied.

This relation is a powerful tool to specify contextual rules to direct the analysis. Since constraints can be added, deleted or changed in the relation, the user can incrementally build the model of the class of scenes under analysis. Thus, begining with a small number of intuitively obtained constraints, he can incrementally add the contextual rules that define the model for the class of scenes uder consideration.

The CONSTRAINTS relation is very simple and has a high development potential. The set of features used can be easily expanded to include more features depending on the class of scenes to be analyzed. In addition, the possible qualifiers or operations can be easily expanded to include more complex functions.

4.6 CONDITION-ACTIONS RELATION

The CONDITION-ACTIONS relation is used by the FOCUS OF ATTENTION process to monitor the current state of analysis as represented in the STM in order to recognize the occurrence of any predefined situation. When a situation is recognized, the FOCUS OF ATTENTION executes all of the actions associated with it. The situations and actions are stored in this relation in the form:

$C \text{ -----> } \alpha$

where C represents a set of conditions C_1, C_2, ..., C_n that must all be satisfied and α represents a set of actions α_1, α_2, ..., α_m that are associated with the situation defined by C. The actions are stored in the form of pointers (names) to procedures which make up PLANS [65, 66, 106] that can be activated by the system. These rules are responsible for directing the system's attention to a specific part of the image. This is done either by:

i) altering the status of the STM so that the sytem's focus of attenticn is directed to a specific part of the image, or

ii) updating the information in the STM to reflect the fact that certain situations have been detected.

As will be described in detail in Chapter V, the attention of the system is focused on the regions at the top of the REGION STACK, which is part of the STM. The HYPOTHESIS VERIFIER/INTERPRETER usually examines the first region on top of the STM REGION STACK. Therefore, it is possible to direct the FOCUS OF ATTENTION of the computation by manipulating the order cf regions in the REGICN STACK.

Although the more important aspects of the FOCUS OF ATTENTION process and the CONDITION-ACTIONS relation have not been implemented, it is interesting to consider them in order to conceive the importance of this approach. This

process is intended to monitor the current status of analysis as is manifested in the STM. When a pre-specified situation is detected, all the actions associated with it are triggered. The situation/action rules are generally model dependent.

At present the user can only express situations that involve the interpretations and confidences of a group of regions. The CONDITION-ACTIONS relation is defined as follows:

CONDITION-ACTIONS = (FLAG, OBJ-NAM1, CONF-1, ACTION-1, ACTION-2, ACTION-3).

where FLAG is an integer quantifier, OBJ-NAM1 is an interpretation of a region and CONF-1 is the confidence associated with that interpretation. For example, we may have these tuples in the relation:

1- SKY	0.90	...	α_1	NONE	NONE
2- EYE	0.95	...	α_2	α_3	NONE
3- TABLETOP	0.85	...	α_4	NONE	NONE

The first specifies that if a region R_i has an interperetation SKY with confidence greater than 90% the FOCUS OF ATTENTION triggers action α_1. Action α_1 may be a heuristic procedure to search the STM updating the interpretations of all regions ABOVE or that CONTAINS region R_i to be either SKY or CLOUDS. The system considers such rules one by one in increasing order of the FLAG domain of

the relation. Thus if the following rule is added to the relation:

FLAG ANY 0 ... α_o

where ANY indicates any object interpretation and the value of the FLAG is the largest of all the tuples, the FOCUS OF ATTENTION will always execute action α_o when none of the situations with lower FLAG values is recognized.

Currently, we have employed one tuple in this relation which is defined as:

1- ANY 0 ... α_o

where α_o is the intrinsic action executed whenever the FOCUS OF ATTENTION is activated. This intrinsic action implements the following:

1- Move region R_i which is at the top of the REGION STACK to the bottom.

2- Bring all the regions ADJACENT to region R_i and place them in decreasing order of confidence at the top of the REGION STACK. In effect, the FOCUS OF ATTENTION will direct the system to examine the least confident region in the neighbourhood of the current tested region R_i as will be described later in Chapter V.

Having described the LTM relations and the model representation of the system, we will present in the next chapter the different processes of the system and the flow

of control during the analysis. We will also describe in detail how the model information is used by the different processes.

CHAPTER V

SYSTEM PROCESSES AND CONTROL

5.1 INTRODUCTION

During the design and implementation of the computer vision system in this thesis, we have identified and maintained certain objectives and guidlines that have affected to a large extent the final system structure and flow of control. In the following discussion, we will present the system and briefly review these objectives and guidelines. As shown in figure 1.2, the system consists of a number of independent processes that communicate and interact dynamically to achieve their processing tasks. The system processes are allowed to communicate only through the STM, which contains the input data as well as the deductions made by the system about the scene under consideration. Hence, all direct communications or interactions between the processes are prohibited. Each process obtains its input data from the STM and transmits its conclusion to it. In this way, the STM enforces the modularity and structural flexibility of the system. It also ensures the independence of each process because each one need only be aware of the STM data model and the communication protocols. We should

stress the fact that a particular process is neither dependent on the data structures of the others nor their processing strategies. Therefore changes in any of the system's processes are completely invisible to the others. Adding or removing processes will not affect their operation as long as the information needed by them is maintained in the STM. It is in this way, that the expandability of the system is facilitated. The STM enables the designer of each process to install and test various versions of his design without affecting the operation of other processes. He must only ensure that the conclusions are delivered to the STM using a specific communication protocol.

As described in Chapter IV, the visual model representation of the class of scenes under analysis (i.e. information about the objects that may exist in the world and their distinguishing features), and the rules that constrain the different spatial co-occurrences of the objects are not embedded in the specific process that uses them. They are maintained separately in the LTM, thereby enhancing the experimentation capabilities of the system. First, the model is completely visible and accessible to the user. He can easily alter it and then study the resulting effects. Second, the system is not restricted to be used in conjunction with a specific world model but can be used with other models without changing or rewriting the computer code

for the different processes. Naturally it is necessary to enter the new model.

Both the STM and LTM are designed as relational databases. The relational database has a very simple data model, the table format, and its use enforces a standard communication protocol in the form of a relational database sublanguage. The concept of competition and cooperation [2, 12, 113, 121, 131, and 133] has been adopted as a basic paradigm for the system interpretation strategy. In addition, the system is data driven, in the sense that the flow of control and the different process activities depend on and are directed by the current state of the data in the STM.

In the next few sections, we shall introduce each of the system processes that has been implemented todate. First, in section 5.2, we discuss the LOW LEVEL PROCESSOR which segments the input colour picture into contrasting regions containing pixels whose features are similar. This process can be looked upon as a means of reducing the number of individual entities to be analyzed at the higher levels, thereby facilitating data handling and storage. Typically, it may result in a compression in the number of entities from the 128X128 pixels in the input (RED, GREEN and BLUE intensity planes) to the order of 200 - 400 regions.

Second, section 5.3 discusses the FEATURE ANALYZER which computes a set of attributes for the segmented input picture and then transmits the result to the STM, to be available to the other system processes. When regions are merged during the iterative computations the FEATURE ANALYZER also computes the new set of attributes for the newly formed regions and updates the STM relations to reflect this.

Third, in section 5.4, we detail the HYPOTHESIS INITIALIZER which takes as its input the region descriptions as they appear in the STM, and uses the model information in the LTM to generate the initial possible interpretations for the regions in the STM. It also associates each interpretation with a number that represents the confidence level in hypothizing that interpretation. The result is transmitted to the STM for later use by the other processes.

Fourth, in section 5.5, we discuss the HYPOTHESIS VERIFIER/INTERPRETER, a process which is intimately connected with the underlying iterative computation implemented by the system. In addition to the model information in the LTM, this process uses the STM information in the form of the region features, the structural relationships between the different regions, and the current set of hypotheses about each region and their

associated confidences. Its basic function is to verify and update the confidence of each interpretation. This is achieved by using the basic paradigm of competition and cooperation which is implemented as a relaxation labelling process [131, 133].

The fifth process, discussed in section 5.6, is the FOCUS OF ATTENTION which is intended to work in a cycle consisting of situation recognition followed by an action or a set of actions. This process controls the order of the analysis of the regions by bringing the appropriate regions to the top of the STM. The system's attention is focused on this top region thereby controlling the sequence of the analysis. Using the situation/action rules stored in the LTM, we are able to specify which part of the image is to be examined next.

In section 5.7, the last process, a sort of super process, called the SCHEDULER is discussed. It is responsible for deciding which process should be activated at any given time. We note that the design considerations require that the system modules be independent and that they interact only by transmitting their results to the STM. No direct transfer of results is allowed between the different processes. In addition, processes are not permitted to activate other processes directly. Both these properties

enforce the desirable level of independence.

5.2 LOW LEVEL PROCESSOR

The goal of a computer vision system can be simply stated as that of developing a machine which can correctly interpret real world pictures. Two aspects come to the fore immediately: first, real pictures imply sensory data represented digitally by a two dimensional array of numbers, and second, interpretations involve a delineation of objects and thence a symbolic description of the scene. While it is possible to represent the information in a scene accurately by a digitized grey-scale array, this representation alone does not put emphasis on the interesting properties of the scene. It would be more convenient to have a description in terms of the natural elements of the picture such as regions, boundaries and lines. The use of the latter description will also reduce the number of individual entities dealt with by the system from the nxn pixels in the original picture to a much smaller subset of entities. In the current implementation, the colour picture is segmented into regions containing pixels whose primary features are similar.

Many algorithms were developed in the last decade for

picture segmentation [16, 39, 48, 75, 127]. In [136] Zucker presents a detailed survey of most of these techniques, too numerous to detail here. We have proposed a new algorithm that uses an adaptive thresholding technique for picture segmentaticn. Perhaps the closest in spirit to our work is the region finding method of Muerle and Allen [75] and Gupta and Wintz [39]. Simply stated, each pixel in the picture attempts to attract as many neighbours as possible on the basis of their cclour differences. It accomplishes this by testing each pixel in the image and attempting to merge it with any of its adjacent four-connected regions[1]. If certain conditions are satisfied it is merged, otherwise it is labelled as a new region which will in turn try to group as many as possible of its neighbouring pixels.

A picture is first stored in the computer as an array P on an nxn grid G^2. The array P is a function on G whose value at each point is the grey scale value of that point. Each pair (s,t) is called a picture element or pixel, whose grey scale is the value P(s,t) and whose coordinates are s and t. In what follows we will often refer to neighbouring

---

[1] In fact, since the array is scanned once in a regular sequential fashion only two pixels are eligible for merging at any given time.

[2] For a colour picture, the array P is replaced by three arrays that represent the RED, GREEN and BLUE grey scale value at each point on G. We will refer to these arrays as P_q where q = 1, 2, 3 for the RED, GREEN and BLUE arrays respectively.

pixels of a pixel (s,t) by which we mean any of the four points that are nondiagonally adjacent to pixel (s,t), (four-connected neighbours). The neighbouring regions of a pixel (s,t) are the regions that contain any of the four adjacent pixels to (s,t). Two points P1 and P2, belonging to a subset R of the picture, are <u>connected</u> with respect to the set R if there exists a sequence of points from R, the first of which is P1 and the last of which is P2, and such that consecutive pairs are neighbours. Thus, a <u>region</u> can be defined as a set $R \subseteq G$ in which any pair of points are connected with respect to R. These regions are the basic elements of the picture representation. For every region R in the picture we define the following:

$$F_{R_i}(q) = \frac{1}{N} \sum_{(s,t) \in R_i} P_q(s,t) \qquad (5.1)$$

$$\sigma_{R_i}(q) = \frac{1}{N} \left[\sum_{(s,t) \in R_i} [P_q(s,t) - F_{R_i}(q)]^2 \right]^{\frac{1}{2}} \qquad (5.2)$$

where

R_i is a region that contains N pixels,

$P(s,t)$ is the value of the feature at pixel (s,t), and

$q = 1, 2,$ or 3.

Then F_{R_i} is the average value of the feature over the region R_i and σ_{R_i} is the standard deviation of the feature over the region R_i.

The algorithm can be described by the following steps:

1- Initially all pixels in the picture assume the null label. That is, they are not labelled as belonging to any region.

2- Starting from the top left corner, we scan the picture and at each pixel (s^*,t^*) we apply the remaining steps 3, 4, 5, and 6.

3- a) Calculate F'_{R_i} and σ'_{R_i} for every region R adjacent to pixel (s^*,t^*) as follows:

$$F'_{R_i}(q) = \frac{1}{N+1}\left[P_q(s^*,t^*) + N \cdot F_{R_i}(q)\right] \tag{5.3}$$

$$\sigma'_{R_i}(q) = \frac{1}{N+1}\left[\left[P_q(s^*,t^*)\right)-F'_{R_i}(q)\right]^2+\sum_{(s,t)\in R_i}\left[P_q(s,t)-F_{R_i}(q)\right]^2\Big]^{\frac{1}{2}} \tag{5.4}$$

F'_{R_i} is the average value of the feature over region R_i including pixel (s^*,t^*) and σ'_{R_i} is the standard deviation of the feature over the region R_i including pixel (s^*,t^*).

b) Calculate the threshold value for that region as:

$$h_{R_i}(q) = \left[1 - \frac{\sigma'_{R_i}(q)}{F'_{R_i}(q)}\right] h \qquad (5.5)$$

where h is a constant threshold[1] value chosen by the user. Thus h_{R_i} represents an adaptive threshold value for the region R_i.

c) Calculate[2] for each region R_i the difference value

$$\Delta F_{R_i}(q) = \left| P_q(s^*,t^*) - F_{R_i}(q) \right| \qquad (5.6)$$

4- If $\Delta F_{R_i}(q) < h_{R_i}(q)$ for q=1, 2, and 3, region R_i is considered as a possible candidate for labelling pixel (s^*,t^*).

5- If there exists more than one candidate, we label pixel (s*,t*) with the region label that is the minimum distance $\sum_q [\Delta F_{R_i}(q)]^2$ to it.

6- If none of the adjacent regions satisfies condition (4), or if the pixel has no adjacent region[3], pixel (s^*,t^*) is labelled as a new region.

---

[1] For the segmented picture in the experiments, h was set to 10.

[2] When using three features, the condition must be satisfied for each feature; e.g. RED, GREEN and BLUE.

[3] The pixel at the top left hand corner does not have any adjacent labelled regions.

Summarizing, the process starts at the pixel on the top left-hand corner and marks it as region number 1 since it has no adjacent labelled regions. Then it scans the picture row by row from left to right, attempting to merge each pixel with any of its adjacent regions (if any). If it cannot merge the pixel with any of its adjacent regions, it is marked as a new region and assigned a new region label. As previously mentioned, merging depends on the threshold values which are based on the standard deviation and the average value of the feature over the region. It can be seen from figure 5.1, which is a plot of equation 5.5, that the value of the threshcld depends on the uniformity of the region, because as the standard deviation of the region approaches zero, the value of the adaptive threshold ($h_{R_i}(q)$) approaches the constant h specified by the user. As the region becomes less uniform, the ratio $\sigma'_{R_i}(q) / F'_{R_i}(q)$ grows, which in turn reduces the value of $h_{R_i}(q)$. Thus the use of equation 5.5 limits the growth of nonuniform regions. As shown in the results (see figure 6.2), the SKY and FENCE regions were each merged into one large region because of their relative uniformity. On the other hand, the TREE region, which lies to the left of the house, was segmented into a large number of small regions. Between these two

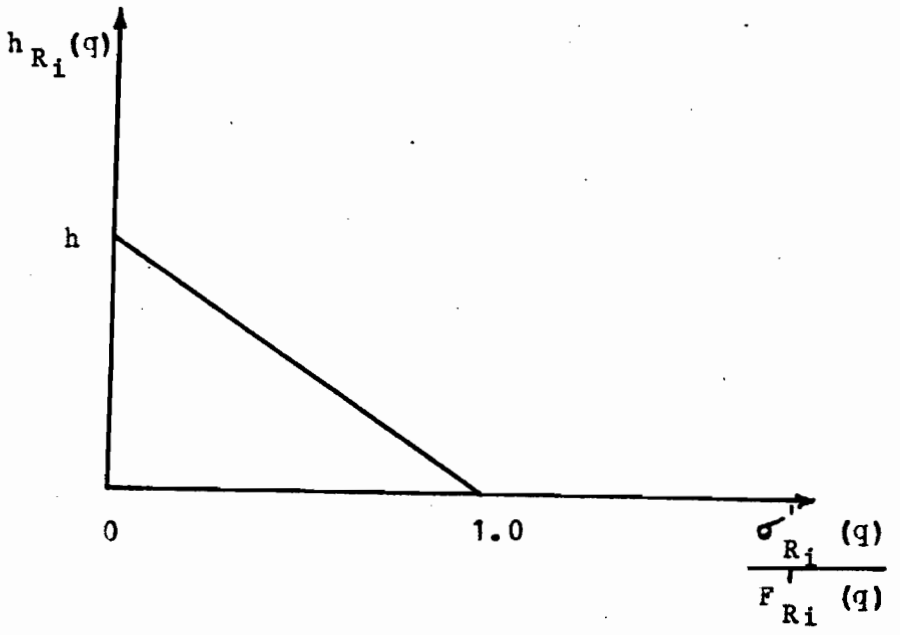


Figure 5.1 Adaptive threshold versus the ratio of standard deviation and average value of feature q over region R_i

extremes is the ROOF of the house, which was segmented into five regions, as it is not as uniform as the SKY, nor as nonuniform as the TREE or WALL regions.

The algorithm described above is simple and effective. It does not require any knowledge of the scene under consideration[1]. It is also completely automatic, in the sense that no user interaction is needed to direct the processing, and does not contain complicated parameters to control its operation. Moreover, it is very fast. For a 128X128 picture, it takes about 30 seconds on a DEC VAX11/780 computer and about 5 minutes on a DEC PDP-15 computer. It is obviously not suitable for segmenting textured areas. However, it was very successful with the GRASS and ROOF areas, of the tested picture, as they are weakly textured.

The algorithm is followed by a clean-up operation to remove regions with very small areas. The user specifies the minimum region area allowed in the final segmentation. Then the pixels of the regions whose areas are less than that minimum area are labelled with the null label. Next, the REGION MAP is scanned and when a pixel with a null label is found, it is merged with any of its adjacent regions, depending on the distance $\sum_q \| \Delta F_{R_i}(q) \|^2$. That is,

---

[1] Zucker [136] has referred to this as "general purpose segmentation".

a pixel marked with the null label is merged to the adjacent region that has the minimum distance $\sum_{q} \| \Delta F_{R_i}(q) \|^2$. However, if a pixel (s^*,t^*) does not have any adjacent regions (i.e. its four neighbouring pixels are labelled with a null label), it would not be labelled in that scan. This whole process is repeated until all pixels in the picture are labelled with an appropriate region label. Finally, this REGION MAP is stored in the STM to be used by the other processes. The results in figure 6.2 used a clean-up operation with minimum area equal to 8 pixels. In this case, six scans were required to label the null labelled pixels.

5.3 FEATURE ANALYZER

The role of the FEATURE ANALYZER is to compute the features stored in the STM. Thus it is responsible for the computations associated with the following relations: REGION-FEATURES, LEFT-OF, RIGHT-OF, ABOVE, BELOW, ADJACENT, CONTAINS, ENCLOSED-BY, and CHAIN-CODE. These features have been defined in Chapter III. The FEATURE ANALYZER uses the REGION MAP and the colour input picture to calculate the following features:

(i) The average features of the region over the three colour planes and the associated standard deviation of these features.

(ii) The position of the region in the pictures in terms of the X and Y coordinates of the centroid of the region.

(iii) The minimum rectangle that contains the region.

(iv) Region area and perimeter.

(v) Region circularity and aspect ratio which are rough measures of shape.

(vi) The structural relationships between the region and its adjacent neighbouring regions such as: LEFT-OF, RIGHT-OF, ABOVE, BELOW, ADJACENT, CONTAINS, and ENCLOSED-BY.

(vii) Region boundary in the form of the chain code.

This process is activated whenever a new region is defined. Obviously it is invoked after the low level segmentation to calculate the features of all the regions in the REGION MAP. It is again called upon when the system decides to merge regions during the analysis. At this time, it is activated to update the STM relations to, for example, alter the REGION MAP by changing the labels of the merged regions to the new region labels. Next, it calculates the new region features as defined in Chapter III. Then it removes all of the information about the merged regions from

the STM relations[1]. The MERGE HISTORY relation is also updated at this time to provide a dynamic history of the functioning of the system. Thus this process is activated when the current description (STM) must be updated because of decisions made by other processes. This activation is initiated by the SCHEDULER which will be described in section 5.7.

It should be noticed that each of the features defined is calculated using a procedure. To add new features, the user need only specify the procedure to be used and define where the features are to be stored in the STM. Therefore, the process is easily expandable to any number of features and is not restricted to the set of features defined here. For example, it can be easily expanded to include the analysis and measurement of edges, boundaries and lines [78].

5.4 HYPOTHESIS INITIALIZER

The function of the HYPOTEHESIS INITIALIZER is to generate a set of hypotheses about each entity in the STM. These entities are restricted to being regions, however the

---

[1] It was not possible to retain all of this information about the merged regions because of the hardware constraints imposed by the PDP-15. However, this could be done with regions flagged after being merged.

same process can be easily extended to include any visual entities such as lines, boundaries, edges or surfaces. For each region in the STM, the HYPOTHESIS INITIALIZER generates a set of possible interpretations. Each of these interpretations is assigned a number that represents the confidence of this process in such an interpretation. In generating these interpretations and their associated confidences, the process uses model information in the LTM stored in the OBJECT FEATURES and the DISTINGUISHING FEATURES relations. In the OBJECT FEATURES relation, each object is described in terms of the average value of its features as well as the acceptable range of variation in these features (see Chapter IV). The DISTINGUISHING FEATURES relation defines which features are the object's distinguishing features and the weight associated with these features. Some of the objects' distinguishing features are designated as having a veto. Any set of features can be used as a distinguishing set of features; however we have found that the most useful features are the three colour features (RED, GREEN and BLUE) and vertical position of the region (CENTROID).

Simply stated, the HYPOTHESIS INITIALIZER attempts to match each region in the STM to all of the objects described in the LTM model. It matches the object's distinguishing features against the region features, and an object

interpretation is added to the region's interpretation list if all of the distinguishing features having vetos are matched. It also assigns to each interpretation a confidence which is proportional to the summation of the weights of the successfully matched features.

Let $F_{R_i}(q)$ be the q-th feature of the region R_i (in the STM) which is to be matched, $F_{I_x}(q)$ be the average of the q-th feature of object I_x, and $d_{I_x}(q)$ be the acceptable variation in the q-th feature of the object I_x (in the LTM). Suppose there is available for each object interpretation I_x in the LTM a set of K distinguishing features, F_{I_x}, where $W_{I_x}(k)$ is the weight of the k-th distinguishing feature, and $V_{I_x}(k)$ is a flag indicating whether or not a feature k has a veto. The basic matching process for the hypothesis initialization can be expressed as follows:

(i) For each of the distinguishing features of the object I_x, set M(k)=1 or 0 depending whether or not there is a match. A match exists for the q-th feature if:

$$| F_{I_x}(q) - F_{R_i}(q) | \leq d_{I_x}(q) \qquad (5.7)$$

(ii) The unnormalized confidence $C^*_{R_i}(I_x)$ is defined as:

$$C^{*}_{R_i}(I_x) = \frac{\sum_{k \in K} M(k)\ W_{I_x}(k)}{\sum_{k \in K} W_{I_x}(k)} \qquad (5.8)$$

(iii) If a distinguishing feature with a veto is not matched according to equation 5.7, then set $C^{*}_{R_i}(I_x)=0$ for that interpretation.

This process is repeated for each object entry in the LTM, thus generating a set of hypotheses about the region R_i as objects $\bar{I}$; that is, I_1, I_2, ... I_n with confidences C_1^*, C_2^*, ... C_n^*. The latter is then normalized by the equation:

$$C^{o}_{R_i}(I_x) = \frac{C^{*}_{R_i}(I_x)}{\sum_{I_y \in \bar{I}} C^{*}_{R_i}(I_y)} \qquad (5.9)$$

to give the initial confidences for the interpretations attatched to region R_i. In case none of the LTM objects matches (i.e. all $C^{*}_{R_i}(I_x)=0$), for a particular region R_i, all the confidences $C^{*}_{R_i}(I_x)$ are assumed to be equal. In effect, the process hypothesizes that the region can be interpreted as any object in the LTM with equal confidence. The list of interpretations $\bar{I}$ for each region R_i and the associated confidences $C^{o}_{R_i}(I_x)$ are then transmitted to the STM and stored in the INTERPRETATION relation.

The HYPOTHESIS INITIALIZER functions on the basis of

an exhaustive comparison of each region in the STM with all of the objects in the LTM. This is a practical approach, even if the number of objects in the LTM is large, since the use of the veto in the DISTINGUISHING FEATURES relation totally eliminates unsuitable initial interpretations.

5.5 COMPETITION AND COOPERATION

In this section we will describe how competition and cooperation is used as a basic approach in the computer vision system. Todate the primary vehicle employed to achieve this goal in the field of computer vision has been the relaxation labelling process [129]. In the following section we will briefly review the use of the concept of competition and cooperation in the form of a relaxation labelling process in computer vision. A more detailed review and discussion of this concept may be found in Mackworth [70].

5.5.1 HISTORICAL BACKGROUND

Descrete relaxation was used very successfully by Waltz [122] in the highly constrained environment of line drawings of polyhedra. He used and extended the vertex

labelling scheme of Huffman [49] and Clowes [18] to include and utilize shadows in the process of interpreting line drawings of scenes that consist of convex trihedral objects. Each vertex is labelled with all of the possible interpretations. Lines are also assigned labels which cover most of the possible edge interpretations in a 3-D (three dimensional) scene and depend on the lighting conditions at the line interfaces. A filtering algorithm that implements descrete relaxation is then applied to remove any of these labels which are impossible, given the current context of the junction (vertex) under consideration. The effect of removing labels from a specific junction propagates through the neighbouring junctions restricting their possible set of labels. The process continues until no more changes in the labels can be made and quickly converges to either a unique interpretation or a small number of scene interpretations. Thus the problem can be described as a network which consists of a labelled directed graph in which the variables are represented by the nodes (junctions) each with an associated set of possible labels attached to the node. The branches connecting the vertices represent the arcs of the network. The task is to allow the labels of the different nodes to interact in order to produce a unique label for each node in the network that is compatible with its neighbouring nodes. The neighbouring nodes of a node X are defined as those nodes that are connected directly to it by

arcs of the network. In descrete relaxation, if a junction label is incompatible with any of its neighbours, the label is removed from the set of labels of this junction. The process iterates, and label removal propagates through the network until no more labels can be removed.

Tenenbaum and Barrow [113] have adopted Waltz's filtering algorithm (descrete relaxation) for segmenting and interpreting real pictures in the IGS (INTERPRETATION GUIDED SEGMENTATION) system. They start with an initial partition of the image that contains elementary regions with identical colour attributes. Each region is assigned an initial set of interpretations. The process proceeds by eliminating those locally possible interpretations that are not consistent with any possible interpretation of an adjacent region. After a region interpretation is eliminated, the interpretations of all the adjacent regions are reexamined to determine whether they are still compatible. Deductions may thus propagate as in the Waltz program.

A series of experiments were designed to test the IGS system. In all of these, human intervention was necessary in order to have the filtering algorithm produce a correct, non-trivial result. In the first experiment, a human operator is asked to assign the correct interpretation for all the regions in the initial segmentation that have an

area greater than 6 pixels. Also, whenever a region grows such that its area becomes greater than 6 pixels, the user is again asked to assign to it the correct interpretation. In another experiment, the user is allowed to outline some areas in the picture and even point to isolated pixels, and designate the correct interpretation for them. The algorithm is then applied in order to assign the correct interpretation to the other parts of the picture.

A third experiment was implemented to segment and interpret an image of an air compressor. A structural model of the compressor developed by Agin [5], together with a model for the relative positions of the camera, assign interpretations to the different pixels in the image. Most pixels start with the correct interpretation. Only the pixels that lie in the overlapping areas of the uncertain parts of the compressor are assigned more than one interpretation. The filtering algorithm is then activated to examine the pixels that have more than one interpretation.

The final experiment was performed on an indoor scene. Human intervention was used in order to have the filtering algorithm produce a correct non-trivial result. This was done by isolating certain regions in the picture and giving them an artificial ("universal" in [113]) interpretation.

In addition, in order to force the process to produce non-trivial results (that is, all the regions are interpreted as one object), it was necessary to label at least one pixel in the image with the correct interpretation, in this case, a pixel in the baseboard. During the filtering algorithm, the elimination of the erroneous interpretations of the different regions propagates from the baseboard pixel to the rest of the image.

Most of the difficulties in the IGS arose from the fact that descrete relaxation process in the form of Waltz's filtering algorithm was applied to a domain which was not highly constrained. Faced with these problems, Tenenbaum and Barrow later proposed MSYS [12] which uses continuous relaxation to interpret pictures, given as an input a complete segmentation provided by a human operator. They used simulated range information to restrict the initial set of interpretations. In this case, the relaxation process was not defined in the usual manner to be a process uniformly applied to all entities in the picture [131]. Instead, a set of heuristically designed likelihood functions were tailored to the set of interpretations they used. Each likelihood function is heuristically built and consists of a product of terms. Each term raises or lowers the likelihood of an interpretation, depending on the type

of constraints and the likelihood that the regions satisfy the constraints. First an estimate is obtained of a consistent likelihood limit for each locally possible region interpretation based on:

(i) apriori likelihood of that interpretation, and
(ii) likelihood limits estimated for other semantically constrained interpretations.

Whenever any region interpretation is changed, the likelihoods of the related regions are reevaluated. This process is repeated until there is no change in the likelihoods. The result is a set of regions that has unique interpretations and another set with more than one interpretation. This necessitated the use of the so-called M* algorithm in order to find the unique interpretation of a region which had more than one interpretation, a situation which may occur.

The MSYS system suffers from several difficulties. First, it assumes a complete segmentation. Second, the likelihood functions do not interact uniformly, because each one is designed heuristically and independently of the other functions. This in turn makes it difficult to introduce an additional heuristic function to the system because its effect on the original system is not known. Tenenbaum and

Barrow describe the results of the system as inconclusive because of the simplicity of the test scene and the reliance of the system on the simulated range data for the intitial set of interpretations [12].

Zucker et al [131] have described and applied a relaxation labelling process (RLP) to the detection of smooth lines and curves. They have used knowledge in the form of constraints about line and curve continuity to resolve local ambiguities. Nine labels are associated with each image point, eight labels indicating line segments at various orientations and one indicating a no-line case. Attached to each label λ is a quantity $P(\lambda)$ which is described as the probability that λ is the correct label for that image point. The relaxation process allows the labels and their probabilities to interact such that line segments in compatible orientations strengthen one another, and incompatible segments weaken one another. The no-line labels are reinforced by no-line neighbours and weakened by line oriented labels. Zucker et al [132] have studied the convergence, speed and stability of the RLP. In [133], Zucker et al analyzed the conditions under which the relaxation process is equivalent to local maxima selection.

Riseman and Hanson [43] have used the relaxation labelling process proposed by Zucker et al [131] in the

low level processing stage of their computer vision system. First, they applied the Kirsch operator to find edge points. These edge points are then aggregated into boundaries using a relaxation labelling process. They also used the relaxation labelling process together with global histograms to classify the pixels of the input picture into clusters. Here, they incorporated the spatial constraints of pixels together with compatability functions defined on these clusters to form regions. From the point of view of a partial segmentation, their results look promising.

Finally, Arbib [2] has reviewed several neural network models (Diday [29], Kilmer et al [58], and Dev [28) which can be analyzed within the common framework of competition and cooperation. He then relates the relaxation labelling process as described in Zucker et al [131] to the paradigm of competition and cooperation in brain theory.

In the following section, we will describe how cooperative computation is used in the segmentation and interpretation of real world scenes.

5.5.2 HYPOTHESIS VERIFIER/INTERPRETER

The HYPOTHESIS VERIFIER/INTERPRETER adopts the concept of competition and cooperation as a basic paradigm for its vision strategy. It uses the competition between the possible hypotheses for a particular region and the competition and cooperation between the immediate neighbours of the region in order to achieve a globally compatible interpretation for all of the regions in the picture. The function of the HYPOTHESIS VERIFIER/INTERPRETER is to test different hypotheses about a specific region in the STM and then to update the confidence associated with each of these hypotheses. This process is based on the continuous relaxation labelling process as defined in Zucker et al [131]. It is the first implemented continuous relaxation process to be applied to high level processing of a partially segmented image. It is designed as an experimental tool in order to be able to test various approaches and definitions of the relaxation labelling process. At present the user is able to choose any one of a set of updating rules defined in Zucker [133].

As described in section 5.2, the input image is broken-up into regions whose primary features (i.e. colour features) are similar. These regions are not necessarily objects, and in extreme cases may just be single pixels.

The HYPOTHESIS INITIALIZER generates a set of initial hypotheses about each region. It also assigns to each interpretation a confidence level. Let I_x represent an interpretation such that $I_x \in \bar{I}$, where $\bar{I}$ is the set of all pertinent object interpretations in the LTM. Then we may define $C_i^k(I_x)$ as the confidence that region R_i has an interpretation I_x at iteration k. Only a subset $\bar{I} \subseteq I$ of all possible interpretations in the LTM are applicable to a particular region in the STM, which at any given time contains M regions. The confidence of an interpretation $C_i^k(I_x)$ satisfies the following constraints:

$$0 \leq C_i^k(I_x) \leq 1.0 \quad \text{for all } I_x \in \bar{I} \subseteq I \text{ and for all } i \in M \tag{5.10}$$

$$\sum_{I_x \in I} C_i^k(I_x) = 1.0 \quad \text{for all } i \in M \tag{5.11}$$

Let region R_i be the region under consideration and A be the set of all the immediately adjacent regions to region R_i. Then $R_j \in A$ is the j-th adjacent region to the current region under examination. The adjacency D_{ij} of two regions R_i and R_j is defined as the ratio of the length of the common perimeter between the two regions to the total perimeter of the region R_i. We should point out here that the adjacencies satisfy the following constraints:

$$0.0 \leq D_{ij} \leq 1.0 \quad \text{for all } i, j \in M \tag{5.12}$$

$$\sum_{j \in A} D_{ij} = 1.0 \quad \text{for all } i \in M \tag{5.13}$$

The process examines each interpretation associated with the region R_i under study attempting to decide whether or not it is compatible with the different interpretations of the adjacent regions $R_j \in A$. Thus we define $\delta_{ij}(I_x, I_y)$ as the compatability of regions R_i and R_j when interpreted as objects I_x and I_y, respectively. The process attempts to determine whether R_i and R_j can be interpreted as I_x and I_y by logically testing all of the possible constraints in the LTM that involve I_x and I_y as objects. For example, if I_x and I_y are SKY and ROOF, it examines the constraints in the LTM that involve SKY and ROOF against the features and structural relationships of the two regions R_i and R_j. Thus,

$$\delta_{ij}(I_x, I_y) = \begin{cases} +1 & \text{if } R_i \text{ and } R_j \text{ are compatible as objects } I_x \text{ and } I_y \\ -1 & \text{if } R_i \text{ and } R_j \text{ are not compatible as object } I_x \text{ and } I_y \end{cases} \tag{5.14}$$

The incremental change in the confidence for each interpretation of the region R_i can be calculated as follows:

$$\Delta c_i^k(I_x) = \sum_{j \in A} D_{ij} \sum_{I_y} (I_x, I_y)\, c^k_j(I_y) \qquad (5.15)$$

The new updated confidence is then given by:

$$c_i^{k+1}(I_x) = \frac{c_i^k(I_x)\,[1 + \Delta c_i^k(I_x)]}{\sum_{I_x \in \bar{I}} c_i^k(I_x)\,[1 + \Delta c_i^k(I_x)]} \qquad (5.16)$$

The denominator is so chosen in order to ensure that :

$$\sum_{I_x \in \bar{I}} c_i^{k+1}(I_x) = 1. \qquad (5.17)$$

The process described above is applied iteratively to each interpretation. In each iteration, the confidence level of the interpretation increases or decreases depending on the context in its immediate neighbourhood. The effect of the changes in the confidence level of the interpretations of different regions are thus propagated through the regions immediately in their neighbourhood. By means of this iterative process, the confidence levels converge to a fixed point as defined in Zucker [135].

The updating equations 5.15 and 5.16 can be interpreted as allowing each adjacent region interpretation I_y to vote for the interpretation I_x of region R_i. Each adjacent region R_j is given a voting power proportional to its adjacency (D_{ij}) to region R_i and thus a region that has a long common boundary with region R_i will cast a stronger

vote. Obviously it is possible to introduce other bases for weighting the vote of the adjacent regions, such as relative area, for example. Also, each interpretation I_y of neighbouring region R_j is given a vote proportional to the current confidence in this interpretation $C_j^k(I_y)$.

To illustrate the process, let region R1 be the region under examination by the HYPOTHESIS VERIFIER/INTERPRETER. As shown in figure 5.2.a, region R1 is adjacent to regions R2, R3 and R4. Each region has a set of interpretations and each interpretation is associated with a number that represents the confidence level of the system in that interpretation. For each interpretation of the region R1 under examination calculate the compatability $\delta_{ij}(I_x, I_y)$ of this interpretation with all of the interpretations of the adjacent regions R2, R3 and R4. For example δ_{12}(SKY,ROOF) is calculated as follows:

1- Find the sets of constraints which involve SKY and ROOF in the LTM CONSTRAINTS relation by using the following relaticnal algebraic expression:

CONSTRAINTS :(OBJ-X = SKY & OBJ-Y = ROOF).

The result is stored temporarily in a relation as shown in figure 5.2. Generally, each tuple in the relation

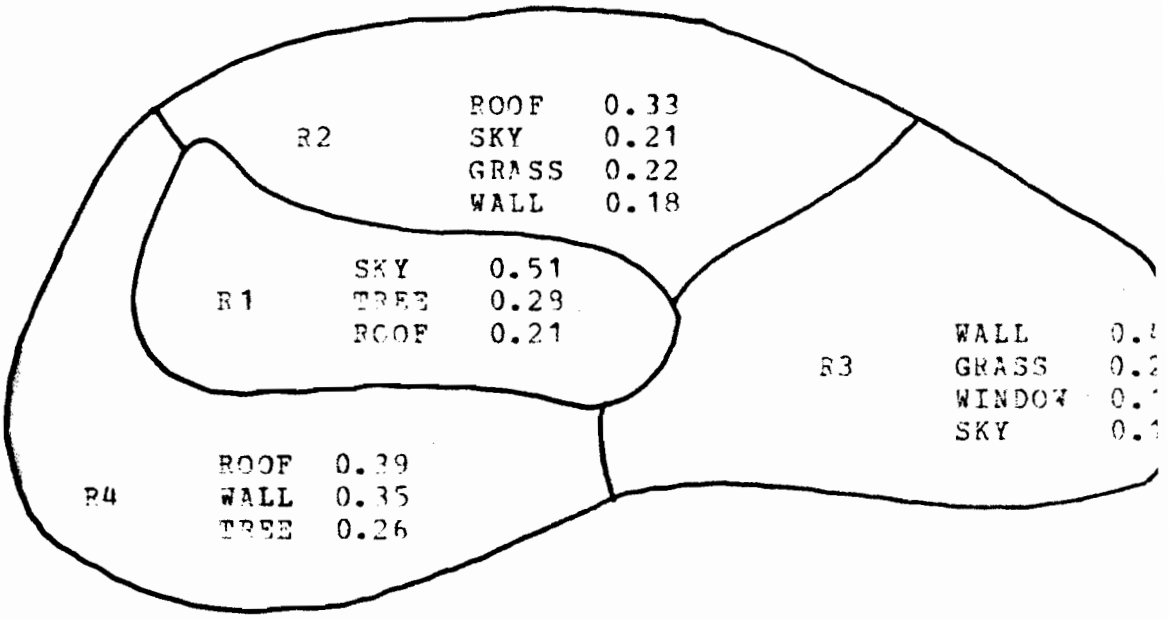


Figure 5.2.a Regions for illustrating the updating process

SKY	ROOF	ABOVE	GT 30	FEATURE1	DIFF-GT 15 . . .
SKY	ROOF	ABOVE	GT 30	FEATURE2	DIFF-GT 15 . . .
SKY	ROOF	ABOVE	GT 30	FEATURE3	DIFF-GT 15 . . .

Figure 5.2.b Relation that contains the constraints involving the objects SKY and ROOF.

represents a single constraint that consists of one or more conditions. A constraint is said to be satisfied, if and only if, all of its conditions are satisfied. In this case, it turns out that there are three possible constraints.

2- The HYPOTHESIS VERIFIER/INTERPRETER compares the features and structural relationships of the two regions R1 and R2 against each of the three constraints. If all the conditions in any one of the constraints are satisfied, regions R1 and R2 are compatible as objects SKY and ROOF respectively; whence, δ_{12}(SKY,RCOF) is set to +1. If none of the constraints is satisfied, δ_{12}(SKY,ROOF) is set to -1 and R1 and R2 are said to be incompatible as objects SKY and ROOF. As an example, consider the HYPOTHESIS VERIFIER/INTERPRETER when it attempts to satisfy the conditions in the first constraint by comparing the value of ABOVE(R1,R2), as specified in the spatial relation ABOVE, tc the threshold value specified in the constraint using the operator GREATER THAN (GT). If the value of the structural relationship ABOVE(R1,R2) is GREATER THAN 0.30, the first condition is satisfied. The second condition is satisfied if the absolute difference between the values of FEAT-1 of regions R1 and R2 is GREATER THAN 15. If both conditions are satisfied, the constraint is satisfied and the other

constraints are not examined. However if any of the two conditions is not satisfied, this constraint is not satisfied and we proceed to check the other constraints. If none of the constraints is satisfied, the regions R1 and R2 are not compatible as objects SKY and ROOF, and δ_{12}(SKY,ROOF) is set to -1. Also, if the relation contains zero tuples (i.e. no constraints are found that relate the two objects) they are considered incompatible. It is possible to specify that two objects are compatible no matter what the feature values of the regions are by inserting a tuple of the form:

OBJECT1 OBJECT2 NULL NULL NULL

in the CONSTRAINTS relation.

3- The above steps are repeated for each interpretation of the adjacent regions R2, R3 and R4.

4- The whole process of steps 1, 2 and 3 is repeated for all of the interpretations of region R1.

5- The HYPOTHESIS VERIFIER/INTERPRETER then calculates:

$$C_1^{k+1}(SKY) = \frac{C_1^k(SKY)\left[1 + \sum_{i=1}^{3} D_{1i} \sum_{I_y} \delta_{1i}(SKY, I_x) C_i^k(I_y)\right]}{\sum_{I_x} C_1^k(I_x)\left[1 + \sum_{j=1}^{3} D_{1j} \sum_{I_y} \delta_{1j}(I_x, I_y) C_j^k(I_y)\right]} \quad (5.18)$$

$$C_1^{k+1}(TREE) = \frac{C_1^k(TREE)\left[1 + \sum_{i=1}^{3} D_{1i} \sum_{I_y} \delta_{1i}(TREE, I_x) C_i^k(I_y)\right]}{\sum_{I_x} C_1^k(I_x)\left[1 + \sum_{j=1}^{3} D_{1j} \sum_{I_y} \delta_{1j}(I_x, I_y) C_j^k(I_y)\right]} \quad (5.19)$$

$$C_1^{k+1}(ROOF) = \frac{C_1^k(ROOF)\left[1 + \sum_{j=1}^{3} D_{1j} \sum_{I_y} \delta_{1j}(ROOF, I_x) C_j^k(I_y)\right]}{\sum_{I_x} C_1^k(I_x)\left[1 + \sum_{j=1}^{3} D_{1j} \sum_{I_y} \delta_{1j}(I_x, I_y) C_j^k(I_y)\right]} \quad (5.20)$$

where:

$C_1^k(SKY)$ is the confidence at iteration k of region R1 as SKY which equals 0.51 in figure 5.2.b.

$C_1^k(TREE)$ is the confidence at iteration k of region R1 as TREE which equals 0.28 in figure 5.2.b.

$C_1^k(ROOF)$ is the confidence at iteration k of region R1 as ROOF which equals 0.21 in figure 5.2.b.

6- Finally, the interpretations of region R1 are ordered according to their new confidences C_1^{k+1} and transmitted to the STM.

Equations 5.15 and 5.16 are the standard updating rules defined in Zucker et al [131]. For practical reasons, the current implementation allows the user to restrict the summation $\sum \delta_{ij} C_j^k(I_y)$ in equation 5.15 to be limited to only a certain number of the most confident interpretations of the neighbouring region R_j. For example, the user may ask that the three most confident interpretations of region R_j be used in the summation. In this case, only the three interpretations of region R_j that have the highest confidence are considered in the interpretation process. This was done in order to decrease the computational requirements for this system when interpreting complex

outdoor scenes. We note that in comparison to the low level applications of the relaxation labelling process, where the same set of conditions had to be considered for every entity in the computation, in our case, the conditions are not independent of the region's interpretations. For example, the constraints that have to be satisfied in examining SKY and ROOF are different from the ones which need to be considered for SKY and SKY. In general a different set of constraints is verified for different possible interpretations, and these are specified in the LTM relations.

One special case of interest is when the user specifies that only the best interpretation of region R_j, that is, the interpretation with the highest confidence, be used in the updating process. Equation 5.15 then becomes:

$$\Delta C_i^k(I_x) = \sum_{j \in A} D_{ij} \; \delta_{ij}(I_x, I_y^*) C_j^k(I_y^*) \qquad (5.21)$$

where I_y^* is the interpretation with the highest confidence level for region R_j and

$$C_i^{k+1}(I_x) = \frac{C_i^k(I_x)[\,1 + \sum_{j \in A} D_{ij}\, \delta_{ij}(I_x, I_y^*) C_j^k(I_y^*)\,]}{\sum_{I_x \in \bar{I}} C_i^k(I_x)[\,1 + \sum_{j \in A} D_{ij}\, \delta_{ij}(I_x, I_y^*) C_j^k(I_y^*)\,]} \qquad (5.22)$$

This therefore implies that only the most confident intepretations have a voting power. It has been shown experimentally (see section 6.5) that the advantage of such

an arrangement is that it provides a faster convergence to the result. In addition, it requires less computation for each region verification. Of course, a disadvantage is that it may possibly lead to the wrong final interpretation. One might view the use of equation 5.21 as a dilution of the available evidence or, on the other hand, the incorporation of all of the available evidence. Because of this, we have allowed the user to select the number of interpretations to be used in the summation in equation 5.15.

In Zucker et al [133, and 135], a generalization of the standard updating rules given in [131] and described before in equations 5.15 and 5.16, was introduced. A new family of updating rules is presented, taking the general form of:

$$C_i^{k+1}(I_x) = C_i^k(I_x) \; \frac{Q_i^k(I_x)}{\sum_{I_x \in I} C_i^k(I_x)\; Q_i^k(I_x)} \qquad (5.23)$$

$$Q_i^k(I_x) = \underset{j \in A}{\text{MEAN}}\; D_{ij} \underset{I_y \in I}{\text{NORM}}\; r_{ij}(I_x, I_y)\; C_j^k(I_y) \qquad (5.24)$$

$$r_{ij}(I_x,I_y) = \begin{cases} 1 \text{ if } R_i \text{ and } R_j \text{ are compatible as objects } I_x \text{ and } I_y \\ 0 \text{ if } R_i \text{ and } R_j \text{ are incompatible as objects } I_x \text{ and } I_y \end{cases} \quad (5.25)$$

where r_{ij} corresponds to δ_{ij}.

Three forms for relaxation updating were introduced:

1- ARITHMETIC RULE:

This can be obtained using the arithmetic mean and L_1 norm.

$$Q_i^k(I_x) = \sum_A D_{ij} \sum_{I_y \in \bar{I}} r_{ij}(I_x,I_y)\ C_j^k(I_y) \quad (5.26)$$

It can also be obtained from the standard relaxation updating rule [131] using the linear transformation:

$$r_{ij}(I_x,I_y) = \frac{1}{p}\,[\ 1 + \delta_{ij}(I_x,I_y)\ ] \quad (5.27)$$

where P is a constant. Thus,

$$\Delta\, C_i^k(I_x) = \sum_j D_{ij} \sum_{I_y} [P\ \ r_{ij}(I_x,I_y) - 1]\, C_j^k(I_y)$$

$$= P\sum_j D_{ij}\sum_{I_y} r_{ij}(I_x,I_y) C_j^k(I_y) - \sum_j D_{ij} \sum_{I_y} C_j^k(I_y) \quad (5.28)$$

Using equations 5.11 and 5.13 we get:

$$\Delta C_i^k(I_x) = [P \sum_j D_{ij} \sum_{I_y} r_{ij}(I_x,I_y)\ C_j^k(I_y)\ ] - 1 \quad (5.29)$$

Thus equation 5.16 can be rewritten as:

$$C_i^{k+1}(I) = \frac{C_i^k(I_x) \sum_j D_{ij} \sum_{I_y} r_{ij}(I_x,I_y)\ C_i^k(I_y)}{\sum_{I_x} C_i^k(I_x) \sum_j D_{ij} \sum_{I_y} r_{ij}(I_x,I_y)\ C_i^k(I_y)} \quad (5.30)$$

2- GEOMETRIC AND PRODUCT RULES:

By using the geometric mean operator and the L_1 norm we get:

$$Q_i^k(I_x) = \prod_{j \in A} [D_{ij} \sum_{I_y} r_{ij}(I_x, I_y)\ C_i^k(I_y)]^{1/\#J_i} \quad (5.31)$$

where $\#J_i$ indicates the number of regions (nodes) adjacent to region R_i. If we further raise $Q_i(I_x)$ to the power $\#J_i$, we obtain the product rule:

$$Q_i^k(I_x) = \prod_{j \in A} [D_{ij} \sum_{I_y} r_{ij}(I_x, I_y)\ C_i^k(I_y)] \quad (5.32)$$

3- THE MAX RULE:

If the geometric mean and the L_∞ norm are used, we get:

$$Q_i^k(I_x) = \prod_{j \in A} \underset{I_y \in \bar{I}}{MAX} [D_{ij}\ r_{ij}(I_x, I_y)\ C_i^k(I_y)] \quad (5.33)$$

Here, the most supportive interpretation for each region R_j is used to represent that region at any iteration. Thus if $r_{ij}(I_x, I_y)$ is restricted to take only the values 0 and 1, a region interpretation which is incompatible with all of the interpretations of an adjacent region is removed (i.e. $C_i^{k+1}(I_x) = 0$). Thus this rule can be considered as a true generalization of the descrete relaxation as discussed in Zucker et al [133].

Although the experimental work done was based on the standard updating relaxation rule, equation 5.16, we have

implemented all of the above rules. Thus the user of the system can interactively select any one of them and experiment with it.

The assumption in the above discussion was that the interpretations for all regions were updated at each iteration. In other words the relaxation process was employed whereby the updating process could be considered as a parallel iterative mechanism[1]. However, even when only the best interpretation is used in the updating process, the computation time is quite excessive. An experimental comparison was made and this is discussed in Chapter VI. Thus the issue of parallel versus sequential updating must be considered. That is, rather than looking at each region in the image and updating the confidences of all the interpretations of all the regions in the image before proceeding to a new iteration, it makes sense to scan regions in the picture in an appropriate sequence, updating a region using, for example, equation 5.16 and then using the new confidences to update other regions. This is the approach that has been taken. After each particular region is considered, the confidences for its interpretation are updated and stored. We have shown experimentally for some test example that the system converges to the same solution

---

[1] No loss of generality occurs when this same computation is structured as a sequential process.

in both parallel and sequential cases[1]. However, the sequential approach converges faster than the parallel.

Each region begins with a set of interpretations and associated confidences which are updated as discussed above. As the analysis proceeds, certain confidences will drop below a threshold, thereby eliminating the associated intepretation. Eventually, the region will remain with one interpretation having 100% confidence, after which time it will not be tested again by the HYPOTHESIS VERIFIER/INTERPRETER. Thus the STM regions will eventually arrive at a mutual agreement and a compatible set of region interpretations will have been attained. In a few situations, the region may have more than one interpretation, and all of them may be compatible with the adjacent regions; in this case the final interpretation will be considered to be the interpretation with the highest confidence level.

We see that the user, in addition to being able to select any of the rules discussed before, also has the option of selecting a parallel or a sequential computation mode. The order in which the regions of the picture are considered is governed by the FOCUS OF ATTENTION process which is discussed in the next section.

---

[1] The sense in which this is meant will be clarified in the next chapter.

5.6 FOCUS OF ATTENTION

In general, the function of the FOCUS OF ATTENTION process is to monitor the current state of knowledge about the scene under consideration as represented in the STM in order to recognize the occurrence of any set of predefined situations stored in the LTM. A set of actions is associated with each of these predefined situations. When a given situation is recognized, the FOCUS OF ATTENTION triggers all of the actions associated with it. The situations and actions are stored as C--->α, where C represents a set of conditions C_1, C_2, C_3, ..., C_n that must be satisfied and α represents a set of actions α_1, α_2, α_3, ..., α_m that are associated with the situation defined by C. The rules are stored in the LTM ACTIONS relation. The actions are stored in the form of pointers (names) of procedures or PLANS that can be activated by the system.

These rules are responsible for directing the system attention to a specific part of the image. They are therefore intimately involved with the control of the analysis as will be seen below. The attention of the analysis, that is the region which is analyzed at any given moment, is accomplished by altering the order of the regions in the REGION STACK, which is stored in the STM. The HYPOTHESIS VERIFIER/INTERPRETER examines the first region in the STACK and updates the confidence of its interpretations using the relaxation updating rules as discussed in the

previous section. Therefore, the FOCUS OF ATTENTION of the computation is adjusted and manipulated by placing a particular sequence of regions on top of the REGION STACK.

Although the more complex aspects of this approach have not yet been fully implemented, it is interesting to consider them in order to understand the importance of the FOCUS OF ATTENTION. The intention is that this process would monitor the current knowledge about the scene under analysis as is manifested by the state of the STM, and attempt to recognize the occurrence of any of the situations as defined in the LTM. When a particular situation would be recognized, this would cause the activation of a set of actions associated with that situation. In general, we observe that these situation/action pairs will be model dependent and therefore act in a similar fashion to what has previously been refered to as "PLANS" [66]. The actions are directed in such a manner as to:

(i) Alter the status of the STM so that the system attention will be directed to a specific part of an image.

(ii) Change the information in the STM.

In the following we present some examples of such PLANS:

1- In face recognition, when a region R_i is interpreted as an EYE with a confidence greater than some threshold, the possible actions of the system might be: Mark all regions that are LEFT-OF/RIGHT-OF that region to be candidates as another EYE. Increase the

confidence of the interpretations FOREHEAD and HAIR of all the regions ABOVE region R_i. Increase the confidence level of the interpretations NOSE, MOUTH, ..., etc, of all the regions BELOW region R_i.

2- If a region is recognized as being SKY with some confidence, a possible action is to increase the confidence of the interpretations SKY and CLOUD of all the regions CONTAINED or ABOVE that region.

3- If a large region is recognized as a TABLE-TOP, it is possible to restrict certain regions to a subset of interpretations:
The regions CONTAINED-BY TABLE-TOP can only be interpreted as either TABLE-TOP, PHONE, or BOOKS depending on the objects that may exist in the model.

From the above discussion, we can easily identify two general classes for the situation/action rules:

1- Rules that use heuristic semantic information to locate certain objects, to restrict and update the interpretations associated with some regions, or even to stop the analysis if certain situations arise.

2- Rules that direct the system attention to a specific part of the image. This is achieved by placing a certain sequence of regions on the top of the REGION STACK. Thus the system will be directed to examine certain specific parts of the picture.

These actions are implemented as heuristic procedures that are model dependent and implement certain vision strategies. We have not designed a language to express these condition/action rules. However, it is expected that this language will be very similar to that of a production system, implemented within the framework of the relational database formalism.

At present the user can only express situations that involve the interpretations and confidences of a group of adjacent regions. Although this capability is incorporated into the FOCUS OF ATTENTION process, it was in fact not used for the experiments to be described in the following chapter. Only the default action described below was employed.

As has been discussed in the previous section, the system's attention is focused at the regions on the top of the REGION STACK. Thus there exists an intrinsic action that is invoked to alter the order of this REGION STACK if no other appropriate action can be triggered. This action implements a worst-first strategy. First, it inserts the region R_x which is at the top of the REGION STACK (i.e. the region whose interpretation has been updated by the HYPOTHESIS VERIFIER/INTERPRETER) at the bottom. In addition, it brings all of the regions which are neighbours of region R_x to the top of the REGION STACK in descending order of their most confident interpretation. The

HYPOTHESIS VERIFIER/INTERPRETER always examines the region which is at the top of the REGION STACK. Obviously, the next region to be examined will be the region, with the least confidence in its best interpretation and in the local neighbourhood of the previously examined region. There are exceptions to this as described below.

Let us consider a simple example as shown in figure 5.3 where the STM contains only nine regions R1, R2, ..., R9. Because the initial order of the REGION STACK is not important as will be seen later, let the initial REGION STACK be ordered as follows: R1, R2, ..., R9. The HYPOTHESIS VERIFIER/INTERPRETER will first test the region on the top of the REGION STACK, that is region R1. The confidence of its interpretations are updated, and the result transmitted to the STM. The FOCUS OF ATTENTION will then place region R1 at the bottom of the REGION STACK and the neighbours of region R1 will be brought to the top of the REGION STACK in decreasing order of the confidence of their best interpretation. Thus, the REGION STACK might be as follows:

R4 R2 R5 R3 R6 R7 R8 R9 R1

The next region examined will be R4 because it is now on top of the REGION STACK. After this iteration the new order of the REGION STACK might become:

R7 R1 R8 R5 R2 R3 R6 R9 R4

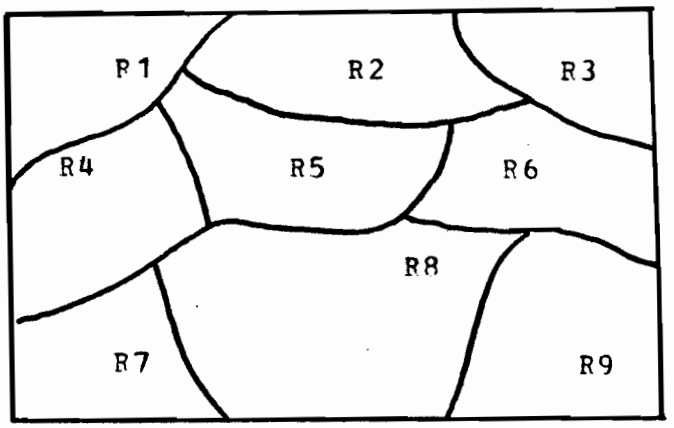


Figure 5.3.a Example of a REGION MAP

ITERATION NUMBER	REGION STACK
0	(1) 2 3 4 5 6 7 8 9
1	(4) 2 5 3 6 7 8 9 1
2	(7) 1 8 5 2 3 6 9 4
3	(8) 4 1 5 2 3 6 9 7
4	(4) 7 6 5 9 1 2 3 8
.	. ...

Figure 5.3.b The REGION STACK changes after each iteration of the region updating process

Thus region R7 will be the next to be examined. As can be seen from the above discussion and figure 5.3.b, where this process has been carried on for a few iterations, the system will be examining regions in a very small neighbourhood. In this way, the FOCUS OF ATTENTION of the system is directed to different regions in the image and can be represented by the directed graph in figure 5.4.a.

The above discussion avoided the situation whereby the system is locked into a sequence of regions. Consider iteration 3 and 4 in figure 5.3.b where we have started with the sequence R8 R4 ... R7 in iteration 3 and have the sequence R4 R7 R6 R5 ... R8 in iteration 4. If iteration 5 results in region R8 being on top again, we will be alternatively testing region R4 and region R8 until their confidences are high enough such that the sequence is broken and another region is tested. However, should the confidences of region R4 and region R8 oscillate, the oscillation would not cease. In fact, this problem is more complex, especially if we have a sequence involving 3 or 4 regions. This situation has the disadvantage of locking the system into a small neighbourhood and not allowing global information to propagate through the image, which is contrary to the spirit of the concept of competition and cooperation.

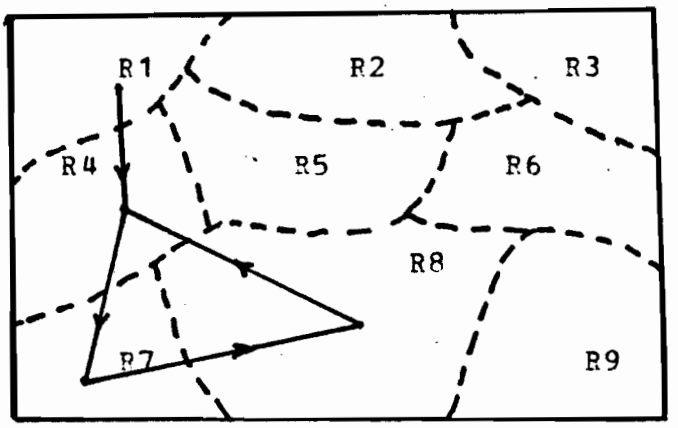


Figure 5.4.a The FOCUS OF ATTENTION without the use of the QUEUE

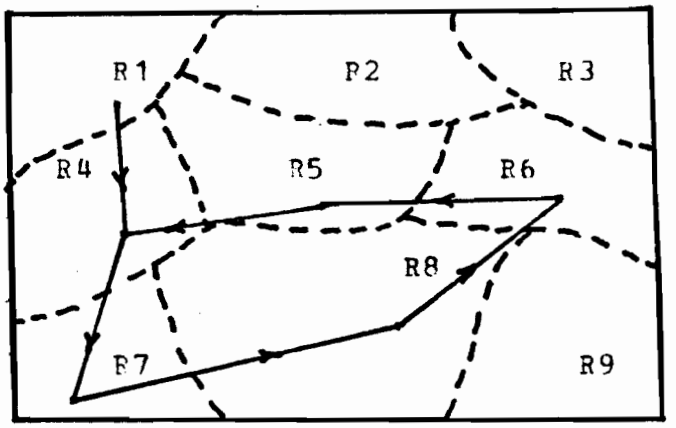


Figure 5.4.b The FOCUS OF ATTENTION with the QUEUE of length 4

We have resolved this problem by introducing a first-in / first-out QUEUE of finite length L. When the top region in the REGION STACK is tested and then placed at the bottom, it is also added to the QUEUE. If a region R_i happens to be on the top of the REGION STACK and is also still in the QUEUE, the HYPOTHESIS VERIFIER/INTERPRETER will neglect it and consider the region below it on the REGION STACK. Consider figure 5.5, for example, which represents the different iterations of the example given in figure 5.3, but now with a QUEUE of length 4. As can easily be seen, neither region R4 nor region R7 are tested in iteration 4 although they are on top of the REGION STACK. The system examines region R6 which was the first region near the top not in the QUEUE. We note that no region will be tested twice unless L regions are first tested. This approach does not allow the attention of the system to remain in a small neighbourhood of the picture and forces it to move to other areas. This area is determined by the QUEUE length L. The larger the value of L, the larger the neighbourhood the system must examine before reconsidering the same region. If L is equal to M-1 where M has been previously defined as the number of STM regions, the HYPOTHSIS VERIFIER/INTERPRETER will have to examine all of the regions before considering the same region again. This is equivalent to implementing a sequential relaxation process

ITERATION NUMBER	REGION STACK	QUEUE
0	(1) 2 3 4 5 6 7 8 9	0 0 0 0
1	(4) 2 5 3 6 7 8 9 1	1 0 0 0
2	(7) 1 8 5 2 3 6 9 4	4 1 0 0
3	(8) 4 1 5 2 3 6 9 7	7 4 1 0
4	4 7 6 5 9 1 2 3 8	8 7 4 1
4	(6) 4 7 5 9 1 2 3 8	8 7 4 1
5	8 5 2 3 9 4 7 1 6	6 8 7 4
5	(5) 8 2 3 9 4 7 1 6	6 8 7 4
6	8 4 6 2 3 9 7 1 5	5 6 8 7
6	(4) 8 6 2 3 9 7 1 5	5 6 8 7
.	...	...

Figure 5.5 Example of the FOCUS OF ATTENTION with a QUEUE of length 4.

with updating after each region is considered. On the other hand, if L is equal to 1, there is no control over how regions are tested, and no gaurantee of stable processing. The parameter L is input by the system user at the outset of the analysis. We have experimented with different values for the QUEUE length L and the detailed results are discussed in the next chapter. The most favourable value of the QUEUE length L has been found experimentally to be in the range of M/4 to M/2 where M is the number of regions in the STM.

With the aid of the PATH relation (Appendix II), it is possible to later track the FOCUS OF ATTENTION of the system and in this way study the effects of the different values of L. Obviously, other vision strategies can be postulated. These would only require the addition of more procedures to the FOCUS OF ATTENTION module.

So far we have considered the different processes implemented in the system without any reference to how they are controlled to run in harmony with each other. In the next section we will consider the SCHEDULER that directs the flow of control between the different processes.

5.7 SCHEDULER

In the previous sections we have presented the individual processes of the computer vision system. Throughout the discussion we have stressed the fact that no direct communication is allowed between different processes, a property which enforces the desirable level of independence. Thus, no process is allowed to access the data structure or the private bookkeeping of another process. Further, individual processes are prohibited from direct activation of other processes because the invocation of a process may depend on a complex set of conditions which is created by several other processes. Such direct activation also contradicts the requirement of independence because a process will now have to contain information about the others activated by it. Whence, we need to define some mechanism by which the different processes are activated. This mechanism should be able to decide which process is to be triggered based on the status of the analysis.

To maintain the original design considerations, two possibilities present themselves for designing such a mechanism. One is to allow each process to monitor the STM for changes in the status of the analysis which may interest it. Thus, each process would be able to initiate itself based on its own measurements. This approach tends to be a

very costly computation and would require some methodology for deciding which process should be initiated when a conflict transpires. An alternative to this approach is to create a special process, the SCHEDULER which monitors the STM status for all the processes, attempting to identify an activity appropriate to the initiation of a particular process. In this way, it acts as a measurement and activation mechanism for all of the processes collectively. This allows for a more efficient use of the computer resources while still not negating the possibility of parallel operation.

The measurements are done by procedures within the SCHEDULER which is continuously monitoring the status of analysis attempting to decide which of the system's processes would be initiated. This mechanism of control is data-directed because a process is only activated if a status of the anaysis as revealed in the STM required its activation. Further, the modification made by any given process is expected to trigger other processes by changing the STM contents such that other processes may be initiated.

In particular, the SCHEDULER monitors whether any of the regions in the REGION STACK lacks interpretations or has a unique interpretation, whether the REGION MAP exists, and whether the STM has all the feature values for all the

regions in the REGION STACK. In addition, it senses when the order of the regions in the REGION STACK has been altered and when the analysis should be stopped. The SCHEDULER does not resemble a centralized control mechanism because it is not a fixed process with predefined conditions such that it is difficult to add, remove or update the system processes or their activation pre-conditions. That is, the SCHEDULER might be conceived as accessing a control relation in the LTM with three domains:

CONTROL = (PROCESS-NAME, PRIORITY, POINTER) where the POINTER is to a procedure that specifies the condition of the activation of that process. Adding a process would require adding the appropriate entry to the CONTROL relation and supplying the appropriate measurement procedure. Changing the pre-conditions or the measurements required by a process would require the definition of a new procedure. Finally, removing a process is simply attained by deleting its entries in the CONTROL relation. Thus, it is possible to specify the triggering of the same process under different preconditions with different priorities. The priority is used here to resolve conflicts if two or more of the processes can be initiated. The priority of the processes in decreasing order are: LOW LEVEL, FEATURE ANALYZER, HYPOTHESIS INITIALIZER, HYPOTHESIS VERIFIER/INTERPRETER, and FOCUS OF ATTENTION.

The LOW LEVEL process is activated whenever it is

determined that a REGION MAP does not exist. In the future, it may be activated to reanalyze a specific part of the picture. For example, if the system suggests the existence of an edge at some point during the analysis, the LOW LEVEL PROCESSOR may be activated to verify the existence of this edge. The later example is not implemented but it is conceivably possible in a more sophisticated LOW LEVEL process. The FEATURE ANALYZER is obviously called upon when a region does not have adequate feature values. This may happen initially after generating the REGION MAP or when a region reaches a state with a unique interpretation (i.e. confidence level of interpretation is greater than a threshold). The latter suggests the possibility that the region may be merged with a region(or regions) that is adjacent to it and has the same unique single interpretation. The HYPOTHESIS INITIALIZER is triggered when any of the regions in the REGION MAP lacks interpretations. The HYPOTHESIS VERIFIER/INTERPRETER is activated when the order of the regions in the REGION STACK of the STM changes.

In addition to measuring the system status and selecting the appropriate process to be initiated, the SCHEDULER also monitors the system to find whether or not there was enough activity in the STM to justify the continuation of the analysis. Currently, the activity of the system can be measured by the changes in the confidence

levels of the regions in the STM. We define ΔP as the maximum change in the confidence level over all the regions in the STM and for all the interpretations of each region. It is given by:

$$\Delta P = \max_{\substack{i \\ I_x \in I}} \left| C_i^{k+1}(I_x) - C_i^{k}(I_x) \right| \qquad (5.34)$$

where $C_i^k(I_x)$ is the confidence level of interpreting region R_i as object I_x. Thus, the SCHEDULER will stop the analysis if ΔP remains below a threshold value determined by the user of the system. During the experimental phase of this work, this threshold value was equal to 0.01. Then if the changes in the confidence of all the regions is less than 0.01, the SCHEDULER stops the analysis.

The stopping criterion is designed as an independent procedure, therefore, the different system users are able to specify new stopping criteria. However, they must supply the procedure that defines these criteria. For example, in some experiments we have used another criterion which is less strict than the one previously defined. We have defined ΔP as:

$$\Delta P = \sum_{i=1}^{N} \left| C_i^{k+1}(I_x) - C_i^{k}(I_x) \right| \qquad (5.35)$$

where the summation is applied to the last N regions tested by the HYPOTHESIS VERIFIER/INTERPRETER. When using this criterion, the user is asked to specify both N and the

threshold value. The first criterion is more strict because it specifies that all regions in the STM should be consulted before stopping the system. On the other hand, the second criterion considers only the last N regions examined. The first criterion ensures that the activity of the system has essentially stabilized after testing each region. The other criterion checks whether there has been sufficient change in the confidence levels to justify the continuation of the analysis.

Finally, the FOCUS OF ATTENTION is brought into play by default, if according to the measurements, none of the other processes can be activated and the status of the system does not justify the stopping of the analysis. In conclusion, the SCHEDULER is a powerful tool for monitoring the system's behaviour represented by the STM activity, and for deciding on the next appropriate action.

In sequential operation, the SCHEDULER is activated after each individual process to decide which process is to be invoked next. However, because of the independence of the system processes and their data-directed activation, it is also possible to organize the system to run in asynchronous (parallel) format.

CHAPTER VI

RESULTS AND CONCLUSIONS

6.1 INTRODUCTION

During the design and implementation of this computer vision system, we have performed a number of experiments to test each of the processes described in the previous chapter, as well as the system's overall performance. These experiments were done on a DEC PDP-15 computer with 32K words of memory. Initially, we used a test picture which was manually constructed for training the system and checking the correctness of the computer code of each process. Further experiments were performed on real outdoor scenes of which figure 6.1 is typical[1] [2]. In the following, we will present the results obtained from the individual processes. Then, we will present some experiments performed on the test picture to investigate the effects of parallel and sequential processing and the effects of different values of the queue length on the speed of convergence. Furthermore, we will discuss the results obtained when the analysis was applied to a real outdoor scene.

---

[1] All colour pictures in this thesis were photographed off a convac RGB monitor refreshed by a Grinnell Frame Memory System.

[2] This test data was provided by Dr. Robert M. Haralick of the Virginia Polytechnic Institute and State University.

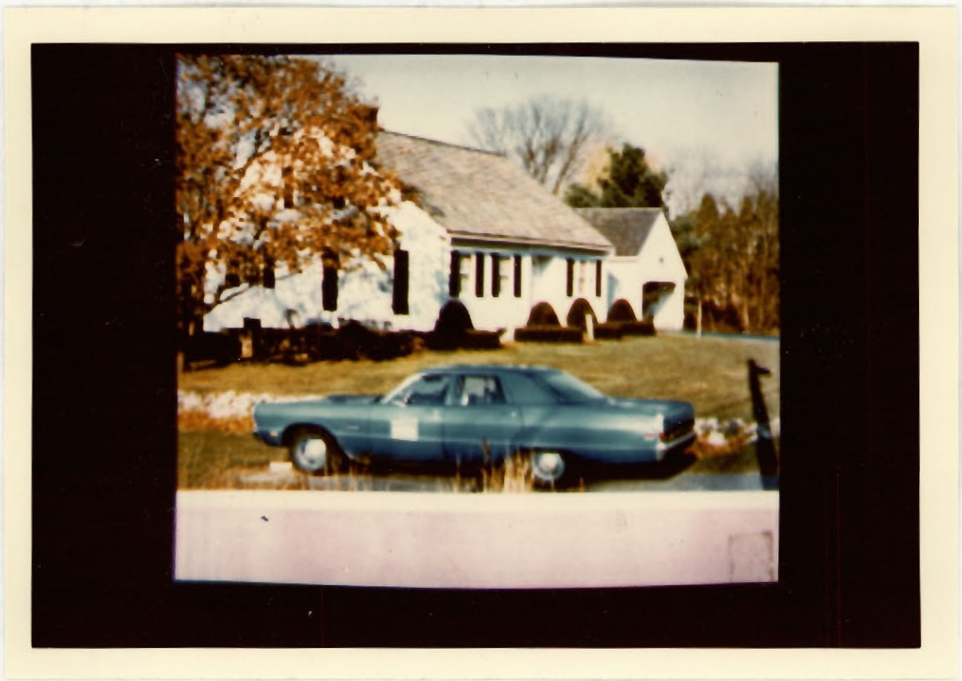
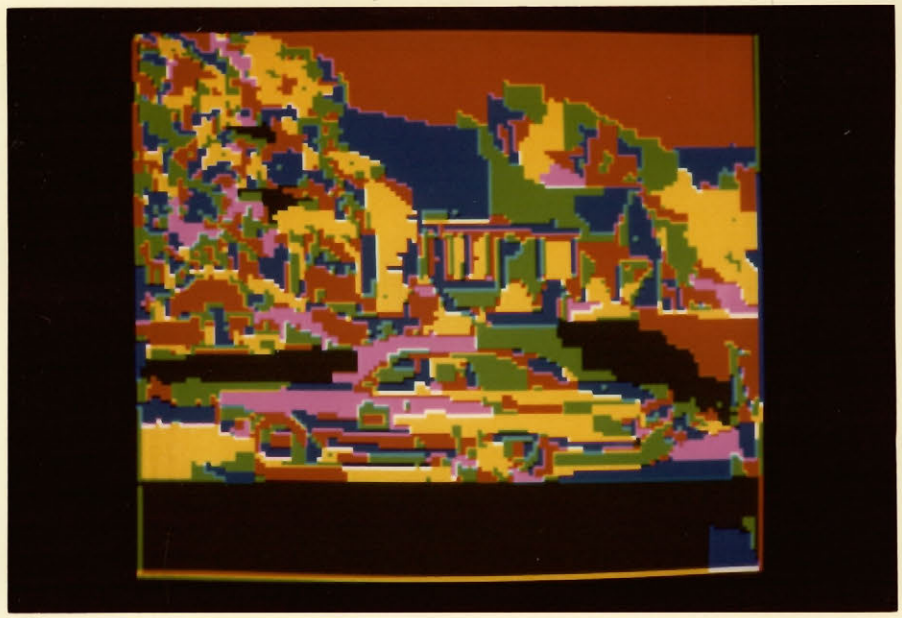


Figure 6.1 A typical real outdoor scene.

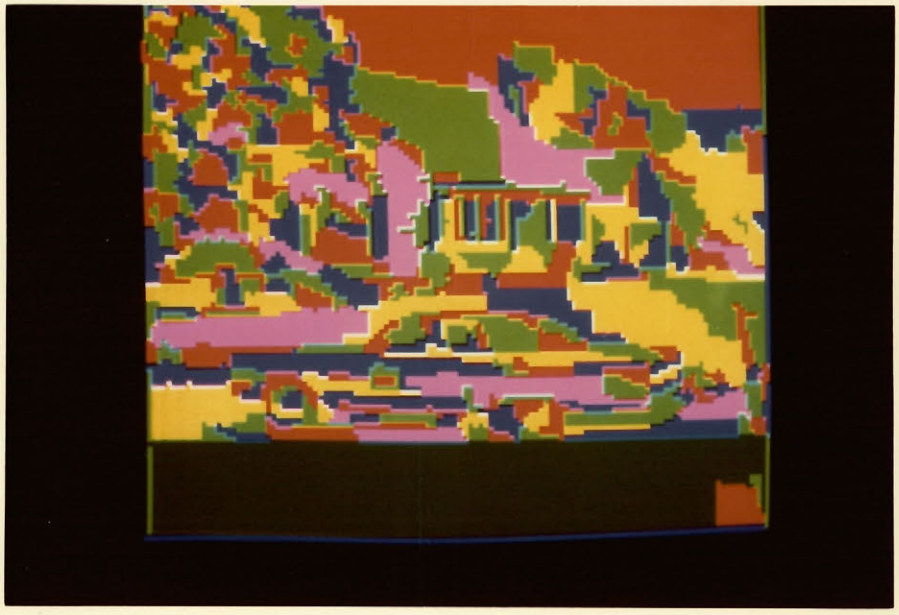
6.2 LOW LEVEL SEGMENTATION

The LOW LEVEL PROCESSOR partitions the input picture into regions containing pixels whose RED, GREEN and BLUE feature values are similar. We have tested this process on the picture shown in figure 6.1. The picture was represented by three 128X128 arrays representing the RED, GREEN and BLUE picture scans. It was segmented into regions using different values for the threshold parameter h in equation 5.5 in the range from 5 to 20.

Figure 6.2.a represents the REGION MAP obtained with threshold value 10. It contains 584 regions. This REGION MAP was then cleaned up by marking the pixels of the regions whose area was less than 8 pixels. These pixels were then merged with the nearest most similar existing region. Thus begining with a 128X128 image, the resulting analysis yields a REGION MAP with 292 regions shown in figure 6.2.b. We note that no formal method for evaluating the goodness of such a partial segmentation is available. However, it was found that the resulting region boundaries did largely conform to the object boundaries in the picture (figure 6.3). In addition, the algorithm was sufficiently sensitive to isolate the very low contrast (with respect to the SKY) CROWN of the TREE occluded by the ROOF of the HOUSE. The



(a) REGION MAP with threshold parameter h=10.



(b) REGION MAP after clean-up operation with clean-up parameter equal to 8.

Figure 6.2 REGION MAP for figure 6.1 with regions arbitrarily coloured.



Figure 6.3 Illustration of the region boundaries with reference to the original picture of the regions in figure 6.2.b.

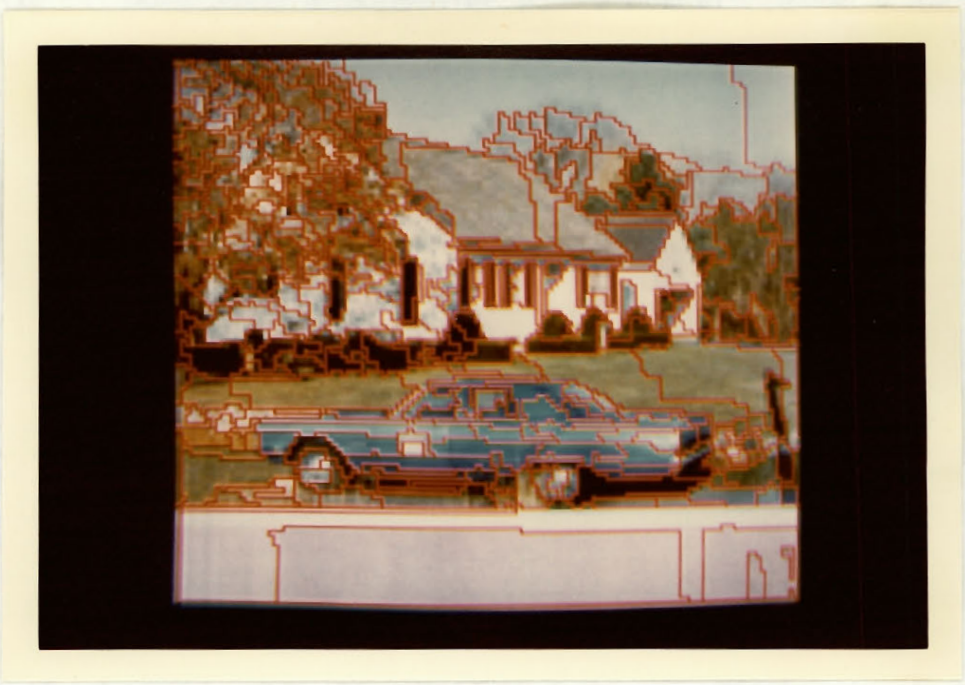
performance was also very good in the smooth areas, for example, the SKY region at the top, as well as the weak textured areas such as the ROOF, GRASS, BUSHES and RIGHT TREE.

We have segmented the same picture using other values for the threshold parameter h and the results are shown in figures 6.4.a and 6.4.b for threshold values of 7 and 15, respectively. It is obvious that as the threshold value decreases, the segmentation process becomes more conservative and the resulting segmentation has a larger number of regions. As the value of h increases, the process is less conservative, and thus regions can attract more pixels and grow larger.

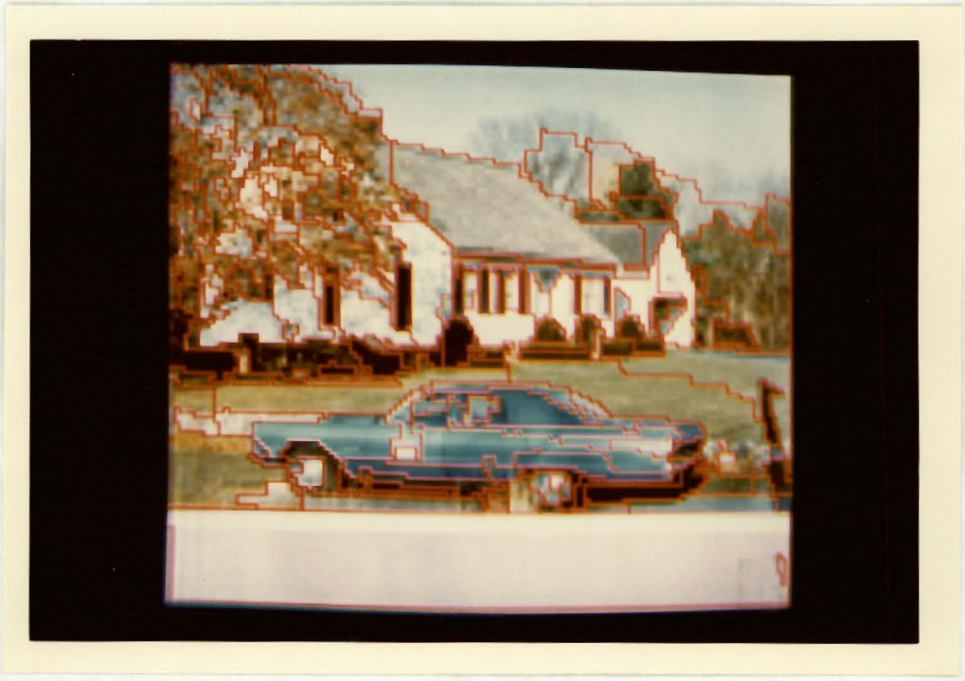
In general, this segmentation process is fast and simple. The only parameters the user must input are the threshold value h and the minimum area that is used in the clean-up operation. The threshold value h is domain dependent and can be determined for different classes of pictures experimentally[1]. The second parameter is used to remove the isolated noise pixels, thus limiting the number of regions in the resulting partial segmentation. If the

---

[1] Ideally we could choose a low value of h and obtain a REGION MAP containing a large number of regions. Such a value would probably be suitable for many different pictures. However, the total analysis time to obtain the final interpretation would be excessive given the computer we used.



(a) h=7



(b) h=15

Figure 6.4 Region boudaries with different threshold values and clean-up parameter equal to 8.

minimum area is specified to be zero, no clean-up is performed.

The CPU time required for the LOW LEVEL segmentation of a 128X128 colour picture on a DEC PDP-15 was in the range of 4-5 minutes. However, when we later implemented the algorithm on a DEC VAX 11/780, it required only about 20-30 seconds. This time reduction can be attributed to the increase in the amount of memory available on the computer system as well as the speed of the machine.

6.3 HYPOTHESIS INITIALIZATION

The function of the HYPOTHESIS INITIALIZER is to generate a set of initial hypotheses about the regions in the STM (REGION MAP). It employs the model information in the LTM which is stored in the DISTINGUISHING FEATURES and OBJECT FEATURES relations. The first relation specifies the features which are used to distinguish the objects in the model, the weight of each feature for the initialization process, and whether or not the feature has a veto. The OBJECT FEATURES relation specifies, for each object, the average value of its distinguishing features and the acceptable range of variation about that average.

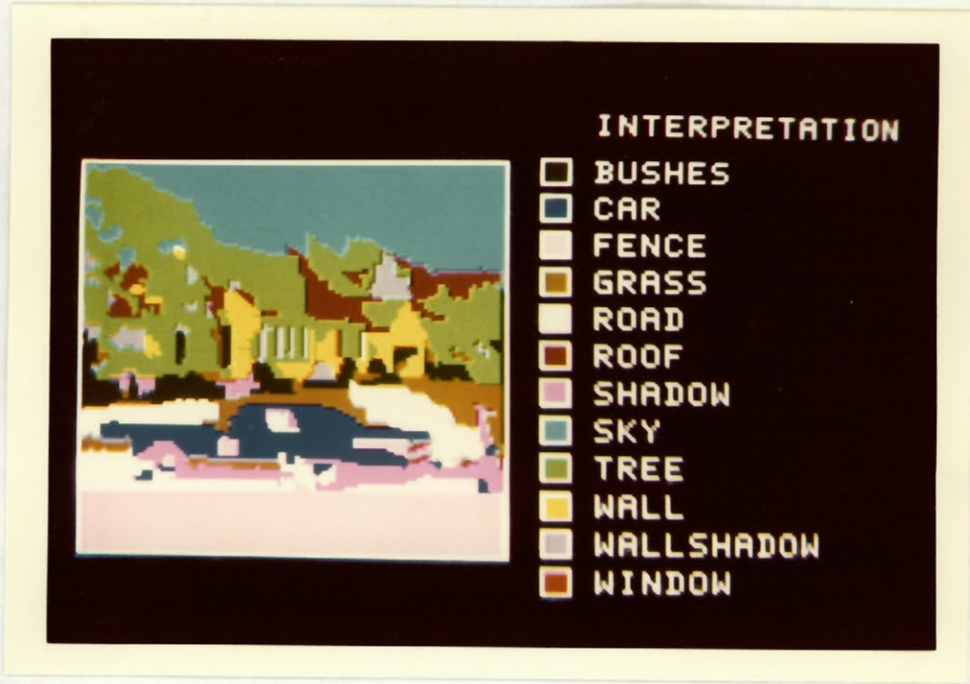
The features used in this initialization stage are the three colour features (RED, GREEN and BLUE) and the vertical position (CENTROID-Y) of the object. We should here stress the fact that the level of analysis under consideration is such that it begins with a partial segmentation and attempts to produce a labelled complete segmentation of the input picture. Since the latter is only achieved as a final result, the analysis is always proceeding on the basis of very fragmented regions. Therefore, information about objects such as their geometric structure (shape) is useless and inapplicable.

To investigate the initialization process we have performed a number of experiments. These experiments used different combinations of the distinguishing features and weights. For each experiment, the HYPOTHESIS INITIALIZER generated a list of all possible interpretations and their associated confidences as described in Chapter V. In every case the results were stored in the STM INTERPRETATION relation. These are displayed in a series of pictures shown in figures 6.5-6.8 where each region is painted with a colour that corresponds to the interpretation given to it. The confidences associated with the interpretations are also represented in colour pictures, with the colour representing the scale displayed to the right of the pictures.

Figures 6.5.(a-c) represent the initial interpretations and their associated confidences using the DISTINGUISHING FEATURES and the OBJECT FEATURES relations shown in APPENDIX II[1]. In this case, the RED, GREEN and BLUE colour features and CENTROID-Y of the objects are used as distinguishing features with relative weights 1:1:1:3, respectively, for all of the objects in the model. In effect, the model assumes that all three colour features have a total weight equal to that of the position. The model also gives a veto to the vertical position of the object. This veto should be understood in the sense that, if the vertical position of the region is not within the acceptable limits specified for the object, the confidence is set to zero as described in section 5.3. The actual feature values and the acceptable variation for each are in Appendix II in the OBJECT FEATURES relation. Shown in figure 6.5 is the resulting set of the first three interpretations and their associated confidences. Some regions start with as much as 8 possible interpretations, while others have only 2 or 3. For example, a region that lies in the upper third of the image may be associated with SKY, TREE, WALL, WALL-SHADOW and ROOF. On the other hand, a region in the middle section might be associated with TREE, WALL, WALL-SHADOW, WINDOW, GRASS, BUSHES, CAR, and so on.

---

[1] Note that although only the first three best interpretations and their associated confidences are given in this thesis, in fact, the number of interpretations for each region may be as high as eight.



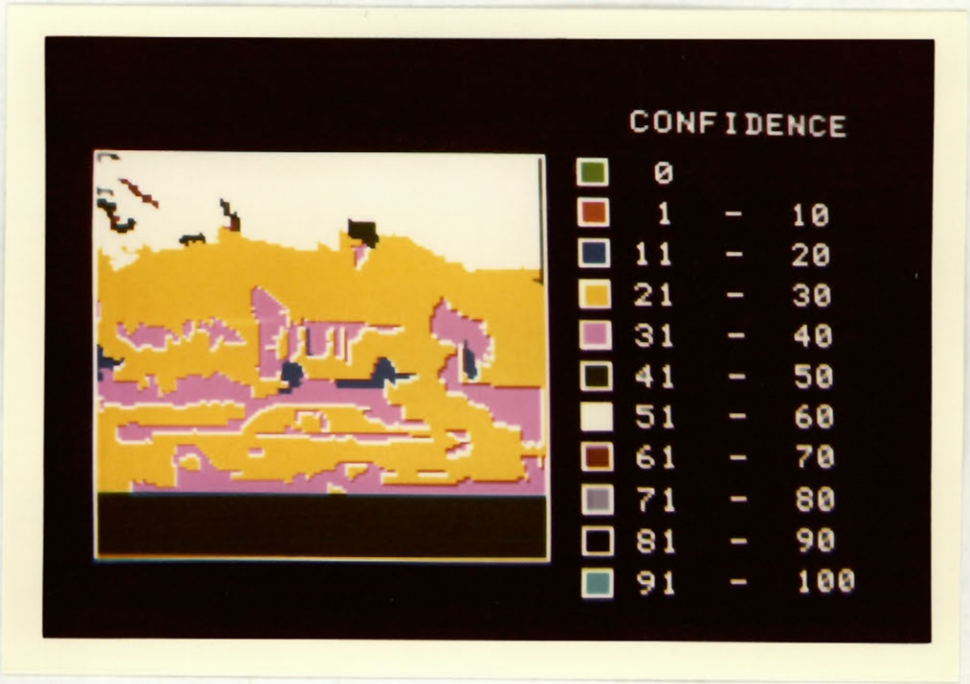
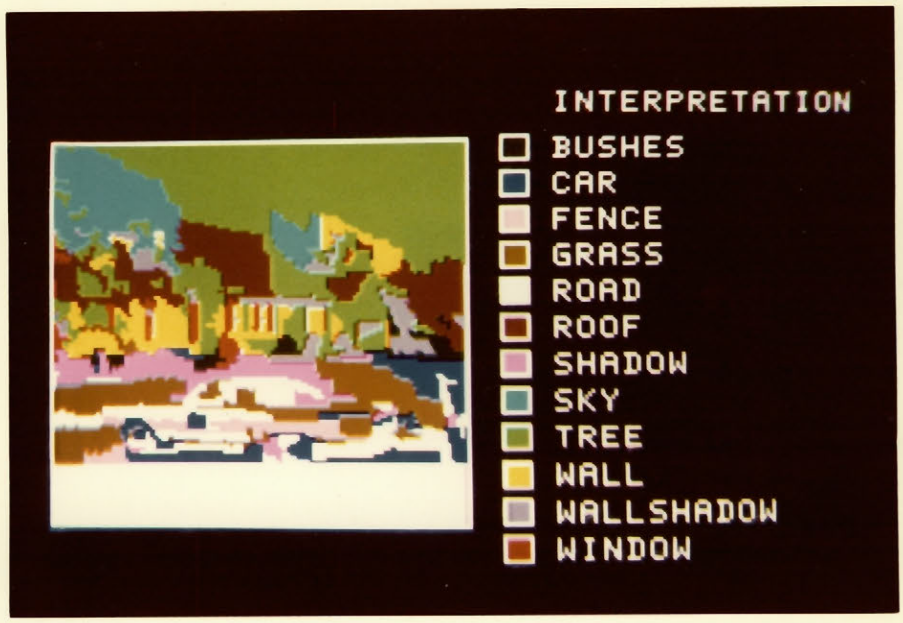


Figure 6.5.a Initial first best interpretations and associated confidences for the REGION MAP in figure 6.2.b when colour and position are used as distinguishing features and position was given a veto power.



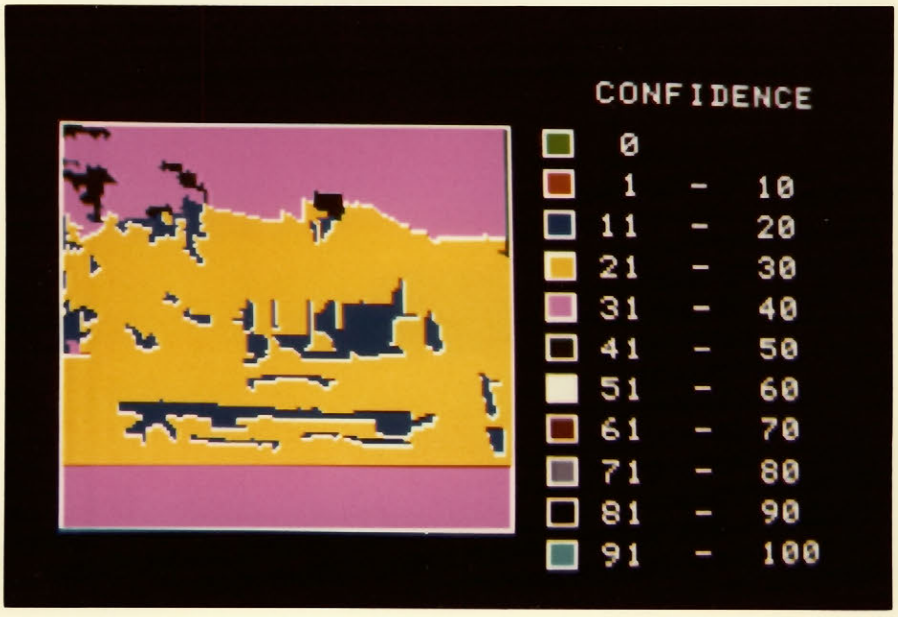
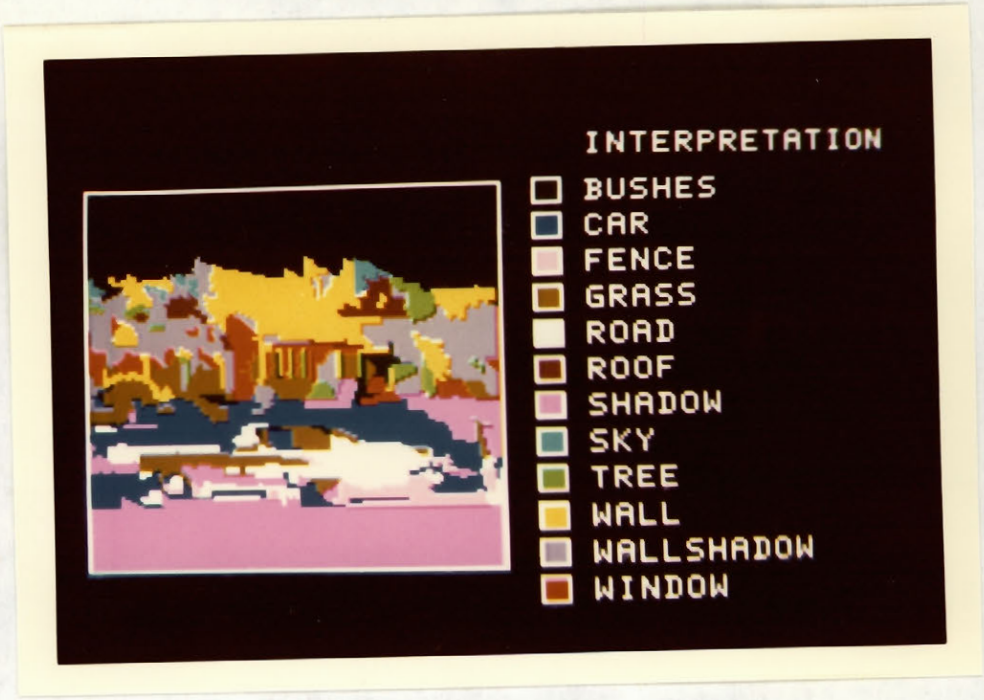


Figure 6.5.b Initial second best interpretations and associated confidences for the REGION MAP in figure 6.2.b when colour and position are used as distinguishing features and position was given a veto power.



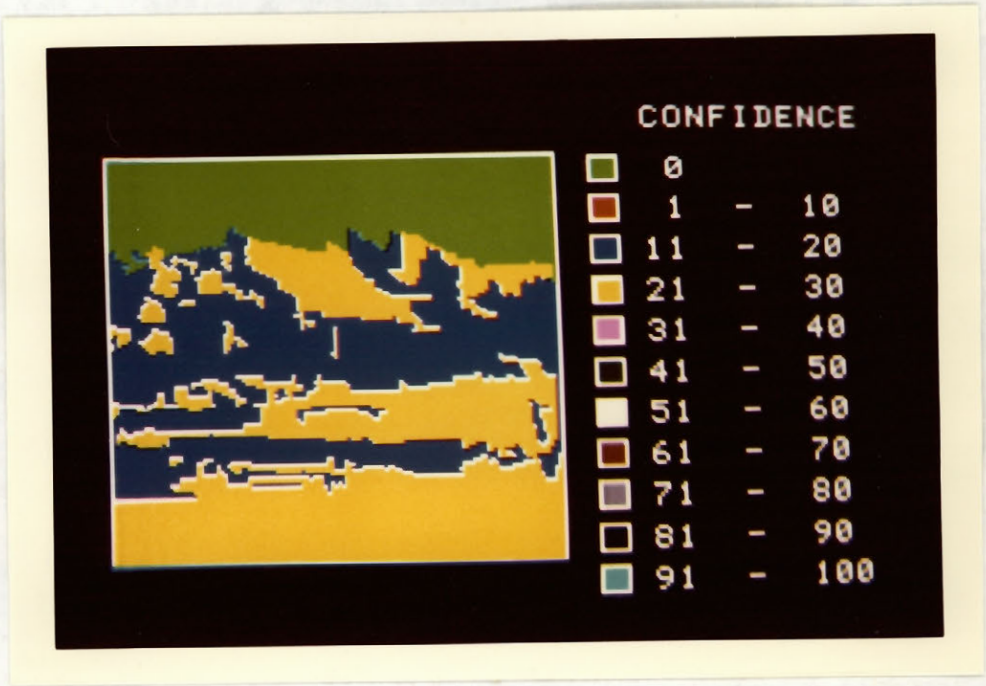


Figure 6.5.c Initial third best interpretations and associated confidences for the REGION MAP in figure 6.2.b when colour and position are used as distinguishing features and position was given a veto power.

It is easy to see that most of the regions are initialized with approximately equal confidences for their first three or four interpretations. In some cases, a region may even begin with all of its interpretations about equally probable. Thus, at the outset, either most of the region interpretations are significantly in error, or at the very least confused. That is, they have many interpretations which are equally probable.

Next, we changed the model such that only the colour information represented by the RED, GREEN and BLUE features are used for the initialization with equal weights and no veto assigned to any of them. The results are shown in figure 6.6. We note that each region commences with a larger number of interpretations than in the previous experiment. In addition the confidence levels of most of the interpretations are equal. Furthermore, some regions are given impossible interpretations. For example, the SKY interpretation was assigned to the large region at the bottom of the picture, and FENCE was given as an interpretation for regions at the top. Thus using the colour features as the only distinguishing features introduces many unrealistic, even impossible, region

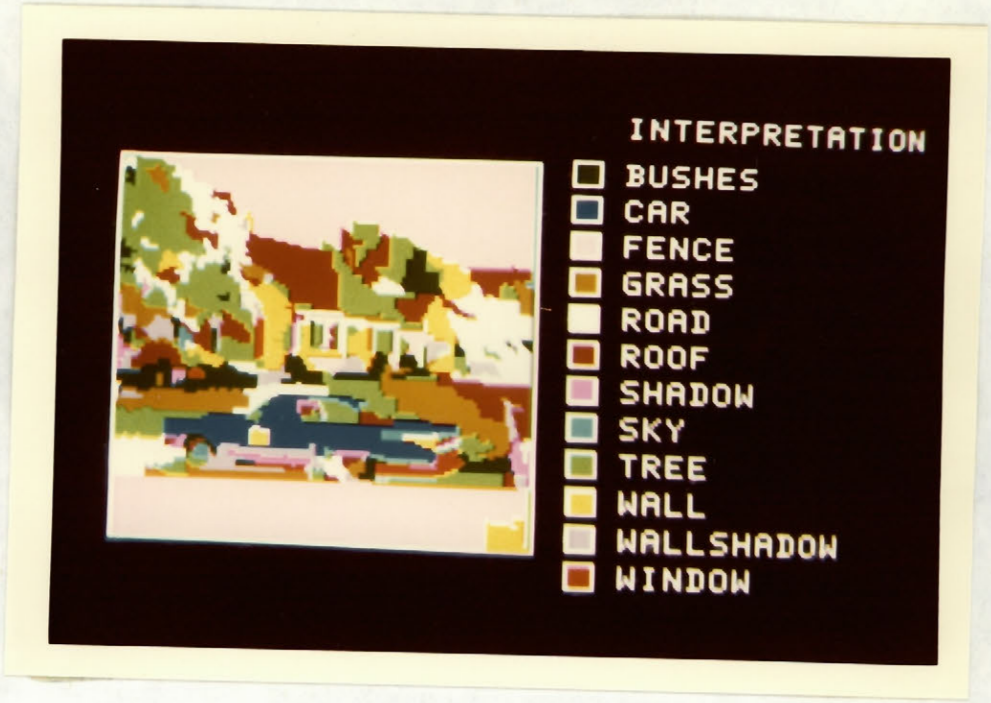




Figure 6.6.a Initial first best interpretations and associated confidences when colour is used as the only distinguishing feature.



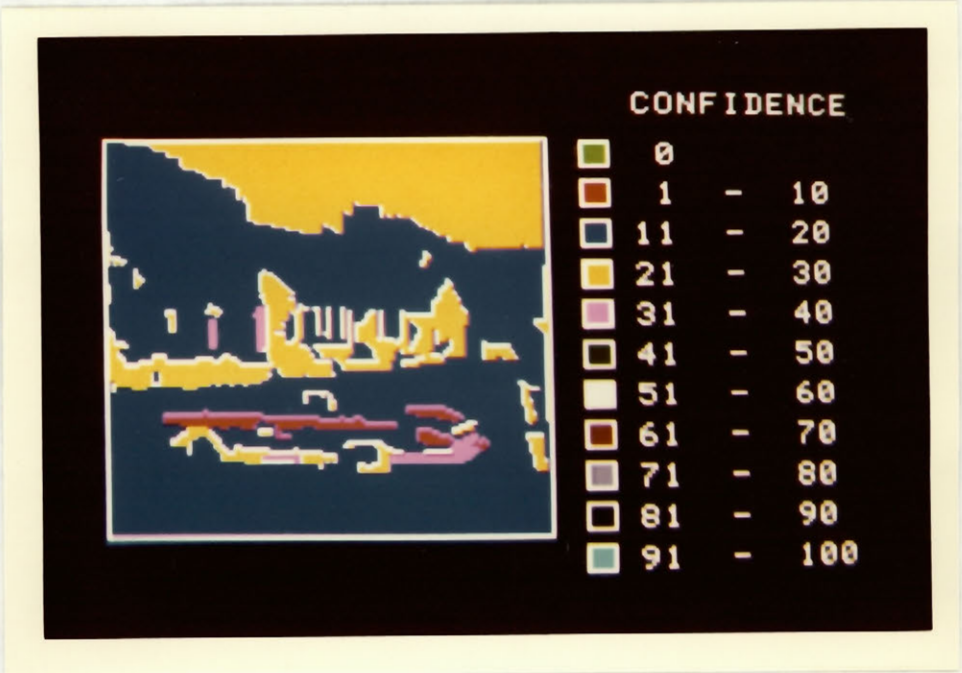


Figure 6.6.b Initial second best interpretations and associated confidences when colour is used as the only distinguishing feature.

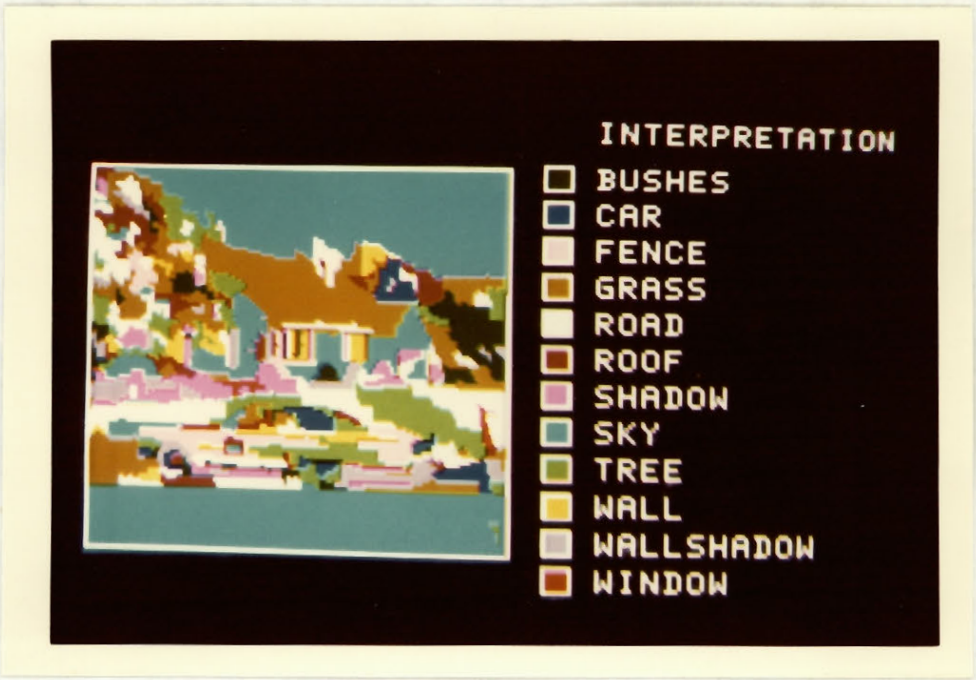




Figure 6.6.c Initial third best interpretations and associated confidences when colour is used as the only distinguishing feature.

interpretations. In view of the fact that the colours of the objects in real outdoor pictures are all clustered around the white reference point in the colour triangle, unless we restrict objects to exist in a very small range around the centre of each object's average colour feature, the colour information provides insufficient evidence for the initialization process.

Generally speaking, using only colour features in the initialization process would have added many unrealistic interpretations to the list of interpretations of each region. This in turn introduces confusing evidence in the updating process. Thus propagation of information during the relaxation process will be diffused. This will increase excessively the time required for the analysis. Furthermore, some regions may acquire erroneous unique interpretations based on the false evidence propagated from their local neighbourhood. This would result in an increase in the amount of errors in the final interpretation. It is for this reason that we introduced the vertical position feature in the next experiment.

In this experiment, shown in figure 6.7, we used the colour features (RED, GREEN and BLUE) and the vertical position. However these features were given equal weight



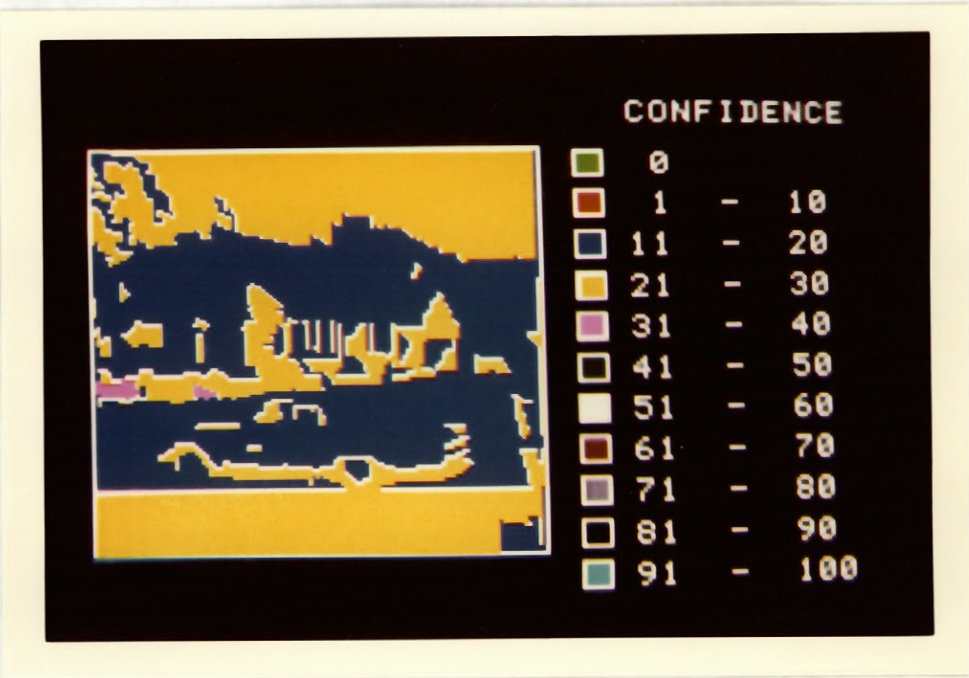


Figure 6.7.a Initial first best interpretations and associated confidences when colour and position are used as distinguishing features with equal weight and no veto.



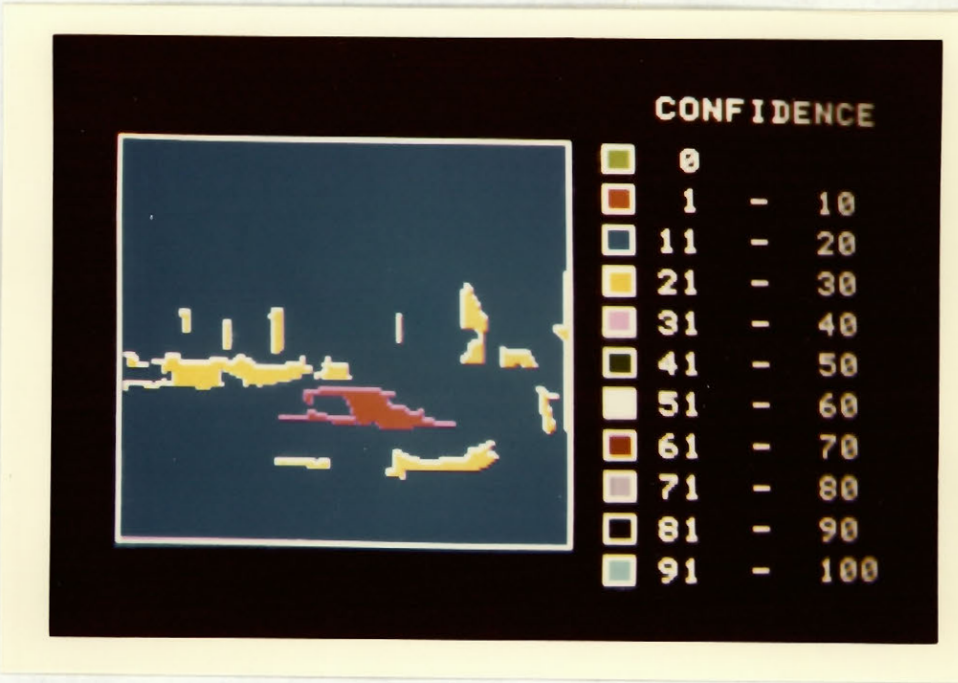
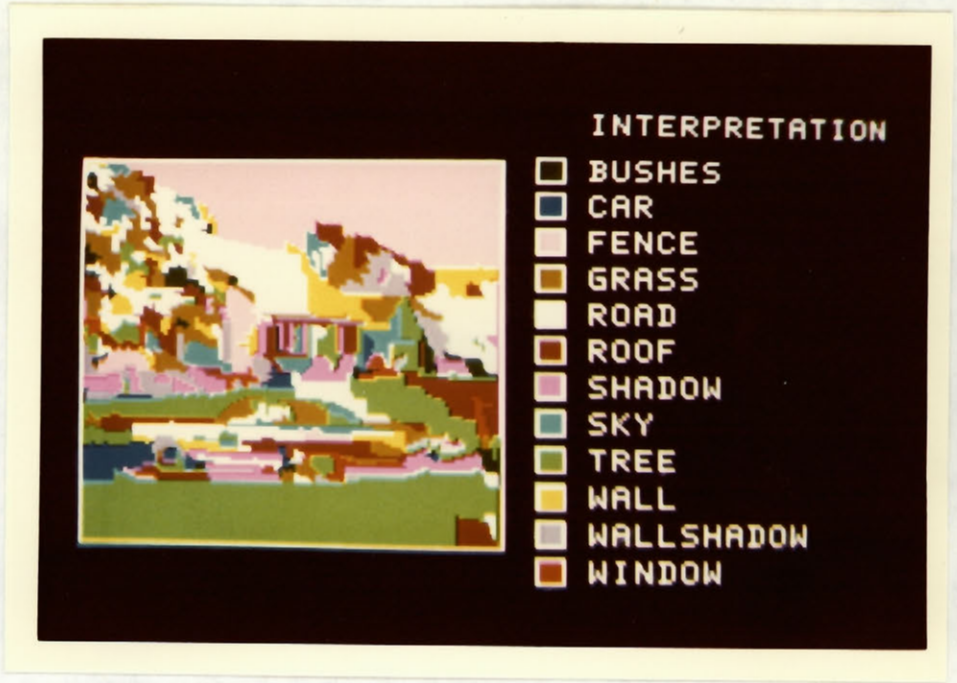


Figure 6.7.b Initial second best interpretations and associated confidences when colour and position are used as distinguishing features with equal weight and no veto.



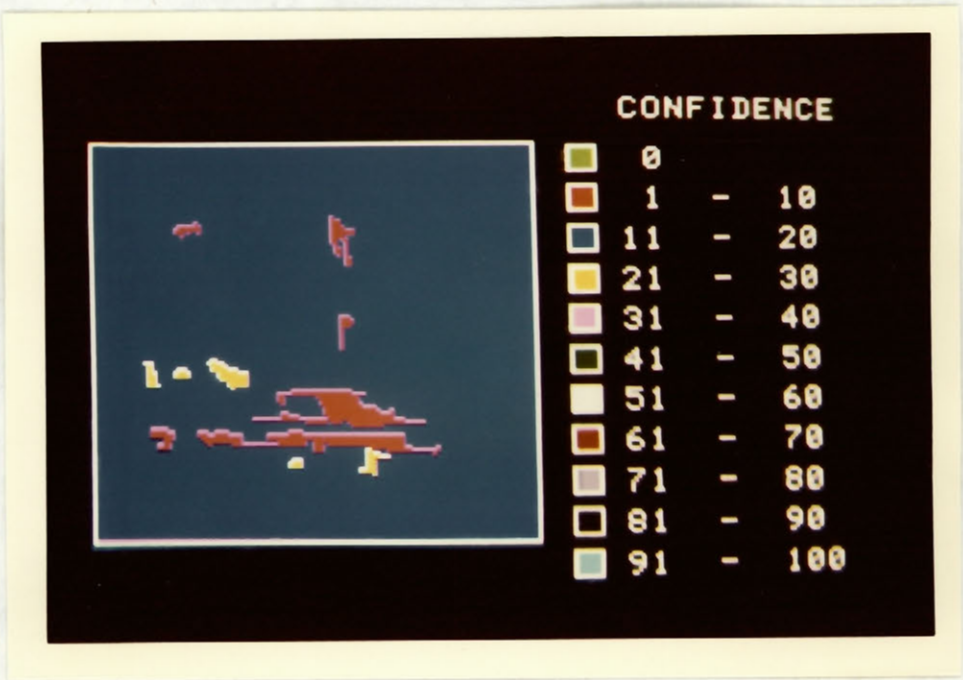


Figure 6.7.c Initial third best interpretations and associated confidences when colour and position are used as distinguishing features with equal weight and no veto.

and no veto power was assigned to any of them. It was noted that introducing the position as one of the distinguishing features provided a little bias (in comparison with figure 6.6), such that the FENCE interpretation given to regions at the top, and the SKY interpretation for regions at the bottom, were given lower confidences. Nevertheless, we still have many unrealistic interpretations. In general, we have found that assigning the veto to the vertical position feature eliminates most of these unrealistic interpretations, which in effect reduces the number of interpretations needed to ultimately interpret the scene under consideration.

We repeated the initialization process using the colour and position as distinguishing features, as shown in the relations in Appendix II, but we also increased the acceptable range of variation of the SKY region by 50% for the colour features and by 100% for the vertical position feature. The results are shown in figure 6.8 where it is noted that the SKY interpretation was assigned to more regions with higher confidences than figure 6.5. In order to perform such an experiment, the user must only change the value of the acceptable range of variation of the object SKY in the relation OBJECT FEATURES to the required values.



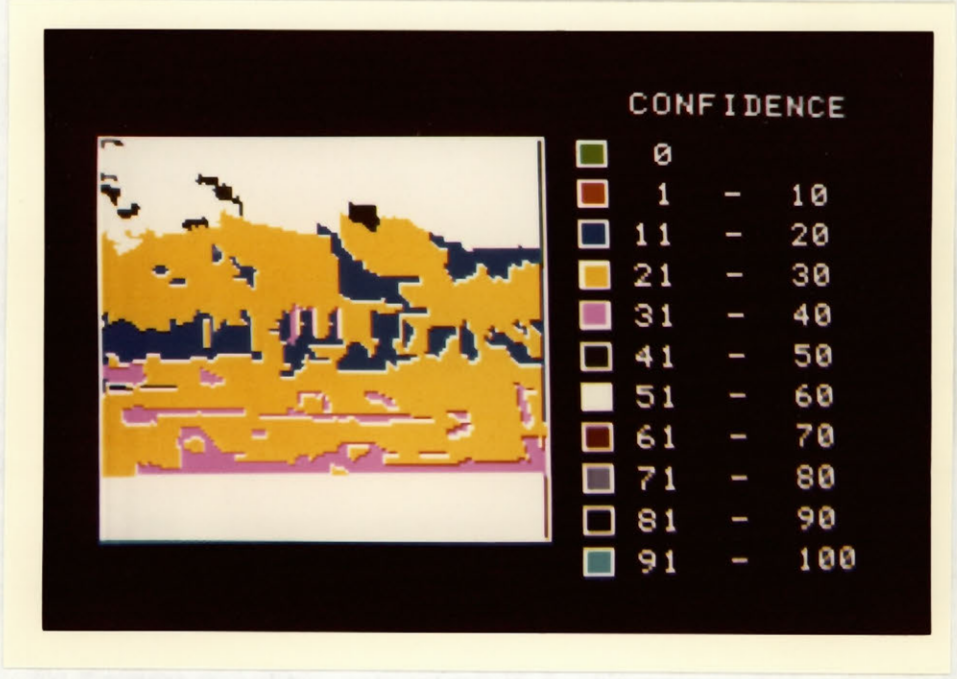
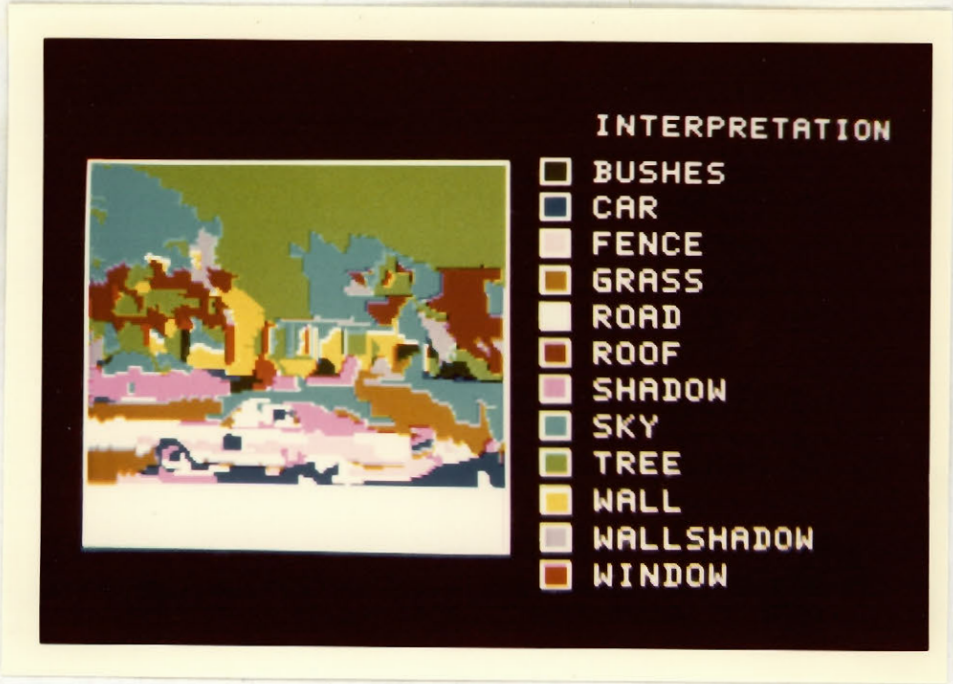


Figure 6.8.a Initial first best interpretations and associated confidences with relaxed SKY features.



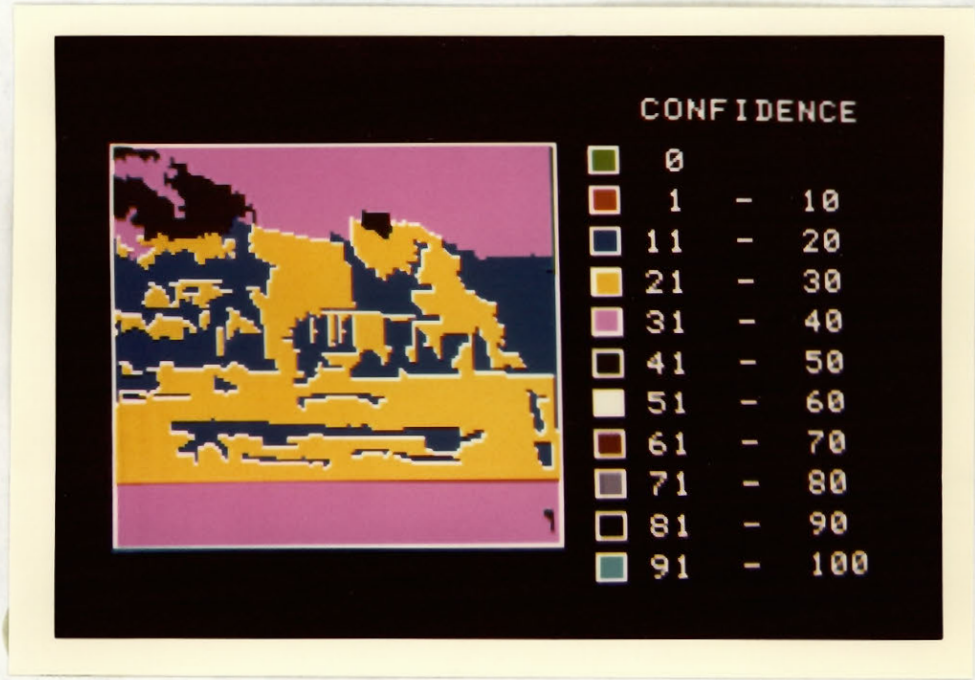
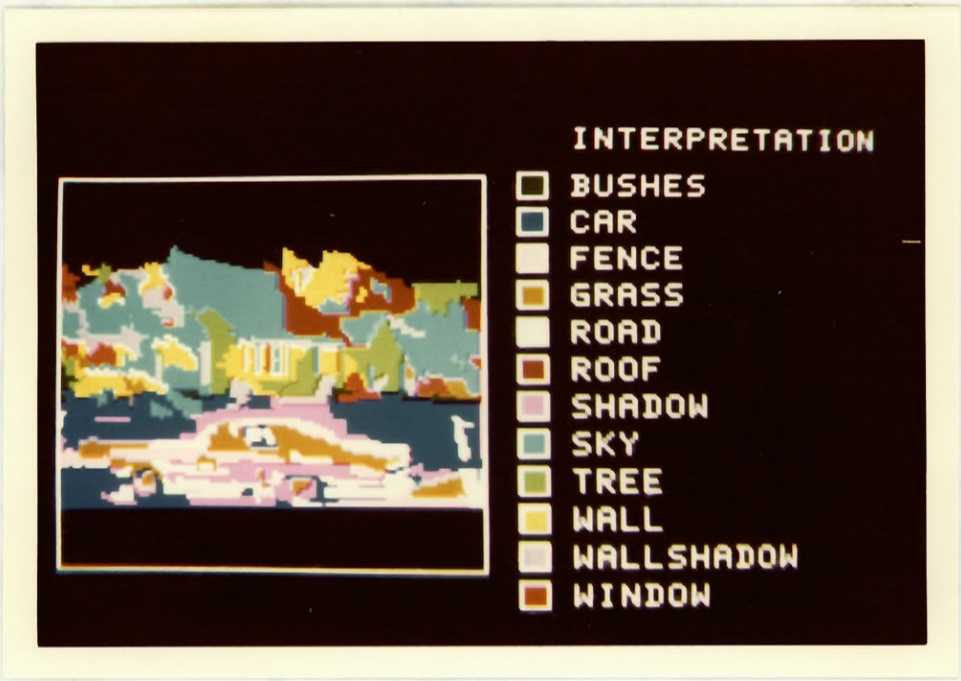


Figure 6.8.b Initial second best interpretations and associated confidences with relaxed SKY features.



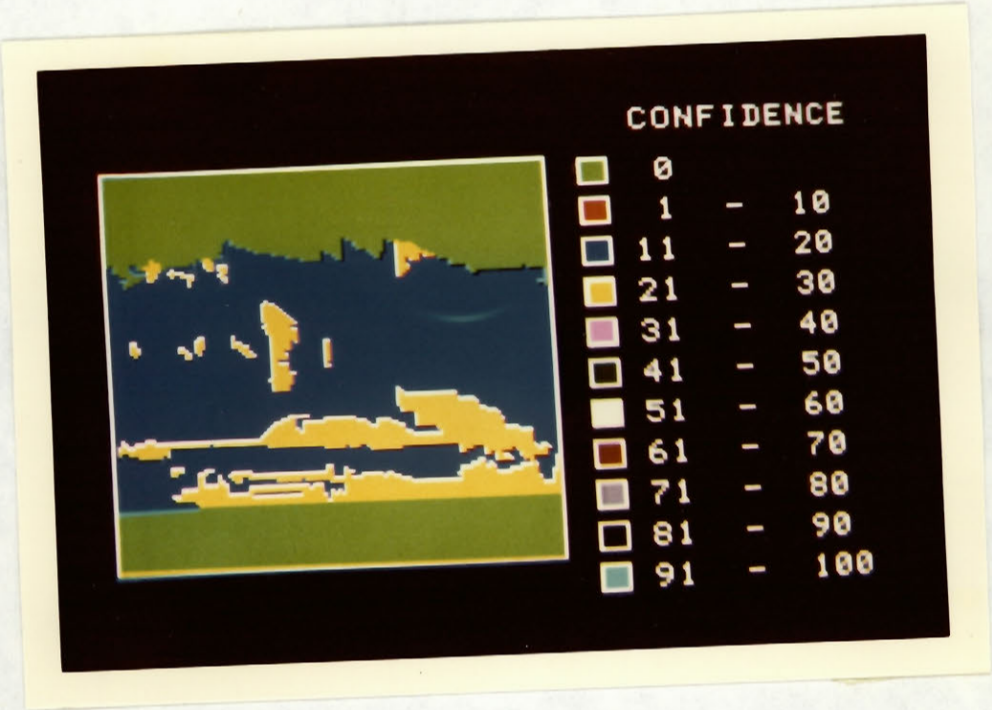


Figure 6.8.c Initial third best interpretations and associated confidences with relaxed SKY features.

This demonstrates the experimental capabilities of the system and allows for incremental model building. In this way, we can start with a very crude model for the class of scenes under analysis and proceed to test the different values for the features and their acceptable range of variation. As we vary these parameters, the initial set of hypotheses will change. The idea is to find the set of parameters that can be used to describe the general class of scenes.

In conclusion, the use of the distinguishing features and the veto power have provided a powerful tool for limiting the number of initial interpretations assigned to the regions. To change the model or to build a new model, the user must only make the necessary changes in the LTM relations defined.

6.4 OVERALL SYSTEM PERFORMANCE

Experiments with real pictures have been performed on outdoor scenes (typically figure 6.1). This picture is indeed quite difficult to analyze since it contains sections which are often ambiguous to the human observer. Many areas of the picture are occluded by objects and a large degree of ambiguity is introduced by shadows, for example, those

appearing on the walls of the house or around the car. We have also previously mentioned that the system presented in this thesis is designed such that it begins with a partial segmentation and attempts to produce a labelled complete segmentation as a final result. Because of this assumption, the analysis is always proceeding on the basis of a very fragmented picture. Therefore, no use was made of the objects' geometric shape or other such properties that refer to a complete segmentation. Thus, the type of constraints that are used are restricted to those which are meaningful to segmented parts of objects. Some constraints employ the fact that the colour of an object does not change abruptly. Two adjacent regions can then be interpreted as the same object only if the difference between the average value of each of the RED, GREEN and BLUE features of the two regions is less than some threshold (e.g see CONSTRAINT relation in Appendix II). Other constraints use structural relationships to govern the conditions under which two adjacent regions can be interpreted as two different objects as seen in relation CONSTRAINTS.

The objects appearing in the current analysis are: BUSHES, CAR, FENCE, GRASS, ROAD, ROOF, SHADOW, SKY, TREE, WALL, WALL-SHADOW and WINDOW. We should again point out that SHADOW and WALL-SHADOW were introduced to avoid the problems associated with shadows. Both might conceivably be

removed if a successful SHADOW ANALYZER were developed.

Spatial constraints that restrict the relative position of two adjacent objects proved extremely useful in eliminating or de-emphasizing certain arrangements of neighbouring regions. For example, consider the constraints that involve ROOF as one object. As shown in Appendix II, a ROOF segment can be ADJACENT to SKY, TREE, WALL, WALL-SHADOW (WALL_SH.) or another ROOF segment. It must also be ABOVE the WALL or WALL-SHADOW, BELOW the SKY and not ABOVE the TREE. In addition, it must differ in at least one colour feature by a specified threshold in order to be ADJACENT to any of the other objects. However, the difference in each of the three colour features must be below a threshold in order that it be ADJACENT to another ROOF segment.

Summarizing the initial stage of the analysis for the input picture (figure 6.1), we note that we used a low level segmentation threshold value h equal to 10 and clean-up parameter equal to 8. Beginning with a 128X128 image, the LOW LEVEL SEGMENTATION yielded a REGION MAP with 292 regions (figure 6.2.b). The HYPOTHESIS INITIALIZER generated a list of all possible interpretations and their associated confidences (figure 6.5) on the basis of the relations given in Appendix II.

The result of analyzing the picture using the initial interpretations described above are shown in figure 6.9[1]. As evident from the figure, the best interpretation for the large majority of the regions is correct. In addition most of the regions have acquired only one interpretation. This can easily be seen by observing that the confidence level of the second best interpretation for most of the regions is zero (figure 6.9.b). Many regions started with incorrect first and second (even some the third) interpretations; however, the final interpretations are ultimately correct. Others begin with all of their interpretations being equally confident, and through the iterative process which involved their neighbouring regions, are eventually correctly interpreted. As example, we note that none of the regions has its best initial interpretation as WINDOW; however, the final result in figure 6.9.a contains the correct labels for most of the WINDOW regions (we have considered the WINDOW-SHUTTERS as WINDOWS).

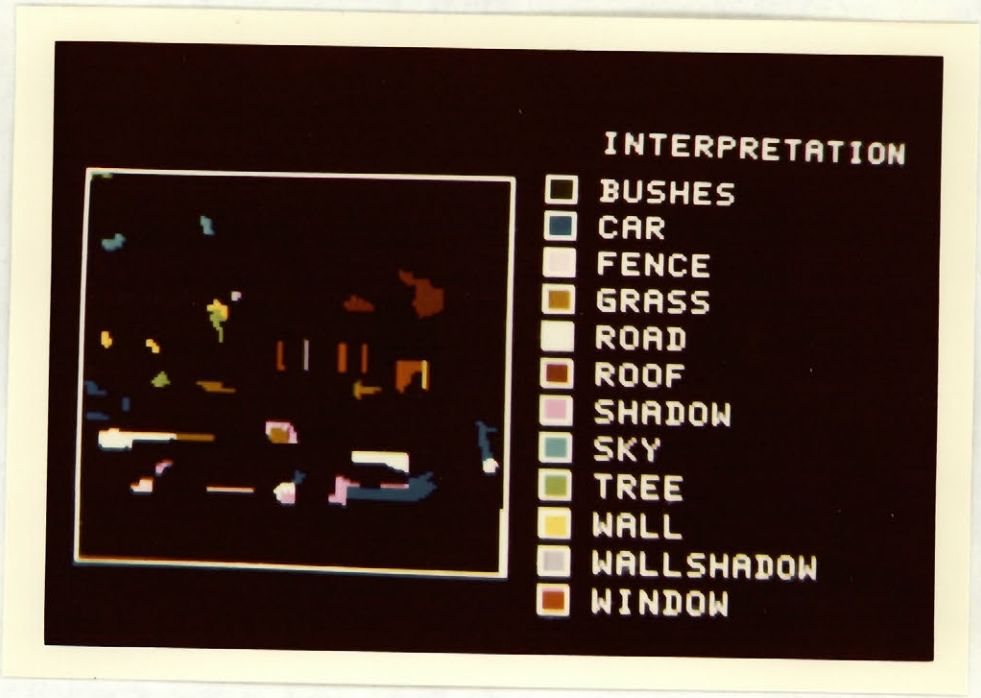
---

[1] Equations 5.14-5.16 in Chapter V were used in the process and the queue length was set equal to 100.





Figure 6.9.a Final first best interpretations and associated confidences using the initialization of figure 6.5.



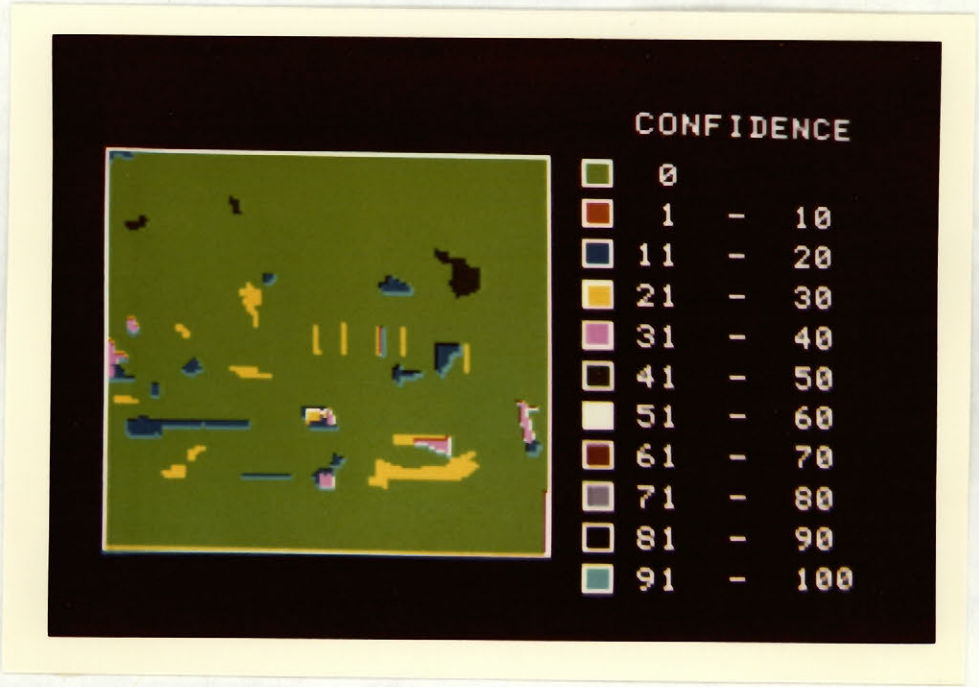
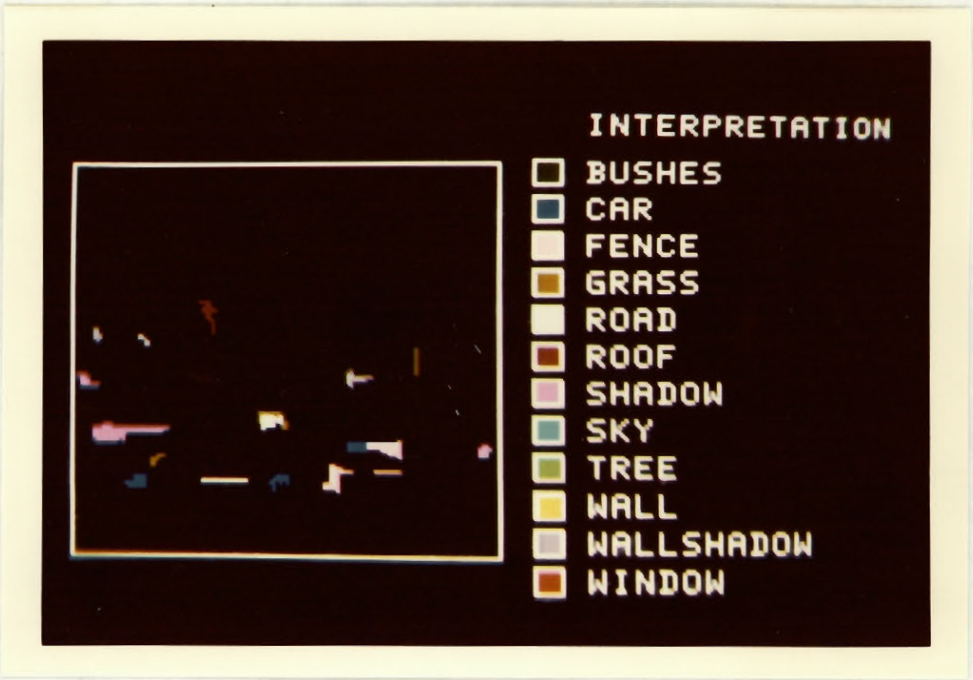


Figure 6.9.b Final second best interpretations and associated confidences using the initialization of figure 6.5.



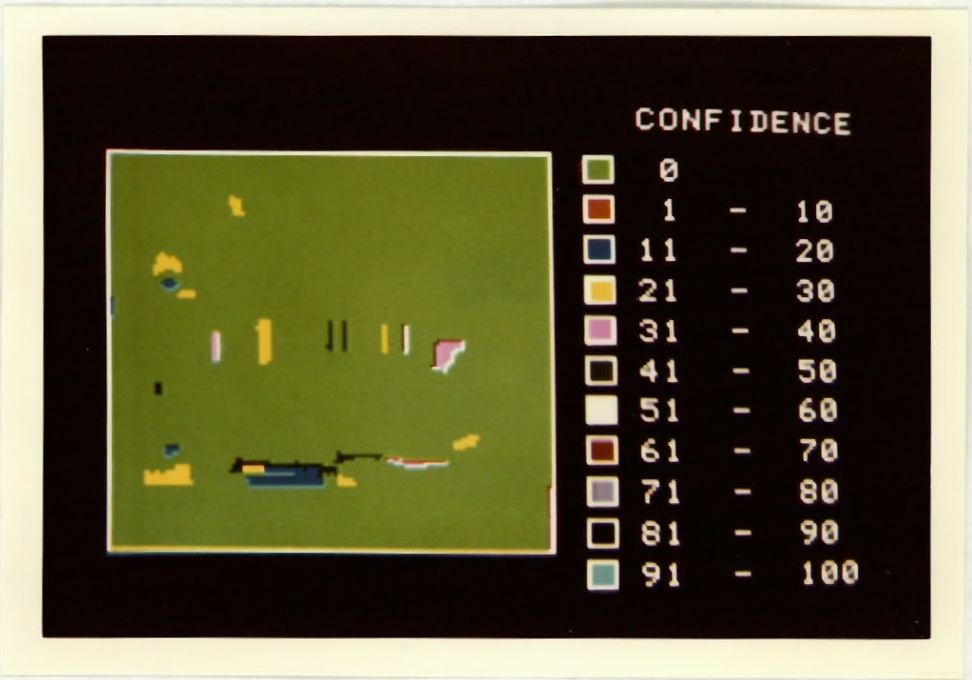


Figure 6.9.c Final third best interpretations and associated confidences using the initialization of figur 6.5.

There are a few minor errors (regions that have incorrect final interpretations) as seen in the final result, but on the whole the segmentation and labelling is quite satisfactory. The TREE to the left of the picture contains small regions that were erroneously labelled as ROOF and there is a small patch of GRASS in the middle of the CAR. The first ambiguity is quite understandable, given the complexity of that part of the picture, which can be best characterized as a rough textured area with tree leaves, shadows and walls, each not very distinct from the others. This error was not avoided by the constraint that a ROOF segment cannot be completely contained by a TREE segment because the confidence of these regions interpreted as ROOF reached the value of 100% long before the large region that contains them was formed.

In general we observe that the resulting errors are minor and represent less than 3% of the area of the picture. These errors result from regions that are confused by their fragmented neighbours in that the latter interpretations provide conflicting evidence. They may be eliminated by reapplying the analysis to a new REGION MAP as will be described later. We also observe that the ROCK-WALL near the car was not included in the LTM because of the difficulty of describing it in terms of a generalized model. It was therefore labelled as GRASS. Examination of the

original picture (figure 6.1) shows that these rocks could easily have been interpreted as part of the CAR. Moreover, the highlights appearing on the CAR (e.g front bumper or rear fender) caused small areas to be interpreted as SHADOW. This happens because highlights were not included in the model. The large differences between the colour features of these regions and their local neighbours prevented them from being interpreted as a CAR. However, the analysis was able to arrive at the only conceivable interpretation which was SHADOW. This proves its ability to perform even with a partial model, in the following sense. If an object that was not described in the model should appear in the scene under study, the system attempts to interpret it with the current available knowledge as best as it can. On the other hand, the white square area on the side of the CAR was interpreted correctly as a CAR. It was merged with its directly adjacent neighbours, because the differences in the colour features were less than the theshold values specified in the constraints.

The colours in natural scenes of the type under consideration are extremely unsaturated, clustering around the white reference point in the colour triangle (Kender [57] and Nazif [78]). This makes it difficult to distinguish such objects as TREE, SKY and GRASS on the basis of absolute colour. As stated in Section 4.2, these

absolute features were only used for initialization; relative values or colour differences were employed for constraint satisfaction. The use of semantic information in the form of spatial constraints helps to restrict and direct the analysis to the correct interpretations. Finally, it is worth observing that occlusion may also cause problems; for example, not all of the windows of the house which are behind the tree have been found.

During the previous analysis of the picture, the queue length was set to 100 and all of the adjacent regions' interpretations were used in the updating process, (equations 5.13-5.16). Begining with a REGION MAP that contained 292 regions, the system produced a new labelled REGION MAP with only 71 regions. These were formed when adjacent regions having the same interpretations with 100% confidence were merged during the analysis. The latter required about 2600 region verifications. The new region boundaries are shown in figure 6.10. On the average, during the main part of the analysis, this means that each region was updated about nine times. Some regions need only 2-4 verifications, while others need about 10-15. Of course, regions with satisfactory interpretations (i.e. the confidence in the best interpretation is above a threshold, 93% in our case) are set to 100% so that towards the end, the system concentrates on improving the results for a small

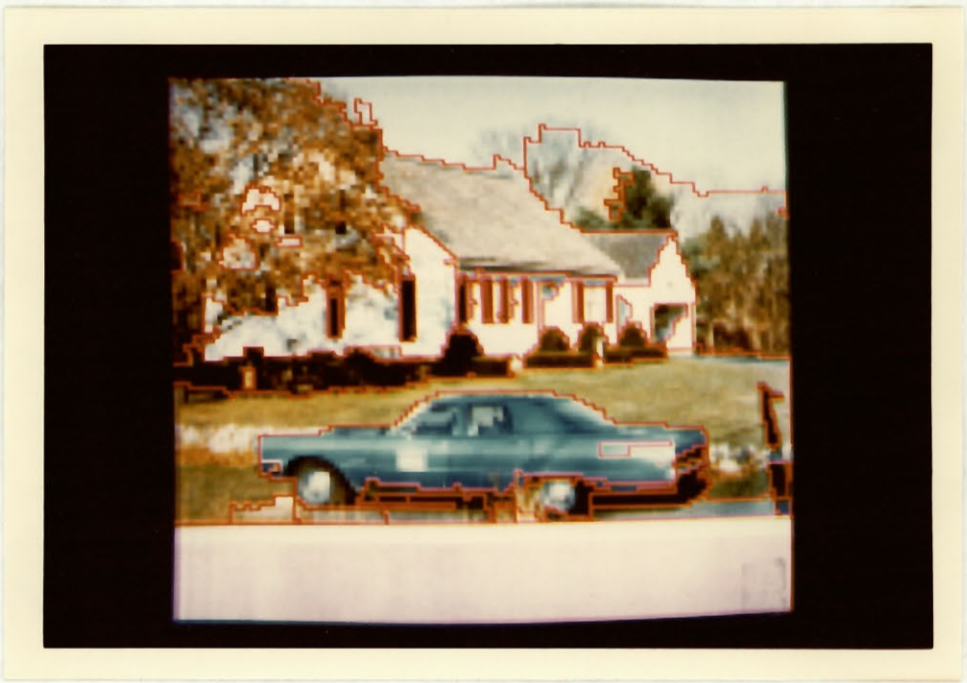
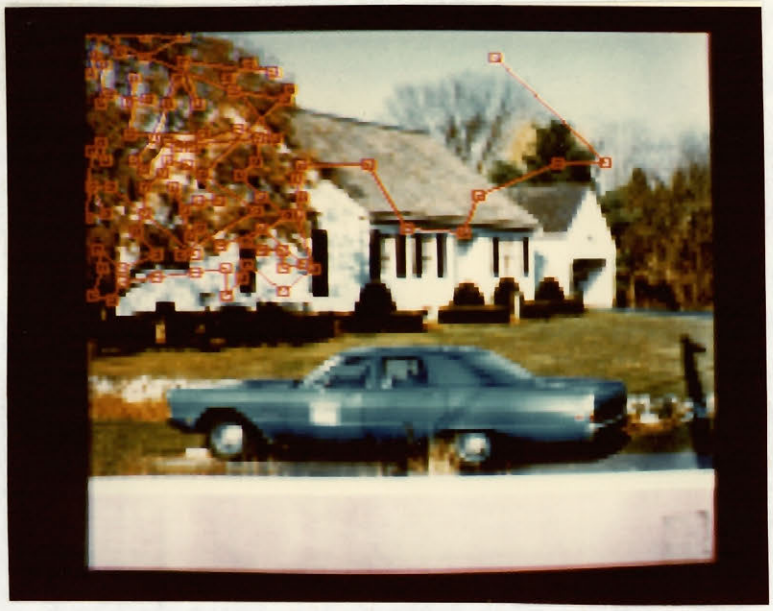


Figure 6.10 Region boundaries after merging as a result of figure 6.9. The number of regions is 71.

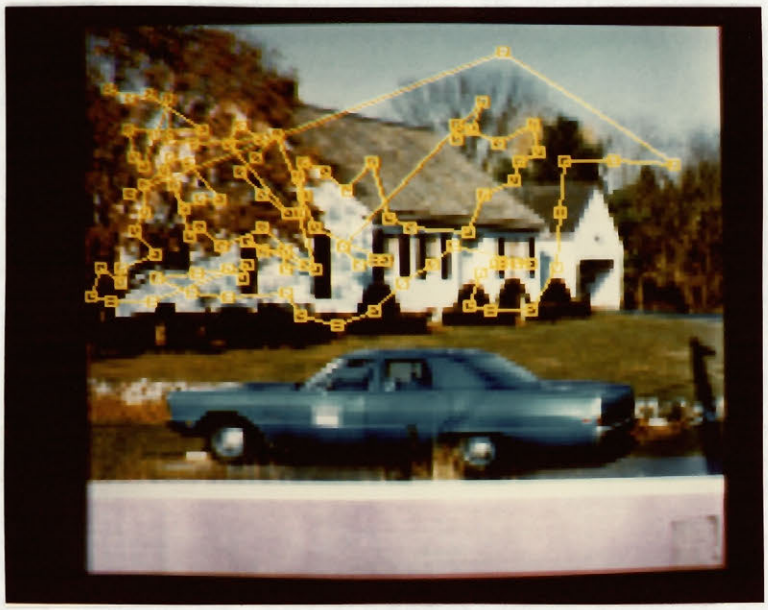
subset of the results in the picture. Figure 6.11 shows the order of the regions' verifications, where each region is represented by a square centered around the centroid of the region. The path over 300 region verifications is shown. As has been described, in section 5.6, the system verifies the interpretation of the regions that have the lowest confidence and which are not in the QUEUE (worst first strategy). The total DEC PDP-15 time required for one region verification is about 18 seconds. Preliminary experiments with the system on a DEC VAX11/780 points to a reduction in the time by a factor of at least 20. An even greater reduction may be expected since the PDP-15 memory is restricted to 32K words, most of which is dedicated to the program rather than the data. Therefore a significant percentage of the 18 seconds per region verification is actually attributed to disc access time. The total time could also be significantly reduced by using equation 5.22 instead of 5.16.

To study the effect of initially using a complete segmentation in the process, we started the analysis using the REGION MAP that resulted from the previous analysis. It contains 71 regions and is shown in figure 6.10. Using the same initialization model, the HYPOTHESIS INITIALIZER was then invoked to provide a new set of initial interpretations and associated confidences for all of the regions, as shown



Number of Verifications

1 - 100



101 - 200



201 - 300

Figure 6.11 Illustration of the PATH for region verifications.

in figure 6.12. When the analysis was reapplied to the initial interpretations of this new REGION MAP, it resulted in the final interpretations shown in figure 6.13. In this case, because the number of regions was only 71, the queue length was set to 30, and again all of the adjacent region interpretations were used for updating. Some of the errors produced, when the original REGION MAP with the 292 regions was used, were now eliminated. For example, the constraint that a ROOF cannot be completely enclosed by a TREE eliminated the errors at the left seen in figure 6.9. This second analysis was performed to ensure the ability of the constraints to cope with a complete segmentation as well as a partial segmentation. It was also necessary to check the validity of the constraints that were possibly not used in the previous analysis. In this stage, the system performed about 700 region verifications, thus averaging about 10 verifications for each region in the new REGION MAP. The new REGION MAP contains 58 regions as shown in figure 6.14.

6.5 EXPERIMENTS WITH DIFFERENT VERIFICATION STRATEGIES

The system described in this thesis was initially tested and verified using a fictitous picture. A REGION MAP

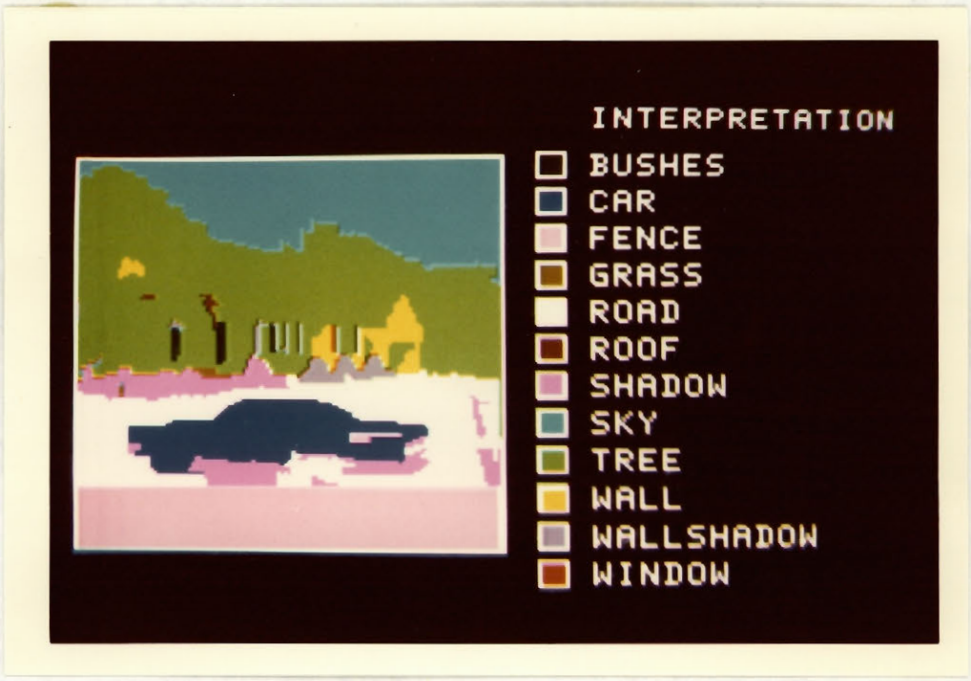




Figure 6.12.a Initial first best interpretations and associated confidences for the new REGION MAP in figure 6.10.



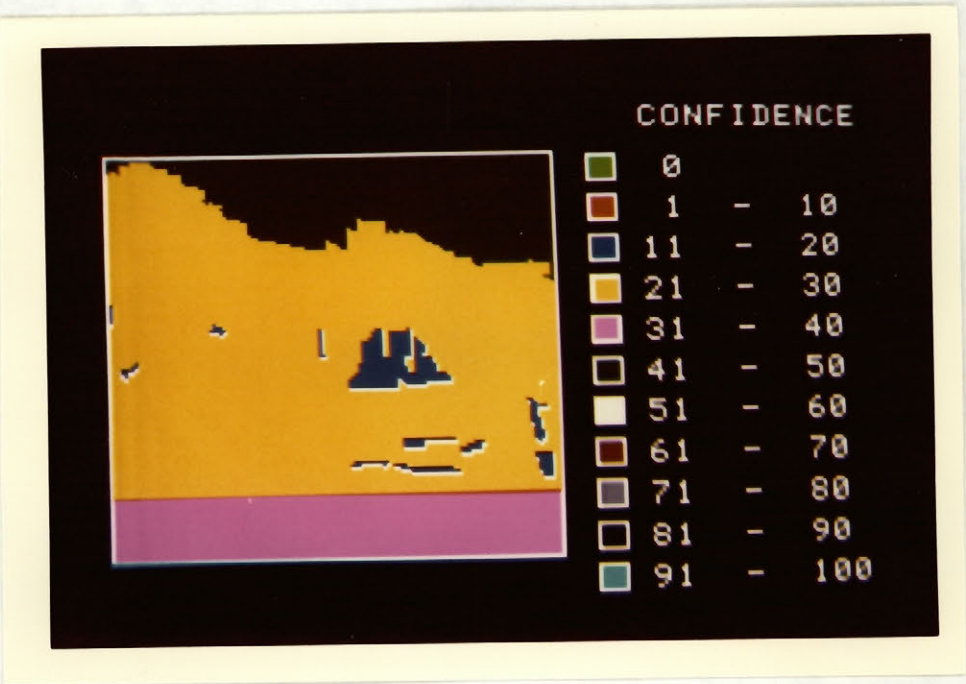
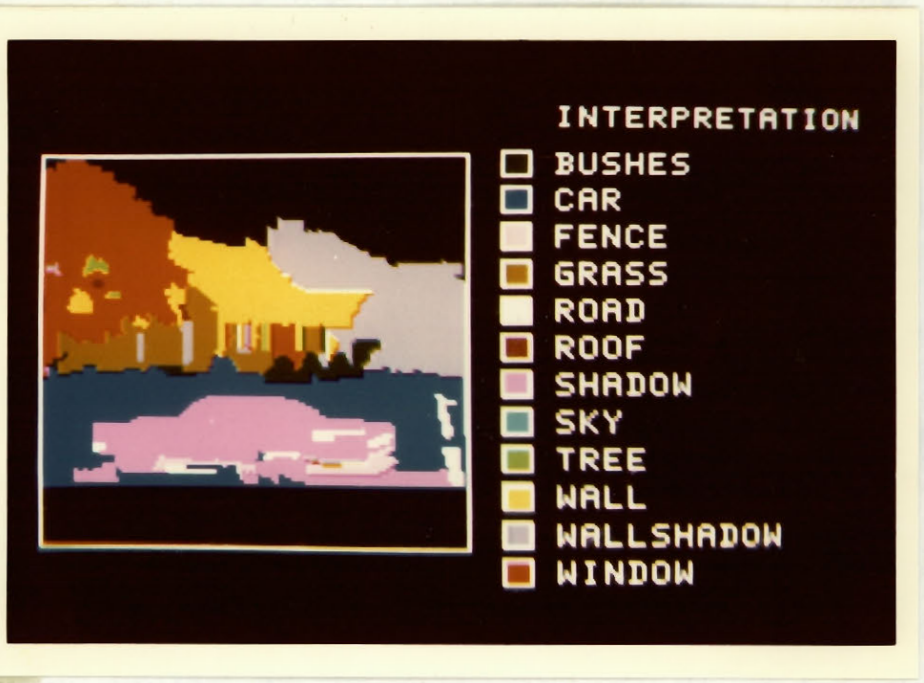


Figure 6.12.b Initial second best interpretations and associated confidences for the new REGION MAP in figure 6.10.



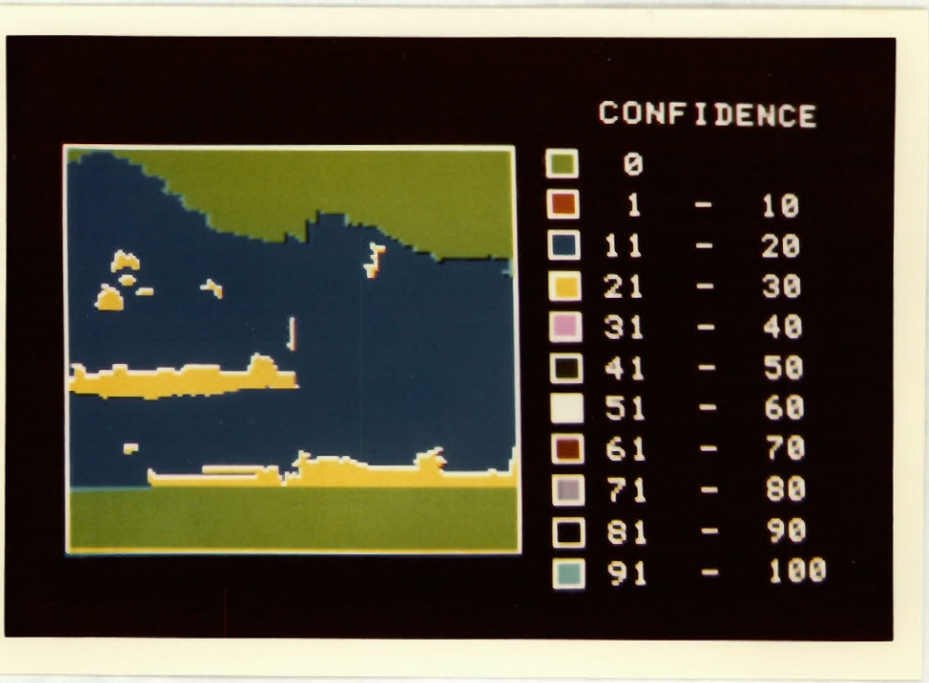
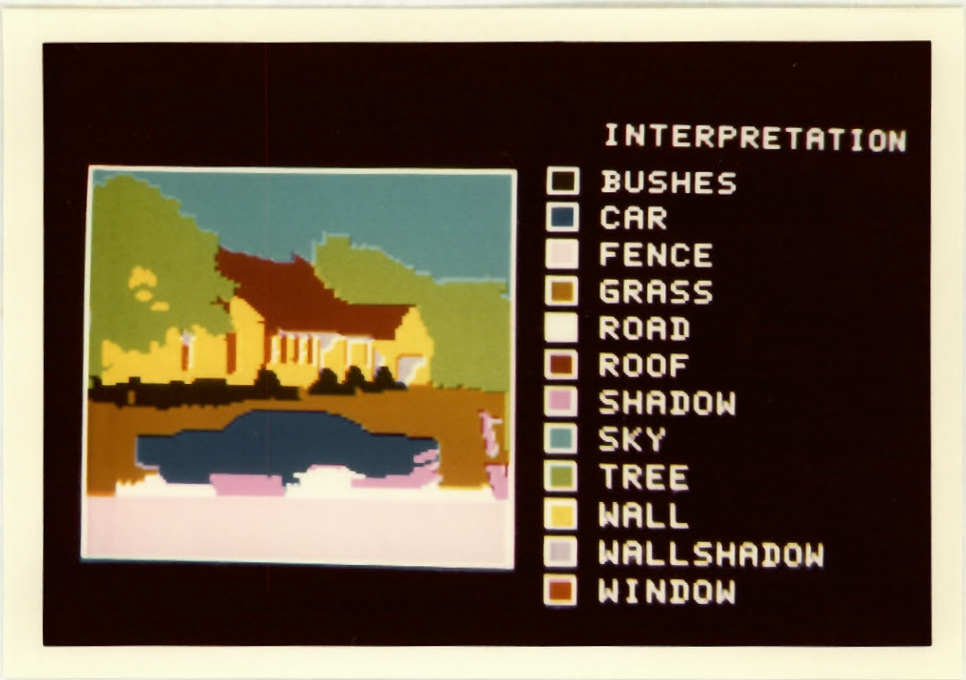


Figure 6.12.c Initial third best interpretations and associated confidences for the new REGION MAP in figure 6.10.



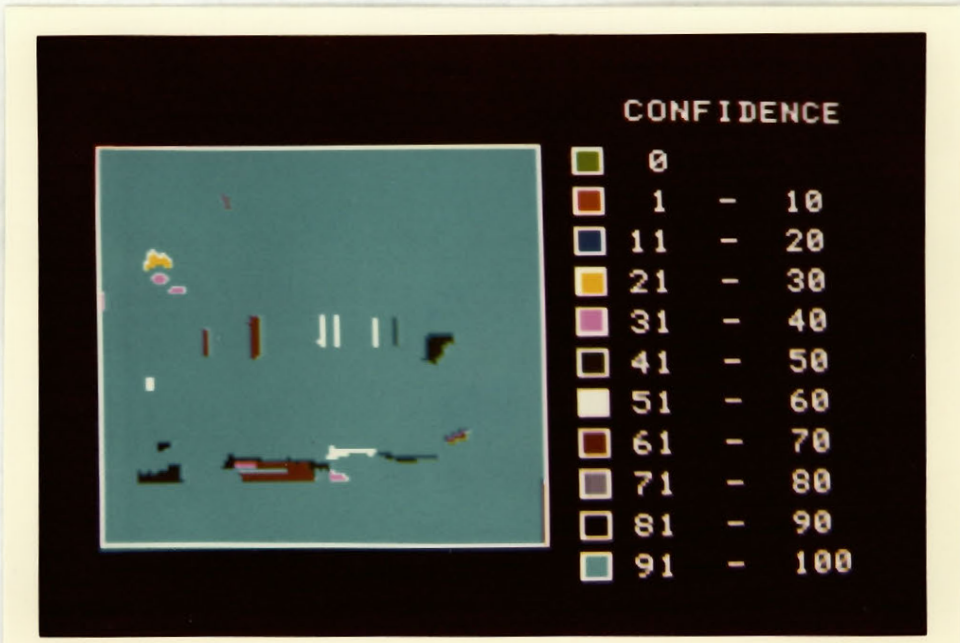
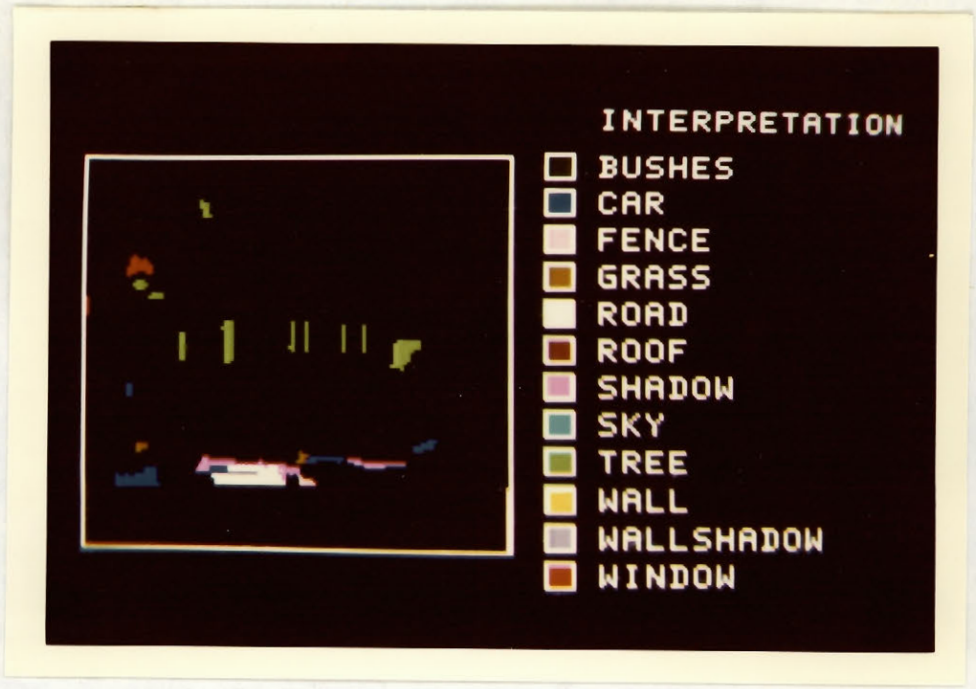


Figure 6.13.a Final first best interpretations and associated confidences using the initialization of figure 6.12.



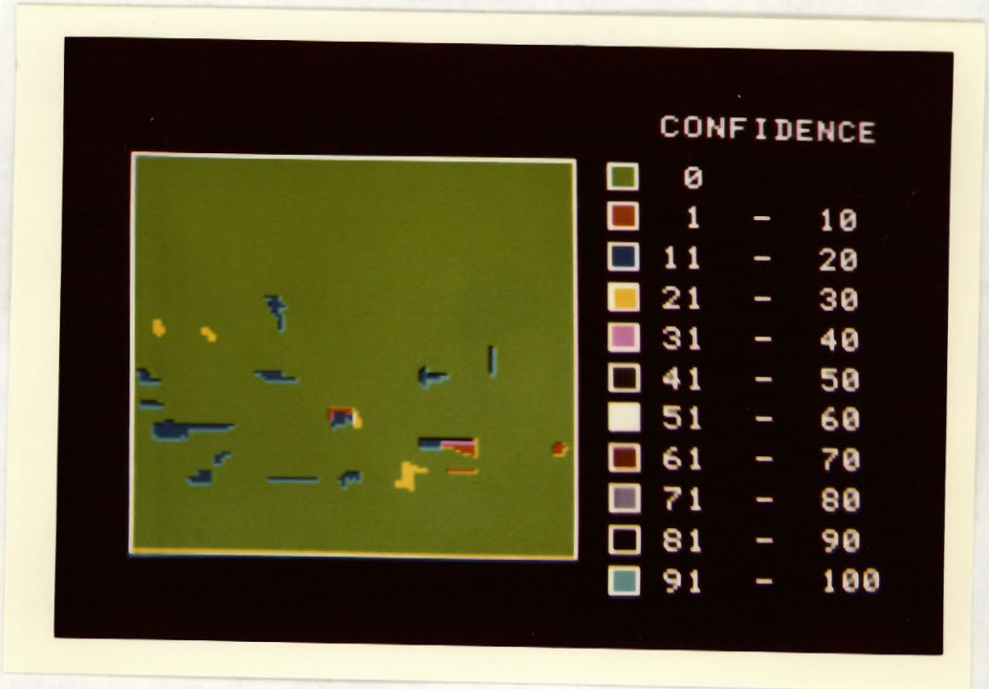
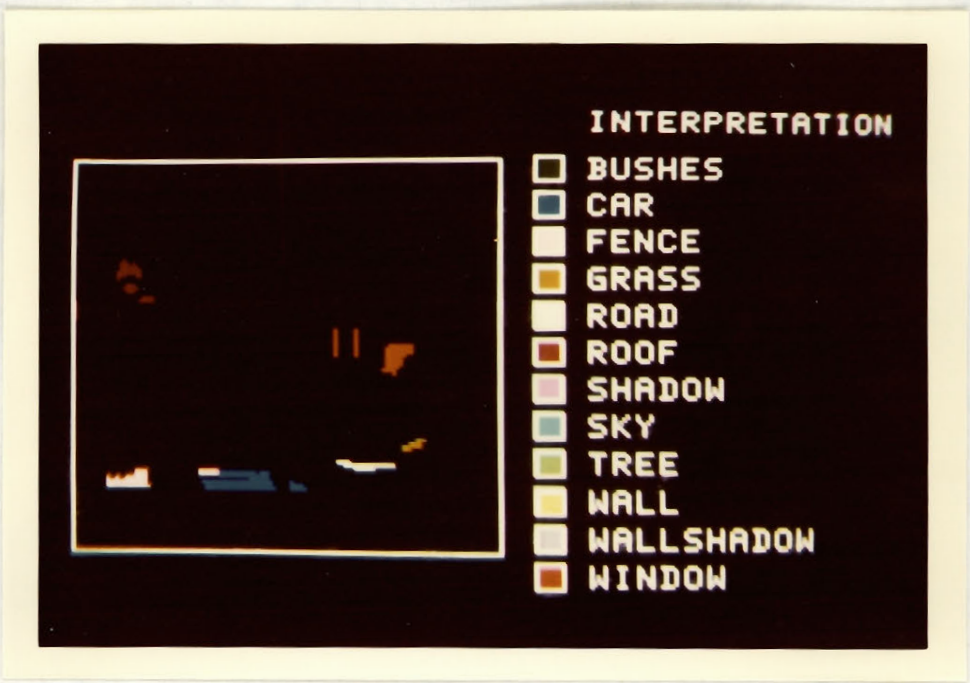


Figure 6.13.b Final second best interpretations and associated confidences using the initialization of figure 6.12.



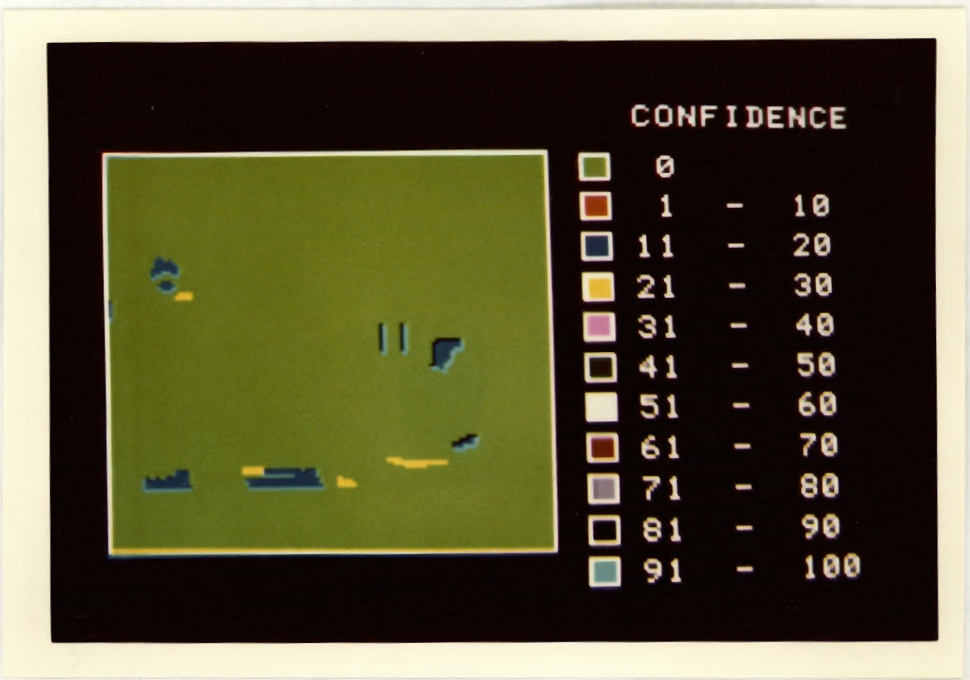


Figure 6.13.c Final third best interpretations and associated confidences using the initialization of figure 6.12.



Figure 6.14 The resulting region boundaries after the analysis. The number of regions is equal to 58.

that consists of 36 regions was manually generated (figure 6.15). This picture was chosen intentionally to be simple so that the convergence properties could be subjectively studied for different processing startigies. It was first used to test the correctness of the computer code of the different processes in the computer vision system. Then we used it to study the effects of the queue length on the system's performance. As shown in figure 6.16, the objects in the picture are: CLOUDS, DOOR, GRASS, ROOF, SKY, TREE, WALL and WINDOW. In interpreting this picture, we employed a set of constraints similar to the ones shown in Appendix II. The initial region interpretations and associated confidences for these experiments are shown in figure 6.17, where only the first best three interpretations are shown.

We used the system to interpret this picture using different queue lengths. We also reconfigured the system so that the relaxation process could be applied as:

(i) a parallel iterative process, or

(ii) a sequential process.

A final interpretation and associated confidence is shown in figure 6.16. All the regions were interpreted with correct interpretations in all cases. However, they differ in the number of region verifications needed to reach a fixed point.

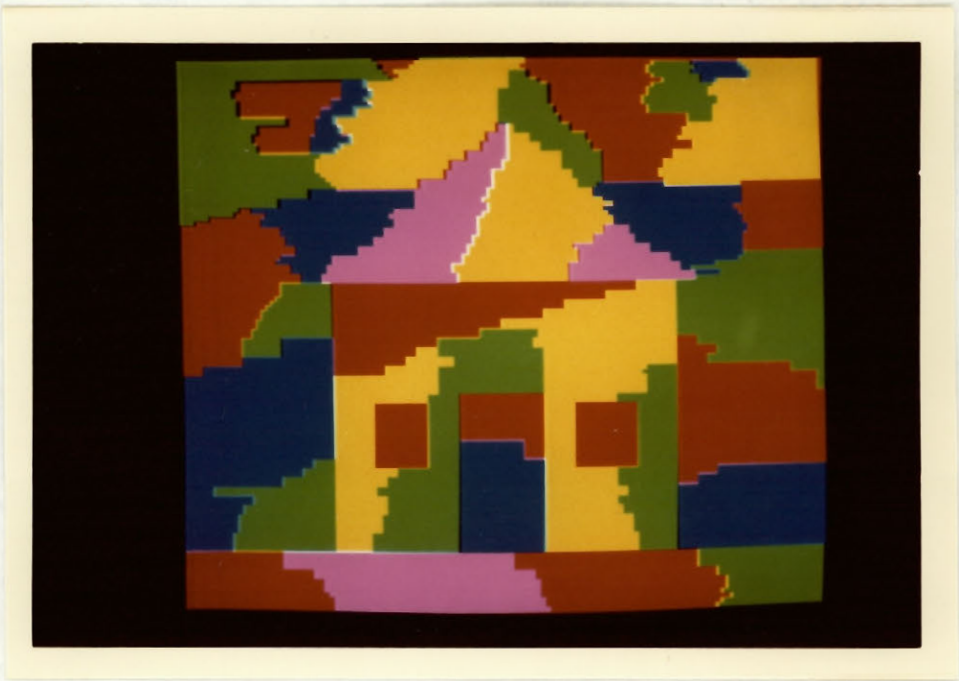
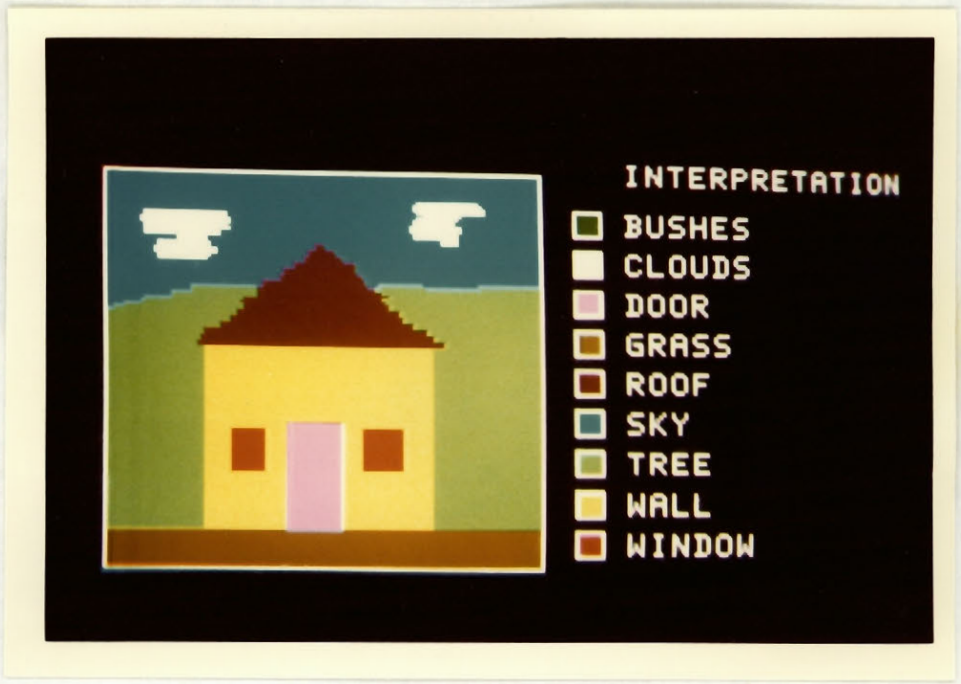


Figure 6.15 A manually created REGION MAP.



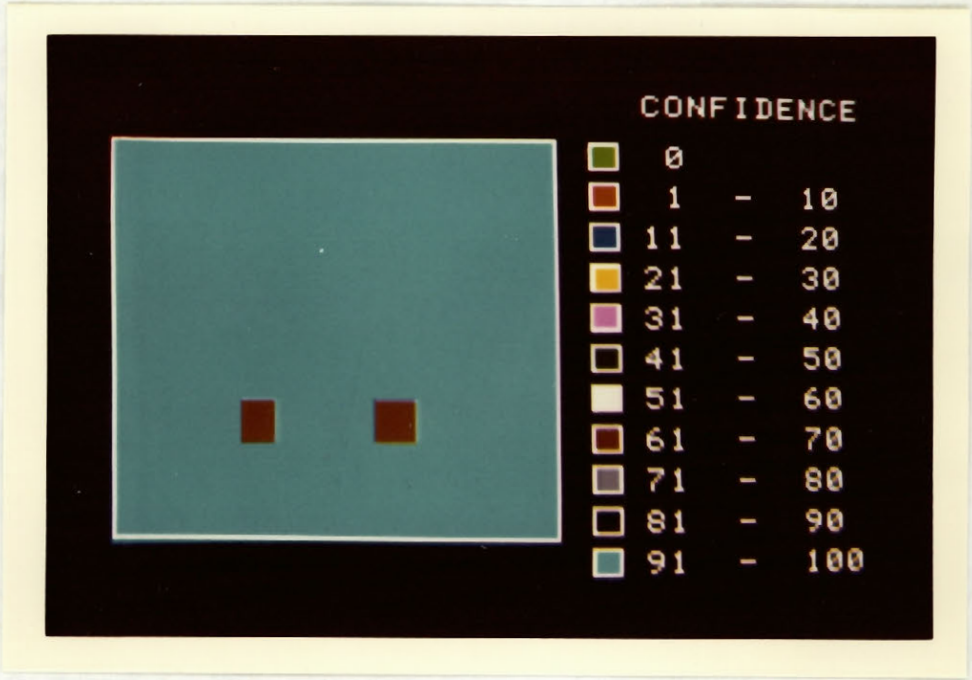
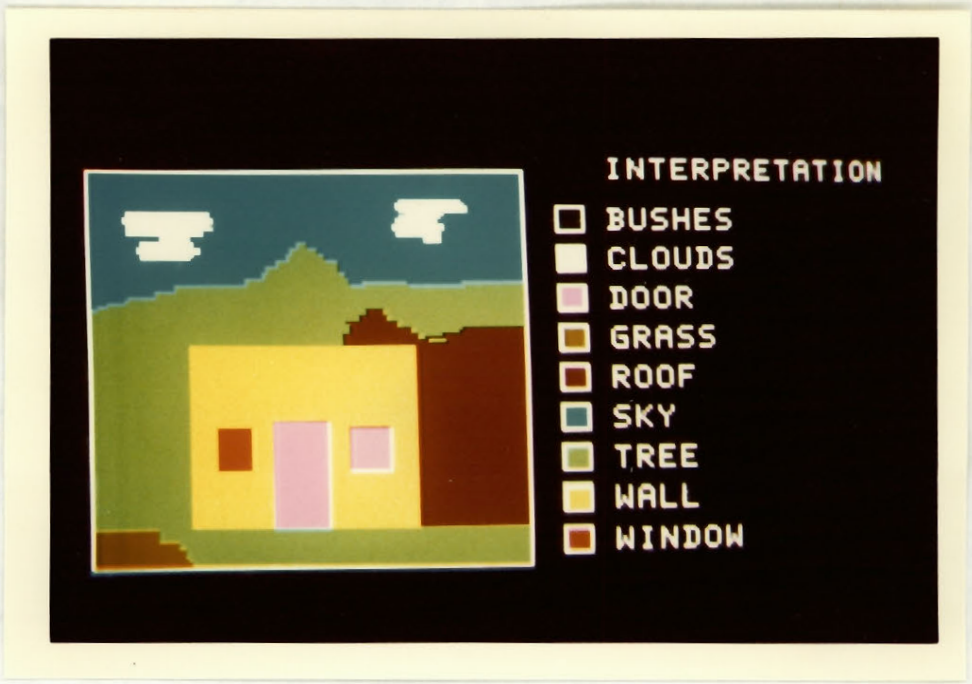


Figure 6.16 Final interpretations and associated confidences for the example in figure 6.15.



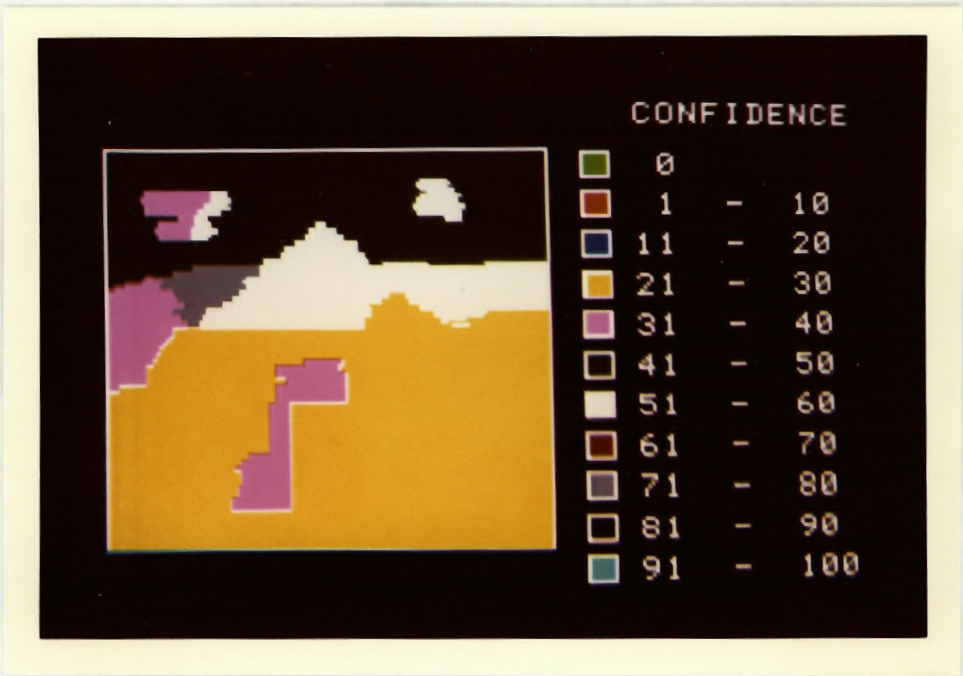
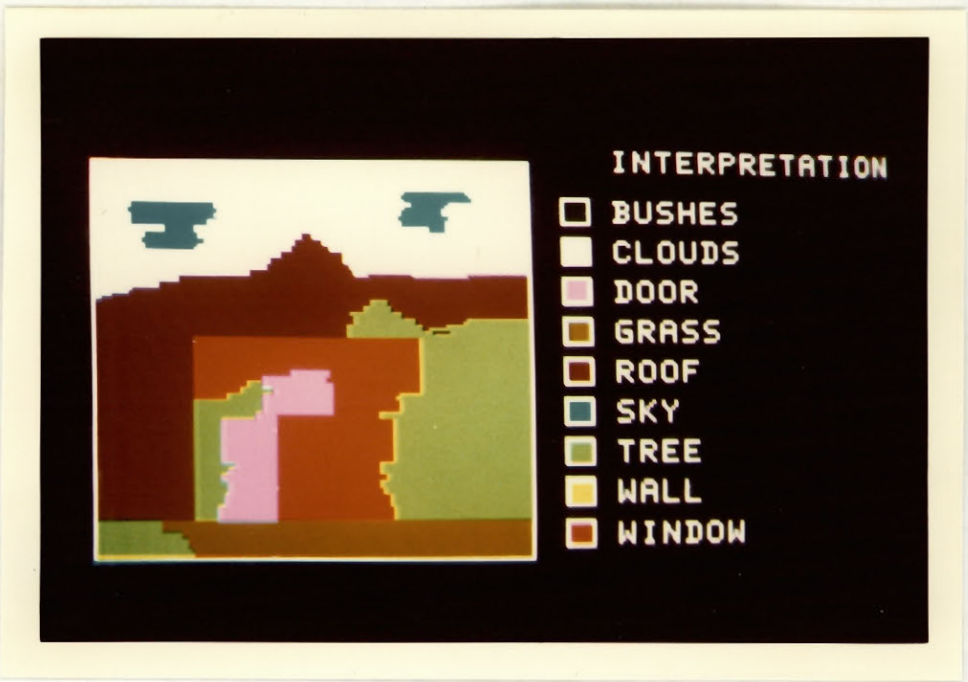


Figure 6.17.a Initial first best interpretations and associated confidences for the REGION MAP in figure 6.15.



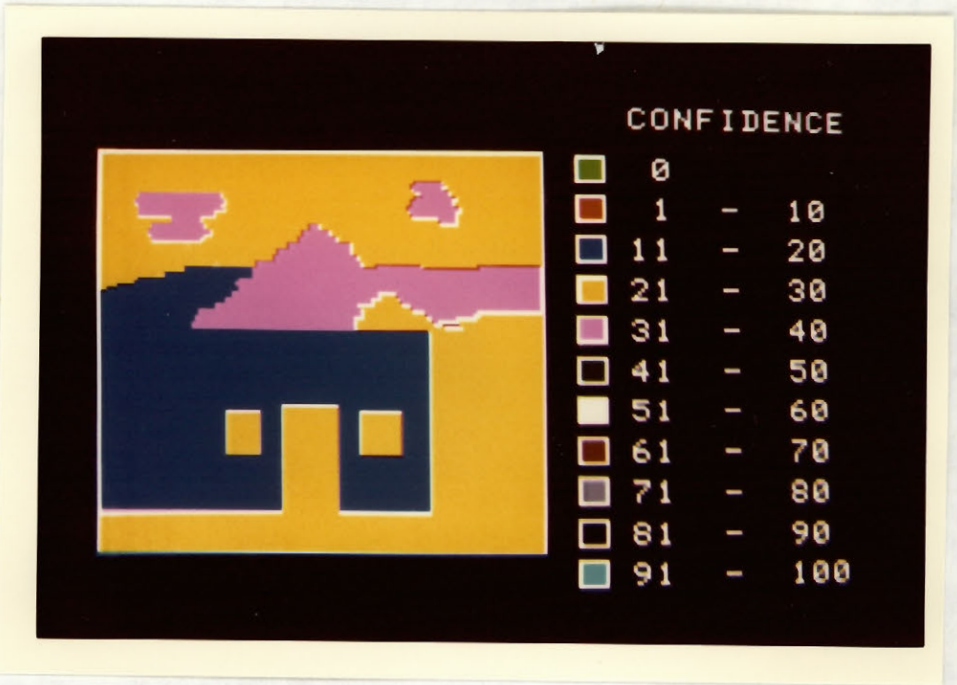
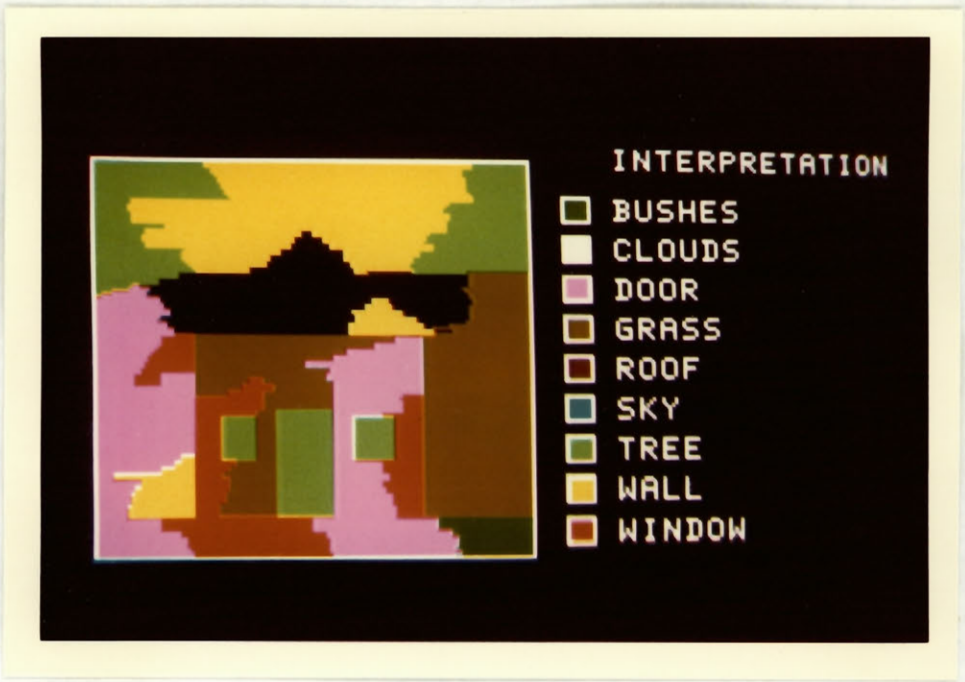


Figure 6.17.b Initial second best interpretations and associated confidences for the REGION MAP in figure 6.15.



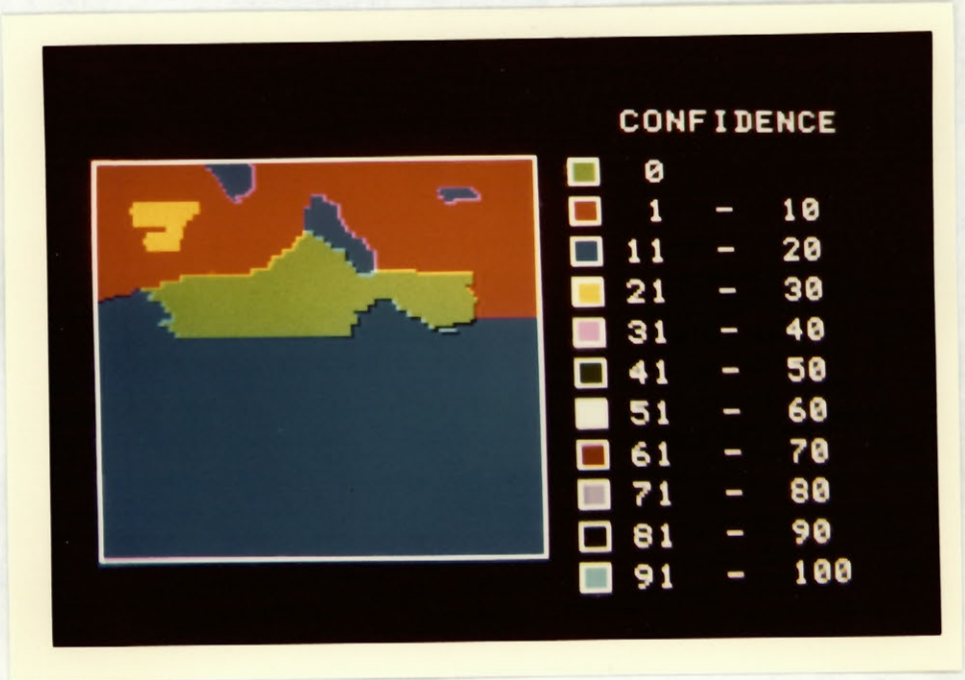


Figure 6.17.c Initial third best interpretations and associated confidences for the REGION MAP in figure 6.15.

To study the convergence properties of the system, we used the the entropy as an information measure [123, 132]. The information measure after T region verifications is defined as:

$$I(T) = -\sum_{i=1}^{M} \sum_{I_x \in \bar{I}} C_i(I_x) \; \text{Ln} \; C_i(I_x) \qquad (6.1)$$

where:

$C_i(I_x)$ is the current confidence of interpreting region R_i as object I_x,

$\bar{I}$ is the set of interpretations associated with region R_i, and

M is the number of regions in the picture.

Note that T represents the total number of times the HYPOTHESIS VERIFIER/INTERPRETER is activated.

First, the system is used to interpret the test picture with different queue lengths (e.g. 5,10,15,20). In the experiments, it was ncted that the queue length has very little effect on the number of regicn verifications needed to arrive at a fixed pcint. In all cases, all the region's interpretations were used in the updating equation (i.e. equations 5.14-16). The number of region verifications required was in the range 215-225 for the different queue lengths used. Figure 6.18 shows the variation of the information measure I(T) for two different queue lengths.

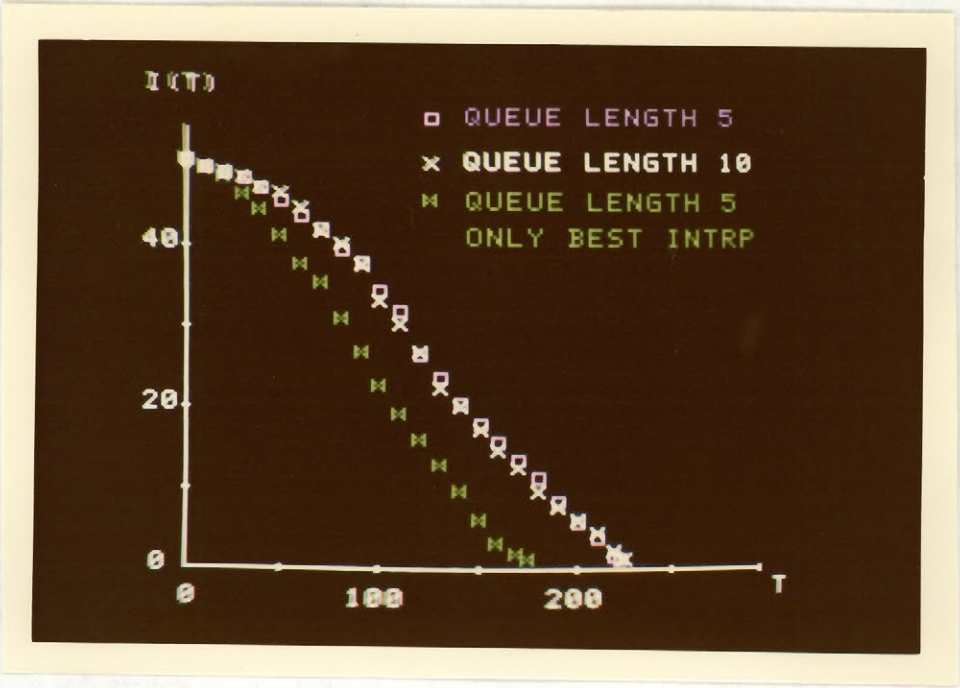


Figure 6.18 Variation of information measure I(T) for different queue lengths.

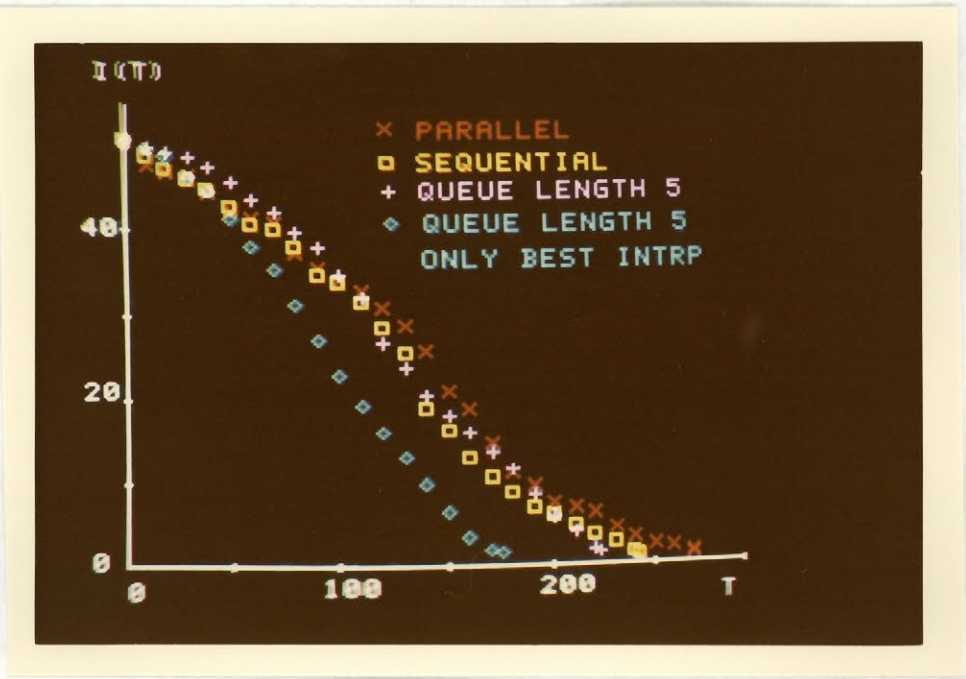


Figure 6.19 Variation of information measure I(T) by processing in parallel, sequentially and with queue length equal to 5.

In another experiment, we reconfigured the system so that the relaxation labelling process was performed in parallel. The FOCUS OF ATTENTION process was disabled and the confidences of all the regions were updated in parallel using equations 5.14-5.16. We then repeated the process so that the regions' confidences were updated sequentially (equations 5.14-5.16). In this case, the regions were selected and updated sequentially according to their region number as given in the REGION MAP. The information measure in both cases is shown in figure 6.19. We also repeated the process using the FOCUS OF ATTENTION as described before, but with the number of interpretations used in the updating process restricted to be one using equation 5.22 (i.e. only the best interpretations of the neighbouring regions were used in the updating equation). When compared with parallel processing, we note from figure 6.19 that sequential processing reduces the number of region verifications required from 275 to 240. On the other hand, when the worst first strategy with all of the adjacent regions' interpretations used in the updating process was employed, the number of region verifications was reduced to 220. When we used both the worst first strategy and restricted the number of interpretations in the updating process to one, the number of region verifications was further reduced to 170. We should here point out that in the last case, the time required for verification of one region is also reduced

by a factor of 1/Z where Z is the average number of interpretations for the regions in the neighbourhood of the region under consideration. In conclusion, we note that generally the worst first strategy reduces the number of region verifications needed to arrive at a fixed point. The savings that result from the sequential worst first strategy are in the range of (20-30)%.

6.6 CONCLUSIONS AND FUTURE RESEARCH

The computer vision system presented in this thesis is a new approach to the design and implementation of such complex systems. It is constructed of several independent processes that communicate only through a common channel (STM). The design of the STM as a relational database ensures the necessary degree of independence between the different processes of the system. This in turn enhances the structural flexibility of the system because a process can easily be removed or a new process added without affecting the others, as long as these processes are able to find the necessary information for their operation in the STM. A new process only requires a knowledge of the data model of the STM (the relational data model) and the communication language to and from the STM. This in turn facilitates the task of adding a new process because the

designer does not have to know any of the storage structures or the strategies of the other processes. This is extremely important, especially for such large systems which will be used by many people over a long span of time.

In this system, we have isolated the model information from the computer code that implements the algorithm. This was achieved by introducing the LTM in the form of a relational database. In turn, this has enhanced the experimentation capabilities of the system. The user need only change the LTM relations to modify and test his model. Thus, he can incrementally build his model by changing it and experimenting with the system. In our case, the major portion of the model information was specified before actually performing the analysis; however, certain trends and fine points were brought to the fore during experimentation, thereby pointing out the need for additional constraints. Thus constraints were incrementally added to the LTM in a learning phase, and hence the LTM structure provided a powerful tool for building and testing a world model. It also gives the experimenter an insight into the importance of the different contextual clues provided by the constraints. Note that although the system was tested on a picture of a real outdoor scene, it is designed as a general experimental tool and not restricted to this single class of scenes. The user need only define

the new model of any other class of scenes (e.g. indoor scenes), enter it into the LTM, and then perform experiments to ensure the validity of his visual model.

The system adopts the concepts of competition and cooperation as a basic paradigm for its vision processing strategy. This was implemented as a relaxation labelling process which was designed and implemented to interpret a partially segmented real image. The user has the option of implementing the relaxation process in parallel, sequentially, or in a sequential mode directed by the FOCUS OF ATTENTION strategy. The system is data directed in the sense that the path of control is determined by the state of the analysis. The FOCUS OF ATTENTION process enables the user to introduce additional heuristics that can direct the anlysis to specific parts of the image or to specific objects. Thus the user is able to define the actions that should be executed when certain situations arise. For example, the FOCUS OF ATTENTION can be used to locate new objects; it will halt the system or direct it to search in a pre-specified area when a region is interpreted as a specific object with a confidence above some threshold.

The system is a new approach that incorporates the high level contextual clues in computer vision. The use of distinguishing features with veto power eliminates many of

the initially unrealistic region interpretations. This in turn reduces the time required for the subsequent analysis. As tested on real outdoor scenes, the results of the analysis are quite satisfactory. However, certain limitations and improvements can be recognized. First, we did not consider any of the three dimensional properties of the scenes. We have also used objects such as SHADOW and WALL-SHADOW to avoid the problems associated with shadows in real world pictures. This may be a viable solution in general. However, further research should be directed to developing different processes for the analysis of shadows, highlights and occlusion. In particular, research should be guided to the design of three dimensional models of real world scenes, which directly relate to the problems of lighting and shadows. It is also of interest to consider models that describe more than one class of scenes or those that contain a large set of objects (200-300). In both cases, we need to develop methods to direct the analysis using a subset of these models. Thus an additional stage of analysis would be needed to first distinguish these submodels.

Another area of future research is the development of a more sophisticated LOW LEVEL PROCESSOR where lines, edges, vertices and regions are used to obtain a complete or partial segmentation. We expect that the current system

structure will facilitate future development of feedback paths between the different processes in the system. For example, the high level stages of processing would use their intermediate and final deductions to direct the low level stages to specific parts of the picture, or to calculate computationally expensive features (e.g. texture).

Moreover, although the current implementation was designed on a sequential machine, it is possible to modify the processes to work in parallel depending on the hardware resources available.

Finally, we need to develop a more advanced language to express complicated constraints and condition/action rules in the declarative format. Further research should be done in identifying and developing heuristic PLANS [66] that can be incorporated in the condition/action rules. These PLANS could be used in different world models to achieve specific strategies.

In conclusion, computer vision systems are very complex to design and implement. We have implemented a system structure that can be used as a framework to incorporate more processes. The use of a competition and cooperation paradigm in the form of a relaxation labelling process proved to be a successful approach to interpreting

real world scenes. We have separated the model from the analysis code, and therefore the system is not restricted to a specific world model. This also facilitates the investigation of real world models because it allows the user to incrementally build the model. Finally, the use of a relational database and the relational algebraic sublanguage facilitates the incorporation of more processes in the system and provides the required level of independence to permit parallel operation. In fact, we foresee the parallel operation of the different processes as a very promising research area, especially with the recent advancement being made in the field of microprocessor design. We can easily imagine each process to be implemented using a separate microprocessor system.

APPENDIX I: SHORT TERM MEMORY RELATIONS

This Appendix contains samples of the following STM relations:

	page
REGION FEATURES RELATION	287
LEFT-OF RELATION	289
RIGHT-OF RELATION	290
ABOVE RELATION	291
BELOW RELATION	292
ADJACENT RELATION	293
CONTAINS RELATION	294
ENCLOSED-BY RELATION	295
CHAIN CODE RELATION	296
REGION INTERPRETATION RELATION	297
MERGE HISTORY RELATION	301
PATH RELATION	302

RELATION REGION FEATURES

DOMAINS: 1 ---> 12

	REG_NO.	FLAG	FEAT-1	FEAT-2	FEAT-3	VAR-1	VAR-2	VAR-3	INTEN	CENT-X	CENT-Y	MIN-X
1-	1	0	42	47	44	3.867710	6.502746	5.184848	44	3	127	1
2-	2	0	45	52	51	2.652132	1.856468	2.097706	49	82	114	8
3-	3	0	29	34	26	2.897938	2.715656	2.849383	30	127	107	126
4-	4	0	44	41	30	2.861380	2.173067	4.017323	38	3	125	2
5-	5	0	40	40	32	3.127413	2.298187	5.138166	37	11	125	6
6-	6	0	37	29	18	3.319763	3.174333	2.409471	28	1	121	1
7-	7	0	36	32	19	4.922763	4.298836	3.620829	29	7	122	6
8-	8	0	36	34	23	4.089544	3.811085	4.421467	31	13	120	9
9-	9	0	35	33	21	5.149019	4.168952	4.865334	30	17	122	15
10-	10	0	35	39	33	3.194382	3.815917	5.984248	36	21	120	18
11-	11	0	30	28	16	6.757285	5.738234	4.861401	25	4	121	3
12-	12	0	27	28	17	4.380450	3.767156	3.685051	24	13	115	8
13-	13	0	35	35	25	4.740714	3.931920	4.625405	32	21	114	17
14-	14	0	34	36	31	5.044868	5.610826	5.462669	34	2	111	1
15-	15	0	33	35	26	5.202510	4.228982	4.827820	31	8	116	6
16-	16	0	37	35	22	3.079631	2.985339	4.670757	31	30	110	23
17-	17	0	38	44	40	1.500000	2.394655	4.472136	40	3	115	2
18-	18	0	38	42	36	1.901633	3.391165	5.319388	39	34	113	30
19-	19	0	34	31	18	3.688266	2.757783	3.263877	28	20	110	18
20-	20	0	37	39	30	4.885465	3.518545	4.538175	35	5	114	3
21-	21	0	36	40	32	4.759845	4.336797	5.530464	36	11	110	7
22-	22	0	41	47	44	1.709815	2.094696	3.579989	44	39	111	38
23-	23	0	39	46	41	3.815174	3.125833	4.074820	42	4	110	2
24-	24	0	38	39	28	5.704265	5.088088	6.000545	35	18	104	15
25-	25	0	37	38	27	1.497683	2.285218	3.402409	34	31	107	30
26-	26	0	42	46	41	2.538591	3.281071	5.079612	43	42	107	41
27-	27	0	36	35	24	4.337337	3.848193	3.876511	32	14	105	13
28-	28	0	37	38	28	5.487350	4.031103	4.416666	34	24	104	21
29-	29	0	36	32	19	4.673239	4.263358	4.035694	29	6	99	1
30-	30	0	29	33	25	3.722818	4.270172	5.170529	29	6	106	6
31-	31	0	38	45	42	2.061485	2.132812	3.395911	42	78	103	74
32-	32	0	43	47	42	2.761792	3.136877	5.045790	44	10	104	7
33-	33	0	27	25	16	3.572114	2.993325	4.500000	23	41	105	39
34-	34	0	37	33	19	5.165612	5.006227	5.519828	30	39	96	34
35-	35	0	38	44	38	2.375525	2.875819	4.342365	40	73	98	71
36-	36	0	39	43	35	3.899660	2.779302	4.445558	39	81	94	77
37-	37	0	43	46	37	3.464542	1.512906	4.730985	42	88	98	86
38-	38	0	26	22	12	3.938349	4.366015	2.395210	20	2	93	1
39-	39	0	43	42	30	2.370928	2.258448	3.415938	38	11	100	11
40-	40	0	47	42	29	3.045115	3.839766	5.693986	39	29	101	26
41-	41	0	40	42	34	3.073666	2.269155	2.652796	39	57	90	42
42-	42	0	34	39	33	5.059341	5.874972	6.830677	35	15	100	13
43-	43	0	39	35	22	4.459181	6.666804	8.585015	32	23	100	21
44-	44	0	42	44	35	3.469711	2.656560	3.422938	40	78	83	65
45-	45	0	44	47	42	4.478723	4.551045	7.147934	44	15	96	13
46-	46	0	42	38	24	3.890773	3.440858	4.078227	35	30	94	27
47-	47	0	52	49	38	1.673320	2.507987	4.561797	46	32	98	31
48-	48	0	33	39	31	1.812544	1.835325	2.489868	34	76	97	75
49-	49	0	34	29	14	3.079671	2.777560	2.331844	26	4	95	3
50-	50	0	41	39	29	2.436698	2.758509	5.633327	36	10	97	9

RELATION REGION FEATURES

DOMAINS: 13 ---> 19

	MIN-Y	MAX-X	MAX-Y	AREA	PERIMTR	CIRCLRTY	ASPECT-R
1-	126	8	128	14	18.24200	0.528679	2.666666
2-	93	127	128	2746	320.2160	0.336530	3.333333
3-	88	128	128	43	82.82800	0.078763	0.073170
4-	124	5	127	12	12.24199	1.006203	1.000000
5-	124	15	127	23	23.31199	0.531836	2.500000
6-	119	3	125	12	13.65600	0.808618	0.428571
7-	119	10	125	25	21.89799	0.655149	0.714285
8-	116	16	125	40	26.72599	0.703722	0.800000
9-	119	20	125	22	20.48399	0.658874	0.857142
10-	117	24	125	28	25.06999	0.559833	0.777777
11-	118	5	124	17	15.65599	0.871559	0.428571
12-	111	19	120	31	32.96799	0.358414	1.200000
13-	108	25	120	40	36.96800	0.367804	0.692307
14-	105	5	119	37	40.96800	0.277026	0.333333
15-	113	11	119	22	20.48399	0.658874	0.857142
16-	101	40	120	157	71.59201	0.384928	0.900000
17-	115	6	117	8	12.24199	0.670802	1.666666
18-	109	38	117	32	29.55399	0.460391	1.000000
19-	106	22	116	22	28.48399	0.340745	0.454545
20-	113	7	115	11	10.82800	1.178977	1.666666
21-	106	17	115	59	36.38199	0.560129	1.100000
22-	109	41	114	14	14.24199	0.867352	0.666666
23-	107	7	113	24	19.06999	0.829314	0.857142
24-	98	22	112	63	48.03800	0.343067	0.533333
25-	106	33	110	12	11.65599	1.109920	0.800000
26-	104	43	110	9	14.24199	0.557583	0.428571
27-	102	16	109	16	17.06999	0.690021	0.500000
28-	100	30	109	35	32.13999	0.425780	1.000000
29-	92	13	108	83	55.45200	0.339197	0.764705
30-	104	8	108	8	10.24199	0.958362	0.600000
31-	99	82	108	60	29.89799	0.843483	0.900000
32-	103	13	107	20	17.06999	0.862526	1.399999
33-	104	42	107	10	10.82800	1.071798	1.000000
34-	85	43	106	93	69.93601	0.238940	0.454545
35-	90	77	105	52	52.96800	0.232908	0.437500
36-	86	85	104	97	48.96800	0.508342	0.473684
37-	93	96	104	60	40.96799	0.449232	0.916666
38-	88	7	103	37	44.13999	0.238641	0.437500
39-	98	13	103	13	12.82800	0.992738	0.500000
40-	99	33	103	22	21.31199	0.608672	1.600000
41-	74	71	103	460	117.5920	0.418033	1.000000
42-	99	18	102	14	15.65600	0.717754	1.500000
43-	99	26	102	11	12.48399	0.886941	1.500000
44-	73	95	101	254	117.3500	0.231780	1.068965
45-	94	17	100	21	17.06999	0.905652	0.714285
46-	91	35	100	36	32.13999	0.437945	0.900000
47-	97	33	100	10	8.828000	1.612444	0.750000
48-	93	79	100	19	18.48399	0.698829	0.625000
49-	93	7	99	16	17.65600	0.644977	0.714285
50-	97	13	99	8	10.82800	0.857438	1.666666

RELATION LEFT_OF

	REG_X	REG_Y	VALUE
1-	1	2	0.875000
2-	1	5	0.443181
3-	2	3	0.962056
4-	2	56	0.175628
5-	2	65	0.508333
6-	4	5	1.000000
7-	4	7	1.000000
8-	5	2	0.111392
9-	5	9	0.900000
10-	6	4	0.083333
11-	6	11	0.666666
12-	7	8	0.475000
13-	7	12	0.177777
14-	8	9	0.583333
15-	8	13	1.000000
16-	9	10	0.214285
17-	10	16	0.623376
18-	11	7	1.000000
19-	11	15	1.000000
20-	12	13	0.568181
21-	12	19	0.722222
22-	12	24	0.183333
23-	13	16	0.499999
24-	14	15	1.000000
25-	14	20	0.066666
26-	14	30	1.000000
27-	16	22	0.666666
28-	16	26	1.000000
29-	16	33	0.811965
30-	16	34	0.236111
31-	17	15	0.800000
32-	18	22	0.888888
33-	19	28	0.433333
34-	20	15	0.400000
35-	21	24	0.527272
36-	21	27	0.045454
37-	22	26	0.750000
38-	23	21	0.833333
39-	23	30	0.416666
40-	24	28	0.625000
41-	24	43	0.607142
42-	24	51	0.425000
43-	26	41	0.274509
44-	27	24	0.333333
45-	28	40	0.100000
46-	29	39	0.519230
47-	29	45	0.923076
48-	29	50	0.170940
49-	29	59	0.215384
50-	29	73	0.032967

RELATION RIGHT_OF

	REG_X	REG_Y	VALUE
1-	1	6	0.225000
2-	2	1	0.991666
3-	2	5	0.844725
4-	2	9	0.737820
5-	2	10	0.653205
6-	2	16	0.348529
7-	2	18	0.361919
8-	2	22	0.288095
9-	2	26	0.239473
10-	3	2	0.312056
11-	3	65	0.250000
12-	3	79	0.083333
13-	4	6	0.250000
14-	5	1	0.518181
15-	5	4	1.000000
16-	5	7	0.055555
17-	7	4	1.000000
18-	7	11	1.000000
19-	8	7	0.625000
20-	9	5	0.833333
21-	9	8	0.499999
22-	10	9	0.285714
23-	11	6	0.666666
24-	12	7	0.527777
25-	12	15	0.333333
26-	13	8	1.000000
27-	13	9	0.180555
28-	13	12	0.484848
29-	15	11	1.000000
30-	15	14	1.000000
31-	15	17	0.833333
32-	15	20	0.466666
33-	16	10	0.797979
34-	16	13	0.666666
35-	16	28	0.055555
36-	19	12	0.488888
37-	20	14	0.066666
38-	21	15	0.045454
39-	21	23	0.909090
40-	21	30	0.675324
41-	22	16	0.083333
42-	22	18	0.750000
43-	24	21	0.425000
44-	24	27	0.583333
45-	24	42	0.071428
46-	24	45	0.291666
47-	26	16	1.000000
48-	26	22	0.666666
49-	27	32	0.750000
50-	27	39	0.750000

RELATION ABOVE

	REG_X	REG_Y	VALUE
1-	1	4	0.083333
2-	1	5	0.083333
3-	1	6	1.000000
4-	2	35	0.225095
5-	2	36	0.322916
6-	2	37	0.273809
7-	2	41	0.408730
8-	2	44	0.550000
9-	2	56	0.666666
10-	2	65	0.907407
11-	3	65	0.589268
12-	3	79	0.912757
13-	3	85	1.000000
14-	4	6	0.333333
15-	4	7	0.300000
16-	4	11	0.750000
17-	5	7	0.300000
18-	5	8	0.357142
19-	5	9	0.300000
20-	6	14	0.857142
21-	7	12	0.603174
22-	7	15	0.857142
23-	8	12	0.100000
24-	8	13	0.136363
25-	9	13	0.614285
26-	10	13	0.255555
27-	10	16	0.341269
28-	11	14	0.630952
29-	11	15	0.571428
30-	12	21	0.100000
31-	12	24	0.723076
32-	13	28	0.762820
33-	14	29	0.545833
34-	14	30	0.400000
35-	15	20	0.171428
36-	15	21	0.349206
37-	16	28	0.016666
38-	16	33	0.150000
39-	16	34	0.450000
40-	16	40	0.683333
41-	17	20	0.666666
42-	19	28	0.336363
43-	20	23	0.666666
44-	21	27	0.266666
45-	21	30	0.414285
46-	21	32	0.675000
47-	22	26	0.499999
48-	23	30	0.547619
49-	24	42	0.222222
50-	24	43	0.222222

RELATION BELOW

	REG_X	REG_Y	VALUE
1-	2	1	0.791666
2-	2	5	0.611111
3-	2	9	0.222222
4-	2	10	0.055555
5-	4	1	0.250000
6-	5	1	0.250000
7-	6	1	1.000000
8-	6	4	0.547619
9-	7	4	0.514285
10-	7	5	0.514285
11-	8	5	0.657142
12-	9	5	0.514285
13-	11	4	0.857142
14-	12	7	0.688888
15-	12	8	0.100000
16-	13	8	0.251748
17-	13	9	0.746153
18-	13	10	0.392307
19-	14	6	0.933333
20-	14	11	0.783333
21-	14	15	0.033333
22-	14	17	0.222222
23-	15	7	0.857142
24-	15	11	0.571428
25-	16	10	0.585714
26-	20	17	0.666666
27-	21	12	0.100000
28-	21	15	0.477777
29-	23	20	0.857142
30-	24	12	0.789743
31-	24	19	0.071794
32-	24	21	0.071794
33-	26	22	0.547619
34-	27	21	0.166666
35-	28	13	0.716666
36-	28	19	0.299999
37-	29	14	0.577205
38-	29	30	0.372549
39-	29	32	0.192513
40-	30	21	0.114285
41-	30	23	0.433333
42-	32	21	0.475000
43-	34	16	0.477272
44-	34	33	0.696969
45-	34	40	0.097902
46-	35	31	0.062500
47-	36	2	0.024671
48-	36	31	0.350877
49-	39	27	0.523809
50-	39	32	0.833333

RELATION ADJACENT

	REG_X	REG_Y	VALUE
1-	1	2	0.142857
2-	1	4	0.500000
3-	1	5	0.285714
4-	1	6	0.071428
5-	2	1	0.008695
6-	2	3	0.152173
7-	2	5	0.039130
8-	2	9	0.013043
9-	2	10	0.043478
10-	2	16	0.043478
11-	2	18	0.047826
12-	2	22	0.043478
13-	2	26	0.043478
14-	2	31	0.078260
15-	2	35	0.047826
16-	2	36	0.021739
17-	2	37	0.078260
18-	2	41	0.113043
19-	2	44	0.052173
20-	2	56	0.078260
21-	2	65	0.095652
22-	3	2	0.760869
23-	3	65	0.152173
24-	3	79	0.065217
25-	3	85	0.021739
26-	4	1	0.388888
27-	4	5	0.055555
28-	4	6	0.166666
29-	4	7	0.111111
30-	4	11	0.277777
31-	5	1	0.125000
32-	5	2	0.281250
33-	5	4	0.031250
34-	5	7	0.218750
35-	5	8	0.250000
36-	5	9	0.093750
37-	6	1	0.076923
38-	6	4	0.230769
39-	6	11	0.461538
40-	6	14	0.230769
41-	7	4	0.066666
42-	7	5	0.233333
43-	7	8	0.233333
44-	7	11	0.133333
45-	7	12	0.200000
46-	7	15	0.133333
47-	8	5	0.222222
48-	8	7	0.194444
49-	8	9	0.138888
50-	8	12	0.305555

RELATION CONTAINS

	REG_X	REG_Y	VALUE
1-	2	31	0.020625
2-	16	25	0.069767
3-	41	107	0.021231
4-	52	72	0.078260
5-	81	149	0.024316
6-	113	147	0.145161
7-	225	240	0.250000
8-	234	246	0.135135
9-	234	256	0.202702
10-	237	258	0.088105
11-	287	290	0.004805

RELATION ENCLOSED_BY

	REG_X	REG_Y	VALUE
1-	25	16	0.069767
2-	31	2	0.020625
3-	72	52	0.078260
4-	107	41	0.021231
5-	147	113	0.145161
6-	149	81	0.024316
7-	240	225	0.250000
8-	246	234	0.135135
9-	256	234	0.202702
10-	258	237	0.088105
11-	290	287	0.004805

RELATION CHAIN CODE

DOMAINS: 1 ---> 12

	REG_NO.	FLAG	LENGTH	X_START	Y_START	CHAIN_1	CHAIN_2	CHAIN_3	CHAIN_4	CHAIN_5	CHAIN_6	CHAIN_7
1-	1	1	17	1	126	02100700	34444446	68888888	88888888	88888888	88888888	88888888
2-	2	1	302	107	93	01000000	00000017	00002222	22222222	22222222	22222222	22222244
3-	2	2	0	0	0	44444444	44444444	44444444	44444444	44444444	44444444	44444444
4-	2	3	0	0	0	44444444	44444444	44444444	44444444	44444444	44444444	44444444
5-	2	4	0	0	0	44444000	00077000	76000766	70007077	00007661	70766617	67666660
6-	2	5	0	0	0	70000700	07000700	00022170	77022222	21702221	70000007	66617017
7-	2	6	0	0	0	00077707	07700760	00076788	88888888	88888888	88888888	88888888
8-	3	1	82	128	88	22222222	22222222	22222222	22222222	22222222	66666666	66666666
9-	3	2	0	0	0	66666666	66666666	66666654	07888888	88888888	88888888	88888888
10-	4	1	11	2	124	10722244	65688888	88888888	88888888	88888888	88888888	88888888
11-	5	1	20	11	124	11761234	44445440	00778888	88888888	88888888	88888888	88888888
12-	6	1	12	1	119	11323266	66668888	88888888	88888888	88888888	88888888	88888888
13-	7	1	19	7	119	03100331	03444666	66788888	88888888	88888888	88888888	88888888
14-	8	1	23	15	116	02222322	43255447	76707678	88888888	88888888	88888888	88888888
15-	9	1	18	18	119	01244312	24457660	76888888	88888888	88888888	88888888	88888888
16-	10	1	23	21	117	01032222	44432665	00766668	88888888	88888888	88888888	88888888
17-	11	1	14	5	118	22222346	67560788	88888888	88888888	88888888	88888888	88888888
18-	12	1	28	19	111	22432443	32433447	07677076	00778888	88888888	88888888	88888888
19-	13	1	32	25	108	23222233	43225432	66661760	76776607	88888888	88888888	88888888
20-	14	1	36	3	105	00223234	32213221	70034356	66666666	76678888	88888888	88888888
21-	15	1	18	8	113	22170232	44436676	76888888	88888888	88888888	88888888	88888888
22-	16	1	60	34	101	00312211	70224443	32314435	33122434	44367546	77666676	75670007
23-	16	2	0	0	0	17768888	88888888	88888888	88888888	88888888	88888888	88888888
24-	17	1	11	2	115	10072444	36688888	88888888	88888888	88888888	88888888	88888888
25-	18	1	25	37	109	02232322	44443657	17007576	78888888	88888888	88888888	88888888
26-	19	1	26	20	106	00243210	23324326	60766666	66888888	88888888	88888888	88888888
27-	20	1	10	5	113	00243446	07888888	88888888	88888888	88888888	88888888	88888888
28-	21	1	31	8	106	00210722	10003332	43543665	66507658	88888888	88888888	88888888
29-	22	1	13	39	109	00322243	66766888	88888888	88888888	88888888	88888888	88888888
30-	23	1	17	6	107	22122443	44660776	78888888	88888888	88888888	88888888	88888888
31-	24	1	41	18	98	01003311	22443222	23347754	47765466	60107656	68888888	88888888
32-	25	1	10	31	106	02124366	57888888	88888888	88888888	88888888	88888888	88888888
33-	26	1	13	43	104	22223265	07666888	88888888	88888888	88888888	88888888	88888888
34-	27	1	15	14	102	22210233	66656678	88888888	88888888	88888888	88888888	88888888
35-	28	1	28	25	100	21100044	43321434	34607666	66778888	88888888	88888888	88888888
36-	29	1	48	7	92	10101144	43321022	44343443	32266667	66661766	07075476	88888888
37-	30	1	9	6	104	21124656	68888888	88888888	88888888	88888888	88888888	88888888
38-	31	1	27	80	99	02122322	24444443	66666700	77788888	88888888	88888888	88888888
39-	32	1	15	8	103	00102134	65444678	88888888	88888888	88888888	88888888	88888888
40-	33	1	10	41	104	02224654	07888888	88888888	88888888	88888888	88888888	88888888
41-	34	1	60	41	85	02332313	21700243	21322122	43432566	57544666	21007665	76666507
42-	34	2	0	0	0	60768888	88888888	88888888	88888888	88888888	88888888	88888888
43-	35	1	48	75	90	02232222	23100244	33243666	66666766	72232217	76666666	88888888
44-	36	1	44	80	86	21021331	00022222	22222436	66565446	65566667	66078888	88888888
45-	37	1	36	86	93	01123217	00217004	34333444	36666666	66668888	88888888	88888888
46-	38	1	40	1	88	02176171	43443213	22223222	26666666	66666666	88888888	88888888
47-	39	1	12	11	98	02221346	66668883	88888888	88888888	88888888	88888888	88888888
48-	40	1	18	28	99	21710722	35444456	07888888	88888888	88888888	88888888	88888888
49-	41	1	106	57	74	21702443	22100007	60000000	03222222	22222332	22224443	44444344
50-	41	2	0	0	0	43444344	44343665	66756076	07070776	60707766	66754076	66888888

RELATION REGION INTERPRETATION

DOMAINS: 1 ---> 12

	REG_NO	FLAG	NO_INTRP	UPDATES	OBJ_NAM1	CONF_1	OBJ_NAM2	CONF_2	OBJ_NAM3	CONF_3	OBJ_NAM4	CONF_4
1-	1	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
2-	2	0	2	0	SKY	0.600000	TREE	0.389999	NONE	0.000000	NONE	0.000000
3-	3	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
4-	4	0	2	0	TREE	0.540000	SKY	0.450000	NONE	0.000000	NONE	0.000000
5-	5	0	2	0	TREE	0.540000	SKY	0.450000	NONE	0.000000	NONE	0.000000
6-	6	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
7-	7	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
8-	8	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
9-	9	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
10-	10	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
11-	11	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
12-	12	0	2	0	TREE	0.660000	SKY	0.330000	NONE	0.000000	NONE	0.000000
13-	13	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
14-	14	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
15-	15	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
16-	16	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
17-	17	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
18-	18	0	2	0	TREE	0.540000	SKY	0.450000	NONE	0.000000	NONE	0.000000
19-	19	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
20-	20	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
21-	21	0	2	0	TREE	0.540000	SKY	0.450000	NONE	0.000000	NONE	0.000000
22-	22	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
23-	23	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
24-	24	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
25-	25	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
26-	26	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
27-	27	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
28-	28	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
29-	29	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
30-	30	0	2	0	TREE	0.660000	SKY	0.330000	NONE	0.000000	NONE	0.000000
31-	31	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
32-	32	0	2	0	SKY	0.500000	TREE	0.500000	NONE	0.000000	NONE	0.000000
33-	33	0	2	0	TREE	0.660000	SKY	0.330000	NONE	0.000000	NONE	0.000000
34-	34	0	5	0	TREE	0.300000	SKY	0.190000	WALL_SH.	0.190000	WALL	0.150000
35-	35	0	5	0	TREE	0.279999	SKY	0.229999	WALL	0.180000	WINDOW	0.139999
36-	36	0	5	0	TREE	0.279999	SKY	0.229999	WALL	0.180000	WALL_SH.	0.139999
37-	37	0	5	0	TREE	0.270000	WALL	0.220000	SKY	0.220000	WINDOW	0.129999
38-	38	0	6	0	WALL_SH.	0.259999	WINDOW	0.170000	TREE	0.170000	ROOF	0.129999
39-	39	0	2	0	TREE	0.540000	SKY	0.450000	NONE	0.000000	NONE	0.000000
40-	40	0	2	0	SKY	0.500000	TREE	0.500000	NENE	0.000000	NONE	0.000000
41-	41	0	5	0	TREE	0.259999	ROOF	0.259999	WALL	0.209999	WALL_SH.	0.129999
42-	42	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
43-	43	0	2	0	TREE	0.600000	SKY	0.389999	NONE	0.000000	NONE	0.000000
44-	44	0	5	0	ROOF	0.259999	TREE	0.259999	WALL	0.209999	WALL_SH.	0.129999
45-	45	0	5	0	SKY	0.250000	TREE	0.250000	WALL	0.250000	WALL_SH.	0.119999
46-	46	0	5	0	TREE	0.279999	SKY	0.180000	WALL_SH.	0.180000	WALL	0.180000
47-	47	0	5	0	WALL	0.250000	TREE	0.250000	SKY	0.190000	WINDOW	0.150000
48-	48	0	5	0	TREE	0.310000	SKY	0.200000	WALL_SH.	0.150000	WALL	0.150000
49-	49	0	5	0	WALL_SH.	0.229999	TREE	0.229999	SKY	0.180000	WINDOW	0.180000
50-	50	0	5	0	TREE	0.279999	WALL_SH	0.180000	WALL	0.180000	SKY	0.180000

RELATION REGION INTERPRETATION
****
DOMAINS: 13 ---> 20

	OBJ_NAM5	CONF_5	OBJ_NAM6	CONF_6	OBJ_NAM7	CONF_7	OBJ_NAM8	CONF_8
1-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
2-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
3-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
4-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
5-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
6-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
7-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
8-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
9-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
10-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
11-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
12-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
13-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
14-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
15-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
16-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
17-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
18-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
19-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
20-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
21-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
22-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
23-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
24-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
25-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
26-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
27-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
28-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
29-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
30-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
31-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
32-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
33-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
34-	WINDOW	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
35-	WALL_SH.	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
36-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
37-	WALL_SH.	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
38-	SKY	0.129999	WALL	0.129999	NONE	0.000000	NONE	0.000000
39-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
40-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
41-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
42-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
43-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
44-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
45-	WINDOW	0.119999	NONE	0.000000	NONE	0.000000	NONE	0.000000
46-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
47-	WALL_SH.	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
48-	WINDOW	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
49-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
50-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000

RELATION REGION INTERPRETATION

DOMAINS: 1 ---> 12

	REG_NO	FLAG	NO_INTRP	UPDATES	OBJ_NAM1	CONF_1	OBJ_NAM2	CONF_2	OBJ_NAM3	CONF_3	OBJ_NAM4	CONF_4
51-	51	0	5	0	TREE	0.279999	WALL_SH.	0.180000	WALL	0.180000	SKY	0.180000
52-	52	0	5	0	WALL_SH.	0.250000	TREE	0.250000	ROOF	0.190000	WINDOW	0.150000
53-	53	0	5	0	TREE	0.270000	WALL	0.220000	SKY	0.220000	WINDOW	0.129999
54-	54	0	5	0	TREE	0.300000	SKY	0.190000	WALL_SH.	0.190000	WINDOW	0.150000
55-	55	0	5	0	TREE	0.270000	WALL_SH.	0.220000	ROOF	0.220000	WALL	0.129999
56-	56	0	5	0	ROOF	0.250000	WALL	0.250000	TREE	0.250000	WALL_SH.	0.119999
57-	57	0	5	0	TREE	0.279999	SKY	0.180000	WALL	0.180000	WALL_SH.	0.180000
58-	58	0	5	0	TREE	0.270000	WALL	0.220000	SKY	0.220000	WALL_SH.	0.129999
59-	59	0	5	0	ROOF	0.270000	TREE	0.270000	WALL_SH.	0.180000	WALL	0.129999
60-	60	0	5	0	TREE	0.270000	ROOF	0.220000	WALL_SH.	0.180000	WALL	0.180000
61-	61	0	5	0	TREE	0.270000	ROOF	0.270000	WALL_SH.	0.180000	WINDOW	0.129999
62-	62	0	5	0	ROOF	0.229999	TREE	0.229999	WALL	0.229999	WALL_SH.	0.139999
63-	63	0	5	0	WALL_SH.	0.229999	TREE	0.229999	ROOF	0.180000	WINDOW	0.180000
64-	64	0	5	0	TREE	0.279999	WALL_SH.	0.229999	ROOF	0.180000	WINDOW	0.139999
65-	65	0	5	0	ROOF	0.259999	TREE	0.259999	WALL	0.209999	WALL_SH.	0.129999
66-	66	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WINDOW	0.139999
67-	67	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WALL	0.139999
68-	68	0	5	0	TREE	0.259999	WALL_SH.	0.209999	ROOF	0.209999	WINDOW	0.170000
69-	69	0	5	0	ROOF	0.259999	TREE	0.259999	WALL	0.209999	WALL_SH.	0.129999
70-	70	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WALL	0.139999
71-	71	0	5	0	TREE	0.270000	ROOF	0.220000	WALL	0.180000	WALL_SH.	0.180000
72-	72	0	5	0	WALL_SH.	0.300000	WINDOW	0.239999	WALL	0.150000	ROOF	0.150000
73-	73	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WINDOW	0.139999
74-	74	0	5	0	ROOF	0.259999	WALL	0.209999	TREE	0.209999	WALL_SH.	0.170000
75-	75	0	5	0	ROOF	0.250000	TREE	0.250000	WALL	0.200000	WALL_SH.	0.160000
76-	76	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WINDOW	0.139999
77-	77	0	5	0	WALL_SH.	0.279999	TREE	0.279999	WALL	0.139999	WINDOW	0.139999
78-	78	0	5	0	TREE	0.270000	ROOF	0.270000	WALL_SH.	0.180000	WALL	0.129999
79-	79	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WALL	0.139999
80-	80	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WALL	0.139999
81-	81	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WINDOW	0.139999
82-	82	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WINDOW	0.139999
83-	83	0	5	0	WALL	0.250000	TREE	0.250000	ROOF	0.250000	WALL_SH.	0.119999
84-	84	0	5	0	ROOF	0.270000	TREE	0.270000	WALL_SH.	0.180000	WALL	0.129999
85-	85	0	6	0	BUSHES	0.229999	WALL_SH.	0.229999	WINDOW	0.150000	TREE	0.150000
86-	86	0	5	0	WALL_SH.	0.300000	TREE	0.190000	WINDOW	0.190000	ROOF	0.150000
87-	87	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WINDOW	0.139999
88-	88	0	5	0	ROOF	0.250000	TREE	0.250000	WALL	0.250000	WALL_SH.	0.119999
89-	89	0	4	0	WALL	0.350000	TREE	0.289999	WINDOW	0.170000	WALL_SH.	0.170000
90-	90	0	5	0	ROOF	0.270000	TREE	0.270000	WALL_SH.	0.180000	WALL	0.129999
91-	91	0	5	0	TREE	0.259999	ROOF	0.259999	WALL_SH.	0.170000	WALL	0.170000
92-	92	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WALL	0.139999
93-	93	0	5	0	ROOF	0.259999	TREE	0.259999	WALL	0.209999	WINDOW	0.129999
94-	94	0	5	0	ROOF	0.259999	TREE	0.259999	WALL_SH.	0.209999	WINDOW	0.129999
95-	95	0	5	0	WALL	0.250000	ROOF	0.250000	TREE	0.250000	WINDOW	0.119999
96-	96	0	5	0	ROOF	0.259999	TREE	0.259999	WALL_SH.	0.170000	WALL	0.170000
97-	97	0	5	0	WALL_SH.	0.310000	WINDOW	0.200000	ROOF	0.150000	WALL	0.150000
98-	98	0	5	0	TREE	0.279999	ROOF	0.229999	WALL_SH.	0.180000	WALL	0.139999
99-	99	0	5	0	WALL_SH.	0.279999	TREE	0.229999	WINDOW	0.180000	ROOF	0.139999
100-	100	0	5	0	WALL_SH.	0.270000	TREE	0.220000	WINDOW	0.180000	ROOF	0.180000

RELATION REGION INTERPRETATION
****
DOMAINS: 13 ---> 20

	OBJ_NAM5	CONF_5	OBJ_NAM6	CONF_6	OBJ_NAM7	CONF_7	OBJ_NAM8	CONF_8
51-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
52-	WALL	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
53-	WALL_SH.	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
54-	WALL	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
55-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
56-	WINDOW	0.119999	NONE	0.000000	NONE	0.000000	NONE	0.000000
57-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
58-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
59-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
60-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
61-	WALL	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
62-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
63-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
64-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
65-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
66-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
67-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
68-	WALL	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
69-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
70-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
71-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
72-	TREE	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
73-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
74-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
75-	WINDOW	0.119999	NONE	0.000000	NONE	0.000000	NONE	0.000000
76-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
77-	ROOF	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
78-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
79-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
80-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
81-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
82-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
83-	WINDOW	0.119999	NONE	0.000000	NONE	0.000000	NONE	0.000000
84-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
85-	GRASS	0.110000	WALL	0.110000	NONE	0.000000	NONE	0.000000
86-	WALL	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
87-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
88-	WINDOW	0.119999	NONE	0.000000	NONE	0.000000	NONE	0.000000
89-	NONE	0.000000	NONE	0.000000	NONE	0.000000	NONE	0.000000
90-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
91-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
92-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
93-	WALL_SH.	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
94-	WALL	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
95-	WALL_SH.	0.119999	NONE	0.000000	NONE	0.000000	NONE	0.000000
96-	WINDOW	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000
97-	TREE	0.150000	NONE	0.000000	NONE	0.000000	NONE	0.000000
98-	WINDOW	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
99-	WALL	0.139999	NONE	0.000000	NONE	0.000000	NONE	0.000000
100-	WALL	0.129999	NONE	0.000000	NONE	0.000000	NONE	0.000000

RELATION MERGE HISTORY

	ITER_NO	REG_X	REG_Y
1-	192	21	24
2-	200	24	30
3-	201	23	24
4-	202	20	23
5-	204	15	20
6-	205	12	15
7-	206	12	19
8-	207	12	28
9-	208	10	13
10-	209	8	10
11-	209	8	12
12-	511	2	18
13-	512	8	16
14-	773	8	51
15-	777	8	25
16-	1042	8	33
17-	1043	2	26
18-	1293	31	36
19-	1304	1	2
20-	1402	127	181
21-	1581	41	44
22-	1588	127	129
23-	1598	104	113
24-	1598	113	143
25-	1622	41	78
26-	1698	188	189
27-	1728	179	183
28-	1728	179	193
29-	1729	179	187
30-	1730	179	186
31-	1738	188	194
32-	1740	179	180
33-	1771	8	57
34-	1773	8	46
35-	1776	8	68
36-	1776	8	87
37-	1778	8	34
38-	1861	182	188
39-	1878	177	182
40-	1882	65	82
41-	1889	127	131
42-	1940	278	279
43-	1963	205	217
44-	1965	205	206
45-	1984	65	79
46-	1991	31	37
47-	1993	8	32
48-	1995	94	108
49-	1998	94	153
50-	2006	177	228

RELATION PATH
****

	ITER_NO	REGION	CENT_X	CENT_Y
1-	1	2	82	114
2-	2	56	104	90
3-	3	52	94	90
4-	4	44	78	83
5-	5	110	75	75
6-	6	109	64	76
7-	7	41	57	90
8-	8	71	44	90
9-	9	75	43	87
10-	10	88	44	83
11-	11	96	43	79
12-	12	121	46	67
13-	13	157	40	62
14-	14	141	33	68
15-	15	145	32	65
16-	16	161	29	61
17-	17	140	29	67
18-	18	144	23	66
19-	19	156	15	65
20-	20	155	8	64
21-	21	169	6	60
22-	22	154	2	61
23-	23	137	4	67
24-	24	114	3	71
25-	25	150	8	67
26-	26	120	15	70
27-	27	98	11	75
28-	28	93	13	79
29-	29	73	13	85
30-	30	74	17	88
31-	31	62	18	91
32-	32	60	22	90
33-	33	91	28	82
34-	34	95	24	82
35-	35	90	21	80
36-	36	83	19	85
37-	37	105	22	74
38-	38	102	24	76
39-	39	115	28	72
40-	40	103	36	76
41-	41	142	44	68
42-	42	146	40	67
43-	43	117	40	71
44-	44	111	36	71
45-	45	116	33	73
46-	46	100	41	79
47-	47	87	36	83
48-	48	69	32	88
49-	49	63	35	91
50-	50	53	36	96

APPENDIX II: LONG TERM MEMORY RELATIONS.

This Appendix contains an example of LTM relations used during the experimentation. The relations are:

	Page
OBJECT FEATURES RELATION	304
DISTINGUISHING FEATURES RELATION	307
CONSTRAINTS RELATION	308

RELATION OBJECT FEATURES

DOMAINS: 1 ---> 12

	OBJ_NAME	OBJ_FLAG	FEAT-1	FEAT-2	FEAT-3	VAR-1	VAR-2	VAR-3	INTEN	CENT-X	CENT-Y	MIN-X
1-	BUSHES	1	20	20	10	0.000000	0.000000	0.000000	16	0	60	0
2-	CAR	1	25	40	45	0.000000	0.000000	0.000000	36	0	40	0
3-	FENCE	1	50	50	50	0.000000	0.000000	0.000000	50	0	20	0
4-	GRASS	1	35	40	25	0.000000	0.000000	0.000000	33	0	50	0
5-	ROAD	1	35	40	25	0.000000	0.000000	0.000000	33	0	20	0
6-	ROOF	1	40	40	35	0.000000	0.000000	0.000000	38	0	85	0
7-	SHADOW	1	20	20	20	0.000000	0.000000	0.000000	20	0	30	0
8-	SKY	1	40	50	50	0.000000	0.000000	0.000000	47	0	128	0
9-	TREE	1	35	35	25	0.000000	0.000000	0.000000	32	0	95	0
10-	TREE	2	40	40	40	0.000000	0.000000	0.000000	40	0	95	0
11-	WALL	1	50	50	50	0.000000	0.000000	0.000000	50	0	80	0
12-	WALL_SH.	1	20	20	20	0.000000	0.000000	0.000000	20	0	80	0
13-	WINDOW	1	10	10	10	0.000000	0.000000	0.000000	10	0	80	0

RELATION OBJECT FEATURES
****

DOMAINS: 13 ---> 24

	MIN-Y	MAX-X	MAX-Y	AREA	PERIMTR	CIRCLRTY	ASPECT-R	DFEAT-1	DFEAT-2	DFEAT-3	DVAR-1	DVAR-2
1-	0	0	0	0	0.000000	0.000000	0.000000	15	15	10	0.000000	0.000000
2-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
3-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
4-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
5-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
6-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
7-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
8-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
9-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
10-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
11-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
12-	0	0	0	0	0.000000	0.000000	0.000000	10	10	10	0.000000	0.000000
13-	0	0	0	0	0.000000	0.000000	0.000000	5	5	5	0.000000	0.000000

RELATION OBJECT FEATURES
****

DOMAINS: 25 ---> 36

	DVAR-3	DINTEN	DCENT-X	DCENT-Y	DMIN-X	DMIN-Y	DMAX-X	DMAX-Y	DAREA	DPERIMTR	DCRCLRTY	DASPCT-R
	********	********	********	********	********	********	********	********	********	********	********	********
1-	0.000000	10	0	10	0	0	0	0	0	0.000000	0.000000	0.000000
2-	0.000000	10	0	20	0	0	0	0	0	0.000000	0.000000	0.000000
3-	0.000000	10	0	20	0	0	0	0	0	0.000000	0.000000	0.000000
4-	0.000000	10	0	20	0	0	0	0	0	0.000000	0.000000	0.000000
5-	0.000000	10	0	30	0	0	0	0	0	0.000000	0.000000	0.000000
6-	0.000000	10	0	10	0	0	0	0	0	0.000000	0.000000	0.000000
7-	0.000000	10	0	30	0	0	0	0	0	0.000000	0.000000	0.000000
8-	0.000000	10	0	35	0	0	0	0	0	0.000000	0.000000	0.000000
9-	0.000000	10	0	35	0	0	0	0	0	0.000000	0.000000	0.000000
10-	0.000000	10	0	35	0	0	0	0	0	0.000000	0.000000	0.000000
11-	0.000000	10	0	20	0	0	0	0	0	0.000000	0.000000	0.000000
12-	0.000000	10	0	20	0	0	0	0	0	0.000000	0.000000	0.000000
13-	0.000000	5	0	20	0	0	0	0	0	0.000000	0.000000	0.000000

RELATION DISTINGUISHING FEATURES
****
DOMAINS: 1 ---> 12

	OBJ_NAME	OBJ_FLAG	FEATUR_1	VETO_1	WEIGHT_1	FEATUR_2	VETO_2	WEIGHT_2	FEATUR_3	VETO_3	WEIGHT_3	FEATUR_4
1-	BUSHES	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
2-	CAR	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
3-	FENCE	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
4-	GRASS	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
5-	ROAD	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
6-	ROOF	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
7-	SHADOW	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
8-	SKY	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
9-	TREE	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
10-	TREE	2	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
11-	WALL	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
12-	WALL_SH.	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y
13-	WINDOW	1	FEAT-1	FALSE	1	FEAT-2	FALSE	1	FEAT-3	FALSE	1	CENT-Y

RELATION DISTINGUISHING FEATURES
****
DOMAINS: 13 ---> 17

	VETO_4	WEIGHT_4	FEATUR_5	VETO_5	WEIGHT_5
1-	TRUE	3	NONE	NONE	0
2-	TRUE	3	NONE	NONE	0
3-	TRUE	3	NONE	NONE	0
4-	TRUE	3	NONE	NONE	0
5-	TRUE	3	NONE	NONE	0
6-	TRUE	3	NONE	NONE	0
7-	TRUE	3	NONE	NONE	0
8-	TRUE	3	NONE	NONE	0
9-	TRUE	3	NONE	NONE	0
10-	TRUE	3	NONE	NONE	0
11-	TRUE	3	NONE	NONE	0
12-	TRUE	3	NONE	NONE	0
13-	TRUE	3	NONE	NONE	0

RELATION CONSTRAINTS
****

DOMAINS: 1 ---> 12

	OBJ_X	OBJ_Y	NUMBER	CONSTR_1	OPERAT_1	VALUE_1	CONSTR_2	OPERAT_2	VALUE_2	CONSTR_3	OPERAT_3	VALUE_3
1-	BUSHES	BUSHES	1	FEAT-1	DIFF_LT	12.00000	FEAT-2	DIFF_LT	12.00000	FEAT-3	DIFF_LT	12.00000
2-	BUSHES	GRASS	1	ABOVE	.GT.	0.100000	FEAT-1	DIFF_GT	10.00000	NONE	NONE	0.000000
3-	BUSHES	GRASS	2	ABOVE	.GT.	0.100000	FEAT-2	DIFF_GT	10.00000	NONE	NONE	0.000000
4-	BUSHES	GRASS	3	ABOVE	.GT.	0.100000	FEAT-3	DIFF_GT	10.00000	NONE	NONE	0.000000
5-	BUSHES	TREE	1	BELOW	.GT.	0.300000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
6-	BUSHES	TREE	2	BELOW	.GT.	0.300000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
7-	BUSHES	TREE	3	BELOW	.GT.	0.300000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
8-	BUSHES	WALL	1	BELOW	.GT.	0.100000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
9-	BUSHES	WALL	2	BELOW	.GT.	0.100000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
10-	BUSHES	WALL	3	BELOW	.GT.	0.100000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
11-	CAR	CAR	1	FEAT-1	DIFF_LT	15.00000	FEAT-2	DIFF_LT	15.00000	FEAT-3	DIFF_LT	15.00000
12-	CAR	GRASS	1	ABOVE	.LE.	0.000000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
13-	CAR	GRASS	2	ABOVE	.LE.	0.000000	FEAT-2	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
14-	CAR	GRASS	3	ABOVE	.LE.	0.000000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
15-	CAR	ROAD	1	ABOVE	.GT.	0.300000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
16-	CAR	ROAD	2	ABOVE	.GT.	0.300000	FEAT-2	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
17-	CAR	ROAD	3	ABOVE	.GT.	0.300000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
18-	CAR	SHADOW	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
19-	CAR	SHADOW	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
20-	CAR	SHADOW	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
21-	CLOUDS	CLOUDS	1	FEAT-1	DIFF_LT	10.00000	FEAT-2	DIFF_LT	10.00000	FEAT-3	DIFF_LT	10.00000
22-	CLOUDS	SKY	1	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
23-	CLOUDS	SKY	2	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
24-	CLOUDS	SKY	3	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
25-	CLOUDS	TREE	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
26-	CLOUDS	TREE	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
27-	CLOUDS	TREE	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
28-	DOOR	DOOR	1	FEAT-1	DIFF_LT	5.000000	FEAT-2	DIFF_LT	5.000000	FEAT-3	DIFF_LT	5.000000
29-	DOOR	GRASS	1	ABOVE	.GT.	0.200000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
30-	DOOR	GRASS	2	ABOVE	.GT.	0.200000	FEAT-2	DIFF-GT	15.00000	CONTAIN	.LE.	0.000000
31-	DOOR	GRASS	3	ABOVE	.GT.	0.200000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
32-	DOOR	WALL	1	ABOVE	.LE.	0.000000	FEAT-1	DIFF_GT	20.00000	CONTAIN	.LE.	0.000000
33-	DOOR	WALL	2	ABOVE	.LE.	0.000000	FEAT-2	DIFF_GT	20.00000	CONTAIN	.LE.	0.000000
34-	DOOR	WALL	3	ABOVE	.LE.	0.000000	FEAT-3	DIFF_GT	20.00000	CONTAIN	.LE.	0.000000
35-	FENCE	FENCE	1	FEAT-1	DIFF_LT	5.000000	FEAT-2	DIFF_LT	5.000000	FEAT-3	DIFF_LT	5.000000
36-	FENCE	GRASS	1	BELOW	.GT.	0.500000	FEAT-1	DIFF_GT	25.00000	NONE	NONE	0.000000
37-	FENCE	GRASS	2	BELOW	.GT.	0.500000	FEAT-2	DIFF_GT	25.00000	NONE	NONE	0.000000
38-	FENCE	GRASS	3	BELOW	.GT.	0.500000	FEAT-3	DIFF_GT	25.00000	NONE	NONE	0.000000
39-	FENCE	ROAD	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
40-	FENCE	ROAD	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
41-	FENCE	ROAD	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
42-	FENCE	SHADOW	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
43-	FENCE	SHADOW	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
44-	FENCE	SHADOW	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
45-	GRASS	BUSHES	1	BELOW	.GT.	0.100000	FEAT-1	DIFF_GT	10.00000	NONE	NONE	0.000000
46-	GRASS	BUSHES	2	BELOW	.GT.	0.100000	FEAT-2	DIFF_GT	10.00000	NONE	NONE	0.000000
47-	GRASS	BUSHES	3	BELOW	.GT.	0.100000	FEAT-3	DIFF_GT	10.00000	NONE	NONE	0.000000
48-	GRASS	CAR	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	15.00000	ENCLOSE	.LE.	0.000000
49-	GRASS	CAR	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	15.00000	ENCLOSE	.LE.	0.000000
50-	GRASS	CAR	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	15.00000	ENCLOSE	.LE.	0.000000

RELATION CONSTRAINTS
****

DOMAINS: 13 ---> 24

	CONSTR_4	OPERAT_4	VALUE_4	CONSTR_5	OPERAT_5	VALUE_5	CONSTR_6	OPERAT_6	VALUE_6	CONSTR_7	OPERAT_7	VALUE_7
1-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
2-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
3-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
4-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
5-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
6-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
7-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
8-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
9-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
10-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
11-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
12-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
13-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
14-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
15-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
16-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
17-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
18-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
19-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
20-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
21-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
22-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
23-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
24-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
25-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
26-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
27-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
28-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
29-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
30-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
31-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
32-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
33-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
34-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
35-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
36-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
37-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
38-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
39-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
40-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
41-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
42-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
43-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
44-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
45-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
46-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
47-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
48-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
49-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
50-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000

RELATION CONSTRAINTS
****

DOMAINS: 1 ---> 12

	OBJ_X	OBJ_Y	NUMBER	CONSTR_1	OPERAT_1	VALUE_1	CONSTR_2	OPERAT_2	VALUE_2	CONSTR_3	OPERAT_3	VALUE_3
51-	GRASS	DOOR	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
52-	GRASS	DOOR	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
53-	GRASS	DOOR	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
54-	GRASS	FENCE	1	ABOVE	.GT.	0.500000	FEAT-1	DIFF_GT	25.00000	NONE	NONE	0.000000
55-	GRASS	FENCE	2	ABOVE	.GT.	0.500000	FEAT-2	DIFF_GT	25.00000	NONE	NONE	0.000000
56-	GRASS	FENCE	3	ABOVE	.GT.	0.500000	FEAT-3	DIFF_GT	25.00000	NONE	NONE	0.000000
57-	GRASS	GRASS	1	FEAT-1	DIFF_LT	12.00000	FEAT-2	DIFF_LT	12.00000	FEAT-3	DIFF_LT	12.00000
58-	GRASS	ROAD	1	ABOVE	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
59-	GRASS	ROAD	2	ABOVE	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
60-	GRASS	ROAD	3	ABOVE	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
61-	GRASS	SHADOW	1	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
62-	GRASS	SHADOW	2	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
63-	GRASS	SHADOW	3	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
64-	GRASS	TREE	1	BELOW	.GT.	0.300000	FEAT-1	DIFF_GT	12.00000	NONE	NONE	0.000000
65-	GRASS	TREE	2	BELOW	.GT.	0.300000	FEAT-2	DIFF_GT	12.00000	NONE	NONE	0.000000
66-	GRASS	TREE	3	BELOW	.GT.	0.300000	FEAT-3	DIFF_GT	12.00000	NONE	NONE	0.000000
67-	ROAD	CAR	1	BELOW	.GT.	0.300000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
68-	ROAD	CAR	2	BELOW	.GT.	0.300000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
69-	ROAD	CAR	3	BELOW	.GT.	0.300000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
70-	ROAD	FENCE	1	ABOVE	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
71-	ROAD	FENCE	2	ABOVE	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
72-	ROAD	FENCE	3	ABOVE	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
73-	ROAD	GRASS	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
74-	ROAD	GRASS	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
75-	ROAD	GRASS	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	NONNE	NONE	0.000000
76-	ROAD	ROAD	1	FEAT-1	DIFF_LT	10.00000	FEAT-2	DIFF_LT	10.00000	FEAT-3	DIFF_LT	10.00000
77-	ROAD	SHADOW	1	BELOW	.GT.	0.050000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
78-	ROAD	SHADOW	2	BELOW	.GT.	0.050000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
79-	ROAD	SHADOW	3	BELOW	.GT.	0.050000	FEAT-3	DIFF_LT	20.00000	NONE	NONE	0.000000
80-	ROOF	ROOF	1	FEAT-1	DIFF_LT	8.000000	FEAT-2	DIFF_LT	8.000000	FEAT-3	DIFF_LT	8.000000
81-	ROOF	SKY	1	BELOW	.GT.	0.300000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
82-	ROOF	SKY	2	BELOW	.GT.	0.300000	FEAT-2	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
83-	ROOF	SKY	3	BELOW	.GT.	0.300000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
84-	ROOF	TREE	1	ABOVE	.LE.	0.000000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
85-	ROOF	TREE	2	ABOVE	.LE.	0.000000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
86-	ROOF	TREE	3	ABOVE	.LE.	0.000000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
87-	ROOF	WALL	1	ABOVE	.GT.	0.100000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
88-	ROOF	WALL	2	ABOVE	.GT.	0.100000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
89-	ROOF	WALL	3	ABOVE	.GT.	0.100000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
90-	ROOF	WALL_SH.	1	ABOVE	.GT.	0.100000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
91-	ROOF	WALL_SH.	2	ABOVE	.GT.	0.100000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
92-	ROOF	WALL_SH.	3	ABOVE	.GT.	0.100000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
93-	SHADOW	CAR	1	BELOW	.GT.	0.100000	FEAT-1	DIFF_GT	20.00000	INTEN	.LT.	20.00000
94-	SHADOW	CAR	2	BELOW	.GT.	0.100000	FEAT-2	DIFF_GT	20.00000	INTEN	.LT.	20.00000
95-	SHADOW	CAR	3	BELOW	.GT.	0.100000	FEAT-3	DIFF_GT	20.00000	INTEN	.LT.	20.00000
96-	SHADOW	FENCE	1	ABOVE	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	INTEN	.LT.	20.00000
97-	SHADOW	FENCE	2	ABOVE	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	INTEN	.LT.	20.00000
98-	SHADOW	FENCE	3	ABOVE	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	INTEN	.LT.	20.00000
99-	SHADOW	GRASS	1	FEAT-1	DIFF_GT	20.00000	INTEN	.LT.	20.00000	NONE	NONE	0.000000
100-	SHADOW	GRASS	2	FEAT-2	DIFF_GT	20.00000	INTEN	.LT.	20.00000	NONE	NONE	0.000000

RELATION CONSTRAINTS
****
DOMAINS: 13 ---> 24

	CONSTR_4	OPERAT_4	VALUE_4	CONSTR_5	OPERAT_5	VALUE_5	CONSTR_6	OPERAT_6	VALUE_6	CONSTR_7	OPERAT_7	VALUE_7
51-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
52-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
53-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
54-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
55-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
56-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
57-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
58-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
59-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
60-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
61-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
62-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
63-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
64-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
65-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
66-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
67-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
68-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
69-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
70-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
71-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
72-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
73-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
74-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
75-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
76-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
77-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
78-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
79-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
80-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
81-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
82-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
83-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
84-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
85-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
86-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
87-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
88-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
89-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
90-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
91-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
92-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
93-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
94-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
95-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
96-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
97-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
98-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
99-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
100-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000

RELATION CONSTRAINTS
****

DOMAINS: 1 ---> 12

	OBJ_X	OBJ_Y	NUMBER	CONSTR_1	OPERAT_1	VALUE_1	CONSTR_2	OPERAT_2	VALUE_2	CONSTR_3	OPERAT_3	VALUE_3
101-	SHADOW	GRASS	3	FEAT-3	DIFF_GT	20.00000	INTEN	.LT.	20.00000	NONE	NONE	0.000000
102-	SHADOW	ROAD	1	FEAT-1	DIFF_GT	20.00000	INTEN	.LT.	20.00000	NONE	NONE	0.000000
103-	SHADOW	ROAD	2	FEAT-2	DIFF_GT	20.00000	INTEN	.LT.	20.00000	NONE	NONE	0.000000
104-	SHADOW	ROAD	3	FEAT-3	DIFF_GT	20.00000	INTEN	.LT.	20.00000	NONE	NONE	0.000000
105-	SHADOW	SHADOW	1	FEAT-1	DIFF_LT	5.000000	FEAT-2	DIFF_LT	5.000000	FEAT-3	DIFF_LT	5.000000
106-	SKY	CLOUDS	1	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
107-	SKY	CLOUDS	2	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
108-	SKY	CLOUDS	3	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
109-	SKY	ROOF	1	ABOVE	.GT.	0.300000	FEAT-1	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
110-	SKY	ROOF	2	ABOVE	.GT.	0.300000	FEAT-2	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
111-	SKY	ROOF	3	ABOVE	.GT.	0.300000	FEAT-3	DIFF_GT	15.00000	CONTAIN	.LE.	0.000000
112-	SKY	SKY	1	FEAT-1	DIFF_LT	5.000000	FEAT-2	DIFF_LT	5.000000	FEAT-3	DIFF_LT	5.000000
113-	SKY	TREE	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	12.00000	NONE	NONE	0.000000
114-	SKY	TREE	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	12.00000	NONE	NONE	0.000000
115-	SKY	TREE	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	12.00000	NONE	NONE	0.000000
116-	TREE	BUSHES	1	ABOVE	.GT.	0.300000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
117-	TREE	BUSHES	2	ABOVE	.GT.	0.300000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
118-	TREE	BUSHES	3	ABOVE	.GT.	0.300000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
119-	TREE	CLOUDS	1	ABOVE	.LE.	0.000000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
120-	TREE	CLOUDS	2	ABOVE	.LE.	0.000000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
121-	TREE	CLOUDS	3	ABOVE	.LE.	0.000000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
122-	TREE	GRASS	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	12.00000	NONE	NONE	0.000000
123-	TREE	GRASS	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	12.00000	NONE	NONE	0.000000
124-	TREE	GRASS	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	12.00000	NONE	NONE	0.000000
125-	TREE	ROOF	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
126-	TREE	ROOF	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
127-	TREE	ROOF	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
128-	TREE	SKY	1	ABOVE	.LE.	0.000000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
129-	TREE	SKY	2	ABOVE	.LE.	0.000000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
130-	TREE	SKY	3	ABOVE	.LE.	0.000000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
131-	TREE	TREE	1	FEAT-1	DIFF_LT	12.00000	FEAT-2	DIFF_LT	12.00000	FEAT-3	DIFF_LT	12.00000
132-	TREE	WALL	1	ABOVE	.GT.	0.100000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
133-	TREE	WALL	2	ABOVE	.GT.	0.100000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
134-	TREE	WALL	3	ABOVE	.GT.	0.100000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
135-	TREE	WALL_SH.	1	ABOVE	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
136-	TREE	WALL_SH.	2	ABOVE	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
137-	TREE	WALL_SH.	3	ABOVE	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
138-	WALL	BUSHES	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
139-	WALL	BUSHES	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
140-	WALL	BUSHES	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
141-	WALL	DOOR	1	BELOW	.LE.	0.000000	FEAT-1	DIFF_GT	20.00000	ENCLOSE	.LE.	0.000000
142-	WALL	DOOR	2	BELOW	.LE.	0.000000	FEAT-2	DIFF_GT	20.00000	ENCLOSE	.LE.	0.000000
143-	WALL	DOOR	3	BELOW	.LE.	0.000000	FEAT-3	DIFF_GT	20.00000	ENCLOSE	.LE.	0.000000
144-	WALL	ROOF	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
145-	WALL	ROOF	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
146-	WALL	ROOF	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
147-	WALL	TREE	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	15.00000	NONE	NONE	0.000000
148-	WALL	TREE	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	15.00000	NONE	NONE	0.000000
149-	WALL	TREE	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	15.00000	NONE	NONE	0.000000
150-	WALL	WALL	1	FEAT-1	DIFF_LT	12.00000	FEAT-2	DIFF_LT	12.00000	FEAT-3	DIFF_LT	12.00000

RELATION CONSTRAINTS
****
DOMAINS: 13 ---> 24

	CONSTR_4	OPERAT_4	VALUE_4	CONSTR_5	OPERAT_5	VALUE_5	CONSTR_6	OPERAT_6	VALUE_6	CONSTR_7	OPERAT_7	VALUE_7
101-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
102-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
103-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
104-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
105-	INTEN	.LT.	20.00000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
106-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
107-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
108-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
109-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
110-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
111-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
112-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
113-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
114-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
115-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
116-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
117-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
118-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
119-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
120-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
121-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
122-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
123-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
124-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
125-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
126-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
127-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
128-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
129-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
130-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
131-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
132-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
133-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
134-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
135-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
136-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
137-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
138-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
139-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
140-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
141-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
142-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
143-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
144-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
145-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
146-	ENCLOSE	.LE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
147-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
148-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
149-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
150-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000

RELATION CONSTRAINTS
****

DOMAINS: 1 ---> 12

	OBJ_X	OBJ_Y	NUMBER	CONSTR_1	OPERAT_1	VALUE_1	CONSTR_2	OPERAT_2	VALUE_2	CONSTR_3	OPERAT_3	VALUE_3
151-	WALL	WALL_SH.	1	FEAT-1	DIFF_GT	15.00000	FEAT-1	.GT.	35.00000	NONE	NONE	0.000000
152-	WALL	WALL_SH.	2	FEAT-2	DIFF_GT	15.00000	FEAT-2	.GT.	35.00000	NONE	NONE	0.000000
153-	WALL	WALL_SH.	3	FEAT-3	DIFF_GT	15.00000	FEAT-3	.GT.	35.00000	NONE	NONE	0.000000
154-	WALL	WINDOW	1	FEAT-1	DIFF_GT	25.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
155-	WALL	WINDOW	2	FEAT-2	DIFF_GT	25.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
156-	WALL	WINDOW	3	FEAT-3	DIFF_GT	25.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
157-	WALL_SH.	ROOF	1	BELOW	.GT.	0.100000	FEAT-1	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
158-	WALL_SH.	ROOF	2	BELOW	.GT.	0.100000	FEAT-2	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
159-	WALL_SH.	ROOF	3	BELOW	.GT.	0.100000	FEAT-3	DIFF_GT	12.00000	CONTAIN	.LE.	0.000000
160-	WALL_SH.	TREE	1	BELOW	.GT.	0.200000	FEAT-1	DIFF_GT	20.00000	NONE	NONE	0.000000
161-	WALL_SH.	TREE	2	BELOW	.GT.	0.200000	FEAT-2	DIFF_GT	20.00000	NONE	NONE	0.000000
162-	WALL_SH.	TREE	3	BELOW	.GT.	0.200000	FEAT-3	DIFF_GT	20.00000	NONE	NONE	0.000000
163-	WALL_SH.	WALL	1	FEAT-1	DIFF_GT	15.00000	FEAT-1	.GT.	20.00000	NONE	NONE	0.000000
164-	WALL_SH.	WALL	2	FEAT-2	DIFF_GT	15.00000	FEAT-2	.GT.	20.00000	NONE	NONE	0.000000
165-	WALL_SH.	WALL	3	FEAT-3	DIFF_GT	15.00000	FEAT-3	.GT.	20.00000	NONE	NONE	0.000000
166-	WALL_SH.	WALL_SH.	1	FEAT-1	DIFF_LT	5.000000	FEAT-2	DIFF_LT	5.000000	FEAT-3	DIFF_LT	5.000000
167-	WALL_SH.	WINDOW	1	FEAT-1	DIFF_GT	12.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
168-	WALL_SH.	WINDOW	2	FEAT-2	DIFF_GT	12.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
169-	WALL_SH.	WINDOW	3	FEAT-3	DIFF_GT	12.00000	NONE	NONE	0.000000	NONE	NONE	0.000000
170-	WINDOW	WALL	1	FEAT-1	DIFF_GT	12.00000	FEAT-2	DIFF_GT	12.00000	FEAT-3	DIFF_GT	12.00000
171-	WINDOW	WALL_SH.	1	FEAT-1	DIFF_GT	12.00000	FEAT-2	DIFF_GT	12.00000	FEAT-3	DIFF_GT	12.00000
172-	WINDOW	WINDOW	1	FEAT-1	DIFF_LT	5.000000	FEAT-2	DIFF_LT	5.000000	FEAT-3	DIFF_LT	5.000000

RELATION CONSTRAINTS
****
DOMAINS: 13 ---> 24

	CONSTR_4	OPERAT_4	VALUE_4	CONSTR_5	OPERAT_5	VALUE_5	CONSTR_6	OPERAT_6	VALUE_6	CONSTR_7	OPERAT_7	VALUE_7
151-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
152-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
153-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
154-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
155-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
156-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
157-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
158-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
159-	ENCLOSE	.LE.	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
160-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
161-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
162-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
163-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
164-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
165-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
166-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
167-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
168-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
169-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
170-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
171-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000
172-	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000	NONE	NONE	0.000000

REFERENCES

[1] Aggrawal, J.K., and Martin, W.N., Survey: Dynamic Scene Analysis, Computer Graphics and Image Processing, Vol 7, 1978, pp. 356-374.

[2] Arbib, M.A., Parallelism, Slides, Schemas, and Frames, in Systems: Approaches, Theories, Applications, Hartnett, W., (Ed.), Reidel, D., Publishing Co., 1977, pp. 27-43.

[3] Arbib, M.A., Two papers on Schemes and Frames, COINS Technical Report 75 C - 9, Computer and Information Science, University of Massachusetts at Amherst, October, 1975.

[4] Arbib, M.A., The Metaphorical Brain: An Introduction to Cybernetics as Artificial Intelligence and Brain Theory, Wiley-Interscience, 1972.

[5] Agin, G.J., Artificial Intelligence-Research and Applications, Nilsson (Ed.), report on contract DAHC04-75-0005, Artificial Intelligence Center, Menlo Park, California, May 1975, pp. 90-100.

[6] Baird, M.L., and Kelley, M.D., A paradigm for picture recognition, Pattern Reecognition 6, 1974, pp. 61-74.

[7] Bajcsy, R. Computer Description of Textured Surfaces, Proc. Third International Joint Conference on Artificial Intelligence, Stanford, August, 1973, pp. 572-579.

[8] Ballard, D.H., Brown, C.M., and Feldman, J.A., An Approach to Knowledge-Directed Image Analysis, in Computer Vision Systems, A. Hanson and E. Riseman (Eds.), Academic Press, New York, 1978.

[9] Ballard, D.H., Model-Directed Detection of Ribs in Chest Radiographs, TR-11 Dept. of Computer Science, University of Rochester, New York, 1978.

[10] Ballard, D., and Sklansky, J., Tumer Detection in Radiographs, Tech. Report TR-72-7, School of Engineering, University of California, Irvine, October 1972.

[11] Barrow, H.G., and Tenenbaum, J.M., Recovering Intrinsic Scene Characteristics from Images, in Computer Vision System, A. Hanson and E. Riseman (Eds.), Academic Press, New York, 1978.

[12] Barrow, H.G., Tenenbaum, J.M., MSYS: A System for Reasoning About Scenes, Technical Report 121, A.I. Center, Stanford Research Institute, Stanford, California, 1976.

[13] Barrow, H.G., Tenenbaum, J.M., Representation and Use of Knowledge in Vision, SIGART Newsletter NO. 57, June 1975, pp. 2-9.

[14] Barrow, H.G., and Popplestone, R.J., Relational Descriptions in Picture Processing, Machine Intelligence 6, Meltzer, B. and Michie, D. (Eds.), University Press Edinburgh, 1971.

[15] Bell, C.J., A Relational model for Information Retrieval and the Processing of linguistic Data, IBM Research Report RC1705, Yorktown, N.Y., 1966.

[16] Brice, C.R., and Fernema, C.L., Scene Analysis Using Regions, Artificial Intelligence 1, 1970.

[17] Casey, R.G. and Delobel, C., Decomposition of a Data Base and the Theory of Boolean Switching Functions. IBM J. R & D 17, No 5, September 1973.

[18] Clowes, M.B., on Seeing Things, AI Journal, Spring 1971.

[19] Clowes, M.B., Perception Picture Processing and Computers, Machine Intelligence 1, Collins, B. and Michie, D. (Eds.), American Elserier, N.Y., 1967, pps. 181-197.

[20] Codd, E.F., Relational Completeness of Data Base Sublanguages, In Data Base Systems, Courant Computer Science Symposium Series, Vol 6, Prentice-Hall 1972.

[21] Codd, E.F., Further Normalization of The Data Base Relational Model, In Data Base Systems, Courant Computer Science Symposia Series, Vol 6, Prentice-Hall 1972.

[22] Codd, E.F., Normalized Data Base Structure: A Brief Tutorial, Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access and Control, Available from ACM, 1971.

[23] Codd, E.F., A Data Base Sublanguage Founded on the

Relational Calculus, Proc. 1971 ACM SIGFIDET Workshop on Data Description, Access and Control, Available from ACM, 1971.

[24] Codd, E.F., A Relational Model for Large Shared Data Banks, CACM 13, No 6, June 1970.

[25] Date, C.J., An Introduction to Database Systems, Addison-Wesley Publishing Co. Inc., Reading, Massachusetts, 1975.

[26] Date, C.J, and Codd, E.F., The Relational and Network Approaches: Comparison of the Application Programming Interfaces, Proc. ACM-SIGMOD Debate " Data Models: Data Structure Versus Relational", Ann Arbor, May 1974, pp. 1-3.

[27] Date, C.J. and Hopewell, P., Storage Structure and Physical Data Independence, Proceedings of ACM SIGFIDET Workshop on Data Description, Access and Control, Available from ACM, 1971.

[28] Dev, P., Segmentation Process in Visual Perception: A Cooperative Neural Model, Int. J. Man-Machine Studies, 1975.

[29] Diddy, R.L., The Similation and Modeling of Distributed Information Processing in the Froog Visual System, Ph. D. Thesis, Stanford University, 1970.

[30] Duda, R.O., and Hart, P.E., Pattern Recognition and Scene Analysis, John-Wiley, New York, 1973.

[31] Engles, R.W., A Tutorial on Data Base Organization,

Annual Review in Automatic Programming Vol 7, Part 1 Halpern and McGee (Eds.). Pergamon Press, July 1972.

[32] Erman, L., and Lesser, V., System Engineering Techniques for Artificial Intelligence Systems, in Computer Vision Systems, Hanson and Riseman (Eds.), Academic Press, 1978.

[33] Erman, L., and Lesser, V., A Multi-Level Organization for Problem-Solving using Many, Diverse, Cooperating Sources of Knowledges, Proc. Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR, September 1975, pp. 483-490.

[34] Falk, G., Computer Interpretation of Imperfect Line-Data As A Three-Dimensional Scene, Stanford AI Memo AIM-132, Stanford University, August 1970.

[35] Freeman, J., The modelling of spatial relations, Computer Graphics and Image Processing 4, 1975, pp. 156-171.

[36] Freeman, H., Computer Processing of Line-Drawing Images, Computing Surveys, Vol. 6, 1974, pp. 57-97.

[37] Garvey, T.D.,and Tenenbaum, J.M., On the Automatic Generation of Programs for Locating Objects in Office Scenes, Proc. Second International Joint Conference on Pattern Recognition, Augest, 1974,pp. 162-168.

[38] Grape, G.R., Model Based (Intermediate Level) Computer Vision, Stanford AI Laboratory, Memo AIM-201, Computer Science Dept., Report CS-73-366, April 1973.

[39] Gupta, J.N., and Wintz P.A., A Boundary Finding Algorithm and its Applications, IEEE Trans. on Circuits and Systems, Vol. CAS-22,NO. 4, April 1975.

[40] Guzman, A., Computer Recognition of Three-Dimensional Objects In A Visual Scene, MAC-TR-59, MIT Project MAC, December 1968.

[41] Guzman, A., Decomposition Of A Visual Scene Into 3-D Bodies, AFIPS 1968 Fall-Joint Computer Conference, 33, part 1, 1968, pps. 291-304.

[42] Hanson, A. and Riseman, E., Computer Vision Systems, Hanson, A. and Riseman, E. (Eds.), Academic Press, New York, 1978.

[43] Hanson, A. and Riseman, E., Segmenting natural Scenes, in Computer Vision Systems, Hanson, A. and Riseman, E. (Eds.), Academic Press, New York, 1978.

[44] Hanson, A. and Riseman, E., VISIONS: A Computer System for Interpreting Scenes, in Computer Vision Systems, Hanson, A. and Riseman, E. (Eds.), Academic Press, New York, 1978.

[45] Harlow, C.A., Image Analysis and Graphs, Technical Report IAL - TR 17 - 72, Image Analysis Laboratory, University of Missouri-Columbia, Columbia, Missouri, September 1972.

[46] Heath, I.J., Unacceptable File Operations in a Relational Data Base, Proc. ACM-SIGFIDET Workshop on Data Description, Access and Control, 1971.

[47] Holland S.W., A Programable Computer Vision System Based on Spatial Relationships, Research Publications GMR-2078, Research Laboratories, General Motors Corporation, Warren, Mich., 1976.

[48] Horowitz, S.L., and Pavlidis, T., Picture Segmentation by a Directed Split-and-Merge Procedure, Proc. Second International Joint Conference on Pattern Recognition, 1974, pp. 497-498.

[49] Huffman, D.A., Logical Analysis of Pictures of Polyhedra, AI group, Technical Note No 6, SRI Project 7494, May 1969.

[50] Huffman, D.A., Impossible Object as Nonsense Sentences, Machine Intelligence 6, Meltzer, B. and Michie, D. (Eds.), American Elserier, New York, 1971, pps. 295-323.

[51] Information Management System/360 Version 2 General Information manual, IBM Form No.GH20-0765.

[52] Jain, R., Militzer, D., and Nagel, H., Separating Non-Stationary From Stationary Scene Components In A Sequence Of Real World TV-Images, Proc. Fifth International Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, August 1977.

[53] Kanade, T., Region Segmentation: Signal vs. Semantics, Proc. Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan, 1978.

[54] Kanade, T., Model Representation and Control

Structure in Image Understanding, Proc. Fifth International Joint Conference on Artificial Intellignce, Cambridge, MA, August 1977.

[55] Kelley, M.D., Edge Detection in Pictures by Computers Using Planning, Machine Intelligence, 6, 1971, pp. 379-409.

[56] Kelley, M.D., Visual Identification of People by Computers, Memo AI-130A, Computer Science Dept., Stanford, Ca., July 1970.

[57] Kender, R.R., Saturation, Hue, and Normalized Color; Calculation, Digitization Effects, and Use, Technical Report, Dept. of Computer Science, Carnegie-Melon University, November 1976.

[58] Kilmer, W., and Blum, J., A Model of the Verterbrate Central Command System, Int. J. Man-Machine Studies, No 1, pp. 279-305.

[59] Kirsch, R.A., Computer Determination of the Constituent Structure of Biological Images, Computers and Biomedical Research, 4, 1971, pp. 315-328.

[60] Kuipers, B.J., An Hypothesis Driven Recognition System for the Block World, Second Joint Conference on Pattern Recognition, Copenhagen, Denmark, August 1974, pp. 169-173.

[61] Ledley, R.S., High speed automatic analysis of bio-medical pictures, Science 146, 1964, pp. 216-223.

[62] Lesser, V., and Erman, L., A Retrospective View of

the HEARSAY II Architecture, Proc. Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977, pp. 790-800.

[63] Lesser, V.R., Fennel, R. D., Erman, L.D., and Reddy, D.R., Organization of the Hearsay-II Speech Understanding System, IEEE Trans. on ASSP, Vol. 23, 1975, pp. 11-23.

[64] Levein, R.E. and Maron, M.E., A Computer System for Inference Execution and Data Retrieval, CACM 10, No 11, November 1967.

[65] Levine, M.D., and Shaheen, S.I., A Modular Computer Vision System for Picture Segmentation and Interpretation, Proc. of the IEEE Conference on Pattern Recognition and Image Processing, Chicago, Augest(6-8), 1979.

[66] Levine, M.D., A Knowledge-Based Computer Vision System, in Computer Vision Systems, Allen R. Hanson & Edward M. Riseman (Eds.), Academic Press, New York, 1978, pp. 336-352.

[67] Levine, M.D., On the Segmentation Process in Scene Analysis, Report No. 73-25, McGill University, Montreal, P.Q., August 1973.

[68] Levine, M.D., O'Handley, D.A., and Yagi, G.M., Computer determination of depth maps, Computer Graphics and Image Processing, 2, 1973, pp. 131-150.

[69] Lieberman, L., Computer Recognition and Description of Natural Scenes, Moore School Technical Report No

74-08, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia 1974.

[70] Mackworth, A.K, Consistency of Netwoorks of Relations, Tech. Report 75-3, Dept. of Computer Science, University of British Columbia, July,1975.

[71] Mackworth, A.K., Interpreting Pictures of Polyhedral Scenes, Proc. Third International Joint Conference on Artificial Intelligmce, Stanford Research Institute, Menlo Park, CA, 1973, pps. 556-563.

[72] Minsky, M., A Framework for Representing Knowledge, The Psychology of Computer Vision, Winston, P. (Ed.), McGraw-Hill, 1975.

[73] Montanari, V., A Note on Minimal Length Polygonal Approximation to a Digitized Contour, Communication of the ACM, Jan. 1977,pp. 41-47.

[74] Montanari, V., Networks of Constraints: Fundamental Properties and Applications too Picture Processing, Information Sciences, Vol.7, 1974, pp. 95-132.

[75] Muerle, J.L.,and Allen D.C., Experimental Evaluation of Techniques for Automatic Segmentation of Objects in Complex Scene, in Pictorial Pattern Recognition, C.G. Cheng (Eds.), Thompson, Washington,1968.

[76] Nagel, H.H., Analysing Sequences of TV-Frames, Proc. Fifth International Joint Conference on Artificial Intelligmce, Massachusetts, August 1977.

[77] Nagin, P.A., Hanson, A.R., and Riseman, E.M., Region

Extraction and Description Through Planning, COINS Technical Report 77-8, University of Massachusetts, May 1977.

[78] Nazif, A., A Survey of Color, Boudary Information and Texture as Features for Low level Image Processing, McGill Technical Report 78-7R, 1978.

[79] Neisser, U., Cognition and Reality: Principles and Implications of Cognitive Psychology, W.H. Freeman and Company, 1976.

[80] Newell, A., and Reddy, D., Image Understanding: Some Notes, Minitues ARPA, Image Understanding Workshop, Science Applications Inc., March 1975.

[81] Nilsson, N.J., Artificial Intelligence, Information Processing Proceedings of IFIP Congress, 1974, pp. 778-801.

[82] Ohlander, R.B., Analysis of Natural Scenes, Ph. D. thesis, Department of Computer Science, Carnegie-Mellon, University Pittsburg, Pennsylvania, 1975.

[83] Orban, R., Removing Shadows in a Scene. AI Memo 192 MIT AI Laboratory, August 1970.

[84] Piaget, J., Biology and Knowledge: An Essay on the Relations and Cognitive Processes, Edinburgh University Press, 1971.

[85] Pingle, K.K., and Tenenbaum, J.M., An Accommodating Edge Follower, Proc. Second International Joint

Conference on Artificial Intellignce, British Computer Society, London, 1971, pp. 1-7.

[86] Rattener, M.H., Exending Guzman's SEE Program, AI Laboratory, Memo 204, MIT Project MAC, Cambridge, Massachusetts, 1970.

[87] Reddy, R., and Akin, O., Knowledge Acquisition for Image Understanding Research, Computer Graphics and Image Processing 6, 1977, pp. 307-334.

[88] Riseman, E.M. and Arbib, M.A., Computational Techniques in Visual Segmentation of Static Scenes, Computer Graphics and Image Processing Vol 6, June 1977, pp. 221-276.

[89] Riseman, E.M. and Hanson, A.R., Processing Cones: A Computational Structural for Scene Analysis, COINS Technical Report 74C-7, Department of Computer & Information Science, University of Massachusetts, Amherst, 1974.

[90] Riseman, E., and Hanson, A., the Design of a Semantically Directed Vision Processor, COINS Technical Report 74C-1, University of Massachusetts, January 1974.

[91] Rissanen, J., and Delobel, C., Decomposition of Files, A Basis for Data Storage and Retrieval, IBM Research Report 1220, May 1973.

[92] Roberts, L.G., Machine Perception of 3-Dimensional Solids, Optical and Electro Optical Information

Processing, Tippett, et al (Eds.), MIT Press, Cambridge, Massachusetts, 1965, pp. 159-197.

[93] Roberts, L.G., Machine Perception of Three-Dimensional Solids. Technical report No 315, MIT Lincoln Laboratory, May 1963 (Reissued May 1965).

[94] Rosenfeld, A., Picture Processing: 1976, TR-500, Computer Science Center, University of Maryland, College Park, Md., 1977.

[95] Rosenfeld, A., Picture Processing: 1975, TR-433, Computer Science Center, University of Maryland, College Park, Md., 1976.

[96] Rosenfeld, A., A Note on Automatic Detection of Texture Gradients, IEEE Transactions On Computers, C-24, Number 10, October, 1975.

[97] Rosenfeld, A., Progress in Picture Processing: 1969-1971, COMPUTING SURVEYS 5, June, 1973.

[98] Rosenfeld, A., Compact Figures in Digital Pictures, Tech. Report TR-173, March 1973.

[99] Rosenfeld, A., Picture Processing: 1972, COMPUTER GRAPHICS AND IMAGE PROCESSING 1, December, 1972.

[100] Rosenfeld, A., Picture Processing by Computer, COMPUTER SURVEYS 1, September, 1969.

[101] Rosenfeld, A., Picture Processing by Computer, Academic Press, New York, 1969.

[102] Rosenfeld, A., Hummel, R. and Zucker, S., Scene Labelling by Relaxation Operations, IEEE Trans.

Systems, Man and Cybernetics, 1976, SMC-6, pps.420-433.

[103] Rosenfeld, A. and Davis, L.S., Iterative Histogram Modification, TR-519, Computer Science Center, University of Maryland, April 1977.

[104] Sakai,T., Kanade, T., and Ohta, Y., Model-Based Interpretation of Outdoor Scene, Proc. Third International Joint Conference on Pattern Recognition, 1976.

[105] Schank, R.C., and Abelson, R.P., Goals, Plans, Scripts, and Understanding: An Enquiry into Human Knowledge Structures, Erlbaum Press, NJ, 1977.

[106] Shaheen, S.I., Levine, M.D., Image Segmentation and Interpretation Using a Knowledge Database, Second National Conference of the Canadian Society for Computational Studies of Intelligence, Toronto, Ontario, 19-21 July, 1978.

[107] Shaheen, S.I., An Implementation of a Relational Database and An Algebraic Sublanguage for Computer Vision System, Tech. Report No. 78-12R, Dept. of Electrical Engineering, McGill University, Montreal, 1978.

[108] Shaheen, S.I., User Manual for MIPS Relational Database,Department of Electrical Engineering, McGill University, 1977.

[109] Shirai, Y., Analyzing Intensity Arrays Using

Knowledge About Scenes, The Psychology of Computer Vision, Winston, P. (Ed.), McGraw Hill, New York, 1975.

[110] Tanimoto, S., Pavlidis, T., A Hierarchical Data Structure for Picture Processing, Computer Graphics and Image Processing, Vol. 4, 1975, pp.104-119.

[111] Tasto, N. and Block, U., Locating Objects in Complex Scenes Using a Spatial Distance Measure, Proc. Second International Joint Conference on Pattern Recognition, Augest 1974, pp. 336-340.

[112] Tenenbaum, J.M., On Locating Objects by Their Distinguishing Features in Multi-sensory Images, SRI Technical Note 84, AI Center, Stanford Research Institute, September 1973.

[113] Tenenbaum, J.m., Barrow, H.G., Experiments in Interpretation-Guided Segmentation, Artificial Intelligence, Vol. 8, No. 3, 1977, pp. 241-274.

[114] Tenebaum, J.M., Garvey, T.D., Weyl, S., and Wolf, H.C., An Interactive Facility for Scene Analysis Research, SRI Technical Report 87, 1974.

[115] Tidhar, A., Using a Structured World in Flexible Recognition of Two Dimensional Patterns. Moore School Technical Report No 75-02, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia.

[116] Ting, D., Intermediate Level Processing For a Computer Vision System, M.Eng. Thesis, Department of

Electrical Engineering, McGill University, Montreal, 1979.

[117] Ting, D., Feature Calculation and Storage, Mips Memo 76-2, Dept. of Electrical Engineering, McGill University, Montreal, May 1976.

[118] Tomita, F., Yashida, Y., and Tsuji, S., Detection of Homogeneous Regions by Structural Analysis, Proc. Third International Joint Conference on Artificial Intelligence, Stanford, California, August 1973, pp. 564-571.

[119] Uhr, L., Recognition Cones and Some Test Results, in Computer Vision Systems, Hanson, A. and Riseman, E. (Eds.), Academic Press, New York, 1978.

[120] Uhr, L., Layered Recognition Cone Networks That Preprocess, Classify, and Describe, IEEE Trans. Computers, 1972, pp. 758-768.

[121] Waltz, D., Understanding Line Drawings of Scenes with Shadows, in the Psychology of Vision, McGraw-Hill, 1975.

[122] Waltz, D.L., Generating Semantic Descriptions from Drawings of Scenes with Shadows. AI TR-271, MIT AI laboratory, November 1972.

[123] Watanabe, S., Knowing and Guessing, A Quantitative Study of Inference Information, John Wiley & Sons Inc., New York, 1969.

[124] Winston, P.H., The Psychology of Computer Vision,

McGraw-Hill, 1975.

[125] Winston, P.H., Learning Structure Description from Examples Mac-TR-76, MIT Project MAC, September 1970.

[126] Yakimovsky, Y., and Feldman, J.A., Decision-theory and artificial intelligence: a semantic-based region analyzer, Artificial Intelligence 5, 1974, pp. 349-371.

[127] Yakimovsky, Y., Feldman, J.A., A Semantics-Based Decision Theory Region Analyzer, Proc. Third International Joint Conference on Artificial Intelligence, Stanford University, 1973, pp. 580-588.

[128] Yashida, and Tsjui, S, A Machine Vision For Complex Industrial Parts With Learning Capability, Advanced papers of the Proc. Fourth International Joint Conference on Artificial Intelligence, September 1975, pp. 819-862.

[129] Zucker, S.W., Vertical and Horizontal Processes in Low Level Vision, in Computer Vision Systems , Hanson and Riseman (Eds.), Academic Press, New York, 1978.

[130] Zucker, S.W., Relaxation Labelling and the Reduction of Local Ambiguities, in Pattern Recognition and Artificial Intelligence, C.H. Chen (Ed.), Academic Press, New York, 1977.

[131] Zucker, S.W., Hummel, R.A., and Rosenfeld, A., An Application of Relaxation Labelling to Line and Curve Enhancement, IEEE Trans. Comp., 1977, C-26, pps.

394-403, and 922-929.

[132] Zucker, S.W. and Krishnamurthy, E.V., and Haar, R., Relaxation Processes for Scene Labeling: Convergence, speed and Stability. IEEE Trans. Systems, Man, and Cybernatics, 1978, SCMC-8, pp.41-48.

[133] Zucker, S.W., Leclerc, Y.G., Mohammed, J.L., Continuous Relaxation and Local Maxima Selection, Conditions for Equivalence, Report No. 78-15R, Department of Electrical Engineering, McGill University, Montreal, December 1978.

[134] Zucker, S.W. and Mohammed, J.L., A Hierarchical System for Line Labelling and Grouping, IEEE Conf. on Image Processing and Pattern Recognition, Chicago, 1978.

[135] Zucker, S.W., and Mohammed, J.L., Analysis of Probabilistic relaxation Labelling Processes, IEEE Conf. on Image Processing and Pattern Recognition, Chicago, 1978.

[136] Zucker, S., Region Growing: Childhood and adolescence, TR-370, Computer Science Dept., University of Maryland, April 1975.