Final Report

(NASA-CR-146667) RESEARCH IN INTERACTIVE SCENE ANALYSIS Final Report (Stanford Research Inst.) 77 p HC $5.00 CSCL 09B

N76-20834

Unclas

G3/60 15220

RESEARCH IN INTERACTIVE SCENE ANALYSIS

By: J. M. TENENBAUM, H. G. BARROW, S. A. WEYL

Prepared for:

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
HEADQUARTERS
ROOM 607
WASHINGTON, D.C. 20546
ATTENTION: CHARLES PONTIOUS

Contract NASW-2086

SRI Project 8721

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

ABSTRACT

SRI, with NASA support, has been developing cooperative (man-machine) scene analysis techniques whereby humans can provide a computer with guidance when completely automated processing is infeasible. An interactive approach promises significant near-term payoffs in analyzing various types of high-volume satellite imagery, as well as vehicle-based imagery used in robot planetary exploration. This report summarizes the work accomplished over the two-year duration of the project and describes in detail three major accomplishments not previously reported:

- The interactive design of texture classifiers.
- A new approach for integrating the segmentation and interpretation phases of scene analysis.
- The application of interactive scene analysis techniques to cartography.

CONTENTS

ILLUSTRATIONS

TABLES

I INTRODUCTION

SRI, with NASA support, has been developing cooperative (man-machine) scene analysis techniques whereby humans can provide a computer with guidance when completely automated processing is infeasible. An interactive approach promises significant near-term payoffs in analyzing various types of high-volume satellite imagery, as well as vehicle-based imagery used in robot planetary exploration. This report summarizes the work accomplished over the two-year duration of the project and describes in detail the major accomplishments not previously reported.

II SUMMARY OF WORK DURING 1973-1974

During this period, we developed and implemented a set of scene analysis programs known collectively as ISIS (Interactive Scene Interpretation System). These programs are loosely integrated by compatible data structures and a common top-level command-driven executive. ISIS currently consists of the key components described below:

(1) ISIS Core [1]*--An extensible library of compatible INTERLISP and Fortran subroutines for picture processing and graphical interaction. These subroutines allow interactive users to observe how graphically designated parts of the scene are perceived by the system's descriptive and relational primitives. This information can then be used in conjunction with available sampling and region growing subroutines to empirically formulate and test automated strategies for distinguishing objects in particular pictorial domains.

(2) Object Finding Subsystem [2]--A program that automatically develops strategies for finding specified objects in a given class of scenes. Objects are designated to the system graphically by outlining pictorial examples. First, the system formulates a description of the object, based on characteristic features that distinguish it from objects previously designated. Then, it develops an efficient strategy based on cost-effectiveness models of the available ISIS modules.

(3) Segmentation Subsystem [3]--A program that uses semantic interpretations solicited interactively from a human collaborator to partition complex scenes into regions that correspond to meaningful objects. The system operates by requesting an interpretation whenever an unidentified region exceeds a threshold size and then by refusing to merge regions that carry different labels.

*The references are listed at the end of this report.

(4) Region Interpretation Subsystem [4]--A program that determines the best joint interpretation for regions in a partitioned scene. Each region is first assigned a set of possible interpretations that are consistent with its local attributes. A deductive mechanism then systematically eliminates improbable interpretations that violate global semantic constraints. For example "door" would be eliminated as a possible interpretation of all regions above a region previously deduced to be "wall."

The object finding, segmentation, and region interpretation subsystems were written to provide ISIS users with packaged paradigms that could be used as high-level components in their own scene analysis programs. Several specialized interactive systems were fabricated using these subsystems: natably Garvey's program for finding objects in room scenes [2], Weyl's program for cooperative (man-machine) partioning of natural scenes [3], and Tenenbaum's program for interpreting a manually partitioned room scene [4].

III SUMMARY OF WORK DURING 1974-1975

During this recently completed period, a number of new core facilities were implemented including a relational data base and a capability for windowing the image to obtain maximum resolution over a selected area of interest. These facilities were used in experiments on the interactive design of texture classifiers for distinguishing textures in a limited domain of scenes. The segmentation and interpretation subsystems were integrated into an automatic scene analysis system distinguished by its ability to capitalize on both general semantic knowledge about the scene domain and direct guidance from a human user. Finally, interactive scene analysis techniques were successfully applied to the problem of extracting cartographic features from aerial photographs. The approach used appears applicable to a variety of other tasks requiring coordinate digitization of graphical data.

The remainder of the report consists of self-contained chapters describing in detail the various aspects of our recent work.

IV INTERACTIVE DESIGN OF TEXTURE CLASSIFIERS

A. Introduction

Texture is an essential feature in the segmentation and interpretation of natural scenes. However, unlike hue and brightness, it is not a monolithic attribute, easily characterized by a single number. Investigators have thus been forced to use a wide variety of features to classify or distinguish particular textures in particular classes of imagery. Attempts to formalize criteria for selecting textural features have not been overly successful. For these reasons we decided to investigate ways in which the interactive facilities of ISIS could be used to determine empirically enough features to distinguish the prominent textures appearing in a limited scene domain.

B. Method of Approach

Representative images from the selected domain are partitioned exhaustively into small rectangular subimages. Manual interpretation is made of the texture types appearing in each subimage. Each subimage is then subjected to a battery of programs that extract texture-related features (see Section C). The results of manual interpretation and feature extraction are stored in a relational file that provides access to the values of texture features associated with a texture interpretation, the texture interpretation(s) associated with a set of texture feature values, the subimages containing a given texture interpretation, and the subimages associated with each original image. Using this data base, the experimenter designs ad hoc functions that test whether a particular texture interpretation is present in a subimage, based on texture features computed over that subimage. Typically, a texture function is first hypothesized on a basis of feature values obtained in subimages containing that texture. The postulated function is then tested automatically on the complete set of interpreted subimages in the data base. If necessary, the design process is iterated by modifying the function to incorporate texture features of misclassified subimages. Implementational details of the relational data base system can be found in Reference [5].

C. Texture Features

Textures can be characterized on an ad hoc basis at several levels of detail. For example, a textured region may be characterized at a microlevel by statistical distributions of the brightness, hue, and saturation of individual picture elements. Microtextures may also be specified by nonstatistical functions on the attributes of picture elements. In one particular scene, regions of sky and lake both contained samples with virtually identical blue hues. However, in the lake, the blue samples were liberally interspersed with distinctive green samples. Thus, in this domain, the texture "lake" might be characterized as a set of picture elements within a prescribed proximity to a distinctive green sample (the hue distribution of these proximate points being unimportant). At the macrotexture level, a region could be described in terms of distinguishing attributes of component regions, as when describing grass as a region containing green, yellow, and brown blobs. A particularly simple macrotexture descriptor is the number or density of smaller regions contained in a standardized window of a subimage. For instance, bushes may appear as a large number of small green regions, while grass, sky, and trees are represented by a few large regions. Other macrotexture features include distributions of the shape, spatial arrangement, and microtexture of the elementary component regions.

A library of programs for extracting both micro- and macro-texture features was written for use in developing texture discrimination functions. The microtexture features consist of approximately 30 statistical properties computed over all the picture elements in a subimage. These features, listed in Table 1, include the mean and distribution of brightness, saturation, and hue over the subimage, as well as measures of the homogeneity of these attributes. The homogeneity of an attribute was estimated by comparing the range of values observed over the whole subimage (excluding the upper and lower 10 percent extrema) with ranges observed over smaller subwindows of the subimage. Two sets of subwindows were used, partitioning the subimage into 4 x 4 and 10 x 10 cells respectively. Homogeneity served both as an intrinsic texture feature and as an indication that perhaps two or more textures were present within a subimage. This latter case might be suspected if the range of variation computed over the whole subimage was large compared with the ranges computed over more localized portions.

Macrotexture features were based on the attributes of regions obtained by subjecting the subimage to a crude segmentation procedure. The procedure used divided the subimage into regions consisting of adjacent picture elements of identical brightness (based on a few significant bits) in all three filter bands. The number of bits was manually chosen to provide a "good" distribution of region sizes: too many bits produces a lot of small regions and no large ones; too few bits produces a few, very large regions. (The process of selecting a suitable number of bits could, of course, be automated, perhaps by basing the number of bits on the global mean and variance values of brightness and hue over the subimage.) Seventeen properties were computed for each significant region, whose size exceeded 6 pixels (picture elements). These are listed in Table 2. The total number of significant regions over the subwindows

Table 1

SUBWINDOW PROPERTIES

1.	Number of significant regions (size >6 pixels)
2-4.	Brightness homogeneity a. Computed over whole subwindow (1) b. Computed over a sixteenth of the subwindow (4) c. Computed over a hundredth of the subwindow (10)
5-7.	Hue homogeneity (1, 4, 10)
8-10.	Saturation homogeneity (1, 4, 10)
11-13.	Average brightness, hue, and saturation (computed over all pixels in subwindow)
14-18.	Crude brightness distribution (5 level histogram)
19-23.	Crude hue distribution
24-28.	Crude saturation distribution
29-31.	Variance of brightness, hue, saturation
32-40	Crude distributions of average brightness, hue, and saturation for regions <6 pixels (3 level histogram for each attribute)
41-45.	Distribution of region sizes in partitioned subwindow (5 level histogram)

together with crude distributions of brightness, hue, and saturation for the insignificant regions were included in Table 1 as additional micro-texture attributes of the subimage.

D. Designing Texture Classifiers

A case study on the interactive design of texture functions was performed in a domain of 12 forest scenes from the Point Reyes National Seashore. Some typical scenes are shown in Figure 1. These were digitized through red, green, and blue color filters at 240 x 340 spatial

Table 2

REGION PROPERTIES
(Computed for each region in partitioned Subwindow)

Brightness, Color

1. Average Brightness, $\overline{b}_r = \sum_{i=1}^{N} \frac{br_i}{N}$, N = # of Pixels in region.

2. Brightness standard deviation, $\sigma_r = \sqrt{\frac{\sum_i |\overline{br} - br_i|^2}{N}}$.

3-5. Average brightness through Red, Green, Blue Filters:

$$\overline{R}_r = \sum_{i=1}^{N} \frac{r_i}{N} , \quad \overline{G}_r = \sum_{i=1}^{N} \frac{g_i}{N} , \quad \overline{B}_r = \sum_{i=1}^{N} \frac{b_i}{N} .$$

6-7. Average Red and Green brightness, normalized by total brightness,

$$\overline{r}_r = \frac{\sum_{i=1}^{N} \frac{r_i}{r_i+g_i+b_i}}{N} \qquad \overline{g}_r = \frac{\sum_{i=1}^{N} \frac{g_i}{r_i+g_i+b_i}}{N} .$$

8-9. Average Hue, Saturation based on $\overline{r}$, $\overline{g}$, $\overline{b}$.

(Computed using formulae described in Appendix A of [1].)

Shape

10. Area = N (number of Pixels in region).

11. Perimeter - number of elementary vectors surrounding region in region data structure.

12. Compactness = $\frac{(\text{Perimeter})^2}{\text{Area}}$.

13. Eccentricity = $\sqrt{\frac{(U_{20}-U_{02})^2 + 4U_{11}^2}{N^2}}$

where $U_{20} = \sum_{i=1}^{N} (\overline{x} - x_i)^2$

$U_{02} = \sum_{i=1}^{N} (\overline{y} - y_i)^2$.

ORIGINAL PAGE IS OF POOR QUALITY

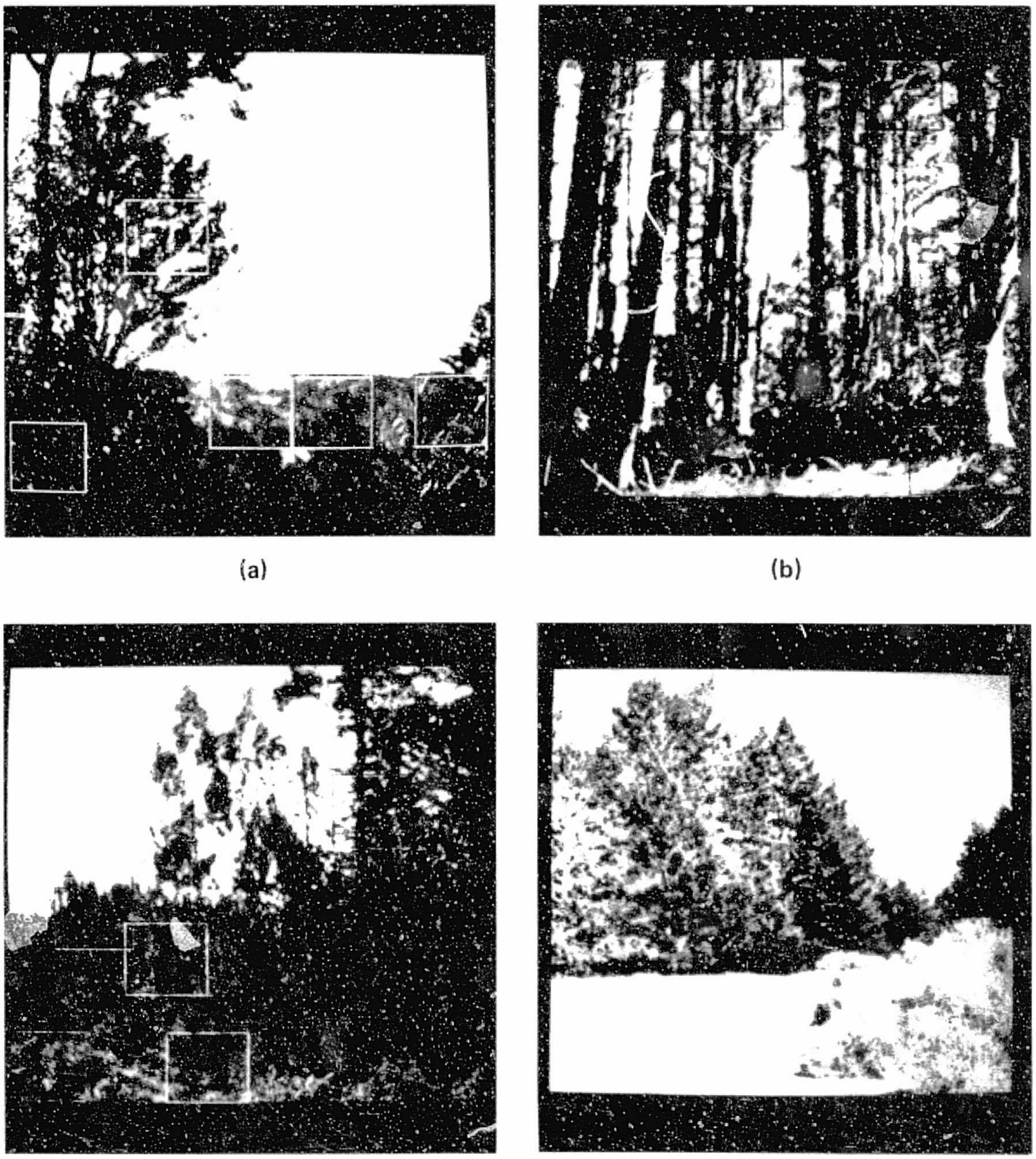
14. Angle of Major Axis, $\theta = \frac{1}{2} \tan^{-1}\left(\frac{2U_{11}}{U_{20}-U_{02}}\right) + \frac{n\pi}{2}$

where n is selected so that $\frac{-\pi}{2} \leq \theta \leq \frac{\pi}{2}$.

15. X Width, ΔX = XMAX - XMIN (of bounding rectangle).

16. Y Width, ΔY = YMAX - YMIN.

17. Fractional Fill = $\frac{\text{Area}}{\Delta X \cdot \Delta Y} = \frac{\text{Area of region}}{\text{Area of bounding rectangle}}$.



(a) (b) (c) (d)

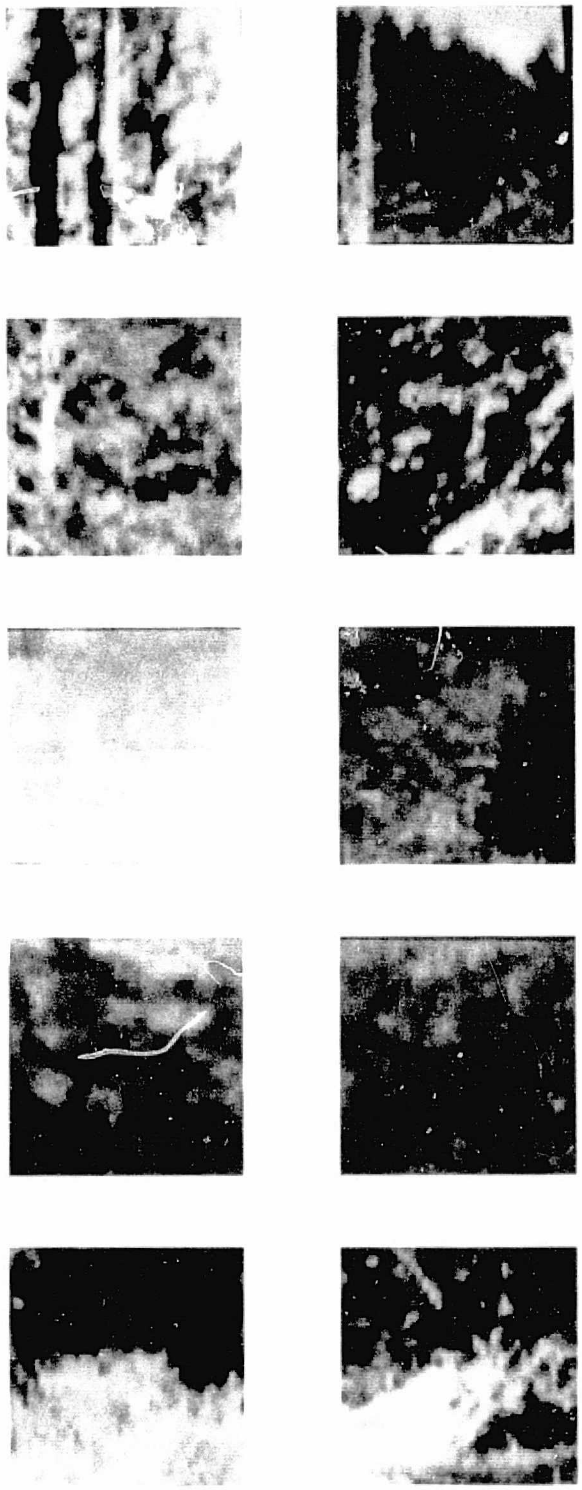
SA-4683-19

FIGURE 1 FOREST SCENES FROM POINT REYES NATIONAL SEASHORE resolution and 8 bits of intensity. From each picture, several 40×40 subimages were selected manually for empirically developing a texture classifier. Each subimage was partitioned into regions of identical color (based on several significant bits/separation) and then exhaustively characterized according to the subimage and region features listed in Tables 1 and 2. The resulting feature values were stored respectively under subimage and region files in the data base. The squares superimposed on the images in Figure 1 outline subimages selected for texture analysis, some of which are shown isolated from the rest of their pictorial context in Figure 2. The subimages in Figures 2a and 2b all come from trees, showing the diverse appearances that textured entities in real-world scenes can assume.

As mentioned earlier, the quality of the partition on which texture analysis is based depends critically on the number of bits from each separation that are used for determining homogeneity. Using too many bits will result in too much detail, reducing the amount of meaningful region shape and orientation information, while using too few bits will cause blurring and elimination of critical details. Figure 3a shows a partition based on too few bits, while Figure 3b shows a good partition. Notice that in the good partition all important details have been captured.

Five texture categories were identified in the current study, "grass," "shrubs," "trees," "sky," and "background." A list of texture interpretations from these categories was manually assigned to each subimage recorded in the data base.

The design of texture functions for these five textures began by observing the distinguishing macrotexture features of subimages containing each texture type. Trees were observed to have a significant number of bright reddish highlight regions in the crown portion, interspaced



(a) TREES

(b) MORE TREES

(c) GRASS

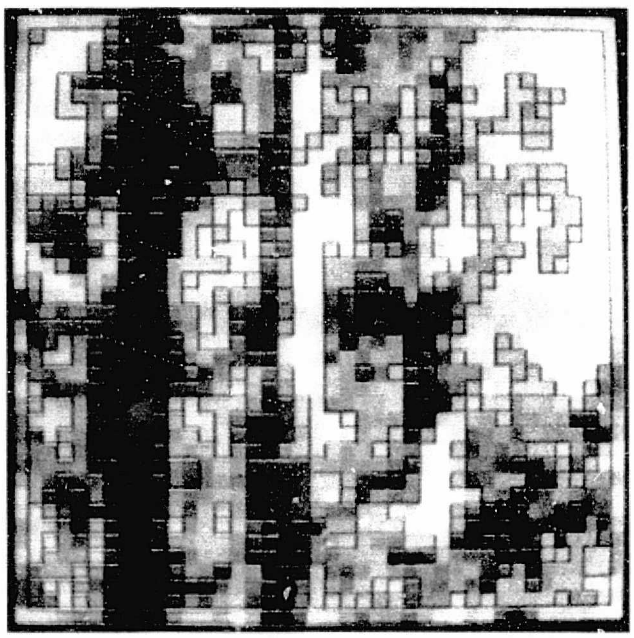
(d) BACKGROUND

(e) GRASS AND SHRUBS

SA-4683-20

FIGURE 2 SUBIMAGES REPRESENTATIVE OF TEXTURE CATEGORIES

(a) OVERLY COARSE PARTITION
(TOO FEW BITS)



(b) GOOD PARTITION

SA-4683-21

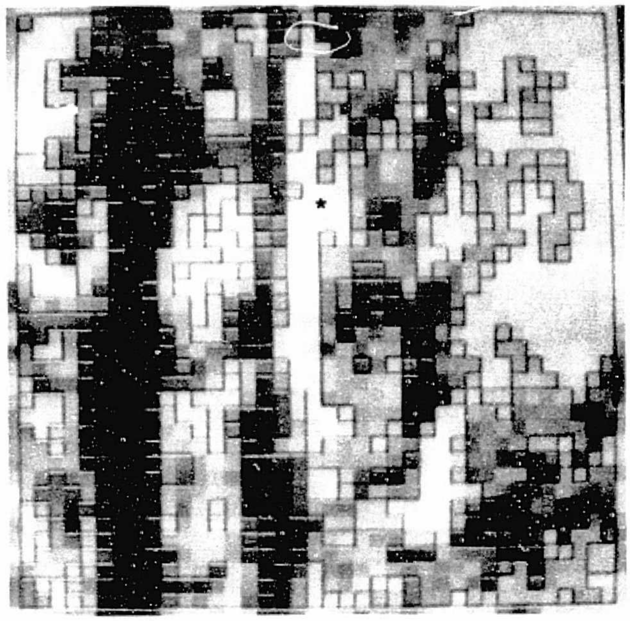
FIGURE 3 PARTITIONED SUBIMAGE CONTAINING TREE (TRUNK) TEXTURE with blue blob-shaped regions of sky. Treebark tended to appear as vertically elongated brown regions. Shrubs, by contrast, were distinguished by large light-red blobs and few small blobs, while grass was a fairly solid brown-green. Many of these distinctions are evident in the sample subimages of Figure 2, despite the absence of color information present in the original RAMTEK displays.

The next step in the design process involved formulating functions of region and subimage properties that could represent analytically the distinctions in texture described above. This crucial design phase is entirely empirical and its success depends largely on the adequacy of the available feature extraction operators.

The typical sequence used in designing texture classifiers can be illustrated with an example. Suppose the user is interested in identifying subwindows containing the class of treetrunk textures exemplified by Figure 3. After displaying a representative subwindow, the user can interrogate the values of selected window attributes and the attributes of distinguishing regions, which he designates with a cursor.

For example in Figure 4a, the user selects a vertical section of the treetrunk in the middle of the subimage, which the computer identifies with a bright star. The properties of this region, shown in Table 3, are then printed out on his terminal.

The user then decides that sections of treetrunk are distinguished by their vertical orientation and horizontal narrowness. He filters the regions, using the predicate for treebark shown in Table 4. The regions selected by this predicate are indicated with stars in Figure 4b. Any subwindows containing a sufficient number of such regions are classified as containing treetrunks. Classifiers for the other categories of texture are also given in Table 4.



(a) REPRESENTATIVE TREEBARK REGION (SELECTED BY USER)

(b) REGIONS MATCHING TREEBARK PREDICATE (SELECTED BY COMPUTER)

NOTE: Selected regions are designated by *.

SA-4683-22

FIGURE 4 INTERACTIVE DESIGN OF TEXTURE PREDICATES

Table 3

PROPERTIES OF TREETRUNK REGION

1.	Average brightness	=	78.2
2.	Brightness s.d.	=	11.7
3.	Average r	=	85.7
4.	Average g	=	47.8
5.	Average b	=	101.1
6.	Average norm r	=	0.36
7.	Average norm g	=	0.20
8.	Area	=	102.0 pixels
9.	Perimeter	=	102.0 elementary vectors
10.	P^2/A	=	102.0
11.	Trace	=	0.011
12.	Eccentricity	=	0.01
13.	Angle of major axis	=	90°
14.	Fractional fill	=	0.54
15.	x width	=	5.0 pixels
16.	y width	=	38.0
17.	Hue	=	283.6°
18.	Saturation	=	0.39

Table 4

TEXTURE CLASSIFIERS

Subwindow contains

1. Tree (trunks)

   if regions with:

   $.33 \leq$ average $r \leq 1.0$

and $80^{\circ} \leq$ angle of major axis (from horizontal) $\leq 100^{\circ}$

and $4 \leq$ width (pixels) ≤ 8

2. Tree (crown)

   if regions with:

   area < 4

and $230 \leq$ brightness ≤ 256

and average $r > 0.33$

(These regions correspond to those preceived as red highlights in Figure 2.)

3. Sky

   if regions with:

   $230 \leq$ brightness

and $r < 0.33$

4. Grass

   if # of regions in subwindow < 200 (partition based on 2 bits/color)

and not (sky)

5. Background

   Brightness variance - small (fuzzy greenish regions)

and average $g > 0.33$

E. Discussion

As of this writing, a comprehensive evaluation of these classifiers on the whole data base has not yet been performed. The actual results obtained by this particular set of ad hoc texture functions is, however, less important than the interactive methodology used in formulating them; ISIS was created to provide experimenters with the tools (such as displays, data base, operators) needed to rapidly formulate effective scene analysis strategies for limited domains of scenes. ISIS was first used to design strategies for finding objects based on their distinguishing features in a limited context. The same methodology has now been applied to the design of texture functions.

The most obvious way to improve the performance and generality of a texture classifier is to add additional texture features to the system's repertoire. Several interesting features were discussed, for example, in our 1974 annual report [6], based on the spatial dependency and Fourier power spectrum of pixel brightness. The performance of the macrotexture features used in the current exercise could be improved by using a better procedure to partition the subimages into regions. Clearly, these regions should correspond closely to interpretable pictorial entities (e.g., to leaves in a tree texture). In the extreme, a detailed scene analysis procedure could be performed within each subimage to obtain a good segmentation on which to base texture classification.

The interactive aids provided by the system could also be augmented with clustering procedures for suggesting good texture features to the user. Our immediate interest, however, is in using textural attributes in the semantically guided segmentation system described in the next chapter.

V EXPERIMENTS IN INTERPRETATION-GUIDED SEGMENTATION

A. Introduction

A truly flexible interactive scene analysis system should be based on an underlying automatic system with the versatility for effectively using manually supplied guidance. Such a system would be capable of functioning, albeit at reduced effectiveness, without any guidance, and its effectiveness would increase steadily with increasing quantity and specificity of user interaction.

The system we propose is based on a generalization of Weyl's semantic segmentation program [3]. The central idea in that program was the use of semantic region interpretations to guide segmentation. The program in its existing form interactively solicits explicit interpretations for large unidentified regions and then refuses to merge regions that carry different labels. The use of a size threshold is, of course, arbitrary; if interpretations could be assigned to every picture element (pixel), then segmentation would be reduced to the trivial process of collecting adjacent pixels with the same labels.

There are two difficulties in automating interpretation at the pixel level, namely the excessive volume of data, and the absence of global attributes (e.g., shape, texture, boundary relations). These attributes emerge only after a region structure has been imposed on the pixels, but without them, interpretation is usually ambiguous.

The integration of segmentation and interpretation is accomplished in our system by proceeding incrementally. Beginning at the pixel level, the system first performs the most complete interpretation possible in the current partition. Next, it performs the safest merge consistent with that interpretation and any prior knowledge about the domain. The process then iterates by revising the interpretation to fit the current partition and performing another merge (see Figure 5).

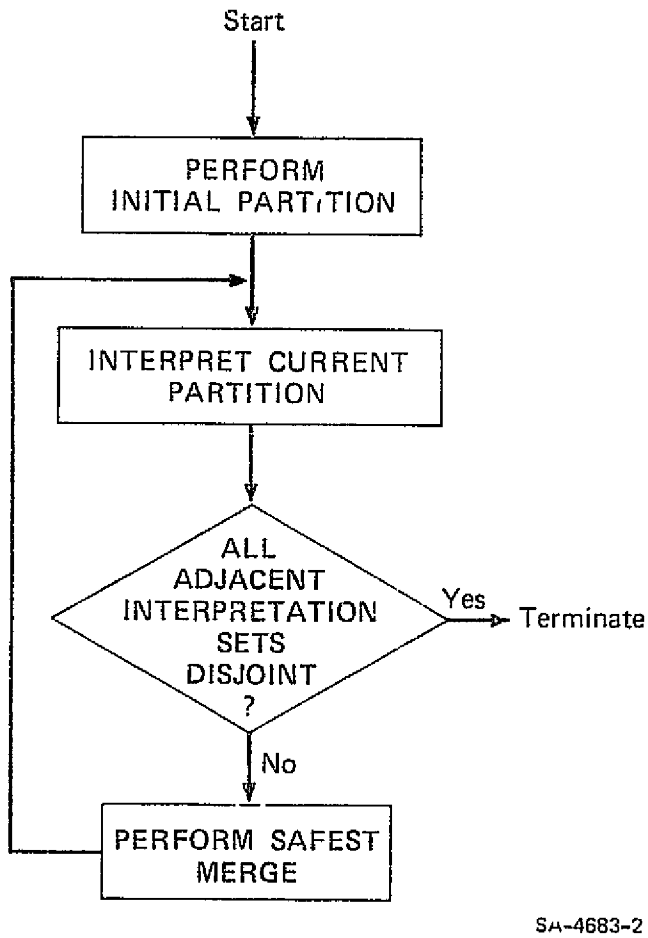


FIGURE 5 OVERVIEW OF INTERPRETATION GUIDED SEGMENTATION PARADIGM

The safety of a merge is evaluated by assessing the likelihood that two adjacent regions (or pixels) are fragments of the same object. Merges are thus never allowed between regions whose interpretations (or sets of possible interpretations) are known to be disjoint. The safest merge involves regions that have already been assigned the same unique interpretation. It is also safe to merge adjacent regions whose interpretations, while not yet unique, have been narrowed to the point where prior knowledge constrains both regions to take the same interpretation. For example, suppose that "pictures" were constrained to hang only on "walls," and thus could never appear adjacent to "doors" in an image. Two adjacent regions with "door" and "picture" as possible interpretations, could thus be safely merged since both regions must be interpreted either as parts of a "door" or as parts of a "picture." If there are no "safe merges," as defined by the above criteria, then the regions separated by the lowest contrast boundary are merged, provided, of course, their possible interpretations are not disjoint. When the possible interpretations of all adjacent regions in the current partition are disjoint, the analysis terminates.

After each merge, the resulting partition is reinterpreted. When regions merge, the resultant region initially inherits the possible interpretations shared by its parent regions. (These are obtained by intersecting the interpretation sets of the parent regions.) Some of these common interpretations may not be compatible with the expanded range of attribute values found in the enlarged region and can therefore be immediately ruled out. A small region, for example, will admit interpretation as either a small object or part of a large object, but a large region can correspond only to a large object.

Interpretations eliminated in the course of region merging may, in turn, allow semantically related interpretations to be dropped as possibilities of other regions. For example, if a newly merged region becomes too large to be a "chairseat," the possibility "chairback" can be dropped for the region above it. These secondary eliminations may themselves propagate additional eliminations extending throughout the image.

Initially, all pixels are assigned all possible interpretations. Hence, any adjacent pixels can be legally merged but no merge is guaranteed to be "safe." Without additional knowledge or interactive guidance, the system will thus function as a conventional region grower, merging regions in order of boundary contrast. Prior knowledge and user interaction act by constraining the possible interpretations of regions and thereby restrict the set of region interpretations with which those regions can be compatibly merged.

A prototype version of the above paradigm was implemented in Fortran, as an extension of a previously described region analysis program [3]. In this prototype version, every pixel was allowed up to 18 possible interpretations that were predefined for a given domain. In room scenes, for example, the interpretations that were defined included "door," "wall," "floor," and so forth. (The possible interpretations of a pixel were physically represented by bits in the left halfword of the image array element containing its brightness.) As an expedient, the initial level of interpretation occurred, not at the pixel level, but after an initial level of partitioning in which adjacent pixels with both identical brightness <u>and</u> identical sets of possible interpretations were grouped into regions.

The remainder of this chapter describes three sets of experiments with the above paradigm involving three distinct sources of knowledge. In these experiments, interpretations were constrained by user interaction, a geometric model, and prior knowledge about the spatial relationships of objects in a limited domain.

B. <u>Experiment I--Interactively Guided Segmentation</u>

1. <u>Introduction</u>

Users can influence the partitioning of particular images by directly assigning interpretations to specified regions. In Weyl's system, an interpretation was solicited from the user whenever merging produced a large uninterpreted region. This capability has been generalized so that users can now volunteer interpretations for regions or sets of regions throughout the analysis by pointing at or encircling them with a display cursor.

Intuitively, guidance received early in the analysis will be most beneficial in preventing erroneous merges. We felt that with relatively little effort, a user could crudely outline and label the major objects in a scene. These labeled outlines would provide most of the region interpretations that had to be solicited individually in Weyl's system and also serve as a good initial partition from which detailed boundaries could be rapidly grown. To test this contention, a program was written that allowed users to draw regions in a displayed image before initial partitioning, and to specify for each region a unique label, a set of possible labels, or a set of labels to be deleted. Users were instructed to rapidly partition and label the image so as to thwart anticipated merge errors. In particular, they were told to crudely inscribe and uniquely label areas of the image containing unobstructed views of large objects and to point at and label at least one pixel in each area of the image containing a sizable but isolated fragment of a major object, such as pieces of "sky" showing through a "tree." They could also attempt to contain spatially amorphous objects, such as "trees," by circumscribing them crudely and then deleting that object's interpretation from all pixels outside the circumscribed region.

2. Methodology

The output of this region-labeling phase was an annotated image array in which every pixel had an associated set of possible interpretations. All pixels contained within a region designated by the user were assigned the interpretation set associated with that region. All other pixels were assigned, as a default, the set of all possible interpretations. An initial partitioning of this array was performed in two steps.

First, adjacent pixels with unique, identical interpretations were grouped into regions, then all remaining adjacent pixels with both identical brightness and identical interpretations were grouped. Grouping uniquely interpreted regions independent of brightness reduced the total number of regions in the initial partition and made the resulting regions more representative of the underlying object structure.

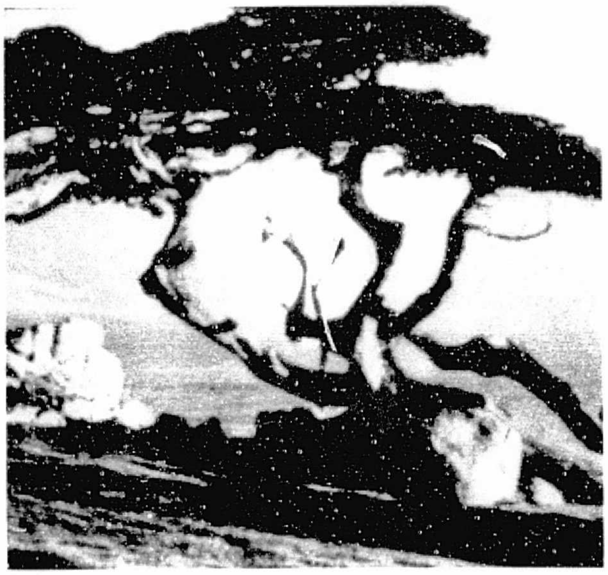
Following this initial partitioning, the merge/interpretation cycle commenced. In this experiment, the system had no general semantic knowledge and hence all merges had to be regarded as unsafe. As such, merging proceeded at each stage by deleting the lowest contrast boundary between adjacent regions with nondisjoint interpretation sets. Additional user interaction was not solicited during this subsequent analysis.

3. Results

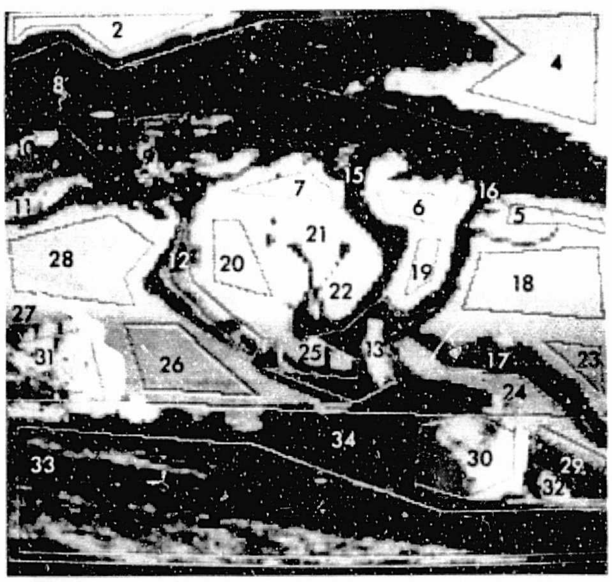
Some typical results are shown in Figures 6 and 7. Figure 6a is an improved digitization of the scene previously analyzed in Reference 3.* This scene contains a large number of isolated fragments of objects occluded by parts of the tree. This necessitated a rather detailed manual labeling stage, the results of which are shown in Figure 6b. The initial partition based on brightness and manual labeling (at 60 × 60 resolution) appears in Figure 6c. (This initial partition is far superior to that shown in Reference 3, which was based solely on brightness at 40 × 40 resolution.) The final partition and labeling appears in Figure 6d. The scene analyzed in Figure 7 contains little occlusion. Consequently, fewer manually inscribed regions were needed to adequately constrain the final partition.

---

*The current digitization was performed at USC on a Muir head drum at 8 bits of brightness resolution.



(a) DIGITIZED IMAGE
(8 BITS AT 256 x 256 RESOLUTION)



Interpretations	Regions
Sky	2,4,5,6,7,10*,11*
Mountain	18,19,20,21*,22*,28
Sea	23,24,25*,26,27*
Ground	33,34**
Rock	30,31,32*
Tree (Crown)	8
Tree (Bark)	12,13,15*,16*,17

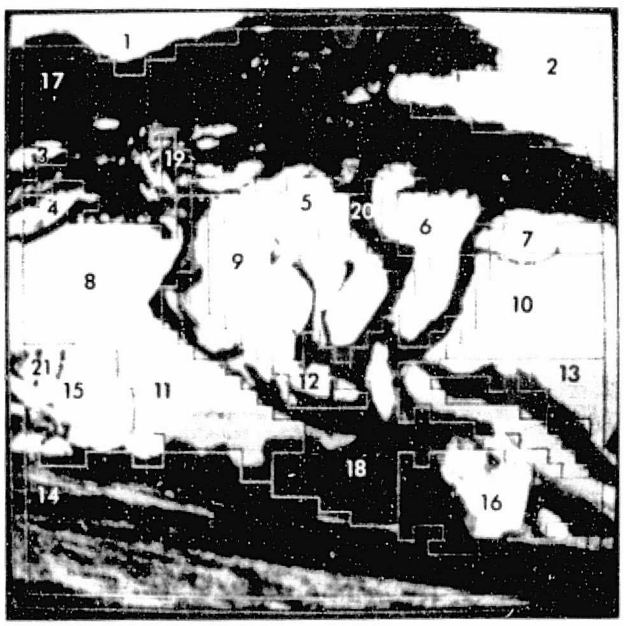
(b) CRUDE MANUAL PARTITION AND LABELING

*Single point region.
**Circumscribed boundary.
All other regions are inscribed boundaries.

SA-4683-15

FIGURE 6 INTERACTIVELY GUIDED SEGMENTATION OF MONTEREY CYPRUS SCENE

(c) INITIAL PARTITION (AT 60 × 60 RESOLUTION) BASED ON BRIGHTNESS AND MANUAL LABELING; CONTAINS 481 REGIONS



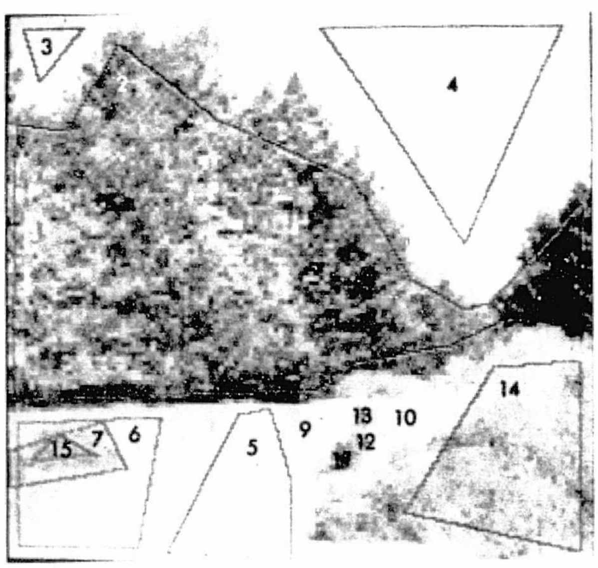
Interpretations	Regions
Sky	1-7
Mountain	8-10
Sea	11-13, 21
Ground	14
Rock	15-16
Tree (Crown)	17
Tree (Bark)	18-20

(d) FINAL PARTITION AND LABELING (21 REGIONS)

SA-4683-16

FIGURE 6 INTERACTIVELY GUIDED SEGMENTATION OF MONTEREY CYPRUS SCENE (Concluded)

(a) DIGITIZED IMAGE
(8 BITS AT 256 x 256 RESOLUTION)



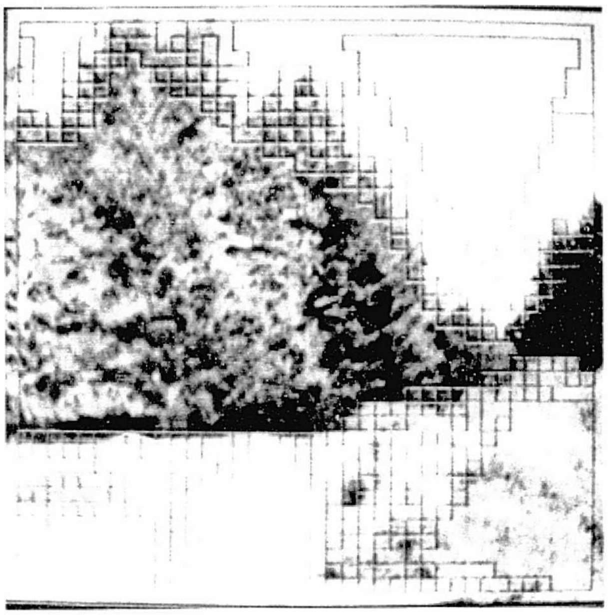
Interpretations	Regions
Sky	3,4
Tree	11*,12*,13*
Tree and Sky	2
Shrubs	14,15
Grass	6,7,9*,10*
Path	5

(b) CRUDE MANUAL PARTITION AND LABELING

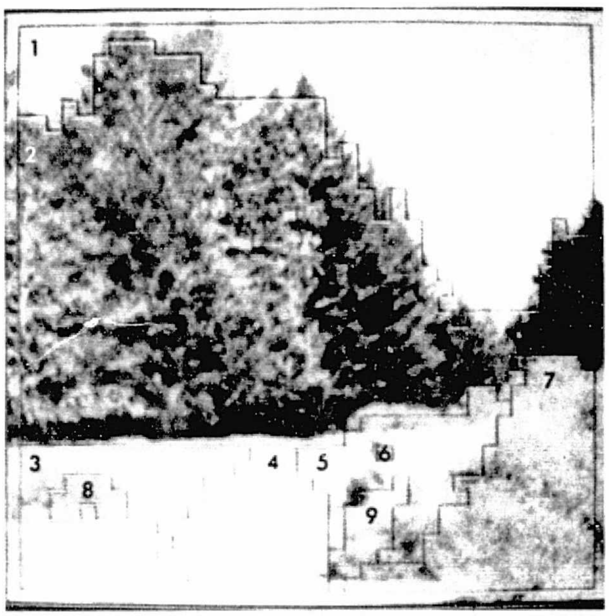
*Single point region.
All other regions are inscribed boundaries.

SA-4683-17

FIGURE 7 INTERACTIVELY GUIDED SEGMENTATION OF POINT REYES SCENE



(c) INITIAL PARTITION
(AT 60 x 60 RESOLUTION);
CONTAINS 286 REGIONS



Interpretations	Regions
Sky	1
Tree	6,9
Tree and Sky	2
Shrubs	7,8
Grass	3,5
Path	4

(d) FINAL PARTITION AND LABELS (9 REGIONS)

SA-4683-18

FIGURE 7 INTERACTIVELY GUIDED SEGMENTATION OF POINT REYES SCENE
(Concluded)

It is difficult to evaluate an experiment whose results are subject to the variability of human input. The results shown are, however, representative of the 10 experiments of this type that have been performed. The final partition in Figure 6 appears subjectively better than the result previously obtained in Reference 3, where interpretations were solicited during the analysis. This improvement is probably due, in large part, to the improved initial partition and the increased resolution.

4. Discussion

The above experiments confirmed that with a little human guidance, reasonable partitioning of complex scenes could be obtained. This interactive mode of partitioning could conceivably provide a practical way to process images that are too difficult to segment completely automatically and also too detailed or numerous to segment by hand (e.g., by tracing detailed boundaries on a digitizing table). We envisage a system that would use crude manual partitioning as a guide to extract detailed region boundaries, and then rely on additional interaction to correct the occasional errors (e.g., small sections of boundary could be traced in detail). We are currently studying the application of such techniques to cartography (see Chapter VI), and are considering additional applications in earth resource assessment, photo interpretation, and radiology.

C. Experiment II--Model Guided Segmentation

1. Introduction

An experiment was performed to demonstrate the feasibility of guiding segmentation with interpretations provided by a three-dimensional geometric model. Specifically, the objective was to segment an image into regions that correspond to the parts of an object articulated in the model. For this experiment, a color photograph of an air compressor was digitized to 32 levels at 60 X 60 resolution (Figure 8). This photograph was initially partitioned into regions composed of adjacent pixels with identical brightness, as shown in Figure 9. Because of the uniform coloring of the compressor, which is typical of mechanical equipment, a nonsemantic region-merging program proved very unsatisfactory. Figure 10, for example, shows the partition that results from successively merging together pairs of adjacent regions with lowest color contrast, until 200 regions remained. Though pointless, this process could obviously be continued until the entire scene was merged into one big region.

A structural model of this compressor was previously developed by Agin, for use in planning assembly and disassembly sequences [7]. The model, shown in Figure 11, contains polyhedral representations for the major components of the compressor, and associated metrical information. Given this polyhedral model and a simple projective camera model, a graphics program can display how the compressor in known position and orientation will appear from an arbitrary viewpoint. With the straightforward addition of a hidden surface algorithm, the display program can also determine which component of the compressor (e.g., tank, pump, motor) will actually be visible at each point in the image. This knowledge can be represented in the form of a visibility matrix, as shown in Figure 12.

2. Methodology

For our experiment, it was assumed that the relative location and orientation of the camera and compressor were known approximately. This uncertainty in relative position introduces a corresponding uncertainty in the prediction of which compressor component will be visible at a given point in the image. The latter uncertainty can be represented by a set of overlapping regions, each of which expresses the composite area of the image that could possibly be occupied by a given interpretation, for all compressor positions within the assumed range of uncertainty. Figure 13

FIGURE 8 DIGITIZED IMAGE OF COMPRESSOR (5 BITS AT 120 x 120 RESOLUTION)

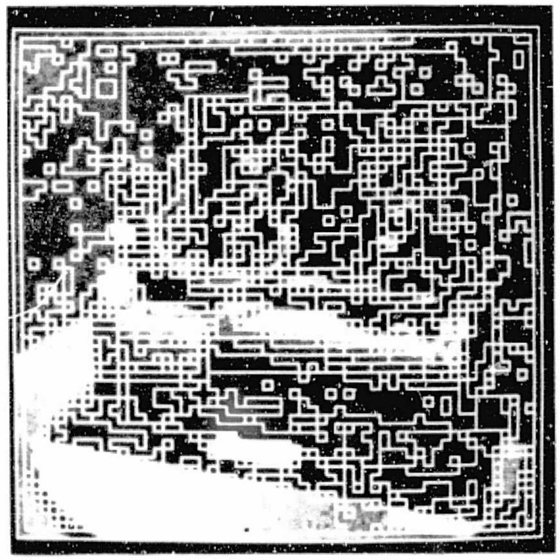


FIGURE 9 INITIAL PARTITION (AT 60 x 60 RESOLUTION); CONTAINS 931 REGIONS

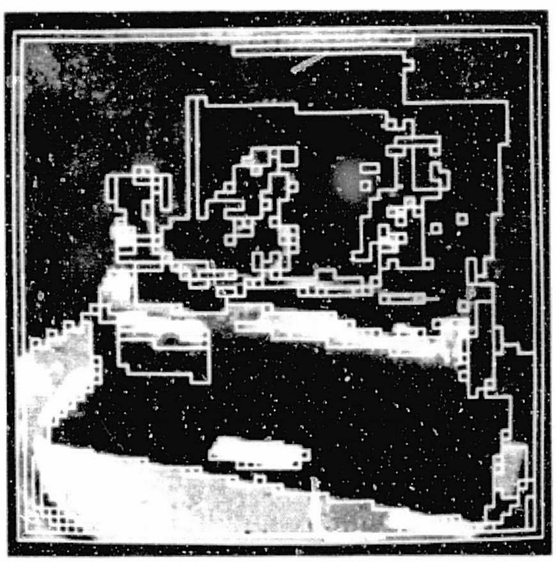


FIGURE 10 ERRORFUL UNGUIDED PARTITION (200 REGIONS)

SA-4683-12

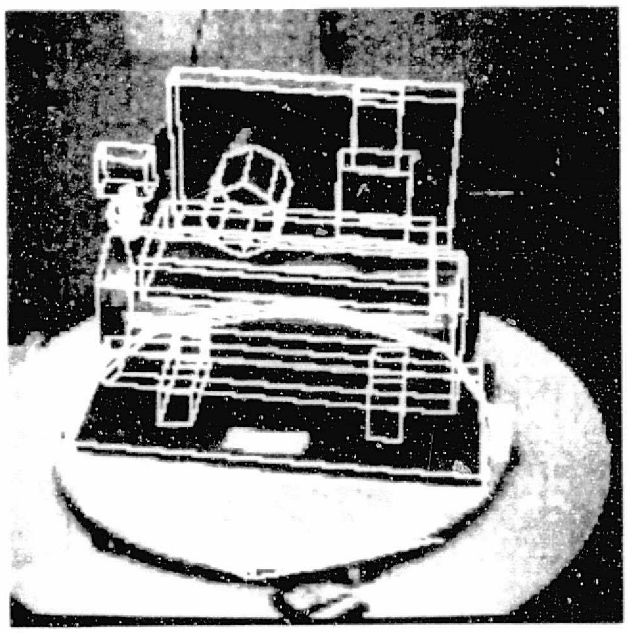
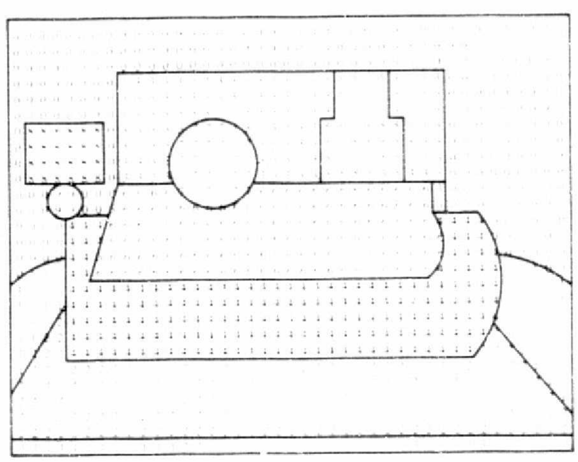


FIGURE 11 POLYHEDRAL (WIRE FRAME) MODEL SUPERIMPOSED ON T-V IMAGE OF COMPRESSOR



0 = Background
1 = Table
2 = Base
3 = Belt Housing
4 = Tank Cylinder
5 = Pump
6 = Tank Platform
7 = Motor
8 = Pressure Switch
9 = Pressure Guage

SA-4683-13

FIGURE 12 VISIBILITY MATRIX SHOWING PIXEL INTERPRETATIONS FOR COMPRESSOR IN KNOWN RELATIVE POSITION TO CAMERA shows the composite regions for the compressor parts distinguished in this experiment. (These regions were transcribed manually from a series of displays showing the compressor at various positions over the allowed range. The transcription process would, however, be straightforward to automate.)

The regions shown in Figure 13 were used to make initial interpretations of each pixel, in the same way that manually designated region interpretations were used in the previous experiment. Specifically, the bit representing the interpretation of each region was turned on for all pixels within that region and turned off for all those outside. An initial partition was then formed in which all adjacent pixels with identical brightnesses and interpretations were grouped into regions. Regions were then merged, as in the previous experiment, in order of weakest boundary contrast subject to the existence of at least one common interpretation. Resultant regions again acquired interpretation sets formed by intersecting the possible interpretations of both parent regions.

3. Results

The merging process terminated with the partition shown in Figure 14, in which all adjacent regions had disjoint interpretations. The result is by no means perfect, but does represent a considerable improvement over the attempt at unguided segmentation. The result could be further improved by using a more detailed model, by iterating the analysis to refine the position estimate of the compressor, and by using additional knowledge about compressors such as the visual appearance of parts.

4. Discussion

The use of structural models for guiding segmentation is well suited to industrial inspection tasks where the structure of a manufactured item is fixed and its position is known approximately. The resulting

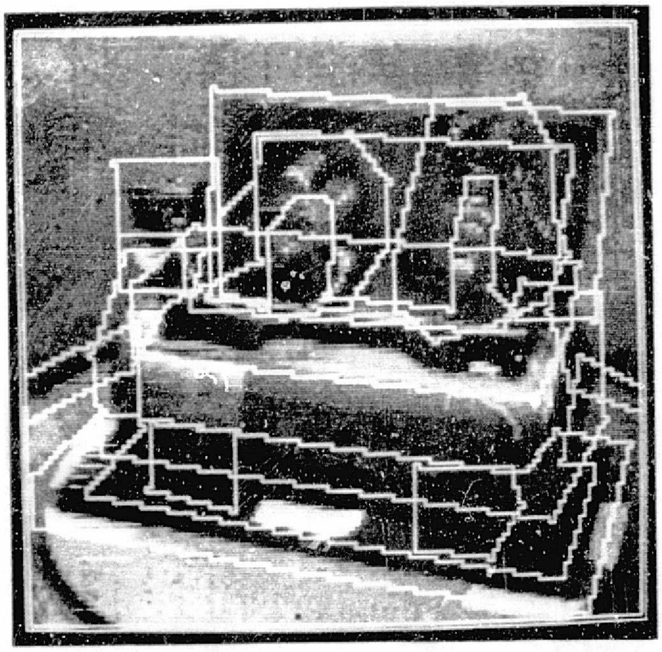
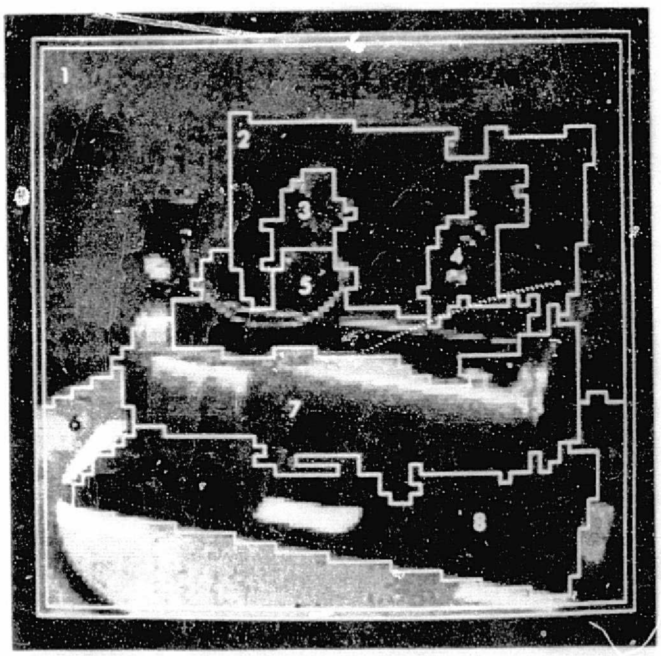


FIGURE 13 COMPOSITE REGIONS DELINEATING POSSIBLE AREAS OF IMAGE FOR EACH INTERPRETATION



Region	Interpretations
1	Background
2	Belt Housing
3	Motor
4	Pump
5	Tank Platform
6	Table
7	Tank Cylinder
8	Base

SA-4683-14

FIGURE 14 FINAL PARTITION AND LABELS AFTER MODEL GUIDED MERGING analysis can be used to locate the item's position exactly and also to locate the boundaries of parts of the item as a prelude to inspection. This inspection scenario is representative of a variety of tasks involving knowledge about the approximate image location of objects in a relatively static scene. Thus, maps can be used in a similar fashion as structural models to guide the interpretation of aerial photographs. Similarly, anatomical maps can guide the interpretation of medical imagery such as x-rays and thermograms. A previous analysis of a scene is yet another source of knowledge about object location that can be used in tasks such as change detection, motion tracking (i.e., analyzing a series of scenes taken from slightly different viewpoints), and the analysis of a sequence of movie frames. Note that when the location parameters of an object model are known exactly but the position of the camera is uncertain, then a model-driven analysis can be used to calibrate parameters of the camera model, or alternatively, the location of a robot vehicle that may be carrying the camera.

D. Experiment III--Constraint Guided Segmentation

In both previous experiments, segmentation was guided by interpretations that were specified for particular regions in a particular scene. Region interpretations can also be deduced using constraints that apply to generic interpretations over all images in a given domain. These constraints specify conditions on the attributes and spatial relationships of regions that must be satisfied for given region interpretations to be valid. For example, constraints might dictate that the interpretation "sky" can apply only to large, blue regions that are not below another region previously labeled "horizon."

1. <u>Deducing Region Interpretations with Relational Constraints</u>

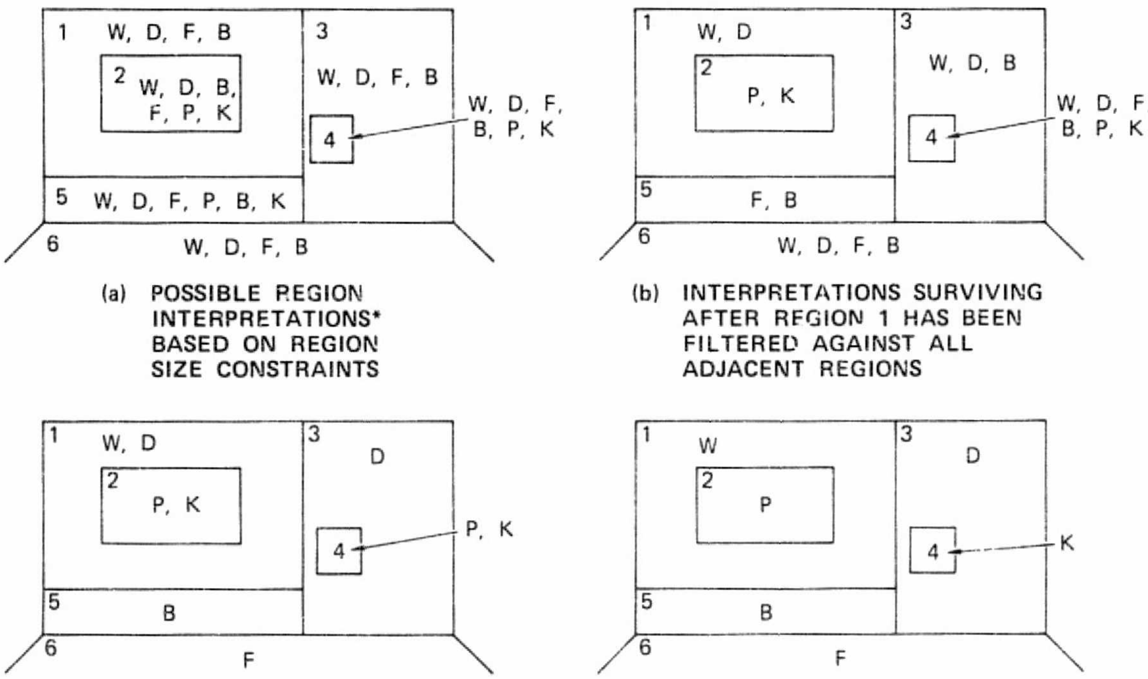
The process of deducing region interpretations using constraints generalizes Waltz's filtering algorithm [8]. Waltz analyzed line drawings by initially assigning all locally possible interpretations to each vertex and then eliminating any vertex interpretation that was inconsistent with all possible interpretations of a neighboring vertex along a common edge. Eliminating a possible vertex interpretation could result in the elimination of additional interpretations from adjacent vertices. This elimination process would often propagate until each vertex was left with a unique interpretation. A similar paradigm can be applied to region analysis by initially assigning all locally possible interpretations to each region and then eliminating interpretations inconsistent with those assigned to neighboring regions sharing a common boundary.

The locally possible interpretations of a region are governed by constraints that specify a range of attribute values a region must have to admit a particular interpretation (e.g., tabletops must be horizontal regions, 2-3 feet high). The global consistency of a region interpretation is determined by relational constraints that specify, for each interpretation, the allowed interpretations for an adjacent region in a specified relationship (e.g., a region labeled "door" can appear above an adjacent region labeled "door," "floor," or "doorknob," but not above one labeled "wall"). It is presumed that the correct interpretation of a region will be supported in every adjacent region by at least one interpretation that satisfies <u>all</u> applicable constraints between that pair of regions. Therefore, any region interpretation that lacks at least one such compatible interpretation in every adjoining region can be immediately ruled out. After eliminating a region interpretation, the interpretations of all adjacent regions must be reexamined to determine whether they are still compatible with the remaining interpretations. Deductions may thus propagate, as in Waltz filtering.

2. Illustration of Filtering

The deduction of region interpretations by filtering is illustrated in Figure 15. The example involves an image of an empty room that has been correctly partitioned into six regions corresponding to the objects "floor," "wall," "door," "baseboard," "picture," and "doorknob." The problem is to determine the correct pairing of interpretations and regions. To simplify the example, it is assumed that all boundaries between regions have nonnegligible contrast. Therefore, invoking Relation 5, no adjacent regions will have the same interpretation. Initially, every region is assigned all six possible interpretations, but immediately "picture" and "doorknob" are dropped from Regions 1, 3, and 6 because their size violates Relation 4. This stage of labeling is shown in Figure 15a. Regions are now filtered in pairs in order of region number, beginning with Regions 1 and 2. Relation 1 (within) applies between these regions and eliminates all interpretations but "wall" and "door" for Region 1 and "picture" and "doorknob" for Region 2. Next, Regions 1 and 3 are filtered with Relation 2 (beside), which eliminates "floor" from the possibility set of Region 3. Finally, Regions 1 and 5 are filtered by Relation 3 (above), leaving Region 5 with "floor" and "baseboard" as possible interpretations. The state of interpretation after filtering Region 1 with all its neighbors appears in Figure 15b. Region 2 is now filtered against its neighbor, Region 1, but there are no further eliminations since neither region has changed interpretation since the last time it was filtered.

The process then proceeds to filter Regions 3 and 4 by Relation 1 (within), eliminating "baseboard" from Region 3 and reducing the interpretation possibilities of Region 4 to "doorknob" and "picture." Region 3 is next filtered against Region 5 by Relation 2 (beside), which leaves Region 5 with the unique interpretation "baseboard" and Region 3 with the unique interpretation "door." Finally, Regions 3 and 6 are filtered by



(a) POSSIBLE REGION INTERPRETATIONS* BASED ON REGION SIZE CONSTRAINTS

(b) INTERPRETATIONS SURVIVING AFTER REGION 1 HAS BEEN FILTERED AGAINST ALL ADJACENT REGIONS

(c) INTERPRETATIONS REMAINING AFTER FILTERING REGION 3

(d) FINAL (CORRECT) REGION INTERPRETATIONS

*Legend: D-Door, W-Wall, F-Floor, P-Picture, B-Baseboard, K-Doorknob

Constraints. 1. Within
(Picture within wall)
(Doorknob within door)

2. Beside
(Wall beside door)
(Baseboard beside door)

3. Above
(Wall above baseboard)
(Baseboard above floor)
(Door above floor)

4. Size
(Small doorknob)
(Small picture)

5. All objects assumed to have uniform brightness

APPLICABLE RELATIONS BETWEEN ADJACENT REGIONS

	1	2	3	4	5	6
1	X	W	B	X	A	X
2	W	X	X	X	X	X
3	B	X	X	W	B	A
4	X	X	W	X	X	X
5	A	X	B	X	X	A
6	X	X	A	X	A	X

W Within Contains
A Above Below
B Beside

SA-4683-5

FIGURE 15 DEDUCING REGION INTERPRETATIONS USING RELATIONAL CONSTRAINTS

Relation 3 (<u>above</u>), yielding "floor" as the sole surviving interpretation of Region 6. The current state of interpretation is now as shown in Figure 15c. Region 4 is next filtered against Region 3 using Relation 1 (<u>within</u>), which leaves Region 4 with the interpretation "doorknob." Regions 5 and 1 are filtered using Relation 3 (<u>above</u>), leaving "wall" as the unique interpretation of Region 1. The initial pass concludes by filtering Regions 5 and 6 by Relation 3 (<u>above</u>), with no effect. Every region now has a unique interpretation except for Region 2, which retains the possibilities "picture" and "doorknob." The process continues by reconsidering all pairs of regions whose interpretation sets have changed since they were last filtered. Since "door" was just eliminated from Region 1, Regions 1 and 2 are refiltered by Relation 1 (<u>within</u>) and, this time, Region 2 loses the interpretation "doorknob." The final (correct) interpretation of the scene is shown in Figure 15d.

3. <u>Integration of Filtering and Segmentation</u>

The use of filtering to guide segmentation is summarized in Figure 16. First, the scene is partitioned into regions of pixels with identical brightness. Every region is assigned the complete set of possible interpretations. Adjacent regions are then filtered by making repeated passes through a table of boundaries, each boundary representing a pair of regions. For each pair of regions, a set of applicable relations is determined, based on properties of the common boundary. For example, the regions may be in the relation above/below and have strong boundary contrast. The interpretations of both regions are then individually filtered against all the possible interpretations of the other region. An interpretation is allowed if at least one interpretation of the other region simultaneously satisfies all the applicable relational constraints in conjunction with the interpretation being filtered. If any region interpretations are eliminated for lack of such a compatible

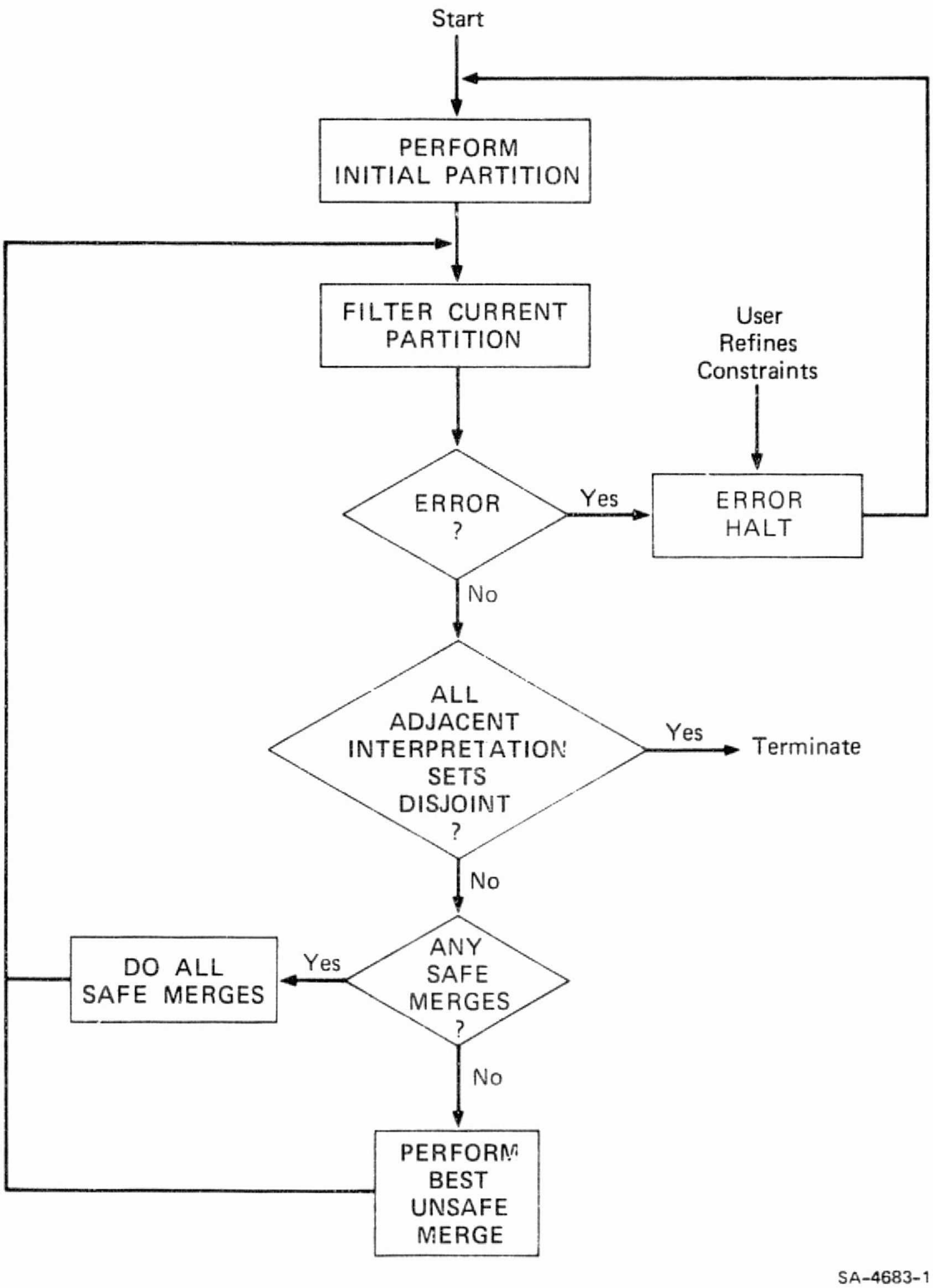


FIGURE 16 FLOWCHART OF CONSTRAINT GUIDED SEGMENTATION

interpretation, all boundaries involving that region are flagged in the boundary table. Initially, a complete pass is made through the boundary table, filtering all adjacent regions. Subsequent passes are made to refilter pairs of regions whose boundaries were flagged on the previous pass. When no flagged boundaries are encountered, filtering is complete.

At the conclusion of filtering, all merges that can be done "safely" are performed. Safe merges incur no risk because the regions involved are known to have the same interpretation (even if the interpretation has not yet been uniquely determined).* After every merge, the boundary table is updated to represent the resulting partition. All boundaries involving the newly created region are flagged.

After all safe merges have been performed, the resulting partition is interpreted by refiltering all flagged boundaries. Note that boundaries are refiltered even when a newly created region has the same interpretation possibilities as both its parents. This is because its boundary relations with adjacent regions may be different from those that previously held for its parent regions. If filtering succeeds in eliminating interpretations, additional safe merges may be possible, which could in turn allow further eliminations. The cycle of safe merges followed by refiltering continues until no further eliminations occur. At this point, if the possible interpretations of all adjacent regions are disjoint, the analysis is complete. Otherwise, a single unsafe merge is performed (between the adjacent regions with at least one common interpretation, which have the weakest boundary contrast) and the interpretation/merge cycle resumed.

---

*A merge between two regions will be safe provided they have the same set of possible interpretations and, moreover, that every region interpretation is supported in the other region only by that same interpretation. This condition is checked with the same routine used for filtering, by testing whether the deletion of each region interpretation would result in the elimination of that interpretation from the other region.

If filtering should ever succeed in eliminating all possible interpretations of any region, the analysis is immediately halted so that constraints can be interactively refined.

4. Error Recovery--The Incremental Acquisition of Knowledge

Errors manifest themselves in three ways: The elimination of all possible interpretations for some region at an interim stage of partitioning, an incorrect final partition, or the incorrect interpretation of regions in the final partition. Error detection is automatic in the first case, but a matter of human judgment in the latter two.

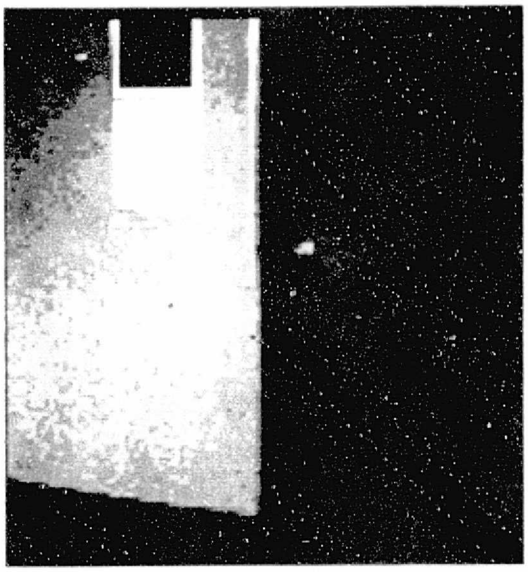
Errors are caused by constraints that are incorrect (e.g., that contain incorrect supporting interpretations), inappropriately applied, or insufficient. Incorrect and inappropriately applied constraints are responsible for eliminations of correct region interpretations and thus for the first and third error manifestations. Insufficient constraints are the primary cause of erroneous unsafe merges, which result because an incorrect region interpretation was not eliminated early enough in the analysis. Ideally, with sufficient constraints, no merge should be unsafe.

Errors resulting from insufficient constraints can be uncovered in a straightforward manner, by examining the resulting partition after each unsafe merge. Erroneous interpretations whose elimination would preclude the erroneous merge can then be identified. Unfortunately, because of the way filtering propagates eliminations, it is frequently difficult to track down the source of errors due to incorrect or inappropriately applied constraints. The fact that some region has been left with no interpretations could be merely an artifact of having eliminated the correct interpretation of some other region much earlier in the analysis. Two key aids are provided to help users deduce the original source of error. First, the analysis can be repeated with an instruction to halt whenever specified interpretations are deleted from regions contained within designated areas. This facility can be used, for example, to halt the analysis as soon as any correct region interpretation is eliminated. Second, upon halting, the user can interrogate the current interpretation possibilities of any region as well as the relations holding between regions in the current partition.

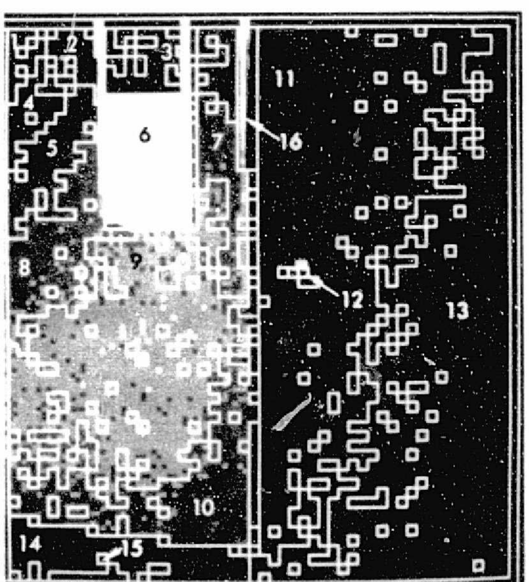
Having located the source of an error, a user can add or modify constraints and then retry the analysis. A correct analysis establishes empirically when the system has sufficient knowledge to process at least the current scene. This incremental mode of acquiring knowledge through debugging proved essential, even in simple scenes, because of difficulties in anticipating the relations that could arise between regions at interim stages of partitioning.

5. Experimental Results

An experimental validation of constraint-guided segmentation was performed in the elementary but nontrivial domain of empty room scenes typified by Figure 17a. Six possible region interpretations were defined: "wall," "door," "picture," "floor," "baseboard," and "doorknob." These interpretations were constrained by the eight relations defined by the boxes in Table 5. Each box gives for each interpretation of a region, R1, the permissible alternative interpretations for a related Region R2. For example, if Region R1 is <u>above</u> R2, then R1 can be "floor" only if R2 can also be "floor." On the other hand, if R1 is <u>below</u> R2, then R1 can be "floor" provided R2 is either "floor," "door," or "baseboard." These constraints were compiled into the filtering program in the form of bit tables so that bits representing required interpretations could be rapidly matched with logical operations against those representing possible region interpretations. Interpretations were not constrained with respect to region attributes such as size, shape, or brightness.



(a) DIGITIZED IMAGE
(8 BITS AT 256 x 256
RESOLUTION)



INTERPRETATION POSSIBILITIES
FOR SELECTED REGIONS
FOLLOWING INITIAL FILTERING

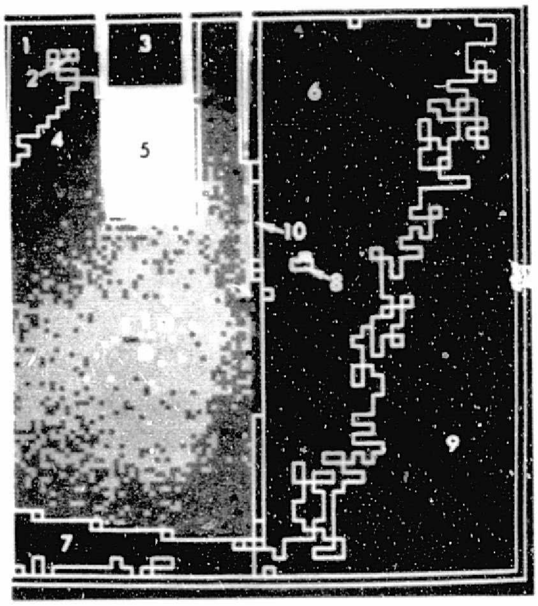
Region	Possible Interpretations
1-5	Picture Wall
6	Picture
7-10	Wall
11	Door
12	Knob
13	Door, Baseboard
14-15*	Baseboard
16*	Universal

(b) INITIAL PARTITION OF ROOM SCENE (264 REGIONS BASED ON 4 SIGNIFICANT BITS OF BRIGHTNESS AT 60 x 60 RESOLUTION)

*Manually assigned interpretation.

SA-4683-10

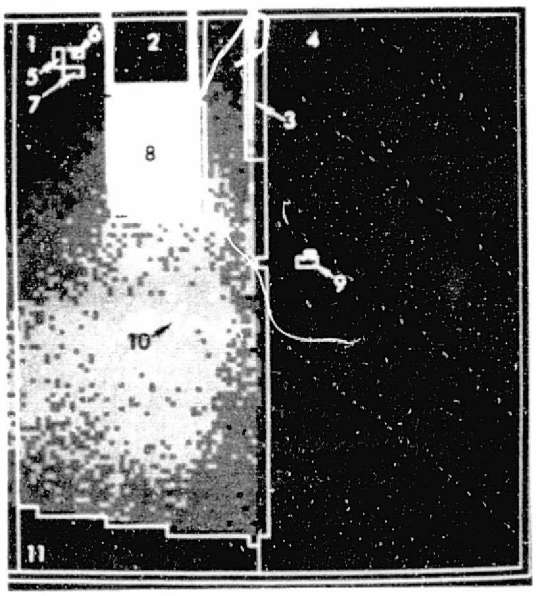
FIGURE 17 CONSTRAINT GUIDED SEGMENTATION OF ROOM SCENE



INTERPRETATION POSSIBILITIES FOR SELECTED REGIONS FOLLOWING REFILTERING

Regions	Possible Interpretations	Average Brightness
1	Picture, Wall	52
2	Picture	188
3	Picture, Wall	16
4	Wall	89
5	Picture	182
6	Door	13
7	Baseboard	27
8	Knob	64
9	Door, Baseboard	17
10	Wall, Door	48

(c) ROOM SCENE PARTITION AFTER 200 SAFE-MERGES



FINAL REGION INTERPRETATIONS

Region	Interpretation
1	Wall
2	Picture
3	Universal
4	Door
5	Picture
6	Picture
7	Picture
8	Picture
9	Knob
10	Picture
11	Baseboard

(d) FINAL PARTITION OF ROOM SCENE

SA-4683-11

FIGURE 17 CONSTRAINT GUIDED SEGMENTATION OF REGION SCENE (Concluded)

Table 5

RELATIONS GOVERNING INTERPRETATIONS OF ADJACENT REGIONS IN ROOM SCENE DOMAIN

R1 Above R2*	
R1	R2
Baseboard	Floor, Baseboard
Door	Knob, Floor, Door
Floor	Floor
Wall	Picture, Wall, Baseboard
Picture	Picture, Wall
Knob	Knob, Door

R1 Below R2	
R1	R2
Baseboard	Wall, Baseboard
Door	Knob, Door
Floor	Floor, Door, Baseboard
Wall	Picture, Wall
Picture	Picture, Wall
Knob	Knob, Door

R1 Beside R2	
R1	R2
Baseboard	Door, Baseboard
Door	Knob, Wall, Door, Baseboard
Floor	Floor
Wall	Picture, Wall, Door
Picture	Picture, Wall
Knob	Knob, Door

R1 Adjacent to R2	
R1	R2
Baseboard	Wall, Floor, Door, Baseboard
Door	Knob, Wall, Floor, Door, Baseboard
Floor	Floor, Door, Baseboard
Wall	Picture, Wall, Door, Baseboard
Picture	Picture, Wall
Knob	Knob, Door

R1 Contrasts With R2	
R1	R2
Baseboard	Knob, Picture, Wall, Floor, Door
Door	Knob, Picture, Wall, Floor, Baseboard
Floor	Knob, Picture, Wall, Door, Baseboard
Wall	Knob, Picture, Floor, Door, Baseboard
Picture	Knob, Picture, Wall, Floor, Door, Baseboard
Knob	Picture, Wall, Floor, Door, Baseboard

R1 No Contrast With R2	
R1	R2
Baseboard	Knob, Picture, Door, Baseboard
Door	Picture, Door, Baseboard
Floor	Knob, Picture, Wall, Floor
Wall	Knob, Wall, Floor
Picture	Knob, Picture, Floor, Door, Baseboard
Knob	Knob, Picture, Wall, Floor, Baseboard

R1 Inside R2	
R1	R2
Baseboard	Baseboard
Door	Door
Floor	Floor
Wall	Wall
Picture	Picture, Wall
Knob	Knob, Door

R1 Outside R2	
R1	R2
Baseboard	Baseboard
Door	Knob, Door
Floor	Floor
Wall	Picture, Wall
Picture	Picture
Knob	Knob

*Box lists the interpretations of Region R2 that are compatible with each interpretation of Region R1, given that R1 is above R2. Other relations are analogously defined.

ORIGINAL PAGE IS OF POOR QUALITY

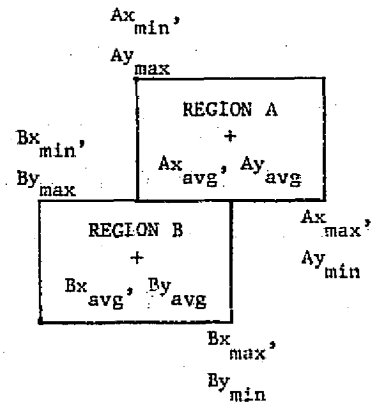
Applicable relations between a pair of regions were determined in this experiment by factors that could be most easily extracted from an existing region data structure. The conditions of applicability are summarized in Table 6. Applicability of the relations above, below, and beside is based on the relative image coordinates of the regions' centers of mass and vertices of their bounding rectangles (derived from X, Y boundary extrema). Region R1, for example, is defined to be above Region R2 provided its highest boundary point is higher in the image than the highest point of R2 and its lowest point is higher than R2's center of mass. It was also required that the horizontal extents of R1 and R2 overlap, and that the size of both regions exceed 5 pixels. The two last requirements decrease (but do not eliminate) the possibility that a relation will be prematurely applied at an early stage of partitioning (see Figure 18 in conclusion). Below is defined as the converse of above. Beside is a symmetric relation that applies when regions with vertical overlap are sufficiently displaced in a horizontal direction.

Adjacency is a universal relation that applies between any regions with a common boundary. Inside and outside refer to regions that are holes within other regions. These three relations are topological properties of the region data structure and not subject to the artifacts of merging. They are therefore applied regardless of region size.

The relation contrast applies whenever the average brightness of two regions exceeds a conservatively large threshold (T1). The relation no-contrast applies when the difference is less than a second, conservatively small threshold (T2). For the current room domain, these thresholds were empirically set at T1 = 42 and T2 = 15 (assuming 256 brightness levels). The contrast constraint insures that two adjacent contrasting regions will not receive the same interpretation if a surface with that interpretation is known to be approximately uniform in brightness. (It is assumed that all objects in the room domain except for "picture" have uniform brightness.)

Table 6

CONDITIONS OF APPLICABILITY FOR RELATIONS BETWEEN ADJACENT REGIONS



1. Above/Below

Region A is above Region B if

$$(Ay_{max} > By_{max}) \wedge (Ay_{min} > By_{avg}) \wedge$$
$$[((Bx_{min} \le Ax_{min}) \wedge (Ax_{min} < Bx_{max})) \vee$$
$$((Bx_{min} < Ax_{max}) \wedge (Ax_{max} \le Bx_{xmax})) \vee$$
$$((Ax_{min} \le Bx_{min}) \wedge (Ax_{max} \ge Bx_{max}))]$$

Region A is below Region B if

$$(Ay_{min} < By_{min}) \wedge (Ay_{max} < By_{avg}) \wedge$$
$$[((Bx_{min} \le Ax_{min}) \wedge (Ax_{min} < Bx_{max})) \vee$$
$$((Bx_{min} < Ax_{max}) \wedge (Ax_{max} \le Bx_{max})) \vee$$
$$((Ax_{min} \le Bx_{min}) \wedge (Ax_{max} \ge Bx_{max}))]$$

2. Region A is beside Region B if

$$[((Ax_{max} > Bx_{max}) \wedge (Ax_{min} > Bx_{avg})) \vee$$
$$((Bx_{max} > Ax_{max}) \wedge (Bx_{min} > Ax_{avg}))] \wedge$$
$$[((By_{min} \le Ay_{min}) \wedge (Ay_{min} < By_{max})) \vee$$
$$((By_{min} < Ay_{max}) \wedge (Ay_{max} \le By_{max})) \vee$$
$$((Ay_{min} \le By_{min}) \wedge (Ay_{max} \ge By_{max}))]$$

3. Region A cortrasts with Region B if

$$|\text{Brightness A} - \text{Brightness B}| > T1$$

4. Region A has no contrast with Region B if

$$|\text{Brightness A} - \text{Brightness B}| < T2$$

Note: Applicability of the relations adjacent, inside and outside is determined by topological properties of the region data structure.

The relation no-contrast insures that two regions with similar brightnesses will not receive different interpretations whose brightnesses are known to be significantly different, for example, "wall" and "door."

The image in Figure 17a was digitized to 8 bits at 60 × 60 resolution. An initial partition of this digitized image, based on the four most significant bits of brightness,* is shown in Figure 17b.

There were 264 regions in the initial partition. All regions, with two exceptions, were initially assigned the set of all possible interpretations. The first exception involved an isolated one-pixel region at the bottom of the image (Number 15 in Figure 17b), which was manually assigned the unique interpretation "baseboard." This assignment was made to explicitly exclude the case where every region in the image receives the interpretation "picture" (i.e., the image portrays a picture of a room scene rather than a room scene). The second exception involved the thin vertically elongated rectangular region (Number 16) at the top of the image between the "door" and "wall." This very bright region was an anomaly, the result of specular reflections from a doorframe that, otherwise, was indistinguishable from the "wall." While such anomalies are undeniably a part of real scenes, we saw no reason to complicate the initial experiment by introducing additional interpretations specifically to account for them. The region was thus manually assigned a special universal interpretation that both supports and is supported by any adjacent interpretation. With this interpretation, the anomalous region was effectively removed from the analysis since it could not participate in filtering or safe merges, and could merge unsafely only with another region that had the same special interpretation.

---

*A 4-bit partition was chosen as an experimental expedient to minimize the number of regions without losing any significant boundaries.

The above set of region interpretations was filtered using relational constraints applicable in the initial partition. The results of filtering are shown for selected regions in the caption of Figure 17b. Note that many parts of the scene have already acquired unique interpretations. These parts include large areas of the "wall" (Regions 7-10) and "door" (11), as well as the "baseboard" (14), the "doorknob" (12), and the lower (bright) half of the "picture" (6). Many of the smaller regions contained <u>within</u> these areas are also uniquely labeled with the same interpretation as the containing region.

During filtering, eliminations propagated from the manually assigned "baseboard" interpretation. The possibilities for Region 14, <u>adjacent</u> to and <u>noncontrasting</u> with Region 15 (known to be baseboard) were immediately reduced to "door" or "baseboard." Regions 10 and 14 could then be filtered by the relations <u>above</u> and <u>contrast</u>, leaving those regions with the unique interpretations "wall" and "baseboard," respectively. The interpretation of Region 13, <u>beside</u> and <u>noncontrasting</u> with "baseboard" Region 14, was then narrowed to the alternatives "door" and "baseboard." The interpretation "wall" propagated upward from Region 10 to Region 9 through the relations <u>above</u> and <u>no-contrast</u>, and subsequently to Regions 5, 7, and 8. This, in turn, allowed Region 6 to be interpreted as "picture" since it is <u>above</u> and <u>contrasting</u> with Region 9, now known to be "wall." Meanwhile, Region 11, which is <u>beside</u> and <u>contrasting</u> with Region 9 ("wall") and <u>adjacent</u> and <u>noncontrasting</u> with Region 13 ("door" or "baseboard"), is uniquely constrained to be "door."

The initial stage of filtering leaves two main areas of the image with uncertain interpretations. Region 13 and its interior regions still admit the possibilities "door" or "baseboard," while Regions 1-6 in the upper left part of the scene can each be interpreted as either "wall" or "picture." The "door"/"baseboard" ambiguity persists because Regions 11 and 13 do not satisfy the formal conditions defining the relation <u>above</u>. The second ambiguity arises because of a brightness gradient across the wall such that Regions 5 and 8 do not fulfill the conditions for either <u>contrast</u> or <u>no-contrast</u>. As a consequence, the interpretation "picture" cannot be eliminated from Region 5 and the resulting "wall"/"picture" ambiguity then propagates to the other regions in the area. A third and relatively minor area of ambiguity exists among the small regions on the border between "wall" and "door." These regions, adjacent to both "door" and "wall," are classified as either "door," "wall," or "baseboard."

Approximately 200 safe merges are performed, based on the interpretations surviving the initial filtering. The resulting partition, containing about 68 regions, is refiltered, yielding the results shown in Figure 17c. The safe merges primarily involved adjacent regions already having the same unique interpretations. Regions in the upper part of the wall with possible interpretations "wall" and "picture" could also be safely merged where the <u>contrast</u> constraint did not apply (since a "picture"/"wall" boundary is required to have <u>contrast</u>). Although the resulting partition appears much cleaner, the same basic ambiguities persist. These ambiguities must now be resolved by postulating unsafe merges, based on the region brightnesses included in the caption of Figure 17c.

The first unsafe merge of consequence occurred with approximately 43 regions remaining. Regions 6 and 9 (in Figure 17c), with a contrast of 4, were merged into a single region with the unique interpretation "door" (the intersection of the interpretation possibilities for Regions 6 and 9) and an average brightness of 15. Next, with approximately 25 regions left, Regions 1 and 4 (contrast 37) were merged to form one large region of "wall" with brightness 87. As a result of this merge, the contrast relation could now be applied to eliminate the interpretation "wall" from Region 3. Finally, with about 20 regions left, the small regions, such as 10, between "door" and "wall" were merged unsafely into "wall." At this point, after a total of 43 unsafe merges and 214 safe merges, the analysis terminated with 11 regions remaining, all having unique and disjoint interpretations.

The final partition and associated region interpretations are shown in Figure 17d. The analysis is essentially correct, given the limited semantics used in the experiment. A wall-mounted thermostat was fragmented into three regions (5-7), which were then interpreted as "pictures." A noisy pixel in the center area of the wall area was also assigned the interpretation "picture." These interpretations occurred because "picture" was the only legal possibility for a contrasting region contained within a region labeled "wall." The interpretation errors could have been avoided by introducing explicit interpretations for "thermostat" and "noise" (which would be distinguished from "picture" by additional constraints on region size). Finally, the so-called picture, actually a Sierra Club calendar, was split into two regions, containing respectively, a landscape and numeric data. These parts of the calendar were physically connected by a spiral binding which was invisible in the digitized image.

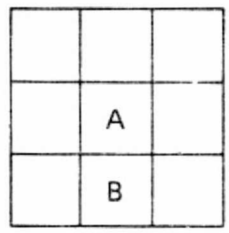
6. Discussion

The present set of constraints was conceived as an initial test of constraint-guided interpretation and, as such, makes no pretense at semantic generality. Thus, it assumes a particular viewing position and is dependent on a number of thresholds concerning region attributes, such as size and brightness. We plan to reformulate the constraints so as to remove these limitations and then evaluate the performance of the paradigm on a reasonable sampling of room scenes.

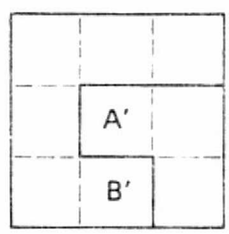
More generally, binary valued relations between adjacent regions often cannot adequately constrain interpretations. First, defining relations between adjacent regions is of questionable value in scenes containing significant occlusion. Attributes of the individual regions, such as size, shape, color, and texture can still be used to prune interpretations. Alternatively, relations such as above, beside, and contrast can be redefined for nonadjacent regions. This would increase overhead in the filtering algorithm but it might also allow the implications of an elimination to propagate faster. A second drawback of the current constraints is their binary "all or nothing" nature. If two large regions touch along a very small fragment of their boundaries, should this be sufficient grounds to exclude absolutely an interpretation that violates an adjacency constraint? In cases such as this, it seems more natural for constraints to decrease the likelihood of that interpretation, but not necessarily all the way to zero. Absolute elimination is particularly risky because the filtering algorithm can propagate the consequences of any error throughout the image, possibly resulting in many other errors. A third limitation concerns the restrictive way in which constraints must currently be expressed: as sets of interpretations that may be compatible with a given interpretation. One might also want to impose stronger must and must-not conditions. It should be possible, for example, to require that an interpretation "doorframe" must be adjacent to at least one region with the possible interpretation "door," or to require that two regions on opposite sides cannot both be uniquely interpreted as "doors." Ideally, it should be possible to formulate constraints as arbitrary procedures. This would also allow conditions of applicability for the constraint to be specified independently for each interpretation. We have, in fact, experimented with a LISP program called MSYS, which performs region interpretations based on real valued, procedurally represented constraints. However, MSYS (which runs slowly) has not yet been integrated with a segmentation program to perform a complete scene analysis.

The above limitations can be viewed as shortcomings of the current implementation. There are, however, a number of deeper conceptual problems concerning the filtering paradigm that have not yet been satisfactorily resolved. A major source of concern is the fact that relations between a region and one of its neighbors can cease to apply when that region is merged with another neighbor (see Figure 18). In other words, a relation that may have already been used to eliminate a correct region interpretation is shown in a subsequent stage of partitioning to have been invalid--an artifact of the grain of the previous partition. Unfortunately, because of the way eliminations propagate, there is no obvious way to either diagnose or recover from such errors.

A second fundamental issue concerns the extensibility of the filtering approach. The present system has been demonstrated in a domain containing less than 10 objects. Whenever a new interpretation type is added, every constraint must be modified to express relations between it and all previously defined interpretations. Obviously, the list of possible interpretations cannot expand without limit. How then, could the paradigm be applied in natural scenes containing innumerable objects? One approach would be to make the initial level of interpretation domain independent. Regions would be interpreted initially in terms of descriptive surface characteristics such as curvature (planar, convex, concave), orientation (vertical, horizontal), texture, and material (e.g., metal, plastic, wood) that are common to many domains. This level of interpretation would be based on domain-independent constraints dealing with shading, illumination, shadowing, occlusion, and so forth. Interpretations at the level of specific objects would then be introduced, together with appropriate constraints, as a consequence of establishing associated surface characteristics. Thus, a large vertical planar surface might invoke the interpretation "wall." Determining whether the interpretation guided segmentation paradigm will actually work with domain-independent interpretations is one of our major research objectives.



(a) SECTION OF IMAGE AT AN EARLY STAGE OF PARTITIONING (REGION A ABOVE REGION B)



(b) SAME SECTION OF IMAGE: REGIONS A AND B HAVE BEEN MERGED INTO REGIONS A′ AND B′ RESPECTIVELY, WITH (REGION B′ ABOVE REGION A′)

SA-4683-3

FIGURE 18 PREMATURE APPLICATION OF ABOVE RELATION AT AN EARLY STAGE OF PARTITIONING

E. Conclusion

The scene analysis paradigm described in this chapter has two main features: segmentation and interpretation are completely and effectively integrated; and many diverse sources of knowledge can be used to guide the analysis. The second feature is particularly significant in that the effectiveness of a scene analysis technique is usually correlated with its ability to capitalize on prior knowledge about the depicted scene. So far, we have experimented with three sources of knowledge: direct manual interaction, geometric models, and relational constraints. Additional sources that have been contemplated include maps, region attributes, and prior analyses of the scene (from similar viewpoints), perhaps by other scene analysis programs. All these knowledge sources can be expressed in a uniform way as constraints on the possible interpretations of regions. Multiple sources of knowledge can thus be combined in a straightforward way so that incremental additions of knowledge (or, equivalently, human guidance) will effect incremental improvements in performance.

Areas for improvement have previously been suggested in the discussions following each experiment. One way of improving performance in all tasks is by improving the underlying region-merging process. First, the current method of obtaining an initial partition is quite crude and incurs a significant risk of grouping pixels from different objects in the same region. Several recently developed segmentation programs can do much better. In particular, a program by Yakimovsky [9] forms a partition based on the output of a sophisticated edge operator; regions in the partition are defined as sets of pixels that can be connected by a path that does not cross a ridge of edge values. Second, the ordering of unsafe merges could be improved by relying on more elaborate region descriptions. Comparing the textures and brightness gradients of regions, in addition to their average colors, should significantly improve the basic decision regarding whether two regions belong to the same surface. (This will certainly be true in monochrome images.) Third, there is at present no provision for splitting regions if a merge error is detected. Such a capability would relax the requirements on both initial partitioning and merging.

VI APPLICATION OF INTERACTIVE SCENE ANALYSIS TECHNIQUES TO CARTOGRAPHY

A. Introduction

The production of maps from aerial photographic data is, despite a large body of mechanical techniques, primarily labor-intensive. One of the most time-consuming steps in this process is the delineation of topographic, cultural, and land-use features, such as lakes, rivers, roads, and drainages. Currently, a trained operator must manually trace the detailed boundaries of features, a lengthy process. Similar problems also occur in digitizing maps for later updating.

In such a labor-intensive craft, it is reasonable to look toward computers as a possible means for eliminating much of the routine work. The idea of a fully automatic, aerial photograph-to-map computer system, while appealing, is not only infeasible at the present time but is likely to remain so for the forseeable future. A more promising approach would be to develop an interactive system which an operator could quickly program to extract specific features in a specific type of terrain. The feasibility of such an interactive approach has been successfully demonstrated at SRI using our ISIS [1].

B. Example

The following scenario illustrates how a user and interactive system might work together on a typical cartographic task, extracting an outline of the large lake in Figure 19.* Human input will be shown by thick white

*Figure 19 is an orthophoto of Fort Sill, Oklahoma, coarsely digitized at 256 x 256 resolution. A coarse digitization was used to speed processing for this example.

lines and the computer's response by thin ones. In Figure 20, the user has designated an area of interest that is then displayed at a magnified scale. In Figure 21, a crude triangular region is drawn by the user to indicate roughly the center of the lake. The computer's initial guess, shown in Figure 22, contains both errors of omission (samples excluded along the periphery of the lake), and of commission (unwanted tail in lower left-hand corner of the lake). The operator crudely encircles the tail (Figure 23) and tells the computer to omit all points in the enclosed region. He also points at several omissions (the crosses in Figure 23). The computer responds with the boundary shown in Figure 24.

C. Method of Approach

The examples and counterexamples of lake were used to develop and debug interactively a computer procedure for distinguishing between pixels (picture elements) from the lake and those from the shore. The resulting procedure was then used by a conventional boundary-following algorithm to extract the lake outline.

This algorithm first detects the lake boundary by scanning outwards from the center of the designated triangle until the discrimination procedure classifies a pixel as "nonlake." It then follows the boundary in a counterclockwise direction. The next boundary point is determined by applying the discrimination procedure to the pixel immediately to the right of the present boundary element and then testing pixels in a counterclockwise arc about the present element until a "lake" classification is encountered.

The interesting part of this work concerns the methodology used to develop the discrimination procedure. The objective is to construct the simplest procedure, using all available feature extraction operators, for distinguishing example points from counterexample points. Table 7 lists typical feature extraction operators, ordered by computational

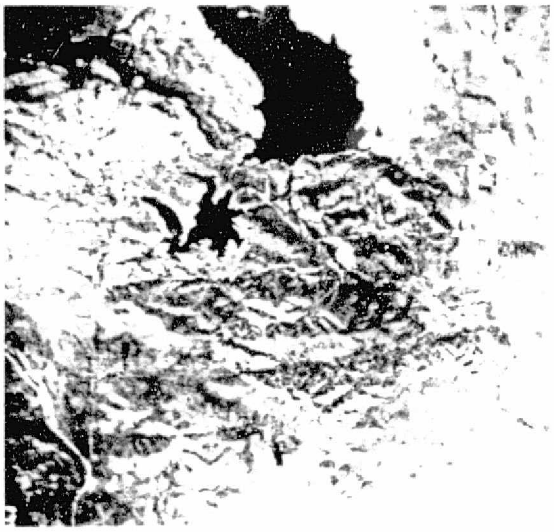


FIGURE 19 DIGITIZED AERIAL VIEW OF FORT SILL, OKLAHOMA

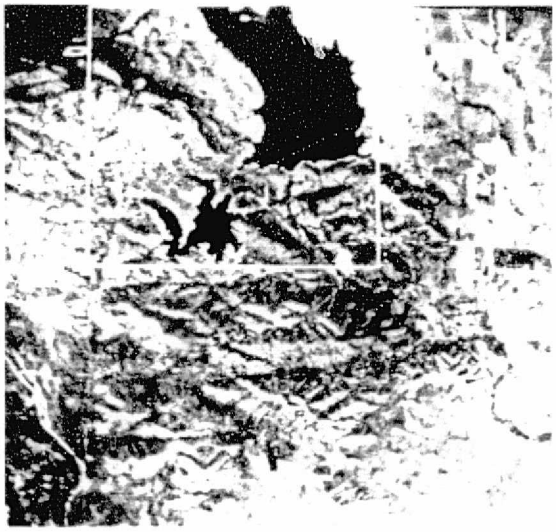


FIGURE 20 WINDOWING TO OBTAIN MAGNIFIED DISPLAY OF WORK AREA

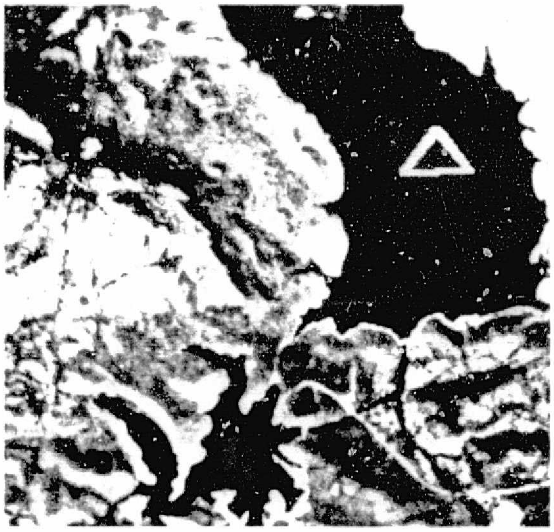


FIGURE 21 USER MANUALLY DESIGNATES A FEW IMAGE POINTS CONTAINED IN LARGE LAKE

FIGURE 22 INITIAL BOUNDARY WITH DEFECTS

SA-4683-7

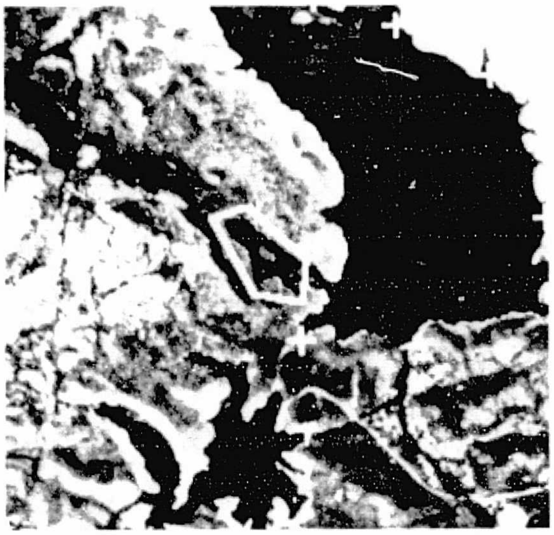


FIGURE 23 USER INDICATES ERRORS

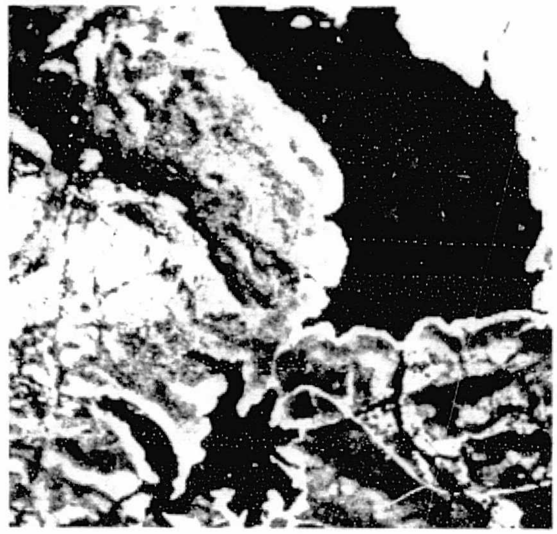


FIGURE 24 FINAL BOUNDARY OF LARGE LAKE AFTER UPDATING MODEL

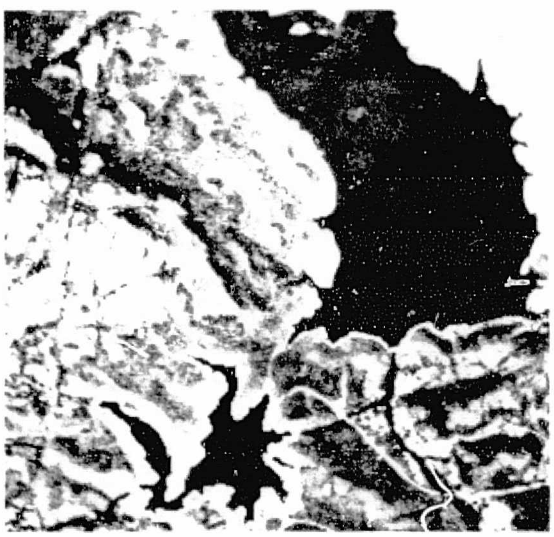


FIGURE 25 FINAL BOUNDARY OF SMALL LAKE

FIGURE 26 OUTLINE OF RIVER AFTER DESIGNATING ONE POINT IN THE UPPER BRANCH

SA-4683-8

Table 7

TYPICAL OPERATORS

Point Operators (applied to individual pixels)
Brightness
Color (hue and saturation)
Elevation
Local Area Operators (applied to sets of contiguous pixels in small circular or oblong areas)
Average of attribute values
Distribution of attribute values
Weighted averages (templates)
Region Operators (applied to sets of contiguous pixels)
Texture over regions
Shape of regions
Size of regions

Table 8

GRAPHICAL INTERACTION (POINTING)
MODES FOR DESIGNATING
EXAMPLES AND COUNTEREXAMPLES

Single Points
Small Regions
Inside
Outside
Crude Outline
Inscribed
Circumscribed
Detailed Outline
Segments
Complete

complexity. Details of these and other operators can be found in standard texts on scene analysis [10-11]. There is a hierarchy of graphic interaction (pointing) modes, as Table 8 indicates, by which the machine can be shown examples. From a single sample pixel, it is possible to construct a program that accepts contiguous pixels whose point attributes (i.e., brightness, hue, or elevation, if available) differ from the indicated pixel by less than a threshold. An implicit inference is being made here that the rest of the pixels on the feature resemble this single pixel. Given an example region, the thresholds can be widened to encompass the range of attributes measured on that region. Counterexamples can than be used to narrow these limits. In general, the more complete the example, the less iteration will be required to develop a good specification. If an example and counterexample cannot be distinguished on the basis of thresholded point attributes, averages or distributions of attribute values over local areas can be used. If this still is not sufficient, an attempt can be made to distinguish between the two on the basis of the size and shape of the regions delineated by outlining. The final procedure will be composed of conjunctions and disjunctions of these processes.

Now, we will examine in detail the interactive process by which the lake-outlining procedure illustrated above was developed. The sampled, digitized image (Figure 19) was read into ISIS and displayed on a RAMTEK self-refreshing CRT. With the cursor, the user then created a small region in the interior of the lake and asked the system for a distribution of brightness values for pixels in this area. From these data he composed a simple program that determined whether a pixel belonged to the lake based on thresholded brightness (the only available point attribute). The edge follower used the program to produce the outline shown in Figure 22.

The user next drew a crude boundary around the "bad" pixels in the tail and again requested a brightness distribution. A significant overlap with the previous distribution of example pixels was observed. Adequate discrimination was achieved empirically by increasing the operator size so that brightness of a point was computed as the average brightness over a circular area centered on the point. This crude spatial filtering acted to exclude dark areas of the image with insufficient width to qualify as lakes. Finally, the brightness threshold was widened to include the brightnesses of the missed points that the user had indicated with the cursor. Using the updated program, the edge follower was able to obtain an outline that tracked fairly accurately the actual lake.

The final procedure for distinguishing lake points from nonlake points is, in fact, a "model" for what pixels from a lake look like to the computer. The program was written on-line in an interactive language (LISP) and then debugged interactively as contingencies arose. Interactive refinement is a powerful concept for a scene analysis programmer. It frees him from the necessity of formulating programs in a language that is understood by the machine but that is cumbersome for people. Instead, it allows direct communication with the program via a common language of images. Debugging is simplified in this system. Instead of predicting the problems that the system is likely to encounter, the program is executed on exemplary images and debugged when errors arise.

D. Further Examples

1. Automatic Extraction of Previously Learned Features

The procedure developed in tracing the first lake can serve as the initial basis for extracting other lakes in similar terrain. Even if the outline is not exact, it provides a good staring point for further interaction. Figure 25 shows a boundary extracted for the small lake using the same discrimination procedure developed for the large lake. In this example, the user manually designated a single pixel in the center of the second lake to initiate the boundary follower. Alternatively, a starting point could have been acquired automatically by scanning systematically through the image for a reasonably sized set of contiguous pixels satisfying the criteria for "lake." Note, that any subsequent interaction required to refine a boundary could be used to further improve or generalize the original discrimination procedure.

2. Linear Features

Linear fatures, such as rivers and roads, may also be outlined using similar interactively generated procedures. In Figure 26, we show the upper branch of the river connecting the two lakes. Here the user pointed at a single river point just above the fork. Starting from this point and using a threshold based on its brightness, the boundary follower tracked the river until it intersected the road. The trainer next indicated additional starting points on each river branch below the road, and using the same threshold, the river boundary was completed. The final river boundary is shown in Figure 27. These crude boundaries could be improved by applying a thinning algorithm [12]. Figures 28 and 29 show the final results of tracing the designated features and then projecting them back onto the orignal, high-resolution image.

E. Possible Extensions

1. Automatic Generation of Discrimination Procedures

The above examples required that the user supply discrimination procedures for distinguishing between the brightness distributions of designated regions. These procedures were interactively formulated using data provided by the system. An obvious next step would be to have these

FIGURE 27 OUTLINE OF RIVER AFTER DESIGNATING AN ADDITIONAL POINT IN EACH LOWER BRANCH

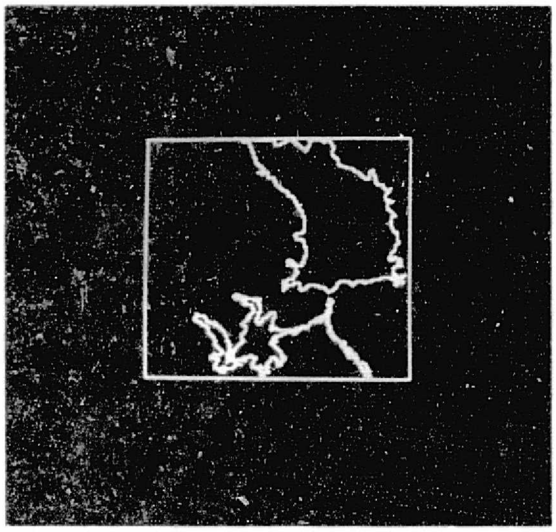


FIGURE 28 COMPLETED MAP OF MAJOR WATERWAYS WITHIN WINDOW

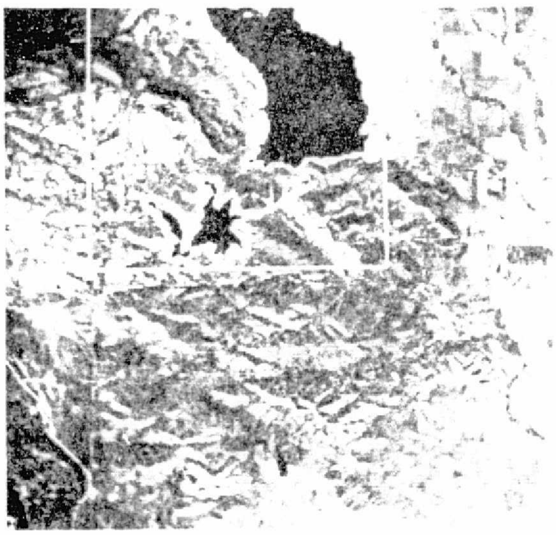


FIGURE 29 COMPLETED MAP SUPERIMPOSED ON ORIGINAL IMAGE

SA-4683-

procedures formulated automatically by the system based on the user designated examples and counterexamples. In this mode of operation, a user might crudely sketch a feature of interest. The system would use this to formulate a discrimination procedure and then attempt to trace a detailed boundary. Any errors made by the system could be refined interactively.

For simple discrimination procedures of the type described in this paper, automatic generation appears straightforward. Existing ISIS subroutines could be used, for example, to select the appropriate threshold and operator size for distinguishing the brightness distributions of example and counterexample points [2]. The same approach should be applicable with the other operators in Table 7 when additional discrimination is required.

2. Elevation Data

The availability of elevation data would make many of the tasks described above much simpler. The constant elevation of a lake, combined with local brightness values, would provide a powerful discriminating test. And, in many cases where the brightness contrast between two features is poor, a difference in slope or elevation may be sufficient to distinguish them. Similarly, features in mountainous terrain would prove more tractable with elevation data.

3. Digitization of Existing Maps

The same techniques used to trace features on aerial photos would also be useful for tracing features on existing maps to reduce them to digital form. In many cases, the processing should, in fact, be easier, because of the better contrast available in maps. These digitized maps could then be updated interactively using recent aerial photographs. Ultimately, information in existing digitized maps could be used in lieu of pointing to indicate preexistant features on the photograph. This would allow the program to use the digitized map to guide the subsequent analysis of the aerial photograph, in the same way as would a person.

4. Elimination of Map Editing

The process we have described should eliminate the need for an independent editing step after the map features have been extracted. The editing is an inherent part of the process of incremental refinement of the outline and, therefore, should not be normally needed as a post-processing step.

F. Conclusions

We believe that the examples described above demonstrate the technical feasibility of applying interactive scene analysis techniques to cartography. Whether or not the techniques developed will prove practical in actual cartographic use is, of course, a matter for further study. The simple feature extraction operators used (essentially a threshold applied to the average brightness computed over a bar-shaped operator) almost certainly will not suffice in more complex aerial scenes. Moreover, processing times may become a key factor at the image resolutions required for cartographic accuracy. An appealing aspect of the interactive approach is that, when necessary, the user can always revert to detailed manual tracing. Thus, our approach would be useful even if it applied in only some of the cases encountered in practice.

In the future, we plan to apply interactive techniques in a variety of other problem domains involving large volumes of graphic and pictorial data that are difficult to extract in digital form by either strictly manual or automatic means.

VII REFERENCES

1. J. M. Tenenbaum, T. D. Garvey, S. Weyl, and H. C. Wolf, "An Interactive Facility for Scene Analysis Research," SRI Artificial Intelligence Center Technical Note 87, Stanford Research Institute, Menlo Park, California (January 1974) (Summarized in proceedings, 2nd International Joint Conference on Pattern Recognition, Copenhagen, Denmark, August 1974).

2. T. D. Garvey, Perceptual Strategies for Purposive Vision," SRI Artificial Intelligence Center Technical Note 117, Stanford Research Institute, Menlo Park, California (February 1976).

3. J. M. Tenenbaum and S. Weyl, "A Region-Analysis Subsystem for Interactive Scene Analysis," Preprints of the 4th International Joint Conference on Artificial Intelligence, Tbilisi, USSR (September 1975).

4. J. M. Tenenbaum and H. G. Barrow, "MSYS: A System for Reasoning About Scenes," SRI Artificial Intelligence Center Technical Note 121, Stanford Research Institute, Menlo Park, California (March 1976).

5. S. Weyl, "An Interlisp Relational Data Base System," SRI Technical Note 116, Stanford Research Institute, Menlo Park, California (January 1976).

6. J. M. Tenenbaum, T. D. Garvey, S. Weyl, and H. C. Wolf, "Research in Interactive Scene Analysis," Annual Report to NASA for 1974, SRI Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (March 1975).

7. G. Agin, "Geometric and Spatial Models--Visible Outlines," p. 90 in <u>Artificial Intelligence--Research and Applications</u>, N. Nilsson, ed, Progress Report to ARPA, SRI Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (May 1975).

8. D. Waltz, "Generating Semantic Descriptions from Drawings of Scenes with Shadows," Artificial Intelligence Technical Report 271, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts (August 1972).

9. Y. Yakimovsky, "Boundary and Object Detection in Real World Images," Technical Memorandum 33-709, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California (March 1975).

10. Azriel Rosenfeld, Picture Processing by Computer (Academic Press, New York, New York, 1969).

11. R. O. Duda, and P. E. Hart, Pattern Classification and Scene Analysis (Wiley-Interscience, New York, New York, 1973).

12. J. M. S. Prewitt, "Object Enhancement and Extraction," p. 120 in Picture Processing and Psychopictorics, Lipkin and Rosenfeld, eds. (Academic Press, New York, New York, 1970).