PHOTOGRAPH THIS SHEET

AD A118738

DTIC ACCESSION NUMBER

II

LEVEL

INVENTORY

Image Understanding Research

DOCUMENT IDENTIFICATION Final Rept., 1 Oct. 81-28 Feb '82 Mar. 82

Contract F33615-80-C-1080 AFWAL-TR-82-1111

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

82

DISTRIBUTION STATEMENT

ACCESSION FOR
NTIS GRA&I
DTIC TAB
UNANNOUNCED
JUSTIFICATION

BY
DISTRIBUTION /
AVAILABILITY CODES

| DIST | AVAIL AND/OR SPECIAL | |
|---|---|---|
| A | | |

DISTRIBUTION STAMP

DTIC
ELECTE
S AUG 30 1982 D
D

DATE ACCESSIONED

DEIB
INSPECTED
2

82 08 27 101

DATE RECEIVED IN DTIC

PHOTOGRAPH THIS SHEET AND RETURN TO DTIC-DDA-2

DTIC FORM OCT 79 70A

DOCUMENT PROCESSING SHEET

AFWAL-TR-82-1111

# IMAGE UNDERSTANDING RESEARCH

Ramakant Nevatia et al.
UNIVERSITY OF SOUTHERN CALIFORNIA
INTELLIGENT SYSTEMS GROUP
DEPARTMENTS OF ELECTRICAL ENGINEERING
AND COMPUTER SCIENCE
UNIVERSITY PARK
LOS ANGELES CA 90007

MARCH 1982

Final Report for Period 1 October 1981 - 25 February 1982

Approved for public release; distribution unlimited.

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433

*NOTICE*

*When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.*

*This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.*

*This technical report has been reviewed and is approved for publication.*

Louis A. Tamburino

LOUIS A. TAMBURINO
Project Engineer

Donald L. Moon

DONALD L. MOON, Chief
Information Processing Technology Branch
Avionics Laboratory

*FOR THE COMMANDER*

Richard H. Boivin

RICHARD H. BOIVIN, Colonel, USAF
Chief, System Avionics Division
Avionics Laboratory

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

*"If your address has changed, if you wish to be removed from our mailing list; or if the addressee is no longer employed by your organization please notify* AFWAL/AAAT-1 *W-PAFB, OH 45433 to help us maintain a current mailing list".*

*Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.*

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>AFWAL-TR-82-1111 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>IMAGE UNDERSTANDING RESEARCH | | 5. TYPE OF REPORT & PERIOD COVERED<br>Final Technical Report<br>1 Oct. 81- 28 Feb. 1982<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Ramakant Nevatia et al. | | 8. CONTRACT OR GRANT NUMBER(s)<br>F-33615-80-C-1080 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Depts. of Electrical Engin. and Computer Science<br>University of Southern Calif., LA.,CA.90007 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>DARPA Order No.3119 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Defense Advanced Research Projects Agency<br>1400 Wilson Blvd.<br>Arlington, Virginia 22209 | | 12. REPORT DATE<br>March 1982<br>13. NUMBER OF PAGES<br>80 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)<br>AF Wright Aeronautical Laboratories (AFWAL/AAAT<br>Wright-Patterson Air Force Base<br>Dayton, OHIO 45433 | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image understanding, scene analysis, image segmentation, image matching, texture analysis, building detection.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes techniques for symbolic matching of aerial images with other images or maps. Aerial images contain natural features and also man-made 3-D structures. Techniques for scene segmentation and description for these matching tasks are presented.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# TABLE OF CONTENTS


| | | Page |
|---|---|---|
| 1.0 | Overview and Summary | 1 |
| 1.1 | Matching High Level Features of an Aerial Image with a Map or Another Image<br>- *Gerard G. Medioni* | 2 |
| 1.2 | Symbolic Matching of Images and Scene Models<br>- *Keith E. Price* | 18 |
| 1.3 | An Edge Based System for Detecting Buildings in Aerial Images<br>- *Andres Huertas* | 41 |
| 1.4 | Segmentation of Images into Regions Using Edge Information<br>- *Gerard G. Medioni* | 65 |
| 1.5 | Using Texture Edge Information in Aerial Image Segmentation<br>- *H.Y. Lee and K.E. Price* | 73 |

# 1.0 OVERVIEW AND SUMMARY

This report summarizes our research under contract F-33615-80-C-1080 during the period of 1 October 1981 to 28 February 1982 and constitutes the final report along with previously published semi-annual progress reports.

During this period, our main concentration has been on improved image to map, (and image to image) correspondence techniques, that work with complex images that include man-made 3-D cultural features. Two such techniques are described in detail in this report. In Sec. 1.1, "Matching High Level Features of an Aerial Image with a Map or Another Image," Medioni describes improvements to a previous segment matching technique and shows results for the DMA supplied Ft. Belvoir Images. In Sec. 1.2, "Symbolic Matching of Images and Scene Models," Price gives new results on matching of image using an improved relaxation labeling technique that uses higher level groupings and is particularly useful when a large number of elements are present.

To support symbolic image to map matching, we have also developed improved segmentation and description techniques. In Sec. 1.3, "An Edge Based system for Detecting Building in Aerial Images," Huertas describes a method to detect building (or building like geometrical structures) from edge segments in high resolution aerial images, without specific a priori map that gives the location of the buildings.

In Sec. 1.4, "Segmentation of Images into Regions Using Edge Information," Medioni presents a relatively simple technique to detect regions from edge segment data with impressive results. In Sec. 1.5, "Using Texture Edge Information in Aerial Image Segmentation," Lee and Price describe a technique for improving segmentation of an image by using texture features.

No hardware development was undertaken as part of the contract during this period.

# 1.1 MATCHING HIGH LEVEL FEATURES OF AN AERIAL IMAGE WITH A MAP OR ANOTHER IMAGE

GERARD G. MEDIONI

## 1. INTRODUCTION

Suppose that we are given a very high resolution aerial picture taken from a known altitude and with a known orientation, together with a detailed map of the area. How can we determine which parts of the picture correspond to given elements of the map? The complexity of this problem stems partly from the fact that a picture is described in terms of pixel intensities while the map is a set of high level abstract entities. There have been several tentative answers to this question. Early systems worked directly with the intensity array [1,2,3], trying to find transformations which map one array into another. Problems arise when the illumination changes substantially or when the texture changes with the seasons.

Price and Faugeras [4,5], extracted linear features and regions to be matched with a map whose characteristics are derived manually. They use relative position constraints and stochastic labeling.

The Hughes Research Laboratories [6,7,8] conducted studies to match two views of a scene using line and vertex features derived from the scene.

We first extract features from the intensity images using the USC linear feature extraction system [9]. The technique consists of convolving the image with 6 directional edge masks, each 5*5 pixels, choosing the maximum, thinning and thresholding the convolved output, linking the resulting edges based on proximity and orientation, and finally approximating by straight lines.

Since we are interested in rivers and roads, we consider a linear feature called APAR (for

antiparallel) which represents two parallel edge segments with a 180 orientation difference. If the scene is to be matched with a map, we encode the linear pieces of the map manually. The problem can now be formulated as: Which elements in the image correspond to the given elements in the map, based on geometrical constraints.

The next section provides assumptions and definitions, the third section describes the kernel method, derived from the relaxation method, the fourth presents results and the conclusion outlines possible extensions.

## 2. ASSUMPTIONS AND DEFINITIONS

We assume that

-The model and the scene have approximately the same orientation.

-The scaling factor from the model to the scene, $\mu$ , is known.

Let us define the following terms:

We will denote the linear features of one image as $a_i$, $1 \leq i \leq n$, and call them objects.

We will denote the linear features of the other image, or of the map, as $\lambda_j$, $1 \leq j \leq m$, and call them labels.

The set $A=\{a_i \mid 1 \leq i \leq n\}$ is the scene.

The set $L=\{\lambda_j \mid 1 \leq j \leq m\}$ is the model.

We are interested in computing the quantity $p(i,j)$ which is the possibility for object $a_i$ to have label $\lambda_j$.

The method presented here principally relies on geometrical constraints, meaning that when we assign a label $\lambda_j$ to an object $a_i$ , we expect to find an object $a_h$ with a label $\lambda_k$ in a certain position depending on $i,j,k$.

This area is denoted $w(i,j,k)$ and is called the window $(i,j,k)$.

For details of window design, please refer to the previous report.

Figure 1 presents an example of such a window.

Finally, we need to define the relation , "is compatible with", between *(i,j)* and *(h,k)* as *(i,j)* IS COMPATIBLE WITH *(h,k)*

$$<==> (i,j) \quad (h,k)$$

$$<==> a_h \text{ in } w(i,j,k) \text{ AND } a_i \text{ in } w(h,k,j) .$$

We need to check both predicates because the relation "is in *w*" is not symmetric, that is $a_h$ in $w(i,j,k)$ does not imply $a_i$ in $w(h,k,j)$.

We now can proceed to explain how the method operates.

## 3. DESCRIPTION OF THE KERNEL METHOD

### 3.1. Brief Overview of the Relaxation Method

Given a set of objects $A=\{ a \mid 1 \leq i \leq n\}$ (scene) and a set of labels $L=\{ \lambda_j \mid 1 \leq j \leq m \}$ (model), we are looking for a subset $M=\{ (a_i,\lambda_j) \mid p(i,j)=1 \}$ of the cartesian product $A*L$ which is the set of objects matched with a label.

Let $M^t$ be the superset of $M$ at the $t^{th}$ iteration.

$$M^t=\{ (a_i,\lambda_j) \mid p^t(i,j)=1 \}$$

If there exists only a partial match, then $M = \phi$.

This situation also occurs if some objects are slightly out of place. Since we are interested in partial matches, we introduce the quantity q and define the iteration formula as follows:

$$p^{t+1}(i,j) = 1 \text{ iff } p^t=1 \text{ and}$$

there exist a subset $S$ of $[1,m]$ with $q$ elements such that

$$\forall\ s \text{ in } S, \exists\ k \text{ in } [1,n] \text{ such that } (i,j)\ C(k,s).$$

$q$ is a measure of the way scene and model agree. Setting $q$ to $m$ means that we know that there is a perfect match "a priori." We will denote the resulting set at the $t^{th}$ iteration as $M^t_q$. The stopping criterion is simply $M^{t+1}_q = M^t_q$

A flow-chart of the procedure is shown in Figure 2.

The main result is that this process converges in a finite number of iterations. For more

detailed analysis, see [10,11]. One example of a successful application is shown in the section describing results.

### 3.2. The Kernel Method

One of the main problems with the method described above is that the number of labels in the model has to be small for the method to be efficient. That leads us to selecting a few labels with no really valid criterion. Let us see how we can improve the method if we know for sure that some pairs $(i,j)$ are in the set $M$.

Let $B$ be a subset of $A$ with $q$ elements.

$$B = \{ b_i \mid 1 \leq i \leq q \} \text{ and } B \quad A.$$

Let $T$ be a subset of $L$ with $q$ elements.

$T = \{ t_i \mid 1 \leq i \leq q \}$ and $T \subseteq L$ such that all pairs $[(b_i,t_i),(b_j,t_j)]$ are pairwise compatible.

Obviously, all these couples are in $M_q$.

Let us call the set of all such couples $K_q$.

Now, a *sufficient condition for any pair* $(a_i,\lambda_j)$ to be in $M_q$ is simply that

either $(a_i,\lambda_j) = (b_k,t_k)$ for some $k$ in $[1..q]$

or $\exists k$ in $[1..q]$ , $(a_i,\lambda_j) \mathcal{C} (b_k,t_k)$

Let us denote this newly obtained set by $N_q$

$$N_q = \{ (a_i,\lambda_j) \mid (a_i,\lambda_j) \text{ in } K_q \text{ or } \forall k \text{ in } [1..q],\ (a_i,\lambda_j) \mathcal{C} (b_k,t_k) \}$$

We can see that $M \subseteq N_q \subseteq M_q$, therefore, the new set is better that the previous one.

### 3.3. Finding the Kernel

One of the problems encountered by the relaxation method is that, having to reduce the number of labels, we had to choose some with no valid criterion. Now, to find a valid kernel, we consider the full model and choose a small number of objects in the scene, that is reverse the role of scene and model. The difference is that all labels look alike, but we can choose objects that have "good looking" attributes, that is, long, isolated, and corresponding to strong

edges. The procedure is then:

- Choose $N$ objects from the scene.
- Match them with the model using the relaxation method with $q < N$.
- Find $q$ matched pairs that verify pairwise compatibility. These $q$ pairs are the kernel.

### 3.4. Discussion

There are some very interesting properties associated with this method:

- We no longer need to limit the size of the model.
- It is a one pass method giving a fast yes/no answer for each object in the scene.
- The map can be replaced by another scene, such as a different view of the same scene.
- Since the method to find the kernel is fast, we can "forget" that we know the relative orientation of scene and model and derive it or refine it.

## 4. RESULTS

These methods have been applied to 2 scenes representing part of the Fort Belvoir Military Reservation in Virginia. The original pictures have been provided by the Defense Mapping Agency and the full resolution images are 2048 * 2048. Figure 3a shows the DMA3 picture at full resolution. Figure 3b shows the DMA2 picture at full resolution. Figure 4 is the part of the map corresponding to the DMA images. As we can see, the original images are very detailed, and in order to segment them, we proceed hierarchically: To find the most prominent features such as large roads and rivers, we use lower resolution images, as shown on figures 5a and 5b that have a resolution of 256 * 256. Now, as explained in the introduction, we extract the edges, thin them and link them to obtain the linear features shown on figures 6a and 6b. As we can see, most small details have vanished. Since we are interested in roads and rivers, we extract the apars (antiparallel lines) with a maximum width of 8 pixels and filter out the very small ones . The resulting scenes are shown in figures 7a and 7b.

To illustrate the relaxation method, we manually generate from the map a model of the main highway, as shown in figure 8, and match it against the scene of figure 7a. The result is shown in figure 9 and is the desired result.

To illustrate the efficiency of the kernel method, we first provide the model as shown in figure 10. This model contains 35 labels. We use a kernel of 4 elements, shown in figure 11, to match the DMA3 scene. The resulting set of matches is shown in figure 12. The processing time was 8.5 seconds, not counting time to compute the kernel. To compare it with the relaxation method, we matched the DMA3 scene with the full model and a value $q = 9$. The result, shown in figure 13, took 750 seconds, and contains more errors.

We also generated a kernel from DMA3, as shown in figure 14, to match DMA3 (scene) with DMA2 (model). DMA2 is a rather complex scene because the original image is very textured, containing many objects; furthermore, long segments are broken into small pieces. However, the method was successful, as shown in figure 15.

## 5. CONCLUSION

This paper demonstrates how a small quantity of "a priori" knowledge can transform a hard problem into a simple one. The "expensive" processing, namely relaxation, is used to find a good match on a small subset of a scene, thus allowing the decision for the other elements of the scene to be simple and fast. This method can be generalized to work on all elements of an image that can be modeled in terms of vectors in a 2-d space; however, it is not suitable to be applied on the edges of the full resolution image (2048 * 2048). We are currently investigating the existence and representation of primitive features with a higher semantic meaning.

## 6. REFERENCES


[1] Azriel Rosenfeld and Avinash C. Kak, Digital Picture Processing, Academic Press, New York, 1976.

[2] B. Widrow, "The 'Rubber-mask' technique," Pattern Recognit. 5, 1973, pp 175-211.

[3] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," IEEE Trans. Comput. C-22, 1973, pp 67-92.

[4] K. E. Price and R. Reddy, "Matching segments of images," IEEE Trans. Pattern Anal. Mach. Intell., vol PAMI-1, Jan. 1979, pp 110-116.

[5] O. D. Faugeras and K. E. Price, "Semantic Description of aerial images using stochastic labeling," IEEE Trans. Pattern Anal. Mach. Intell., vol. F `MI-3,Nov. 1981, pp 633-642.

[6] S. A. Dudani, A. L. Luk, J. P. Stafsudd, C. S. Clark, B. L. Bullock, "Model based scene matching," Hughes Research Report 509, Malibu, 1977.

[7] C. S. Clark, A. L. Luk, and C. A. McNary, "Feature-based scene Analysis and model matching," Proceedings of NATO Advanced Study Institute on Pattern Recognition and Signal Processing, Paris, June 1978.

[8] C. S. Clark, W. O. Eckhardt, C. A. McNary,R. Nevatia, K. E. Olin, and E. M. VanOrden, "High accuracy model matching for scenes containing man-made structures," Proc. of Symposium on Digital Processing of Aerial Images, SPIE, vol. 186, 1979, pp 54-62.

[9] R. Nevatia and K. R. Babu, "Linear feature extraction and description," Computer Graphics and Image Processing, vol. 13, 1980, pp 257-269.

[10] G. G. Medioni, "Matching of a map with an aerial image," USC-IPI Report 1050, Sept. 1981, pp 12-37.

[11] R. M. Haralick and L. G. Shapiro, "The consistent labeling: Part 1," IEEE Trans. Pattern Anal. Mach. Intell., vol. PAMI-1, Apr. 1979, pp 173-184.

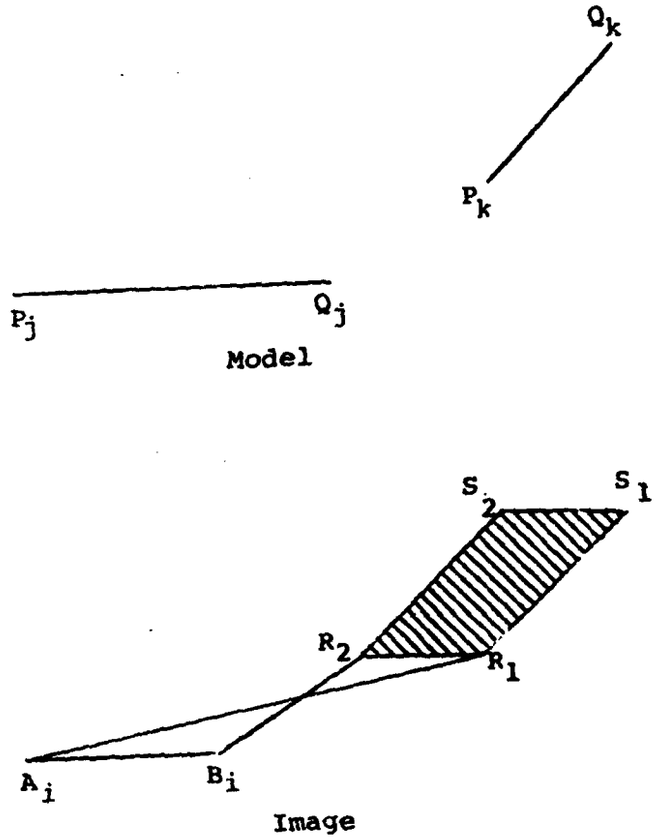



Figure 1. Example of window design

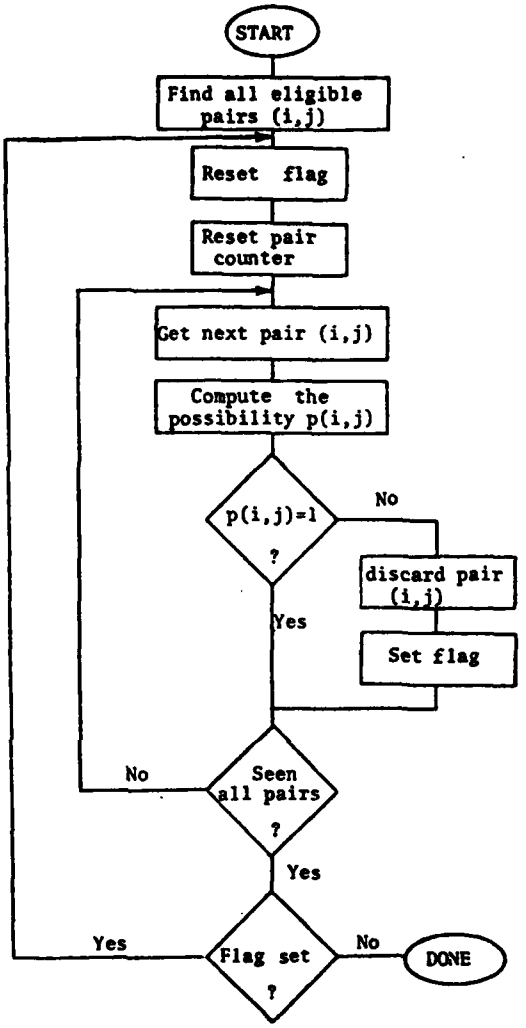



Figure 2. Flow chart of the relaxation method

Best
Available
Copy

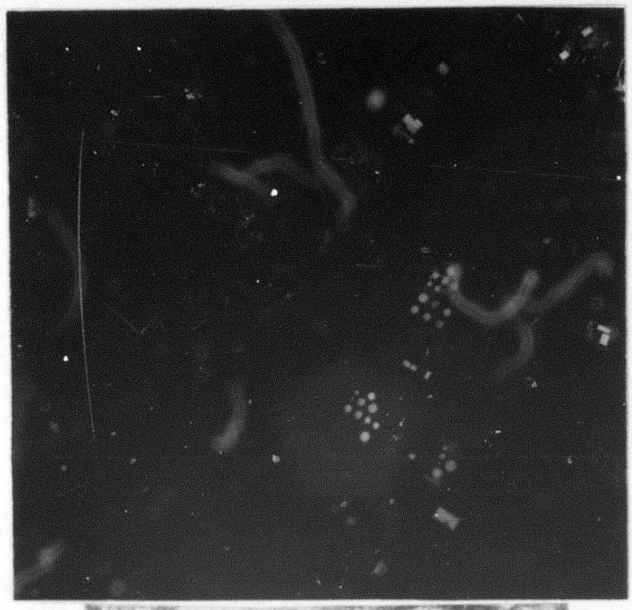


Figure 3a. Full resolution DMA3 image

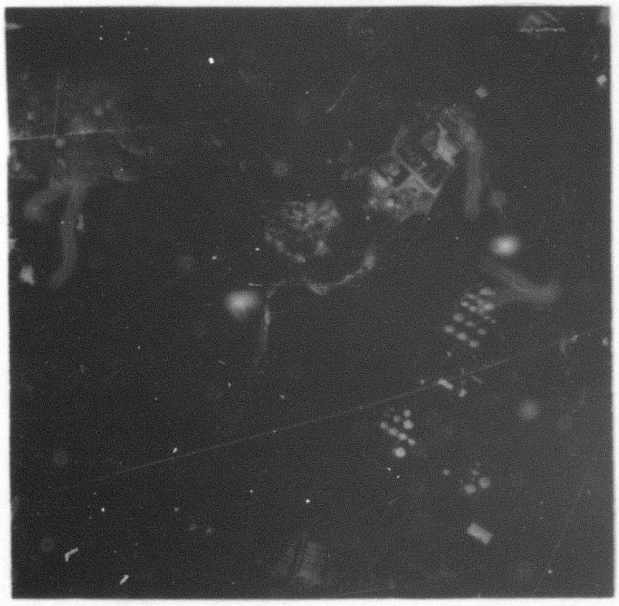


Figure 3b. Full resolution DMA2 image

Figure 4. Map of the area

Figure 5a. DMA3 image 256*256 resolution

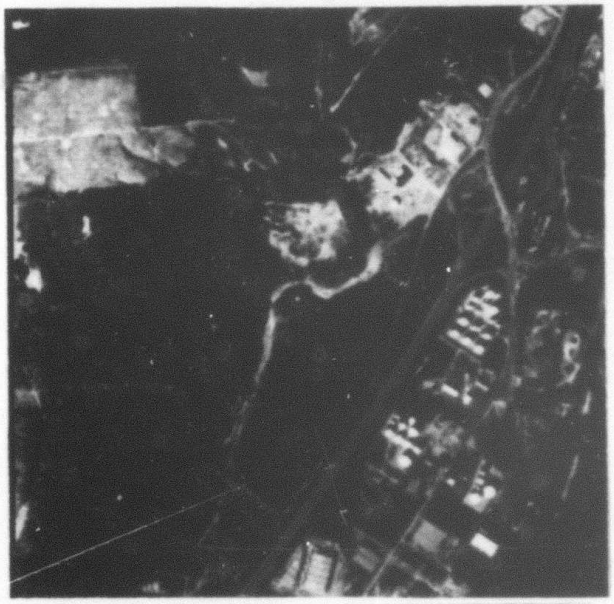


Figure 5b. DMA2 image 256*256 resolution

Figure 6a. Segments of DMA3

Figure 6b. Segments of DMA2

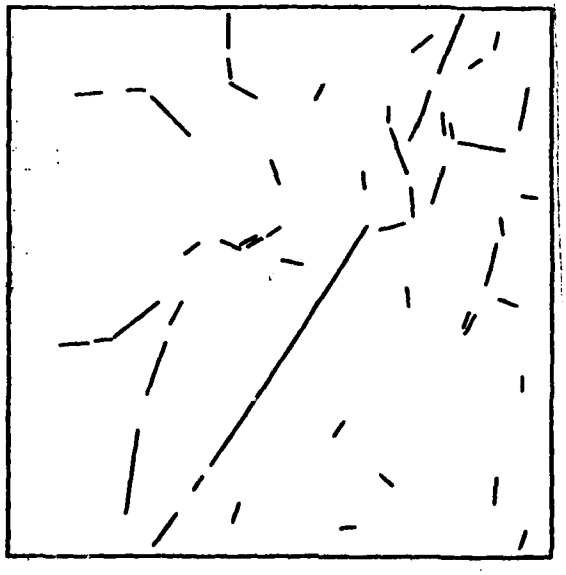


Figure 7a. Apars of DMA3

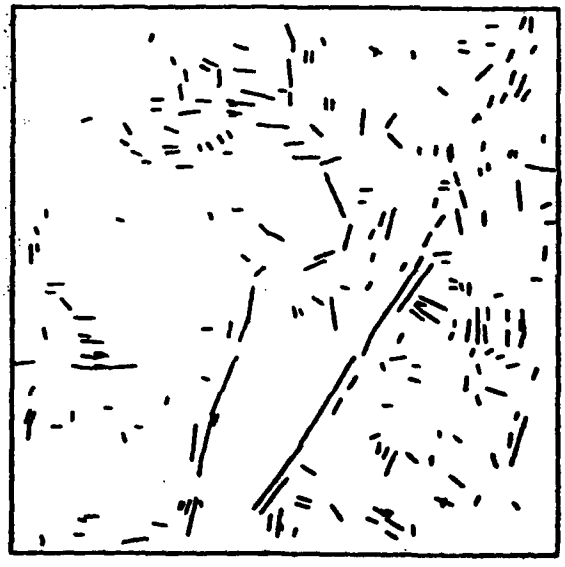


Figure 7b. Apars of DMA2

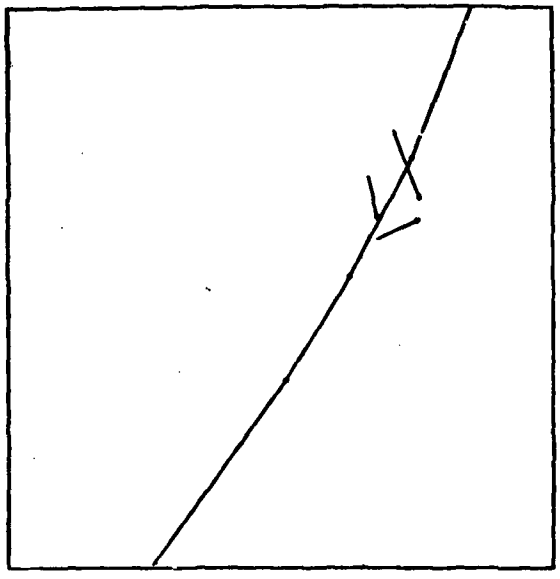


Figure 8. Model of the highway

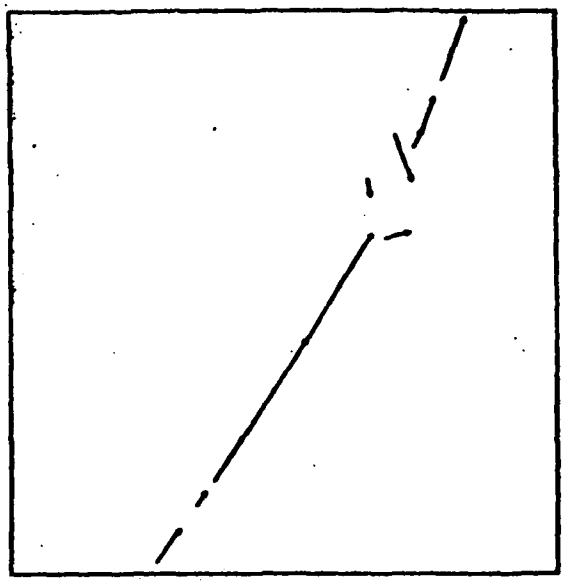


Figure 9. Matched apars from DMA3

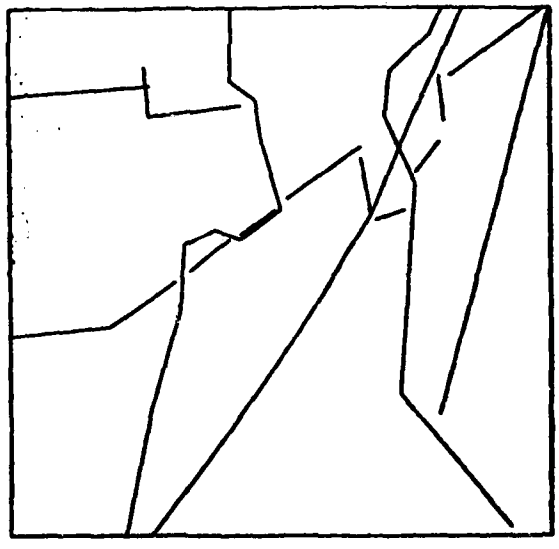


Figure 10. Full model of the area

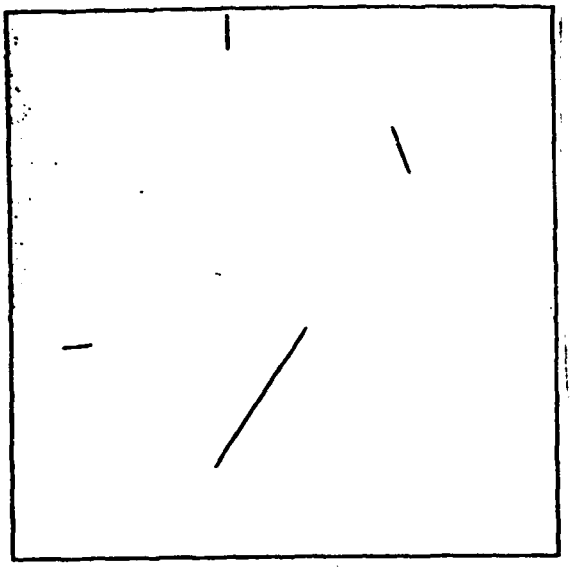


Figure 11. Kernel to match DMA3 with the model

Figure 12. Matched apars of DMA3 using the kernel

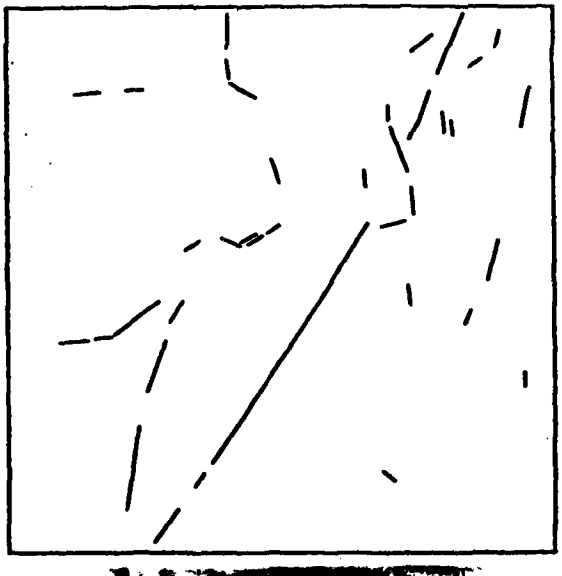


Figure 13. Matched apars of DMA3 using relaxation

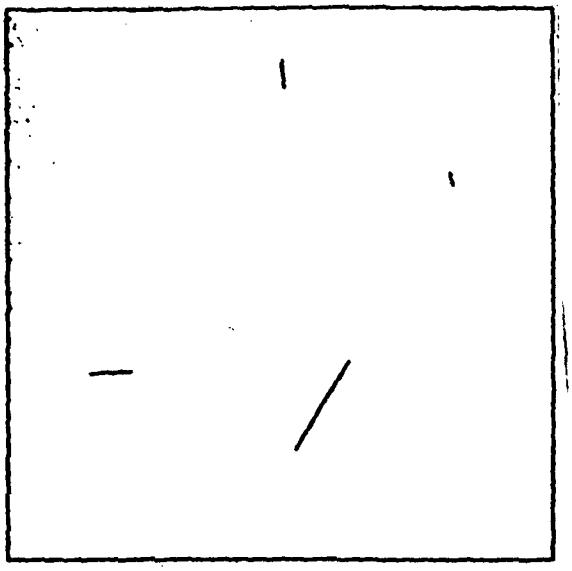


Figure 14. Kernel to match DMA2 with the model

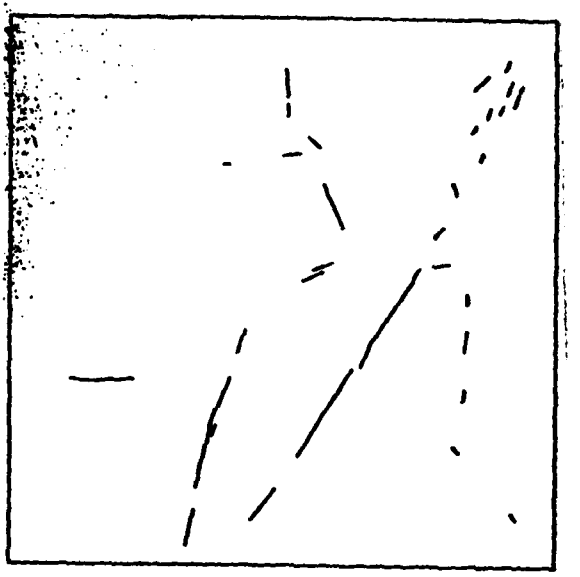


Figure 15. Matched apars of DMA2

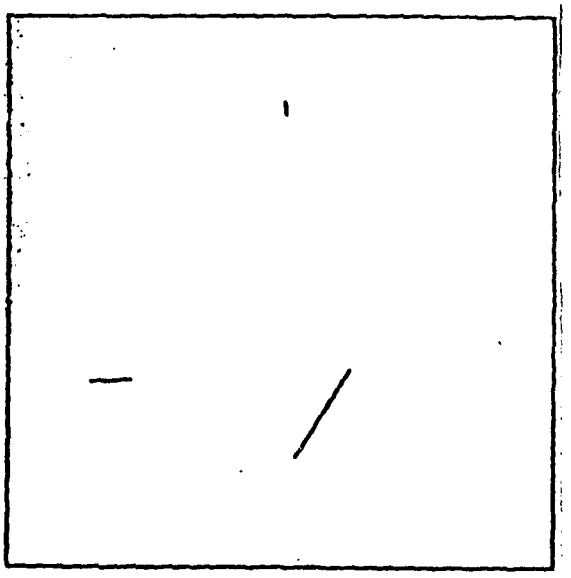


Figure 16. Kernel to match DMA2 with DMA3 as model

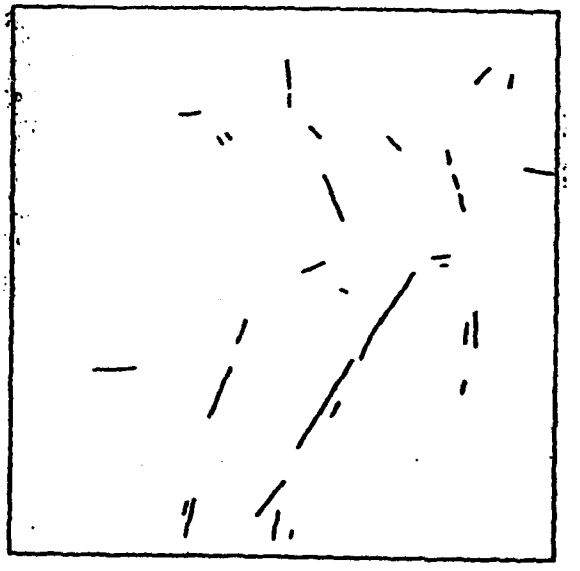


Figure 17. Matched apars of DMA2

# 1.2 SYMBOLIC MATCHING OF IMAGES AND SCENE MODELS

KEITH PRICE

## 1. INTRODUCTION

Matching of images and descriptions has many different uses and can be performed at several different levels. Some matching tasks require that very precise corresponding locations are computed (e.g., stereo depth computation, pixels level change detection). But for many tasks, matching at a grosser level (i.e., finding correspondences between large areas) is best. This paper discusses results of a general symbolic level image matching system applied to the task of matching an image and an a priori description of the scene (a model) to locate objects in the image and the task of matching two images to find the location of an object in two different views. thus we use this program to find correspondences between areas of the images (or objects) rather than to find a pixel level matching between them.

## 2. BACKGROUND

The work reported here represents an extension of earlier relaxation based symbolic matching efforts [1]. A variety of image matching techniques have been developed for different tasks. Moravec [2] has developed a system which locates feature points in one image (essentially corners) and uses a correlation based matching procedure at multiple resolutions to efficiently find a set of corresponding points in the two images. This system is intended for land based robot navigation which uses the three dimensional information from these feature points for navigation. A stereo system developed by Baker [3] generates a complete disparity map starting from edge correspondences which can be used for depth computations if the camera

positions are known. These two (and many other similar efforts) concentrate on precise matching of image data.

Several systems which work on a variety of symbolic representations have also been developed. Barnard and Thompson [4] have developed a relaxation based motion analysis program which finds corresponding feature points in two images. The feature points are similar to those of Moravec [2], but they are located in both images. Wong et al. [5] also use a relaxation procedure to match corners which are detected in pairs of images. This system allows arbitrary translations and rotations of the camera. Clark et al. [6] have developed a system to match line like structures (generally edges or region boundaries). The program uses three initial matching lines to get a mapping between the two images. The quality of the match depends on how well all the other lines match, and the best match is determined by trying all possible triples of matching lines. The number of possible triples is limited by the allowable transformations, i.e., given one match, the possible matches for the other two are very restricted. Gennery [7] extracts objects and uses a tree searching procedure to find the best match.

The relaxation procedure used here is developed more fully in [1, 8] and differs from other methods in its gradient optimization approach. There are several alternative relaxation updating schemes such as the basic method of Rosenfeld et al. [9], Peleg [10], and Hummel and Zucker [11]. We have implemented these other methods and use them for a comparison of the results.

## 3. DESCRIPTION AND MATCHING

This matching system uses feature-based symbolic descriptions for its input. The description of an idealized version of the scene (a model) is developed by the user through an interactive procedure. The image descriptions are derived automatically from input images. The underlying descriptive mechanism is a semantic network. The nodes of the network are the

basic objects with associated feature values and the links indicate the relations between objects.

The basic objects used in the image description are regions or linear features extracted by automatic segmentation procedures [12, 13]. These procedures produce a set of objects composed of connected regions which are homogeneous with respect to some feature in the input image and long narrow objects which differ from the background on both sides and can be represented as sequence of straight line segments. Only the important objects are described in the model. The automatic image segmentation produces many objects which are not included in the model (as many as 100-300 elements). The model description determines the outcome of the matching procedure and can also be used to guide the segmentation procedure [14].

The description is completed by extracting features of the regions and linear objects. The features are those which can be easily computed from the data and which are reasonably consistent. These features include average values of the image parameters (intensity, colors, etc.), size, location, texture, and simple shape measures (length to width ration, fraction of minimum bounding rectangle filled by the object, perimeter$^2$/area, etc.). Relations included in the description are those which are easily computed; such as adjacency, relative position, (north of, east of, etc.), near by, and an explicit indication of not near by.

The basic goal for the matching procedure is to determine which elements in the image correspond to the given objects in the model. Most of the objects cannot be recognized based on features alone. They require contextual information to be accurately located. An important idea used by the matching system is to locate a small set of corresponding objects using feature values and weak contextual information. These initial islands of confidence provide the context needed for finding correspondences for the less well defined objects. Finally, when most objects are assigned, the matching can be done solely on the basis of context, i.e., radical changes in a few objects do not cause the matching program to fail.

The basic operation of the matching system is outlined in Fig. 1. In the large outer loop a

set of possible matching regions is located for every element in the model. Each of these possible assignments has a rating (probability) based on how well the model and image elements correspond. These ratings are refined by the relaxation procedure in the inside loop, until one or more model elements have one highly likely assignment (usually a probability greater than 0.7 or 0.8). At this point a firm assignment is made and the likely assignments are recomputed using these assigned elements to give the context for the match. The inner relaxation procedure updates the probabilities of the assignment based on how compatible the assignment is with the assignments of its neighbors in the graph (i.e., objects linked by relations). We use a variety of relaxation schemes [1, 8, 9, 10, 11, 15] in this loop, with the criteria optimizing method in [1, 8] giving the best results.

**Matching Details**

The quality of match between two elements (one each from the model and image or from two different images) is given by the weighted sum of the magnitude of the feature value differences,

$$R(u,n) = \sum_{t=1}^{m} |V_{ut} - V_{nt}| W_t S_t \qquad (1)$$

where $u$ is an element from the model $n$ from the image, $m$ is the number of features being considered, and $V_{ut}(V_{nt})$ is the value of the $R^{th}$ feature of element $u(n)$. $W_t$ is a normalization weight (the same for all tasks) to equalize the impact from all features. $S_t$ is the task dependent strength of a given feature. These strength values distinguish between important, average, and unimportant features. The ratio of the strength values is 5:1 and there is a fourth strength zero which indicates a feature is not used. This rating function is converted to the range [0, 1] by

$$f(u,n) = \frac{a}{R(u,n)+a} \qquad (2)$$

where $a$ is a constant which controls how steep the differences function is. A value of 1 (a

sharply declining function) produces the best results with the optimization updating approach. Relations are easily included in this scheme. $V_{mt}$ is the number of relations of type $t$ which are specified in the model and $V_{nt}$ is the number which actually occur in the image. Figure 4 illustrates how these values are computed for a given $u_i$. For each possible corresponding region $n_k$, check all $u_j$ (in the model) which are related to $u_i$ to see if the given correspondence ($n_l$) for $u_j$ is properly related to $n_k$. When computing the initial probabilities of a match, only those $u_j$ which have been previously assigned can be considered. The basic compatibility measure is computed when given two potential assignments, therefore rather than using all the other units in the model use only the specified unit $u_j$.

The relaxation procedures require a function which measures the compatibility of a particular assignment $n$ with the current assignments at all neighboring (related) units. This is defined by

$$Q_i(n_k) = \sum_{u_j \text{ in } N_i} Q_{ij}(n_k) + \alpha |N| f(u_i, n_k) p_i(n_k) \tag{3}$$

and

$$Q_{ij}(n_k) = \sum_{n_l \text{ in } W_j} c(u_i, n_k, u_j, n_l) p_j(n_l) \tag{4}$$

Where $N_i$ is the set of objects related to $u_i$, $|N|$ is the number of neighbors, $\alpha$ is a factor between 0 and 1 that adjusts the relative importance of features versus relations (0.1 to 0.25 is the usual range), $p_i(n_k)$ is the current probability for assigning $u_i$ to $n_k$, $W_j$ is the set of likely assignments of $u_j$ (for efficiency and improved results we generally use only the one most likely assignment here). $c(u_i, n_k, u_j, n_l)$ is the same as $f(u_i, n_k)$ except that only relations between $u_i$ and $u_j$ are considered. The vector $\vec{Q}_i$ is normalized to give a probability vector $\vec{q}_i$ which is used by the updating step. The iterative updating is given by

$$\vec{p}_i^{(n+1)} = \vec{p}_i^{(n)} + \rho_n P_i[\vec{q}_i^{(n)}] \quad n = 0,1,2... \tag{5}$$

where $\rho$ is a positive step size to control the convergence speed, $P_i$ is a linear projection operator to maintain the constraint on $\vec{p}_i^{n+1}$ that is is a probability vector, and $g_i^{(n)}$ is an explicit gradient function determined by the criteria to be optimized.

$$g_i(n_k) = -q_i(n_k) - p_i(n_k) N \alpha_i(u_i, n_k)(1 - q_i(n_k))/D_i - \sum_{\substack{u_j \text{ such that} \\ u_i \text{ in } N_j}} 1/D_j \sum_{n_l \text{ in } W_j} c(u_j, n_l, u_i, n_k)(p_j(n_l) - p_j q_j) \tag{6}$$

where

$$D_i = \sum_{k=1}^{m} Q_i(n_k) \tag{7}$$

where $m$ is the number of possible assignments.

## 4. EXTENSIONS TO THE MATCHING

This basic matching procedure is able to adequately perform the match for many tasks, but there are extensions which are required for others. These include extensions to handle multiple levels of descriptions for the scene and those to facilitate the image to image matching process.

### A. Groups

The matching procedure, as described so far, handles relations between two specific elements, if relations among three or more objects are desired they are specified by combinations of binary relations. They may not yield unique matches, but explicit higher order relations are too expensive to compute and use. We extend the matching and description system to include relations between groups of elements. These groups are specified in the model and can be composed of an object or a collection of separate objects that can be more easily related to others as a group. For example, in Fig. 2, the area of San Francisco can be considered as a group composed of the urbanized area, and the park-like areas. The bay,

bridges, and islands can form another group. The separate clusters of storage tanks or buildings could be used to form groups in Fig. 3.

We make several assumptions about the groups of objects. (1) The components of a group are spatially close, not widely scattered through the image. (2) Relations (adjacency, above, etc.) between elements within a group are meaningful, but usually not between individual elements in two separate groups. (3) Relations between groups are consistent and predictable. (4) The feature values for individual objects relative to the averages for the group are well defined (e.g., intensity greater than average, $x$ location in the top fifth). This easily handles structures in aerial images and might be extended to three-dimensional structures possibly with some changes in assumptions.

Thus we simply extend the basic network description of the model to include for each element a pointer to the group, and feature values relative to the group average values, with relations between the groups specified as links between the group nodes. Group features are not available for the image description until the correspondences are located. Initial groupings could be computed in limited cases by creating sets of objects where each is near at least one other member of the set. We could consider groups as descriptions at higher levels in a generalized pyramid structure [16], but our description of the higher level object is based solely on the lower level descriptions of its parts rather than the description of the object at lower resolutions. For a description which encompasses more than two levels, a general multilevel description should be used, but a matching scheme would require a means for linkage between levels.

These group features and relations are incorporated into the matching procedure much the same as the initial features and relations (Eqs. 1-6). But, we apply a second relaxation step in the inner loop (see Fig. 1) using only the group features and relations to compute the compatibility measures. The average feature values and the location of each group are

computed from the current most likely assignment for each of the components of the group (i.e., the top one after the previous relation updating step). Figures 4 and 5 illustrate how group relations enter in compatibilities. As illustrated in Fig. 4 the measure for standard relation is given by whether the relation specified in the model between two elements actually occurs between the two possible assignments. The test for group relations is a bit different. The compatibility measure ($c(u_i, n_k, u_j, n_l)$) can be computed only for $u_i$ in group $G_i$ and $u_j$ in group $G_j$ where $G_i$ is related (is a neighbor in the graph) to $G_j$. The problem is then to determine if $n_k$ is properly related to $G_j$ (e.g., above) and $G_i$ is also related to $n_l$ (above). $R(u, n)$ as given in Eq. 1 is computed in the same manner except all possible second model units ($u_j$) are considered. (Possible in this case means that the two groups are related and $u_j$ is assigned.)

The relations between groups are specified by the model and the test between $n_k$ and $G_j$ must be computed each time since specifications of the group (location, extent, etc.) may change on every iteration. The relations between simple elements in the model should correspond to relations between elements in the segmentation of the image so that these can be computed once and stored. This difference results in an increased computation time for relations between groups compared to relations between basic elements.

**B. Image to Image Matching**

Matching of images at a symbolic level can provide information similar, though not identical, to pixel level image matching. The result is a set of pairs of corresponding objects. From these it is easy to extract global transformations (scale, position, orientation, intensity shifts, etc.), relative displacements (for relative object heights), and local object changes. The matching system is identical to that used for the model to image matching, but there are some differences in how it is used.

Some of the differences are caused by the differences in the nature of the descriptions of images and scene models. The scene model contains only important objects and only those

feature values and relations which are relevant or consistent. The image segmentation cannot be restricted in the same way, thus there are many extra unimportant regions, all feature values are available for all regions and all possible relations between two regions are included in the description.

The increased number of regions is addressed first, rather than trying to find a match for all regions in an image we can restrict the search to those regions which meet a given criterion. We can filter the image to eliminate ill formed regions (using the shape parameters with very loose thresholds), or can restrict the match to some other subset of regions (the brightest half).

The availability of all features is more a benefit than a liability. We can use absolute locations as very strong features, after initial matches are located which can provide the necessary transformations (translation, etc.). By using absolute position, the matching can be performed when differences occur in image segmentations and feature descriptions. Initially the absolute position cannot be used in the matching since we allow arbitrary translations, but when several matches are located we can generate global transformations which will approximately register the images. Because of distortions, height differences, segmentation differences, etc., no global transformation will work perfectly, but the object positions can be used as important features. This is implemented by adding a transformation generation step prior to the determination of initial likelihoods. The transformation is generated using the objects with translations closest to the mean translation. This selection can be done in many ways with different degrees of complexity, we chose a simple method since we do not require subpixel level accuracy in the location transformation. The strengths of the position features are increased from low to medium to high as more correspondences are located. Additionally the number of iterations to try before termination must be reduced when there are few (less than 10) regions remaining to be assigned.

The final change for image to image matching is to perform the match in both directions

independently. This means that when we match two images *A* and *B* we treat *A* as the model and find the correspondences in *B*, then treat *B* as the model and find the correspondence in *A*. The final result is all the pairs of regions which are located in both cases. This eliminates a few correct matches which are found in one case but also eliminates most (all in the examples) incorrect matches since these are predominantly caused by segmentation differences (combined or missing regions).

## 5. RESULTS

We have applied this system to a variety of images (generally two views of each scene, see Fig. 2, 3). For different views of the same scene, we use the same model. The results are presented as overlays on the original images, showing the border of regions or center lines of linear features. The labels are taken from the name given in the model, either the user derived model or the image which serves as a model. Table 1 summarizes the results.

Figure 6 shows the results of matching the model to two images of San Francisco area (Fig. 2). The errors in the second view (Fig. 6b) are caused by the segmentation errors. The two sections of the Bay Bridge are missed by the linear features program and this causes these two to be missed plus the island which is adjacent to the bridges and both portions of the bay is mismatched. (Note that the two sections of the bay were intended to be split by the bridges.) Figure 7 is the same result except that the group features and relations are used. The results are the same except that one section of the bay bridge in view 2 is not matched (which is the correct result) and a second match is found for a park area in view 1. The computation times (with or without group features) are similar.

Figure 8 gives the results for a subwindow of the low altitude aerial images (Fig. 3) without the group information. Figure 9 shows the improvement when group features and relations are used. In the first view 2 fewer mistakes are made. In the second view mistakes

are reduced by 7 and correct matches are increased by 3. Because of the cost of group relations the computation time increased substantially. Different objects are segmented poorly in the two views, but the matching still works well for both. In the seven errors (see Table 1), 3 are objects with no correct match, 3 are multiple matches where the correct match also occurs and one is an extra match to a small nearby region. Figure 10-12 illustrate the image to image matching process. In Fig. 10 the first view A is used as the model, and the second is used in Fig. 11 (the image used as the model is the one on the left). Figure 12 shows those pairs which occur in both cases. Table 2 gives the computed disparities for each of these 31 matched objects.

## 6. SUMMARY AND CONCLUSIONS

This paper presents an extension to an earlier symbolic matching program. The extensions improve the performance of the matching procedure for model to image matching when there are groups or clusters of objects. Additional changes improve the performance of the image to image matching task. The matching results are very good, but not perfect. There is no post processing to eliminate matches which are not consistent with the others which could reduce the errors.

## 7. REFERENCES


[1] O. Faugeras and K. Price, "Semantic Description of Aerial Images Using Stochastic Labeling," *IEEE T-PAMI*, Vol. 3, No. 6, Nov. 1981, pp. 638-642.

[2] H. Moravec, "Rover Visual Obstacle Avoidance," *Proc. 7-IJCAI*, Vancouver, B.C., Canada, Aug. 1981, pp. 785-790.

[3] H. Baker and T. Binford, "Depth from Edge and Intensity Based Stereo," *Proc. 7-IJCAI*, Vancouver, B.C., Canada, Aug. 1981, pp. 631-636.

[4] S. Barnard and W. Thompson, "Disparity Analysis of Images," *IEEE T-PAMI*, Vol. 2. No. 4, July 1980, pp. 333-340.

[5] C. Wong, H. Sun, S. Yada, and A. Rosenfeld, "Some Experiments in Relaxation Image Matching Using Corner Features," Univ. of Maryland, Computer Vision lab, Computer Science Center, TR-1071, 1981.

[6] C. Clark, D. Conti, W. Eckhardt, T. McCulloh, R. Nevatia, and D. Tseng, "Matching of

Natural Terrain Scenes," in *S-ICPR*, Miami, Fla., Dec. 1980, pp. 217-222.

[7] D. Gennery, "A Feature Based Scene Matcher," in *7-IJCAI*, Vancouver, B.C., Canada, Aug. 1981, pp. 667-673.

[8] O. Faugeras and M. Berthod, "Improving Consistency and Reducing Ambiguity in Stochastic Labeling: An Optimization Approach," *IEEE T-PAMI*, Vol. 3, July 1981, pp. 412-424.

[9] A. Rosenfeld, R. Hummel, and S. Zucker, "Scene Labeling by Relaxation Operations," *IEEE T-SMC*, Vol. 6, June 1976, pp. 420-453.

[10] S. Peleg, "A New Probabilistic Relaxation Scheme," *IEEE T-PAMI*, Vol. 2, No. 4, July 1980, pp. 362-369.

[11] R. Hummel and S. Zucker, "On the Foundations of Relaxation Labeling Processes," in *5-ICPR*, Miami, Fla., Dec. 1980, pp. 50-53.

[12] R. Ohlander, K. Price, and R. Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Comp. Graphics and Image Proc.*, Vol. 8, pp. 313-333, 1978.

[13] R. Nevatia and K. Babu, "Linear Feature Extraction and Description," *Comp. Graphics and Image Proc.*, Vol. 13, 1980, pp. 257-269.

[14] K. Price, and G. Medioni, "Segmentation Using Scene Models," to be published.

[15] L. Kitchen, "Relaxation Applied to Matching Quantitative Relational Structures," *IEEE T-SMC*, Vol. 10, Feb. 1980, pp. 96-101.

[16] A. Hanson and E. Riseman, "VISIONS: a Computer System for Interpreting Scenes," in *Computer Vision Systems*, A. Hanson and E. Riseman (eds.), Academic Press, New York, 1978, pp. 303-333.

TABLE 1. SUMMARY OF MATCHING RESULTS.

| Figure | "Model" | "Image" | Correct | Wrong | Iterations | Time | |
|---|---|---|---|---|---|---|---|
| 8 left | Model | View 1 | 35 | 5 | 27 | 1:25 | |
| 8 right | " | View 2 | 32 | 11 | 23 | 1:18 | |
| 9 left | " | View 1 | 35 | 3 | 28 | 3:05 | Group |
| 9 right | " | View 2 | 35 | 4 | 13 | 1:46 | Group |
| 10 | View 1 | View 2 | 34 | 6 | 23 | 1:19 | |
| 11 | View 2 | View 1 | 37 | 3 | 25 | 1:25 | |
| 12 | Combine 1, 2 | | 31 | - | - | - | |
| 6a | Model | View 1 | 15 | 0 | 8 | :29 | |
| 6b | " | View 2 | 12 | 3 | 8 | :36 | |
| 7a | " | View 1 | 16 | 0 | 8 | :34 | Group |
| 7b | " | View 2 | 12 | 2 | 8 | :37 | Group |

TABLE 2. TRANSLATIONS COMPUTED FROM MATCHING REGIONS. THESE ARE GROUPED BY THE CLUSTERS (TOP TO BOTTOM) TO SHOW SIMILARITIES AMONG NEARBY OBJECTS.

| Region ID | ΔR | ΔC | Which cluster | |
|---|---|---|---|---|
| 8 | 91.0 | 195.7 | 1 | |
| 21 | 89.4 | 197.6 | 1 | |
| 22 | 90.0 | 198.2 | 1 | |
| 32 | 89.6 | 196.4 | 1 | |
| 31 | 89.7 | 196.0 | 1 | |
| 10 | 90.4 | 197.8 | 1 | |
| 18 | 89.9 | 198.8 | 1 | |
| 26 | 88.7 | 195.9 | 1 | |
| 23 | 89.6 | 197.4 | 1 | |
| 9 | 87.5 | 198.1 | 1 | |
| 17 | 89.5 | 194.2 | 1 | |
| Group range | 3.5 | 4.6 | | |
| 35 | 84.2 | 193.7 | 2 | |
| 30 | 85.7 | 198.6 | 2 | |
| 3 | 77.0 | 187.7 | 2 | Region broken in half |
| 12 | 83.7 | 198.7 | 2 | |
| Group range | 2 | 5 | | |
| 36 | 79.7 | 200.6 | 3 | |
| 11 | 80.5 | 201.3 | 3 | |
| 39 | 81.9 | 200.7 | 3 | |
| 25 | 80.7 | 201.0 | 3 | |
| 7 | 79.7 | 199.9 | 3 | |
| 19 | 79.6 | 202.3 | 3 | |
| 20 | 81.4 | 199.4 | 3 | |
| 13 | 78.2 | 202.8 | 3 | |
| 14 | 77.2 | 200.0 | 3 | |
| 4 | 77,6 | 202.1 | 3 | |
| Group range | 4.7 | 3.4 | | |
| 29 | 77.6 | 193.6 | 4 | |
| 15 | 77.2 | 193.8 | 4 | |
| 16 | 79.1 | 198.2 | 4 | |
| 34 | 76.4 | 194.1 | 4 | |
| 2 | 77.4 | 194.0 | 4 | |
| 5 | 76.5 | 193.8 | 4 | |
| Group range | 2.7 | 4.6 | | |
| Overall range | 14.6 | 9.2 | | |

TABLE 3. COMPARISON WITH OTHER RELAXATION TECHNIQUES. COMPLETE IS THE OPTIMIZATION APPROACH [1, 8]. PROJECT ONLY USES THE PROJECTION FUNCTION BUT NOT OPTIMIZATION. PRODUCT COMBINES THE INDIVIDUAL MATCHING VALUES USING PRODUCE RATHER THAN SUM (AS IN EQ. 4). ORIGINAL IS FROM [9] AND IS GIVEN FOR HISTORICAL PURPOSES.

| Scene | Relaxation type | Right | Wrong | Iterations | Time | |
|---|---|---|---|---|---|---|
| Top group of 14 | Complete | 14 | 0 | 10 | :07 | |
| | Project only | 14 | 0 | 11 | :07 | |
| | Product combination | 14 | 3 | 4 | :02 | |
| | Original updating | 14 | 3 | 13 | :09 | |
| High altitude model with view 1 | Complete | 15 | 0 | 8 | :29 | |
| | Project only | 7 | 0 | 5 | :21 | |
| | Product | 18 | 4 | 11 | :27 | (14 on iteration 1) |
| | Original | 14 | 8 | 15 | time limit | |
| Low altitude model with view 2 | Complete | 35 | 4 | 13 | 1:46 | |
| | Project only | 32 | 2 | 11 | 1:20 | |
| | Product | 33 | 12 | 4 | :09 | (30-6 on iteration 1) |
| | Original | 38 | 8 | 11 | 1:45 | |
| Low altitude view 1 with view 2 | Complete | 34 | 6 | 23 | 1:19 | |
| | Project | 24 | 3 | 21 | 1:13 | |
| | Product | 35 | 43 | 20 | 0:40 | Very quick decisions |
| View 2 with view 1 | Complete | 37 | 3 | 25 | 1:25 | |
| | Project | 23 | 2 | 14 | 0:55 | |
| | Product | 36 | 33 | 12 | 0:23 | |

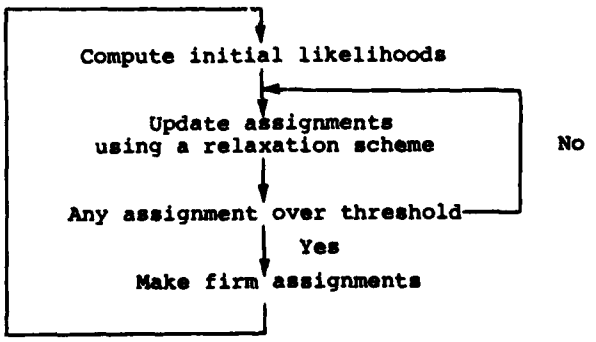



Figure 1. Overview of symbolic matching system

Best
Available
Copy

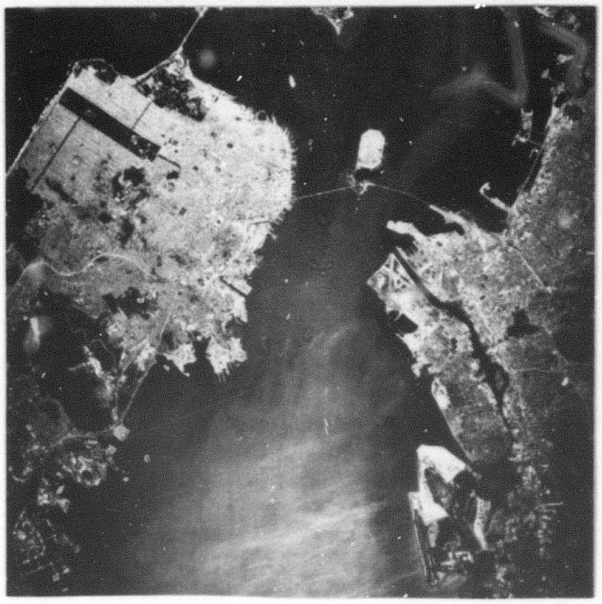
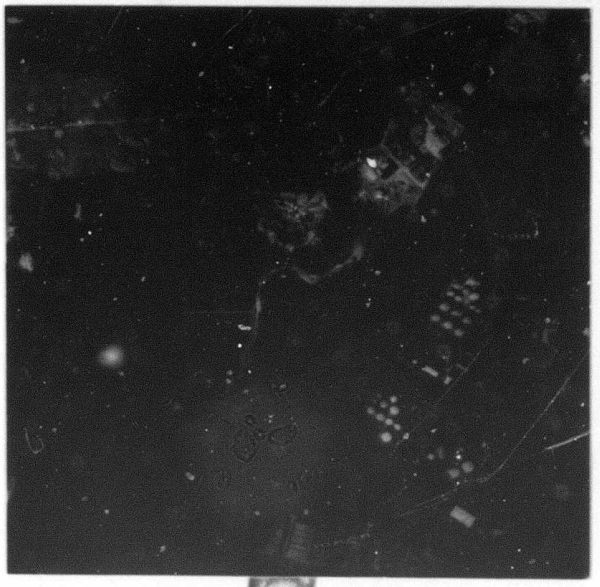
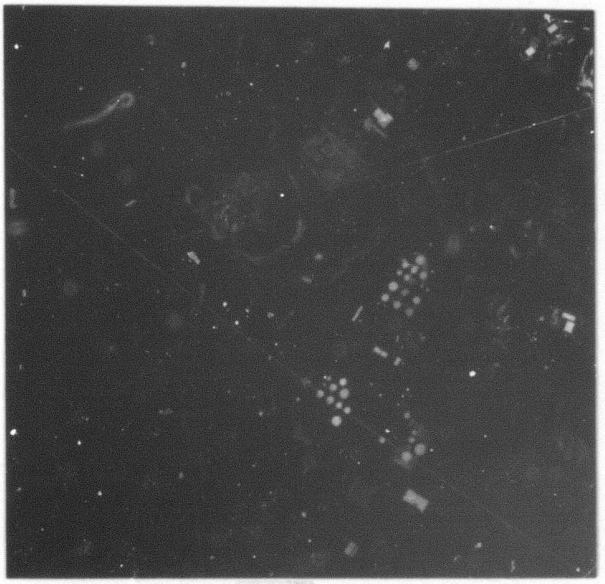


Figure 2. High altitude view of San Francisco area



(a)



(b)

Figure 3. Low altitude aerial image
(a) view 1 (October),
(b) view 2 (August)

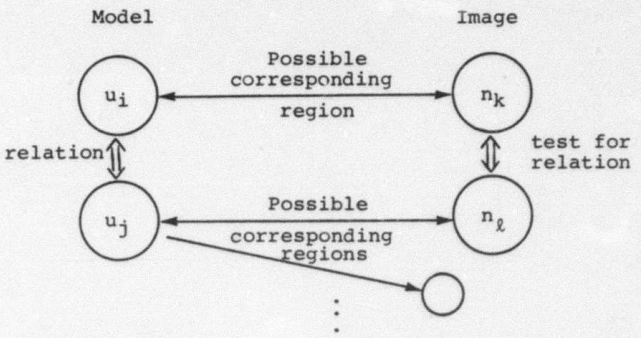



Figure 4. Use of relations in compatibility computation

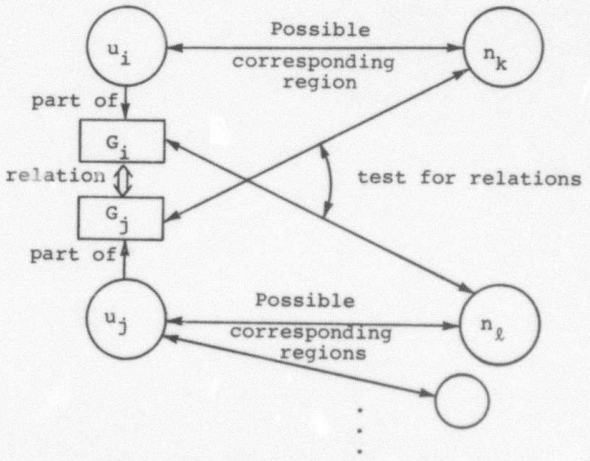
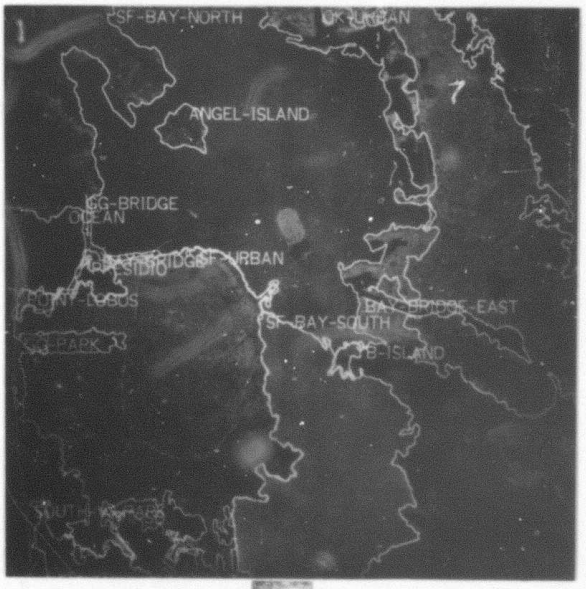



Figure 5. Use of relations between groups in compatibility computation

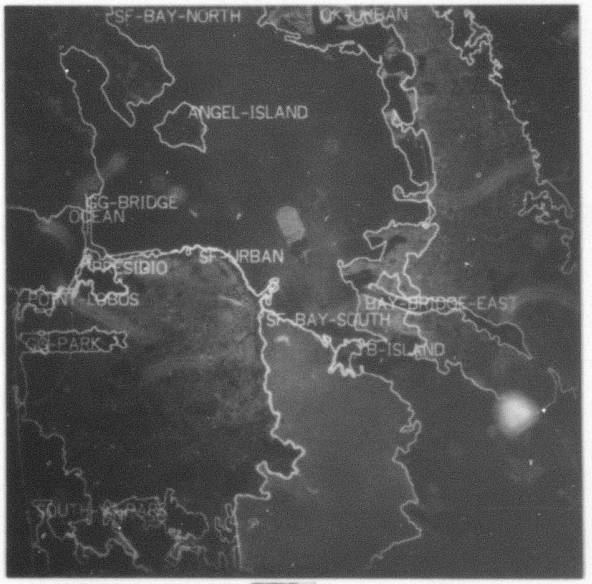

(a)




(b)

Figure 6. Matching results without group features
(a) view 1, (b) view 2


(a)




(b)

Figure 7. Matching results using group features
(a) view 1, (b) view 2

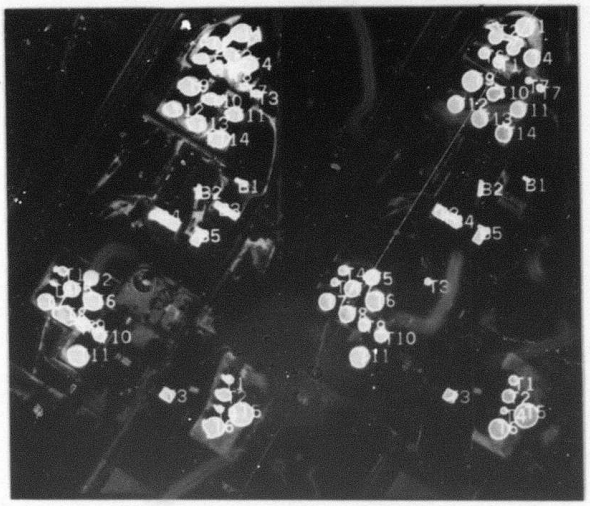


Figure 8. Low altitude model image matching without group information
Left: view 1, Right: view 2

Figure 9. Low altitude model to image matching using group features
Left: view 1, Right: view 2

Figure 10. Low altitude image to image (view 1 used as model) matching
Left: view 1, Right: view 2

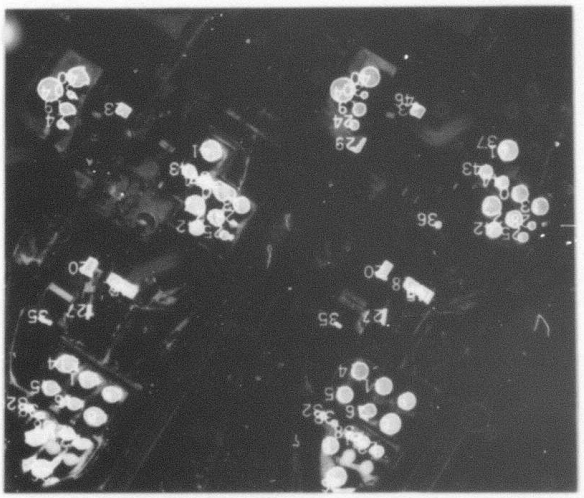


Figure 11. Image to image (view 2 used as model)
Left: view 2, Right: view 1

Figure 12. Image to image matching, the combined results of Fig. 10 and Fig. 11
Left: view 1, Right: view 2

# 1.3 AN EDGE BASED SYSTEM FOR DETECTING BUILDINGS IN AERIAL IMAGES

ANDRES HUERTAS

## 1. INTRODUCTION

The detection of man -made objects in aerial images is a difficult task if only edge or region information is available. In this section we discuss a method that uses the line segments approximating the intensity edges in the image and the intensity data to support interpretations of the image edges.

The appearance of most buildings in aerial images is highly geometric in nature and in that sense "contrasts" against the natural appearance of the surrounding objects such as forests and lakes. Human observers rely on clues such as shadows and nearby roads, but perhaps the most important visual clue are the observed geometric features formed by building sides [1]. This suggests a hierarchy of increasingly complex features from edges to line segments to simple geometric features to geometric regions, that could be interpreted respectively as physical, illumination or reflectance boundaries and corners and, ultimately, as objects projected on the ground surface. These interpretations are obtained by applying geometric, algebraic, illumination, and reflectance constraints imposed on the interpretations of observed edges, groups of edges, and the regions they surround.

Previous work on the interpretation of geometric structure has concentrated on the constraints imposed on the boundary junctions by certain geometric objects ([2], [3], [4]). Binford and Lowe [5] discuss the derivation, use and implementation of more general constraints on the interpretation of image curves. These contraints are derived from general assumptions regarding illumination, object geometry, and the imaging process [5], to carry out

geometric interpretations up to the volumetric level. [redacted] has been tested on simulated image curve data derived by hand with encouraging results. Our system uses edge information derived automatically and, for improved performance, we use a priori information that can be easily available such as direction of illumination.

Previous work on object detection in aerial images by Nagao, et al. follows a region-based approach [6] to perform an elaborate structural analysis of the image. Rosenfeld and Tavakoli [7] use estimates of an ideal building gray level to evaluate edge segments as a building sides. Feature probabilities are then computed to eliminate non-building segments. The remaining few segments are pair-wise linked and grouped according to gray level, and geometric compatibilities (to differentiate them from road segments with similar gray levels). Building segments are extracted by finding closed and semiclosed groups of antiparallel pairs [8] surrounding a bright uniform region or, groups of isolated pairs of antiparallel segments for which lines drawn between the midpoints of the pair components intersect between the pairs. With this approach, important edge information may be lost in the filtering step and, edges may be misinterpreted. In many cases, edges missing or unseen, due to the angle of illumination for example, prevent the detection of antiparallels, or result in groupings that include segments from nearby objects. Since no shadow information was used, nonbuilding groups may be extracted as well.

In our method we rely heavily on two facts: buildings are three dimensional objects which cast shadows (for nonvertical sun angles) and, buildings have a definite geometric appearance. These facts are invariant to illumination conditions, and therefore we use them to extract the most obvious and strongest information first: the geometric features formed by two or more edge segments that can be interpreted as object or shadow boundaries, on the basis of constraints imposed on them by assumptions on building geometry and the direction of illumination. At all times we keep all the edge and intensity information available, and we make extensive use of shadow information. We do not compute probabilistic measures to make absolute decisions since these measures tend to be image dependent. Instead, we progress hierarchically from

simple geometric features towards building features allowing the increasing evidence to reinforce or disprove previously made hypotheses about the edge segments.

Our current edge detection techniques ([8], [9]) provide fairly good linear features to work with, and in some cases, the information available allows to hypothesize missing or unseen elements. A set of derived constraints suitable for detecting buildings that can be viewed as forming box-like objects has been programmed in SAIL and the results obtained for several images are presented and discussed.

## 2. FROM EDGES TO BUILDINGS

Our primary assumption is that aerial images are orthographic projections of the objects on the surface. In most cases we can expect that the boundaries of a projected building have a regular geometric shape such as a rectangle or a combination of rectangles. These geometric properties are combined with properties that the observed geometric regions of interest should have in order to form groups of edges that can be said to correspond to the boundaries of buildings in the scene.

Five levels of features are considered: pixels, edges, simple geometric features, geometric regions, and buildings. In the current implementation the simple geometric features considered are the 90 degree L junctions (corners) formed by pairs of edge segments approximating the intensity edges, and the geometric regions are those surrounded by groups of segments (boxes) forming corners.

The features in a given level are composed from features in the previous level and may consist of disjoint sets (object boundaries and shadow boundaries are both composed of edge segments). The processing associated with the first two levels (edge points and segments) is non-purposive. The rest is purposive, reducing the amount of search required, therefore increasing speed, as the system progresses towards the higher levels in spite of the increased complexity in the representation of the features.

**First Level: Image Intensity Data.** We have worked with portions of high resolution digitized aerial images provided by the Defense Mapping Agency (DMA). At this level we first extract and link the intensity edge points using our previously described techniques [8], with Laplacian-Gaussian masks for local edge detection [9]. In addition, this level makes the intensity data available for pixel measurements required to verify region properties at higher levels. These measurements are performed by placing small windows over the intensity data to gather simple statistics in the neighborhood of a feature of interest. Our basic assumption is that small windows are sufficient to provide evidence in support of the interpretations made about the observed geometric features.

**Second Level: Edge Segments.** The detected edges are fit into directed (the bright side is to the right of the edge) straight edge segments using our Linear Feature Extraction System [8]. The purpose of this level is to maintain a record for each edge segment incluing its endpoints, length, and other attributes of interest.

Fairly long edge segments can be the result of a physical edge in the image, an illumination boundary as in the case of a cast shadow, or a reflectance boundary as in the case of markings painted on a surface. Segments corresponding to physical edges might correspond to boundaries of objects such as buildings, roads, canals and cultivated fields. The basic premise for their interpretation lies in the geometric and illumination constraints imposed on these objects giving rise to distinguishable properties. We make the following assumptions about individual edge segments:

a) Long straight edge segments correspond to long continuous straight edges in the scene unless a curved physical edge coincidentally lies in a plane aligned with the observer. A constant contrast across the edge is likely to indicate the boundary between the side of an object and its shadow if the source of illumination is on the bright side of the edge. Straight shadow edges must have been cast by a straight physical edge and vice versa if the physical edge lies between the shadow edge and the source of illumination.

b) Segments terminating at a long segment form a *T* or a double *T* (*dT*) junction (A *dT* junction is a two-stemmed *T* junction, in which the stem segments are parallel, have opposite directions and correspond to the sides of a thin long region). In the first case, the top of the *T* is likely to be a physical edge occluding a shadow or, the side of an object, coplanar with the observer, where two non-coplanar surfaces or surfaces with different reflectance end (see figure 1a). The stem of the *T* is likely to be the boundary between the two surfaces, in particular if there is a break in the shadow cast. In the second case, the stem of the *dT* is likely to be a marking occluded by the top of the *dT*.

**Third Level: Corners**. At this level we detect and maintain a record of the corners formed by pairs of edge segments and determine an initial interpretation for their segment components. Corners are the result of three occurrences:

a) There is a break in the image edge due to a sharp physical corner (*L* junction). The detected corners are important, and in particular those formed by long edge segments and surrounding uniform regions. Being localized and oriented, they allow strong assumptions and hypotheses to be made about the corner components. The edge segment components of a hypothesized shadow corner (its segment components surround a dark uniform region, possibly a shadow, and the corner bisector is parallel to the sun rays and directed towards the projected source of illumination) must have been cast by the components of a physical corner located between the shadow corner and the source of illumination. Hence, if the direction of illumination is known, some physical corners and boundaries are strongly constrained to cast shadows in a unique direction. Others are strongly constrained not to cast shadows at all.

b) The image edge is curved. If the two edge segments are nearly collinear there is a link point. If one of the segments is a physical edge, the other one is also a physical edge if the two segments have the similar direction.

c) There is a split *T* (*sT*) or a double split *T* (*dsT*) junction. At a *sT* junction three edges meet, two of which are collinear. At a *dsT* junction four edges meet, two of which are collinear

and the other two form a thin antiparallel pair perpendicular to the collinear edges (see figure 1b):

c.1) At a *sT* junction three regions meet at a point. If the collinears have opposite directions, away from the junction point, the stem component must be directed towards the junction point (since the bright side of the stem is to its right) and the collinear segments are likely to have different interpretations: the edge between the two darkest regions is possibly a physical shadow casting edge, and the edge between the two brightest regions is possibly a reflectance boundary.

If the orientation of the collinear segments is along the projected sun rays, the stem component and the collinear segment directed towards the projected source of illumination and away from the link point are likely to correspond to a physical corner, and the remaining collinear segment to a shadow boundary (see figure 1b). A similar situation is found when the collinear segments are directed towards the junction point since the stem must be directed away from the junction point.

c.2) If the two collinear segments have the same direction, the segment directed towards the junction point will form a corner with a stem that is directed away from the junction point. Alternatively, the collinear directed away from the junction point will form a corner with a stem directed towards the junction point. Both corner components are likely to have the same interpretation.

c.3) At a double split *T* junction, the stem components are likely to be part of a surface marking, an alley between two buildings, or a line painted across and to the edge of an object. In either case, the collinear segments are likely to have the same interpretation and the stem components are also likely to have the same interpretation.

The initial interpretations of the segments forming corners are based on illumination constraints imposed by the direction of illumination and building geometry (see (a) above and

figure 2). A corner is bright (dark) if its edge components surround a bright (dark) region. The color and disposition of each corner and its components with respect to the projected direction of illumination is very likely to indicate whether the corner components correspond to physical or illumination boundaries. In some cases, readily available shadow edge information is used to match object to shadow corners to allow strong hypotheses to be formed.

Corners $C2$ and $C4$ in figure 2, for example, have their bisectors parallel and directed towards the projected source of illumination. Their edge components have a constant contrast along the edges. Pixel measurements indicate the presence of a dark region outside the corners. Furthermore, there exist corners $C7$ and $C9$, dark, along the direction of illumination with bisectors parallel and directed towards the projected source of illumination. Since the bright corners are located between the dark corners and the projected source of illumination, $C2$ matches $C7$ and $C4$ matches $C9$ and, we can interpret the components of $C2$ and $C4$ as possible building sides and those of $C7$ and $C9$ as shadow boundaries.

Notice also in figure 2 that corners $C6$ and $C8$ are both bright and have the same disposition with respect to the position of the light source. In addition to pixel measurements, the existence of $C3$, matching $C6$ determines the interpretation for the edge components of $C8$ (and $C3$). Similarly, pixel measurements and the nonexistence of a matching corner for $C6$ determine the interpretation for its components.

In the absence of shadow boundary information, the width of a shadow is computed later by analyzing the region adjacent to hypothesized object segments that are constrained to cast a shadow. The method consists of growing several one-pixel wide slices over the shadow, starting at the edge of the object, and in the direction of illumination. The slices grow linearly until the shadow boundary is found. The average of the lengths of the slices is taken as the width of the shadow. Shadow width measures obtained by measuring the distance between matched object-shadow corners or by growing slices, together with the angle formed by the sun rays with a normal to the ground surface, allow the height of the buildings to be hypothesized (see figure

3).

There are cases in which coincidences result in corner segments being misinterpreted or in a corner having more than one valid interpretation. Dark roads surrounding a bright region could be misinterpreted as shadows when analyzing a corner locally or physical edges might be occluding illumination boundaries (see figure 4). Misinterpretations are carried over to the next level where a more global analysis is performed.

**Fourth Level: Geometric regions.** The purpose at this level is to form consistent disjoint sets of corners called boxes which satisfy definite shape constraints. In the current implementation, a box is defined as the boundary of a geometric region whose sides form 90 degree corners. In addition, the surrounded regions must have a uniform gray level. The resulting set of boxes are hypotheses that the grouped edge segments are the boundaries of box-like objects in the input image.

Two geometric constraints are used at this level for shape analysis:

a) Corner consistency (*CC*), defined as the number of concave (bright) corners minus the number of convex (dark) corners. *CC* is always equal to four if the surrounded region is brighter than its background or equal to minus four if the surrounded region is darker than its background. This is due to the fact that the simplest regular shape of interest, the rectangle, has four vertices and four sides. Any additional pair of sides added to this basic shape to form more complex figures adds one convex vertex plus one concave vertex.

b) Side consistency (*SC*), based on the fact that box sides can have only two orthogonal orientations (the corresponding segments can have two possible directions for each orientation). Hence, for each orientation the sum of the lengths of the segments in one direction must be the same as the sum of the lengths of the segments along the same orientation but with opposite direction.

When forming box hypotheses, the most obvious information is processed first: two

corners sharing a component (strongly compatible) are very likely to be in the same box; two corners whose bisectors are orthogonal or parallel (simply compatible) are likely to be in the same box; pairs of corners whose bisectors are neither parallel nor orthogonal or distant from each other (incompatible) are very likely to be in different boxes; corners having short components which are compatible only with themselves can be discarded. The basic assumption is that all the elements contributing to a box are within a circular window, called the compatibility window, whose size is equivalent to that of the largest building of interest in the image. In our experiments we have used a window with a 70 pixel radius.

Strong compatibility among several corners is a necessary and sufficient condition to set up a box hypothesis including them. Simple compatibility is necessary but not sufficient. Isolated corners with long edge components previously matched to its corresponding shadow strongly suggests a three dimensional object, which is sufficient although not necessary to set up a box hypothesis. Using this criteria the disjoint subsets of corners are initially set up to contain only corners that are strongly compatible, and single element subsets for simply compatible, matched or non-matched corners.

For each box hypothesis, *CC* and *SC* roughly indicate how complete a box is, or alternatively, how many elements, corners, and sides are needed to have a complete box. A box is complete if its corner and segment consistency constraints aresatisfied. Whenever a box is or becomes complete a building hypothesis is formed for later validation and description. Otherwise, tasks are activated to search for nearby evidence to geometrically match compatible corners and to reconstruct incomplete boxes. These tasks update the shape consistency ratings and regroup the subsets of corners into stronger box hypotheses:

a) Searching for nearby evidence of incomplete boxes is achieved by performing a trace of the boundary in the direction of the segments in the boundary, and if necessary, in the opposite direction, too. During the trace the geometric constraints for corner detection are relaxed, and as a result, new corners and link points are detected (see figure 5).

b) The corners in the boxes that have been traced but remain incomplete are matched to compatible corners in other boxes. We take each corner in a box and predict the likely position and attributes of a matching corner (see figure 6). Next, all corners within the compatibility window centered at the predicting corner are matched against the predicted corner. A successful match is based on the disposition of the corners, the interpretation given to the corner components, and the attributes of the region between them. A successful match results in the merging of the boxes including the matching corners.

c) Boxes with one or two missing sides facing the source of illumination are reconstructed if the resulting complete box satisfies *CC* and *SC* (see figure 7).

The output of the system is the validated box hypotheses that are candidates for buildings in the scene.

## 3. RESULTS

Figures 8, 9, 10, and 11 show the results obtained for four scenes from a high resolution image of the Fort Belvoir area:

Figure 8(a) shows two buildings, (b) the edge segments extracted, (c) the corners detected, and (d) the boxes found. Two boxes are found. One of them is nearly closed and easily detected. The other remains partitioned into two sets after searching for nearby evidence. The box is only formed after a mutual match is found between a corner in one partition and the predicted corner for a corner in the other partition. The partitions are merged after the smoothness of the region between them is verified. Although there is no object corner-shadow corner matches for either box pixel measurements indicate that both objects cast shadows. The width of their shadows is obtained with the slice method. The following is a description of the buildings found:

| BUILDING: 1 | | BUILDING: 2 | |
|---|---|---|---|
| Perimeter: | 131.5824 | Perimeter: | 109.9089 |
| Area: | 832.2404 | Area: | 875.3179 |
| Orientation: | 250.8210 | Orientation: | 158.4986 |
| Centroid: | <81,111> | Centroid: | <126,108> |
| Corners: | 4 | Corners: | 4 |
| Height: | 5.775503 | Height: | 9.2673054 |

Figure 9(a) shows two main buildings, (b) the extracted edge segments, (c) the detected corners, and (d) the boxes found. Two boxes are found. One of them is closed and the other is nearly closed and consistent. Both objects cast shadows. The width of the shadows is obtained with the slice method. The following is a description of the buildings found:

| BUILDING: 1 | | BUILDING: 2 | |
|---|---|---|---|
| Perimeter: | 122.4878 | Perimeter: | 114.4059 |
| Area: | 888.4081 | Area: | 816.0882 |
| Orientation: | 129.6107 | Orientation: | 306.8699 |
| Centroid: | <60,138> | Centroid: | <142,68> |
| Corners: | 4 | Corners: | 4 |
| Height: | 4.426352 | Height: | 5.581053 |

Figure 10(a) shows six buildings, (b) the extracted edge segments, (c) the detected corners, and (d) the boxes found. Six boxes are found. The leftmost box includes a corner that is matched to a shadow corner, confirming a three dimensional object. This strong evidence supports the decision to reconstruct its missing side, facing the source of illumination. The side is reconstructed after being unable to extract it from sufficient nearby evidence. The box next to it is completed with existing nearby evidence. The two top most boxes are assumed to be missing one side each and are reconstructed. The width of their shadows is computed by the slice method. The lower box is the result of merging two partitions after searching for nearby evidence fails to complete the box. The box on the right is the result of reconstruction. The box on the lower left is too small to be a building. The evidence on the top right is conflicting and requires further analysis. Verification and measurement of the width of the shadows cast is also made with the slice method. The following is a description of the buildings found:

| BUILDING: 1 | | BUILDING: 2 | |
|---|---|---|---|
| Perimeter: | 99.86020 | Perimeter: | 72.46572 |
| Area: | 601.0075 | Area: | 312.4100 |
| Orientation: | 57.38077 | Orientation: | 354.8056 |
| Centroid: | <93,35> | Centroid: | <50,107> |
| Corners: | 4 | Corners: | 4 |
| Height: | 5.196152 | Height: | 5.965953 |

| BUILDING: 3 | | BUILDING: 4 | |
|---|---|---|---|
| Perimeter: | 54.33446 | Perimeter: | 105.9317 |
| Area: | 154.0292 | Area: | 585.9832 |
| Orientation: | 173.9910 | Orientation: | 329.0000 |
| Centroid: | <74,101> | Centroid: | <104,63> |
| Corners: | 4 | Corners: | 6 |
| Height: | 13.85641 | Height: | 4.233902 |

| BUILDING: 5 | | BUILDING: 6 | |
|---|---|---|---|
| Perimeter: | 107.5956 | Perimeter: | 96.08443 |
| Area: | 287.3164 | Area: | 388.7220 |
| Orientation: | 333.4350 | Orientation: | 57.99463 |
| Centroid: | <132,109> | Centroid: | <84,153> |
| Corners: | 8 | Corners: | 4 |
| Height: | 8.852704 | Height: | 9.814955 |

Figure 11(a) shows two that appear to be two buildings with surface markings, (b) the extracted edge segments, (c) the detected corners, and (d) the boxes found. Nine boxes are found. Three of them are initially complete, five others become complete after searching for nearby evidence, and one becomes complete after reconstruction. The box on the left is small to be a building. The rest are treated as separate boxes although they could be merged by deleting the thin dark antiparallel segments generated by the surface markings. The undetected box on the lower left include misguiding nearby evidence that requires a more complex analysis. The following is a description of the buildings found:

| BUILDING: 1 | | BUILDING: 2 | |
|---|---|---|---|
| Perimeter: | 203.8170 | Perimeter: | 182.0516 |
| Area: | 1997.171 | Area: | 1273.349 |
| Orientation: | 255.4111 | Orientation: | 254.2680 |
| Centroid: | <50,178> | Centroid: | <75,171> |
| Corners: | 4 | Corners: | 4 |
| Height: | 9.253628 | Height: | 8.981462 |

| BUILDING: 3 | | BUILDING: 4 | |
|---|---|---|---|
| Perimeter: | 202.2840 | Perimeter: | 50.02153 |
| Area: | 1939.353 | Area: | 93.19094 |
| Orientation: | 255.4111 | Orientation: | 147.8043 |
| Centroid: | <99,165> | Centroid: | <39,106> |
| Corners: | 4 | Corners: | 8 |
| Height: | 8.981462 | Height: | 5.715476 |

| BUILDING: 5 | | BUILDING: 6 | |
|---|---|---|---|
| Perimeter: | 81.22091 | Perimeter: | 117.6088 |
| Area: | 286.6831 | Area: | 95.79863 |
| Orientation: | 141.6325 | Orientation: | 233.9726 |
| Centroid: | <75,94> | Centroid: | <100,78> |
| Corners: | 8 | Corners: | 6 |
| Height: | 6.531973 | Height: | 3.265986 |

| BUILDING: 7 | | BUILDING: 8 | |
|---|---|---|---|
| Perimeter: | 173.6268 | Perimeter: | 201.3521 |
| Area: | 1272.808 | Area: | 1528.893 |
| Orientation: | 69.44396 | Orientation: | 255.6221 |
| Centroid: | <116,147> | Centroid: | <131,162> |
| Corners: | 4 | Corners: | 4 |
| Height: | 7.620635 | Height: | 8.036229 |

Table 1 shows some experimental data associated with the processing of these four scenes.

## 4. CONCLUSION

A successful building detector was described which takes advantage of the geometric appearance that most buildings have. The system uses both the line segments approximating the intensity edges in the image and the intensity data for pixel measurements at the various stages of the process.

A hierarchy of features from edges to line segments to geometric features to geometric regions to buildings is constructed. The features at each level in the hierarchy are interpreted on the basis of geometric and illumination constraints imposed on the observed image edges by

the geometric properties of the objects of interest and by the direction of illumination.

Three-dimensional objects on the surface are constrained to cast shadows in a unique direction for nonvertical sun angles. We make use of this fact throughout the process. Future work will involve a more extensive use of shadows and three dimensional analysis of aerial images.

**TABLE 1. EXPERIMENTAL DATA**

All programs were run on a DEC KL-10 processor under TOPS-20.

| | Scene 1 (Fig. 8) | Scene 2 (Fig. 9) | Scene 3 (Fig. 10) | Scene 4 (Fig. 11) |
|---|---|---|---|---|
| Convolution | | | | |
| -Image Size | 200x200 | 200x200 | 200x200 | 200x256 |
| -Filter Size | 11 | 17 | 11 | 17 |
| -Time | 69s | 145s | 72s | 197s |
| Zero-Crossings | | | | |
| -Time | 5.1s | 7.3s | 5.4s | 10.3s |
| Segments | | | | |
| -Number | 795 | 554 | 921 | 766 |
| -Time | 14.0s | 9.7s | 16.2s | 13.6s |
| Corners | | | | |
| -Number | 16 | 38 | 35 | 67 |
| -Components | 28 | 70 | 70 | 120 |
| (minimum length of either segment component: 10) | | | | |
| -Time | 1.5s | 1.8s | 2.0s | 2.2s |
| Boxes | | | | |
| -Hypotheses | 8 | 26 | 26 | 42 |
| -Actual boxes | 2 | 3 | 7 | 9 |
| -Time | 6.2s | 8.2s | 8.8s | 15.5s |
| Buldings | | | | |
| -Detected | 2 | 2 | 6 | 8 |
| -Actual Number | 2 | 3 | 6 | 9 |
| -Time (included above) | | | | |
| TOTAL TIME | 95.8s | 172.0s | 104.4s | 238.6s |

## 5. REFERENCES


[1] A. Huertas, "Corner Detection for Finding Buildings in Aerial Images," USCIPI Technical Report No. 960, pp. 61-68, September 1981.

[2] M.B. Clowes, "On Seeing Things," *Artificial Intelligence*, Vol. 2, pp. 79-116, February 1971.

[3] D.A. Huffman, "Impossible Objects as Nonsense Sentences," *Machine Intelligence*, Vol. 6, pp. 295-323, July 1971.

[4] D. Waltz, "Understanding Line Drawings of Scenes with Shadows," *The Psychology of Computer Vision*, P. Winston (ed.), McGraw-Hill, 1975, pp. 19-92.

[5] D. Lowe and T. Binford, "The Intrepretation of Geometric Structure from Image Boundaries," *Proc. Image Understanding Workshop*, pp. 39-46, 1981.

[6] M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*, New York: Plenum, 1980.

[7] M. Tavakoli and A. Rosenfeld, "Building and Road Extraction from Aerial Photographs," *IEEE Trans. SMC*, Vol. 12, pp. 84-91, January 1982.

[8] R. Nevatia and K.R. Babu, "Linear Feature Extraction and Description," *Computer Graphics and Image Processing*, Vol. 13, pp. 257-269, 1980.

[9] A. Huertas and R. Nevatia, "Edge Detection in Aerial Images Using Laplacian-Gaussian Filters," USCIPI Technical Report No. 1010, pp. 16-26, September 1981.

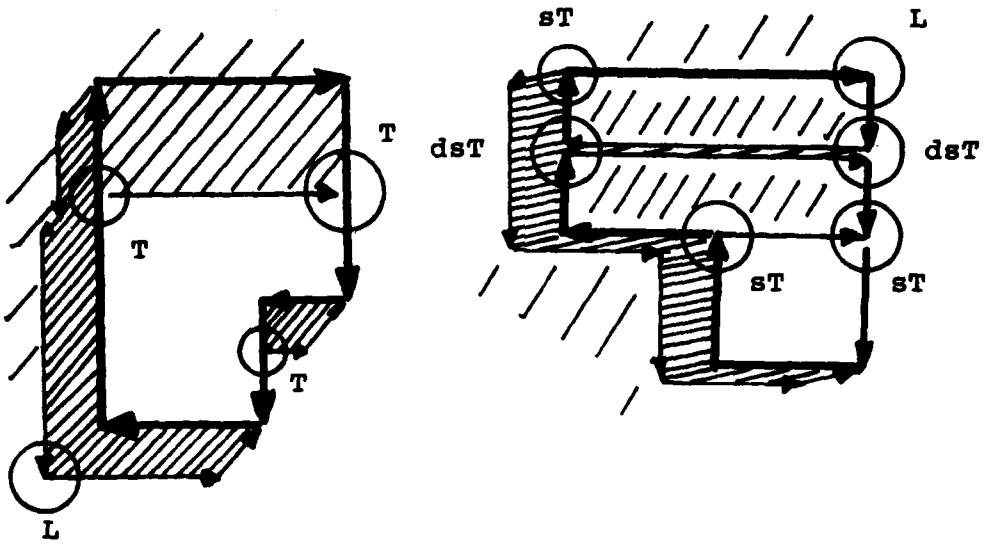



Figure 1. Junctions

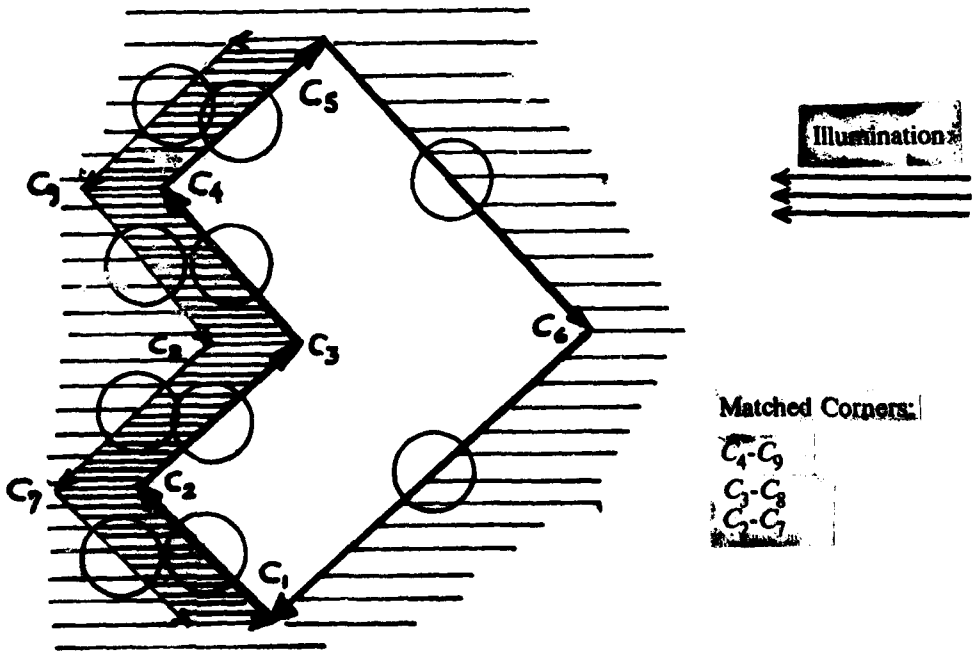



Figure 2. Pixel measurements and corner matching

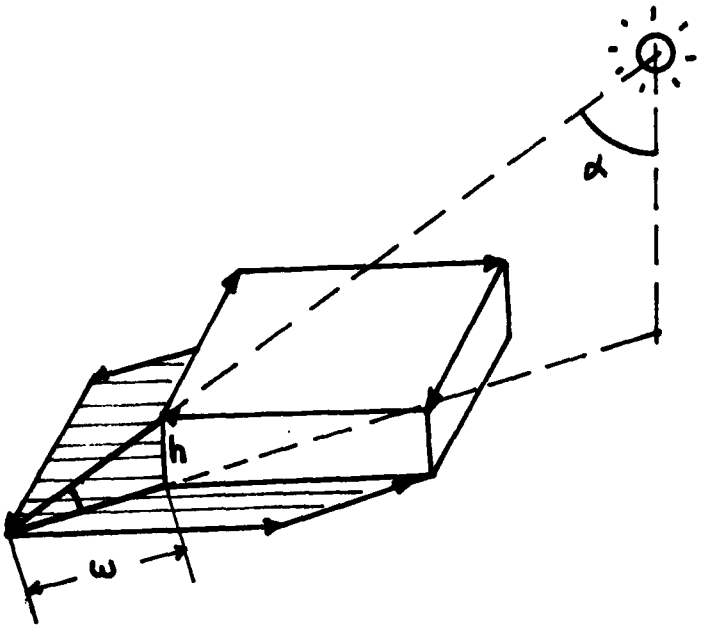



Figure 3. Height hypothesis

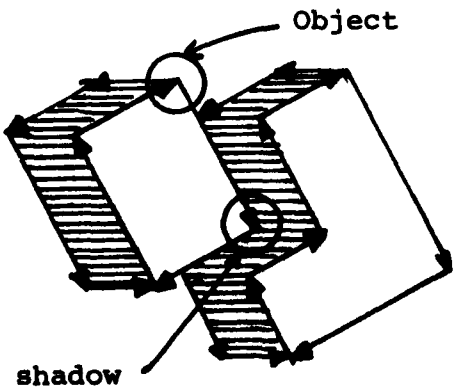



Figure 4. Conflicting interpretation

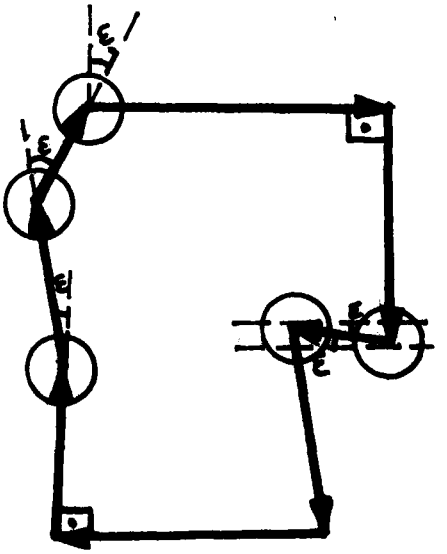
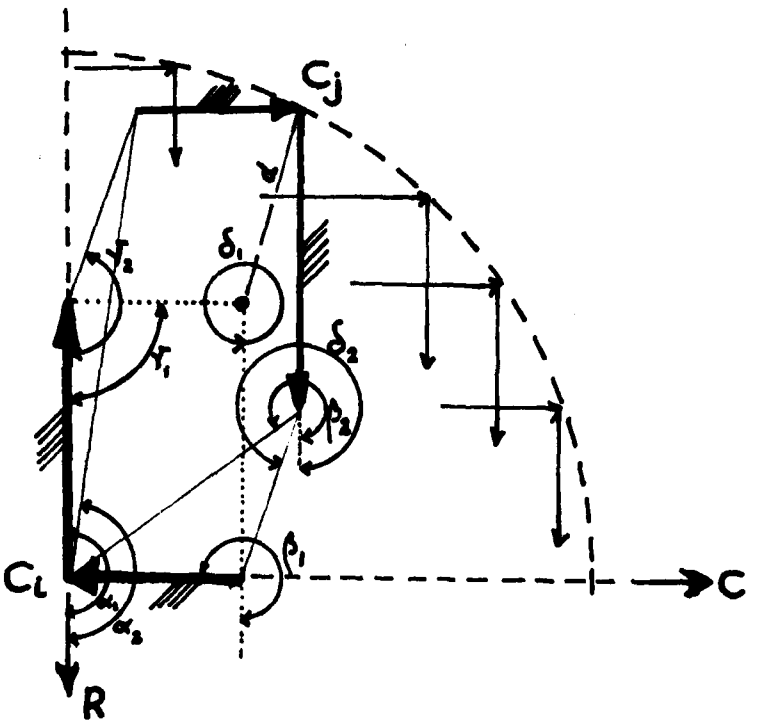
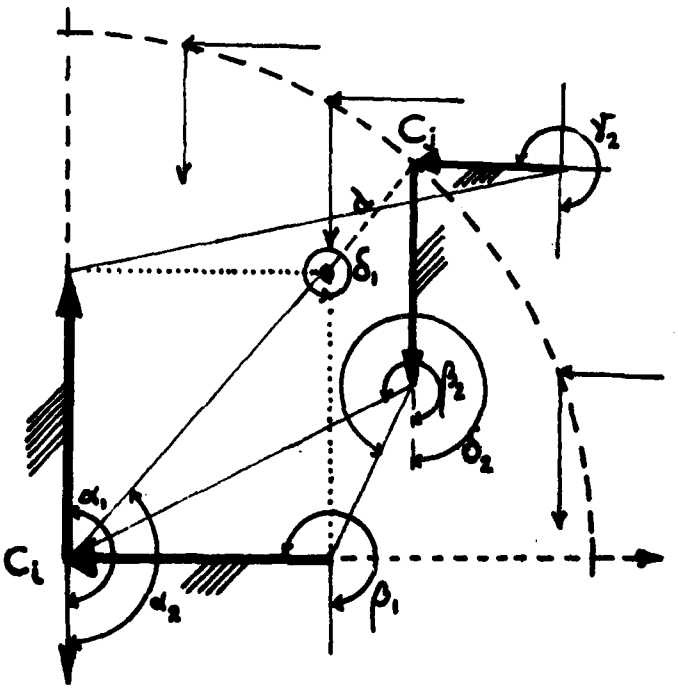


Figure 5. Boundary trace

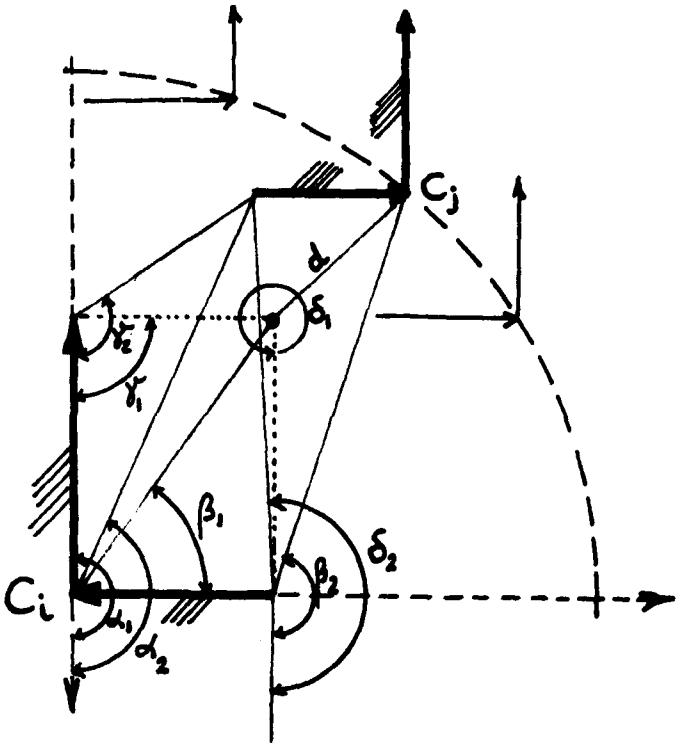



(a) Diverging corners

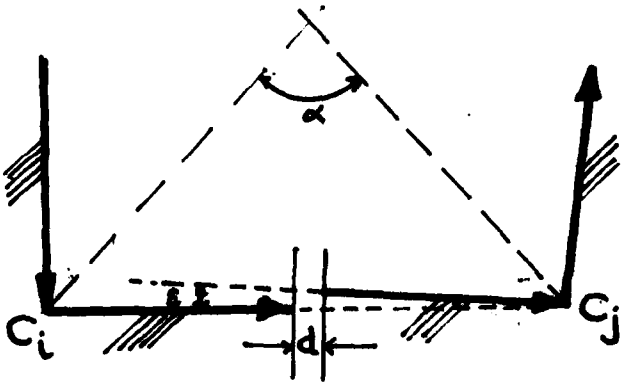



(b) Inverted corners

Figure 6. Matching corners: $\alpha_2 > \alpha_1; \beta_2 > \beta_1; \nu_2 > \nu_1; \delta_2 > \delta_1$.




(c) Inverted corners




(d) Opposite corners

Figure 6. Continued

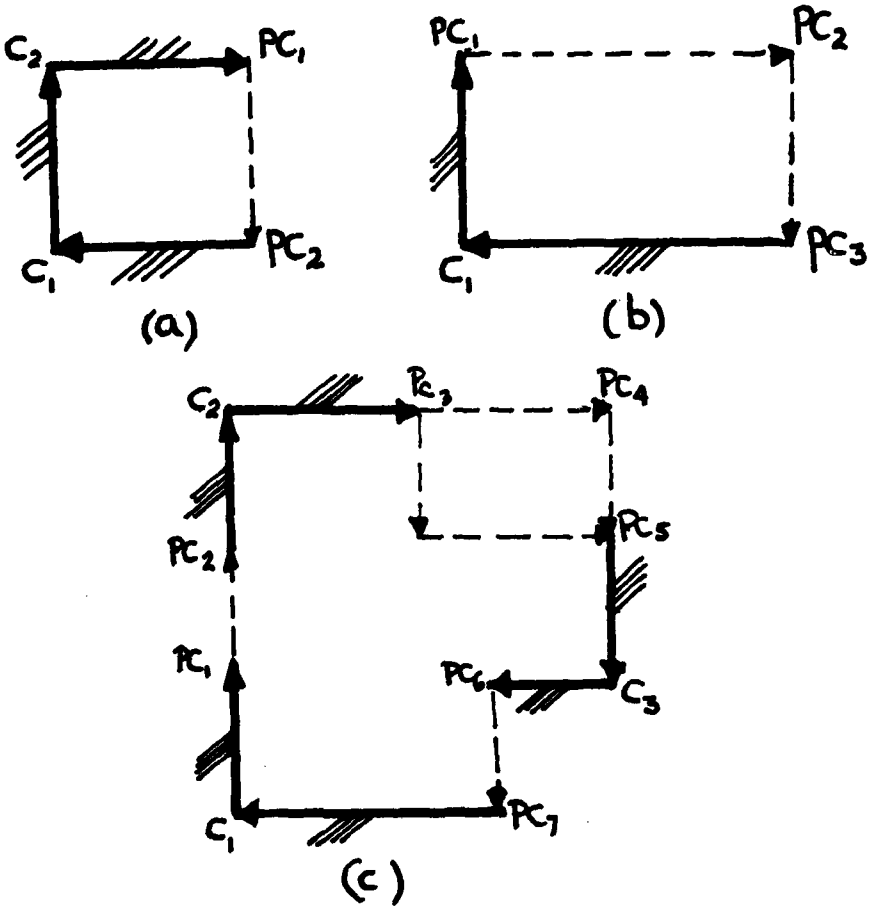
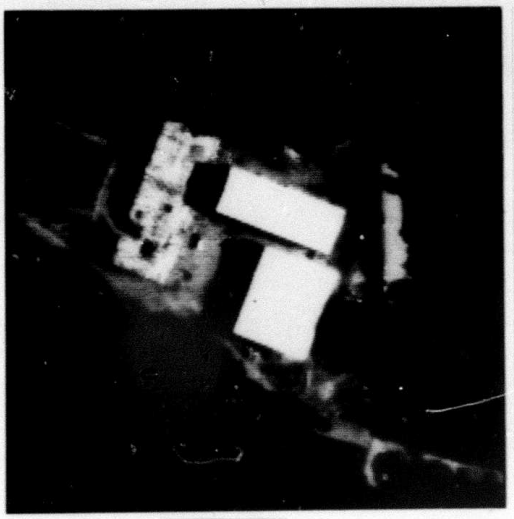
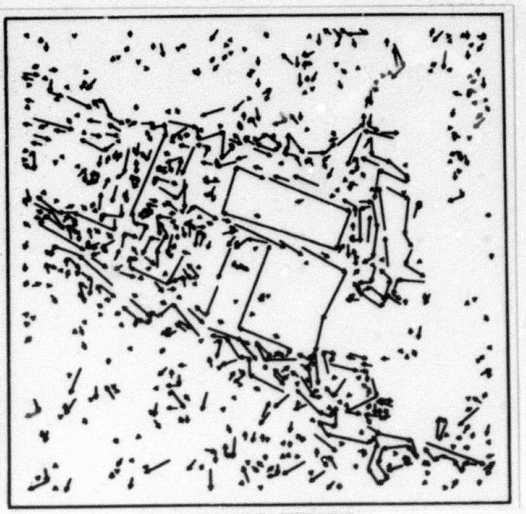


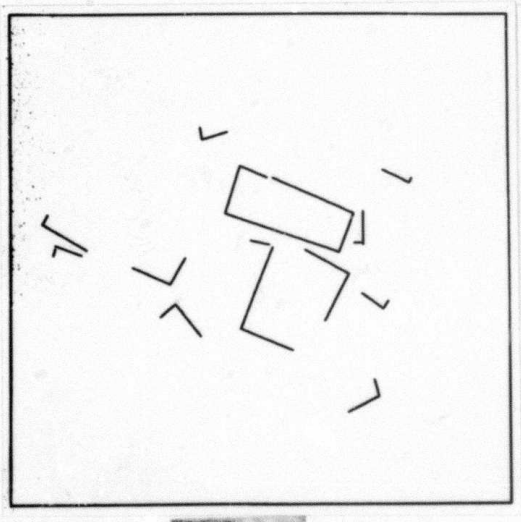

Figure 7. Hypotheses for missing elements



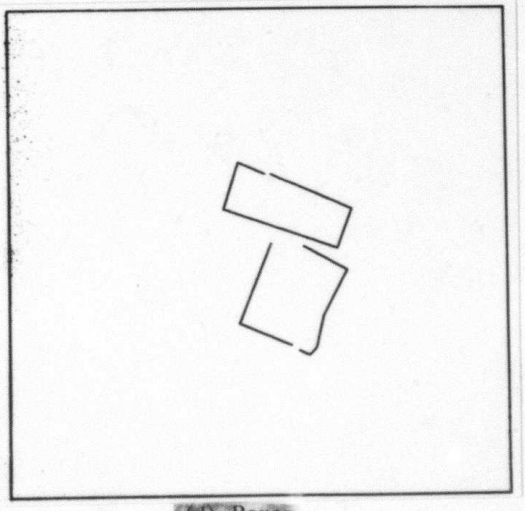
(a) Image



(b) Edge segments

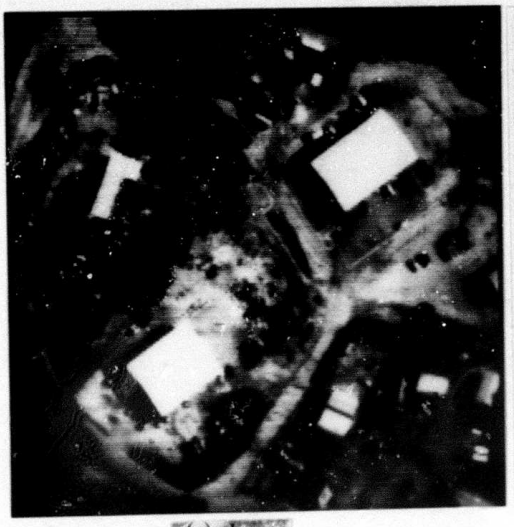


(c) Corners

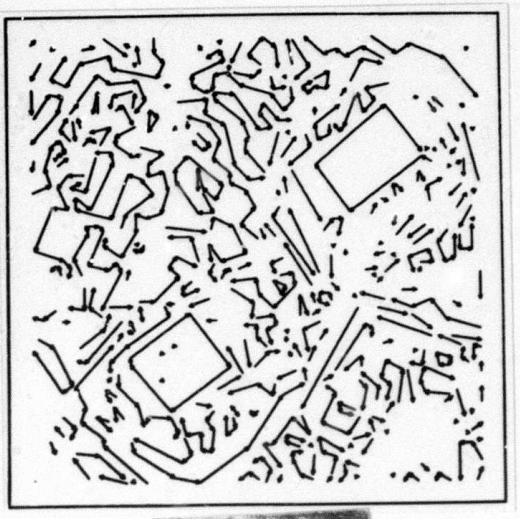


(d) Boxes

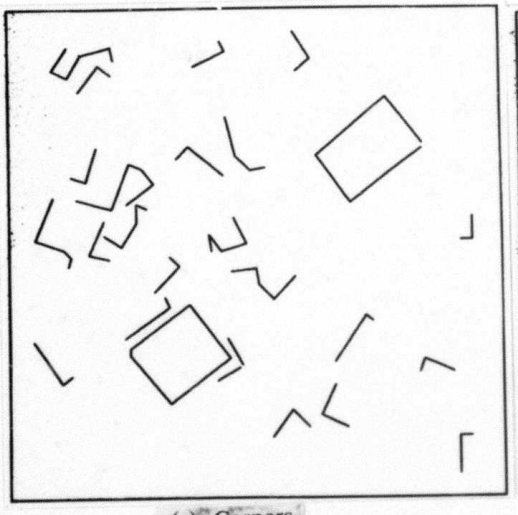
Figure 8. Scene 1



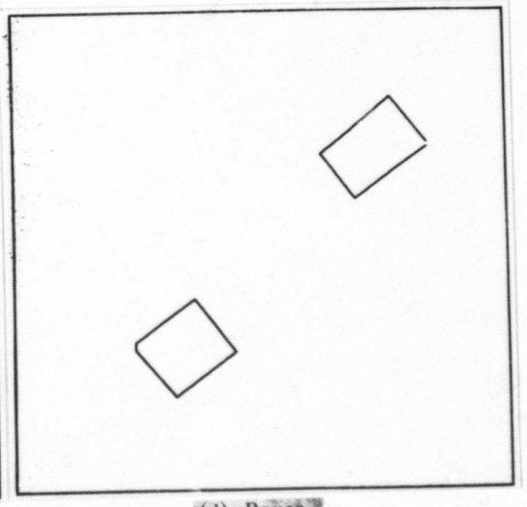
(a) Image



(b) Edge segments



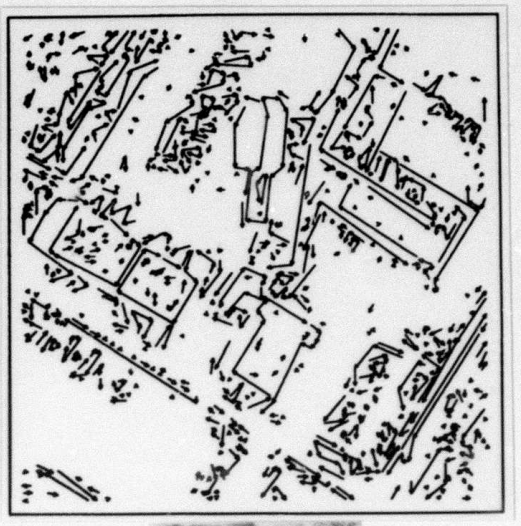
(c) Corners



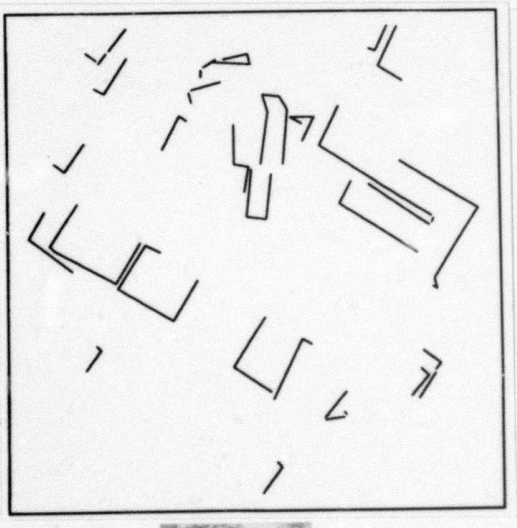
(d) Boxes

Figure 9. Scene 2

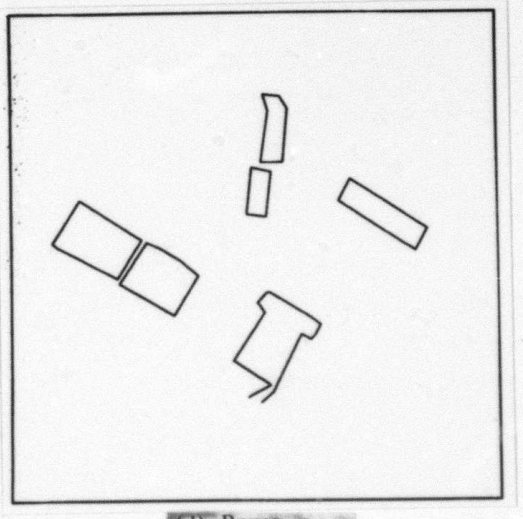
(a) Image



(b) Edge segments

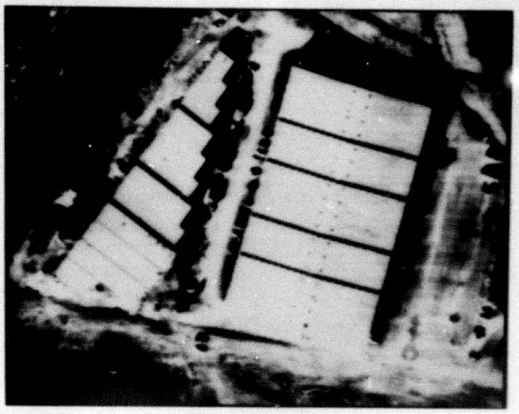


(c) Corners

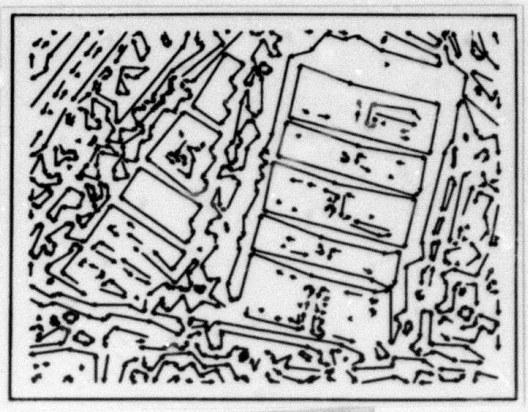


(d) Boxes

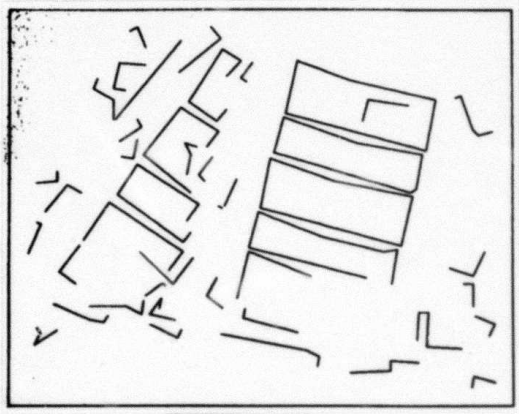
Figure 10. Scene 3



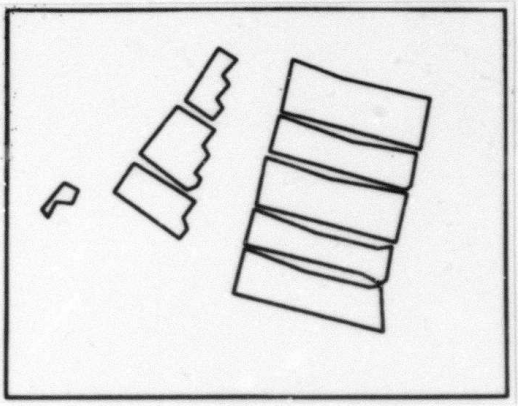
(a) Image



(b) Edge segments



(c) Corners



(d) Boxes

Figure 11. Scene 4

# 1.4 SEGMENTATION OF IMAGES INTO REGIONS USING EDGE INFORMATION

GERARD G. MEDIONI

## 1. INTRODUCTION

There have traditionally been two main approaches to the segmentation of images, edge based and region based.

- □ Edge based methods proceed by locating discontinuity points in the intensity image and connecting them to obtain primitives. They have the advantage of preserving most of the information present in the intensity picture but produce very low level primitives even after further processing (segments) [1]. They are very appropriate to describe elongated objects such as roads and rivers.

- □ Region based methods proceed either by merging regions that have similar intensity and a weak boundary separating them [2,3], or by recursively splitting regions using a threshold defined by histograms [4]. This last technique is very effective on multispectral images. These methods produce higher level primitives (regions with a set of attributes), but most of the time these regions do not correspond to physical entities unless their intensity differs everywhere from the background. If the contrast is too weak, the object will "leak" and will be merged with its background.

We present here a method trying to combine the good points of the 2 methods described above. It substantially differs from the expansion-contraction approach used by Perkins [5] to bridge gaps in the edge image, and does not require an object's interior to contrast with its surround as in Milgram's "superlice" technique [6].

## 2. DESCRIPTION OF THE METHOD

From the grey-level image, we extract the edge points and organize them into linear segments. Using this edge information, we create a new image in which pixels belonging to an edge segment get an intensity depending upon the contrast of the edge at this point and the total length of the segment. We now bridge the gaps in the edges by replacing the intensity at each point by the sum of the intensities in a small square window centered at this point. By thresholding this new image, we obtain a binary image from which we extract the connected regions of intensity 0. These regions are smaller than the expected ones because of the smoothing process, so we expand each one individually to obtain the final result.

### 2.1. Processing the grey-level image

We first extract the edges from the image, thin them and link them using the technique developed by Nevatia and Babu [1]. The final primitives we obtain are SEGMENTS, linear pieces approximating a set of edge points. The attributes of a segment are its 2 end points, its length $L$ and its strength $S$, which is the sum of the contrast of each point. Since we want to eliminate the gaps in the edges following the boundary of an object and reduce the influence of small random or textured edges, we create an image $f(i,j)$ as follows:

if $(i,j)$ belongs to a segment SEG

then if LENGTH[SEG] < MINLENGTH

then $f(i,j)$ = LENGTH/STRENGTH

else $f(i,j)$ = STRENGTH

else $f(i,j) = 0$.

This non-linear process permits us to recognize long (>MINLENGTH) segments and to give a high weight to their points.

### 2.2. Summing the Image

Given the image $f(i,j)$, we use a simple texture/no-texture discrimination process by creating a new image $g(i,j)$ as follows:

$$g(i,j) = \sum_{l=i-n}^{i+n} \sum_{m=j-n}^{j+n} f(l,m)$$

That is, $g(i,j)$ is the sum of $f(i,j)$ in a square window of size $2n+1$ centered in $(i,j)$. We then threshold this image to get a binary version of it:

$h(i,j) = 0$ if $g(i,j) < \text{THR}$
$= 1$ otherwise.

### 2.3. Extracting Regions

From the image $h(i,j)$, we extract all connected regions of intensity 0. Each region represents a shrunk version of a region in which no edges, or very small and weak edges, are present and the gap between the edge and the border created by the edge is n pixels, n being defined above. In order to reconstruct the physical region, we use a growing procedure on each region as described in [7]: for each pixel, we consider a square window of size $2n+1$ centered at that point, and set the pixel to 1 if any pixel in the window is 1. One problem with this technique is that some corners get rounded.

### 2.4. Interpretation

Each region now corresponds to a set of edges forming a nearly closed boundary enclosing this region. These regions can be further filtered by looking at their attributes, such as area, ratio of perimeter$^2$/area and others. They can be the input of a region matching program or can be looked at individually to see if there is an adjoining projected shadow.

## 3. RESULTS

We tried the above procedure on 2 views of the same scene showing part of the Fort Belvoir Military Reservation in Virginia. The original images have a resolution of 600 by 600 and are shown on figures 1a and 2a. Figures 1b and 2b show the segments extracted from the

intensity array. Note that the boundary of the large building in the lower left of fig. 1b is not closed, or even nearly closed. Figures 1c and 2c show the image after summation. The following parameters were used: $n = 4$ (that is, windows are 9 by 9). MINLENGTH = 12 (minimum length of a segment for non-linear processing). From these images we extract connected regions of intensity $< 150$. We now expand each region individually and filter out all regions with a value of perimeter$^2$/area $> 35$ to obtain the final result, as shown on figures 1d and 2d. As we can see, no buildings are missed and their shape is rather well conserved. Figures 1e and 2e show the set of regions obtained by a conventional region splitting [4]. In both images the large building in the lower left is totally lost and some other buildings are merged into a single region.

## 4. CONCLUSION

The method described above provides better segmentation than region growing or region splitting techniques without semantic information. Computing histograms, especially on monochromatic images, does not always provide a good threshold, even though edges define a clear boundary. We are currently investigating the exact effect of the parameters and a segmentation method coordinating edge information and region splitting.

## 5. REFERENCES


[1] R. Nevatia and K. Ramesh Babu, "Linear Feature Extraction and Description," Computer Graphics and Image Processing, Vol. 13, June 1980, pp. 257-269.

[2] C. R. Brice and C. L. Fennema, "Scene Analysis Using Regions," Artif. Intell. 1:3, 1970, pp. 205-226.

[3] Y. Yakamovsky and J. A. Feldman, "A Semantics-Based Decision Theory Region Analyser," Proc. IJCAI-73 Stanford, 1973, pp. 580-588.

[4] R. Ohlander, K. Price and R. Reddy, "Picture Segmentation Using a Recursive Splitting Method," Computer Graphics and Image Processing, Vol. 8, 1978, pp. 313-333.

[5] W. A. Perkins, "Region Segmentation of Images by Expansion and Contraction of Edge Points," Proc. IJCAI-79 Tokyo, Aug. 79, pp. 699-701.

[6] D. Milgram, "Region Extraction Using Convergent Evidence," Computer Graphics and Image Processing, Vol. 11, No. 1, 1979, pp. 1-12.

[7] A. Rosenfeld and A. C. Kak, "Digital Picture Processing," Academic Press, 1976.

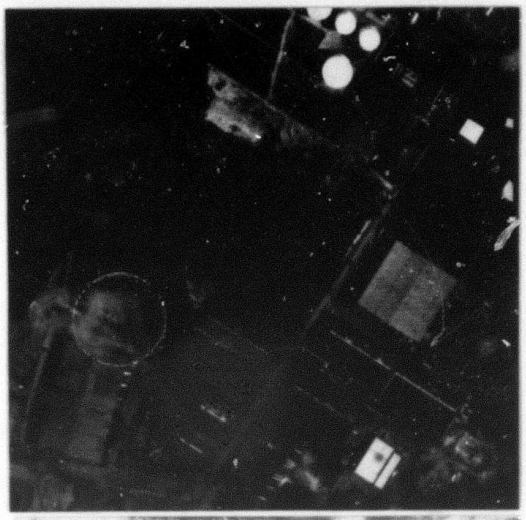


Figure 1a. Original image resolution 600x600

Figure 1b. Segments extracted from the original image

Figure 1c. Summed image

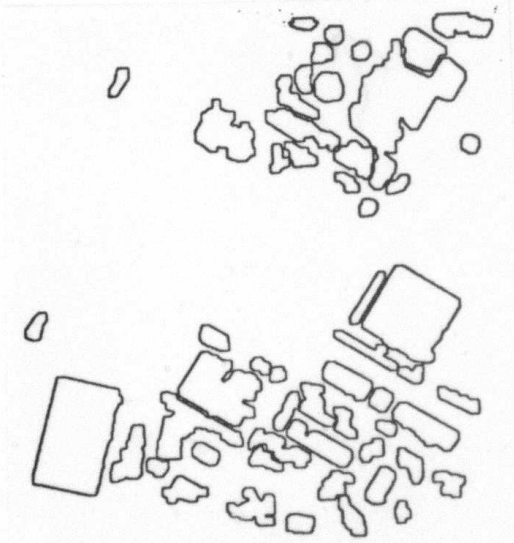


Figure 1d. Regions from the summed image

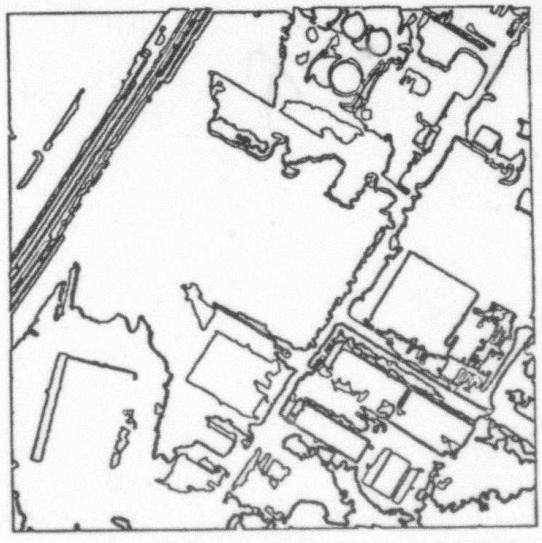


Figure 1e. Regions obtained by splitting method

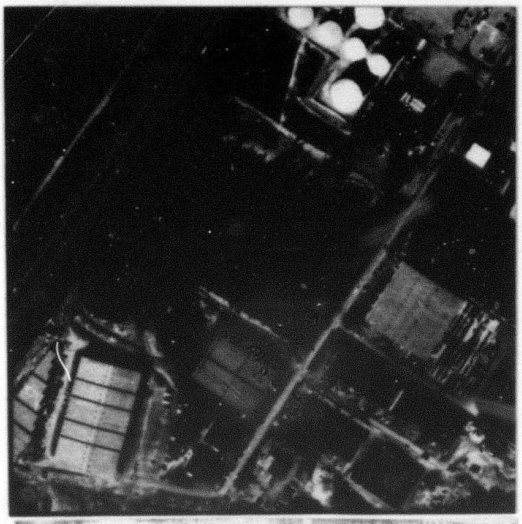


Figure 2a. Original image resolution 600x600

Figure 2b. Segments extracted from the original image

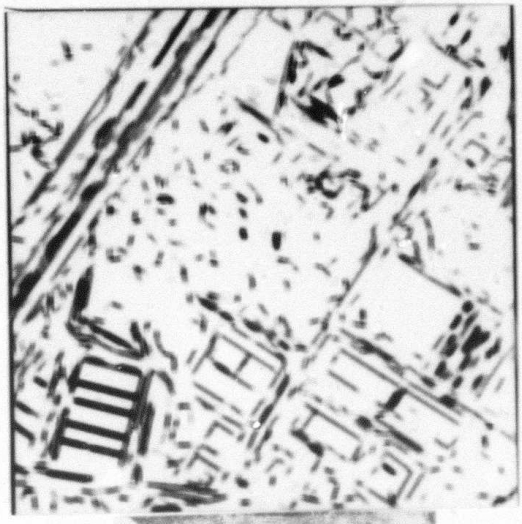


Figure 2c. Summed image

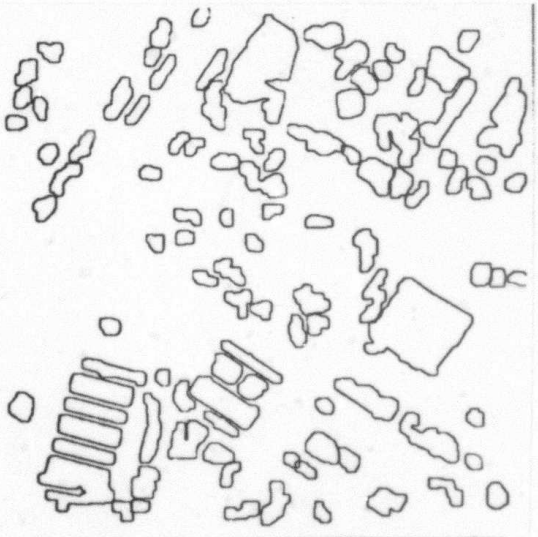


Figure 2d. Regions from the summed image

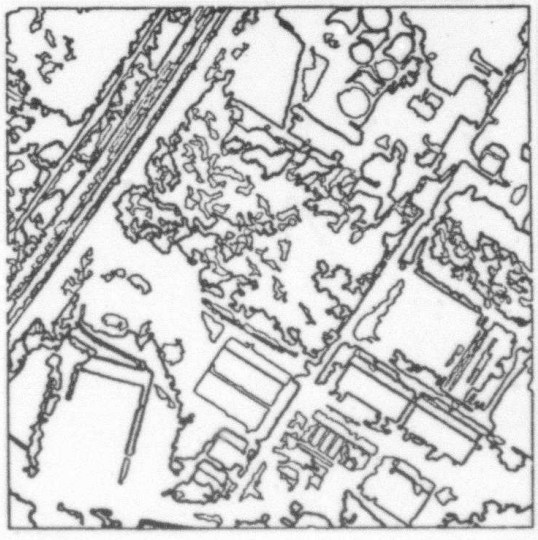


Figure 2e. Regions obtained by splitting method

# 1.5 USING TEXTURE EDGE INFORMATION IN AERIAL IMAGE SEGMENTATION

H.Y. LEE AND K.E. PRICE

## 1. INTRODUCTION

Extraction of the textured regions in an aerial image is important because it will ease the task of segmenting more important objects (e.g. buildings, roads, rivers) which are more likely to be in the untextured regions. Because of the use of large windows in the texture measure computation, the textural property computed over a window overlapping two different regions has a mixture of the two textural properties corresponding to the regions. As a result, the histogram of a texture measure for a natural scene seldom has well separated peaks. Hence thresholding type procedures in texture segmentation have difficulties in finding the level for threshholding and rarely gives accurate region boundaries. On the other hand, edge-based techniques have their own difficulties in obtaining closed boundaries of the regions and dealing with false alarms.

Milgram [1] tried to combine these two approaches in a luminance thresholding technique in which the initial threshold level is adjusted according to the rate of the boundary points coincidence with the edge points. Since that kind of effort is even more needed in texture segmentation, this report proposes a procedure with the initial segmentation by a thresholding technique followed by an iterative growing/shrinking of the "core" object points using texture edge information.

## 2. TEXTURE MEASURES

Based on previous work by Julesz [2] and Pratt et al. [3], Faugeras and Pratt [4] proposed a technique of texture feature extraction based on measurements of the autocorrelation function of the texture field and the first order histogram of the decorrelated field. Another technique was proposed by Faugeras [5] based on a human visual model where textures are analyzed through a bank of nonlinear channels. Laws [6] expanded on both works and proposed to analyze textures with a set of filters of small spatial extent (analogous to the decorrelation operator in [4] or the bandpass filters in [5]) and compute local "energy" values in the output planes after convolution (anologous to the $L$-norms in [5]). If $N$ filters are used in the process, texture is represented at every pixel by an $N$-dimensional vector. Moreover, Laws has been able to exhibit a limited set of filters (of the order of 4) that seem to perform best in terms of classification accuracy on a fairly large set of natural texture. Because of its simplicity and good performance, we adopted Laws' method of texture feature extraction summarized in Fig. 1.

## 3. INITIAL SEGMENTATION

The basic scheme is the Ohlander-Price region splitting method [7] which uses the histograms of the available measures in a recursive thresholding procedure. The original sceme is aimed at segmenting the "nonbusy" parts of the scene and uses the histograms of the nine color coordinate signals which are measured over those parts of the scene that are relatively devoid of texture. If the "property" histograms are not all unimodal, a threshold selection procedure is invoked to determine the best property and the best level for thresholding of that property. After a threshold level has been determined, the image is subdivided into its segmented parts. The procedure is then repeated on each part until the resulting property histograms become unimodal or the segmentation reaches a reasonable stage of separation.

Since our task, however, is to separate the textured regions from the nontextured parts of the scene, those color signal histograms are replaced by the histograms of the texture energy measures. One thing to note is that one of the convolution filters mentioned in the previous section can be a weighted-average, low-pass filter. The resulting texture energy measure is for the cases where there is no significant textured region or average intensity of the texture present bears more information than other structural texture measures. The latter is the case of the initial segmentation performed on DMA2.512 (Fig.2). The resulting region boundaries are shown in Fig.3. Though the fine textural structure of the large forest regions is notable, intensity information prevailed in extracting them.

## 4. REFINEMENT OF THE BOUNDARIES

The previously developed texture edge detector[8], which is patterned after the methods for detecting intensity edges using directional derivatives, gives reasonably accurate information about the locations of the texture edges along with their orientations. To preserve the closed boundary, compact nature of thresholded regions, an iterative refining procedure is developed. At each loop, the previous boundary of a segmented region is refined using the local context of the texture edge information.

After the texture edge orientation of a boundary point determines the neighborhood belonging to it, the point remains to be a boundary point if it has the local maximum of the texture edge magnitudes of that neighborhood. If the texture edge magnitudes decrease or increase along the outward direction in the neigborhood, it is deleted or becomes a inside point by adding any of the four-connected neighbors which did not belong to the region in the previous stage. The procedure is then repeated until the convergence slows down enough or a predetermined maximum iteration is reached. The convergence criterion used is the ratio of the total number of boundary points with maximum texture edge magnitude to the number of all the boundary points. Because different portions of a region boundary usually face different

neighboring regions, texural measures used in the texture edge computations may be changed several times during one boundary tracking. The whole iterative procedure can be applied to a selected set of regions or to all the regions initially segmented. The refined boundaries of Fig.2 after 10 iterations for each region are shown in Fig. 4. Most of the regions show convergence (above 80% of the ratio) after the tenth iteration.

## 5. CONCLUSIONS

Some final boundaries show significant improvements while others can not justify the relatively costly process. This may be due to the inadequecy of the input textural measures. Instead of using the large fixed set of measures which may turn out to be useless, we can confine our task to the extraction of specific regions, such as forest regions only, so that outside knowledge about the characteristics of the forest model can be used to look for the specific peak in the histogram of the specific texture measure.

Another important question is whether the texure information is really useful for meaningful image segmentation. In many image understanding systems, especially for the general purpose ones, there might be many cases where the use of texture would hinder the simpler solution of the problems. This is the reason why the texture operators should work with all the other lower or higher level operators in very cooperative and selective ways. Hence a criterion for determing texture presence/absence is being considered o avoid excessive involvement with textures in early stages of processing. The other interesting subject is to find a remedy for the problem with the threshold selection in the histograms of textural properties mentioned in the introduction.

## 6. REFERENCES


[1] D.L. Milgram, "Region Extraction Using Convergent Evidence," *Computer Graphics and ImageProcessing*, Vol. 11, 1979, pp. 1-12.

[2] B. Julesz, "Visual Pattern Discrimination," *IRE Transactions on Information Theory*, Vol. IT-8, No. 1, pp. 84-9, February 1962.

[3] W.K. Pratt, O.D. Faugeras and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Fields," *IEEE Transactions on Systems, Man, and Cybernetics*, November 1978.

[4] O.D. Faugeras and W.K. Pratt, "Decorrelation Methods of Texture Feature Extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, November 1980.

[5] O.D. Faugeras, "Texture Analysis and Classification Using a Human Visual Model," *IJCPR* Proceedings, Kyoto, Japan, November 1978.

[6] K.I. Laws, "Texture Image Segmentation," Department of Electrical Engineering, USC, Ph.D. Dissertation, January 1980.

[7] R. Ohlander, K. Price and R.Reddy, "Picture Segmentation Using a Recursive Region Splitting Method," *Computer Graphics and Image Processing*, Vol. 8, 1978, pp. 313-333.

[8] O.D. Faugeras and H.Y. Lee, "Texture Edge Detection,"Technical Report USCIPI 960, 1980.

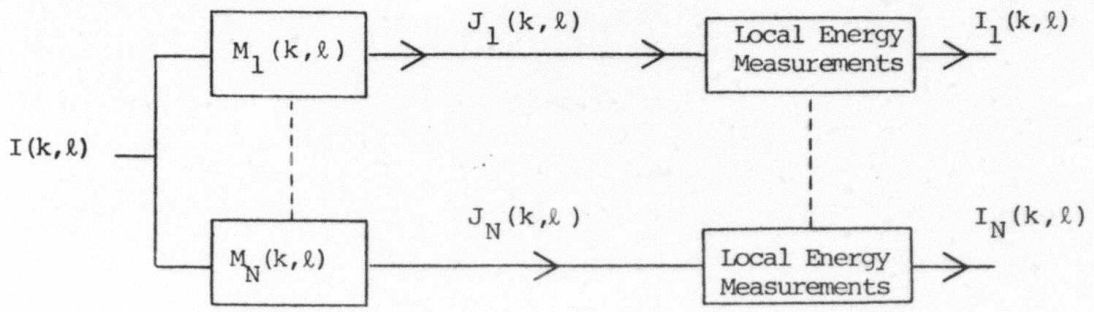



Figure 1. Texture feature extraction by parallel processing of an image with convolution masks $M_1(k,l,...,M_N(k,l)$. The local energy measurement are given by

$$I_n(k,l) = \sum_{i,j} |w_n(k-i,l-j)J_n(i,j)|^P \quad n=1,...,N$$

where $P$ is a positive integer and the window function $w_n$ may or may not depend on $n$.

Figure 2. Original DMA2.512 image

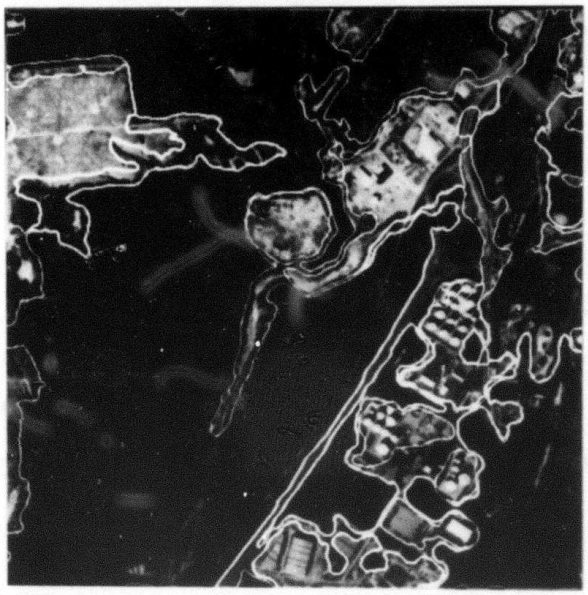


Figure 3. Region boundaries of the initial segmentation

Figure 4. Refined region boundaries