

AD-767 695

SCENE ANALYSIS USING A SEMANTIC BASE FOR
REGION GROWING

Yoram Yakimovsky

Stanford University

Prepared for:

Advanced Research Projects Agency

June 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM-209
STAN-CS-73-380

AD 767695

SCENE ANALYSIS USING A SEMANTIC
BASE FOR REGION GROWING

by

Yoram Yakimovsky

SUPPORTED BY

ADVANCED RESEARCH PROJECTS AGENCY
ARPA ORDER NO. 457

June 1973

Reproduced by
NATIONAL TECHNICAL
INFORMATION SERVICE
U S Department of Commerce
Springfield VA 22151

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY



B D C
OCT 15 1973
REGISTERED
E

135
r

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Stanford University
Department of Computer Science
Stanford, California 94305

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

SCENE ANALYSIS USING A SEMANTIC BASE FOR REGION GROWING

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

technical report, June 1973

5. AUTHOR(S) (First name, middle initial, last name)

Yoram Yakimovsky

6. REPORT DATE

June 1973

7a. TOTAL NO. OF PAGES

133

7b. NO. OF REFS

35

8a. CONTRACT OR GRANT NO.

SD-183

b. PROJECT NO.

ARPA ORDER NO. 457

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)

STAN-CS-73-380

AIM-209

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

DISTRIBUTION STATEMENT A

10. DISTRIBUTION STATEMENT

Releasable without limitations on dissemination

Approved for public release;
Distribution Unlimited

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

The problem of breaking an image into meaningful regions is considered. A probabilistic semantic basis is effectively integrated with the segmentation process, providing various decision criteria. Learning facilities are provided for interactively generating the Bayesian probabilistic basis. A programming system which is based on these ideas and its successful application to two problem domains are described.

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY
MEMO AIM-209

JUNE 1973

COMPUTER SCIENCE DEPARTMENT REPORT
STAN-CS-73-380

SCENE ANALYSIS USING A SEMANTIC BASE FOR
REGION GROWING

by

Yoram Yakimovsky

ABSTRACT: The problem of breaking an image into meaningful regions is considered. A probabilistic semantic basis is effectively integrated with the segmentation process, providing various decision criteria. Learning facilities are provided for interactively generating the Bayesian probabilistic basis. A programming system which is based on these ideas and its successful application to two problem domains are described.

This research was supported in part by the Advanced Research Projects Agency of the Office of Defense under Contract No. SD-183.

The views and conclusions in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Reproduced in the USA. Available from the National Technical Information Service, Springfield, Virginia 22151.

ACKNOWLEDGMENTS

I wish to express my thanks to Professor Jerome Feldman for his invaluable help and advice.

I would like to thank Dr. Thomas Binford and Professor John McCarthy for their suggestions.

Many ideas presented in this work arose out of discussions with fellow workers, in particular with Marty Tenenbaum and Randy Davis.

I would like to acknowledge the contributions from many people at the Stanford Artificial Intelligence Project and in particular from Jim Low who helped significantly in the software implementation.

I am indebted to Barbara Barnett who patiently spent hours to bring this thesis into a readable form.

This work was submitted to the Department of Computer Science and the Committee on Graduate Studies of Stanford University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

TABLE OF CONTENTS

SECTION	PAGE
1.0 INTRODUCTION	
1.1 GUIDE	1
1.2 THE SEGMENTATION PROBLEM IN A.I.	4
1.3 METHODOLOGY	6
1.4 HISTORY AND LITERATURE REVIEW	9
2.0 REGION GROWING - CLUSTERING BY MELTING WEAKEST-BOUNDARY-FIRST	
2.1 OVERVIEW	14
2.2 INITIALIZATION	16
2.3 WEAKEST-BOUNDARY-FIRST REGION GROWER	20
2.4 THE REGION MELTER	22
2.5 THE BOUNDARY STRENGTH LIST	22
2.6 SOME IMPLEMENTATION DETAILS	23
2.7 STOPPING CRITERIA	24
2.8 EDGE FOLLOWING	25
3.0 REGION GROWING WITHOUT USING SEMANTICS	
3.1 PLACEMENT OF THE SAMPLE POINTS	35
3.2 LOCAL FEATURE DETECTORS	39
3.3 BOUNDARY STRENGTH EVALUATION	42
3.4 BOUNDARY STRENGTH EVALUATION THROUGH EDGE	46
	FOLLOWING
3.5 STATE EVALUATION	48
3.6 MAINTENANCE OF REGION AND BOUNDARY PROPERTIES	50
3.7 FINAL COMMENTS ON THE NON SEMANTIC REGION	

TABLE OF CONTENTS

	GROWER	52
4.0	WORLD MOEEL	
4.1	THE STATISTICAL PROBLEM DEFINITION	56
4.2	THE BASIC ASSUMPTION	59
4.3	THE TASK OF THE WORLDO MOEEL	65
4.4	EXTENSIONS OF MOEEL TO 3-0	67
4.5	EXTENSION OF MOEEL TO INCLUDE GLOBAL	
	CONTEXT PARAMETERS	70
4.6	EXTENDED FIRST ORDER	71
5.0	WEAKEST-BOUNDARY-FIRST REGION GROWER BASED ON SEMANTICS	
5.1	SEMANTICS BOUNOARY STRENGTH EVALUATION	73
5.2	SEMANTIC STOPPING CRITERIA	76
6.0	STATE EVALUATION FOR A GIVEN IMAGE SEGMENTATION	
6.1	INTERPRETATION OF THE SCENE - LOWER BOUND	
	EVALUATION	77
6.2	REGION GROWING BASED ON ASSIGNMENT	83
6.3	UPPER BOUND FOR STATE VALUE	84
7.0	THE PROBABILITY SYSTEM IMPLEMENTATION	
7.1	GENERAL CONSIDERATION	86
7.2	THE PROBABILITY APPROXIMATION TECHNIQUE	88
7.3	THE CLASSIFICATION TREE	91
7.4	PROBABILITY COMPUTATION	95
7.5	LEARNING: PROBABILITY ADJUSTMENT	97
7.6	LEARNING: TREE GENERATION	98

TABLE OF CONTENTS

7.7	LEARNING: GENERATING NEW CLASSIFIERS	100
7.8	NEAREST NEIGHBOR CONSIDERATIONS	101
8.0	RESULTS	
8.1	PROBLEM DOMAINS	103
8.2	REGION PROPERTIES	105
8.3	BOUNDARY FEATURES	107
8.4	RESULTS	110
8.5	CONCLUSIONS	113
8.6	ILLUSTRATIONS	115
	BIBLIOGRAPHY	123

LIST OF FIGURES

FIGURE	PAGE
2.1 Initialization Structure	28
2.2 Example of Structure After a Few Region Growing Iterations	29
2.3 Example Of Effect of Merging Step	30
2.4 Example of Regions and Boundaries Structure	31
2.5 The effect of merge on data structure	32
2.6 Some Special Cases For Melter	33

SECTION 1

INTRODUCTION

1.1 GUIDE

The reader is advised to start reading this paper by briefly reviewing the illustrations of some examples of picture processing on pages 115-122, and returning to this point. The illustrations are pictures taken by a Polaroid camera from a television monitor, then processed to get negatives which are used to generate plates. These plates are then used for offset printing of the illustrations (that is why they are so "sharp"). The white lines are overlaid by the program on the original picture. These lines represent the boundaries between regions that exist in the program's segmentation of the image. The programming system was applied to two problem domains. The first domain was images of the type shown in illustrations A through E which are road scenes. The second domain was left ventricular angiograms illustrated in F and G. (A-7), (B-5), (C-4), (D-5), (F-5) and (G-3) are examples of the desired output. These are images segmented the way humans would segment them while

trying to describe them in the specific context of the problem domain. The achievement of our system is that this segmentation was done automatically after the program was taught on the general problem domain (semantics). Even though it is not apparent in the images, the program also understands the segmented images properly. That is, it assigns the same interpretation as humans assigns to the regions. The captions of the different images will give the reader some idea of the terms used in this paper and the problem domains. Illustrations (B-1) through (B-5) show the different stages of processing (problem reduction steps). (A-2), (A-4), (A-5) and (D-3) are examples of possible errors resulting from carrying any of the problem reduction steps beyond their proper stopping criteria.

Since this paper describes the implemented system, the ideas are usually presented in the order of their application to the system. Section 1 is an introduction to the image processing problem domain. Section 2 describes both the general data structures and the flavor of region growers in general, particularly the weakest-boundary-first region grower. Section 3 is a detailed description of the initialization and reduction of the problem by region growing without semantics. Section 4 starts by redefining the problem in statistical terms and continues by describing the assumptions and structure of the semantics representation. Section 5 describes the application of

the semantics to weakest-boundary-first region growing. Section 6 is devoted to describing an interpretation algorithm which is applied on the segmented image to assign meaning to the regions. Extensions of this algorithm which drive a region grower, evaluate the partition and provide for stopping criteria are described as well. Section 7 describes the method which we adopted for collecting the probabilistic knowledge on the problem domain. Section 8 describes specific feature detectors available for regions and boundaries, as well as the results of applications of the whole system to two problem domains.

Most of the ideas presented in this paper were implemented in the programming system, but some of them are included as suggestions for future research and development. Since these suggestions are scattered in the paper we note them explicitly here. Subsection 3.2 suggests improvements in local feature detectors and texture operators. Subsections 2.8 and 3.6 suggest using edge-following to achieve accurate shape contour and improvement in the existing shape description capabilities. Subsections 3.3 and 3.5 call for evaluation of the quality of various general boundary strength and state evaluation procedures. In subsections 4.4 and 4.6 extended representations of the semantics are suggested for implementation. Section 6.2 describes extensions of the meaning assignment algorithm

to drive a region grower with backup capabilities. Subsections 7.6-7.8 contain ideas for various aspects of automation of the learning which should be implemented to increase the effectiveness of the classification.

1.2 THE SEGMENTATION PROBLEM IN A.I.

The problem of segmentation, breaking a complex image into sections, is a central problem in machine perception. The analogous problem arises in the analysis of speech [VIC] and, for that matter, in any problem of overwhelming size. We will concentrate on the image segmentation problem, but most of the ideas are of wider applicability. The main ideas are the application of Bayesian decision theory techniques and the use of problem-dependent information (semantics) to attack the image segmentation problem.

The theory and implementation of a picture processing system which utilizes semantics will be described in this thesis. The segmentation process for pictures means breaking the picture area into regions fitting each other in a jig-saw puzzle sense. The interpretation of the segmented picture means naming (assignment of

meaning to) the different regions. In addition to interpreting regions, boundaries and vertices will be interpreted. The naming of a region means at least identifying the 3-D (three dimensional) surface for which that region is part of the image in the current scene. For boundaries the interpretation will be the 3-D structure associated with it (in addition to naming it as a boundary between the two interpretations of the regions defining it).

The segmentation problem for television images is as follows: given a picture of some scene, we have a rectangular grid composed of some 200x300 points and for each point some information about the light intensity and perhaps color. For any further processing 60000 points are far too many; depending on the perception task that we have in mind, the image should be segmented into regions. That is, the 60000 grid points should be clustered into relatively few regions, where each of these regions should be meaningful in the problem domain and the relevant information needed for the specific task should be easily obtainable. Meaningful segmentation for us means that each of the resulting regions may be named as being one of the regions known to the system a priori (like sky, grass, road, etc.), and the properties of the resulting segmentation structure will match the properties expected of that structure given the interpretation and a priori knowledge of the system about the problem domain. More rigorous definition of the problem will be given in Subsection 4.1.

1.3 METHODOLOGY

In the past, segmentation and interpretation were executed in two levels of programs, a low level and a high level. The interaction between the two levels was done on the basis of failure interrupts. The low level portion segmented the input. The high level tried to make sense of the segments produced by the low level. In case of difficulty in the high level, the low level was recalled to resegment the troubled portion of the picture with a different set of parameters. Certain limited success has been achieved utilizing that approach [ROS]. Some meteorological images can be segmented effectively using such techniques. However, for images like those arising in road scenes or confronting assembly-line robots, the existing algorithms do not suffice. A major problem is that the existing algorithms use absolute and local criteria such as intensity difference, boundary strength [BF, BP], etc. to form regions. But the criteria for what is a "region" will surely vary with context. Certain shades of green, yellow and brown might be merged into a single region of grass in a scene, yet distinguishing the same set of colors might be crucial for region separation in another scene or even in another part of the same scene (like distinguishing a yellow car from green grass that it partially occludes). Another critical consideration is the goal of the perceiver. For some

problems, separating the green grass from yellow grass will be essential; in others it will be completely redundant and cause needless complication.

The importance of goal direction and context-dependent information (semantics) for effective problem solving is now well understood and established in artificial intelligence and scene analysis is just another example. One can certainly write a special purpose region analyzer for a fixed class of images and it will work better than any general algorithm. This, in fact, has been done in various systems [BF , HE] and is sometimes just the right thing to do. The obvious difficulty with this ad-hoc approach is that it requires a lot of work to build or modify each individual program.

The current implementation tries to tie organically the two tasks of segmentation and interpretation so as to get a more reliable partition and interpretation of the input. The general structure of the system can be applied to any combination of segmentation and interpretation process subject to the limits of the system with respect to the special structure of the semantics representation, and the classification capabilities.

In all work done on segmentation of visual input which are known to

the author, semantics was hardly used. When the semantics was used it was used in an ad-hoc fashion. Our system provides direct incorporation of the semantics into the segmentation process. However, for practical reasons the representation of the semantics had to be constrained. We developed a structure which is in some sense first order semantics. It cannot be used to describe all that we know about the problem domain, but what is describable can be directly incorporated in the segmentation process.

Before describing the system in more detail, we must make one additional point of clarification. It is a tenet of artificial intelligence research that any information that can be brought to bear will be helpful in a given task. This is especially true in machine perception, but our current efforts do not attempt to exploit it fully. Region analysis is assumed to be a preliminary (relatively fast) partitioning of an image before further processing. For this reason, we have made no attempt to include semantic features like three-dimensional shape analysis in the current region analyzer. We are still studying the capabilities of our semantic structure. As results of more experiments become available, we will be able to determine which information should be used in the segmentation process and which should be left for higher level processing.

1.4 HISTORY AND LITERATURE REVIEW

The following is a brief review of successful computer systems for A.I picture processing and a brief literature review. No attempt is made to cover all the literature relevant to image processing. The reader who is interested in getting familiar with the literature is encouraged to consult the literature surveys [ROS1 ROS2 ROS3] which survey over thousand recent articles on image processing topics. Relevant papers to our work appear in those surveys under the titles "Edge and curve detection", "Pictorial pattern recognition", "Picture parts", "Picture description" and "Scene analysis". We will reference only papers that had direct effect on our work or deal with closely related topics.

The hand-eye system at Stanford uses the edge detection approach. A procedure was developed [HUEC] which when applied on a circle around a point finds a best fit of a linear step function to the light intensity function in a neighborhood of the point

$$\text{step}(x,y) \hat{=} \begin{cases} u & \forall x,y \mid a*x+b*y \leq c \\ v & \forall x,y \mid a*x+b*y > c \end{cases}$$

$$u \neq v$$

Depending on the quality of the fit and the difference between u and v the probability of the existence of an edge line between two different light intensity regions passing in the circle is computed. The 2-d fit is needed to overcome noise by the use of the 2-d structure of the edge line. Noise arises from both hardware noise and small irregularities in homogeneous region. Alternative edge detectors were developed by other researchers like [GR] which approach the problem as a statistical decision with yes/no answer. Some researchers have tried to use gradient techniques but it seems that gradient derived operators are very sensitive to noise).

The recognition of edge segments using the Hueckel operator is very reliable for simple scenes. The main problem is incorporating the local edge segments detected in various points into a whole picture description. This becomes a very complicated task of edge following and making decisions as to how to close edges to create part (or all) of the contour line of a region and then to interpret the resulting objects by the world model [FALK GG]. Then comes a complicated feedback loop to call the edge detection and following process again with different parameters to recognize predicted edges that were missed, or to delete some erroneous edges [PT TEN]. Algorithms for connecting reliably and efficiently edge pieces were developed by many researchers. The common alternatives to edge following are

various algorithms derived from the minimal spanning tree algorithm where attempt to pass the shortest path through all edge segments is done. An alternative approach is represented by [MON] which utilizes a simple version of dynamic programming for optimal curve detection.

A region growing algorithm was tried at S.R.I. [BF]. This algorithm involved actually melting in random order all boundary lines whose strength was less than some threshold. This threshold was supposed to be given a priori, and had to be adjusted for different pictures. The strength of the boundary was computed as a function of the length of the boundary and the structure of the differences in light intensity across it. The main problems with that system are the heavy computational load resulting from lack of any sampling facilities, limited reliability because of randomly ordered merging of boundaries whose strength was less than some absolute a priori threshold and the lack of any facility to incorporate semantics directly into the region grower. A few researchers have tried to develop techniques for local adjustment of the thresholds of the region growers mainly through local histogram analysis in various parts of the image; such work is now in progress in J.P.L. (oral communication). The work on region growing described in [HE] is in many respects the nearest to our work. It

is an attempt to tie region growing process with specific problem knowledge. The main difference is that our system is more general and more reliable.

World modeling for pictures was developed for planar surface scenes (block world of cubes and wedges) in parallel at Stanford A.I. laboratory by G.Falk and G.Grape for real images, and in M.I.T A.I laboratory by Guzman and Waltz for idealized images. The result of this effort was a well understood world model of planar surface bodies which was able to sustain quite a lot of segmentation errors by the lower level portion.

An attempt to use a semantic graph with some hints of associating probabilities with the links was developed in [PEP]. This was an attempt to model hand-input and hand-segmented images of outdoor scenes.

Our world model is an extension of these models to use both probabilistic world knowledge collected by the system, and an option to utilize the model directly while segmenting the picture. The problem knowledge is collected by the system from training examples and is not limited to planar surface objects. The major deficiency of our current model with respect to the planar surface models is the

absence of vertices and explicit 3-d structure in our model, which are of major importance in the planar surface model. Preliminary investigation indicates that vertices and 3-d structure information may be added to our model without significant change in the structures.

The first application of our system was to road scenes it is worth mentioning in that connection that outdoor scenes analysis tends to be a good source of texture oriented problems. [RBJ] describes work on texture which involved also texture derived from outdoor scenes. Though we provide easy hooks for utilizing textures, texture is not being used in our current system.

SECTION 2

REGION GROWING - CLUSTERING BY MELTING WEAKEST-BOUNDARY-FIRST

2.1 OVERVIEW

This section is a description of the general region growing mechanism. The control mechanism of the weakest-boundary-first region grower will be described briefly, while the specific details of decision criteria will be described in later sections. Section 3 will deal with growing regions without direct use of the semantic model. Section 4 will show the semantics representation, and the following sections will show how we incorporate the model into the region growers and image interpreter.

I will start with a brief overview of the system. The system consists of a sampling mechanism, region growing subsystem and optional edge following. Together they are intended to generate the basis for an efficient and reliable image segmentation system. The region growing algorithm will generate a sequence of partitions of the pictures and will maintain an approximate description of regions, boundaries and

vertices for each partition of the picture observed. The features of the regions, boundaries and vertices will be used to make the decisions that control the algorithm. The basic step of the region grower is to take pairs of regions with a common boundary and merge them to generate one bigger region. When using the weakest-boundary-first region grower, the decision will be to melt the weakest boundary between two regions in the current partition. The evaluation of the strength of the boundaries will control the algorithm. Successful evaluation of the strength of a boundary will be the key to the success of the system. A large portion of the thesis surveys options used to compute the strength with and without the use of the specific problem knowledge.

An evaluation of the quality of partitions of the image is needed to decide how to terminate the algorithm. This evaluation scheme will be used to identify the best partition observed and to restore it on termination. The evaluation procedure provides also for an alternative region grower which is driven from the model directly [see Subsection 6.2]. The semantics of the model would be used to determine the evaluation of the quality of the partitions observed.

Initially the picture will be approximated using sampling to save computing time, but in any stage the option to call an accurate

tracing routine for the contour line of the regions will be available. This procedure will use the approximate description of the regions and boundaries in the current partition to get the accurate contour of the existing regions. After application of that procedure an accurate shape description will be available. The optimal partition is to be passed along for further processing by special purpose routines which are determined by the specific task at hand. This special purpose routine can make much better use of special information about the problem domain which was not expressible in terms of the limited structure of the semantics used in the region growing mechanism.

2.2 INITIALIZATION

Prior to the application of the region growing algorithm, the image to be processed is covered with many small regions. With each iteration of the region grower, two regions will be merged to become a larger region. It is not desirable to start the process with each single grid point as a separate region. There is too much redundancy if we do that. The properties at each point are not reliable enough because of noise. Furthermore for practical application the smallest

region that may be encountered will be composed of very many grid points. For these reasons we place sample points over the image grid. In our experiments, placing the sample points at every fifth point yielded a reasonable density. Each of the sample points is assumed to be representative of a different region for initialization. Doing this we gained two things: first, local operators may be applied around each of the sample points to find more accurately the local properties (reduce noise), and second, the number of regions is $1/25$ of the number of grid points.

We start by placing sample points on the picture rectangle. The placement of the points is such that they cover the picture in some desired density. If information is available on the picture we may want to place the samples so that they will be concentrated near edges of regions and less frequent in the center of regions (here "regions" means the regions that we want to terminate with as defined by the world model). Local operators are applied to determine the local structure around each sample point. This information may be the dominant color, color texture, various histograms, color gradient and 3-d local structure information, depending on what is available and is considered important in the problem domain. This information is stored in a feature vector that is associated with the sample point. In the current implementation it is just the dominant color

and light intensity around the sample point. Substantial amount of research is still required to develop good local texture operators [see Subsection 3.2 for more details].

On initialization each sample point is assumed to be representative of a different region, resulting in an implied region (area) associated with each sample point. Take any sample point and call it "sp". Then the implied region around "sp" is the intersection of all half planes which include "sp" and are defined by the perpendicular bisector of the line which connects "sp" and some other sample point. Practically, we do not need to take all such intersections because of the special structure of the placement of sample points [see Figure 2.1]. Between implied regions there are implied boundaries. The implied boundary between two sample points, if it exists, is the common line segment of their closed implied regions which is on their perpendicular bisector. The single segment of an implied boundary will be called the basic implied boundary and will stand for two adjacent sample points from different regions. Later on, when more than one sample point belongs to a single region the implied boundaries will be composed of several segments of basic implied boundaries. We define a contour line to be the closed path that surrounds a region or a hole inside a region. This contour line may be composed of several boundaries which generate a closed circular path [see Figure 2.2].

One point that should be mentioned is the treatment of the limit of the field of vision. Consider a point on the extreme end of the grid. By our previous definition it would have an unbounded implied region. We want to fix this case so that it will be treated uniformly. An easy way to take care of this case is to have an artificial region which will stand for the domain outside the field of vision. This region will be called O . This region has a common boundary with all sample points that are on the border of the field of vision. This boundary will never be melted and will be used to close the contour line around sample points that had an unbounded domain associated with them by our previous definition. Now whenever we want to check if a region touches the border of the vision field all we have to do is to look for a common boundary of this region with the outside domain (the artificial region). This special boundary provides for contour lines that are always closed paths around regions. This simplifies the edge tracing. The existence and shape of a common boundary between a region and the outside domain are extremely important in recognition of regions. For instance if a region touches the top of the vision field it increases its probability of being sky in the context of outdoor scenes. The shape of these boundaries indicates which of the four sides the region touches and the length of each of the boundaries.

2.3 WEAKEST-BOUNDARY-FIRST REGION GROWER

The description of the weakest-boundary-first region growing algorithm is simplified if we consider the structure of regions and boundaries as a graph structure where the nodes are regions and the edges (links) are the boundaries (Figure 2.3). Each link, representing a common boundary between two regions, has a value associated with it. This value reflects the probability that the two regions are of different interpretation in our world model. These values are called the boundary strengths. The evaluation of the boundary strengths is the responsibility of the control mechanism. By means of evaluating the boundary strengths, the control mechanism controls the region grower. The successful evaluation of these values is the key to the successful processing of pictures by the system.

The basic step of this region grower is to take the weakest boundary in the current image segmentation and merge the two regions for which this is the common boundary into one bigger region. In the corresponding graph structure this means collapsing into a single node the two nodes joined by the weakest link. The resulting node (region) will include all the points of the two regions. The links (boundaries) of the new node (region) will be assigned new values

(strength evaluation) and the next iteration will start (subject to non-termination condition).

The structure that is created after several collapsing stages is demonstrated in Figure 32.2 and in Figure 2.4. Each region is composed of one or several sample points. The boundaries between regions now are lists of pairs of sample points. Each such pair has one point from each of the two regions that the boundary connects. These two points are adjacent to each other. The pairs of sample points which define a boundary are ordered by the order that the real boundary line passes through them. There are two such orders, clockwise and counter-clockwise as seen from each of the two regions. The contour line is composed of one or more boundaries. We maintain a circular list for each contour line of a region, which is the list of the boundaries as they are encountered along the contour line. The maintenance of the above structure is necessary to the description of the boundary shape in each stage of the growth algorithm and for the later accurate contour tracing. No attempt was made at this stage to optimize the representation of the boundary and it is clear that better encoding and more compact approximations are available.

2.4 THE REGION MELTER

With each unification of two regions the data structure needs to be updated. The regions themselves are not ordered, so the combined region is just the union of the points included in the two subregions that compose it. Since the boundary structure is ordered, more elaborate updating is needed. The major complexity results from the fact that the boundary that was melted can be composed of several discontinuous paths. In such a case the resulting region may not be simply connected and hence its boundary will be composed of several closed contour lines. Another minor complexity occurs when a third region has a common boundary with both unified sub-regions. In this case if these two boundaries are continuous then they should be combined into one boundary for the new combined region. To cope with all possible combinations a special algorithm for updating was developed (see Figure 2.5 and Figure 2.6).

2.5 THE BOUNDARY STRENGTH LIST

On each iteration of the weakest-boundary-first region grower we need to find the weakest boundary. To reduce the search time, a list of

the boundaries is maintained. This list is sorted according to the strength of the boundaries on initialization of the algorithm. After each iteration of the region grower, the values of the strengths of boundaries around the new region are evaluated. The boundaries with new values are then relocated in the boundary list to maintain the proper order (by boundary strength). The updating time usually is reduced when starting the search for the new position of the new boundary from the location of the corresponding boundary of the old smaller region. It turned out that in many cases the new value is about the same as the old one. To utilize this property the strength list is doubly linked so that it is easy to float a boundary to its appropriate position in the list as determined by its new strength.

2.6 SOME IMPLEMENTATION DETAILS

The program relies heavily with LEAP features of SAIL [SAIL]. Each sample point is an ITEM (pointer to data structure) which contains the local feature vector at that point. The regions are set ITEMS which contain all the sample points that belong to that region. Associated with each region ITEM is a region feature vector which contains properties of that region. This vector is updated whenever

the region is merged into one of its adjacent regions. The boundaries are lists of pairs of sample points. Each boundary is associated with the two regions which define it. The boundary list is ordered so that the pairs of points are ordered in the order of the boundary path that passes between them. Since there are two such orders there is an indication from which of the two regions the pairs are seen ordered in clockwise direction. With each boundary is associated a boundary feature vector which is updated as the boundary grows. Now for each region and for each of its boundaries we indicate which boundary is next when going clockwise along the contour line of that region. This is done by the association structure. It should be noticed that the boundary may close the region and hence follow itself and there may be also several closed contour lines for a region when it is not simply connected (has holes).

2.7 STOPPING CRITERIA

One decision that has to be made is when to stop melting boundaries. There are three possible options for doing that. One is to stop when the weakest boundary is stronger than some threshold. Another

possibility is to have a state evaluation and to use a back up mechanism to get to the most promising segmentation (that is the one with highest state value) from all partitions generated by the region growing mechanism on its way to a single region. The third option is to find the best interpretation for the scene given the current segmentation. If the resulting interpretation does not interpret any two adjacent regions as parts of the same region (in the world model sense) then we quit merging (see Subsection 6.1). In the current implementation the first and the last options are used.

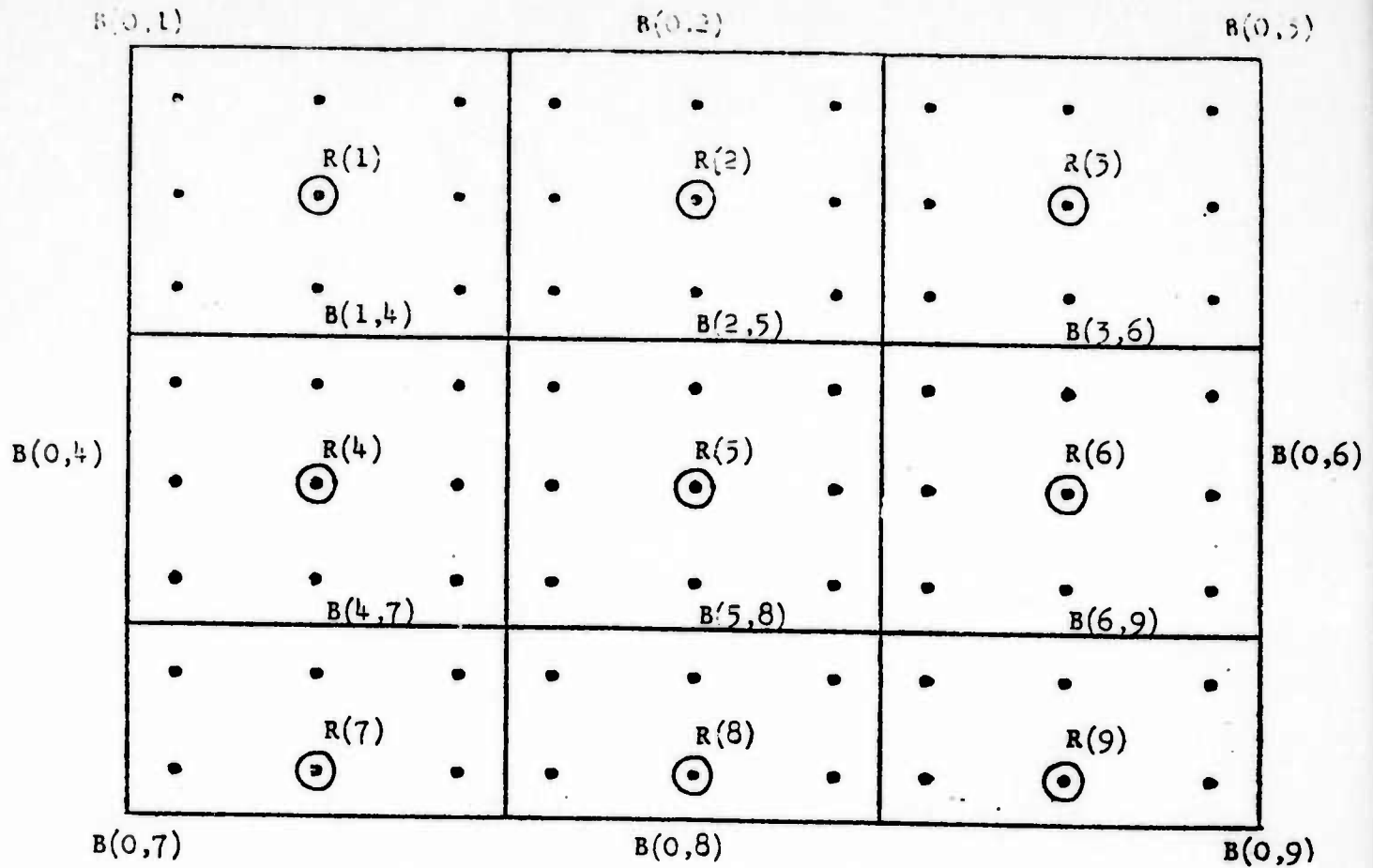
2.8 EDGE FOLLOWING

Edge following can be used for refinement and verification of the boundary structure between sample points for a given picture partition. In such a partition, the implied boundary structure generates implied contour lines around each of the implied regions. This contour line is an approximation of the real contour line of the region. Each basic implied boundary is a segment that is located along the perpendicular bisector of the two sample points. By edge following, we want to replace this implied boundary by a real boundary; that is we want to replace it by the actual pairs of grid

points which define the boundary. To do that, we scan along the line between each pair of sample points which define the basic implied boundary (two adjacent sample points from different regions) to find the exact edge point. Since the two sample points belong to different regions there must be a point along the line that connects them which is the best real edge between the two regions. We can use any available edge operators to detect the optimal location for the edge. This task is especially easy since we know the distinguishing properties between the two regions. We repeat this process for all pairs of sample points which define the basic implied boundaries. Next we want to connect the edge points that we collected, and to find the exact edge curve that passes along the boundary. The implied boundary structure also includes the linkage between the basic implied boundaries. This linkage is the order that the contour line passes between the sample points. Our task is to connect the pieces of edges that we found to create the whole contour line. We do it pairwise for adjacent edge points. (We know the adjacency by the linkage structure). This is done using edge tracing which is relatively easy since we know the properties of each of the two regions that define that edge line and we know two edge points that we want to connect with a simple edge line.

We may expect to find some discrepancies between this implied boundary

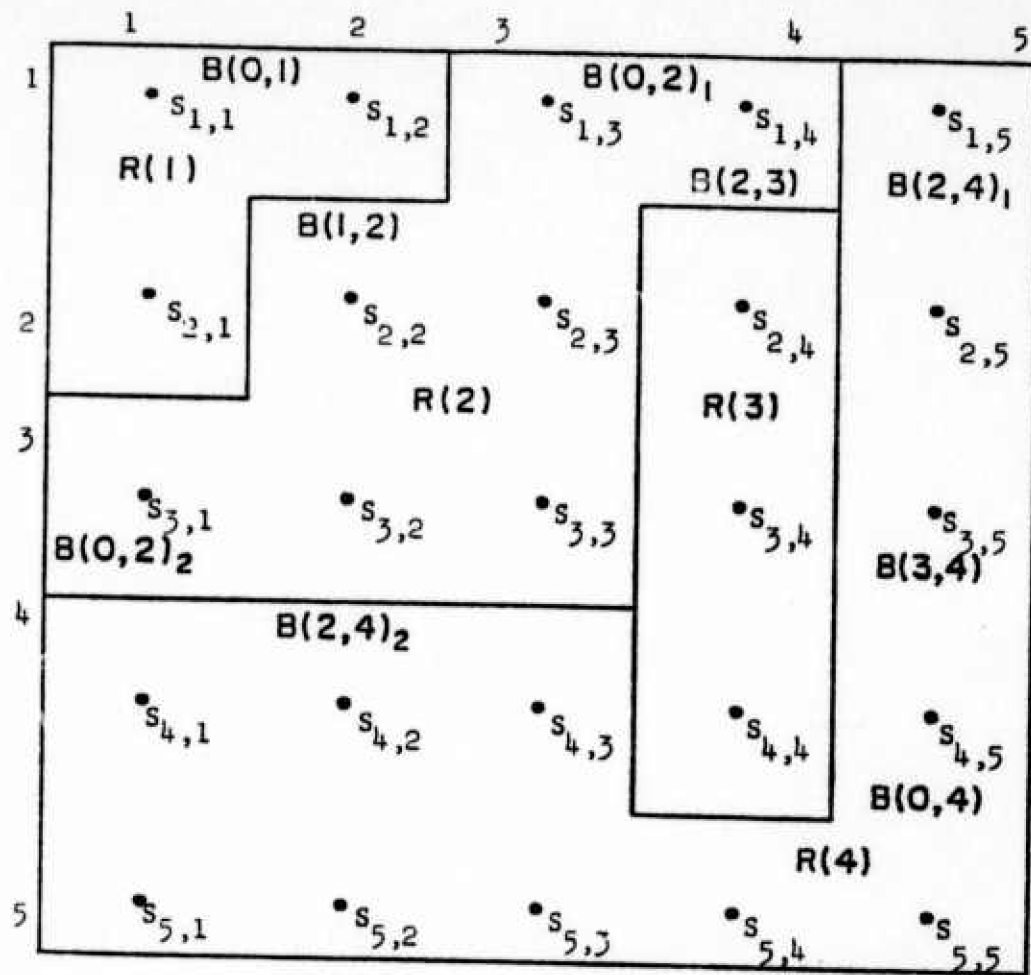
structure and the real boundary structure. There are two sources of problems. One problem is the discovery of new regions when scanning the lines between two sample points. This may happen because the sampling was not dense enough. The other occurs when two regions that were assumed disconnected turn out to be connected by a bottle-neck that was missed by the sampling process. Both problems require special treatment. In the current implementation we assume such cases will not occur. This means we assume a dense enough sampling that fine details will not be lost. If special purpose techniques were used they would be along the lines of those described in Subsection 3.4. In the current implementation this edge following is still missing, but it will become essential when more region and boundary shape descriptors are added to our system.



- . - A grid point
- ⊙ - Sample Point
- R(0) - Is the outside domain
- R(1)-R(9) - The initialization regions, each composed of one sample point.
- R(1) - Stands for 9 grid points $P_{1,1}$ $P_{1,2}$ $P_{1,3}$ $P_{2,1}$ $P_{2,2}$ $P_{2,3}$ $P_{3,1}$ $P_{3,2}$ $P_{3,3}$
- B(0,*) Is the Boundary of R(*) with the outside domain.
- B(I,J) $I \neq 0$ $J \neq 0$ Is the implied boundary between regions R(I) and R(J).

On initialization it stands for a single pair of adjacent sample points from different regions.

Initialization Structure - Rectangular
Cover, Sampling Every Third Point



. - A sample point

R(1) - ($s_{1,1}$ $s_{1,2}$ $s_{2,1}$)

R(3) - ($s_{2,4}$ $s_{3,4}$ $s_{4,4}$)

B(1,2) - ($s_{1,2}$ $s_{1,3}$) ($s_{1,2}$ $s_{2,2}$) ($s_{2,1}$ $s_{2,2}$) ($s_{2,1}$ $s_{3,1}$)

R(1) - Sees B(1,2) going clockwise

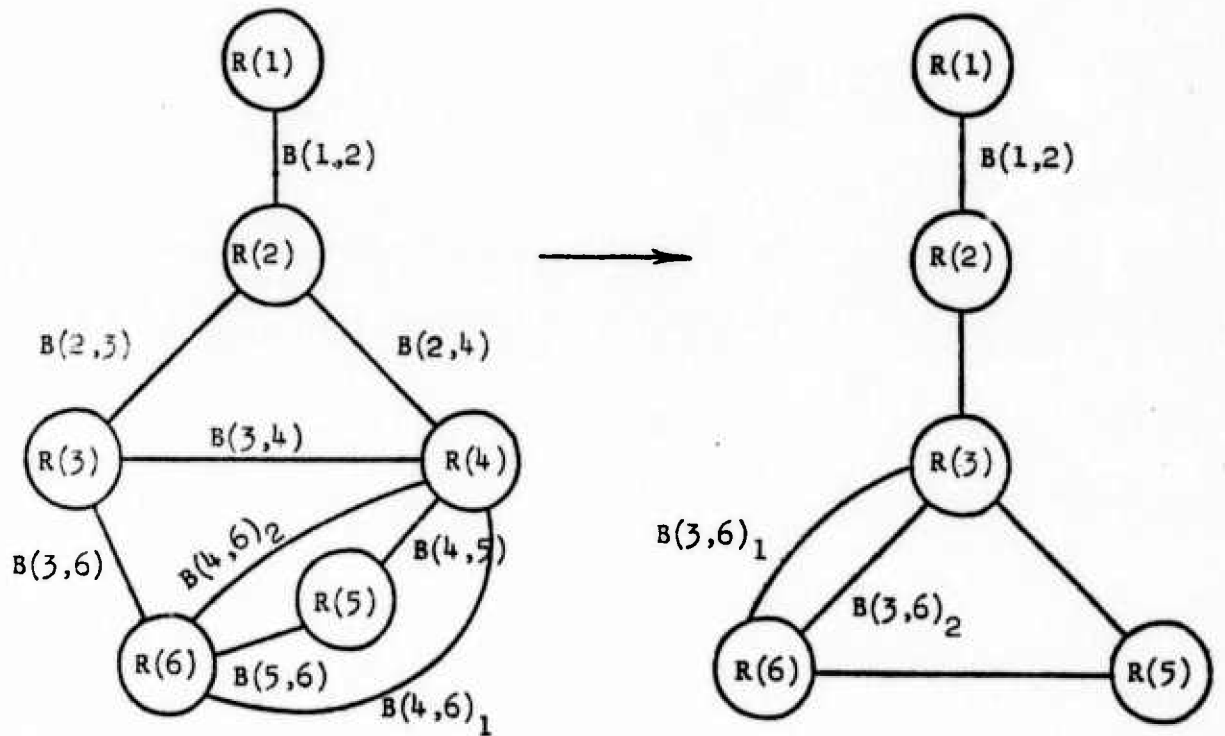
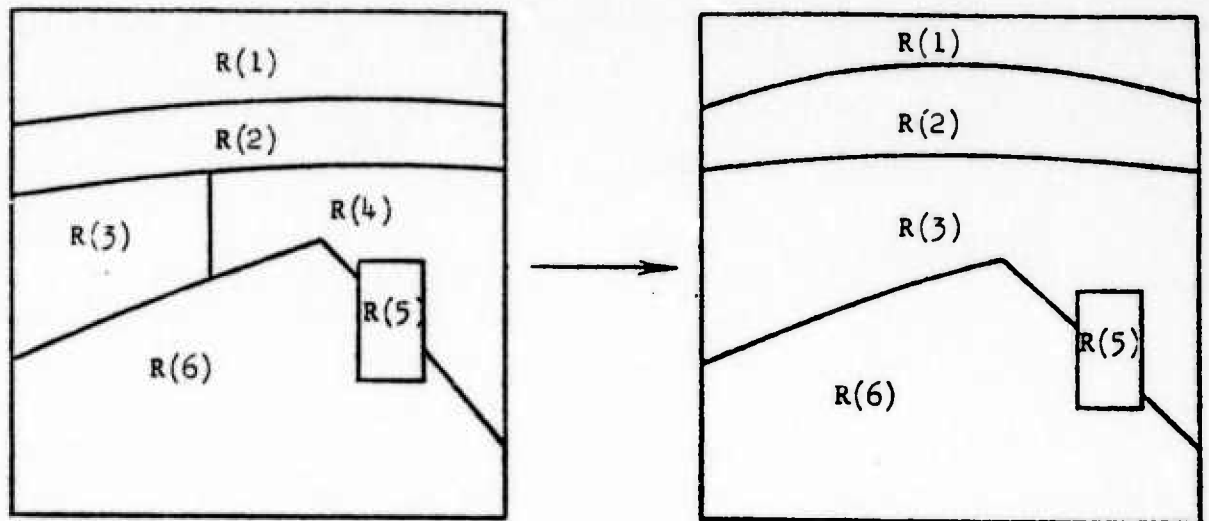
R(2) - Sees B(1,2) going counter clockwise.

R(2) - Contour line:

B(0,2), → B(2,4), → B(2,3) → B(2,4)₂ → B(0,2)₂ → B(1,2) →

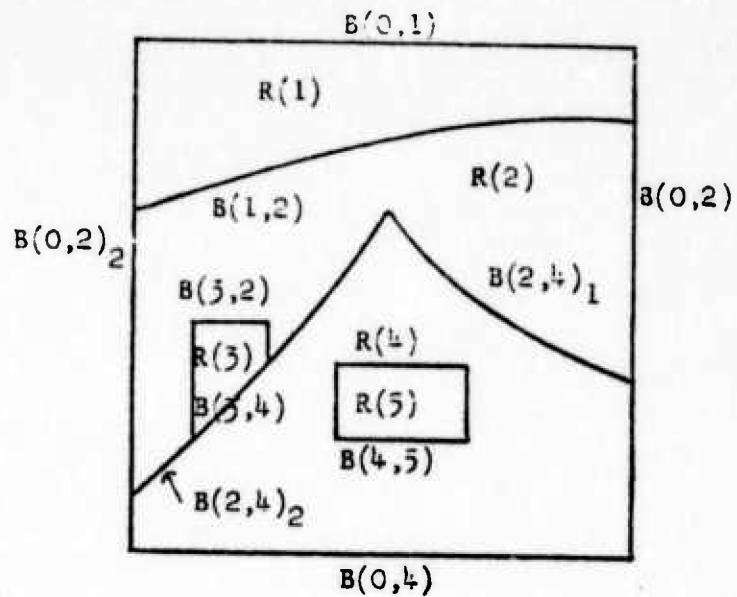
Example of Structure After
Few Region Growing Iterations

Figure 2.2



Example of Effect of Merging
A Pair of Regions on the Structure

Figure 2.3



Regions: R(1), R(2), R(3), R(4), R(5)

External Boundaries: B(0,1)₁ B(0,2)₁ B(0,2)₂ B(0,4)

Internal Boundaries: B(1,2), B(2,4)₁, B(2,4)₂, B(3,2), B(3,4), B(4,5)

Contour Lines:

R(1): $\boxed{\rightarrow B(0,1) \rightarrow B(1,2) \leftarrow}$

R(2): $\boxed{\leftarrow B(1,2) \rightarrow B(0,2)_1 \rightarrow B(2,4)_1 \rightarrow B(3,2) \rightarrow B(2,4)_2 \rightarrow B(0,2)_2 \leftarrow}$

R(3): $\boxed{\leftarrow B(3,2) \rightarrow B(3,4) \leftarrow}$

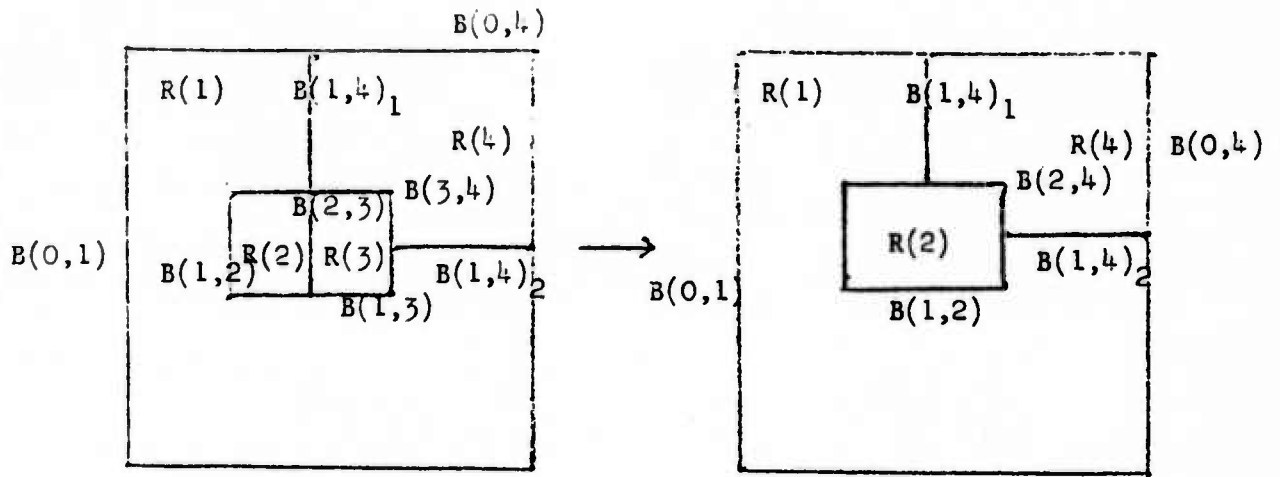
R(4): $\boxed{\leftarrow B(2,4)_1 \rightarrow B(0,4) \rightarrow B(2,4)_2 \rightarrow B(3,4) \leftarrow}$

$\boxed{\leftarrow B(4,5) \leftarrow}$

R(5): $\boxed{\leftarrow B(4,5) \leftarrow}$

Example of Regions and Boundaries Structure

Figure 2.4



$B(2,3)$ Melted
 $R(3)$ Deleted
 $B(1,2) \rightarrow B(1,2) \cup B(1,3)$

Contour Lines:

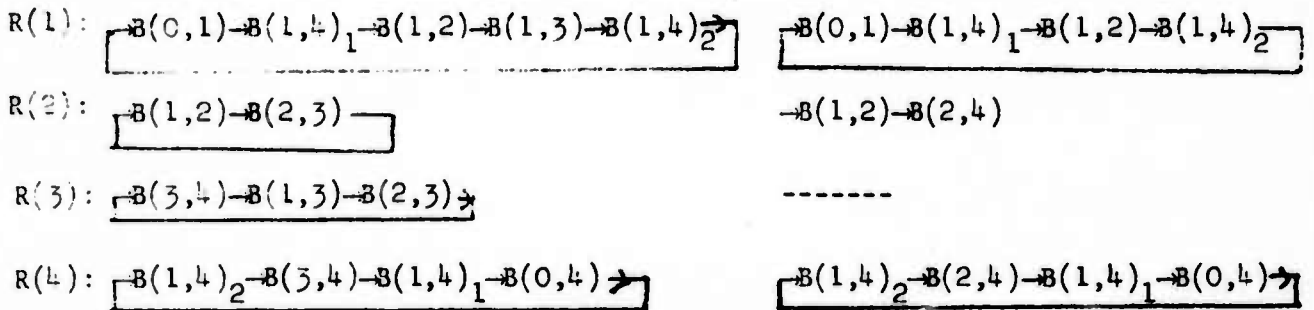
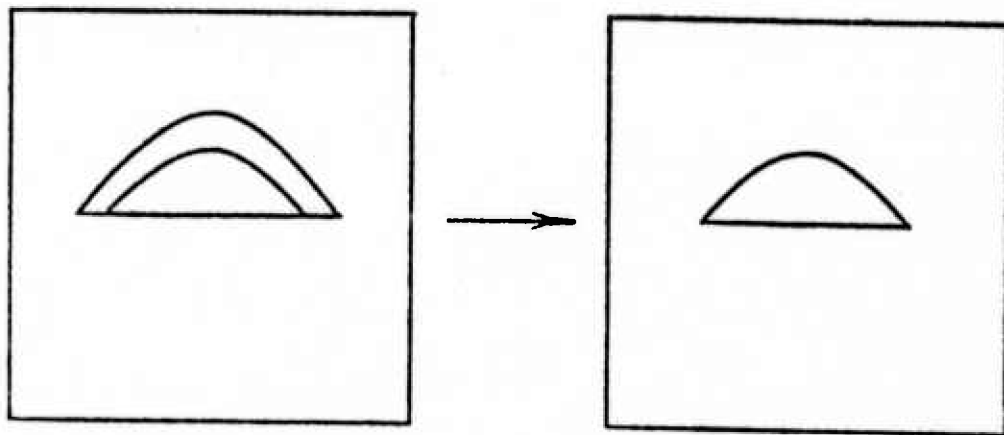
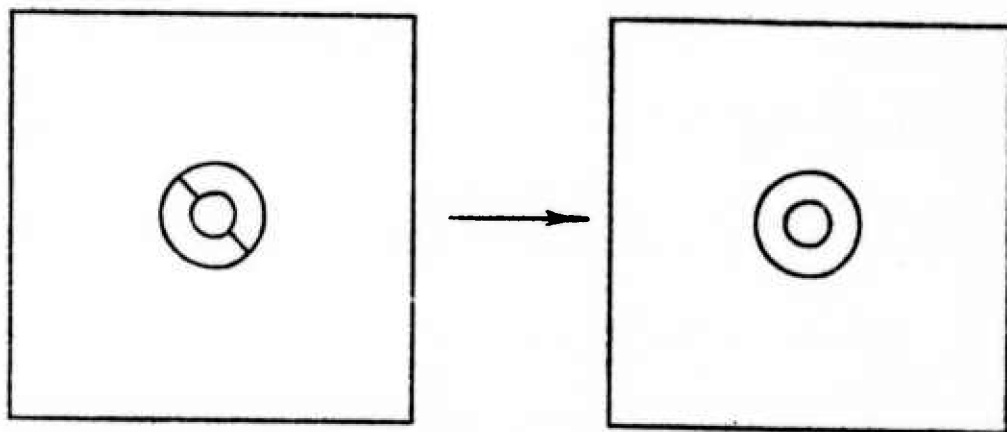
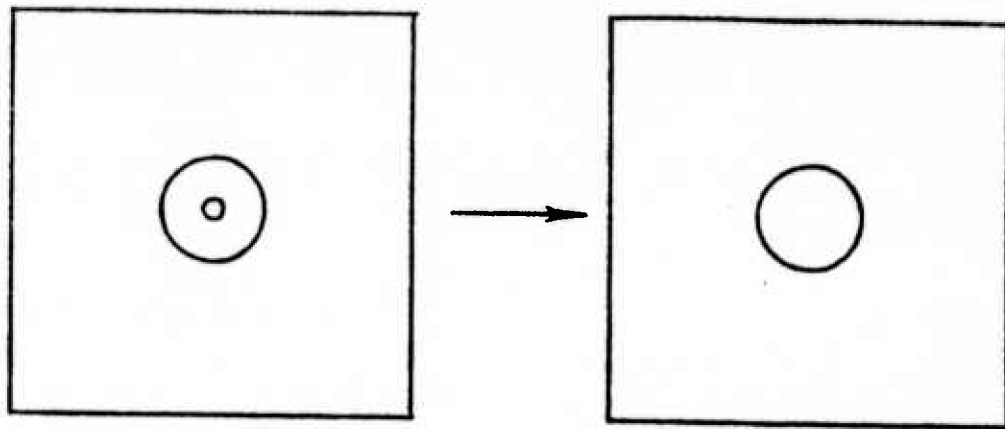


Figure 2.5

The effect of merge on data structure



Some Special Cases For Melder

Figure 2.6

SECTION 3

REGION GROWING WITHOUT USING SEMANTICS

This section more thoroughly describes that part of the system which is active prior to the incorporation of the semantics into the region grower. This portion of the system is intended to be a problem reducer. It tries to reduce the complexity of the image from 60000 points to about 100 regions. The resulting regions are assumed to be subparts of the regions with which we want to terminate. That is, we assume that only very few and minor false merges occur in this phase, and if errors do occur they will be both tolerable, and anticipated by the next phases of the system which utilizes the problem semantics (e.g the semantic world model is generated by working experimentally on real typical images of the problem domain and hence false merges occur while training the system and hence stored in its semantic base). To minimize the risk of erroneous merges, this region grower is stopped with very conservative stopping criteria. This level is more efficient computationally than the run with the semantic model because of a simpler decision mechanism. On the other hand, it is much less reliable and for that reason it has to be stopped quite early, before the decision as to which region to merge becomes unreliable in the world model sense.

The aspects of the system that will be described here are:

- 1) Placement of sample points.
- 2) The local measurements at each sample point.
- 3) Evaluation of boundary strength.
- 4) Evaluation of a given partition.
- 5) The information on regions and boundaries carried with the grower algorithm.

These details are not essential to understand the subsequent sections so the reader may skip points that are too technical to be interesting.

3.1 PLACEMENT OF THE SAMPLE POINTS

The initialization of the region growing algorithm is done by placement of the sample points. Each of these sample points is

considered on initialization representative of a different region. There are two conflicting goals here. On the one hand we want to have as few sample points as possible so that the computational load will be reduced. On the other hand we want dense sampling so that the finer details of the picture will not be lost. The density of the sampling should satisfy the following two conditions: first, from each region that we want to terminate with, at least one sample will be taken. Second, every "bottle neck" in a region will be sampled, meaning that a connected region will not appear disconnected. In many but not all classes of scenes we can find a satisfactory density, which is also sparse.

To ease the computation effort, a fast way of eliminating redundant sample points is provided. This will effectively allow us to increase the sampling density and still keep the number of samples low. We assume that two adjacent sample points are in the same region if the differences between their property vectors is less than some threshold. This implies that a sample point for which the difference between its feature vector and feature vectors of all neighboring sample points is less than that threshold will always be inside a region and not on a boundary. Such points are not interesting for us and we can ignore them, and connect their neighboring points into one region immediately. This way we reduce the number of samples. As a

result samples from the center of big homogeneous regions are ignored. In the current system it was found to be effective to set this threshold as .05 of the maximum possible difference between the property vectors of two sample points as computed by the histogram of light intensity over the grid. Using a higher value caused failures by collapsing into one region sample points that should belong to different regions. This sample point reduction is faster computationally than the region grower. Because this reduction is faster than the melting procedure it allows effectively denser sampling. Most of the simple region growing systems use versions of this path-wise connectivity criterion as the major tool in their region growing algorithm. This clustering mechanism is extremely sensitive to noise and causes severe errors very early. That is why we stop it with a very conservative stopping criterion. Ideally with a more efficient implementation of the weakest-boundary-first region grower this stage could be avoided completely. Note that the elimination of points from center of regions is more conservative than the path connectivity, because in a sense we demand "wide" path connectivity.

If some prior approximation to the location of boundaries is given, then we will place the samples mainly in the neighborhood of the boundaries. This way with fewer sample points we still get a good

description of the regions and boundaries (this will be very effective in multi-picture processing where we have slowly varying pictures). In the case where no prior information was available, the most effective initial placement of the sample points is done by placing the points so that initially all regions will be equal regular hexagons. In this case, the smallest region or bottle neck detected is twice the radius of the hexagon. The advantage of the regular hexagon cover over square or equilateral triangular covers of the picture area is the symmetry of its boundary structure. If squares or equilateral triangles are used as the basic units of the cover, there are pairs of regions that have only a single vertex in common. These vertices make the two covers based on equilateral triangle and on square units ambiguous, because it is not clear whether two regions that have only one common boundary point (vertex) in common should be considered adjacent. We ignore the single vertex boundaries for the rectangle cover.

If we chose any of the special structure covers of the plane (like the one which is composed of equal squares), the initialization of the boundary and region structure becomes trivial, because the special structure conveys directly the structure of the regions and boundaries. The distance between the sample points will be called the quantization factor, which in the current application is a number

between 5 and 20. This number reflects the quality of the description of the shape of the regions and boundaries that we want to get. The full size picture frame is a rectangle of 200x300 points so we have 2400 to 150 sample points (depending on the sampling density). In simple scenes (composed of a few relatively homogeneous regions) the use of the initial point reduction reduces the number of regions in the denser case to about 500.

3.2 LOCAL FEATURE DETECTORS

The information associated with each sample point depends on the hardware available. In passive input devices it is the local light intensity which reaches the image plane of the videcon at each of the grid points. In our case we measure the intensity through three filters (red, green and blue), to get color information. In active input devices where the source of light is available (mainly laser beams), depth information and 3-d surface orientation are also available. (For the capabilities of an active light system see [GJA]). laser light for scanning over the scene.

In the current application, only color and light intensity

information was used. The color was used only to find the dominant color around the sample point. A problem arises when the sample point directly hits an edge between two regions. Dealing effectively with such cases will require application of "structure operators". These operators will try to find a compact description for the light intensity and color as a three valued function in a two dimensional neighborhood of the sample point. If we had such structure operators they would have recognized the edge. Currently though we do not apply them. To reduce the confusion resulting from such a case the dominant color is taken to be the most frequent color, not the average color. This way, in most cases, the properties of one of the regions near that point will be associated with the point.

More elaboration will be needed to effectively use sensitive input devices (more than the current 16 gray level input for each color) to detect gradual changes and texture. Gradual change can be detected easily by approximation of a planar fit to each of the color components in some neighborhood of the sample point, instead of just finding the dominant color. Detection of edges has been investigated quite thoroughly and a few good edge detectors and operators are available [HUEC GRIF]. Detection of texture is extremely hard, and largely an unsolved problem. It is likely that texture and edge detection will be tried in cases where the planar fit for the light

intensity is insufficient. We anticipate that in that case a sequential and conditional application of texture classification operators will be called to classify the texture. One such class of operators correlates the local intensity (color) with itself shifted in different directions. These operators will detect directionality and frequency in the local texture. Another approach is to locally partition the picture into small regions (using threshold or local clustering) to detect the local shape of the small regions which compose the texture. The most powerful system for texture recognition known to the author was implemented by [RBJ], and is based on local Fourier analysis. It is probable that sequential classification of texture of the same statistical nature as the classification of objects for the world model (see Subsection 7.3) will be very helpful in texture recognition. There are many other local measurements which may prove useful in certain scenes. The understanding of which measurements distinguish objects in scenes is a central problem in machine perception. A major advantage of the system described here is that new operators can be incorporated easily as they are developed.

3.3 BOUNDARY STRENGTH EVALUATION

The evaluation of the boundary strength depends on the context of the scene which is being analyzed. This strength should reflect the probability that the boundary is 'real' in the semantic sense. In the description that follows we try to present some general parameters that can be considered in the evaluation of boundary strength. None of these schemes uses the world model. Only direct use of the boundary properties is utilized. No attempt is made to understand what each boundary means in the world model semantics. On the other hand the semantics of the world can be used to help evaluate the weights of the different criteria used in evaluating the boundary strength. We will return to this point in the descriptions of the semantic boundary evaluation.

The first factor in evaluating the boundary strength is the difference between the values of properties of the sample points at each pair along the boundary. A strong boundary will usually be one where the differences across it are high and consistent along the boundary line. This is quite standard, although no previous work known to the author utilizes multi-property differences. All of them worked with a single property for evaluating boundary strength.

Let $(\vec{x}(i), \vec{y}(i))$ $_{i=1, N}$ be the feature vectors of the pairs of sample

points along the boundary. $\vec{x}(i)$ is the value of the measurements at the i -th point on one side and $\vec{y}(i)$ the value of the measurements at the point on the other side.

The average difference in properties along the boundaries will then be

$$\left| \sum_{i=1, N} (\vec{x}(i) - \vec{y}(i)) \right| / N$$

In our system we have a 3-vector associated with each sample point. This 3-vector is derived from the three readings of the dominant intensity of each color components in a small neighborhood of the sample point. If (r, g, b) are the light intensities through the red, green and blue filters, then

$$v(1) \leftarrow r + g + b$$

$$v(2) \leftarrow (-r + \cos(2\pi/3) * (g + b)) / v(1)$$

$$v(3) \leftarrow (\sin(2\pi/3) * (b - g)) / v(1)$$

$v(1)$ is the intensity, $v(2)$ and $v(3)$ are the x,y coordinates in the color plane.

It is reasonable to assume that in general we want to give different weights to different components in the feature vector. The values of the weights are not obvious. We may want to scale each property so that the maximum difference will be at most 1, and this is done in our current implementation. It may turn out to be useful to reduce the weight of a property when the variance of the differences of that property along the boundary is high. A high variance of a property inside each region may also decrease the weight of the differences of that property. We also tried to give very high weight to the two color components as compared to intensity, under the assumption that color is a function of the material of the region and hence less sensitive to lighting conditions (shadows and orientation), but it turned out in those limited experiments not to be of any help.

In the future when more structure than just the dominant color at each sample point will be used, the consistency of the features of the two regions will be more complicated to evaluate. There will be more involved structure in the properties that will be compared. Such a property is, for instance, local variance of the color around the point which we want to match. If it is high we may want to

compare distribution structure on the two sides and not just the mean or histogram peak. Gradual changes would be detected by the slope of the 2-d linear fit to the property in a 2-d neighborhood of the point. If such a fit is done, then the inconsistency between the fits in the two sample points will be the measurement of the boundary strength. More elaborate matching evaluation will be needed if texture detection in areas around sample points is computed.

The size and shape of the boundary and the regions should be used in the boundary strength evaluation. In general the shorter the boundary relative to the area of the regions defining it, the stronger we require it to be.

In the current implementation the following scaling is used.

Let:

$$\alpha = \frac{\sqrt{\text{size of 1-st region}} + \sqrt{\text{size of 2-nd region}}}{\text{length of boundary}}$$

then:

(eq 8)

$$\text{strength of boundary} = \text{average differences} * \frac{[K2 + \alpha]}{K3 + \alpha} \quad \text{where } K2 < K3$$

In cases where the sampling is fine enough to reflect the shape of the boundary, the strength of a jagged boundary is likely to be decreased. This will be especially true when the surfaces of our objects have smooth edges. In this case jagged boundaries will usually result from some gradual lighting change on a smooth surface. The jaggedness is not trivial to compute because of the noise effect of the quantization of the picture which results from application of the sampling. It can be measured as the local deviation from a smooth approximation (like straight line or low order polynomial) scaled by the quantization size. This principle was considered in [BF] to evaluate reliably the boundary strength. Other considerations in boundary strength evaluation may involve more shape evaluation and broader context. These considerations are not incorporated, though they can be used, mainly because they are left for the semantic region growers which makes better use of many additional properties.

3.4 BOUNDARY STRENGTH EVALUATION THROUGH EDGE FOLLOWING

There is another possible approach to the evaluation of the boundary strength. One could scan the picture frame along the line segment

connecting the pair of sample points and look for edge structure across this line. The use of an edge detection operator like the Hueckel operator [HUE] could help to evaluate the probability that there is an edge line between the two sample points and that they therefore belong to different regions. This option was used in early versions of the system. It was dropped in favor of denser sampling because of the complexity it added to the program structure. When this option is used, the scan is done once on initialization. Scanning is effective mainly for dealing with gradual changes and reducing the requirement on the density of the initial sampling. The strength of the boundary between two adjacent sample points is taken as the strength of the strongest edge structure which intersects the straight line segment connecting the two sample points using some edge detector. In case a new region is detected, a new sample point will be placed and a new region will be generated.

The treatment of a new region discovered by the scan between sample points is relatively easy. For the sake of uniform treatment a new sample point is taken from the new region. The implied region and implied boundaries of the new point are generated, in the same manner as for the initial sample points. For pairs of implied regions whose common boundary is changed by the new sample point, the boundary structure is updated. An intricate case may arise if the new region

is very thin and hence a line of some sort. Then we want to invoke a line following routine and avoid disconnecting the regions. Currently there is no treatment of this case in the system. But it should be included if a world model that includes line shapes (like characters) were added to the system. It turned out to be very useful immediately on creation of the new sample point to check if its boundary strength with neighboring points is less than the lowest value of current implied boundaries. In this case it is immediately collapsed into the nearest region. In order to make sure that we do not generate too many new sample points there should be a threshold that prevents generation of new sample points if the strength of the boundary of the new point (region) with one of the two points that defined this point is less than this threshold. This threshold will be set to be greater than or equal to the value of the weakest existing implied boundary, and will be increased over that value as there are more sample points generated so as to promise termination and simplicity.

3.5 STATE EVALUATION

The evaluation of the quality of a partitioning of the picture

without using the semantics can be done only on the basis of a simplicity criterion. Two such effective decision criteria are available:

to maximize

$$\frac{\text{sum of strength of boundaries}}{\text{total number of regions}}$$

or to maximize

$$\frac{\text{sum over regions of the region's average boundary strengths}}{\text{total number of regions}}$$

excluding the outside of the picture and the external boundaries.

These simplicity criteria try to minimize the complexity (the number of regions) and maximize the confidence (strength of boundary). It was found that in a few experiments that the optimal partition with respect to these simplicity criteria was very near to the optimal partition in the world model sense.

It should be mentioned in connection with these quality criteria that the strength of the weakest boundary does not necessarily increase as

the region grower is carried further. If a mistaken unification happens, the differences along the boundary may be inconsistent and result in a weaker boundary (recall that the boundary strength is computed by using the absolute sum of the differences and not the sum of their absolute values along the boundary and scaling them by their length relative to the area of the regions defining them).

These state evaluation functions are not used in the current system. This phase of region growing is used as a problem reducer for the semantic region grower. Currently we stop the merging by threshold. That is, once the weakest boundary is stronger than some threshold we quit. The threshold chosen is relatively conservative and was taken to be .15 of the maximum possible boundary strength for road scenes and 200 regions for the angiograms. We chose a conservative stopping criterion so as to reduce the problem and still keep the risk of erroneous region merges low.

3.6 MAINTENANCE OF REGION AND BOUNDARY PROPERTIES

Throughout the run of the algorithm the basic properties of regions and boundaries need to be maintained. The current portion does not

make thorough use of them but the semantics controlled part does. Most of the needed features except shape are easy to maintain, mainly because the measurements are derived from various integrations. On initialization each region is given its basic color, size and position (the same as the sample point that constitutes it). When two regions are collapsed, the two feature vectors are just added correspondingly and associated with the new region. This sum is used to compute the average of the property over the region, but we need to remember that the average is not always what we want. If the variance of some property is required, then the sum of squares of that property is kept and the variance is easily obtainable. The same holds true for the length of the boundaries. For the differences along the boundary for different color components and different directions, the direction of the differences is important. A convention based on the clockwise and counter-clockwise convention of the regions and boundaries structure is used to decide whether to add or subtract properties of the two growing boundaries. It should be noted also that in the current implementation we have a very rich representation of the structure. We do not make any attempt to compact the data. The reason is that this is an experimental system where we wanted to have maximum convenience of access to information when needed. Thus the finer details of compacting data were completely ignored. A substantial saving in compute time and storage

requirement is achievable by compact approximation to properties and description of regions and boundaries instead of keeping them in the raw form that we are using currently.

If in addition to features derived from integrations, we have other properties, then updating of properties will become more complicated. For example such properties are shape descriptors which require keeping extreme points in various directions (extrema of a linear functional along the boundary path as function of the length) and cross section length.

3.7 FINAL COMMENTS ON THE NON SEMANTIC REGION GROWER

We can compare this part of our system with other region analysis algorithms. First, this algorithm, which uses sampling over the grid, is substantially more efficient than other algorithms that use exhaustive search on the whole picture and treat all grid points. Our approach in a sense allows us to concentrate our attention very rapidly on the important portions of picture, the boundaries between regions. Secondly we do not collapse regions in random order as long as the boundary is weaker than some a given priori threshold. We

first merge the globally weakest boundary on the whole picture. This makes the region grower much more reliable (see illustration A), and enables us to use more sophisticated stopping criteria.

The major gain resulting from ordering the merges is that doubtful merges will occur after obvious merges. The result is very often that a long boundary that has a local weak part will not be destroyed, since often by performing more obvious merges, the boundary will grow to its full length and then the strength computed by the average differences will be high. A stopping criterion which is more general and uses state evaluation can be applied to stop and back up to an optimal state. The optimality can be determined using general criteria on the types of regions and their anticipated inter-relations or complexity.

It is possible to keep with each region a binary tree which will trace how a region was generated (the pairs of regions whose merge generated that region). Such a tree can be used further by higher level processing, either to get finer resolution on parts of the regions or to decrease the number of regions by reunification.

Some of the simple region growing heuristics used in the past have gross difficulties. Consider the following simple example. Assume

that we grow regions by melting all boundaries with a value less than some threshold independent of order. This is usually done by starting with a point and trying to grow around it the region of all points which satisfy the following property: there is a path of adjacent points with property differences less than some given threshold connecting them to the first point. Often the threshold is not an effective criterion as shown in the following example:

1 1 1 0 0 0

2 2 2 0 0 0

3 3 3 0 0 0

Here we consider a 6x3 grid where the distance between nodes is just the absolute difference between the values in the grid points. If we give a threshold of less than 1 it will end with 4 regions but any threshold greater than 1 will result in a single region which will be the whole grid with an external boundary. On the other hand, our technique use the weakest boundary first, with the boundary evaluation as in eq. 8. The result is that going down from 4 regions the areas with values 1, 2 and 3 will always be merged first before collapsing them into the 0 region (remember that we also count the length of boundary relative to size of region in boundary strength evaluation). This means that we have a more reliable mechanism to overcome smooth changes where pieces of the boundary are

END

3.7

obscured, a situation very common in real pictures where shading causes loss of some pieces of edges.

SECTION 4

WORLD MODEL

4.1 THE STATISTICAL PROBLEM DEFINITION

The world model is statistical in nature, and in order to define it more rigorously we need to define statistically the problem that we face. In abstract terms we have $\lambda(i)$, $i=1,L$, the possible meanings of a grid point, where $\lambda(i)$ is the name of the object in the real world for which this grid point is part of the image. Assume that we have a grid of $\vec{x}(i,j)$ points, where $\vec{x}(i,j)$ is the feature vector $i=1,N, j=1,M$ of that point (in our case x,y coordinate and r,g,b of the three measured color components). An interpretation of the scene will be an assignment of some $\lambda(i)$ to each point, that is, identifying image points with objects. Our task is to find a good assignment. We will adopt the maximum likelihood principle. That is, we want an assignment

$$I: N \times M \rightarrow \lambda$$

DEFINITION

4.1

(where $I(i, j)$ is one of the $\lambda(k)$ $\substack{k=1, L$ which is assumed to be the meaning of the (i, j) -th point which has feature vector $\vec{x}(i, j)$), such that the total joint probability of

$$P_I(\vec{x}(i, j) \substack{i=1, N \quad j=1, M})$$

is maximized over all possible I .

Unfortunately this probability measure in that space is extremely hard to approximate, and even if we had it in terms of this raw assignment function, finding the optimal assignment would require a horrendous amount of search. We are interested in image domains where there is a variety of changes between images. It may be easy to compute some probabilities like

$$P_{\lambda(k)}(\vec{x})$$

that is, the probability that the point has property \vec{x} if it is of meaning $\lambda(i)$. However it is extremely difficult to extend this to the joint probability of all features of points in a scene, since

there is a high degree of dependence between properties and meaning of different points. Our attempt is to reduce the dimensionality of the possible assignment by grouping points into domains (regions) where we constrain all points in the domain to be of the same meaning. By this reduction we gain two things: first, the number of possible choices is reduced significantly, and secondly we claim that it is much easier to express the structure (and hence to approximate the joint probability function) in terms of the domains and their properties. The problem is then transformed into the problem of segmenting the global scene into regions so that all points in a single region will be of the same meaning, and trying to find maximal segments. That is, we do not want to be left with two adjacent domains of the same meaning. In the initialization process, which was described in the previous sections, we assumed that adjacent points that have about the same local features are of the same meaning independent of what the meaning is. The clustering process was carried out using this assumption to reduce the problem. However to play safe we had to use a very conservative criterion for similarity which left us with about 100 regions and more reduction is desirable.

4.2 THE BASIC ASSUMPTION

In this section we try to represent a probabilistic model with the following claims: First, it is a good approximation to the real probabilistic structure for many picture domains. Secondly, it may be used effectively in reducing the problem by allowing reliable clustering which is far more advanced than the one based directly on the feature vectors. For region analysis, we define the utility to be:

$$P(\text{global_interpretation} \mid \text{context, values of measurements})$$

This expression actually stands for

$$\prod_{i=1, N} \prod_{j=1, M} P(\vec{x}(i, j) \mid I(i, j)) \times P(I(i, j) \mid \text{a priori}) / P(\text{a priori})$$

where $\vec{x}(i, j)$ are all the measurements in all points and $I(i, j)$ is the meaning assigned to point (i, j) , which will be its interpretation. The context here means the underlying probability space of the picture domain, which we collect experimentally (see section 7 on learning). The probability space is defined for each problem domain by the variations in the scenes that are in that problem domain. We

will keep the conditionality on context to remind the reader of the special probability space which is problem dependent, and perhaps also variable. We should also note that our discussion is immediately extendable to more complex utility functions than the linear utility which is identical in this case to the maximum likelihood principle.

An interpretation divides the image into regions and attaches a meaning to each region. One choice of the overall interpretation evaluation would be attained by considering each region independently. If for a given partition of the image into regions we have $R(i)$ regions, then the interpretation assigns label $INT(i)$ to region $R(i)$. The values of $INT(i)$ will be sky, grass, road, etc., depending on the context and goals. If we assume independence between region features, we want to maximize the expression

$$\prod_{R(i)} P[R(i) \text{ is } INT(i) \mid \text{context, values of measurements on } R(i)]$$

over all partitions of the image into regions and assignments of labels to regions. This is quite conventional so far and is, in fact, too simple for our purposes. We want to account for two additional considerations. First we must use the model to get a good segmentation of the image into regions. For example, we might want

to merge green, yellow and brown patches to create the whole area that we call grass. Secondly we want to use additional semantic constraints (like the grass is below the sky) to influence the total probability that an analysis of the scene is correct.

In an attempt to enrich the semantic structure to support more of the problem knowledge and to provide for a control mechanism on the region growing algorithm, the semantic structure was allowed to have also a "first order structure". In addition to the properties of each individual region, we have, for each pair of adjacent regions of some interpretation, expected relative properties and some expected features of their common boundary line. For instance, if we have two adjacent regions, one of which is named "sky" and the other "hill", then we expect that the sky is above the hill, is a brighter blue than the hills, and that the boundary is usually a more or less horizontal, smooth line. The relative properties are usually more significant than the absolute properties since they are less sensitive to variation between pictures. This semantic model is too limited to describe all that is known of a scene, but many classes of scenes can be segmented properly with first order methods. The model is limited to first order to avoid the combinatorial explosion in the number of terms that have to be considered.

Remember that we want to get a partition of the input and interpretation for the regions (segments) and boundaries so as to maximize the likelihood of having the right interpretation. Let $R(i)$ be the i -th region, $B(i,j)$ the boundary between region $R(i)$ and $R(j)$ (if it exists) and the label of $R(i)$ be $INT(i)$. Then with our first order assumption, the expression that we want to maximize is:

eq. 1

$$\begin{aligned}
 &P[\text{global interpretation} \mid \text{values of measurements}] = \\
 &\quad \prod_{R(i)} P[R(i) \text{ is } INT(i) \mid \text{values of measurements on } R(i)] \\
 & * \prod_{B(i,j)} P[B(i,j) \text{ is between } INT(i) \text{ and } INT(j) \\
 &\quad \quad \quad \mid B(i,j)\text{'s measurements}]
 \end{aligned}$$

The use of eq. 1 represents more than just our belief that properties of individual regions and boundaries will suffice for our semantics. It also entails an assumption that the probability can be factored into the product above. This amounts to assuming that the probabilities of interpretations of each region (boundary) are dependent on the local properties of the individual region (boundary) and are independent of all other measurements. The interpretations of regions and boundaries are tied only by the consistency constraint,

that is, a boundary $B(i,j)$ which is the boundary between $R(i)$ and $R(j)$ must be evaluated as a boundary between $INT(i)$ and $INT(j)$, where $R(i)$ is labeled $INT(i)$ and $R(j)$ is labeled $INT(j)$. For example, if $INT(i)$ is "sky" and $INT(j)$ is "hill", the evaluation of the common boundary of $R(i)$ and $R(j)$ will include factors involving the expected direction, smoothness, etc. of a boundary between sky and hill. These factors are assumed to be independent of the particular color etc. of the sky and hill. This assumption that we can find local properties for regions that will be independent of both the relative properties of the regions and the boundaries' properties is essential in making our approach feasible. Assuming independence, we do not need to consider all cross combinations of the two classes of features. For instance if we have sky that may be cloudy or bright then we will use boundary properties of the sky with the hill which are independent of the particular type of sky. However, if such properties are insufficient to classify the sky boundaries, we will have to use two separate objects cloudy sky and bright sky each as separate possible interpretations. If the independence assumption seems to be unreasonable, consider the following argument:

.eq 2

$$P [\text{interpretation} \mid \text{values of measurement, context}] = \\ P [\text{values of measurements} \mid \text{interpretation, context}] \\ * P(\text{interpretation a priori} \mid \text{context}) \\ / P(\text{values of measurements} \mid \text{context})$$

Now

$$P [\text{values of measurements on } R(i) \mid R(i) \text{ is INT}(i), \text{ context}]$$

and

$$P [\text{values of measurements on } B(i,j) \mid R(i) \text{ is INT}(i) \text{ and } R(j) \text{ is} \\ \text{INT}(j), \text{ context}]$$

are plausibly considered independent of each other. A similar argument can be used for the factorization of the other two terms in the expression on the right of eq 1.

4.3 THE TASK OF THE WORLD MODEL

For a given utility function (in our case the maximum likelihood eq. 1) there are standard techniques in decision theory for finding the maximum utility. Unfortunately, the general techniques are too slow and much of our effort has gone into developing algorithms for efficiently computing an approximately optimal partition. The region growing algorithm starts with many small regions, and on each iteration, merges two adjacent regions (regions with a common boundary). The two basic decisions are which pair of regions to merge on each iteration and when to stop the algorithm. These two decisions can be controlled directly by the limited probabilistic semantic world model that we have. In general, on each iteration of the weakest-boundary-first region growing, the pair of regions whose common boundary is the weakest in the current image partition will be merged. Hence the control of the region growing algorithm is by evaluation of the boundary strength. We will show how our semantic representation can be used directly to compute the boundary strength. Alternatively, we can grow regions based directly on assignment procedures [see Subsection 6.2].

The second task of the semantics is to produce the stopping criterion. In our case we want to maximize:

P[interpretation | measurement values, context].

Ideally, the optimal partition will be the one that has the interpretation which maximizes this likelihood estimate over all partitions and all possible interpretations of partitions. In order to have an effective way to determine that probability, we need a relatively fast way to compute or estimate for a given partition the value of its optimal interpretation. In the next section, we will describe relatively fast methods for computing upper and lower bounds on the optimal value of the probability of a given partition. These bounds will be used as follows: The algorithm will collapse regions, and generate a sequence of image partitions. For each partition generated, the bounds on the possible value of the best interpretation will be evaluated. Then, when the region collapsing has been carried too far (as observed by a strong decline of the possible state value) the system will back-up to the most promising partitions observed while growing the regions (as indicated by the lower and upper bounds estimates of the quality of the partition observed). Next we will search for the best interpretation for the partitions observed whose bounds were high enough to make it possible that they are the best partitions observed. The current algorithm will simply choose the best of these, but more sophisticated procedures can be used if necessary.

It turned out that maximizing the utility (eq. 1) for a given partition frequently yields the best global interpretation. However to compare different partitions we need some modifications that are not available in our current implementation. The major modification required is teaching the system about properties of false merges, that is, what are the properties of a region resulting from merging two regions that should not be merged. In the current implementation the system is taught on false boundaries (that is, boundaries between sub-regions of the same terminal region like a boundary between part of a hill and another part of a hill). When evaluating the quality of a partition we should not allow any region to be interpreted as a merge of two regions of different meaning, and no boundary should be interpreted as a boundary between two regions of the same meaning. Currently we use a different approach. We allow false boundaries in the interpretation. If any of the boundaries is interpreted as false for the best interpretation found for the current structure, then we continue merging. Otherwise we stop. The assignment procedure used is described in Subsection 6.1.

4.4 EXTENSIONS OF MODEL TO 3-D

The simplified structure of first order relations between objects is just an approximation of the real world. It is clear that more involved relations between regions and boundaries hold. One group of such relations is the relation whose terms are the regions and boundaries meeting at a vertex. These relations were found to be key relations in analyzing plane surfaced objects [WAL GG], mainly, as constraints on the 3-d structure of the surfaces and boundaries on that vertex. If 3-d structure analysis were added to the model then the vertices would be essential. In this case we would have three classes of objects: regions, boundaries between regions, and vertices (intersections of several boundary lines). For each class of objects each object can take one of a few possible meanings which will be its interpretation.

The interpretation for a region will be the name of the 3-d surface for which the region is part of the image. (We say part to provide for partial occlusions or for the early stages of the region growing algorithm, when the regions are only portions of what they should be). Some such interpretations are: hill, road, horizontal face of cube, or the x-ray image of a rib. In addition to naming regions, some assumption about their 3-d structure will be made (like orientation, distance, etc.).

The boundaries will be named as boundaries between two regions of some meaning e.g the boundary between sky and hill. In addition, each boundary will have its own interpretation, which is the 3-d structure associated with it. If a boundary is the common boundary of regions X and Y then it may happen that:

- 1) X occludes Y.
- 2) Y occludes X.
- 3) X and Y create a convex corner.
- 4) X and Y create a concave corner.

(concave or convex relative to the included solid volume)

- 5) X and Y surfaces are smoothly continuous.

(There may be other more complex 3-d structures which we will ignore currently).

Vertices are the intersections of several boundary lines. The vertices were found to be extremely important in processing scenes of planar surface objects. Their main use was to constrain the geometrical structure associated with the boundaries. In scenes of curved surface objects their role may diminish, but it seems that they are going to be an important tool. In our current

interpretation they are not used, mainly because we are not trying to solve explicitly for the 3-d structure. An important extension of the current system would cope with the 3-d structure achievable through the use of the vertex and boundary 3-d structure.

In addition to being potentially useful in 3-d analysis, vertex properties may turn out to be useful for adding edge following information to the region grower. That is, we can check to what extent the regions and boundaries meeting at the vertex continue each other. Hence vertex properties may aid in boundary strength evaluation and the interpretation procedures.

4.5 EXTENSION OF MODEL TO INCLUDE GLOBAL CONTEXT PARAMETERS

One major deficiency of our system is the lack of global parameters which are changeable as information is collected. One such parameter could be the domain from which the current image is drawn. That is having the system also define the class of pictures from which the current image was drawn. For instance interpreting a region as a telephone or part of telephone will increase the probability that we have an indoor scene, while interpreting a region as a tree (or part

of tree) increases the probability that the observed scene is an outdoor image. It is likely that such capabilities can be added and tied easily to the tree search for optimal assignment as additional variables that are updated on each assignment of meaning to a region (see Subsection 6.1). Other parameters of this nature are orientation and position of the camera which observed the image. These parameters may scale all the features to normalize to standard observer orientation. In general when these parameters are used the context parameters will be additional variables that we will want to use in optimizing.

4.6 EXTENDED FIRST ORDER

The relations between regions that the current system observes are relations between pairs of adjacent regions. We may extend this to relations between any pair of regions. All the current structure and algorithms will remain valid with minor modification, but the combinatorics will grow prohibitively. If before we had, for N regions, approximately $4 \times N$ relations, now we will have N^2 relations to be considered in the various algorithms. There are ways to reduce the number of relations by restricting the classes of relations of

non adjacent regions. For instance, we might allow only relations between non-adjacent regions of specific meaning or of special relative properties.

SECTION 5

WEAKEST-BOUNDARY-FIRST REGION GROWER BASED ON SEMANTICS

5.1 SEMANTICS BOUNDARY STRENGTH EVALUATION

We return here to the description of the system. The initialization levels were used to reduce the problem to about 100 regions. Our next step is to try to evaluate the boundary strength based on the world model. This part of the algorithm first computes additional properties (like shape) of the regions and boundaries resulting from the initialization. It then assigns probabilities to the alternative interpretations of the regions, i.e. computes

$$P[R(i) \text{ is } X \mid \text{values of measurement on } R(i)].$$

The boundary strength may be evaluated by two related methods: 1) The probability that the boundary is a real boundary (a boundary between different objects in our semantic world model), and 2) the change in the value (probability of correctness) of the interpretation as a result of eliminating the boundary. We will describe here the first

of these which is the one currently used for the weakest boundary melted first region grower. The second method has some advantages and will be discussed below.

We approximate the probability of the boundary to be real as follows. The estimate of the probability that the boundary $B(i,j)$ which is between $R(i)$ and $R(j)$ should not be there (false boundary), is :

$$P_{\text{false}} = \frac{\sum X_{\text{int}}}{X_{\text{int}}}$$

$P[B(i,j) \text{ is a boundary between two subregions of } X \mid \text{measurement values on } B(i,j)]$

* $P[R(i) \text{ is } X \mid \text{measurement values on } R(i)]$

* $P[R(j) \text{ is } X \mid \text{measurement values on } R(j)]$.

The estimate of the probability that boundary B(i,j) is real is:

$$P_{\text{real}} = \frac{P(B(i,j) \text{ is a boundary between } X \text{ and } Y \mid \text{ measurement values on } B(i,j))}{P(B(i) \text{ is } X \mid \text{ measurement values on } R(i)) + P(B(j) \text{ is } Y \mid \text{ measurements' values on } R(j))}$$

P[B(i,j) is a boundary between X and Y |
measurement values on B(i,j)]

* P[R(i) is X | measurement values on R(i)]

* P[R(j) is Y | measurements' values on R(j)].

This is the Bayesian probability (which is in our case the utility) that, given the properties of the boundary and two regions defining it, the boundary is a boundary between sub-parts of images of different objects.

The strength of the boundary is then computed to be

$$\frac{P_{\text{real}}}{P_{\text{real}} + P_{\text{false}}}$$

($P_{\text{real}} + P_{\text{false}}$ may be different from 1 since the independency assumption is only approximation of the reality).

5.2 SEMANTIC STOPPING CRITERIA

We may apply three possible stopping criteria. The simplest one is the threshold stopping criterion, that is once the strength of the weakest boundary in the segmentation is above a certain threshold we quit merging. The second stopping criterion is to look for a good interpretation for the current segmentation and if there is no boundary which is interpreted as a false boundary then quit merging, otherwise continue merging (see Subsection 6.1 for the assignment algorithm). Alternatively we can use the state evaluation for backup and hence avoid using a stopping criterion. That is, back up to the segmentation with the highest state value observed while the region grower is working. The current interpretation provides for utilizing the first two options, or a combination of the two.

SECTION 6

STATE EVALUATION FOR A GIVEN IMAGE SEGMENTATION

State evaluation is required for effectively recognizing the most promising state of image segmentation. Evaluating an image partition will also involve a search for the best interpretation for all regions simultaneously, and hence will effectively provide a way for really understanding the scene. Currently, we use the state evaluation only as a procedure to assign meaning to all regions (and hence boundaries). The assignment procedure is used to verify that the system really understands the segmented image, and to provide for a stopping criterion for the region grower. The difference in state value could also be used in region merging as criterion for melting boundaries, though it is not being used this way in the current implementation.

6.1 INTERPRETATION OF THE SCENE - LOWER BOUND EVALUATION

A lower bound on the value of an image partition is computed by actually finding a good global interpretation using a simple fast

meaning assignment algorithm. Briefly, we take the region of highest confidence interpretation and assign to it its most probable interpretation. Next, using the boundary features of the newly assigned region, the probabilities of different interpretations of adjacent regions of the newly interpreted region is updated. Then the region of highest confidence from all un-interpreted regions is assigned, etc. This is essentially a depth first search of the tree of region interpretations and yields a value for the partition which is the desired lower bound. Extending this search to a full tree search would yield the optimal interpretation. More details on the sequential assignment process are given below.

Recall that we want to approximate the maximum possible value of the expression

eq 1 :

$$\prod_{R(i)} P[R(i) \text{ is } INT(i) \mid \text{values of measurements on } R(i)] *$$

$$* \prod_{B(k,l)} P[\text{boundary } B(k,l) \text{ between } R(k) \text{ and } R(l) \text{ is a boundary} \\ \text{between } INT(k) \text{ and } INT(l) \mid B(k,l) \text{ features}]$$

over all possible values of $INT(i)$ for a given picture partition.

The assignment algorithm that we use to estimate the best possible assignment of $INT(i)$ for all $R(i)$ for a given image partition is as follows:

- [1] Compute for each region the ratio (based just on local measurements of the region) between the most likely interpretation and the next most likely interpretation. This ratio will be called the $CONF1(REG)$. Let x_1 be such that

$$p\{ R(i) \text{ is } x_1 \mid \text{values of measurements on } R(i) \}$$

is maximized for R and let x_2 be such that

$$P\{ R(i) \text{ is } x_2 \mid \text{values of measurements on } R(i) \}$$

is the next highest. Then

$$\text{confi}(R(i)) = \frac{P\{ R(i) \text{ is } x_1 \mid \text{measurements of } R(i) \}}{P\{ R(i) \text{ is } x_2 \mid \text{measurements of } R(i) \}}$$

- [2] Sort the regions by their confidence ratio.

(3) Assign the region with highest confidence (the one with highest ratio) its most likely interpretation.

(4) Update probabilities for various interpretations of regions that are not currently assigned meaning, assuming that the last assignment is true. Let the region assigned most recently be $R(1)$ and its interpretation be $INT(1)$. Now if $R(i)$ has boundary $B(1,i)$ with $R(1)$, then for any interpretation x of $R(i)$ in evaluating eq 1 above, there will be a term of the form

$$P[R(i) \text{ is } x \mid \text{values of measurement on } R(i)]$$

from the first product and one of the form

$$P[B(1,i) \text{ is boundary between } INT(1) \text{ and } x \mid B(1,i)\text{'s features}]$$

from the second product. Since both terms have only one variable x now, a better approximation of the probability of $R(i)$ being x , assuming that $R(1)$ is $INT(1)$, is

$$P_{\text{new}}[R(i) \text{ is } x] = P_{\text{old}}[R(i) \text{ is } x] * \\ P[B(1,i) \text{ is a boundary between } x \text{ and } INT(1) \mid \\ B(1,i)\text{'s features}]$$

Thus we use the new information to find updated probabilities for the different possible assignments for $R(i)$, by counting the newly interpreted region $R(1)$.

We do that updating to all possible interpretations for all adjacent regions of $R(1)$.

[5] Compute the new confidence ratio and sort the regions by the new confidence ratios.

[6] If any region is still unassigned goto [3] else exit.

This process of assigning interpretations iteratively provides a good guess about the possible best interpretation, but it does not guarantee the total maximization of our product. We can extend the current algorithm into a full tree search (undoing some assignments and trying alternative ones) to get the best interpretation. This will be a depth first search in the tree of all possible assignments. Each node in the tree will stand for the assignment of an interpretation to a region. In all sons of such a node the assignment done in that node will be assumed to be true. The terminal nodes will stand for a totally interpreted scene. For efficiency purposes we can use various pruning and tree search techniques [NILSSON ch 3]

to reduce the number of terminal nodes needed to be observed to secure optimality. Our current algorithm is the portion of depth first tree search of the assignment tree up to the point where we get to the first terminal node (first global assignment).

One should also note that the same sequential assignment and extension into tree search can be applied to the extended first order world model described, where we allow relations between any two regions (not necessarily adjacent) if we continue to assume independence. The only difference is that the probabilities and confidence ratio of not only the adjacent regions of the newly interpreted region, but of all related regions, will have to be updated.

Working on extended models, where relations involving more than two variable assignments exist, will cause only minor changes. Whenever a region is interpreted, all the relations in which it appears will be reduced by one degree. That is, an n -ary relation will become an $n-1$ -ary relation. If there is an $n-1$ -ary relation already existing with the same variables, the two will be united (by multiplication). Except for that difference, everything will stay the same.

6.2 REGION GROWING BASED ON ASSIGNMENT

We use the assignment procedure described in the last section as a region grower by taking all pairs of adjacent regions that were assigned the same meaning and merging them. To avoid false merging, we consider all regions which were assigned meaning after the first assignment of a meaning to a region with low confidence level meaningless and hence not mergeable into other regions. This approach may be extended by adding into the meaning assignment algorithm another step [3.5].

[3.5] If any adjacent region of the newly interpreted region is already assigned a meaning identical with the meaning assigned to the newly interpreted region, then merge the two together. Undo the effect of the two small regions on their neighbors interpretations. From this point on, the unified region will be considered in updating probabilities of other, not yet interpreted regions.

We can use the two extensions (merging while assigning meaning, and full depth first tree search) together. This will generate a very reliable meaning assignment concurrent with a region growing procedure which has backup capabilities. It will, however, be relatively slow.

6.3 UPPER BOUND FOR STATE VALUE

The upper bound could be computed by relaxing the consistency constraint. This condition means that a boundary between two regions of known interpretation has to be counted in as a boundary between those two interpretations. We could relax this condition by breaking the product (eq.1) into local sub-products and finding the best local interpretation for the terms involved in this subproduct. We would take the best possible value for each sub-product separately, and multiply them, with proper scaling of common terms. This would result in an upper bound on the value of the best global interpretation. For example such relaxation is to consider all regions and boundaries independently and to assign for each the best possible interpretation considering only its own properties. The product of all these probabilities is an upper bound on the value of eq 1. It is this sort of estimate which could be used to approximate the single step improvement in the second method of boundary evaluation mentioned above. An exact computation of the change in interpretation value would be too time consuming. We do not yet know whether this boundary strength computation will be better than the one described in subsection 6.1.

The local upper bound estimation may be used also to get more

reliable evaluation of the meaning of a region, by considering also its neighbors in evaluating the probabilities of different interpretations for that region. This is analogous to various graph isomorphism algorithms, which use deeper structure around a node for finer node type classification.

SECTION 7

THE PROBABILITY SYSTEM IMPLEMENTATION

7.1 GENERAL CONSIDERATION

Up to now we were dealing with our probabilities in abstract terms without worrying about how to get these probabilities, or which measurements (features) to consider. This problem is actually one that appears generally in pattern recognition problems and decision analysis. The general problem is to try to develop a classification system for the objects which will be able to indicate often and with high probability the real meaning of the object. This section describes the structure of the probability model implementation in general terms. The next section describes the specific measurements applied to our two classes of objects: regions and boundaries.

The thing that makes our case somewhat special is the fact that the probabilities are dependent upon themselves. In the region grower algorithms the decision as to which pair of regions to merge is based on the probabilistic world model. We are working with probabilities

of events produced by the algorithm (remember that p[interpretation | measurement] means the probability that an object with those measurements will be produced by the region grower algorithm as part of an object of that interpretation). For that reason we should be careful about generating "steady state" probabilities. In practice the recursive effect was ignored with the assumption that the model will be stabilized after a few learning cycles. However, this effect should be modeled theoretically to see the effect of the recursive relation.

Another difficulty is the effect of the state of the algorithm. There are good reasons to assume that the probabilities of occurrences of events depend on the state of the algorithm. In the early stages of the algorithm there will be quite a few small regions which are portions of the regions with which we want to terminate. In an advanced state of the algorithm most of the regions will be bigger and near the whole terminating region. For this reason it is desirable to break the model into sub-models, each of them applied in different stages of the algorithm. An indicator for the state of the algorithm is the number of regions. In our practical implementation the process was broken into only two phases: the initialization, where no attempt to assign meaning was done, and the second portion which exploited the world model. In the initialization we used only

a uniformly applied procedure for boundary strength evaluation [section 3]. In the second portion, the probability model is used, and all the probabilities are counted only over that portion of the region grower.

The system has to provide for experimental collecting of the probabilities. We need to collect the probabilities of measurements experimentally. It turned out that in most cases we did not have a good a priori idea of the distribution of the measurements and the program had to learn them experimentally. Apparently, in most cases our conscious knowledge of the visual world provides only a very rough idea of the distribution of measurements.

7.2 THE PROBABILITY APPROXIMATION TECHNIQUE

At present we use a simple form of learning in which the computer only helps in updating the probability estimates inside a given classification scheme. This is a version of the traditional non-parametric adjustment of the probability density function. In the future we intend to use a more advanced learning phase in which the program will keep a complete historical list of objects observed in

the past along with their real interpretations, and will use it to improve the existing classification scheme. In both cases the learning will be supervised (since the intended meaning of the training objects will be given manually).

The probabilistic model that we have is as follows: we have $\alpha(i)$ $i=1, N$ possible parameters (meaning of the object). Picking an object randomly, it will be of type $\alpha(i)$ with a priori probability $P_{\alpha(i)}$ (i). We are given a set $X(j)$ $j=1, M$ of random variables a priori

which are our measurements. We try to estimate $P_{\alpha(i)}(\vec{X})$ (that is

the probability that an object of type $\alpha(i)$ will have properties \vec{X}).

We do that by estimating $P_{\alpha(i)}(S)$ where S is a subset of \vec{X} space (the

features space). Once these two terms are available (approximated) we can compute the Bayesian probability (likelihood) that an unknown object whose measurements fell into S is drawn from $\alpha(i)$ as:

$$\frac{P_{\alpha(i)}(S) * P_{\alpha(i)} \text{ a priori}}{\sum_j P_{\alpha(j)}(S) * P_{\alpha(j)} \text{ a priori}}$$

$$\sum_j P_{\alpha(j)}(S) * P_{\alpha(j)} \text{ a priori}$$

We use sequential classification estimates to generate a good cell structure. Our task is to estimate the joint probability of the random variables in this space. This is done by breaking the random variable space into cells $S(i)$ (not necessarily Cartesian) and assuming that the densities (of each of the probability density functions) are uniform on the subcell. Using training runs, we count the number of objects of each meaning whose measurements fall into a cell. This gives us an effective way of estimating the Bayesian probability (likelihood) that an object is of some meaning if its measurements fell into a cell. This estimate is the standard Bayesian probability estimate and in our case is

$$p(\text{object is } \alpha(i) \mid \text{measurements fall into the cell}) =$$

$$\frac{\text{\# of objects of meaning } \alpha(i) \text{ whose measurements fell into that cell}}{\text{total \# of objects observed whose measurements fell into that cell}}$$

Our task is to break the random variable space (that is the space of all possible combinations of measurements) into cells that will enable us to get an effective classification. That is, given that the values of the measurements of an object fall into some cell, we frequently want to have a high probability estimate for the real

meaning of the object. Given such a fixed partition of the random variable space into cells, learning the probabilities of different interpretations of objects whose measurement values fall into a cell will be done automatically. This is done by simply keeping, for each cell and for each possible interpretation, the count of how many times in the past the value of the measurements of objects of that interpretation fell into this cell. The real meaning of the objects is indicated manually (supervised learning), and the learning is applied for both regions and boundaries.

7.3 THE CLASSIFICATION TREE

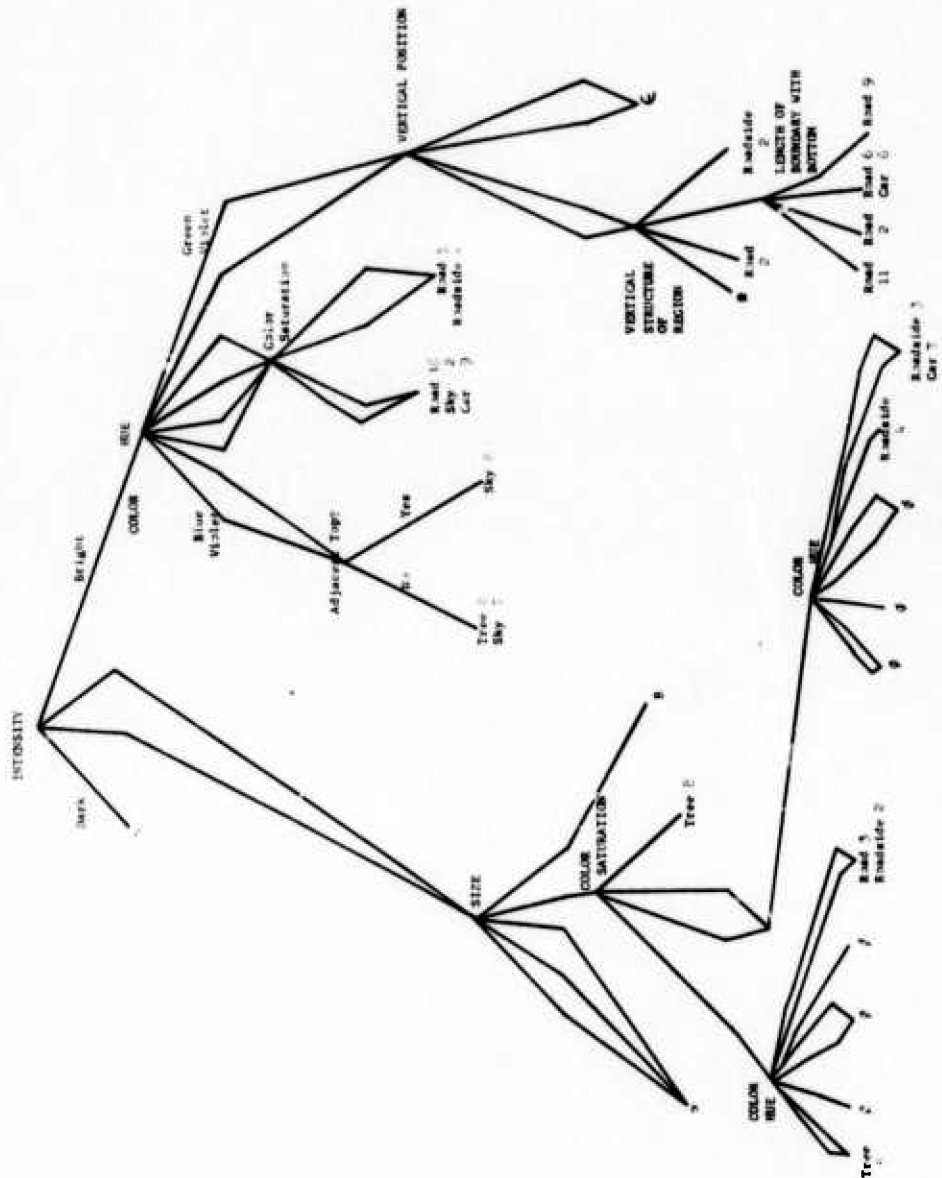
This brings us to the classification tree structure which tries to generate a cell structure with as few cells as possible while attaining a good classification among the possible interpretations. It is critical to keep the number of cells down. Otherwise the whole approach becomes impractical. For this purpose we utilize an augmented decision tree whose leaves correspond to the cells into which we broke the space of all possible combinations of measurements. This structure, which is a version of sequential classification, enables us to treat in a special way special sub-

spaces and hence to apply very special-purpose classifying procedures when necessary. The option to use an augmented tree allows us also to utilize, if known, the independency of some measurements and thus reduce the number of terminal nodes.

The classification tree is quite standard. It corresponds to sequential application of measurements. In each call the current measurement called depends on the values returned by the previous measurement. This way we may apply very specialized measurements if necessary to classify objects, and still keep the classification inexpensive since the special measurements will be used only when needed, as indicated by results of already evaluated measurements. By calling only on very effective features the number of terminal nodes is minimized, and this way we still have an effective way of computing the probabilities (keeping the counts for each terminal node).

The tree structure is as follows. There are three types of nodes: terminal nodes, parallel branch nodes and function call nodes. A terminal node stands for a subspace of the random variable space. With each terminal node we keep counters of how many times in the past the measurements of objects of some interpretation fell into this subspace. A function call has an integer function associated

with it. This function can return a number from 1 to n where n is the number of sons of that node. This function is a function of the measurements of the object. Each of the sons corresponds to part of the subspace associated with the father node which includes all points of the father's subspace for which the function returns that value. That is, if a function has n values we break the subspace associated with the node into n subspaces, one for each possible answer. Obviously the root of the tree has the whole space associated with it (all possible combinations of measurements). We allow also for branch nodes where we allow several independent branches to propagate from them on in parallel, and the value propagated from that node back up will be the product of each of the sons multiplied, and scaled to one (see below). The parallel branch nodes were allowed in order to reduce the number of terminal nodes when it is known that some features may be treated independently. For instance, we may want to treat color features independently from shape features of a region. Suppose that the color feature space was broken into n cells (equivalent to having n terminal nodes for classification based only on color), and suppose that the shape features give us m terminals. Then, treating them without assuming independence, if we consider all possible combinations, there will be $n*m$ terminals. Treating them assuming independence will produce $n+m$ terminal nodes.



a, b, c, e, are branches that g' deeper and are not shown because of lack of space.
 Two branches that meet means that the full separation available was not used.
 The counts at terminal nodes indicate the number of objects of that meaning which fell into the node.
 ' means that n objects fell yet to that node.

PART OF CLASSIFICATION TREE FOR REGIONS USED IN THE FIRST PROBLEM DOMAIN

7.4 PROBABILITY COMPUTATION

Computing the probability vector (the probability for each interpretation) for a given object for a fixed tree structure is done by a recursive procedure described below. In the description the value returned is always a vector of all the probabilities of each interpretation for that object. All these values will be non-negative. By scaling such a vector to one we mean that we sum all those non-negative numbers and divide each of the numbers by the sum so the new sum will be one. The product (division) of two such vectors means here the pointwise multiplication (division) of the elements of the vectors which results in a vector and not the scalar product. $\vec{P}_{\text{a priori}}$ is the vector which for each interpretation has the probability that an object picked at random will be of that interpretation.

The probability vector returned is:

$$\vec{\text{answer}} = f(\text{top_node}) * \vec{P}_{\text{a priori}}$$

scaled to one.

$\vec{P}_{\text{a priori}}$ is the vector resulting from scaling to one the vector whose i -th element is the total # of objects of type i observed).

$f(\text{node}) =$

if node is a terminal then the returned vector is:

(count of occurrences of obj1 in that node
total number of counts of obj1

count of occurrences of obj2 in that node
total number of counts of obj2

.
.
.

counts of occurrences of objn in that node
total number of counts of objn)

scaled to one.

If the node is a function call then

$$f(\text{node}) = f(i\text{-th son of node})$$

where:

i is the value returned for the current object by function associated with the node.

If the node is a parallel branch node then

$$f(\text{node}) = \prod_{\text{son}} f(\text{son}) \quad \text{scaled to one.}$$

7.5 LEARNING: PROBABILITY ADJUSTMENT

Keeping the counts of occurrences of each interpretation for each terminal node is done by pointing at an object and indicating its intended meaning (the meaning the user likes it to have). The program then increments the count associated with this interpretation in all

terminals into which the properties of the current object lead. We may have several such because of parallel branch nodes where we have one terminal for each parallel branch. To ease the chore of naming regions (and hence boundaries) we developed an interactive graphic system. The system displays one region at a time by drawing its boundaries over the original image on a television monitor and asks for its meaning. Once all regions are interpreted, there is an option to have a training run in which the region grower makes use of what it knows on the real meaning of regions in order to increment the counts associated with the real meaning of the object every time an object is observed, while growing the regions in all terminal nodes into which the properties of the object lead (an object is either region or boundary).

7.6 LEARNING: TREE GENERATION

At present, generating the tree and increasing its effectiveness are done interactively. The user may look at terminal nodes that cause errors in the region grower (in the training runs, these errors are detected automatically by the system), or at terminal nodes where the classification is not reliable. The latter are terminal nodes where

there are relatively many occurrences of objects which are not of the meaning which is the most frequent meaning in that terminal cell. What the user can do in this case is to change the terminal node into a non-terminal node, hence replace the cell associated with that node by finer sub-cells such that in each of the smaller cells the classification will be more reliable.

In the future we intend to use an automatic system to generate the classification tree. For an automatic generation of a sub-optimal classification tree the system will keep a historical list which contains objects observed in the past, their properties and their real meanings. Based on this history the system could try to order the application of measurements so as to get good and cheap classification, by creating as few as possible cells (leaves), and still keeping the good classification probability high. It will be able to point out cells that are not sufficiently discriminating so that they may be worked on interactively (as it is now) or automatically (mainly breaking each such cell into finer subcells such that, for each subcell the classification is more reliable). Techniques for organizing the classification tree so as to get near-optimal sequential classification are described in [SR]. In [SR] the tree generation is considered as a game with nature where the score is a quality measure of the classification. Game (α - β) type tree search is utilized in creating the decision tree.

These types of learning are general to many pattern recognition and sequential decision problems. A vast amount of research, both theoretical and experimental, has been done in this area. [FU] is a good description of the theory and [DH] is a good introduction to the various applicable techniques. [SR] which was mentioned above is an interesting example of trying practical automatic generation of a sub-optimal classification tree.

7.7 LEARNING: GENERATING NEW CLASSIFIERS

One additional phase of learning is generating the discriminating procedure. This may be both setting thresholds for already available real-valued functions (to get integer answers), and the generating of the functions themselves. There are some standard techniques for generating such functions, mainly various linear discrimination procedures (see DH). It is not reasonable to assume though, that this level will be automatic in the near future and it is likely that generation of discriminating functions will rely on human intuition.

7.8 NEAREST NEIGHBOR CONSIDERATIONS

It is interesting to compare our technique with the nearest neighbor classification which is investigated in various papers [COV]. This principle is to take for a new unknown occurrence of an object the interpretation of the object observed in the past whose features are nearest to the features of the new object. There are two deficiencies in this approach. First, only rarely is there an obvious metric on the space of values of measurements, and hence only rarely is it clear exactly how to measure distance between the feature vectors of two objects. Secondly, it is very hard to search for the nearest object observed in the past (unless we are in one dimension) since we have to compute the distance from all examples observed to get the minimal distance. An effective way of reducing the search time will call for breaking the space into cells the way we do. That is, locating first the cell into which the measurements of the new object fall and then searching for the nearest one only among known objects whose measurements fall into that cell (and stored associatively with that cell), ignoring objects which fall into other cells. Thirdly, the answer returned is just one possible interpretation and not a list of different possible interpretations with various probabilities. Extending the nearest neighbor principle to find the n -nearest objects and computing the probabilities of

different interpretations based on them will make the computation even less efficient because of search time and will force even more reliance on space partitioning than the method we currently use.

It seems that when the historical list is added to our system to allow automatic generation of the classification tree, then we will have associated with each cell the properties and meaning of objects which fell into the cell. In this case it may be worthwhile to use versions of nearest neighbor classification or some continuous parameter probability adjustment procedures for each cell to improve the classification.

SECTION 8

RESULTS

8.1 PROBLEM DOMAINS

We applied our software system to two picture domains. The first domain was road scenes as they may be seen while driving a car. The second domain was left ventricular angiograms (x-ray images of the left ventricle made visible by injection of a radio-opaque dye). These angiograms are useful for various cardiologic applications since they allow observation of myocardial movement. In the first domain the system was taught about existence and properties of regions which are whole or parts of images of the following objects: sky, tree, road, car, shadow of cars and roadside vegetation. The semantics used in the second domain described the heart interior, chest cavity background and the dark frame border. Illustrations given at the end of this section indicate the results of the experiments. All the pictures are taken from a computer graphic terminal with gray level capabilities. There are six bits available per image point. Five bits are used for displaying the original

picture, and the high order bit is used for the overlay of the boundary lines between regions.

The library of integer value procedures currently available for generating the classification tree nodes for regions and boundaries is still quite limited. We have only crude estimates of the features of regions and boundaries, and there is still a long way to go before a good description system is available. Our attempt was mainly to implement the ideas presented in this thesis on the A.I. laboratory hardware-software system to prove the feasibility and effectiveness of our approach. We consider the result a positive indication of the feasibility of getting an automatic analysis of real world images by computer.

The properties which are currently available in the system are described below. Before getting into their detailed description I would like to make the following general comment. It turned out that individual region properties are very much special purpose mainly because of the weakness of the shape descriptor. Variations between pictures, and the necessity of classifying sub-regions of the terminal regions (as produced by the region growing algorithms) are mainly responsible for the weakness of any classification based on region properties alone. The weakness of classification based solely

on regions features is very significant near initialization when almost only local (point-wise) properties (light intensity color and position) are available. However when we consider boundaries and relative properties of regions the description becomes much more general and less sensitive to variations between pictures.

8.2 REGION PROPERTIES

The region properties available are:

- [1] The size, computed as size of region relative to the whole picture area (five degrees (ranges) logarithmically).
- [2] Vertical position of center of gravity of the region in the picture frame (five degrees).
- [3] Horizontal position of center of gravity of the region in picture frame (five degrees).
- [4] Does the region touch top of picture frame? (yes/no)
- [5] Does the region touch bottom of picture frame? (yes/no).

- [6] Length of boundary with bottom of picture (four degrees).
- [7] Average light intensity of the region relative to the histogram of the light intensity in the entire image (four degrees).
- [8] Color saturation (4 degrees).
- [9] Color hue (eight degrees)
- [10] Does the region touch the frame of the picture on the side? (yes/no).
- [11] Ratio of height to width of the minimal upright rectangle which bounds the region. This rectangle has vertical and horizontal sides.
- [12] Position of center of gravity of the region relative to the center of the minimal upright rectangle which bounds the region.

8.3 BOUNDARY FEATURES

The boundary properties available are:

- [1] Light intensity differences between the two regions defining the boundaries (six degrees).
- [2] Shape of boundary based on breaking into four sets the pairs of sample points which define the boundary. The four classes of pairs of points are defined by the position of the sample point from the reference region relative to the other sample point in the pair it may be below, above, left or right. For each of the four sets we compute whether it is null, and if it is not the average location of the points in the set. See Figure 8.1 for the twenty one basic boundary types which this procedure recognizes. (21 degrees).
- [3] Relative size of the two regions (6 degrees).
- [4] Boundary length relative to the length of the whole image perimeter (5 degrees).
- [5] Relative position of the two minimal upright rectangles that

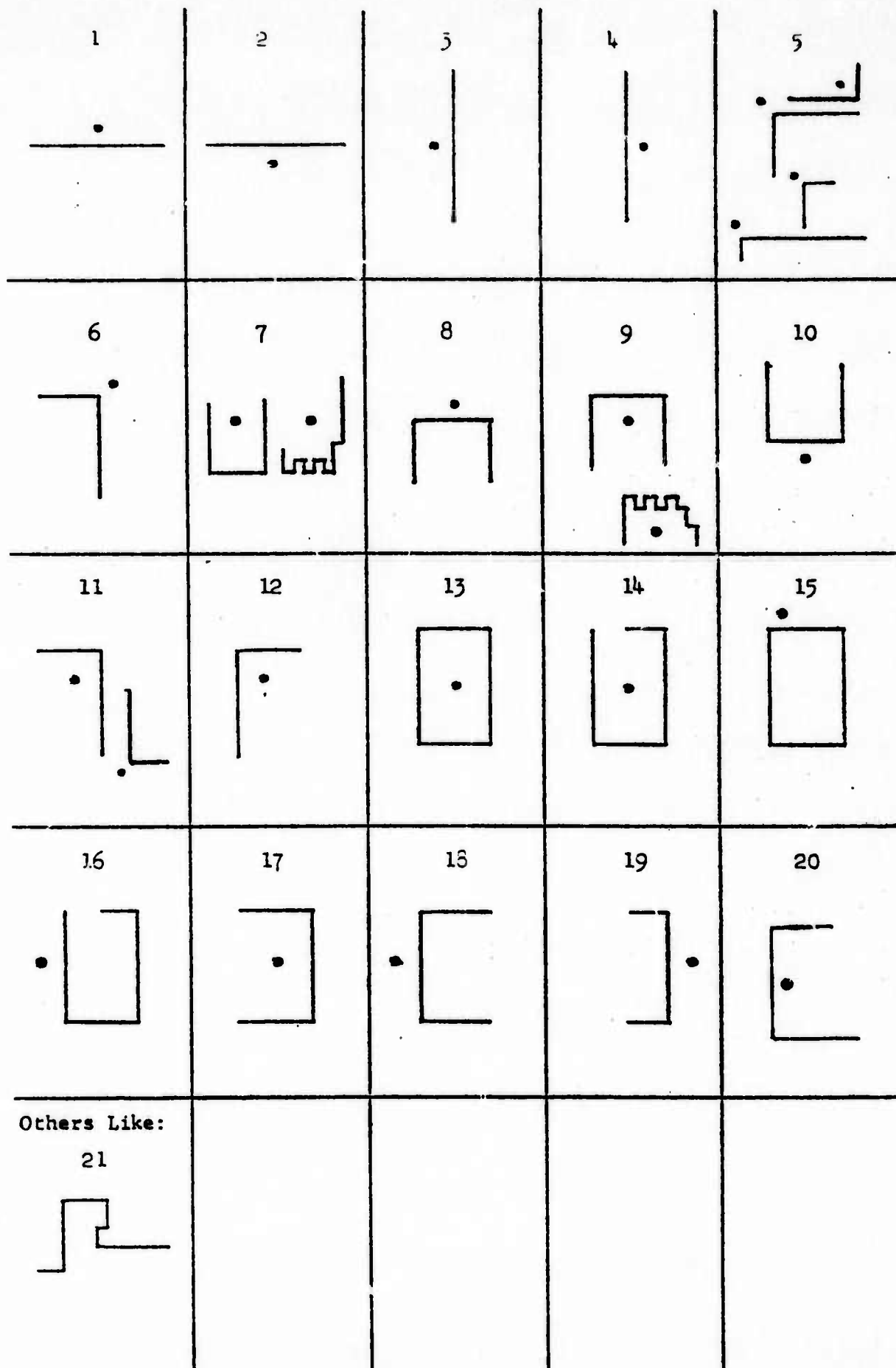
bound the two regions defining the boundaries [4 functions with 5 possible classes for each].

[6] Location of center of gravity of boundary in picture frame (5 horizontal degrees and 5 vertical).

[7] Some quantitative measurements on the relative length of the boundary in the four directions defined in [2].

[8] Color differences between the two regions (4 functions with 3 degrees each).

. - Stands for the Side of the Reference Region



8.4 RESULTS

In the first problem domain, the road scenes, the interpretation for regions can be road, tree, sky, roadside vegetation, car and shadow of car. The possible boundary interpretations are all the 6x6 combinations possible (remember that boundary properties are asymmetric with respect to the reference region). The learning (collecting the probabilities and interactively refining and extending the classification tree) was done by training the system on five pictures and then the collected probability estimates were applied to another five pictures and worked successfully. (See illustrations below for some sample runs). The non-semantic weakest-boundary-first region grower threshold was set to .13 of the maximum possible boundary strength, or 100 regions, whichever came first. The semantic weakest-boundary-first region grower was stopped with strength threshold 0.1. From that point on the region grower derived from the sequential assignment was used until no two adjacent regions were interpreted as parts of the same object. The total computing time for processing one picture is about five minutes. We believe it

can be speeded up on the current hardware by a factor of 100 by optimizing the code and data structures.

Illustration C is a good demonstration of some of the limitations of the system. First, notice that the small car on the top right part of the road is considered to be part of the roadside vegetation. If we used the relative position of the two we would have done better. The major difficulty is that in this case we need more involved relations than the purely first order ones available now. We may need to consider the road, the small car and the roadside vegetation, in order to distinguish the small car from similar structure of roadside vegetation and road on the bottom left part of the picture. Also better shape descriptors are needed in order to recognize more accurately the boundary between the car and its shadow.

The assignment algorithm is driven by the confidence values of regions (the ratio between the probability of the most likely interpretation and the probability of the next most likely interpretation). The recognition of the bigger regions like the sky, road and the bigger parts of the trees and roadside vegetation usually have unique interpretations even on the basis of local region properties alone, and hence the assignment usually starts by assigning them their correct meaning. For instance the bigger part

of the sky is unique because there is only the sky which is big, is very bright, touches the top of the picture along a long line and has nearly blue color. The bigger part of the road is also unique because only it can touch the bottom of the picture with a long line and be relatively bright and almost colorless and horizontally near the center. The bigger parts of the tree are usually unique because they are big, very dark and near the top of the image (see an example of classification for demonstration). Only later are regions which are parts of the car or its shadow interpreted, based on their local properties and the structure of their boundaries with the road and the bigger roadside vegetation areas. Later still smaller parts of the roadside vegetation and trees are interpreted, mainly because it is usually unclear which of the two interpretations to assign to them. In cases where we are looking for the road, we may use a utility that assigns a very low price to a confusion between a tree and roadside vegetation, because such confusion has only a minor effect on the analysis of the road. Currently though, we assign equal value for all errors.

In the problem domain of the left ventricular angiograms, no color was available. As a result light intensity, position and shape are the major recognition tools. In addition the non-semantic region grower had to stop at a relatively early stage because of noise and

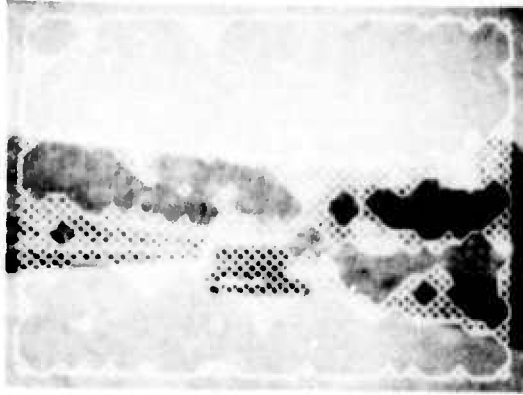
lack of high contrast border. The number of regions on termination of the non-semantic region grower was two hundred. From that point on the sequential assignment region grower was used (see illustration). The run time was again around five minutes per image. It is encouraging that the adjustment to the second domain was very easy. We hope that in the future a general and rich library of feature extracting routines with the capability of working on many models will be developed.

8.5 CONCLUSIONS

The successful application of the system to two problem domains is very encouraging. Especially so, because it is clear that we can do much better on each of the components of the system. The author knows of no previous system able to work on such complex images successfully. Our system is also based on a general structure that provides hooks for incorporating sophisticated subsystems for each of the components. This paper suggests in many places ways of improving the current implementation. The author believes that major improvements may be achieved by the following developments: first, automatically generating a sub-optimal classification tree; second,

improving the available shape classifiers (which will require the option to pass from the sample point description to the finer grid point description); and third, adding options for more complex relations in the semantics representation.

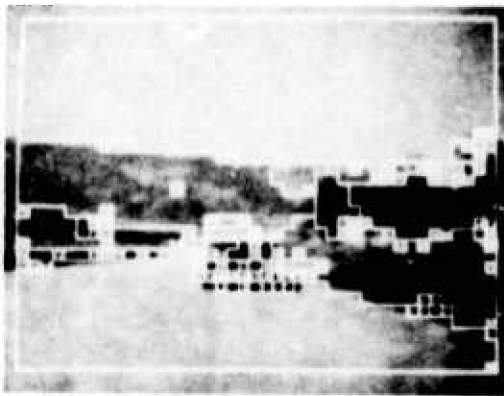
To conclude, the generality of the ideas behind the system provides for ways of incorporating improvements and special knowledge in every one of the components. The author hopes that the generality of the system will enable researchers to concentrate on each of the components separately of the system, hence allowing this young experimental research field to mature as an unified research field.



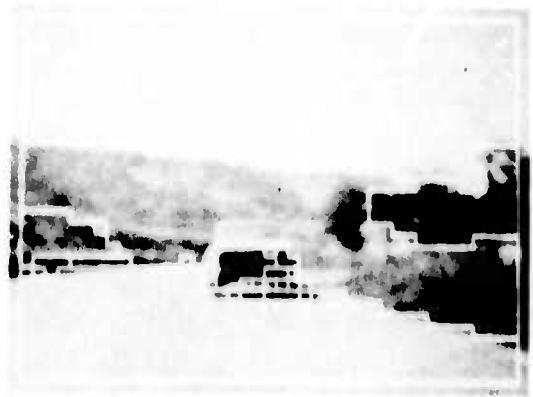
(A-1) The output of the segmentation based on path connectivity when it is stopped by the default stopping criterion. The resulting image is segmented into a few hundred regions.



(A-2) The effect of reducing the number of regions to 40 using the path connectivity region grower with a more liberal threshold than our current stopping threshold.



(A-3) The output of the region grower which melts weakest boundary first, with non-semantic boundary strength evaluation. This is the result of stopping with the default stopping criterion.



(A-4) Result of merging regions down to 30 regions using weakest boundary first algorithm and non-semantic boundary strength evaluation. Note that the top of the car is melted into the roadside vegetation.



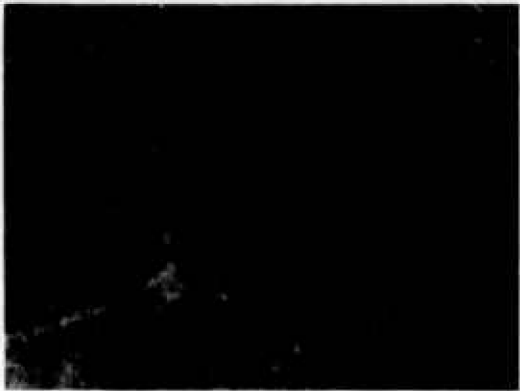
(A-5) Result of attempt to reduce the number of regions to 20 without using semantics (melting weakest boundary first non-semantic boundary strength evaluation).



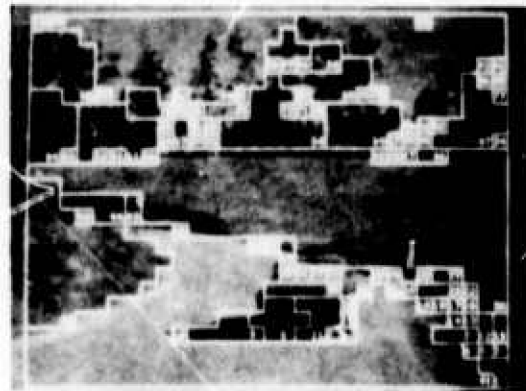
(A-6) Output of region grower based on semantics. (Melting weakest boundary first where boundary strength is computed using the semantic world model).



(A-7) Final grouping of regions based on the interpretation assigned to them by the world model. Regions whose meaning was assigned with confidence less than 10 are not mergable. They occur usually on the real boundary between two regions.



(B-1) Original picture.



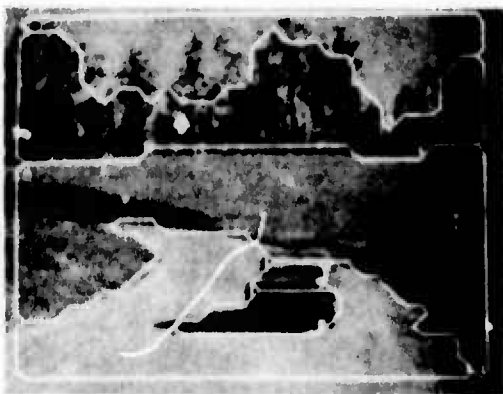
(B-2) Output of the non-semantic weakest boundary melted first region grower.



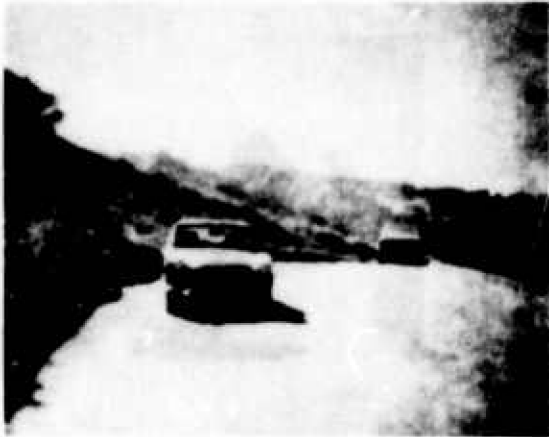
(B-3) Output of the semantic based region grower



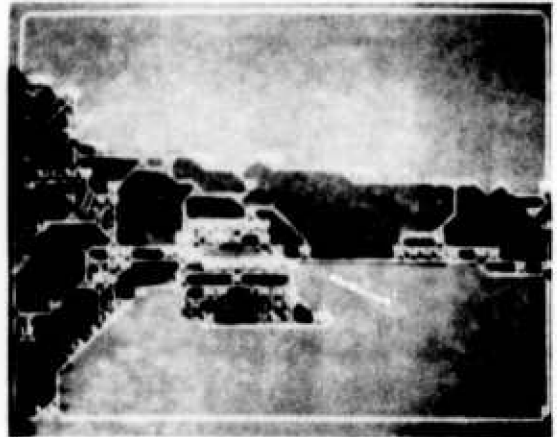
(B-4) Result of grouping regions by their assigned meaning, taking only regions which were assigned meaning with confidence greater than 10.



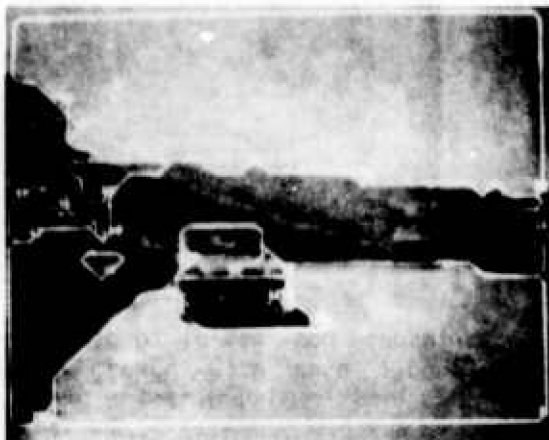
(B-5) Grouping regions by their assigned meaning, all regions considered mergable.



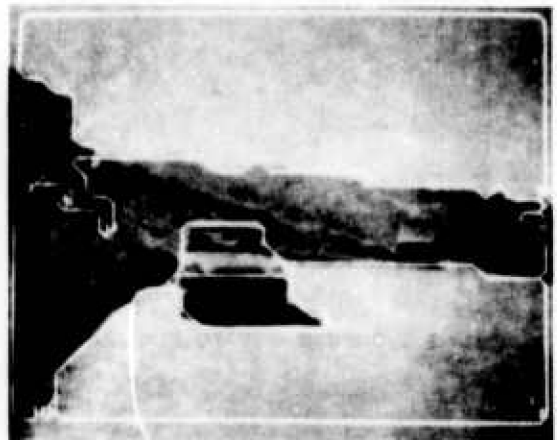
(C-1) Original.



(C-2) Output of non-semantic region grower.



(C-3) Output of the semantic region grower.



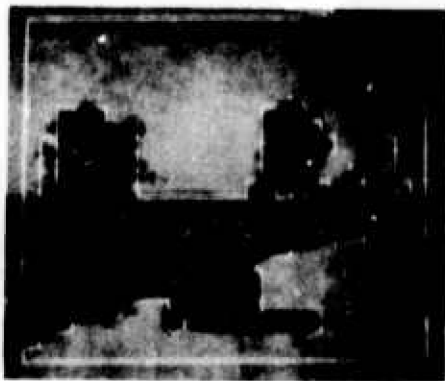
(C-4) Grouping regions by meaning with confidence 10.



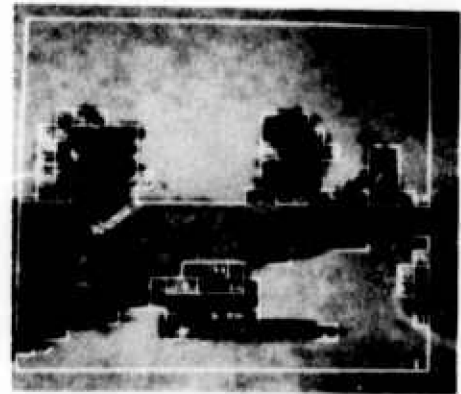
(D-1) Original



(D-2) Output of non-semantic region grower.



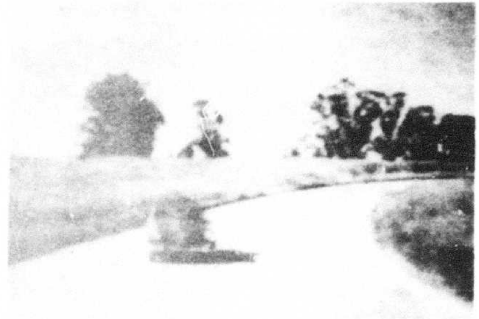
(D-3) Attempt to use non semantic region grower with more liberal stopping criterion.



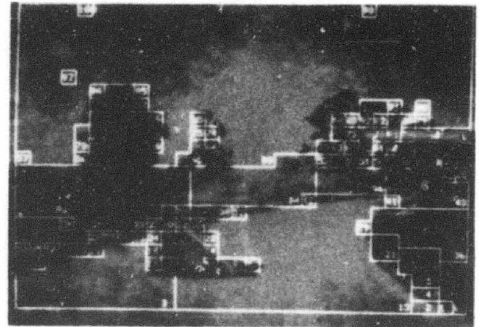
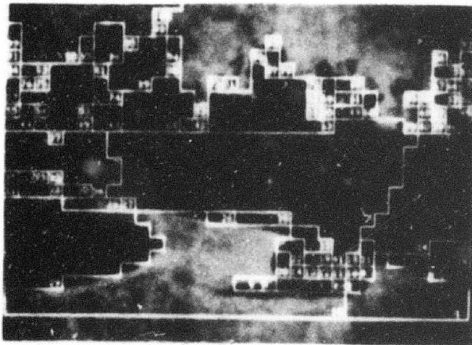
(D-4) Output of the semantic region grower.



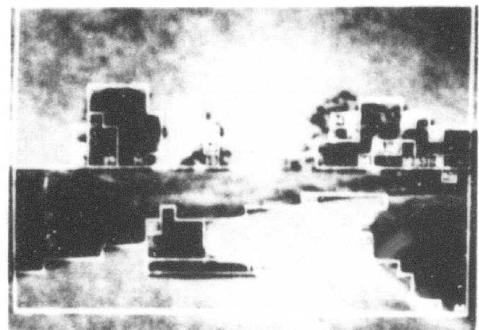
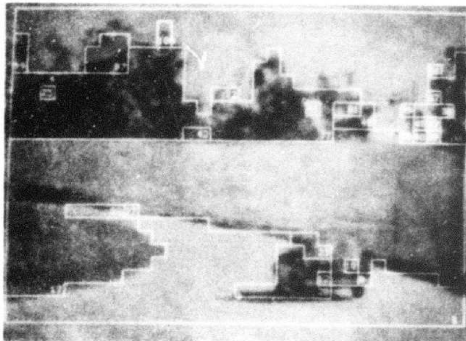
(D-5) Final output after grouping regions by their assigned meaning.



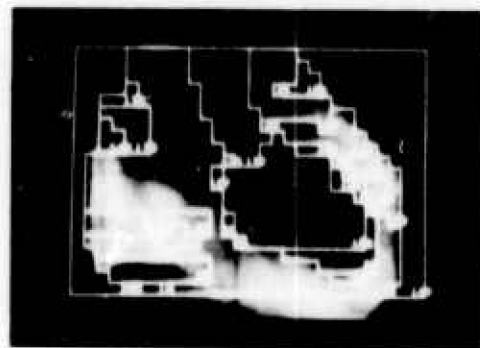
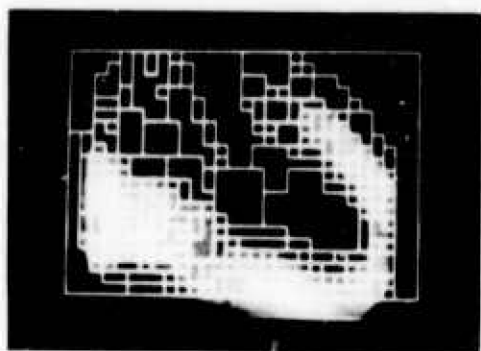
(E-1) Initial.



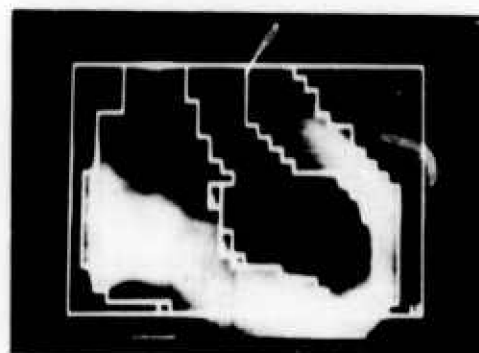
(E-2) Output of non-semantic region grower.



(E-3) Output of semantic region grower.



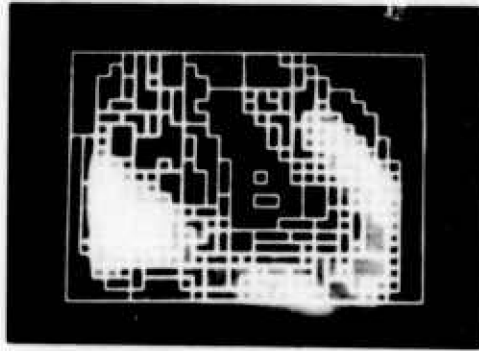
(F-1) Left ventricular angiogram. Output of the non-semantic boundary first region grower. The stopping criterion is to stop when the merger gets down to two hundred regions.



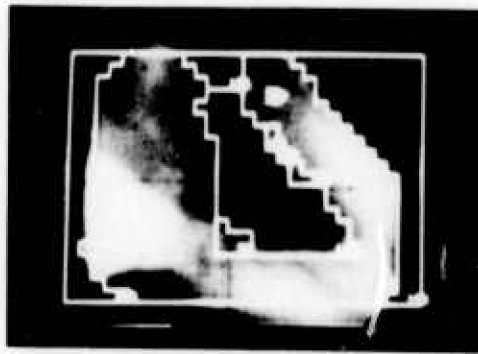
(F-2-3-4) Iterations of semantic region grower. Regions are grown by grouping all adjacent regions which are assigned the same meaning by the sequential assignment procedure, before the first assignment with low confidence level occurs. On each iteration the confidence threshold is lowered.



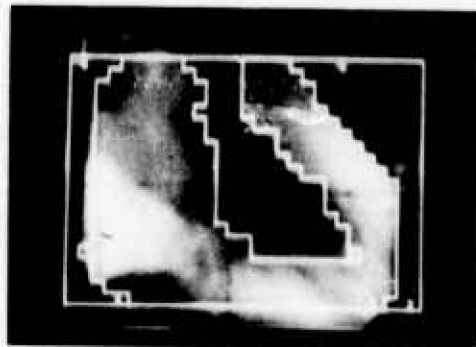
(F-5) Final output . The heart interior is the dark center, around it is the chest cavity and on the two sides there is the dark frame border.



(G-1) Output of the non-semantic weakest boundary first region grower.



(G-2) First iteration of semantic region grower. The region grower used here is grouping of adjacent regions that are assigned the same meaning, before the first assignment with low confidence was done.



(G-3) Another iteration like (G-2) where all assignments are considered valid.

BIBLIOGRAPHY

- [AGIN] Agin, G., "Description and representation of curved objects", Ph.D. Thesis, Stanford University, September 1972.
- [BAB] Barrow, H.G., Ambler, A.P. and Burstal, R.M. "Some techniques for recognition structures in pictures" - Department of Machine Intelligence and Perception, University of Edinburgh. 1972
- [BP] Barrow, H.G. and Popplestone R.J. (1971) "Relational description in picture processing" - Machine Intelligence 6 pp 377-396
- [BF] Brice, C.R. and Fennema, C.L. "Scene analysis using regions " J. ART. INT. 1 (1970) (pg 205-226)
- [CH] Chernoff, H. and Moses, L. - "Elementary decision theory" Wiley New-York, 1959
- [COV] Cover, T.M. and Hart, P.E "Nearest neighbor pattern classification " IEEE Trans. Information Theory, vol. IT-13, pp. 21-27, January 1967

- [DEG] De-Groot, M.H. "Optimal statistical decisions", McGraw-Hill
N.Y., 1970
- [DH] Duda, R.D. and Hart, P.E. "Pattern classification and scene
analysis" 1973 Wiley
- [FAL] Falk, G. "Computer interpretation of imperfect line data as
a three dimensional scene", Stanford U. AIM-139, 1970
- [FER] Ferguson, T.S. "Mathematical statistics a decision analysis
approach" 1967 Academic-Press
- [FEL] Feldman, F.A. "A way of looking at things" Department of
Mathematics, The Hebrew University, Jerusalem, January 1971
- [FU] Fu, K.S. "Sequential methods in pattern recognition and
machine learning", Academic, New-York, 1968
- [GR] Griffith, A.K. "Mathematical models for automatic line
detection", JACM 1973, pp. 62-80
- [GUZ] Guzman, A. "Computer recognition of visual objects in a
three-dimensional scene", M.I.T. 1969

- [HE] Harlow, C.A. and Eisenbeis, S.A. "The analysis of radiographic images ", technical report, department of E.E, U. of Missouri-Columbia
- [HUEC] Hueckel, M.H, "An Operator Which Locate Edges in Digitize Pictures", Stanford AIM-105, October 1969
- [KEL] Kelly, M.D. "Visual identification of people by computer", Stanford AIM-30, 1970
- [MON] Montanari, U. "On the Optimal Detection of Curves in Noisy Pictures". CACM, May 1971, 335-345.
- [NIL] Nilsson, N.J. "Problem Solving Methods in Artificial Intelligence", McGraw-Hill, 1971
- [PER] Preparata, F.P. and Ray, S.R. "An Approach to Artificial Nonsymbolic Cognition" information science 4, pp. 65-86, 1972
- [PT] Pingle, K.K., and Tenenbaum, J.M., "An accomodation edge follower", IJCAI-2.
- [QU] Quam, L.H. "Computer comparison of pictures" Stanford AIM-44

- [RAI] Raiffa, H. " Decision analysis - introductory lectures on choice under uncertainty", Addison Wesley, July 1970
- [RBJ] Bajcsy, R., "Computer identification of textured scenes", Stanford AIM 180 CS 72-321.
- [ROS] Rosenfeld A. "Picture processing by computer", Academic Press New-York, 1969
- [ROS1] Rosenfeld A., "Picture processing by computer", Computing Survey 1, Sep. 1969, 147-176
- [ROS2] Rosenfeld A., "Progress in picture processing: 1969-71", University of Maryland C.S TR-176, Jan. 1972.
- [ROS3] Rosenfeld A., "Picture processing: 1972", University of Maryland C.S TR-217, Jan. 1973.
- [SAIL] Van-Lehn, K. "SAIL user manual", Stanford AIM-204, C.S 373 , 1973
- [SAM] Samuel, A.L. "Some studies in machine learning using the game of checkers. II-recent progress" IBM Jour. of Res. and Dev., 11, 1967, pp. 601-617

BIBLIOGRAPHY

8.6

- [SR] Slagle, J.R. and Lee, R.C. "Application of Game Tree Searching Techniques to Sequential Pattern Recognition" CACM Feb 71, pp. 103-110
- [THO] Thomas, A. "Color perception" Internal report, Stanford U. 1973
- [TS] Thosar, R.B. and Samuel, A.L. "Some preliminary experiments in speech recognition using signature table learning" Stanford AIM- 198, 1973.
- [VIC] Vicens, P. "Aspects of speech recognition by computer " CS-27 Stanford AIM-85, 1969.
- [WAL] Waltz, D.L. "Generating Semantic Description From Drawing of Scenes with Shadows", M.I.T AI TR-271, 1972.