E83-10118

NASA-CR-167778

(E83-10118) PROCEEDINGS OF THE NASA WORKSHOP ON IMAGE ANALYSIS (Texas A&M Univ.) 149 p HC A07/MF A01 CSCL 05B

N83-15766 THRU N83-15773

Unclas

G3/43 00118

DEPARTMENT OF MATHEMATICS

TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS

PROCEEDINGS OF THE

NASA WORKSHOP ON IMAGE ANALYSIS

Texas A&M University
College Station, Texas
April 28-30, 1982

Prepared for

Earth Resources Research Division
NASA/Johnson Space Center
Houston, Texas 77058

by

L. F. Guseman, Jr.
Principal Investigator
Department of Mathematics
Texas A&M University
College Station, Texas 77843

under

NASA Contract NAS 9-16447
"Studies in Mathematical Pattern Recognition
and Image Analysis"

Introduction

by

K. S. Fu
Purdue University

The NASA Workshop on Image Analysis held on April 28-30, 1982 at Texas A&M University, College Station, Texas, provided an opportunity for experts in the areas of pattern recognition, image processing, and remote sensing to assess past progress and to project future development in the area of image analysis with respect to remote sensing applications.

A block diagram of the general image analysis system is given in Figure 1. The preprocessing stage usually refers to filtering, enhancement, and/or coding of raw imagery data. The segmentation stage involves the determination of various regions of importance in the image. Features such as shape and texture measurement are then extracted from each region; a classification technique is often employed to recognize these regions. Once each region has been recognized and the relations among these regions have been identified, a complete description and possibly the interpretation of the image can be obtained through a structural analysis. A priori knowledge (the so-called "world model") of the images under study plays an important role in the design of each stage.

The program of the three-day workshop was devoted to the three major topics of image analysis: segmentation, shape and texture analysis, and structural analysis. A survey paper and two or three special papers were presented on each topic. Formal presentations were followed by panel discussions which assessed past progress and identified future research problems in each topic area.

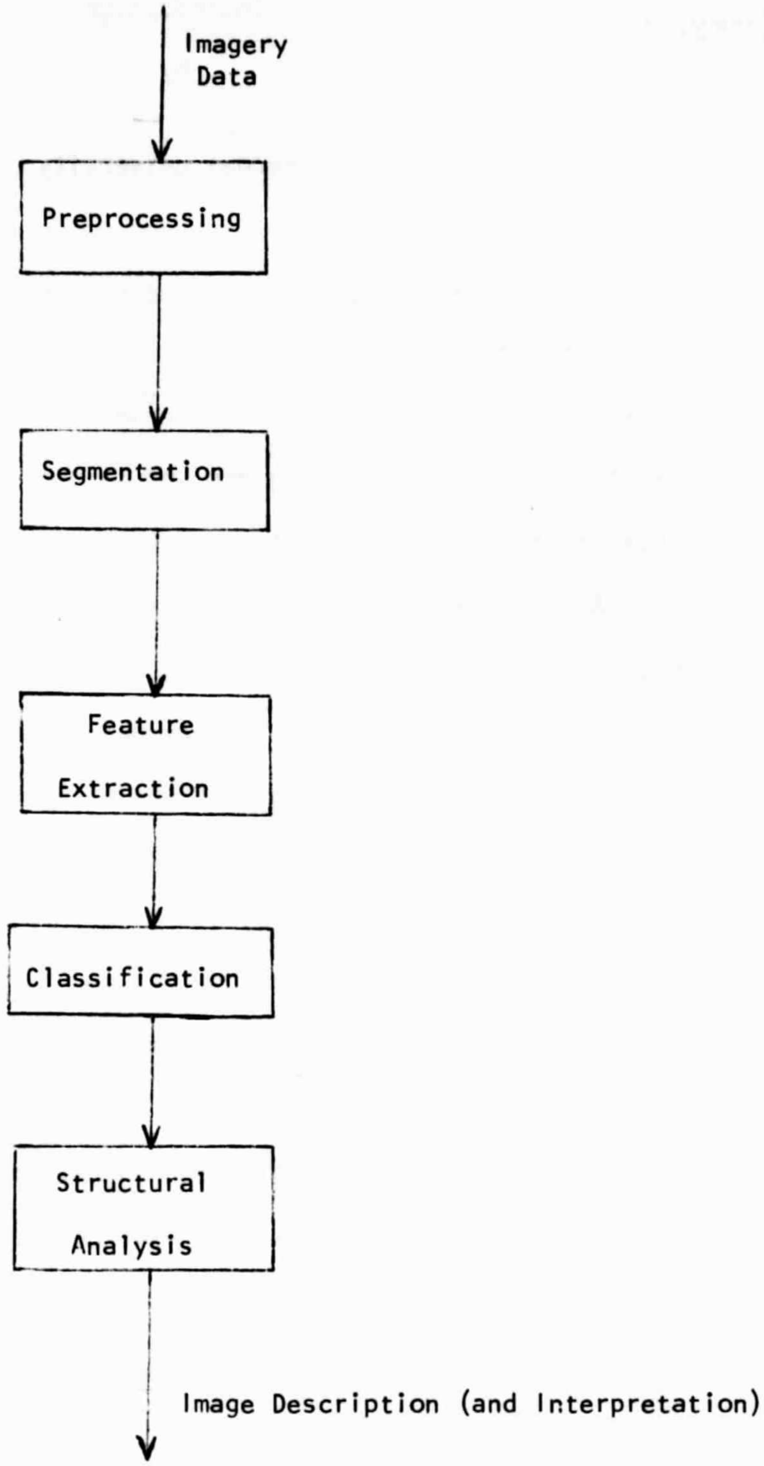


Figure 1. Block diagram of image analysis system

OMIT TO
P. 7

TABLE OF CONTENTS

NASA WORKSHOP ON IMAGE ANALYSIS

Texas A&M University
April 28-30, 1982
Room 510, Rudder Tower

Wednesday, April 28:

8:15 - 8:30	Coffee and donuts
8:30 - 9:30	Introduction R. B. MacDonald, NASA/Johnson Space Center Opening Remarks L. F. Guseman, Jr., Texas A&M University
9:30 - 10:00	Overview: Image Analysis K. S. Fu, Purdue University
10:00 - 10:30	Coffee Break
10:30 - 11:30	Image Segmentation: A Survey Robert M. Haralick, Virginia Polytechnic and State University
11:30 - 12:30	Cooperative Processes in Image Segmentation Larry Davis, University of Maryland
12:30 - 2:00	Lunch
2:00 - 3:00	Dual Problems in Image Segmentation Jack Bryant, Texas A&M University Susan Jenson, EROS Data Center
3:00 - 3:30	Coffee Break
3:30 - 5:00	Panel Discussion--Image Segmentation Moderator: Azriel Rosenfeld Panelists: R. K. Aggarwal Jack Bryant Larry Davis Robert M. Haralick

Thursday, April 29:

8:15 - 8:30	Coffee and donuts
8:30 - 9:30	Shape and Texture Azriel Rosenfeld, University of Maryland

PRECEDING PAGE BLANK NOT FILMED

NASA WORKSHOP ON IMAGE ANALYSIS, cont'd.

9:30 - 10:30	Shape Identification Using 3-D Features Carolyn M. Bjorklund, Lockheed Palo Alto
10:30 - 11:00	Coffee Break
11:00 - 12:00	Automatic Photointerpretation Via Textural Feature Extraction Julius F. Tou, University of Florida
12:00 - 1:30	Lunch
1:30 - 2:30	Target Screening R. K. Aggarwal, Honeywell
2:30 - 3:00	Coffee Break
3:00 - 4:00	Panel Discussion--Shape and Texture Moderator: Robert M. Haralick Panelists: Carolyn M. Bjorklund Azriel Rosenfeld Julius Tou

Friday, April 30:

8:15 - 8:30	Coffee and Donuts
8:30 - 9:30	Structure Analysis Techniques for Remote Sensing Linda Shapiro, Virginia Polytechnic and State University
9:30 - 10:30	Determining 3-D Motion and Structure from Image Sequences Thomas S. Huang, University of Illinois
10:30 - 11:00	Coffee Break
11:00 - 12:30	Panel Discussion--Image Structure Analysis Moderator: Thomas S. Huang Panelists: K. S. Fu Linda Shapiro

NASA WORKSHOP ON IMAGE ANALYSIS

April 28-30, 1982

Participants and Other Attendees:

Raj Aggarwal
Honeywell, MN17-2349
Systems and Research Center
2600 Ridgeway Parkway
Minneapolis, MN 55413

Martha L. Amis (C31)
Lockheed E.M.S.C.O.
1830 NASA Road 1
Houston, TX 77258

Carolyn M. Bjorklund
Lockheed Palo Alto Research Labs
3251 Hanover Street, Dept. 52-53
Palo Alto, CA 94304

Jack Bryant
Department of Mathematics
Texas A&M University
College Station, TX 77843

William A. Coberly
Division of Mathematical Sciences
University of Tulsa
Tulsa, OK 74104

Larry Davis
Computer Science Center
University of Maryland
College Park, MD 20742

K. S. Fu
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

Larry F. Guseman, Jr.
Department of Mathematics
Texas A&M University
College Station, TX 77843

Robert M. Haralick
Department of Electrical Engineering
Virginia Polytechnic and State Univ.
Blacksburg, VA 24061

Richard P. Heydorn/SG3
NASA/Johnson Space Center
Earth Observation Division
Houston, Texas 77058

Thomas S. Huang
Coordinated Science Laboratory
University of Illinois
1101 W. Springfield Avenue
Urbana, IL 61801

Susan Jenson
EROS Data Center
Applications Branch
Sioux Falls, SD 57198

Michael Merickel
Lockheed E.M.S.C.O.
1830 NASA Road 1
Mail Code C31
Houston, TX 77258

R. S. Rajaram/C31
Lockheed
1830 NASA Road 1
Houston, TX 77258

Azriel Rosenfield
Computer Science Center
University of Maryland
College Park, MD 20742

Linda G. Shapiro
Department of Computer Science
Virginia Polytechnic and State Univ.
Blacksburg, VA 24061

NASA Workshop on Image Analysis
Participants and Attendees, cont'd.

Larry L. Schumaker
Department of Mathematics and Center for
Approximation Theory
Texas A&M University
College Station, TX 77843

Julius T. Tou
Center for Information Research
University of Florida
Gainesville, FL 32611

Curtis E. Woodcock
Department of Geography
University of California
Santa Barbara, CA 93106

N83 15767 D1

Image Segmentation Survey

Robet M. Haralick
Virginia Polytechnic Institute and State University
Dept. of Electrical Engineering
Dept. of Computer Science
Blacksburg, VA 24061

Image segmentation can be accomplished by a variety of techniques which in this survey we classify as:

- Single linkage schemes
- Hybrid linkage schemes
- Centroid linkage schemes
- Histogram Mode Seeking Schemes
- Spatial Clusterning schemes
- Split and Merge Schemes

Single Linkage Image Segmentation

Single linkage image segmentation schemes regard each pixel as a node in a graph. Neighboring pixels whose properties are similar enough are joined by an arc. The image segments are maximal sets of pixels all belonging to the same connected component. Single linkage image segmentation schemes are attractive for their simplicity. They do, however, have a problem with chaining, because it takes only one arc leaking from one region to a neighboring one to cause the regions to merge.

The simplest single linkage scheme defines similar enough by pixel difference. Two neighboring pixels are similar enough if the absolute value of the difference between their gray tone intensity value is small enough. For pixels having vector values, the obvious generalization is to use a vector norm of the pixel difference vector. Instead of using a Euclidean distance, Asano and Yokoya (1981) suggest that two pixels be joined together if this absolute value of their difference is small enough compared to the average absolute value of the center pixel minus neighbor pixel for each of the neighborhoods the pixels belong to. Haralick and Dinstein (1975), however, do report some success using the simpler Euclidean distance on LANDSAT data. The ease with which unwanted region chaining can occur with this technique limits its potential on complex or noisy data.

Hybrid single linkage techniques are more powerful than the simple single linkage technique. The hybrid techniques seek to assign a property vector to each pixel where the property vector depends on the KxK neighborhood of the pixel. Pixels which are similar, are similar because their neighborhoods in some special sense are similar. Similarity is thus established as a function of neighboring pixel values and this makes the technique better behaved on noisy data.

One hybrid single linkage scheme relies on an edge operator to establish whether two pixels are joined with an arc. Here an edge operator is applied to the image labeling each pixel as edge or non-edge. Neighboring pixels, neither of which are edges, are joined by an arc. The initial segments are the connected components of the non-edge labeled pixels. The edge pixels can either be left assigned edges and be considered as background or they can be assigned to the spatially nearest region having a label.

The quality of this technique is highly dependent on the edge operator used. Simple operators such as the Roberts and Sobel operator may provide too much region linkage, for a region cannot be declared as a segment unless it is completely surrounded by edge pixels. Haralick (1982) reports some success with this technique using the zero-crossing of second directional derivative edge operator.

Another hybrid technique first used by Levine and Leemet (1976) is based on the Jarvis and Patrick (1973) shared nearest neighbor idea. Using any kind of reasonable notion for similarity, each pixel examines its KxK neighborhood and makes a list of the N pixels in the neighborhood most similar to it. Call this list the similar neighbor list, where we understand neighbor to be any pixel in the KxK neighborhood. An arc joins any pair of immediately neighboring pixels if there are enough pixels common to their shared neighbor lists; that is, if the number of shared neighbors is high enough.

To make the shared neighbor technique work well each pixel can be associated with a property vector consisting of its own gray tone intensity and a suitable average of the gray tone intensity of pixels in its KxK neighborhood. For example, we can have (x,a) and (y,b) denote the property vectors for two pixels where in the first pixel, x is its gray tone intensity value and a is the average gray tone intensity value in its neighborhood. Likewise, for the second pixel, y is its gray tone intensity value and b is the average gray tone intensity value in its neighborhood. Similarity can be established by computing

$$S = w_1(x-y)^2 + w_2(x-b)^2 + w_3(y-a)^2$$

where w_1, w_2 and w_3 are non-negative weights. The pixels are called similar enough for small enough values of s.

Region Growing / Centroid Li' age

In contrast to single linkage, in centroid linkage pairs of neighboring pixels are not compared for similarity. Rather, a pixel's value is compared to the centroid of an already existing but not necessarily completed segment. If the values are close enough, then the pixel is added to the segment and the segment's centroid is updated. If no neighboring region has a centroid close enough, nen a new segment is established having the given pixel's value as its first member. Such a region growing technique was first suggested by Brice and Finnema (1970). Instead of using the absolute value of the difference as the measure of dis-similarity, Gupta, Kettig, Landgrebe, and Wintz (1973) suggest using the more appropriate t-test.

Simple single pass approaches which scan the image in a left right top down manner are, of course, unable to make the left and right sides of a V-shaped region belong to the same segment. To be more effective, the single pass must be followed by some kind of connected components algorithm in which pairs of neighboring regions having centroids which are close enough are put into the same segment.

One minor problem with centroid linkage schemes is their inherent dependence on the order in which pixels are examined. A left right top down scan does not yield the same initial regions as a right left bottom up scan or for that matter a column major scan. Usually, however, differences caused by scan order are minor.

Histogram Mode Seeking

Histogram mode seeking is a measurement space clustering process in which the clusters in measurement space are mapped back to the image domain where the maximal connected components of the clusters constitute the image segments. For images which are single band images, calculation of this histogram in an array is direct. The measurement space clustering can be accomplished by determining the valleys in this histogram and declaring the clusters to be the interval of values between valleys. A pixel whose value is in the i^{th} interval is labeled with index i and the segment it belongs to is a connected component of all pixels whose label is i.

For multiband images such as LANDSAT, determining the histogram in a multi-dimensional array is nöt feasible. For example, in a six band image where each band has intensitites between 0 and 99, the array would have to have $100^6 = 10^{12}$ locations. A large image might be 10,000 pixels per row by 10,000 rows. This only constitutes 10^8 pixels, a sample too small to estimate probabilities in a space of 10^{12} values were it not for some constraints of reality: (1) there is typically a high correlation between the band to band pixel values and (2) there is a large amount of spatial redundancy in image data. Both these factors create a situation in which the 10^8 pixels can be expected to contain only between 10^4 and 10^5 distinct 6-tuples. Based on this fact, the counting required for the histogram is easily done by hashing the 6-tuple into an array.

Clustering using the multidimensional histogram is more difficult than univariate histogram clustering. Goldberg and Shlien (1977, 1978) threshold the multidimensional histogram to select all N-tuples situated on the most prominent modes. Then they perform a measurement space connected components on these N-tuples to collect together all the N-tuples in the top of the most prominent modes. These measurement space connected sets form the cluster cores. The clusters are defined as the set of all N-tuples closest to each cluster core. A variation on this idea is discussed by Matsumoto, Naka, and Yamamoto (1981)

An alternate possibility is to locate the highest mode and region grow around it in the multi-dimensional measurement space. The region growing includes all successive neighboring N-tuples whose probability is no higher than the N-tuple from which it is growing. This procedure identifies the most prominent mode and its associated mountain as the first cluster core. Then the same procedure is repeated on the remaining N-tuples until all multi-dimensional peaks and their associated cores have been accounted for. The clusters are defined as the set of all N-tuples closest to each core.

Rather than accomplish the clustering in the full measurement space, it is possible to work in multiple lower order projection spaces and than reflect these clusters back to the full measurement space. Suppose, for example, that the clustering is done on a four band image. If the clustering done in bands 1 and 2 yield clusters c_1, c_2, c_3 and the clustering done in bands 3 and 4 yield clusters c_4 and c_5 than each possible 4-tuple from a pixel can be given a cluster label from the set $\{(c_1,c_4), (c_1,c_5), (c_2,c_4), (c_2,c_5), (c_3,c_4), (c_3,c_5)\}$. A 4-tuple (x_1,x_2,x_3,x_4) gets the cluster label (c_2,c_4) if (x_1,x_2) is in cluster c_2 and (x_3,x_4) is in cluster c_4.

Spatial Clustering

It is possible to determine the image segments without performing an independent clustering in measurement space. Such techniques are called spatial clustering. In essence spatial clustering schemes combine the histogram mode seeking technique with the region growing/centroid linkage technique. Haralick and Kelly (1969) suggested it be done by locating, in turn, all the peaks in measurement space. Then determine all pixel locations having a measurement on the peak. Beginning with a pixel corresponding to the highest peak not yet processed, simultaneously perform a spatial and measurement space region growing in the following manner. Initially, each segment is the pixel from which we begin. Consider for possible inclusion into this segment the neighbors of this pixel (in general, the neighbors of the pixel we are growing from) if the neighbor's N-tuple value is close enough in measurement space to the pixel's value and if its probability is not larger than the probability of the pixel's value we are growing from.

<u>Split and Merge</u>

The split method for segmentation begins with the entire image as the initial segment. Then it successively splits each of its current segments into quarters if the segment is not homogeneous enough. Homogeneity can be easily established by determining if the difference between the largest and smallest gray tone intensities is small enough. Algorithms of this type were first suggested by Roberston (1973) and Klinger (1973)

Because segments are successively divided into quarters the boundaries produced by the split technique tend to be squareish and slightly artificial. Sometimes adjacent quarters coming from adjacent split segments need to be joined rather than remain separate. Horowitz and Pavlidis (1976) suggest a split and merge strategy to take care of this problem Chen and Pavlidis (1980) suggested using statistical tests for uniformity rather than examination of the difference between largest and smallest gray tone intensities.

The data structures required to do a split and merge on images larger than 512x512 are extremely large. Execution of the algorithm on virtual mememory computers results in so much paging that the dominant activity is paging rather than segmentation. Browning and Tanimoto (1982) give a description of a split and merge scheme where the split and merge is first accomplished on mutually exclusive subimage blocks and the resulting segments are then merged between adjacent blocks to take care of the artificial block boundaries.

References

T. Asano and N. Yokoya, "Image Segmentation Schema For Low-Level Computer Vision," Pattern Recognition, Vol. 14, No. 1, 1981, pp.267-273.

C. Brice and C. Fennema, "Scene Analysis Using Regions," Artificial Intelligence, Vol. 1, 1970, p205-226.

J.D. Browning and S.L. Tanimoto, "Segmentation of Pictures into Regions With a Tile by Tile Method," Pattern Recognition, Vol. 15, No 1, 1982, pp.1-10.

P.C. Chen, and T. Pavlidis, "Image Segmentation as an Estimation Problem," Computer Graphics and Image Processing, Vol. 12, 1980, pp. 153-172.

M. Goldberg and S. Shlien, "A Clustering Scheme For Multispectral Image," IEEE Systems, Man, and Cybernetics, Vol. SMC-8, No 2, 1978, pp.86-92.

M. Goldberg and S. Shlien, "A Four-Dimensional Histogram Approach to the Clustering of LANDSAT Data," Machine Processing of Remotely Sensed Data, IEEE CH 1218-7 MPRSD, Purdue University, West Lafayette, Indiana, June 21-23, 1977, pp.250-259.

J.N. Gupta, R.L. Kettig, D.A. Landgrebe, and P.A. Wintz, "Machine Boundary Finding and Sample Classification of Remotely Sensed Agricultural Data," Machine Processing of Remotely Sensed Data, IEEE 73 CHO 834-2GE, Purdue University, West Lafayette Indiana, October 16-18, 1973, p4B-25 to 4B-35.

R.M. Haralick and G.L. Kelly, "Pattern Recognition with Measurement Space and Spatial Clustering for Multiple Image," Proceedings of IEEE, Vol. 57, No. 4, April 1969, pp.654-665.

R.M. Haralick and I. Dinstein, "A Spatial Clustering Procedure for Multi-Image Data," IEEE Circuits and Systems, Vol. CAS 22, No. 2, May 1975, pp.440-450.

R.M. Haralick, "The Digital Step Edge," to be published in 1982.

S.L. Horowitz and T. Pavlidis, "Picture Segmentation by a Tree Traveral Algorithm", Journal of the Association for Computing Machinery, Vol 23, No 2, April 1976, pp.368-388.

R.A. Jarvis and E.A. Patrick, "Clustering Using A Similarity Measure Based on Shared Near Neighbors," IEEE Transactions on Computers, Vol. C-22, No. 11, 1973, pp.1025-1034.

A. Klinger, "Data Structures and Pattern Recognition," 1st International Joint Conference on Pattern Recognition, Washington D.C., Oct 1973 pp.497-498.

M.D. Levine and J. Leemet, "A Method For Non-Purposive Picture Segmentation," 3rd International Joint Conference on Pattern Recognition, 1976.

K. Matsumoto, M. Naka, and H. Yamanoto, "A New Clustering Method For LANDSAT Images Using Local Maximums of a Multi-Dimensional Histogram," Machine Processing of Remotely Sensed Data, IEEE CH 1637-8 MPRSD, Purdue University, West Lafayette, Indiana, June 23-26, 1981, p321-325.

T.V. Robertson, "Extraction and Classification of Objects In Multispectral Images," Machine Processing of Remotely Sensed Data, IEEE 73 CHO 837-2GE, Purdue University, West Lafayette Indiana, October 16-18, 1973, p3B-27 to 3B-34.

N83 15768

Dz

Cooperative processes in image segmentation

Larry S. Davis

This talk will survey recent research into the role of cooperative, or relaxation, processes in image segmentation. Cooperative processes [1] can be employed at several levels of the segmentation process as a preprocessing enhancement step [2,3,4], during supervised or unsupervised pixel classification [4,5] and, finally, for the interpretation of image segments based on segment properties and relations [6].

1. L. Davis and A. Rosenfeld, "Cooperative processes for low-level vision: A survey," A.I. 17, 245-263, 1981.

2. L. Davis and A. Rosenfeld, "Noise cleaning by iterated local averaging," IEEET-SMC, 8, 706-710, 1978.

3. L. Davis and A. Mitiche, "MITES: A new tool for image segmentation," to appear in CGIP.

4. K. Narayanan and A. Rosenfeld, "Image smoothing by local use of global information," IEEET-SMC, December 1981.

5. L. Davis, C. Wang and H. Xie, "Some experiments in multispectral, multi-temporal crop classification using relaxation," Univ. of Maryland Computer Science TR-1131, December 1981.

6. L. Kitchen, "Scene analysis by region based constraint propagation," in preparation.

Cooperative Processes in Image Analysis

Larry S. Davis
Computer Vision Laboratory
University of Maryland
College Park, MD 20742

COOPERATIVE PROCESSES

GOAL: ASSIGN SYMBOLIC AND NUMERICAL LABELS TO PICTURE PARTS

- ASSIGN SYMBOLIC LAND-USE CATEGORIES TO PIXELS
- ASSIGN NUMERIC STEREO DISPARITY LABELS TO PIXELS

CONSTRAINT: IMAGES ARE LARGE, SO LABELING PROCESS MUST BE FAST

- SEQUENTIAL: TOO SLOW
- PARALLEL: TOO ERROR PRONE

<u>SOLUTION</u>: COOPERATIVE PROCESSES

- ASSESS EACH PART INDEPENDENTLY (PARALLEL)
- COMPARE ASSESSMENTS OF "RELEVANT" PARTS (PARALLEL)
    - COMPARISONS MUST BE LOCAL
    - ENTIRE PROCESS IS ITERATIVE

ORGANIZATION

A) INITIAL, INDEPENDENT PART ASSESSMENT

B) ADJUSTMENTS OF ASSESSMENTS BASED ON RELATIONSHIPS BETWEEN PARTS

C) ITERATION OF STEP (B)

2. Application to Motion Detection

Assume

1) Image intensity is a continuous, differentiable function $f(x,y,t)$
2) The intensity corresponding to any given scene point does not change over time
3) Both the motion (u,v) and the time interval, τ, between frames is small enough that a Maclauren series expansion is a good local approximation to the picture function.

Approximate $f(x+u,y+v,t+\tau)$ by a series expansion about (x,y,y) (which can be regarded as $(0,0,0)$).

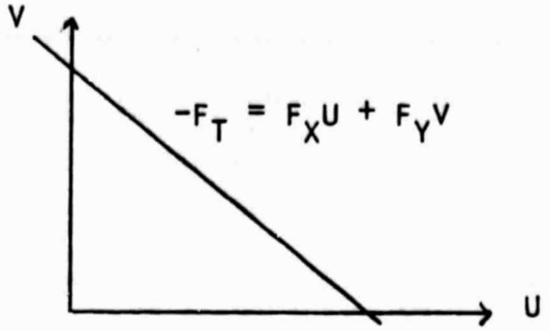
$$f(x+u,y+v,t+\tau) = f(x,y,t)+f_x\cdot u+f_y\cdot v+f_T\cdot\tau$$
$$+\text{higher order terms}$$

Arbitrarily set $\tau=1$ and note that (2) implies $f(x+u,y+v,t+\tau) = f(x,y,t)$

$-F_T = F_X U + F_Y V$: MOTION CONSTRAINT

F_T = TEMPORAL DERIVATIVE

F_X, F_Y = SPATIAL DERIVATIVES



NOTE:

1) IF $F_X, F_Y \approx 0$, THEN MOTION INFORMATION CANNOT BE ACCURATELY DETERMINED

2) IF $F_X = 0$, THEN $-F_T = F_Y V$ SO THAT V IS DETERMINED BUT U IS UNKNOWN

3) LET G AND ℓ DENOTE THE GRADIENT AND LEVEL DIRECTIONS AT A PIXEL.

$$G = \tan^{-1} F_Y / F_X$$
$$\ell \perp G, \; F_\ell = 0$$

THEN $F_T = F_G \, D_G / D_T$ SO COMPONENT OF VELOCITY IN THE GRADIENT DIRECTION IS KNOWN, BUT NOT IN THE LEVEL DIRECTION.

2. Rough estimates by Pseudo-i

ORIGINAL PAGE IS
OF POOR QUALITY

Combining Motion Constraints

1. Assume that u,v are constant over small regions of the image and obtain a local least square solution for u,v.

   C. Cafforio and F. Rocca, "Tracking moving objects in T.V. Images," Signal Proc., 1, 1979, 133-140.

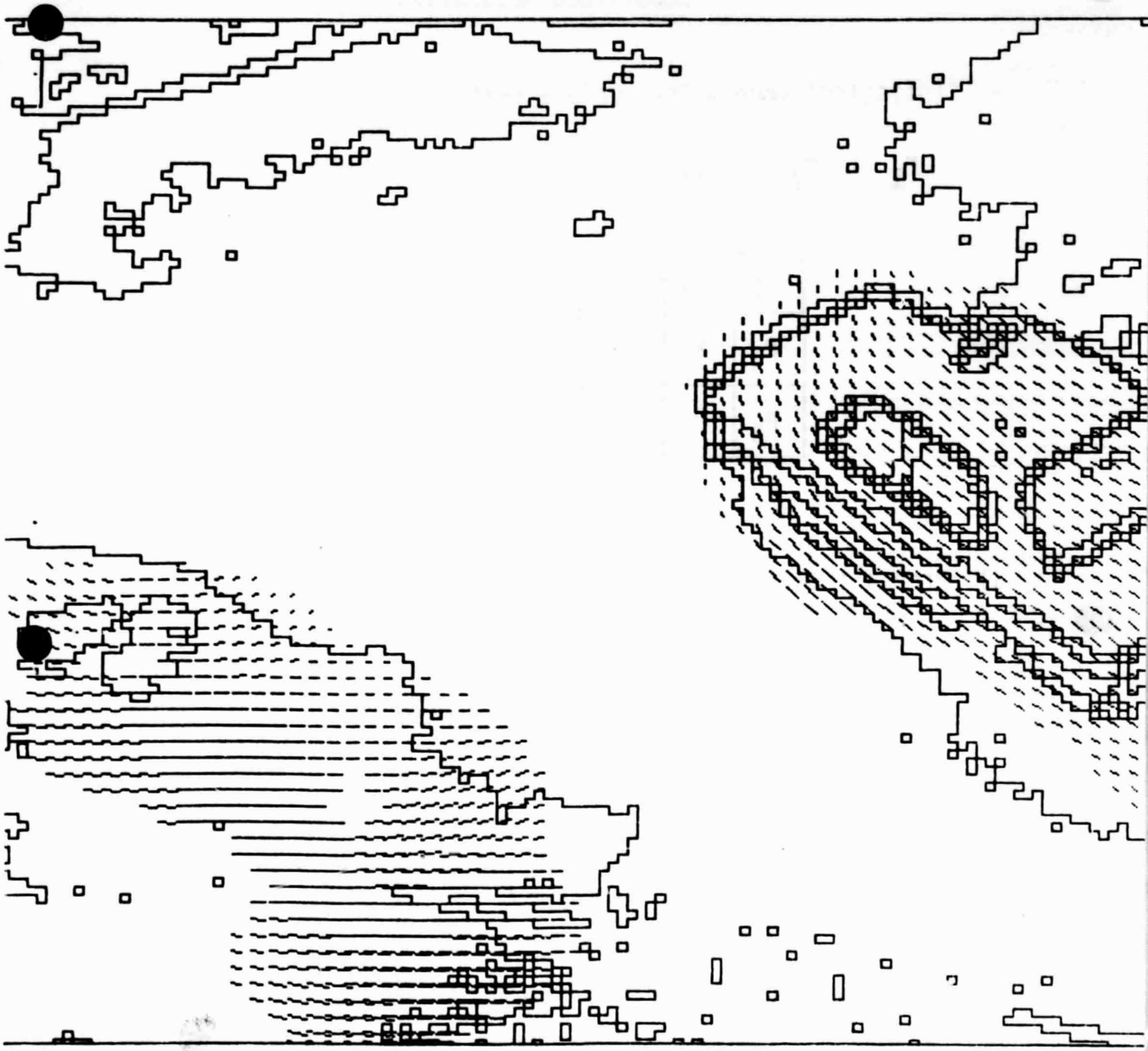
   J. Limb and J. Murphy, "Estimating the velocity of moving images in T.V. signals," CGIP, 4, 1975, 311-327.

2. Assume that u,v vary smoothly over the image and use relaxation techniques to compute "optimal" values at u,v.

   B.K.P. Horn and B. Schunck, "Determining optical flow," Artificial Intelligence, 17, 1981, 185-203.

Feb 27 15:42 1982 vec4.7 7 iterations

ORIGINAL PAGE IS
OF POOR QUALITY



Supervised Relaxation

- Statistical models for various textures

$M_1 \quad M_2 \quad \ldots \quad M_n$

P_1	P_2	P_3
P_4	X	P_5
P_6	P_7	P_8

1) $\text{Prob}[X \text{ interior to } M_i] = \prod_{i} \text{prob}[p_j \in M_i] \cdot \text{prob}[x \in M_i]$

2) $\text{Prob}[X \in M_i$ and on a vertical edge separating

M_j from $M_i] =$

$$\prod_{j=1,4,6} \text{prob}[p_j \in M_j] \cdot \prod_{j\neq 1,4,6} \text{prob}[p_j \in M_i] \cdot \text{prob}[X \in M_i]$$

$\vdots$

Label Set = {m_1, m_2, m_3, ... , m_n

V_{12}, V_{23}, ... , $V_{n-1,n}$,

H_{12}, H_{23}, ... , $H_{n-1,n}$}

Iterative procedure:

For each pixel, X:

1) compute the most probable label for X

2) Smooth X "appropriately"

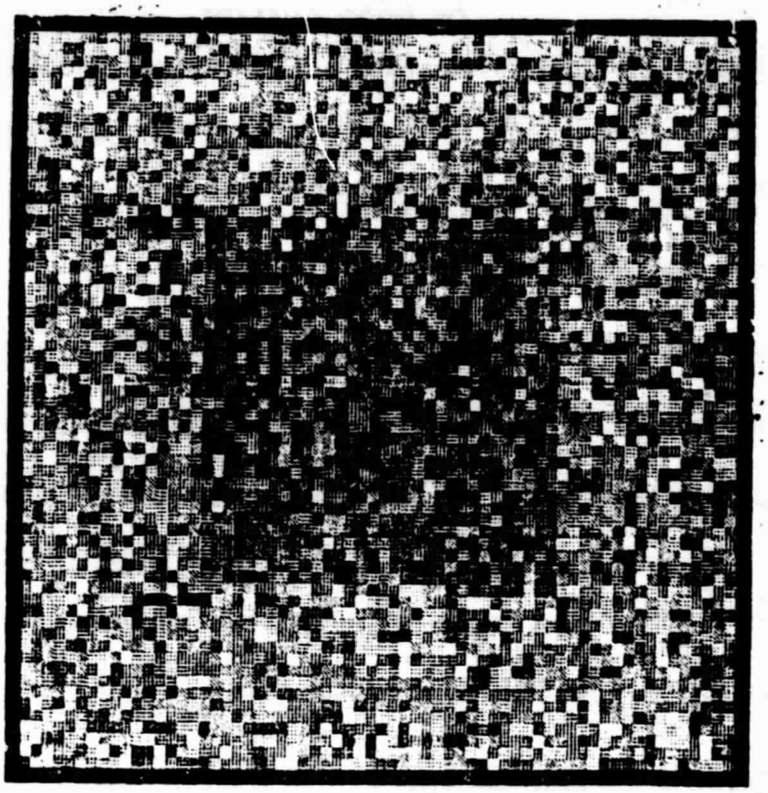
EX: 1) X interior - average with all neighbors

2)

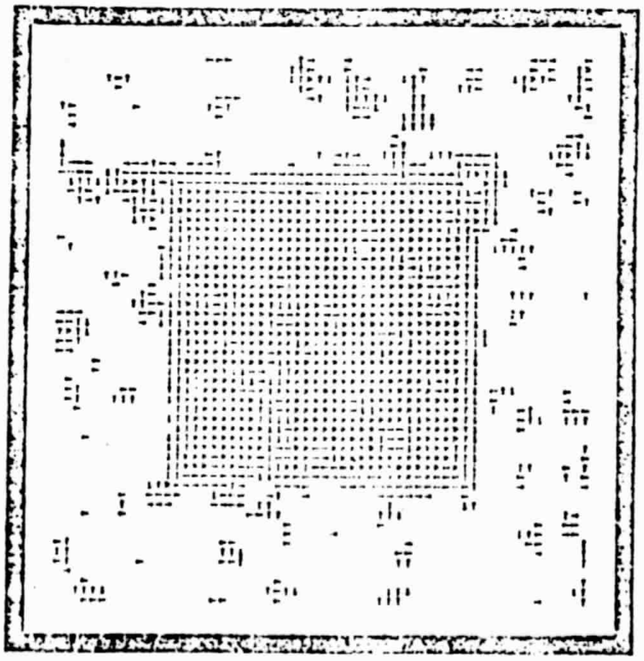
P_1 M_1	P_2 M_1	P_3 M_4
P_4 M_1	X M_1	P5 M_4
P_6 M_1	P_7 M_1	P_8 M_4

average X with

{P_1,P_2,P_4,P_6,P_7}

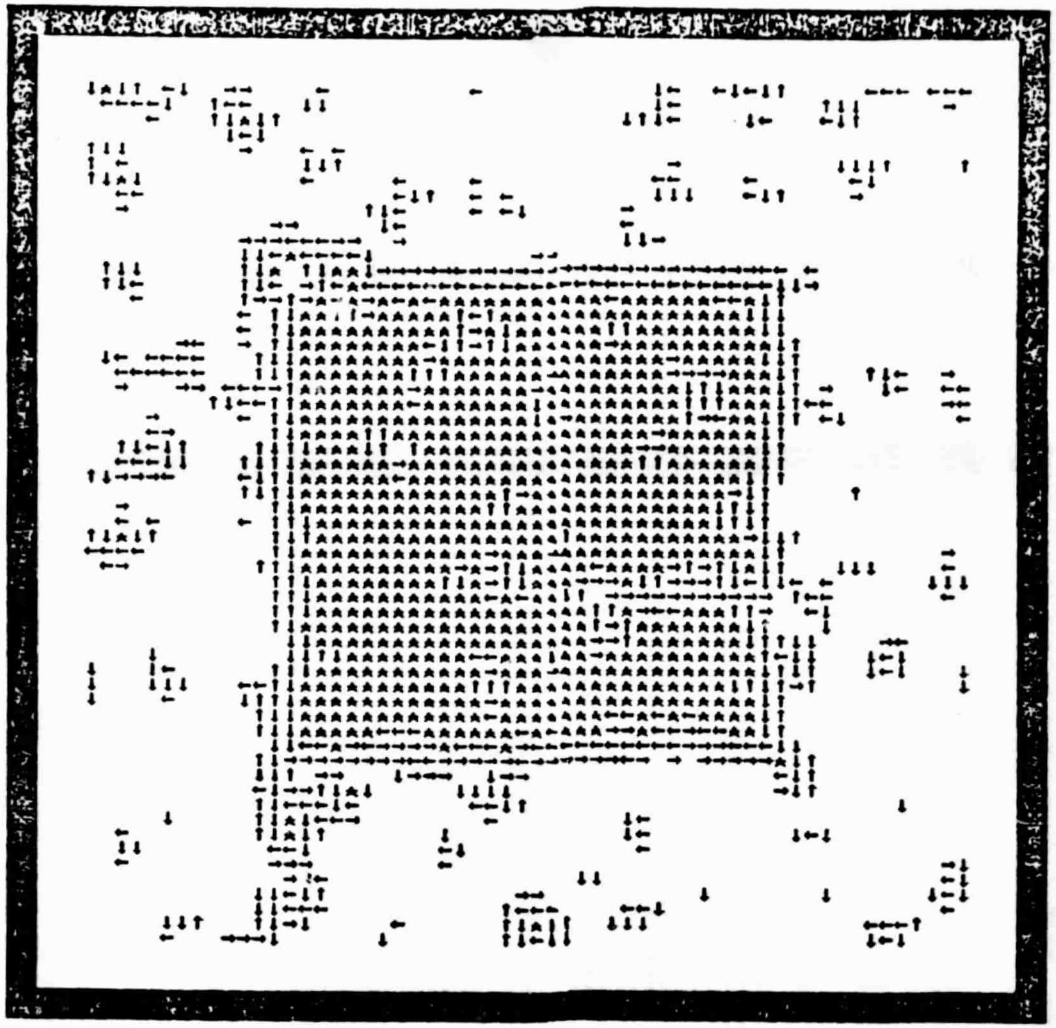


(a)



(b)

Figure 3. Noisy squares (a) and initial labelling (b).

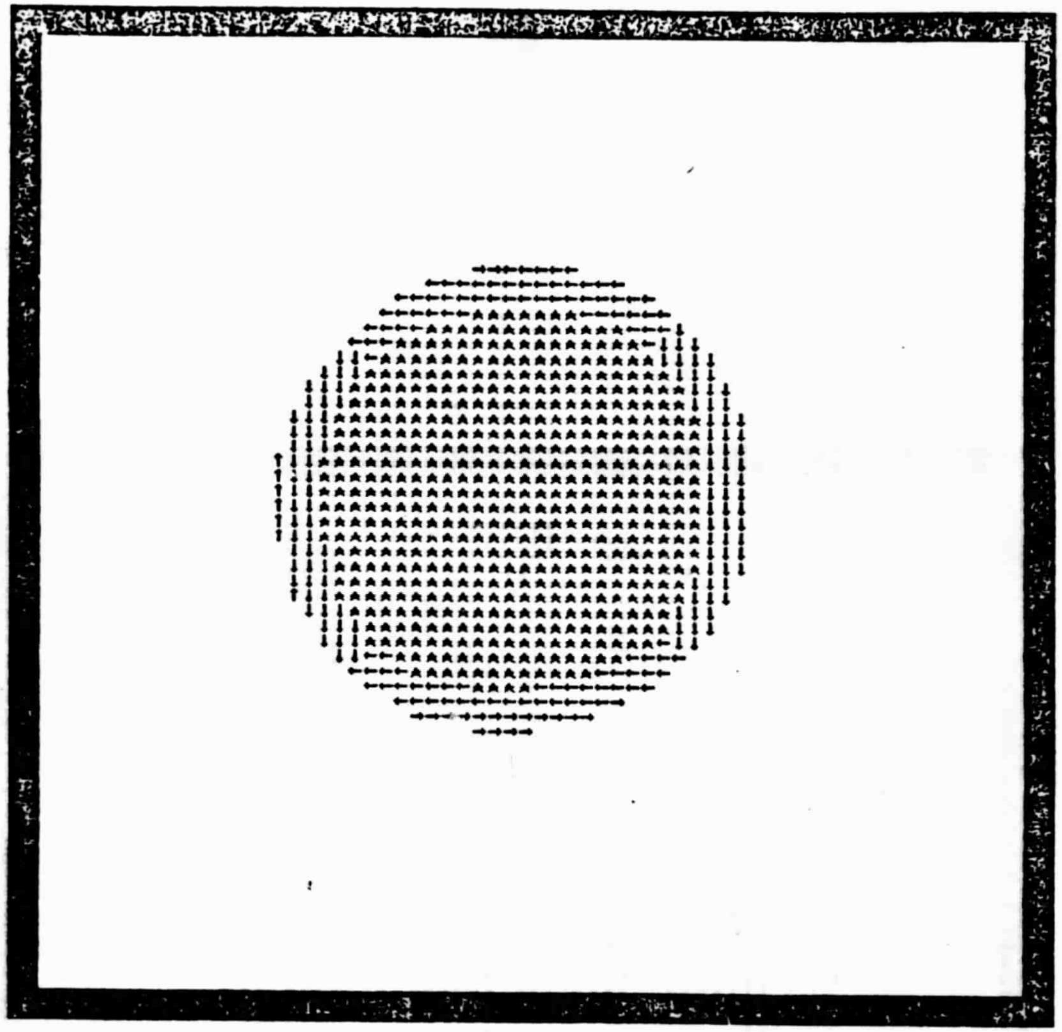


INITIAL LABELLING, NOISY SQUARES, MITES (3 x 3)

Fig 3b

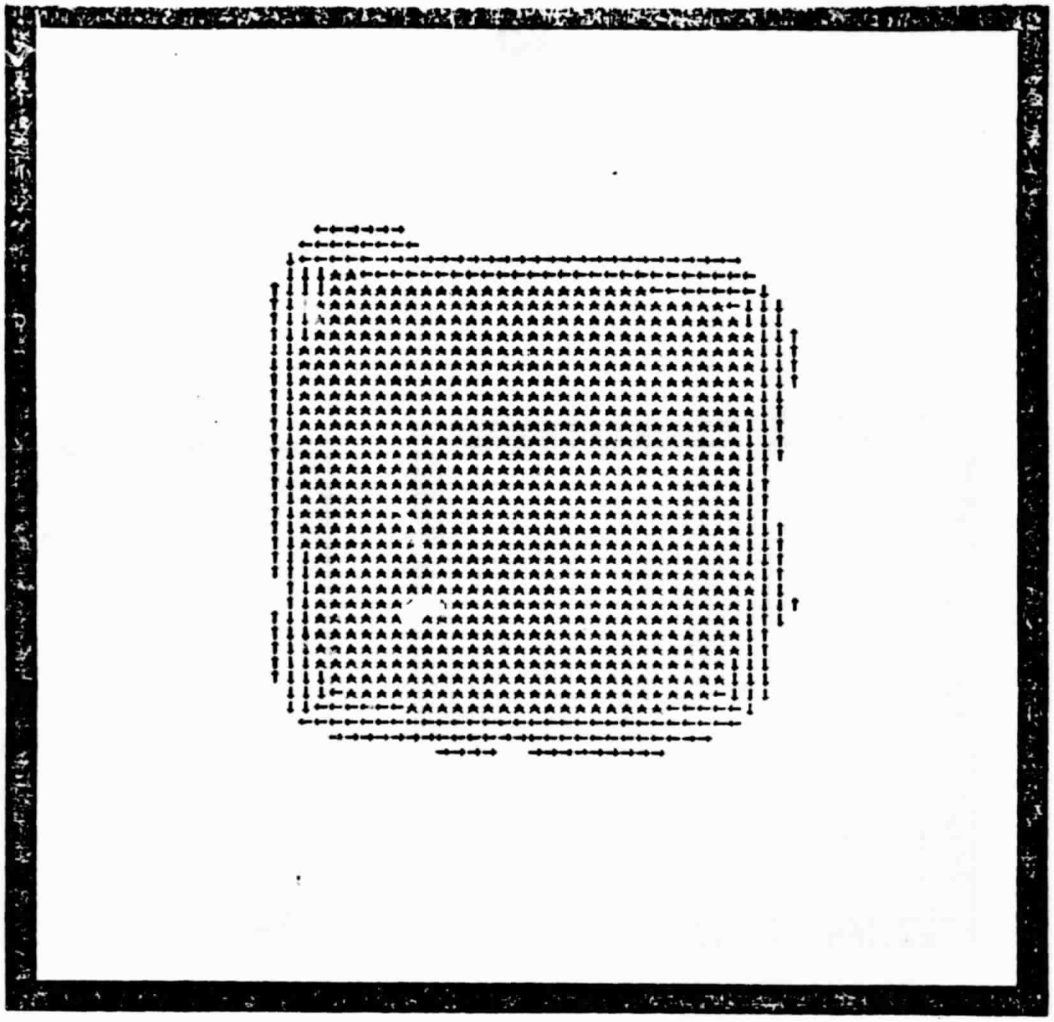
ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY



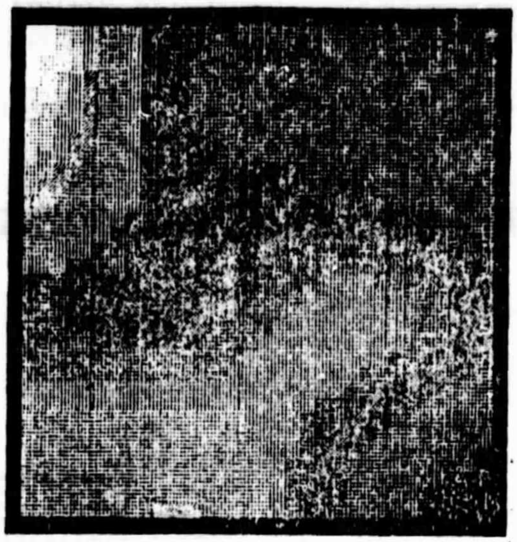
15TH ITERATION ON NOISY SQUARES: 7 x 7 SIMPLE SMOOTHING

Figure 13b

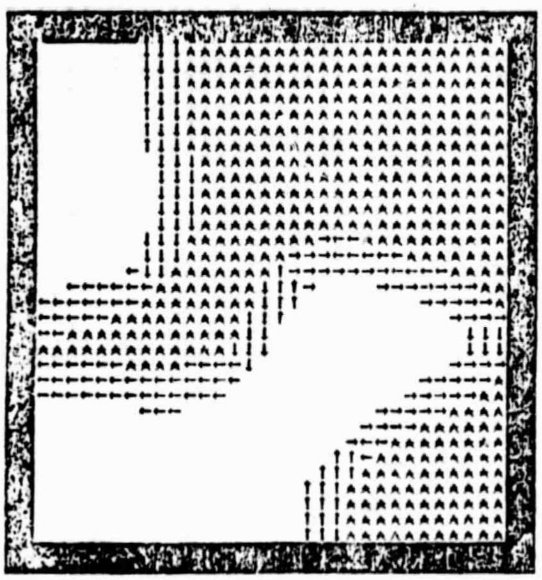


15TH ITERATION ON NOISY SQUARES: MITES 7 x 7

Figure 15a

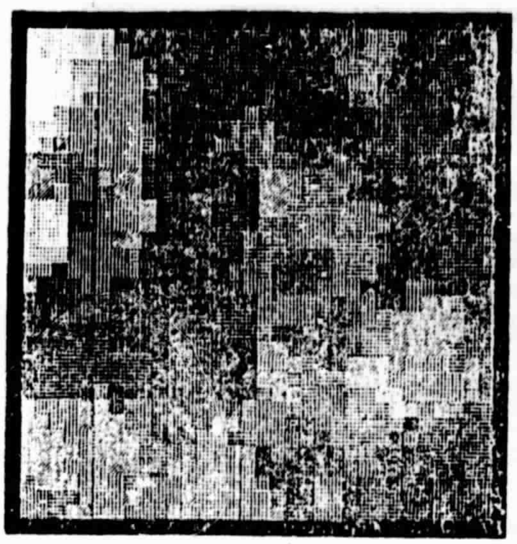


(a)

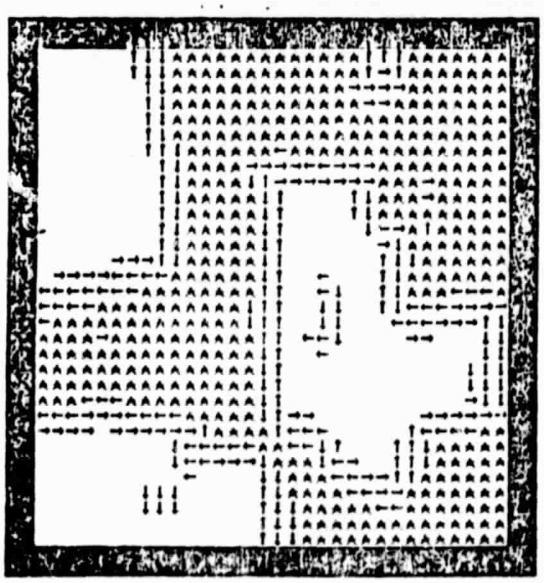


(b)

Figure 32. 15th iteration smoothing (a) and labelling (b). 3 x 3 simple smoothing.



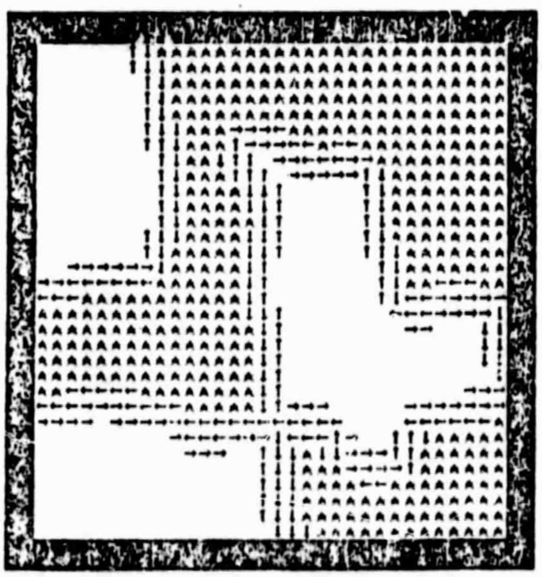
(a)



(b)

Figure 33. 15th iteration smoothing (a) and labelling (b) -E^5.

(a)

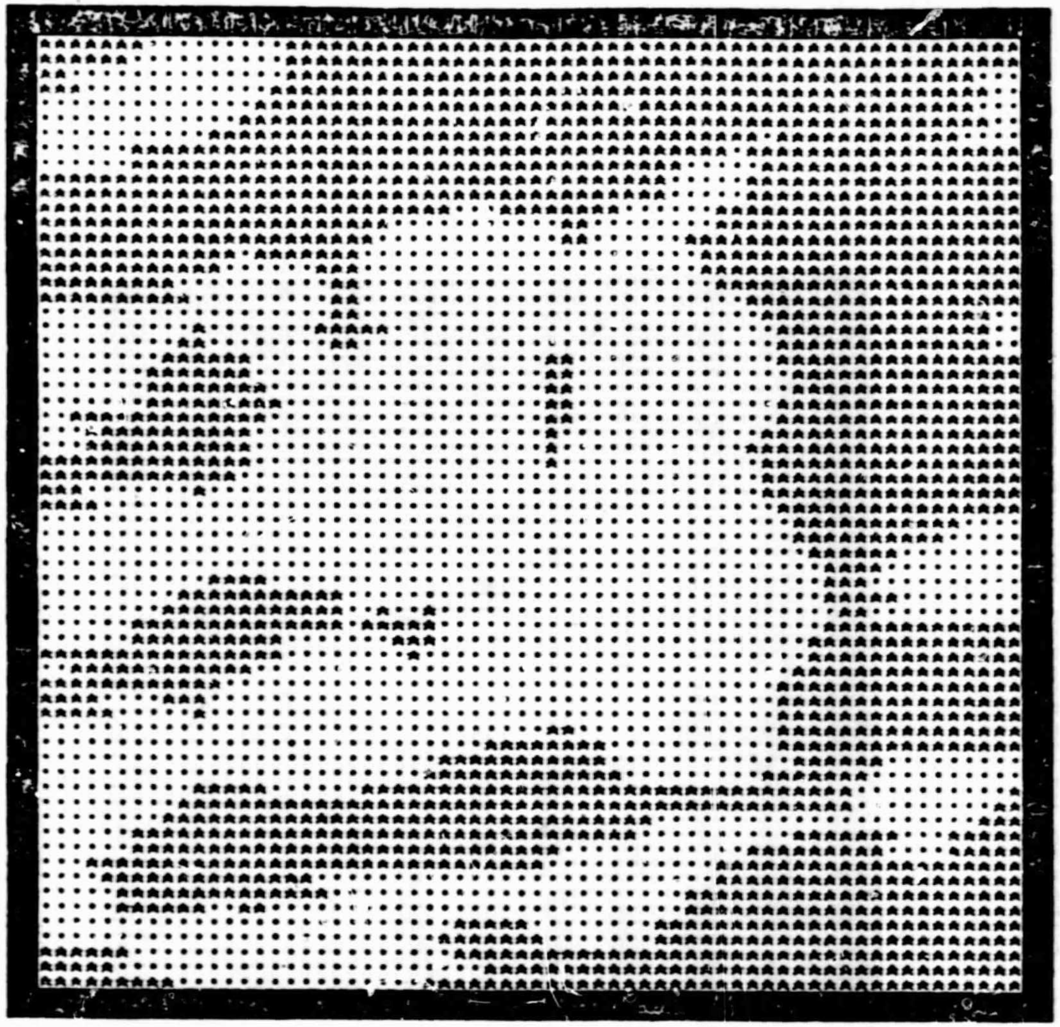


(b)

Figure 31. 15th iteration smoothing (a) and labelling (b) MITES 3 x 3.

(a)

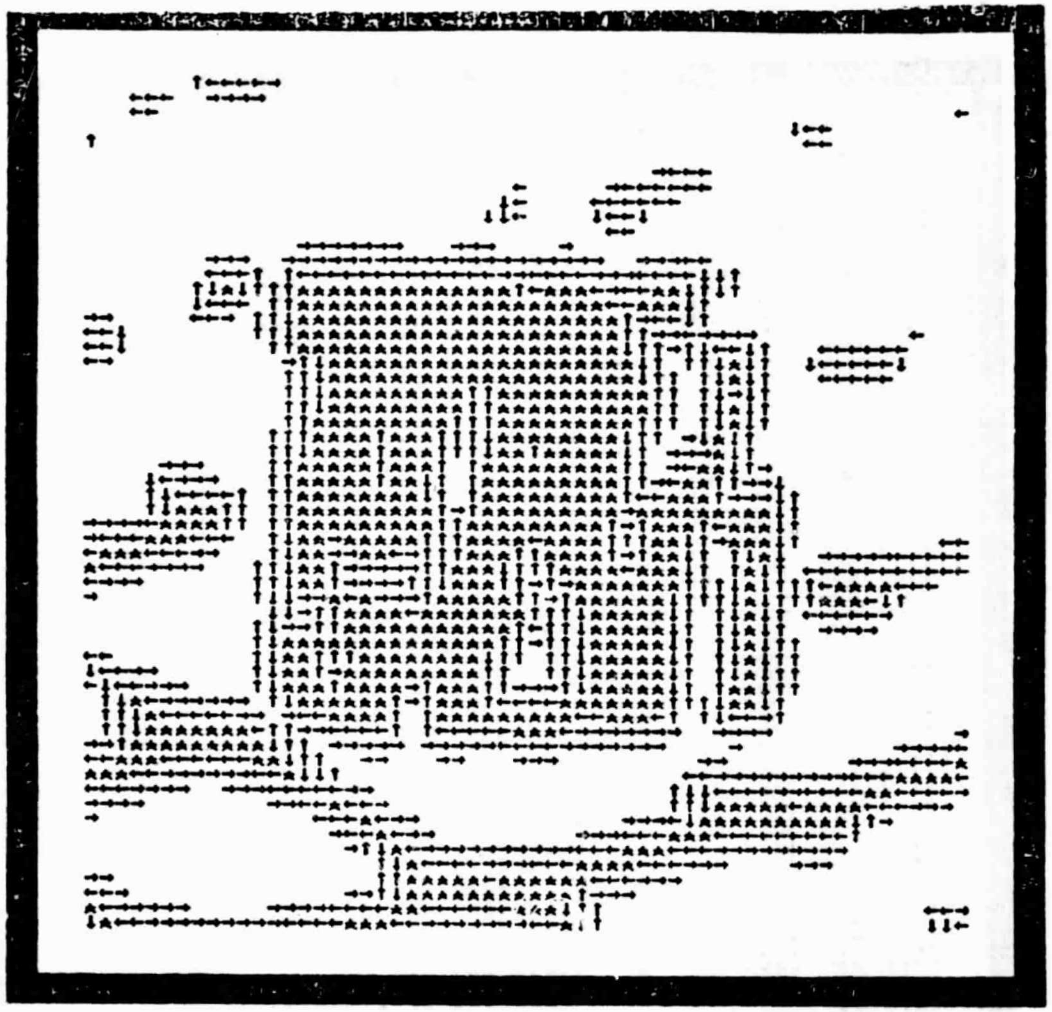
Figure 19 (a) Concrete in Grating
(b) Grating left and Bricks right



UMITES

E/G Initial Cluster on Contrast

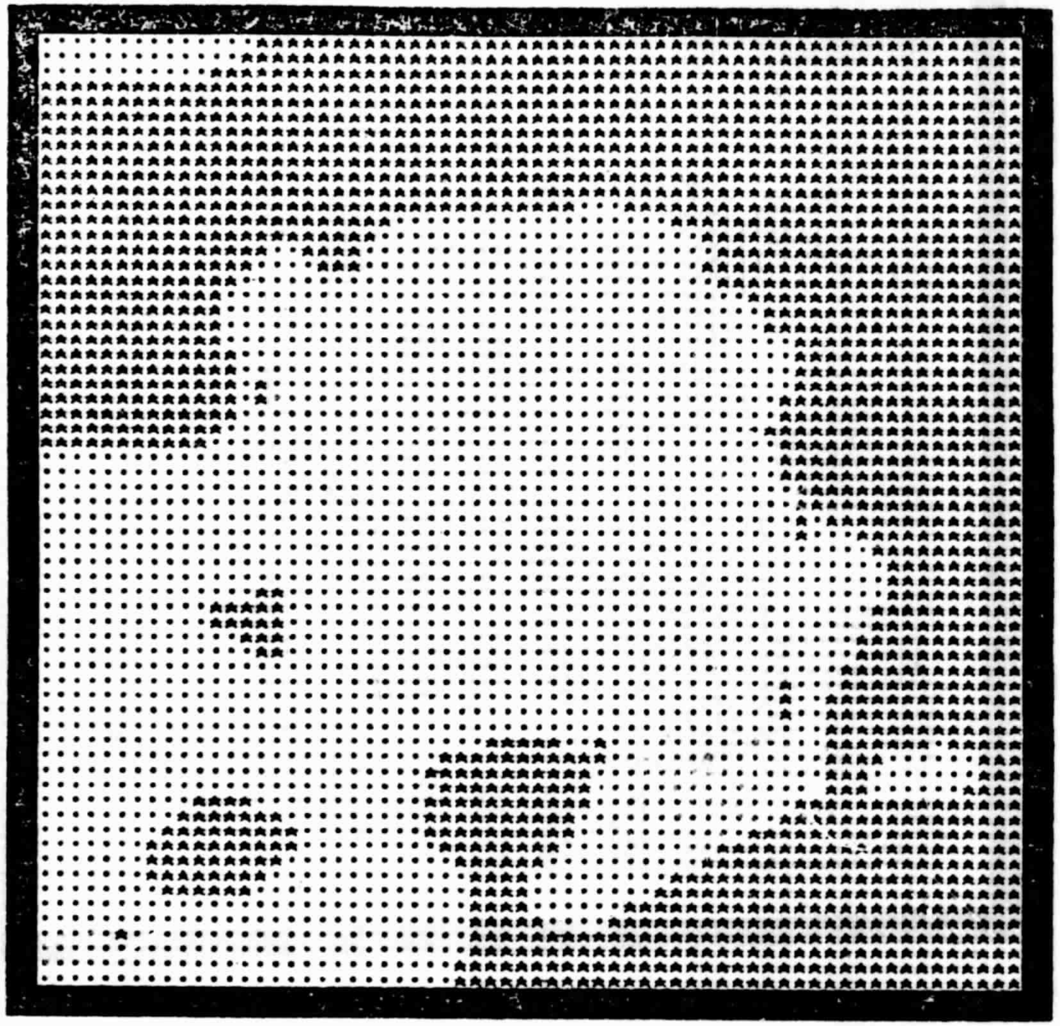
Fig 20a



UMITES

E/G INITIAL LABELLING

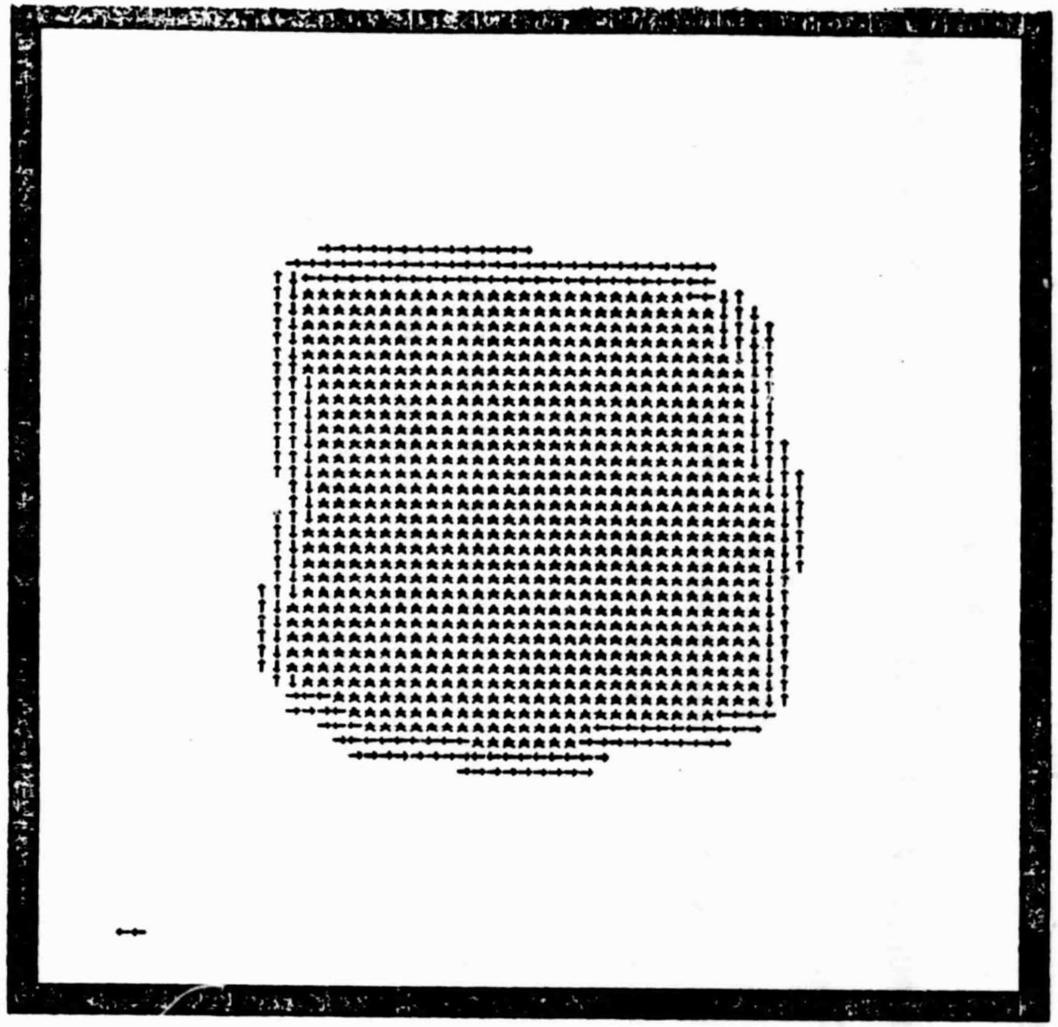
Fig 22a



UMITES

E/G 7TH ITERATION CLUSTER ON CONTRAST

Fig 20c



UMITES

E/G 7TH ITERATION LABELLING

Fig 22C

MULTI-TEMPORAL IMAGE SEGMENTATION

OBJECTS - PIXELS

LABELS - REGION CLASS NAMES

RELAXATION RULES CAN BE HEURISTICALLY DERIVED FROM THE FOLLOWING OBSERVATION:

FIELDS ARE, IN GENERAL, MUCH BIGGER THAN A SINGLE PIXEL AND DO NOT CHANGE LABELS DURING A SINGLE GROWING SEASON.

$P_{\lambda}^{K}(I,J,T)$ - PROBABILITY THAT THE PIXEL AT SPATIAL LOCATION (I,J,T) AND TIME T IS IN CLASS λ AFTER K ITERATIONS OF THE RELAXATION RULE

RELAXATION RULES

RULE SCHEMA

$$P_\lambda^{K+1}(I,J,T) = \frac{1}{N} P_\lambda^K(I,J,T) \times \prod_{T'\neq T} P_\lambda^K(I,J,T')$$

$$*F(\{P_{\lambda'}^K(I',J',T): \lambda' \in L, (I',J') \text{ A NEIGHBOR OF } (I,J)\})$$

SPECIFIC RULES

$$F_1 = \max_{I=0}^{7} P_\lambda^K(X_I, T)$$

X_7	X_0	X_1
X_6	(I,J)	X_2
X_5	X_4	X_3

$$F_2 = \max_{J=0}^{7} \prod_{I=0}^{2} P_\lambda^K(X_{I+J}, T)$$

$$F_3 = \sum_{J=0,2,4,6} P_\lambda^K(X_J, T)$$

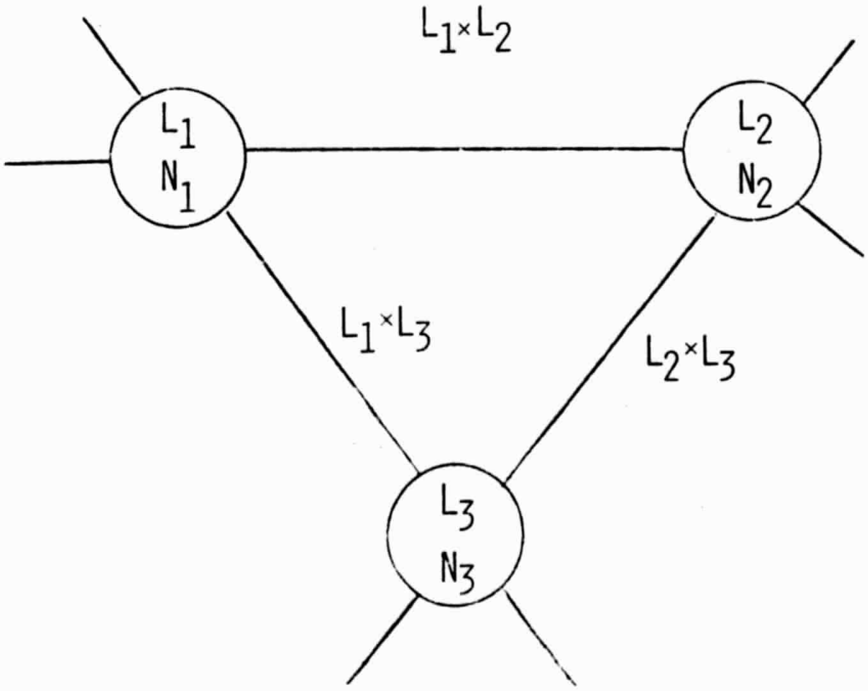
SCENE ANALYSIS - TANKS WORLD

TANKSWORLD

GROUND
SKY
SMOKE
TANK
TREE
TREE FRAGMENTS

} LABELS

UNARY CONSTRAINTS
BINARY CONSTRAINTS
EXISTENTIAL CONSTRAINTS



EXAMPLES

BINARY CONSTRAINTS (ARCS)

TANK CANNOT SURROUND SMOKE

IF (TANK,SMOKE) $\in L_I \times L_J$ AND IF N_I SURROUNDS N_J, THEN DELETE (TANK,SMOKE) FROM $L_I \times L_J$.

EXISTENTIAL CONSTRAINTS

A TREE FRAGMENT MUST BE ADJACENT TO A TREE FRAGMENT

IF TREE FRAGMENT $\in L_I$ AND IF FOR NO ADJACENT L_J IS (TREE FRAGMENT,TREE FRAGMENT) $\in L_I \times L_J$, THEN ELIMINATE TREE FRAGMENT FROM L_I.

COOPERATIVE PROCESSES IN IMAGE ANALYSIS

L. S. Davis

Computer Vision Laboratory, Computer Science Center, University of Maryland, College Park, Maryland 20742

Abstract. This paper contains an overview of the organization of cooperative (or relaxation) processes for low-level vision. Two examples (one involving pixel classification and the other motion disparity estimation) are used to illustrate the various steps in applying a relaxation algorithm to image analysis problems.

Note: This paper also appears in the Proceedings of IFAC 82.

1. ISSUES

Many image analysis problems can be regarded as problems of assigning a label to each element in a set of picture parts (pixels or regions). For example, pixels can be assigned symbollic land-use category labels based on their spectral signatures, or numerical motion disparity labels based on local comparisons amongst consecutive frames in a time-varying image. Both of these problems will be used as examples throughout this paper.

The large number of pixels in a digital image demands that such labelling processes be very fast. One obvious solution to this problem is to make the labelling process highly parallel. However, in a parallel process, each picture part would be analyzed independently of all other picture parts. Thus, parallel processes fail to make use of contextual information (which is often available), and make many labelling errors.

In order to overcome this problem, one can assess the labelling possibilities for every part independently, and then compare each part's assessments to those of other, related parts, in order to correct inconsistencies. Since both the assessment and the comparison can be done independently for every part, each stage of the process is parallel. On the other hand, context is now being used at the comparison stage, when related parts are able to communicate and 'cooperate'. To keep the computational cost low, the comparisons should be local; they should involve only parts that are directly related (e.g., neighboring pixels). This localness can be compensated for by iterating the comparison process, in order to allow information to propagate.

These considerations lead naturally to the design of a 'cooperative' approach to labelling picture parts which allows context to be used in the labelling process while still permitting fast parallel implementation and low computational cost. Such processes are called 'relaxation' processes, because of their resemblance to certain iterative processes used in numerical analysis. Very generally, a relaxation process is organized as follows:

(a) A list of possible labels is independently selected for each part, based on its intrinsic characteristics. A measure of confidence can also be associated with each possible label.

(b) The possibilities (and confidences) for each part are compared with those for related parts, based on a model for the relationships between the possible labels of picture parts. Labels are deleted or modified or confidences are adjusted, to reduce inconsistencies.

(c) Step (b) can be iterated as many times as required.

This approach is very general: We have not specified how to formulate label relationship models, choose possibilities, estimate confidences, or adjust them; nor have we discussed when the process should be iterated, and if so, how many times.

These issues are discussed in some detail in [1], and, in general, are a function of the problem at hand. For a general discussion, the reader should consult [1]. We will, instead, consider two specific problems in detail - pixel classification and motion disparity estimation. The first involves a symbolic label set and the second a numeric one. The first involves confidence adjustment and the second, additionally, involves modification of numeric labels. Sections 2,3, and 4 will consider steps a,b, and c above for these two problems.

2. INITIAL LABEL ASSIGNMENT

The first step of a relaxation process involves assigning initial labels to each picture part. If the given label set is symbollic, then this is ordinarily done using techniques from statistical pattern recognition. Here, measurements are computed for each picture part and, based on a priori models of the class conditional densities of these measurements a probability that each label is the correct one for a picture part can be computed. These probabilities serve as the measures of confidence referred to in Section 1. Some examples of symbollic label sets, L, and associated picture part measurements, M, are:

(1) In a "spring-loaded" template matching problem [2] L would be the set of subtemplate names and M the cross-correlation of the subtemplate at a particular image position. See [3] for details.

(2) In dot cluster detection, L = {interior point, edge point, noise point} and M are measurements of local dot density. See [4] for details.

(3) In supervised pixel classification, L is the set of given class names, and M ordinarily contains spectral and, possibly, local textural features.

Pursuing the pixel classification in more detail, Figure 1a-g shows six images of a LACIE test site (1473). These six images are from three time acquisitions and two bands (2 and 3). Ground truth is available for this site, and it can be used to compute the class conditional probability densities at each time for the 2-vector of measurements at each time. By modeling these densities as normal, we can adopt the following simple procedure for determining class probabilities for each pixel at each time:

(1) Compute the following distance measure of a pixel, x, from a class, d:

$$d_\lambda(x) = \log|\Sigma_\lambda| + (x-y)^T \Sigma_\lambda^{-1}(x-y)$$

Here, y_λ is the mean vector for class λ and Σ_λ is its covariance matrix.

(2) Next, compute the probability that pixel x belongs to class λ by:

$$p_\lambda(x) = [1/d_\lambda(x)]/\sum_{\lambda'}[1/d_{\lambda'}(x)]$$

If one were forced to decide on a fixed label for a pixel, x, then one would choose the x for which p (x) is maximal. As mentioned earlier, such an independent classification contains many errors. Figure 2a-i illustrates this. In the next section we describe how a simple relaxation process can improve on these results.

In many applications, the natural label set is a numeric property value rather than symbollic, and the label set at each picture part is often initially represented by the most current estimate of the most likely property value at that picture part and a measure of confidence often related to the variance of local property values. For example, in relaxation algorithms for grey level image enhancement, the initial property value at each pixel is the pixel's grey level, and the measure of confidence, for piecewise constant images, would be the grey level variance in some neighborhood of the picture.

A more complicated example is motion disparity estimation. Here, the initial property value is a motion vector which can be computed as follows. At each pixel, one can compute a linear constraint on the x and y components (u,v) of that pixel's motion based on the equation:

$$-I_t = I_x u + I_y v \qquad (1)$$

Here, I_t is the temporal image intensity change and I_x and I_y represent the spatial image intensity gradient. We assume here that grey level is an invariant of motion. Assuming that the image motion is locally a translation, then one could combine the equations at two adjacent points to obtain a unique velocity vector at each point, or, more generally, one can compute the "pseudo-intersection" of the linear constraints in a neighborhood of each pixel [5]. In this case, the error of the fit can be used as a measure of confidence at each pixel, although we will ignore the confidences in what follows.

As an example, Figure 3 contains two frames in a time varying image, and Figure 4 contains the initial velocity vector computed by the pseudo-intersection technique.

3. RELAXATION

After initial labellings (and confidences) are computed, one begins a sequence of iterations where the labels and confidences at each picture part are modified based on the distribution of labels and confidences on related picture parts and some model of how labellings affect one another. Thus, the relaxation algorithm is based on:

a) a model for the neighborhood of a picture part, and

ORIGINAL PAGE IS
OF POOR QUALITY

b) a model for the interactions between labellings of adjacent picture parts.

The neighborhood model for a relaxation process specifies which pairs of picture parts directly communicate with one another in the relaxation process, and determines the topology of the graph on which the relaxation process operates. This graph has individual picture parts as nodes. Its arcs connect those pairs of parts that communicate with one another. The neighborhood model is usually designed to establish connections only between 'nearby' parts.

A neighborhood model is specified by a set of neighbor relations $r = \{r_1, r_2, \ldots, r_n\}$. Each r_i is a binary relation defined over the appropriate set of picture parts. For example, if the picture parts are pixels, then the neighborhood model might specify that a pixel is connected to every pixel its 3×3 neighborhood. In this case, there are still several possibilities for the relations contained in the set r. For example, r might be the set {directly above, directly below, etc.} which would distinguish between pairs of points that are horizontally adjacent, vertically adjacent, etc., or it could be the singleton relation "in the 3×3 neighborhood". In the latter case, the connections between pairs of pixels would not be recoverable from the graph on which the relaxation process will operate. The choice of r will, in general, be determined by the isotropy of the universe of labels. For example, if we are designing a relaxation process for edge reinforcement, then the relative positions of pixels are crucial since edges generally 'line up', while if we are designing a relaxation process to enhance an image's grey levels, then the positional information may not be required.

When the picture parts are regions rather than pixels, then connections might be formed between adjacent regions only. In some situations, it might be necessary to distinguish between regions that are above, below, inside, surrounding, etc.

The interaction model defines how a picture part changes its labelling based on the labellings of its neighbors. An interaction model is composed of two parts:

(1) a knowledge representation for the relationships between labels, and

(2) a mechanism, or procedure, for applying the knowledge in (1) to change, or update, labellings.

For discrete symbollic labellings the simplest knowledge representation is a set of the pairs of labels that can simultaneously be associated with pairs of neighboring picture parts. It can be represented by a binary relation R defined over the universe of labels D. Intuitively, $(d,d') \in R$ if a pair of neighbors can simultaneously be labelled with d and d'. In general, there is a binary relation associated with each neighbor relation.

The most obvious updating mechanism for discrete, symbollic labelling is a label discarding process, which looks at pairs of picture parts at a time. A label, d, can be deleted from the labelling of a picture part if, for some neighboring picture part, that neighbor does not contain a label, d', in its labelling with $(d,d') \in R$.

The binary relation knowledge representation can be generalized to symbollic labellings with confidences for each symbollic label by specifying a real-valued compatibility function, C, whose domain is D×D. As before, in general, a compatibility function is defined for each picture relation in the set r. A variety of applications have used compatibility functions whose range is (-1,1). Intuitively, if $C(d,d') = -1$, then d and d' are maximally incompatible, and the strong presence of d' at one picture part (i.e., d' has a high likelihood at that part) should depress the likelihood of d at a neighboring picture part. If $C(d,d') = 1$, then d and d' are maximally compatible, and the strong presence of d' at a picture part should increase the likelihood of d at a neighboring picture part. Finally, if $C(d,d') = 0$, then the presence of d' at a picture part should have no effect on the likelihood of d at a neighboring part. Intermediate values of C should have intermediate effects.

Several mechanisms have been suggested for applying this knowledge representation to updating labellings. For example, Rosenfeld et al. [6] suggested the formula

$$p_i'(d) = p_i(d)(1 + Q_i(d))/N \qquad (2)$$

where

$$Q_i(d) = \sum_j m_{ij} \sum_{d'} C(d,d')p_j(d') \qquad (3)$$

and N is a normalizing factor which guarantees that $p_i(d) = 1$. The m_{ij} values can be used to give higher weight to some neighbors at part i than others. $Q_i(d)$ measures the overall support of the neighborhood of part i for label d; it takes on values in the range [-1,1] and can be interpreted similarly to C. The above operation is applied in parallel at every part and for every label. The p' values then replace the p values, and the operation can be iterated.

Very often, however, a general updating rule like (2) is inappropriate because it fails to take advantage of knowledge about the specific problem at hand. For example, in the crop classification problem several plausible relaxation rules can be derived from heuristic arguments based on the following general observation:

ORIGINAL PAGE IS
OF POOR QUALITY

Fields are, in general, much bigger than a single pixel and do not change class during a single growing season.

If we let $p_\lambda^k(i,j,t)$ denote the probability that the pixel at spatial location (i,j) and time t is in class λ after k iterations of the relaxation rule, then the following rule schema can be used to derive various specific rules:

$$p_\lambda^{k+1}(i,j,t) = \frac{1}{N} p_\lambda^k(i,j,t) \cdot \prod_{t' \neq t} p_\lambda^k(i,j,t')$$

$$\cdot f(\{p_{\lambda'}^k(i',j',t): \lambda' \in \Lambda, (i',j') \text{ a neighbor of } (i,j)\})$$

Here, N is a normalization factor $(= \sum_{\lambda \in \Lambda} p_\lambda^{k+1}(i,j,t))$ and Λ is the set of possible classes. The second term reflects the fact that the class of a pixel does not change over time, while the third term must represent the spatial dependence of (i,j) being in class λ on the likelihood that its neighbors are in various classes. We can identify at least three plausible functions for f:

$$1)\ f_1 = \max_{i=0}^{7} p_\lambda^k(x_i,t).$$

Here, x_i is an 8-neighbor of (j,k) (see figure below) and this rule can be interpreted as saying: If (j,k) is in class λ, then since fields are large, at least one spatial neighbor of (j,k) must be in class λ.

x_7	x_0	x_1
x_6		x_2
x_5	x_4	x_3

$$2)\ f_2 = \max_{j=0}^{7} \prod_{i=0}^{2} p_\lambda^k(x_{i+j},t), \text{ subscripts mod 8.}$$

If fields are roughly rectangular, then any point will have at least three consecutive neighbors in the same field. Notice that simply choosing

$$f_2' = \prod_{j=0}^{7} p_\lambda^k(x_j,t)$$

would at first iteration lower the probabilities of correct labels at border pixels, and propagate these low probabilites into the center of the field at subsequent iterations.

$$3)\ f_3 = \sum_{j=0}^{6,2} p_\lambda^k(x_j,t)$$

One potential advantage of the sum over the product is its insensitivity to one or two erroneously low probabilities. Using f_3 also gives more equal weight to the temporal and spatial information in the relaxation rule. In the next section we will consider the effect of applying this updating rule to the images in Figure 1.

Next, consider the motion estimation problem. At each pixel we have v^k, the estimate of the velocity after the k^{th} iteration of relaxation. Now, we will assume that, locally, the pattern of image motion vectors can be well described by a rigid image plane motion consisting of a translation and rotation (over large regions this might be a poor assumption since the image motion is the projection of a 3-D rigid motion).

Consider a 3×3 neighborhood of velocities

v_1	v_2	v_3
v_8	v_0	v_4
v_7	v_6	v_5

where v_i is the velocity at p_i.

Since any 2-D rigid motion can be represented as a translation plus a rotation about a fixed point, if our assumption of local motions being rigid image plane motions is correct, then the 3×3 pattern of velocities

v_1-v_0	v_2-v_0	v_3-v_0
v_8-v_0	0	v_4-v_0
v_7-v_0	v_6-v_0	v_5-v_0

should be a rotation about the center point. In general, if point a is rotating about point b with velocity v_a, then

$$v_a \cdot d_{ba} = 0$$

when d_{ba} is the vector from b to a. Therefore, it is straightforward to compute a least squares estimate of the angular velocity, w_0, of p_0, in any 3×3 (or m×m) image neighborhood.

In order to compute a value for v_0^{k+1} we proceed with the following parallel operation: Each pixel is a member of the outer ring of 8 3×3 neighborhoods (one centered at each of its 8 neighbors). Choose the neighbor, i, with minimal least square error and set $v_0^{k+1} = v_i^k + w_i$.

4. ITERATION

An important problem in the application of relaxation processes is determining a termination criteria for the iterations. The two obvious termination criteria are:

1) "convergence" in the sense that the change (in probability vectors, numeric labels, etc.) from one iteration to the next is sufficiently small (see [6] for a discussion of convergence o relaxation processes).

2) exhausting allocated computational resources - e.g., in motion detection, one may only have a short time (< 1 sec) to iterate the relaxation process.

Sometimes one finds that while the first several iterations of a relaxation process tend to improve upon the initial labeling, allowing the iterations to proceed actually degrades the results. In such cases one would want to terminate after only a few iterations.

For the multi-temporal, multi-spectral pixel classification, we find empirically, that the largest increase in classification accuracy occurs in the first iteration, with subsequent iterations having little effect on the results. Figure 5 contains the classification maps for each major class in Figure 1 after 5 iterations of the relaxation algorithm described in Section 2 using function f_3. From these figures, it is clear that the relaxation algorithm has substantially enhanced the detection of the fields and the overall classification strategy.

Next, consider the motion estimation problem. Here, the critical factor in determining the number of iterations of the relaxation process is time, since frames are arriving every .03 seconds. Figure 6 shows the motion vectors after 5 iterations of relaxation. Although the vectors in Figure 6 appear smoother than those of Figure 4, one needs some quantitative measure for comparing them. One such measure is to see how well the velocity vectors predict image structure (say, for coded transmission). If $g(\bar{x},t), t=1,2$ is the intensity at pixel $\bar{x}$ at time t, then let

$$e_k(\bar{x}) = |g(x,1)-g(x+v_{\bar{x}}^k,2)|$$

and, finally,

$$E_k = \sum_{\bar{x}} e_k(\bar{x})$$

If the relaxation is truly improving the velocity vectors, then we would expect E_k to be monotonically nonincreasing with k. Figure 7 shows E_k for k=0,...,5 for our example.

5. CONCLUDING REMARKS

Relaxation processes have potential speed advantages because they can be implemented in parallel (hardware permitting). They have been successfully applied to a wide variety of labelling problems by a growing number of investigators. In spite of these successes, little is as yet known about the design and control of these processes. However, a number of promising approaches to their theoretical formulation are being pursued, and it is hoped that a deeper understanding of their nature will soon be achieved.

The support of the National Science Foundation under Grant.MCS-79-23422 is gratefully acknowledged, as is the help of Janet Salzman in preparing this paper.

REFERENCES

1. Davis, L. S., and A. Rosenfeld (1981). Cooperating processes for low-level vision: A survey. Artificial Intelligence, 17, 245-263.

2. Fischler, M., and R. Elschlager (1973). The representation and matching of pictorial functions. IEEE-Computers, 22, 67-92.

3. Davis, L. S., and A. Rosenfeld (1976). Applications of relaxation labeling: Spring-loaded template matching. In Proc. 3rd Int. Joint Conf. on Pattern Recognition, San Diego, CA, 591-597.

4. Zucker, S. W., and R. A. Hummel (1979). Toward a low-level description of dot clusters: Labeling edge, interior and noise points. Computer Graphics and Image Processing, 9, 213-233.

5. Limb, J., and J. Murphy (1975). Estimating the velocity of moving images in TV signals. Processing, 4, 311-327.

6. Rosenfeld, A., R. A. Hummel, and S. W. Zucker (1976). Scene labeling by relaxation operations. IEEE-SMC, 6, 420-433.

a b c d e f

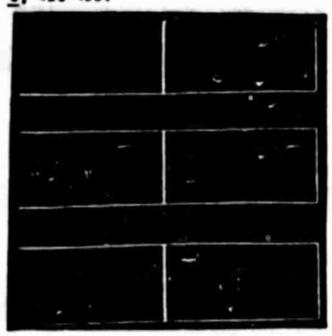


Figure 1. Three acquisitions of LANDSAT for 2 bands

ORIGINAL PAGE IS OF POOR QUALITY

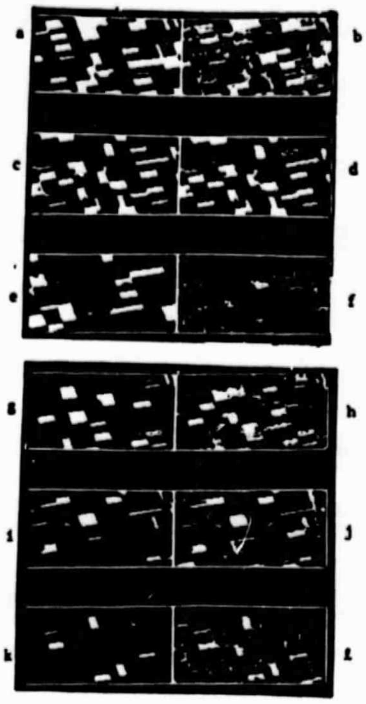


Figure 2. Initial labeling (left column is ground truth, right contains pixels classified into that ground truth class)

Figure 3. Two frames of a traffic sequence

Figure 4. Initial motion vectors

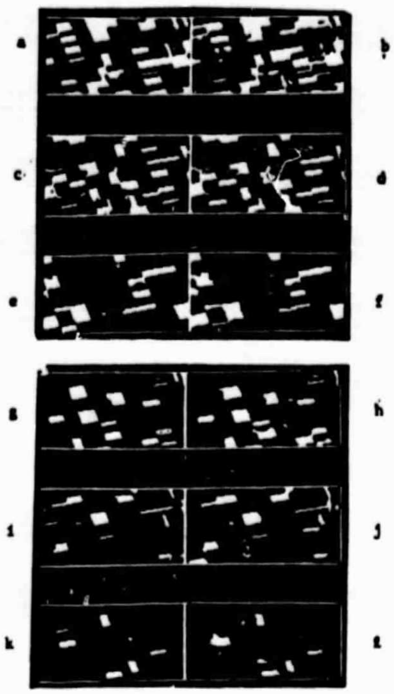


Figure 5. Labeling after 5th iteration using f_3.

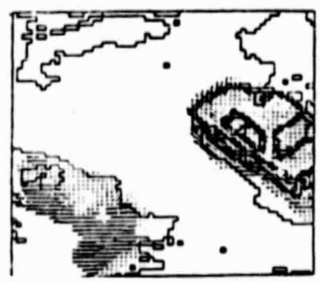


Figure 6. Motion vectors after 5 iterations of relaxation

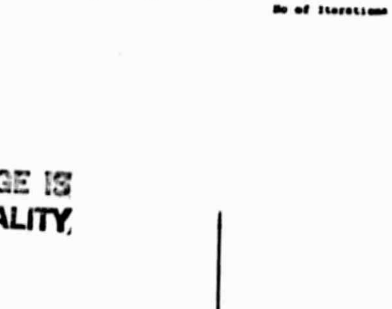


Figure 7. Normalized values of E_k, $k=0,\ldots,5$

ORIGINAL PAGE IS
OF POOR QUALITY

OMIT TO P57

D3

Dual Problems in Image Segmentation

Jack Bryant* and Susan Jenson+

Summary. The obvious duality between edge finding and segmentation was exploited in [1]. This work has been refined, including a more general model for multi-image data, and many more tests of the clustering program AMOEBA. Possible directions for future work in edge following and clustering are suggested, using (now) the duality between segmentation and classification made possible by regarding AMOEBA as a segmentation to classification mapping.

1. J. Bryant, On the clustering of multidimensional pictorial data, Pattern Recognition 11 (1979), 115-125.

---

* Jack Bryant, Department of Mathematics, Texas A&M University, College Station, Texas 77843.

+ Susan Jenson, Applications Branch, EROS Data Center, Sioux Falls, South Dakota 57198.

PRECEDING PAGE BLANK NOT FILMED

PAGE 50 INTENTIONALLY BLANK

D3

SHAPE AND TEXTURE

Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

Regions in a segmented image are characterized, for purposes of description and recognition, by their geometrical properties (size, shape, etc.), as well as by properties that depend on their pixel values (lightness, color/spectral signature, texture). Such properties are also used to define or modify the segmentation process itself, as discussed in the session on segmentation.

The methods used to measure the geometrical properties of a region depend on the data structure used to represent the region. The simplest representation is a binary "overlay" array that has 1's at region pixels and 0's elsewhere. However, other types of representations are often used that are more compact, and that may make it easier to extract certain types of geometrical information. One classical approach is to represent regions by border codes, defining the sequence of moves from neighbor to neighbor that must be made in order to circumnavigate the border; curves can also be represented by such move sequences ("chain codes"). Another standard way of representing regions is as unions of maximal "blocks" contained in them - e.g., maximal "runs" of region points on each row of the image, or maximal upright squares contained in the region; the set of run lengths on each row, or the set of centers and radii of the squares (known as the "medial axis"), completely determines the region. The square centers tend to lie on a set of arcs or curves that constitute the "skeleton" of the region; if we specify each such arc by a chain code, and also specify a radius function along the arc, we have a representation of the region as a union of "generalized ribbons", which are 2D analogs of the "generalized cylinders (or cones)" often used to represent 3D objects.

There has been recent interest in the use of hierarchically structured representations that incorporate both coarse and fine information about a region or feature. A hierarchical maximal-block representation based on recursive subdivision into quadrants, where the blocks can be represented by the nodes of a degree-4 tree (a "quadtree"), is described in (Samet and Rosenfeld, 1980). A hierarchical border or curve representation based on recursive polygonal approximation, with the segments represented by the nodes of a "strip tree", is discussed in (Ballard, 1981); on a border or curve representation based on quadrant subdivision see (Shneier, 1981).

Classically, textural properties have been derived from the autocorrelation or Fourier power spectrum; for example, the coarser the texture (in a given direction), the slower its autocorrelation falls off in that direction from the origin (zero displacement) and the faster its power spectrum falls off in that direction from zero frequency. A related approach, studied extensively by Julesz and Haralick, characterizes textures by their second-order intensity statistics, i.e., by the frequencies with which given pairs of gray levels occur at given relative displacements. It has long been realized, however, that first order statistics of various local property values (e.g., responses of operators sensitive to local features such as edges, lines, line ends, etc.) are at least equally effective in texture discrimination.

More recent work (Beck et al., 1982) suggests that local processes of linking between local features, giving rise to "texture elements" or "primitives", also play a significant role in the perception of texture differences. Texture discrimination based on second-order statistics of local features (e.g., occurrences of edge elements in given relative positions and orientations) has begun to be investigated (e.g., Davis et al., 1979). Texture analysis based on explicit extraction of primitives has also been explored (e.g., Maleson et al., 1977); here statistics derived from properties of the primitives, or of pairs of adjacent primitives, are used as textural properties.

References

1. D. H. Ballard, Strip trees: a hierarchical representation for curves, Comm. ACM 24, 1981, 310-321.

2. J. Beck, K. Prazdny, and A. Rosenfeld, A theory of textural segmentation, in J. Beck and A. Rosenfeld, eds., Human and Machine Vision, Academic Press, NY, 1982, in press.

3. L. S. Davis, S. Johns, and J. K. Aggarwal, Texture analysis using generalized cooccurrence matrices, IEEE Trans. Pattern Analysis Machine Intelligence 1, 1979, 251-259.

4. J. T. Maleson, C. M. Brown, and J. A. Feldman, Understanding natural texture, Proc. DARPA Image Understanding Workshop, October 1977, 19-27.

5. H. Samet and A. Rosenfeld, Quadtree representation of binary images, Proc. 5th Intl. Joint Conf. on Pattern Recognition, 1980, 815-818.

6. M. O. Shneier, Two hierarchical linear feature representations: edge pyramids and edge quadtrees, Computer Graphics Image Processing 17, 1981, 211-224.

For a general survey on shape analysis see:

T. Pavlidis, A review of algorithms for shape analysis, Computer Graphics Image Processing 7, 1978, 243-258.

T. Pavlidis, Algorithms for shape analysis of contours and waveforms, Proc. 4th Intl. Joint Conf. on Pattern Recognition, 1978, 70-85.

For a review of texture see:

R. M. Haralick, Statistical and structural approaches to texture, Proc. IEEE 67, 1979, 786-804.

Annual bibliographies appear that contain many references on all aspects of image analysis; the most recent is

A. Rosenfeld, Picture processing: 1980, Computer Graphics Image Processing 16, 1980, 52-89.

SHAPE AND TEXTURE

Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

Regions in a segmented image are characterized, for purposes of description and recognition, by their geometrical properties (size, shape, etc.), as well as by properties that depend on their pixel values (lightness, color/spectral signature, texture). Such properties are also used to define or modify the segmentation process itself, as discussed in the session on segmentation.

The methods used to measure the geometrical properties of a region depend on the data structure used to represent the region. The simplest representation is a binary "overlay" array that has 1's at region pixels and 0's elsewhere. However, other types of representations are often used that are more compact, and that may make it easier to extract certain types of geometrical information. One classical approach is to represent regions by border codes, defining the sequence of moves from neighbor to neighbor that must be made in order to circumnavigate the border; curves can also be represented by such move sequences ("chain codes"). Another standard way of representing regions is as unions of maximal "blocks" contained in them - e.g., maximal "runs" of region points on each row of the image, or maximal upright squares contained in the region; the set of run lengths on each row, or the set of centers and radii of the squares (known as the "medial axis"), completely determines the region. The square centers tend to lie on a set of arcs or curves that constitute the "skeleton" of the region; if we specify each such arc by a chain code, and also specify a radius function along the arc, we have a representation of the region as a union of "generalized ribbons", which are 2D analogs of the "generalized cylinders (or cones)" often used to represent 3D objects.

There has been recent interest in the use of hierarchically structured representations that incorporate both coarse and fine information about a region or feature. A hierarchical maximal-block representation based on recursive subdivision into quadrants, where the blocks can be represented by the nodes of a degree-4 tree (a "quadtree"), is described in (Samet and Rosenfeld, 1980). A hierarchical border or curve representation based on recursive polygonal approximation, with the segments represented by the nodes of a "strip tree", is discussed in (Ballard, 1981); on a border or curve representation based on quadrant subdivision see (Shneier, 1981).

Classically, textural properties have been derived from the autocorrelation or Fourier power spectrum; for example, the coarser the texture (in a given direction), the slower its autocorrelation falls off in that direction from the origin (zero displacement) and the faster its power spectrum falls off in that direction from zero frequency. A related approach, studied extensively by Julesz and Haralick, characterizes textures by their second-order intensity statistics, i.e., by the frequencies with which given pairs of gray levels occur at given relative displacements. It has long been realized, however, that first order statistics of various local property values (e.g., responses of operators sensitive to local features such as edges, lines, line ends, etc.) are at least equally effective in texture discrimination.

More recent work (Beck et al., 1982) suggests that local processes of linking between local features, giving rise to "texture elements" or "primitives", also play a significant role in the perception of texture differences. Texture discrimination based on second-order statistics of local features (e.g., occurrences of edge elements in given relative positions and orientations) has begun to be investigated (e.g., Davis et al., 1979). Texture analysis based on explicit extraction of primitives has also been explored (e.g., Maleson et al., 1977); here statistics derived from properties of the primitives, or of pairs of adjacent primitives, are used as textural properties.

References

1. D. H. Ballard, Strip trees: a hierarchical representation for curves, Comm. ACM 24, 1981, 310-321.

2. J. Beck, K. Prazdny, and A. Rosenfeld, A theory of textural segmentation, in J. Beck and A. Rosenfeld, eds., Human and Machine Vision, Academic Press, NY, 1982, in press.

3. L. S. Davis, S. Johns, and J. K. Aggarwal, Texture analysis using generalized cooccurrence matrices, IEEE Trans. Pattern Analysis Machine Intelligence 1, 1979, 251-259.

4. J. T. Maleson, C. M. Brown, and J. A. Feldman, Understanding natural texture, Proc. DARPA Image Understanding Workshop, October 1977, 19-27.

5. H. Samet and A. Rosenfeld, Quadtree representation of binary images, Proc. 5th Intl. Joint Conf. on Pattern Recognition, 1980, 815-818.

6. M. O. Shneier, Two hierarchical linear feature representations: edge pyramids and edge quadtrees, Computer Graphics Image Processing 17, 1981, 211-224.

For a general survey on shape analysis see:

T. Pavlidis, A review of algorithms for shape analysis, Computer Graphics Image Processing 7, 1978, 243-258.

T. Pavlidis, Algorithms for shape analysis of contours and waveforms, Proc. 4th Intl. Joint Conf. on Pattern Recognition, 1978, 70-85.

For a review of texture see:

R. M. Haralick, Statistical and structural approaches to texture, Proc. IEEE 67, 1979, 786-804.

Annual bibliographies appear that contain many references on all aspects of image analysis; the most recent is

A. Rosenfeld, Picture processing: 1980, Computer Graphics Image Processing 16, 1980, 52-89.

SHAPE AND TEXTURE

Azriel Rosenfeld

Computer Vision Laboratory
Computer Science Center
University of Maryland
College Park, MD 20742

1. Shape

a) Moments

In this section, we review moments, which are a very useful class of shape properties. The (i,j) moment of f is defined by

$$m_{ij} \equiv \sum_x \sum_y x^i y^j f(x,y)$$

(in the continuous case, $\Sigma\Sigma$ becomes $\iint dxdy$). The first few moments of the picture

```
2 1 1
3 1 0
3 2 1
```

are as follows, if we take the origin at the pixel in the lower left-hand corner of the picture:

i	j	m_{ij}
0	0	14
1	0	8
0	1	12
2	0	12
1	1	7
0	2	20

Moments can be given a physical interpretation by regarding gray level as mass, i.e., regarding f as composed of a set of point masses located at the points (x,y). Thus m_{00} is the total mass of f, and m_{02} and m_{20} are the moments of inertia of f around the x and y axes, respectively. The moment of inertia of f around the origin $m_o \equiv \Sigma\Sigma(x^2+y^2)f(x,y)=m_{20}+m_{02}$. It is easily verified that m_o is invariant under rotation of f about the origin. Moreover, if f is rescaled, say by the factor c, it is not hard to see that m_o is multiplied by c^4. Thus we can normalize f with respect to magnification by rescaling it to give m_o a specified value. Alternatively, a ratio of two moments that have the same value of i+j, e.g. m_{01}/m_{10}, is invariant under magnification.

If we substitute -x for x in the definition of m_{ij}, we obtain $\Sigma\Sigma(-x)^i y^j f(-x,y) = (-1)^i \Sigma\Sigma x^i y^j f(-x,y)$, so that if f is symmetric about the y axis (i.e., $f(-x,y) = f(x,y)$ for all x,y), we have $m_{ij} = (-1)^i m_{ij}$. Thus if i is odd, m_{ij} must be zero. Similarly, if f is symmetric about the x axis and j is odd, $m_{ij}=0$; and if f is symmetric about the origin ($f(-x,-y)= f(x,y)$ for all x,y), and i+j is odd, $m_{ij} = 0$. Moments for which i, j, or i+j is odd can thus be used as measures of asymmetry about the y axis, x axis, and origin, respectively.

If f is a binary-valued picture, say with S as its set of 1's, the moments of f provide useful information about the spatial arrangement of the points of S. To compute moments from the binary array representation χ_S of S, we simply sum the $x^i y^j$ values for all (x,y) in S. To compute them from the run length representation of S, we compute them for each run and sum the results; for example, the (i,j) moment of the run whose endpoints are (x',y) and (x",y) is $y^j \sum_{x=x'}^{x''} x^i$. Similarly, they can be computed from the quadtree representation of S by computing them for each black leaf node, based on its position in the tree, and summing the results. They are not easy to compute from the MAT representation, since the blocks overlap. They can be computed from the crack or chain code representations of the borders of S in much the same way that area is computed from these representations. As an example, for each horizontal crack c_k, let S_k be the vertical rectangle of width 1 extending from the bottom of the picture to c_k; then S is the union of the S_k's for which c_k is an upper boundary of S, minus the union of those for which c_k is a lower boundary. The coordinates of c_k determine the moments of S_k, just as in the case of runs; to compute the (i,j) moment of S, we add the (i,j) moments of all the upper-boundary S_k's, and subtract the sum of the (i,j) moments of all lower-boundary S_k's.

b) <u>The centroid; central moments</u>

The <u>centroid</u> of f is the point $(\bar{x},\bar{y})$ defined by

$$\bar{x} = m_{10}/m_{00}$$
$$\bar{y} = m_{01}/m_{00}$$

Thus the centroid of the 3 by 3 picture shown earlier is (4/7,6/7). It is easily verified that if f is shifted, its centroid shifts by the same amount. (<u>Proof</u>: If we shift f by (α,β), the origin is now at $(-\alpha,-\beta)$, and the new coordinates of (x,y) are $(x+\alpha,y+\beta)$. Hence $\Sigma\Sigma(x+\alpha)f(x,y)/\Sigma\Sigma f(x,y) = m_{10}+\alpha = \bar{x}+\alpha$, and similarly for $\bar{y}$.) Thus if we take the origin at the centroid of f, we have normalized f with respect to translation. Note that since the centroid does not have integer coordinates, if we take it at the origin we should redigitize f; alternatively, we can normalize f by taking the origin at the integer-coordinate point closest to the centroid. (Analogous remarks apply in the case of normalizing with respect to magnification in subsection (a).)

When we take the origin at the centroid, moments computed with respect to this origin are called <u>central moments</u>, and will be denoted by $\bar{m}_{ij}$. Evidently $\bar{m}_{00}=m_{00}$, and it can be verified that $\bar{m}_{10}=\bar{m}_{01}=0$. (Proof: Take $(\alpha,\beta)=(-\bar{x},-\bar{y})$ in the preceding paragraph to obtain $\bar{m}_{10} = \Sigma\Sigma(x-\bar{x})f(x,y) = m_{10}-\bar{x}m_{00} = 0$, and similarly for m_{01}.)

c) The principal axis

The moment of inertia of f about the line $(y-\beta)\cos\theta = (x-\alpha)\sin\theta$, which is the line through (α,β) with slope θ, is

$$\Sigma\Sigma[(x-\alpha)\sin\theta - (y-\beta)\cos\theta]^2 f(x,y)$$

We can find the α, β, and θ for which this is a minimum by differentiating it with respect to α and β and equating the results to zero; this yields

$$\Sigma\Sigma[(x-\alpha)\sin\theta - (y-\beta)\cos\theta]f(x,y) = 0 \quad (\text{from } \partial/\partial\alpha \text{ or } \partial/\partial\beta)$$

$$\Sigma\Sigma[(x-\alpha)\cos\theta + (y-\beta)\sin\theta]f(x,y) = 0 \quad (\text{from } \partial/\partial\theta)$$

Multiplying the first equation by $\sin\theta$, the second by $\cos\theta$, and adding gives $\Sigma\Sigma(x-\alpha)f(x,y) = 0$, so that $\alpha=\Sigma\Sigma xf(x,y)/\Sigma\Sigma f(x,y) =\bar{x}$. Similarly, multiplying the first equation by $\cos\theta$, the second by $\sin\theta$, and subtracting gives $\beta=\bar{y}$. Thus the minimum-inertia line passes through the centroid of f. This line is called the principal axis of f.

To find the slope of the principal axis, take the origin at the centroid; then the moment of inertia of f about the line $y=x\tan\theta$ is

$$\Sigma\Sigma(x\sin\theta-y\cos\theta)^2 f(x,y) = \bar{m}_{20}\sin^2\theta-2\bar{m}_{11}\sin\theta\cos\partial+\bar{m}_{02}\cos^2\theta$$

Differentiating this with respect to θ and equating to zero gives

$$2\bar{m}_{20}\sin\theta\cos\theta-2\bar{m}_{11}(\cos^2\theta-\sin^2\theta)-2\bar{m}_{02}\cos\theta\sin\theta = 0$$

or $\quad \bar{m}_{20}\sin2\theta - 2\bar{m}_{11}\cos2\theta - \bar{m}_{02}\sin2\theta = 0$

so that $\quad \tan2\theta = 2\bar{m}_{11}/(\bar{m}_{20} - \bar{m}_{02})$

Since $\tan 2\theta = 2\tan\theta/(1-\tan^2\theta)$, we can obtain $\tan\theta$ as a root of the quadratic equation

$$\tan^2\theta + \frac{m_{20}-m_{02}}{m_{11}} \tan\theta - 1 = 0$$

It is easily seen that the last equation above is equivalent to

$$(m_{11}\tan\theta+m_{20})^2 - (m_{20}+m_{02})(m_{11}\tan\theta+m_{20}) + (m_{20}m_{02}-m_{11}^2) = 0$$

This implies that $m_{11}\tan\theta+m_{20}$ is an eigenvalue of the matrix

$$\begin{pmatrix} m_{20} & m_{11} \\ m_{11} & m_{02} \end{pmatrix}$$

Show that the principal axis is in the direction of the eigenvector correponding to the larger eigenvalue of this matrix.

A standard method of normalizing f with respect to rotation is to rotate it so that its principal axis has some standard orientation, say vertical. (Here again, this involves redigitization.) More generally, f can be normalized with respect to various types of geometrical distortions by transforming it so as to give standard values to various combinations of its moments.

The principal axis of f can be regarded as a line that "best fits" f. More generally, one can find higher-order curves that "best fit" f in various senses. For example, given a general quadratic curve

$$q(x,y) \equiv ax^2 + bxy + cy^2 + ux + vy + w = 0$$

we can attempt to find the values of the coefficients a, b, c, u, v, w such that

$$\Sigma\Sigma(q(x,y))^2 f(x,y)$$

is a minimum. The curve having these coefficients would be a sort of "quadratic principal axis" for f. Given f's best-fitting quadratic curve q_0, one can attempt to perform "shape normalization" on f by transforming coordinates so that q_0 becomes some standard type of curve--for example, if q_0 is an ellipse, we could transform to make it a circle.

.2. Texture

This section discusses statistical picture properties, and in particular, properties that can be used to describe the "visual texture" of a picture, or better, of a statistically homogeneous region in a picture. We will not attempt to define conditions under which a region would be called uniformly textured. Such regions are often described as consisting of large numbers of small uniform patches, or "primitive elements," arranged according to "placement rules," where the patch shapes and positions are governed by random variables.

a) <u>Gray level statistics</u>

The histogram $p_f(z)$ of a digital picture f tells us how often each gray level occurs in f; it provides an estimate of the gray level probability density in the ensemble of pictures of which f is a sample. If there are k possible gray levels, $z_1,\ldots,z_k$, p_f is a k-element vector. Statistics computed from p_f give us general information about this gray level population. For example,

1) The <u>mean</u> gray level of f, $\mu_f \equiv \frac{1}{N}\Sigma z p_f(z)$, where $N \equiv \Sigma p_f(z)$ is the number of points in f, is a measure of the overall lightness/darkness of f. The <u>median</u> gray level, i.e., the gray level m_f such that (about) half the points of f are lighter than m_f and half are darker, is another such measure.

2) The gray level <u>variance</u> of f, $\sigma_f^2 \equiv \frac{1}{N}\Sigma(z-\mu_f)^2 p_f(z)$, and the <u>standard deviation</u> σ_f, are measures of the overall contrast of f*; if they are small, the gray levels of f are all close to the mean, while if they are large, f has a large range of gray levels. Another such measure is the <u>interquartile range</u> r_f, which is defined as follows: Let m_{1f} be the gray level such that 1/4 of the pixels of f are lighter than m_{1f} and 3/4 are darker; let m_{3f} be defined analogously, with the

---

*Note that if we define one-dimensional moments by $m_i \equiv \Sigma z^i p_f(z)$, we have $N = m_0$ and $\mu_f = m_1/m_0$ (so that μ_f is the centroid of p_f); moreover, if we define central moments $\bar{m}_i \equiv \Sigma(z-\mu_f)^i p_f(z)$ by taking μ_f as the origin, we have $\sigma_f^2 = \bar{m}_2/\bar{m}_0$

1/4 and 3/4 interchanged; then $r_f \equiv |m_{1f}-m_{3f}|$. Other percentiles can be used here in place of the quartiles m_{1f} and m_{3f}.

b) Second-order gray level statistics

Statistics computed from the histogram p_f are of only limited value in describing f, since p_f remains the same no matter how the points of f are permuted--for example, p_f is the same when f is half black and half white, when f is a checkerboard, or when f consists of salt-and-pepper noise. More insight into the nature of f is obtained by studying how often the possible pairs of gray levels occur in given relative positions.

Let $\underline{\delta} \equiv (\Delta x, \Delta y)$ be a displacement, and let M_δ be the k-by-k matrix whose (i,j) element is the number of times that a point having gray level z_i occurs in position $\underline{\delta}$ relative to a point having gray level z_j, $1 \le i,j \le k$. For example, if f is

$$\begin{matrix} 1 & 1 & 2 & 2 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 1 \end{matrix}$$

and $\underline{\delta}$ is (1,0), then M_δ is

$$\begin{matrix} 2 & 1 & 2 \\ 1 & 1 & 1 \\ 0 & 2 & 2 \end{matrix}$$

Note that the size of M_δ depends only on the number of gray levels, not on the size of f. Elements near the main diagonal of M_δ correspond to pairs of gray levels that are nearly equal, while elements far from the diagonal correspond to pairs that are very unequal.

Let N_δ be the number of point pairs in f in relative position $\underline{\delta}$; this is less than the total number of points in f, since if (x,y) is near the border of f, $(x+\Delta x, y+\Delta y)$ may lie outside f. Then in the matrix $P_\delta \equiv M_\delta/N_\delta$ (i.e., if we divide each element of M_δ by N_δ), the (i,j) element is an estimate of the joint probability that a pair of points in relative position $\underline{\delta}$ will have the pair of gray levels (z_i, z_j). P_δ is called a gray level <u>cooccurrence matrix</u> for f.

The matrices P_δ, for various δ's, provide useful information about the spatial distribution of gray levels in f. For example, suppose that f is composed of patches of approximately constant gray level of a certain size s. If the length of $\underline{\delta}$ is small relative to s, then the high-valued entries in P will be concentrated near its main diagonal, since a pair of points $\underline{\delta}$ apart will often have nearly the same gray level. On the other hand, if $\underline{\delta}$ is long relative to s, the entries in P will be more spread out. If f consists of elongated streaks oriented in a given direction, the spread of values in P_δ will depend on both the length and slope of $\underline{\delta}$. If directionality is not important, we can use matrices $\overline{P}$ that are averages of P_δ's (or matrices $\overline{M}$ that are sums of M_δ's) for sets of displacements of a given size in various directions. For example, if f is the 4-by-4 picture shown above, and we use the displacements (1,0), (0,1), (-1,0), and (0,-1), then the combined matrix $\bar{M}$ is

$$\begin{matrix} 8 & 4 & 4 \\ 4 & 6 & 5 \\ 4 & 5 & 8 \end{matrix}$$

(Note that $\bar{M}$ is symmetric, since the set of directions used is symmetric.)

In principle, a large set of P_δ matrices is needed to completely specify the second-order gray level statistics of f. In practice, however, matrices corresponding to large displacements are not necessary. As $\underline{\delta}$ becomes long, the pairs of gray levels separated by $\underline{\delta}$ become uncorrelated, and $P_\delta(i,j)$ approaches the probability that a pair of randomly chosen points of f have gray levels z_i and z_j. Thus for practical purposes we need not use $\underline{\delta}$'s having lengths greater than the distance over which f's gray levels remain correlated, or greater than the size of the "patches" of which f is composed. In fact, the most important P_δ's are usually those for which δ has length 1. Historically, gray level transition probabilities $p(z_j|z_i)$ have been used to characterize textures; $p(z_j|z_i)$ is the probability that a point has level z_j given that the preceding point (with respect to a scan of the picture) has level z_i. Note that the joint probability $p(z_i,z_j)$, which is equal to $p(z_i)p(z_j|z_i)$, is just the (i,j) element of P_δ for $\underline{\delta} = (1,0)$. Other investigators have characterized textures by fitting a time series model to the sequence of gray levels, and using the parameters of this model as texture descriptors; this approach will not be discussed here in detail.

Haralick has suggested a number of statistics that can be used to describe a given cooccurrence matrix P_δ. Four of these are:

1) "Contrast", $\sum_i \sum_j (i-j)^2 P_\delta(i,j)$; this is the moment of inertia of P_δ about its main diagonal. Evidently, it is low when the diagonal concentration of P_δ is high, and vice versa.

2) "Inverse difference moment", $\sum_i \sum_j P_\delta(i,j)/[1+(i-j)^2]$; this is high when the diagonal concentration is high.

3) "Angular second moment", $\sum_i \sum_j P_\delta^2(i,j)$; this is lowest when the $P_\delta(i,j)$'s are all equal, and high when they are very unequal, so that in particular it tends to be high when the diagonal concentration is high.

4) "Entropy", $-\sum_i \sum_j P_\delta(i,j)\log P_\delta(i,j)$; this is highest when the $P_\delta(i,j)$'s are all equal, and hence is low when the diagonal concentration is high.

It should be pointed out that the arrangements of values in the cooccurrence matrices depend not only on the coarseness or busyness of the given picture, but also on its lightness and contrast. For example, if we stretch the grayscale of a picture, the entries in the matrices will spread away from the diagonal, since the pairs of gray levels will be farther apart. Features (1-2) defined above will be especially sensitive to such changes (this is why feature (1) is called "contrast"), while features (3-4) will be less so. To avoid confusing the effects of the first and second order statistics of the picture, it is common practice to normalize its grayscale (e.g., by histogram flattening) before computing the matrices, so that the first order statistics have standard values.

We can define cooccurrence matrices that may be more sensitive to the spatial structure of the given texture by using only selected pairs of points in constructing the matrices, rather than using all possible pairs having a given relative position. For example, suppose that we consider only point pairs (Q,R) in which Q is on an edge (e.g., is at a local maximum of the gradient magnitude), and R is a given distance δ away from Q in the gradient direction. In the matrix P'_δ defined in this way, diagonal concentration still corresponds to coarseness, since if δ is small relative to the texture patch size, R should be interior to the patch on the edge of which Q lies. However, P'_δ may be more sensitive to coarseness changes than the P_δ matrices were, since P'_δ is not influenced by point pairs that are both interior to patches.

c) Local property statistics

Another way of obtaining information about the spatial arrangement of the gray levels in f is to compute statistics of various local property values f' measured at the points of f.

As an illustration of how local properties can be used for texture description, let $\underline{\delta} \equiv (\Delta x, \Delta y)$ be a displacement, let $f_\delta(x,y) \equiv f(x,y)-f(x+\Delta x, y+\Delta y)$, and let p_δ be the histogram of f_δ. Suppose that f is composed of patches of size s. If $\underline{\delta}$ is short relative to s, the high entries in p_δ will be concentrated near 0, since pairs of points $\underline{\delta}$ apart will usually have small differences in value; but if $\underline{\delta}$ is large, the entries in p_δ will be more spread out. (Note, in fact, that $p_\delta(z)$ is the sum of the entries in the matrix M_δ along the line parallel to its main diagonal for which $i-j = z$.) Thus the concentration of p_δ near 0 is a measure of the "coarseness" of f relative to δ, or equivalently, the spread of p_δ away from 0 is a measure of the "busyness" of f. Here again, these properties may depend on direction. Similar remarks apply if we use absolute rather than signed differences; this simply folds p_δ over on itself at the origin.

The gray level (absolute) difference histograms p_δ are not affected by shifting the grayscale (as cooccurrence matrices are), but they are affected by stretching it; thus they too should be used in conjunction with grayscale normalization. Various statistics can be used to describe p_δ, including its mean ($\frac{1}{N}\,\Sigma z p_\delta(z)$, if we use absolute differences), its second moment ($\Sigma z^2 p_\delta(z)$; this is proportional to the "contrast" statistic for the corresponding cooccurrence matrix), its entropy ($-\Sigma p_\delta(z) \log p_\delta(z)$), and so on.

A wide variety of local properties f' can be used in place of f_δ for texture description. For example, we can use combinations of differences, such as the gradient (magnitude) or Laplacian; matches to local templates, such as spot, line, corner, or line end detectors; and so on. f' can be a predicate, e.g. 1 if an above-threshold difference is present and 0 otherwise; in this case, the histogram consists of only two values, and its mean tells us how many edges (or spots, lines, etc.) are present in f per unit area. Another possibility is to count local gray level maxima and minima in f; evidently, both the number of edges and the number of extrema per unit area are measures of "busyness". More generally, we can count occurrences of arbitrary local patterns of values in f.

We can use second order as well as first order local property statistics as texture descriptors, by constructing cooccurrence matrices of the values of f' in given relative positions. If desired, we can use only selected pairs of points in constructing the matrices, e.g., pairs of extrema or pairs of above-threshold edge points, and we can use displacements at each point that depend on f', e.g., displacements in the gradient direction, as at the end of subsection (a).

Rather than using a set of local properties, e.g., differences computed for a set of displacements, we can use a single property and measure it for pictures derived from the original one by a set of local operations. For example, suppose that we use a sequence of local min (or max) operations, and at each step, measure the average gray level; the rate at which this decreases (or increases) is a measure of the coarseness of the high-valued (low-valued) patches in f. For a binary-valued f, the analogous idea is to shrink (or expand) the 1's in f repeatedly, and at each step, count the number of 1's. This approach, using generalized shrinking and expanding operations, has been extensively used for texture analysis in microscopy.

d) Autocorrelation and power spectrum

In the previous two subsections we saw how various statistics of the cooccurrence matrix, difference histogram, etc. for a given displacement $\underline{\delta} \equiv (\Delta x, \Delta y)$ provide useful information about a texture. Thus the set of values of a given statistic α as a function of $\underline{\delta}$ (in particular, for relatively short δ's) can be used as a texture descriptor.

For example, let $f''_{\delta} \equiv f(x,y)f(x+\Delta x,\ y+\Delta y)$, and let α be the mean of f''_{δ}; then α as a function of $\underline{\delta}$ is just the autocorrelation R_f, i.e., the expected value of the product of the gray levels of a pair of points $\underline{\delta}$ apart. By the Cauchy-Schwartz inequality, this takes on its maximum value for $\underline{\delta} = (0,0)$.

(Proof: $\dfrac{\Sigma f(x,y)f(x+\Delta x,\ y+\Delta y)}{[\Sigma f(x,y)^2 \Sigma f(x+\Delta x,\ y+\Delta y)^2]^{1/2}} \le 1$; but the two factors in the denominator are the same, so that the denominator is equal to $\Sigma f(x,y)^2$, which is $R_f(0,0)$.) The rate at which R_f falls off as $\underline{\delta}$ moves away from (0,0) is a measure of the coarseness of f; the falloff is slower for a coarse texture, and faster for a busy one.

Similarly, let $f^2_{\delta} = [f(x,y)-f(x+\Delta x,\ y+\Delta y)]^2$, and let $\alpha \equiv v$ be its mean, i.e., the expected squared gray level difference at two points $\underline{\delta}$ apart; this descriptor is sometimes called the variogram of f. (Compare the use of the mean of the absolute difference histogram as a texture descriptor in subsection (c).) The rate at which its value rises as $\underline{\delta}$ moves away from (0,0) is a measure of the coarseness of f; the rise is slow for a coarse texture and fast for a busy one. Note that $v(\underline{\delta}) = E\{[f(x,y)-f(x+\Delta x, y+\Delta y)]^2\} = E\{f^2(x,y)\} + E\{f^2(x+\Delta x, y+\Delta y)\} - 2E\{f(x,y)f(x+\Delta x, y+\Delta y)\}$; here the first two terms are just $R_f(0,0)$ and the third is $-2R_f(\underline{\delta})$, so that $v(\underline{\delta}) = 2(R_f(0,0)-R_f(\underline{\delta}))$. If f is isotropic, the values of R_f and v depend only on the length of $\underline{\delta}$, not on its direction, so that they become functions of a single variable.

A texture can be modeled as a correlated random field, e.g., as an array of independent identically distributed random variables, to which a filtering operator has been applied. This model suggests that a texture can be described by its autocorrelation and by the probability density of the original random variables; the latter can be approximated by a histogram after a "whitening" operation has been applied to decorrelate the texture. If we use the gradient or Laplacian as an approximate whitening operation, the histogram is just a histogram of difference values, as in the preceding subsection

The Fourier power spectrum $|F|^2$ and the autocorrelation R_f are Fourier transforms of each other. Thus $|F|^2$ can also be used as a texture descriptor. The rate at which $|F|^2$ falls off as the spatial frequency (u,v) moves away from (0,0) is again a measure of the coarseness of f; the falloff is faster for a coarse texture and slower for a busy one, since fine detail gives rise to more power at high spatial frequencies. Samples of $|F|^2$ taken over rings centered at (0,0), or over sectors emanating from (0,0) (to detect directional biases), have often been used as texture features.

Other transforms of f can also be used as a source of texture features. In practice, features based on $|F|^2$ (or R_f) seem to be somewhat less effective for texture discrimination than features based on second-order or local property statistics. At the same time, computation of $|F|^2$ is more costly than computation of a few statistical features (recall that they usually need only be computed for a few $\underline{\delta}$'s), unless we compute it optically.

e) <u>Region-based descriptions</u>

The texture descriptors considered so far are derived from local or point pair properties. We conclude by briefly discussing texture description in terms of homogeneous patches or "primitive" regions. Several types of texture models are based on such decompositions into regions. For example, textures can be generated by using a random geometric process to tessellate the plane into cells, or to drop objects onto the plane, and then selecting gray levels (or gray level probability densities) for the cells or objects in accordance with some probability law.

If we can explicitly extract a reasonable set of primitives from f, we can describe the texture of f using statistics of properties of these primitives--e.g., the mean or standard deviation of their average gray level, area, perimeter, orientation (of principal axis), eccentricity, etc. Second-order statistics can also be used--i.e., we can construct matrices for pairs of values of the area (etc.) at pairs of neighboring primitives (perhaps in directions defined by each primitive's orientation. Of course, this approach depends on being able to extract a good set of primitives from f at a reasonable computational cost. A related, but much simpler, idea is to extract maximal homogeneous blocks (e.g., runs of constant gray level in various directions) from f, and describe f in terms of (first or second order) statistics of the block sizes (e.g., run lengths).

In general, the description of textures in terms of primitives may be hierarchical; the primitives may be composed of subprimitives, etc., or they may be arranged into groupings which in turn form larger groupings, etc. This makes it possible to define placement rules for the primitives in the form of stochastic grammars. Texture analysis can thus be carried out, in principle, by parsing with respect to a set of such grammars.

SHAPE IDENTIFICATION USING 3-D FEATURES

C.M. Bjorklund
R.S. Loe

Lockheed Research Labs
Palo Alto, CA 94304

Image analysis continues to pose significant difficulties for automatic computer analysis due to the unpredictability of object signatures, variability of scene and illumination content and information lost using two-dimensional imagery. Systems providing both intensity and range information permit geometric analysis of the scene to be performed concurrently with grey scale analysis. Laser range imagery provides this capability. Pixel values in a range image measure the distance to the nearest surface along the ray; thus, physical measurements of shapes can be extracted. Two applications will be described. In the first, planar surfaces are identified and extracted for matching in scenes containing buildings [1]. In the second, vehicles (e.g. trucks and tanks) are discriminated based on features extracted from the 3-D data.

1. D.L. Milgram and C.M. Bjorklund, "Range Image Processing: Planar Surface Extraction", 5th ICPR, Dec. 1980, pp. 912-919.

D5 D4

Automatic Photointerpretation via Texture and Morphology Analysis

Julius T. Tou
Center for Information Research
University of Florida

Abstract

This paper discusses computer-based techniques for automatic photointerpretation based upon information derived from texture and morphology analysis of images. By automatic photointerpretation, we mean the determination of semantic descriptions of the content of the images by computer. Such descriptions include a narrative report identifying the objects in the image and describing their characteristics and relationships. Our approaches consist of two major tasks: (1) Morphology analysis, and (2) textural analysis. Morphology and shape information enables us to make a preliminary identification of the objects or regions. In both tasks, a growing knowledge-base is generated from past experience and a priori information. Objects with distinctly different morphology and shape are recognized and these contents in an image are interpreted. To make a finer identification and more accurate interpretation, we make use of textural information.

To perform semantic analysis of morphology, we have developed an heirarchical structure of knowledge representation. The simplest elements in a morphology are "strokes", which are used to form "alphabets". The "alphabets" are the elements for generating "words", which are used to describe the function or property of an object or a region. The "words" are the elements for constructing "sentences", which are used for semantic description of the content of the image. We realize that in many cases morphology alone may not be sufficient to make positive identification and accurate interpretation. Photointerpretation based upon morphology is then augmented by textural information.

To perform textural analysis, we make use of the pixel-vector approach. Each pixel or cluster of pixels is represented by a property vector which characterizes the pixels belonging to an object or a région in an image. Pixels of similar properties are extracted by a correlation and clustering technique. Since an object or a region may contain several types of pixels, an object may be decomposed into several clusters of pixels with different types and properties. The features of the decomposed objects are used in automatic photointerpretation. When objects can be decomposed into similar clusters, we determine the textural rythm as positive identification of the object or the region in addition to morphology information. The knowledge-base is augmented with acquired information from image analysis.

Some experimental results of our knowledge-based photointerpretation system will be discussed.

N83 15771 $_{89}$ ~~D6~~ D6

FLIR TARGET SCREENING

ORIGINAL PAGE IS
OF POOR QUALITY

Raj Aggarwal
Systems & Research Center
Honeywell Incorporated
2600 Ridgway Parkway
Minneapolis, Minnesota 55413

Summary

The remote sensing applications typically involve a sensor that acquires data from the phyiscal world, a processor to process this data and most likely a controller that performs certain mission related functions. Remote sensing is characterized by lack of control over the sensing environment and the scene being sensed. Thus, in addition to the problems posed by the recognition task itself, there are special problems due to uncontrolled environmental factors, including noise and coherent clutter, and in some cases, uncooperativeness on the part of the objects that are to be recognized (countermeasures, camouflage).

One application of remote sensing has been in the area of FLIR target recognition. The sophistication of reconnaissance and strike systems is constantly increasing due to the high threat operational environment. Thus, advanced forward looking infrared sensors are integrated on high performance aircraft. The fast loading and high information rate of advanced sensors has made it imposssible for a human to perform the target search/deduction/recognition task accurately, consistently, and in real time. A lot of work has been done by university/industry teams towards the development of FLIR target screening technology.

A typical target screener would consist of segmentation, detection, and recognition stages. Segmentation step would typically involve the ability to locate the regions of interest. The detection stage is to separate out the clutter from potential targets and the recognition step is to label the type of the target. The process to date includes not only the simulation of this technology in various laboratories around the country but also the development of real time hardware. Some of the hardware boxes have been tested in real time in helicopters, etc.

In FLIR much work has been done on the segmentation and recognition of invidual targets, some of it using structural as well as statistical methods. One key problem that still remains is the ability of these techniques to work under varied conditions without extensive retraining of the algorithms. This deficiency is what leads us to the development of what are known as multi-scenario target screeners. There the key ability for the screener is to learn from what it sees through its sensor and adapt accordingly without operator intervention. The work being done by Honeywell in this direction will be discussed at the conference.

N83 15772 91 D6

Structural Analysis Techniques for Remote Sensing

Linda G. Shapiro
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061

SUMMARY

Structural analysis uses knowledge of the properties of an entity, its parts and their relationships, and the relationships in which it participates at a higher level to locate and recognize objects in a visual scene. For example, Bajcsy and Tavakoli [1] used spectral and shape properties of roads along with knowledge about required connections to other roads in a system for computer recognition of roads from satellite images. Tenenbaum, et. al. [7] set up geometric correspondences between sensed images and symbolic reference maps to aid in monitoring or tracking predefined targets. We will discuss the basic techniques required for structural analysis.

One problem is the representation of structural knowledge. Production systems and relational descriptions are the two major classes of representation used so far. Production systems include picture grammars [2] that use production rules to parse pictures and expert systems [4] where knowledge about pictures can be stored and retrieved. Relational descriptions include trees, graphs, n-ary relations, complex relational structures, semantic nets [4], and frames [3].

A second problem is the development of efficient algorithms for using the structural information to help analyze an image. In [5] we defined a general model of a structural description and compared several inexact matching algorithms for finding the correspondence between models and images. A simple scheme called forward checking showed the most promise. Parallel hardware can also be used to speed up the matching process. In general, the more the problem can be constrained by knowledge, the faster the matching can be done.

A third problem consists of techniques for storage and retrieval of relational models. A knowledge database will typically consist of a large number of models and/or frames. Matching an unknown image against all of them is impossible. Thus schemes for organizing the database of models for fast retrieval of those models most appropriate to a given image are essential. In [6] we discuss some preliminary schemes for database organization. The problem of organizing structural knowledge and knowledge in general is an important topic of current research.

REFERENCES

1. Bajcsy, R. and M. Tavakoli, "Computer Recognition of Roads from Satellite Pictures", Department of Computer and Information Science, University of Pennsylvania, 1974.

2. Fu, K.S., Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.

3. Minsky, M., "A Framework for Representing Knowledge", in The Psychology of Computer Vision, P. Winston, Ed., McGraw-Hill, 1975.

4. Nilsson, N. J., Principles of Artificial Intelligence, Tioga Publishing Co., 1980.

5. Shapiro, L. G. and R. M. Haralick, "Structural Descriptions and Inexact Matching", IEEE Trans. PAMI, Vol. PAMI-3, No. 5, 1981.

6. Shapiro, L. G. and R. M. Haralick, "Organization of Relational Models for Scene Analysis", TR CS81011-R, Department of Computer Science, Virginia Tech, 1981.

7. Tenenbaum, J. M., H. G. Barrow, R. C. Bolles, M. A. Fischler, and H. C. Wolf, "Map-Guided Interpretation of Remotely-Sensed Imagery", Proceedings of IEEE PRIP, 1979, pp. 610-617.

C-2

STRUCTURAL KNOWLEDGE OF AN ENTITY includes

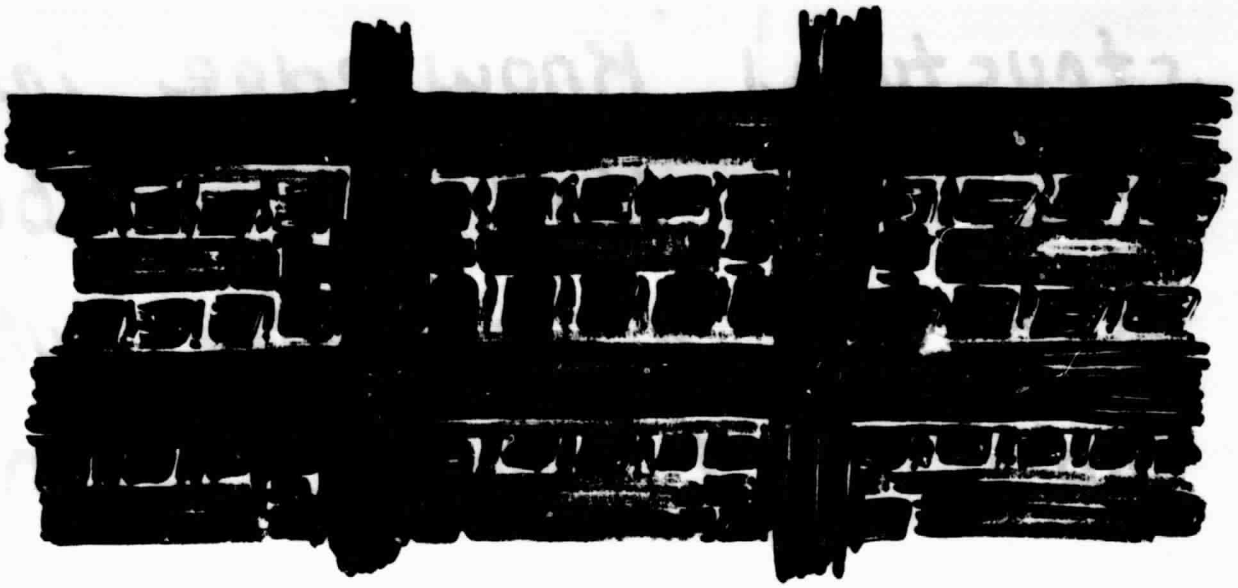
1. Knowledge of its properties

2. Knowledge of its parts and their relationships

3. Knowledge of the relationships in which it participates at a higher level

Example:

ROAD NETWORK IN AN URBAN SETTING



1. One property is its charcoal gray color

2. its parts are thin straight regions arranged in a grid

3. arranged along it are rows of houses and shrubbery

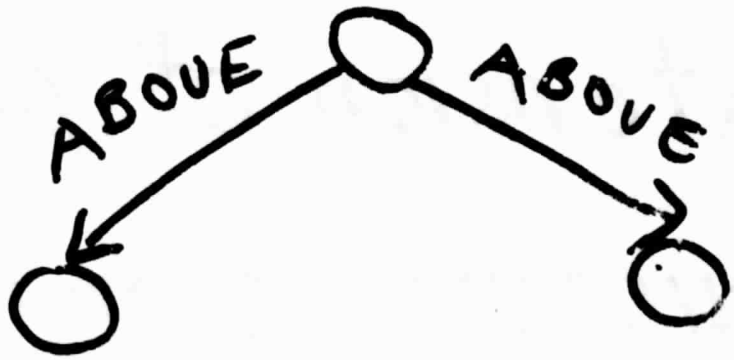
THE REPRESENTATION PROBLEM

How can we represent structural knowledge in a way so that it can be used by a computer whose job it is to analyze scenes?

$\left\{\begin{array}{l}\text{SYNTACTIC}\\ \text{GRAMMATICAL}\end{array}\right.$ DESCRIPTIONS

$$S \rightarrow (\,)$$
$$S \rightarrow (S)$$
$$S \rightarrow S\,S$$

RELATIONAL DESCRIPTIONS



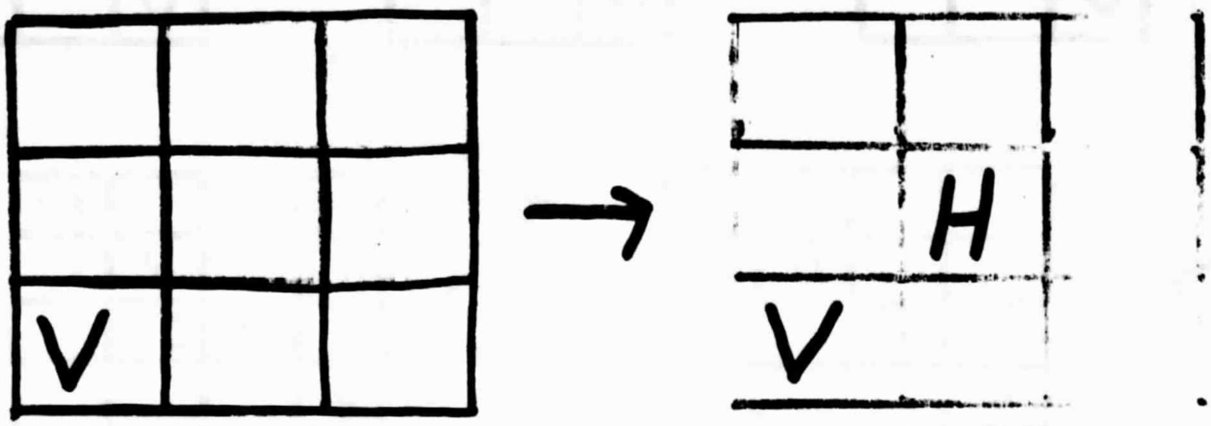
REPRESENTING
KNOWLEDGE
ABOUT PICTURES

PICTURE GRAMMARS

Extension of phrase
structure grammars and
associated parsers
to 2D data

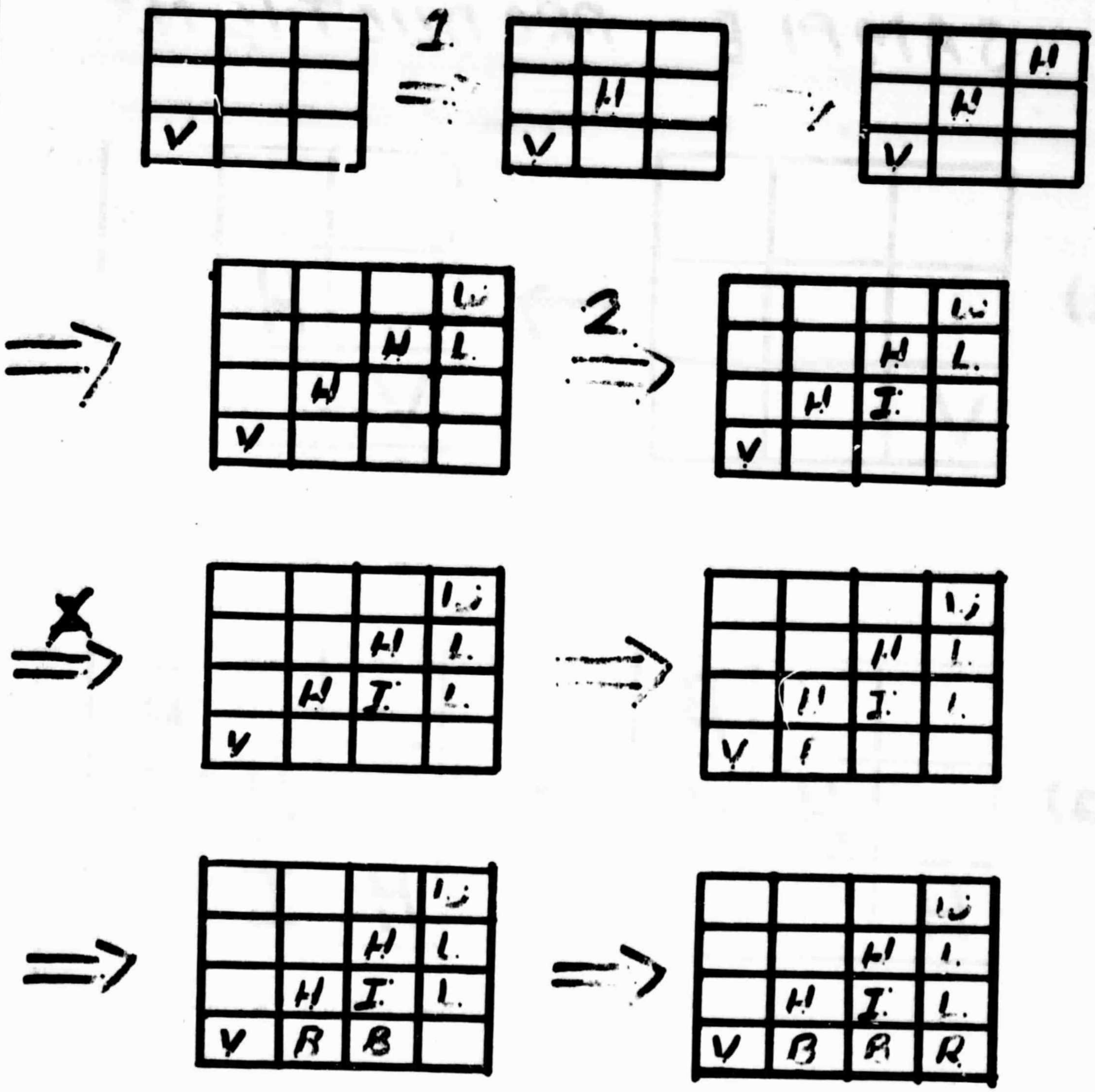
ARRAY GRAMMAR

SAMPLE PRODUCTIONS

1) 

2) 

DERIVATION OF A RIGHT TRIANGLE



ANOTHER TRIANGLE GRAMMAR

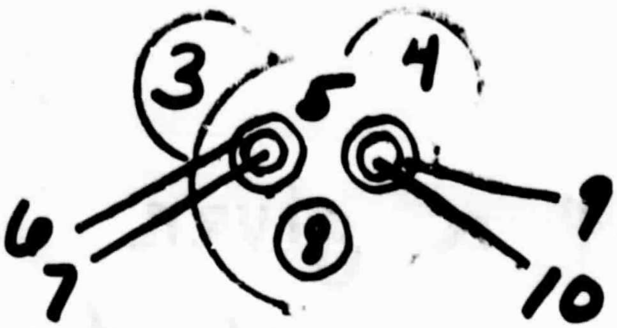
```
(TRIANGLE (X Y Z)
((VERTEX X) (VERTEX Y)
(VERTEX Z) (ELS X Y)
(ELS X Z) (ELS Y Z)
(NON COLL X Y Z)
((VERTICES
  (LIST X Y Z))))
```

HOW DO YOU PARSE A PICTURE?

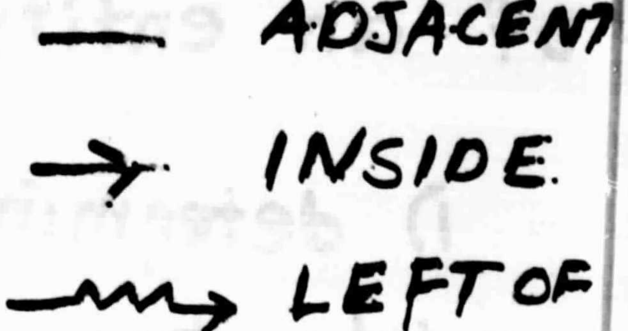
METHOD 1: CREATE A STRING DESCRIBING THE PICTURE AND PARSE IT.

METHOD 2: USE THE GRAMMAR TO GUIDE THE SEARCH FOR CERTAIN RELATIONSHIPS IN THE PICTURE

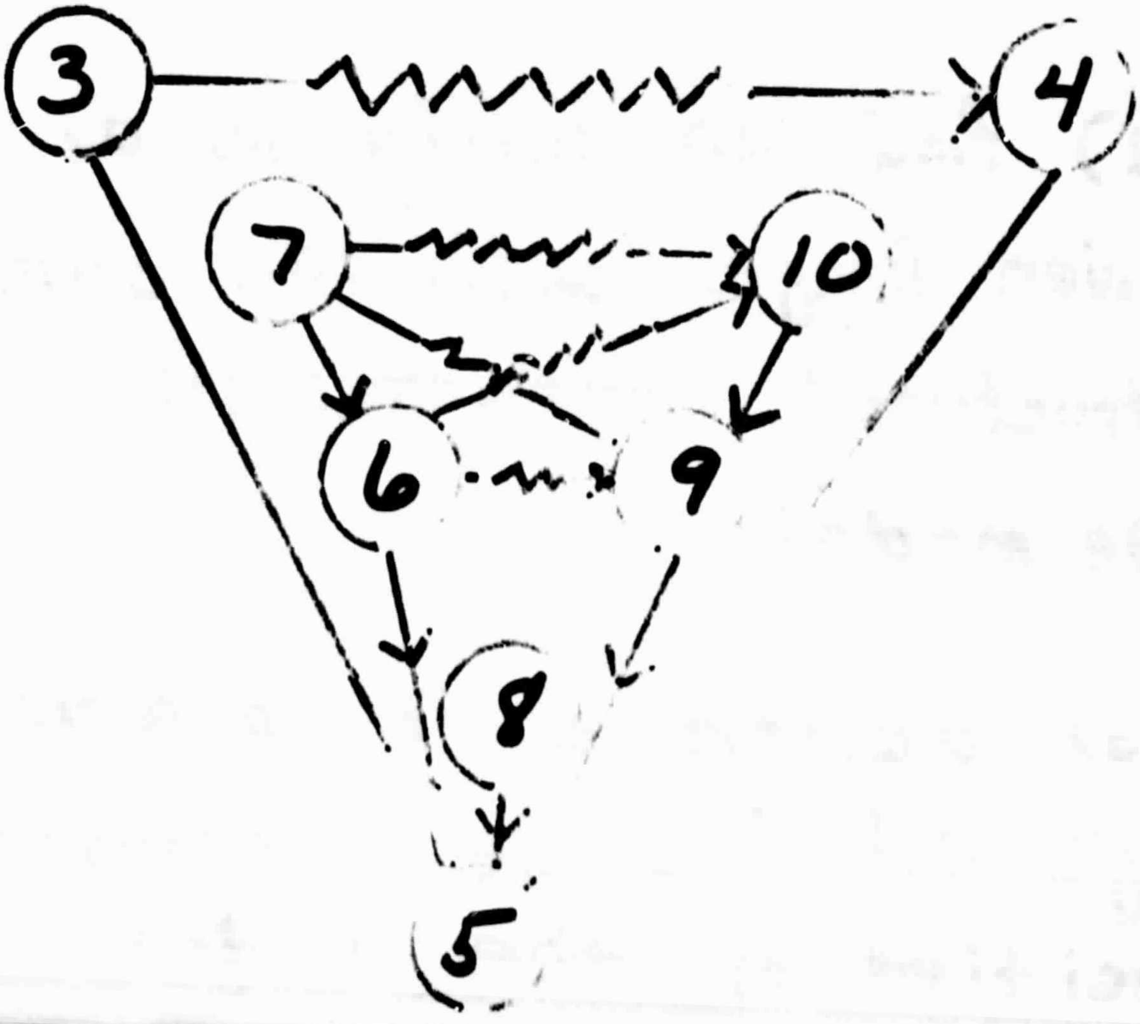
RELATIONAL STRUCTURES



PICTURE



RELATIONS



THE MATCHING PROBLEM

Given a structural model of an entity,

1) determine if a given entity has the same structural characteristics as the model,

or 2) find an entity in a given image with the same structural characteristics as the model,

or 3) determine if an entire image satisfies the constraints specified by the model.

EXACT MATCHING

Structural Description of an Object

$D = (P, R)$.

$P = \{P_1, \ldots, P_m\}$ is a set of PRIMITIVES.

$P_i \subseteq$ Attributes $\times$ Values, $1 \le i \le m$

$R = \{PR_1, \ldots, PR_K\}$ is a set of named N-ary relations over P

$PR_k = (NR_k, R_k)$ $1 \le k \le K$

NR_k is a NAME for R_k

$R_k \subseteq P^{M_k}$ for some M_k is an M_k-ary relation

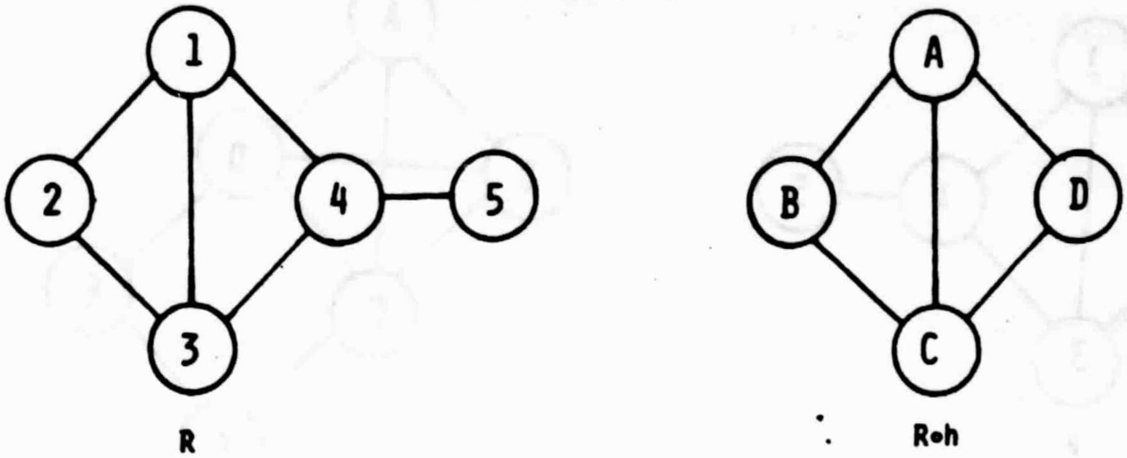
EXACT PRIMITIVE MATCHING

Candidate Primitive C_j

MATCHES

Prototype Primitive P_i

if $P_i \subseteq C_j$



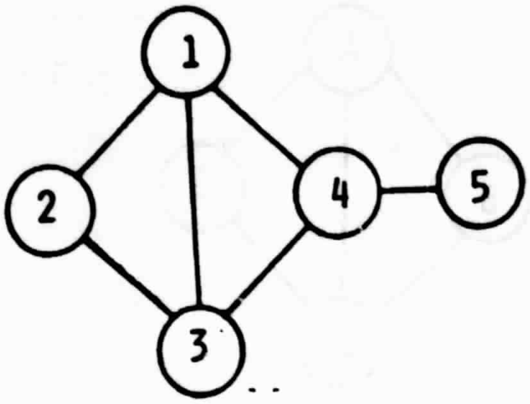
h: 1 → A
2 → B
3 → C
4 → D
5 → D

Figure 1.1 illustrates the composition of binary relation R with mapping h.

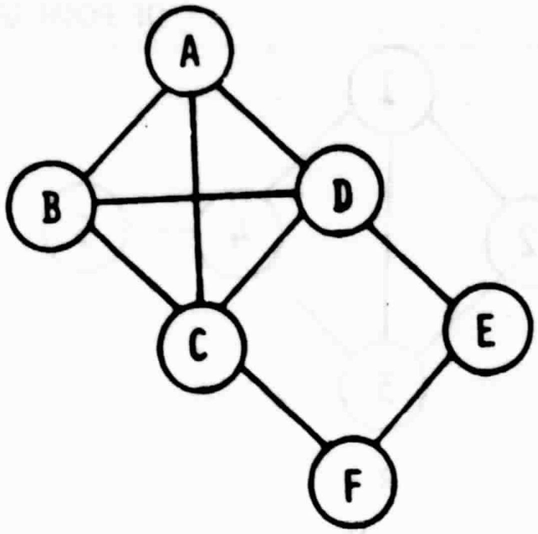
Composition of a Relation with a Function

Let $R \subseteq P^N$ and $h: P \to Q$

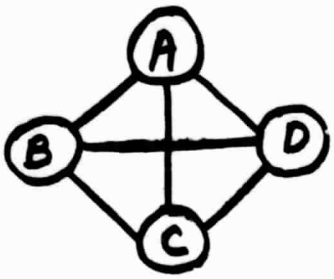
$R \circ h = \{(q_1, \ldots, q_N) \in Q^N \mid$ there exists $(p_1, \ldots, p_N) \in R$ with $h(p_i) = q_i,\ 1 \le i \le N\}$



R



S



$R \circ h$

h: 1 → A
2 → B
3 → C
4 → D
5 → B

$R \circ h \subseteq S$

Figure 4.2 illustrates a relational homomorphism h from binary relation R to binary relation S.

Let $R \subseteq P^N$ and $S \subseteq Q^N$.
A RELATIONAL HOMOMORPHISM from R to S is a mapping $h: P \to Q$ satisfying $R \circ h \subseteq S$

EXACT MATCH DEFINITION

Let $D_P = (P, R)$ be a prototype structural description

$P = \{P_1, \ldots, P_m\}$

$R = \{(NR_1, R_1), \ldots, (NR_K, R_K)\}$

Let $D_C = (Q, S)$ be a candidate structural description

$Q = \{Q_1, \ldots, Q_m\}$

$S = \{(NS_1, S_1), \ldots, (NS_K, S_K)\}$

D_C MATCHES D_P if there is a mapping $h : P \to Q$ satisfying

1) $h(P_i) = Q_j \Rightarrow P_i \subseteq Q_j$

2) $NR_i = NS_j \Rightarrow R_i \circ h \subseteq S_j$

INEXACT MATCHING

Weighted Prototype Structural Description.

$$D = (P, w_p, R, W_R)$$

$$P = \{P_1, \ldots, P_n\}$$

$w_p : P \to [0,1]$ satisfies $\sum_i w_p(i) = 1$

$$R = \{(NR_1, R_1), \ldots, (NR_K, R_K)\}$$

$$W_R = \{w_1, \ldots, w_K\}$$

$w_k : R_k \to [0,1]$ satisfies

$$\sum_{r \in R_k} w_k(r) = 1 \quad \text{for } 1 \le k \le K$$

---

w_p is the primitive weighting function

w_k, $1 \le k \le K$, are the N-tuple weighting functions

ϵ- homomorphisms

Let $R \subseteq P^N$, $w: R \to [0,1]$,
$S \subseteq Q^N$.

An **ϵ-homomorphism** from R to S with respect to w is a mapping $h: P \to Q$ satisfying

$$\sum_{\substack{r \in R \\ r \circ h \notin S}} w(r) \leq \epsilon$$

the ERROR of h

INEXACT MATCH DEFINITION

Let D_p be a weighted prototype structural description

$D_p = (P, W_p, R_p, WR_p)$

$P = \{P_1, \ldots, P_m\}$, $R_p = \{(NR_1, R_1), \ldots, (NR_K, R_K)\}$

$WR_p = \{w_1, \ldots, w_K\}$

Let D_c be a candidate structural description

$D_c = (C, R_c)$

$C = \{C_1, \ldots, C_m\}$, $R_c = \{(NS_1, S_1), \ldots, (NS_K, S_K)$

D_c inexactly matches D_p
with respect to
attribute value thresholds $T = \{t_a \mid a \in A\}$
missing parts threshold t_m, and
relation thresholds $E = \{\epsilon_i \mid PR_i \in R_p\}$
if there is a mapping

$$h : P \rightarrow C \cup \{\text{null}\} \text{ satisfying}$$

1) $h(P_i) = C_j \Rightarrow C_j$ 'inexactly matches' P_i with respect to T.

2) $$\sum_{\substack{P_i \in P \\ h(P_i) = \text{null}}} w_p(P_i) \leq t_m$$

3) $NR_i = NS_j \Rightarrow h$ is an
ϵ_i-homomorphism wrt W_i from R_i to S_j

FINDING E-homomorphisms

Tree search with

a) BACKTRACKING

b) FORWARD CHECKING

c) LOOKAHEAD BY ONE

BACKTRACKING

Instantiate pairs

(P_i, Q_j)

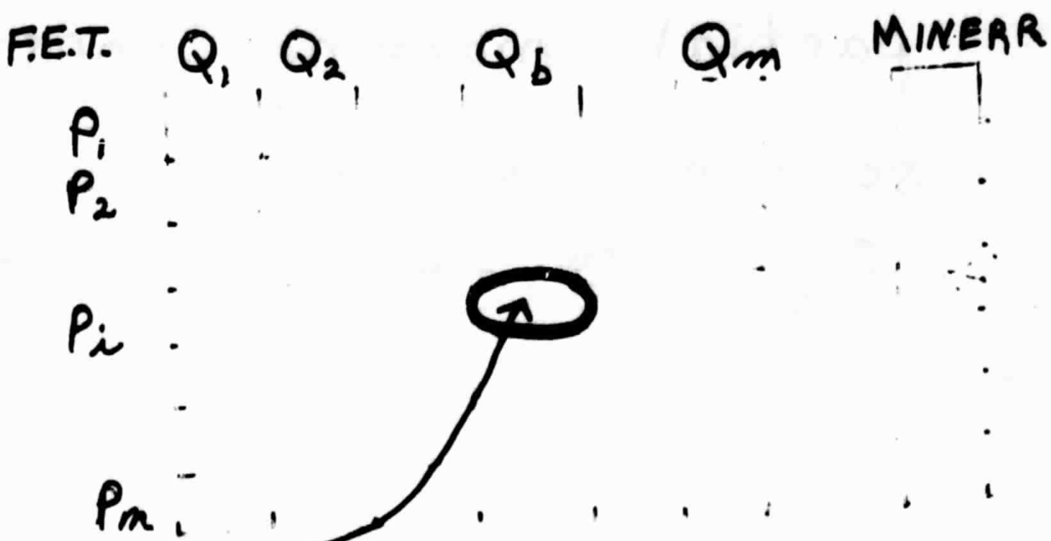
until the error of the partial mapping constructed so far is greater than ϵ.

Then BACK UP.

FORWARD CHECKING

Like backtracking, BUT...

Keep a Future Error Table



F.E.T.(i, j) = the error that mapping P_i to Q_j will cause based on the partial mapping instantiated so far.

MINERR(i) = the smallest error in row i of F.E.T.

PROCEDURE FOR UPDATING F.E.T.

expressed for binary relation $R \subseteq P^2$ and binary relation $S \subseteq Q^2$ (both symmetric)

```
Procedure UPDATE(F.E.T., Pc, Qc, Pasterr)
Sum_of_Errors := 0
for each uninstantiated primitive Pi do
  begin
    for each Qj still O.K. for Pi do
      begin
        if (Pc, Pi) ∈ R and (Qc, Qj) ∉ S
        then F.E.T.(i,j) = F.E.T.(i,j) + w(Pc, Pi
      end
    MINERR(i) := the new minimum in row i
    Sum_of_Errors := Sum_of_Errors + MINERR(i)
    If Sum_of_Errors + Pasterr > ∈
    then fail return
  end
return (Sum_of_Errors)
end
```

IN FORWARD CHECKING

Before instantiating some pair

(P_c, Q_c)

1) check if

error of partial labeling so far
+ F.E.T. (P_c, Q_c)
+ Sum of MINERR(i) for all uninstantiated primitives P_i

is less than or Equal to ϵ

2) call UPDATE to update F.E.T. and return new future error sum (and possibly to FAIL)

LOOKAHEAD BY ONE

Like forward checking BUT also take into account error that can come about due to pairs of yet uninstantiated primitives $(p_i, p_{i'})$.

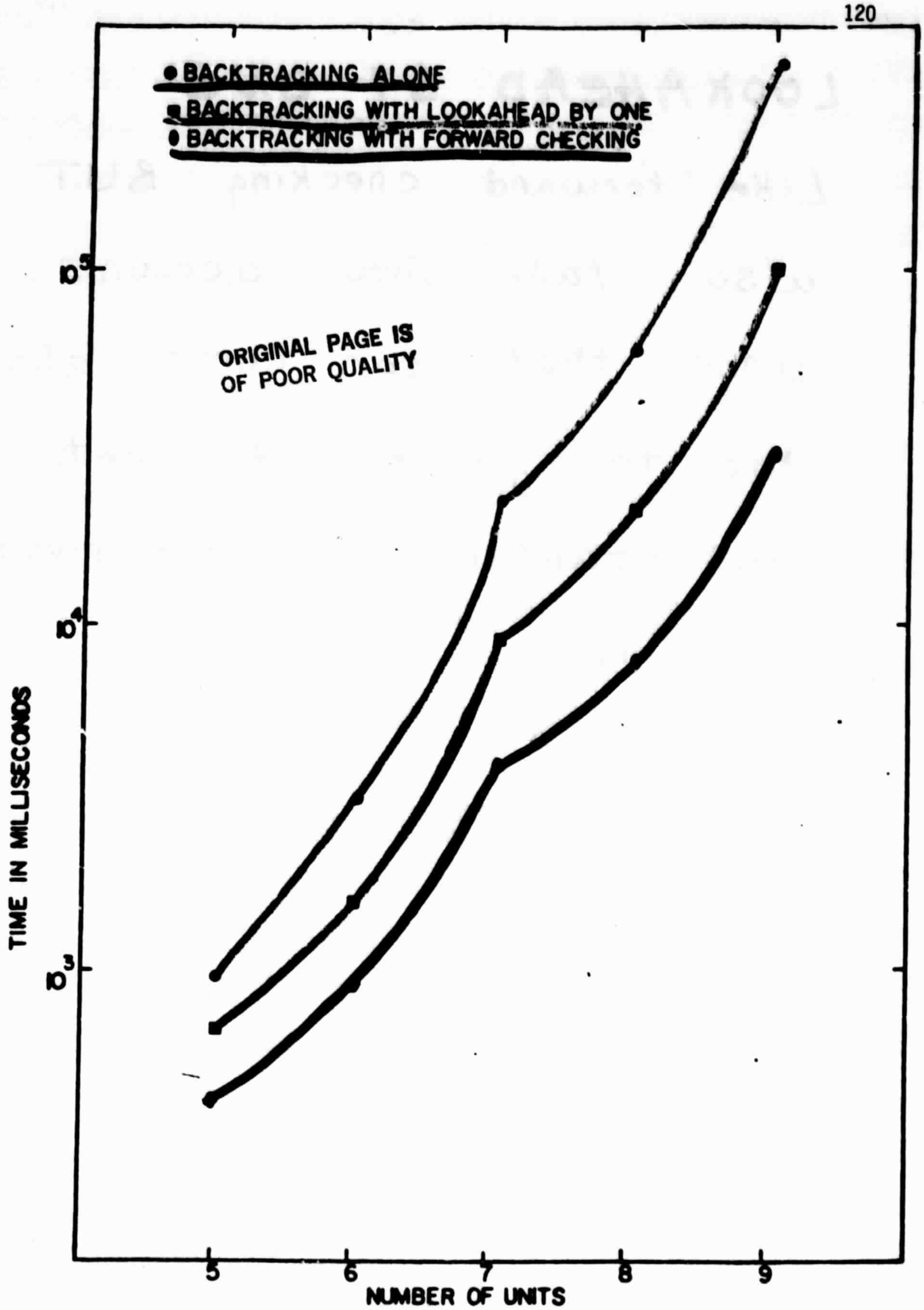


Figure II.(illustrates the number of milliseconds of CPU time on an IBM 370/158 as a function of number of units for $p=.5$, $\epsilon=.1$, and three different search methods.

THE ORGANIZATION PROBLEM

How can we organize a large database of structural models for fast retrieval of those models most appropriate for use in analyzing a given image.

Approach

1) Define a relational distance measure that is a <u>metric</u>

$D(R,S) = 0 \leftrightarrow R$ isomorphic to S

$D(R,S) = D(S,R)$

$D(R,S) \leq D(R,T) + D(T,S)$

for all relational models R, S, T.

2) Using the relational distance meausure, cluster the models into groups of similar models

3) Select for each cluster C a representative R_C.

example representative
Best Model B satisfying

$$\sum_{R \in C} D(R,B) = \min_{R' \in C} \sum_{R \in C} D(R,R')$$

4) Match unknown entities to the cluster representatives and only examine further those clusters whose representatives look promising

For best results, the clustering algorithm and representative selection algorithm should satisfy

For each cluster C and its representative R_C, there is a threshold T_C so that

1) $R \in C \Rightarrow D(R, R_C) \leq T_C$

2) $R \notin C \Rightarrow D(R, R_C) > T_C$

WHAT ABOUT INEXACT MATCHING?

What if unknown entity U is not identical to any model in the database, but is closest to some model M in cluster C with representative R_C and threshold T_C.

$$D(U, R_C) \leq D(U, M) + D(M, R_C)$$

$$D(M, R_C) \leq T_C$$

If $D(U, M) \leq \epsilon$, then

$$D(U, R_C) \leq T_C + \epsilon.$$

In practice, we can choose an ϵ and increase each threshold by ϵ so that any entity within ϵ of a model will match the representative of its cluster

A REMOTE SENSING PROBLEM

Given: LANDSAT image of a mountainous area

Problem: Automatically identify ridges, valleys, rivers, streams, direction of flow.

This problem is being solved at VPI with the use of STRUCTURAL KNOWLEDGE.

Given: an image already segmented into bright and dark regions

APRIORI KNOWLEDGE

1) picture was taken in April -- no foliage

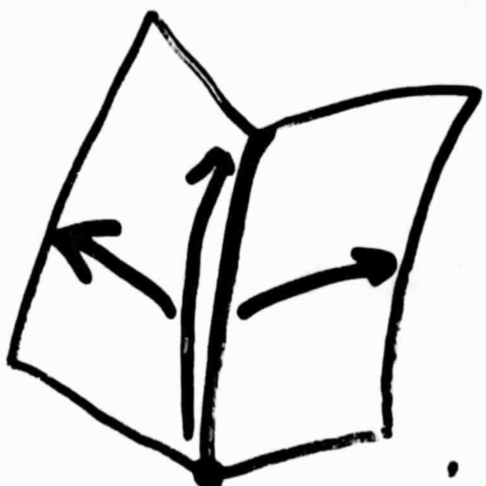
2) sun illumination is from east to west

RIDGE - boundary that is dark on the left and bright on the right

VALLEY - boundary that is bright on the left and dark on the right.

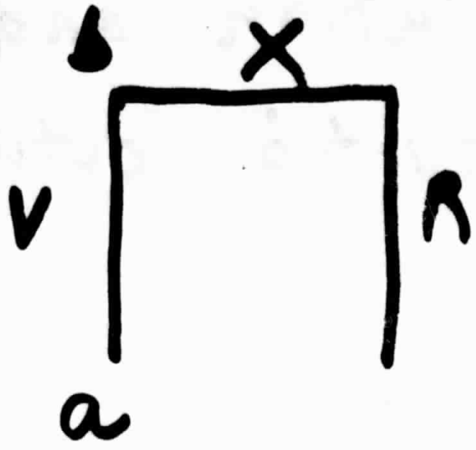
ELEVATION MODEL

Start with known points of lowest elevation and grow the elevation outwards.



We grow along the valley and outward from the valley

CLASSIFYING HORIZONTAL SEGMENTS



a higher than b

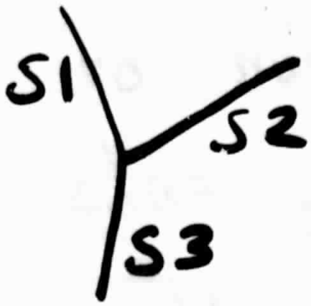
⇒ X = valley

a lower than b

⇒ X = ridge.

PEAK - junction of four boundary segments with more ridges than valleys and with the length of its longest ridge greater than a threshold

STRUCTURAL KNOWLEDGE ABOUT RIVER JUNCTIONS



ANGLES

S1-S3	S1-S2	S2-S3	UP STREAM	DOWN STREAM
=180	<90	>90	S1, S2	S3
=180	>90	<90	S2, S3	S1
<180	>90	>90	S1, S3	S2

⋮

D8

N83 15773

 D7

DETERMINING 3-D MOTION AND STRUCTURE FROM IMAGE SEQUENCES*

T. S. Huang
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1101 West Springfield Avenue
Urbana, Illinois 61801

Summary

The determination of 3-D motion and structure from image sequences has many applications, including interframe TV coding, target tracking, and robot trajectory planning using visual feedback. Past work in the area led to results which involved the iterative solution of nonlinear equations [r-3], and the questions of convergence and uniqueness were not resolved. In this talk, we shall present a new method [4] of determining 3-D motion and structure from two image frames. This method requires eight point correspondences between the two frames, from which 3-D motion and structure parameters are determined by solving a set of eight linear equations and a singular value decomposition of a 3X3 matrix. We also show that the solution thus obtained is unique,

References

1. S. Ullman, The Interpretation of Visual Motion, MIT Press, 1979.

2. J. W. Roach and J. X. Aggarwal, Determining the movement of objects from a sequence of images, IEEE Trans. on PAMI, Vol. 2, pp. 554-562; Nov. 1980.

3. T. S. Huang and R. Y. Tsai, 3-D motion estimation from image-space shifts, Proc. IEEE International Conf. on ASSP, March 30-April 1, 1981; Atlanta, GA.

4. R. Y. Tsai and T. S. Huang, Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces, Report R-921, Oct. 30, 1981, Coordinated Science Laboratory, University of Illinois, Urbana, Illinois 61801.

---

* To be presented at the NASA Workshop on Image Structures, April 28-30, 1982, Texas A&M University, College Station, Texas 77842.

PRECEDING PAGE BLANK NOT FILMED

3-D Motion + Structure from Image Sequences

Roger Y. Tsai and
Thomas S. Huang
Coordinated Science Laboratory
Univ. of Illinois, Urbana-Champaign

Applications: $\begin{cases} \text{Image coding (BW Compression)} \\ \text{Enhancement} \\ \text{segmentation} \end{cases}$

Robotics -
use visual feedback
To plan collision-free
trajectories

Notation and Statement of Problem

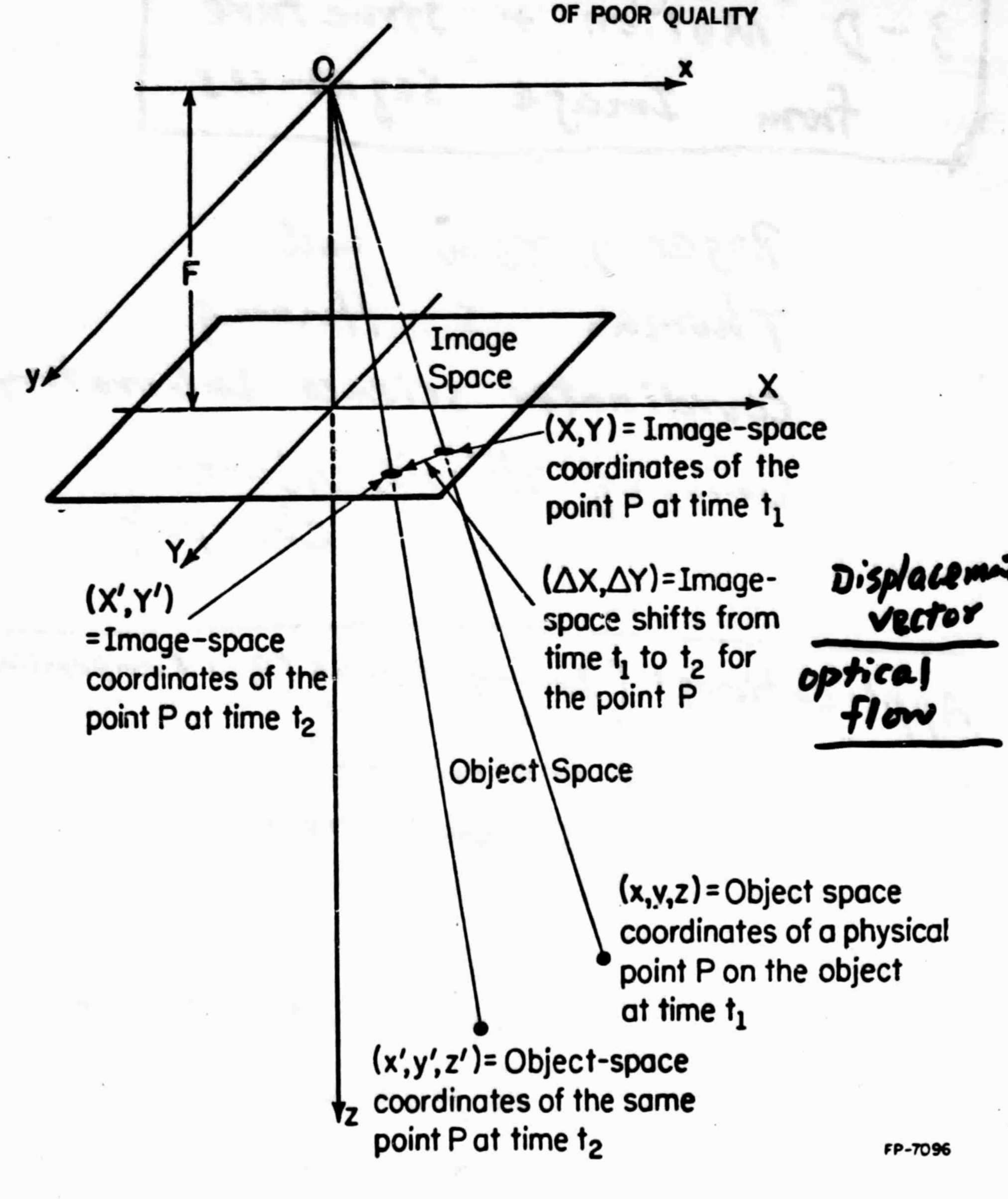


Fig. 1 Basic geometry for three-dimensional motion estimation.

General 3-d motion

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{1.22}$$

where $(\Delta x, \Delta y, \Delta z)$ is the amount of translation, and R is a rotation matrix [1.6]

transpose!

$$R = \begin{bmatrix} n_1^2 + (1 - n_1^2)\cos\theta & n_1 n_2 (1 - \cos\theta) + n_3 \sin\theta & n_1 n_3 (1 - \cos\theta) - n_2 \sin\theta \\ n_1 n_2 (1 - \cos\theta) - n_3 \sin\theta & n_2^2 + (1 - n_2^2)\cos\theta & n_2 n_3 (1 - \cos\theta) + n_1 \sin\theta \\ n_1 n_3 (1 - \cos\theta) + n_2 \sin\theta & n_2 n_3 (1 - \cos\theta) - n_1 \sin\theta & n_3^2 + (1 - n_3^2)\cos\theta \end{bmatrix} \tag{1.23}$$

where n_1, n_2 and n_3 are the directional cosines of the axis of rotation

$$n_1^2 + n_2^2 + n_3^2 = 1 \tag{1.24}$$

and θ is the amount of rotation between the two frames.

Assume the amount of rotation, θ, is small. Then

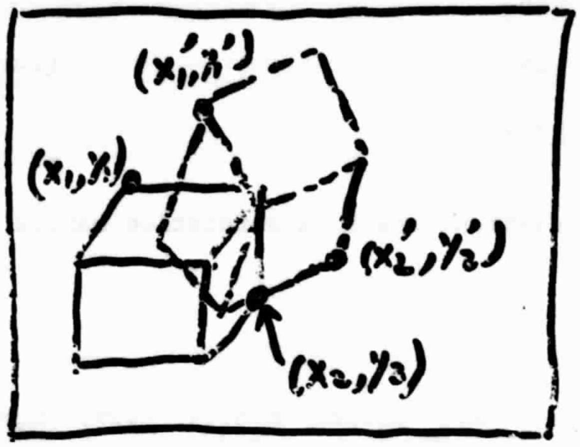
$$R = \begin{bmatrix} 1 & n_3\theta & -n_2\theta \\ -n_3\theta & 1 & n_1\theta \\ n_2\theta & -n_1\theta & 1 \end{bmatrix} \tag{1.25}$$

And we have, using Eqs. (1.22) and (1.15) and after some algebraic manipulation,

Problem: Find R and $\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$ from the two images at t_1 and t_2.

Motion Estimation: Approach

$\boxed{\text{Step 1}}$



$t = t_1$

$t = t_2 (> t_1)$

Find point correspondences

$$(x_1, y_1) \longrightarrow (x_1', y_1')$$

$$(x_2, y_2) \longrightarrow (x_2', y_2')$$

$$\vdots$$

$\boxed{\text{Step 2}}$

$$\begin{cases} x' = x'(x, y, \text{motion parameters}) \\ y' = y'(x, y, \text{motion parameters}) \end{cases}$$

Non linear equations!

solve for the motion parameters.

Examples of derivation of the Nonlinear Equations:

S. Ullman, The Interpretation of Visual Motion, MIT Press, 1979.

J. W. Roach and J. K. Aggarwal, Determining the movement of objects from a sequence of images, IEEE Trans. on PAMI, Nov. 1980.

T. S. Huang and R. Y. Tsai, Three-dimensional motion estimation from image-space shifts, Proc. IEEE Int'l Conf. ASSP, March 30 - April 1, 1981, Atlanta, Georgia.

→ Prazdny.
→ Nagel.

Nonlinear equations - solve by iterative methods

Huang + Tsai

Let $F=1$.

Let $\Delta z = 1$. $\begin{cases} \Delta X = x' - X \\ \Delta Y = y' - Y \end{cases}$

For small rotation,

$$\begin{cases} \Delta X = \dfrac{(Y\varphi_3 - (1+X^2)\varphi_2 + XY\varphi_1)z + \Delta x - X}{(\varphi_2 X - \varphi_1 Y)z + z + 1} \\[2ex] \Delta Y = \dfrac{(-X\varphi_3 + (1+Y^2)\varphi_1 - XY\varphi_2)z + \Delta y - Y}{(\varphi_2 X - \varphi_1 Y)z + z + 1} \end{cases}$$

where $\varphi_1 = n_1\theta$, $\varphi_2 = n_2\theta$, $\varphi_3 = n_3\theta$.

Eliminating z,

$$\frac{\Delta x - X - \Delta X}{\Delta X(\varphi_2 X - \varphi_1 Y + 1) - [Y\varphi_3 - (1+X^2)\varphi_2 + XY\varphi_1]}$$

$$= \frac{\Delta y - Y - \Delta Y}{\Delta Y(\varphi_2 X - \varphi_1 Y + 1) - [-X\varphi_3 + (1+Y^2)\varphi_1 - XY\varphi_2]}$$

unknowns : $\varphi_1, \varphi_2, \varphi_3, \Delta x$, and Δy.

Need at least 5 displacement vectors.

End of Story? No!

uniqueness?

Given the 2 image frames, can we determine the motion parameters uniquely (aside from a scale factor in translation)?

A better motion estimation procedure where we can be sure of our solution?

Solving NL eqs. iteratively has problems – $\begin{cases} \text{nonconvergence.} \\ \text{local min.} \end{cases}$

This paper presents answers to these 2 questions.

Motion estimation for a Rigid planar patch

Assume the object points we consider lie in a plane

$$ax + by + cz = 1$$

at time $t = t_1$.

Then :

$$\begin{cases} X' = \dfrac{a_1 X + a_2 Y + a_3}{a_7 X + a_8 Y + 1} \\[2ex] Y' = \dfrac{a_4 X + a_5 Y + a_6}{a_7 X + a_8 Y + 1} \end{cases}$$

where

$$A \triangleq \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix} = k \left\{ R + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} [a, b, c] \right\}$$

Question 1: Given the 2 image frames, are $\{a_1, a_2, \cdots, a_8\}$ unique? How do we find them?

1(a): How many pts.?

Question 2: Given A, are R, $T \triangleq \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$, and $g \triangleq \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ unique?

How do we find them?

Theorem Given a one-one mapping from R^2 to R^2: $(x, y) \rightarrow (x', y')$

The matrix $A \triangleq \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$

is uniquely determined.

Theorem The matrix A is uniquely determined by a minimum of 4 point correspondences where no 3 points lie on a st. line.

$$A = \kappa \left\{ R + T g^T \right\}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix} = \kappa \left\{ R + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} [a \;\; b \;\; c] \right\}$$

singular value decomposition (SVD) of A:

$$A = U \Lambda V = U \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} V^T$$

Theorem

case i $\quad \lambda_1 > \lambda_2 > \lambda_3$

2 solutions for R, T, g.

case ii $\quad \lambda_1 = \lambda_2 \neq \lambda_3$

1 solution for R, T, g.

case iii $\quad \lambda_1 = \lambda_2 = \lambda_3$

$$\begin{cases} \text{1 solution for } R. \\ T = 0. \\ g \text{ arbitrary.} \end{cases}$$

R, T, g can be expressed in terms of $U, V, + \Lambda$.

(note. scale factor w in $Tg = (Tw)(gw^{-1})$ is excluded.)

Theorem

Given 3 frames, i.e.,

$$R^2(x,y) \rightarrow R^2(x',y') \rightarrow R^2(x'',y''),$$

the motion parameters R, T, and the geometry g are uniquely determined.

Theorem

The motion parameters and the geometrical parameters are uniquely determined from 3 frames of 4 points (no 3 of them lie on a st. line).

Note: the 3 frames have to be distinct.

Estimating the three-dimensional motion parameters of a rigid body with curved surface from 2 image frames

Theorem 1. The motion parameters are uniquely determined given 7 image point correspondences, if the 7 points do not lie on 2 planes one of which contains the origin nor on a cone containing the origin.

Theorem 2. Given 8 point correspondences, the motion parameters can be determined in 2 steps:

(i) solve a set of 8 linear eqs. to get the 8 "essential parameters"

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & 1 \end{bmatrix}$$

(ii) Determine R and T by singular value decomposition of E.

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T$$

where
$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \qquad T = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

$$\begin{cases} X' = \dfrac{(r_1 X + r_2 Y + r_3) z + \Delta x}{(r_7 X + r_8 Y + r_9) z + \Delta z} \\[2ex] Y' = \dfrac{(r_4 X + r_5 Y + r_6) z + \Delta y}{(r_7 X + r_8 Y + r_9) z + \Delta z} \end{cases} \qquad (F=1)$$

$$z = \frac{\Delta x - \Delta z \cdot X'}{X'(r_7 X + r_8 Y + r_9) - (r_1 X + r_2 Y + r_3)}$$

$$z = \frac{\Delta y - \Delta z \cdot Y'}{Y'(r_7 X + r_8 Y + r_9) - (r_4 X + r_5 Y + r_6)}$$

$$[x' \; Y' \; 1] \; E \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = 0 \qquad (1)$$

where

$$E \triangleq \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & 1 \end{bmatrix} = G R$$

$$G = \begin{bmatrix} 0 & \Delta z & -\Delta y \\ -\Delta z & 0 & \Delta x \\ \Delta y & -\Delta x & 0 \end{bmatrix}$$

(i) Determine E.

Given 8 pt. corresp.: $(X_i, Y_i) \to (X_i', Y_i')$

$i = 1, 2, \cdots, 8$

From (1),

$$\begin{bmatrix} x_1'x_1 & x_1'Y_1 & x_1 & Y_1'x_1 & Y_1'Y_1 & Y_1' & x_1 & Y_1 \\ \vdots & & & & & & & \\ x_8'x_8 & x_8'Y_8 & x_8 & Y_8'x_8 & Y_8'Y_8 & Y_8' & x_8 & Y_8 \end{bmatrix} \begin{bmatrix} e_1 \\ \vdots \\ e_8 \end{bmatrix} = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}$$

(ii) R + T from E.

$$E = U \Lambda V^T \qquad \text{(SVD)}$$

Then
$$R = U \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & s \end{bmatrix} V^T$$

or
$$U \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & s \end{bmatrix} V^T$$

where $s = \det(U) \cdot \det(V) = +1$ or -1

and
$$T = \begin{bmatrix} (-\phi_1^T\phi_1 + \phi_2^T\phi_2 + \phi_3^T\phi_3)^{\frac{1}{2}} \\ (\phi_1^T\phi_1 - \phi_2^T\phi_2 + \phi_3^T\phi_3)^{\frac{1}{2}} \\ (\phi_1^T\phi_1 + \phi_2^T\phi_2 - \phi_3^T\phi_3)^{\frac{1}{2}} \end{bmatrix}$$

where ϕ_i^T = ith row of E.

<u>Note</u>. only one of the 2 solutions for R yields positive z.

Technicolor Graphic Services, Inc.

RE: OAB5-39

May 10, 1982

Dr. Larry Guseman
Department of Mathematics
Texas A&M University
College Station, Texas 77843

ORIGINAL PAGE IS
OF POOR QUALITY

Dear Dr. Guseman:

I would like to thank you for inviting me to the NASA Workshop on Image Analysis. Since I work in the area of technology transfer and applications of remote sensing, I viewed the conference from the perspective of how fundamental research influences applied research.

In the area of image segmentation, we now have many tools for clustering and classification. The extension to be made from current capabilities will perhaps make use of ancillary data for classification refinement via digitized soils data, terrain illumination correction, etc. With many layers in a data base, total registration accuracy is a weak point, as well as mechanism to quickly process mixtures of raster and polygonal formats.

Shape and texture analysis seems to me to present the most exciting possibilities - particularly given the finer resolution of Landsat D. Since we know a human analyst can identify features from shape and texture, perhaps images can be enhanced to iteratively improve the interpretability until ultimately some features can be machine-recognized. The role of color in this process has been relatively untapped which to me implies that the extension of algorithms from 2-D to N-D is not at all trivial and deserves a great deal of study.

The area of structural knowledge relates to our needs in that we often model a desired variable (irrigability, exploration potential, grazing capacity) as a function of other variables. A structure within which we can analyze and weight the variables, and arrangements of the variables, would be beneficial.

As to where to look for the evolution of new image processing capabilities, I personally have been dissatisfied with statistical approaches. They seem to deal with images in measurement space only and make assumptions that are not true often enough. The best forward strides I have seen are practical and arise from a well-defined problem.

Again, I enjoyed meeting you, found the workshop stimulating, and am looking forward to visiting A&M again.

Sincerely,

Sue Jenson

Susan K. Jenson
Senior Applications Scientist
Geoscience Section

SOUTH DAKOTA OPERATIONS - EROS DATA CENTER • SIOUX FALLS, SOUTH DAKOTA 57198 • TELEPHONE (605) 594-6511

COMPUTER VISION LABORATORY

Image Analysis
Picture Processing

301-454-4526

May 13, 1982

Dr. Larry Guseman
Dept. of Mathematics
Texas A&M University
College Station, TX 77840

Dear Larry,

The following are some suggestions for future research areas under the NASA Fundamental Research Program in Pattern Recognition:

1. Application of AI methodology to develop "expert systems" for various interpretation tasks.

2. Automatic registration of multisensor data, and of images with maps.

3. Study of VLSI architecture requirements for remote sensor data processing and analysis.

4. Development of database management techniques applicable to remote sensor data.

We hope these ideas will be useful. We enjoyed the meeting and look forward to seeing you again soon.

Sincerely,

Larry

Larry S. Davis
Associate Professor

Azriel Rosenfeld
Research Professor

AR:job

Computer Science Center, University of Maryland, College Park, Maryland 20742, U.S.A.

University of Illinois at Urbana-Champaign

College of Engineering
COORDINATED SCIENCE LABORATORY

1101 West Springfield Avenue
Urbana, Illinois 61801-3082
(217) 333-2511

217-333-6912

May 21, 1982

Dear Prof. Guseman:

Sorry for the delay, but here are copies of some of my viewgraphs. I hope you still have time & include them in the workshop proceedings.

As I stated during the workshop, I think we need basic research in the area of relating 2-d images to 3-d scenes.

Sincerely,

Tom Huang