RECEIVED BY DTIC MAR 17 1970

Coo-1018-1198

MASTER

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS 61801

STUDIES IN IMAGE PROCESSING:

A Systems Viewpoint

by

Jacob L. Woolfson

Charles H. Heider

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:
A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.
As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

File No. 825
February 12, 1970

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

leg

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ABSTRACT

The first part (written by Charles H. Heider) of this analysis, "Studies in Image Processing: A Systems Viewpoint", discusses the literature search, develops the outline of the image processing modules, and presented specific examples of modules used in the first four major categories of pattern recognition processing: Optical Filtering, Image Encoding and Digitization, Image Enhancement and Pattern Articulation. The second part of the Paper, by Jacob L. Woolfson, then presents examples for the final two major categories of pattern recognition processing: Diagraph Representation and Cognition.

It is recognized that the treatment of the image processing modules presented here is still very preliminary and in some instances totally missing. The material is being issued at this time to provide an annotated bibliography for the Computer Science 491 Seminar: Image Processing (Spring 1970).

1.0 INTRODUCTION

This paper reports the results of a systems analysis of image processing research which has been conducted during the past ten years. The purpose of this study was to develop the concept of a modular image processing system incorporating the more successful methods and techniques of many different image processing projects. The system was initially conceived to be a collection of modules or procedures which independently solve a portion of the pattern recognition problem. Linking together of the various modules with their associated parameters and outputs could lead to a specific approach for any particular pattern recognition problem.

To fully specify the system modules it is necessary to analyze image processing into procedures which could be classified as stand alone elements. Following this breakdown it is then necessary to establish the parameters of each system element (global and local), the input/output formats for all data transfers and the description of available algorithms required to accomplish each process. Subsequent studies could then be made to determine operating parameters such as machine time to execute each procedure, the storage capacity required and data transfer requirements.

A program for accomplishing the objectives of this study began with a detailed review of the literature, with an eye to cataloging and classifying the image processing approaches and techniques reported in the literature. After an initial classification of methods was complete, preliminary module descriptions and specifications were then developed. The program needs yet to be concluded with a comparative analysis of the system modules with the relevant literature to test for conformance and verify that the application could be accomplished with the newly specified modules.

This paper is organized into six sections with Section 2 concerned with a description of the literature search and the classification process. Section 3 describes the modules and their parameters. Section 4 contains the bibliography which resulted from the literature research.

2.0 LITERATURE SURVEY: Image Processing Methods and Procedures

Image processing refers to the analysis and/or generation of pictures or images by a digital computer with or without human intervention. Frequently termed pictorial pattern recognition or picture processing, image processing is actually a relatively young field of study, a fact which may be easily verified by a review of the relevant literature where papers dated prior to 1960 are infrequent.

There are basically two approaches to image processing: one method concentrates on measuring certain properties of the image and then applies property space analysis to discriminate or classify the image, the second method utilizes linguistic methods incorporating the notions of syntax and semantics and then applies artificial language processing to the recognition process. There is more literature devoted to research in the former field than in the latter. However recent research utilizing linguistic methods--such as the Illiac III project--have begun to demonstrate the advantages of the second approach when applied to complex scene analysis. Although this study considered the property space methods, the major effort has been directed at the linguistic approach to image processing.

The categorization of the image processing sequences was performed utilizing the following major categories:

1. Optical Filtering
2. Image Encoding and Digitization
3. Image Enhancement
4. Pattern Articulation
5. Digraph Representation
6. Cognition

Optical Filtering - refers to the application of spectral or holographic filtering to the original image or scene before it is converted to electrical signals by the scanning mechanism.

Image Encoding and Digitization - includes the raster specification, sampling strategy, gray scale quantification and transmission codes for the conversion of the image or scene to machine compatible form.

Bandwidth compression techniques would also be considered in this category if required by the transmission of the digitized image.

Image Enhancement - a preprocessing step, is concerned with enhancing the digitized image before the recognition process is begun. Included in this category are spatial filtering, affine, geometric and gamma transformations, skeletonization, line and edge enhancement and texture discrimination. These processes are all local in nature and could therefore be accomplished in parallel operations with a considerable decrease in execution time as compared to serial operations.

Pattern Articulation is the first feature extraction process and is also local in nature. Processes considered here include object extraction, line and arc segment extraction, node extraction, labelling, connectivity analysis, skeleton extraction, texture identification and parameter extraction.

Digraph Representation is a higher level process which uses both local and global information and results in the generation of graph or tree decompositions of the scene or image. The ingredients of this category are: containment relations, neighborhood relations, texture analyses, shape analyses and property space analysis.

Cognition represents the final recognition process and includes syntax, semantics and discriminant function analysis. The first two processes require the adaptation of artificial language techniques while the third process uses statistical methods such as clustering and property space analysis to perform the recognition process.

OUTLINE OF IMAGE PROCESSING MODULES

- I. OPTICAL FILTERING
 - A. Spectral
 - B. Holographic
- II. IMAGE ENCODING & DIGITIZATION
 - A. Raster Specification
 - B. Sampling Strategy
 - C. Gray Scale Quantification
 - D. Transmission Code
- III. IMAGE ENHANCEMENT (PRE-PROCESSING)
 - A. Spatial Filtering (Template Matching, Object Enhancement, etc.)
 - B. Affine Transformations
 - 1. Rotation
 - 2. Magnification
 - 3. Translation
 - C. Geometric Distortion Correction
 - D. Gamma Transformations
 - E. Skeletonization
 - F. Edge Enhancement
 - 1. Spatial Differentiation
 - 2. Thresholding
 - G. Line Enhancement
 - 1. Line thinning
 - 2. Gap Filling
 - H. Texture Discrimination
- IV. PATTERN ARTICULATION (FEATURE EXTRACTION)
 - A. Primitive Object Extraction
 - B. Line and Arc Segment Extraction
 - C. Nodal Extraction
 - D. Labelling
 - E. Connectivity Analysis/Path Extraction
 - F. Skeletonal Extraction
 - G. Texture Identification

V. DIGRAPH REPRESENTATION

- A. Containment Relations
- B. Neighborhood Relations
- C. Texture Analysis
- D. Shape Analysis

VI. COGNITION

- A. Syntax
- B. Semantics
- C. Discriminant Function Analysis
- D. Context Analysis

GAMMA TRANSFORMATIONS

The slope of the line defining the logarithmic ratio of the brightness of corresponding points on an original image and on a reproduced image has been given the name "gamma" in photography. A gamma of 1 signifies that the relative contrasts of the replica and object images are equal. If the slope (gamma) is greater than 1 the contrast of the replica image will be greater than the object image and the replica image will appear harsh or crisp. If the gamma is less than 1 the contrast of the replica image is less than the object image and it will appear soft or flat.

This concept is quite applicable to pattern recognition as it provides a means of varying the contrast of an image after it has been encoded. A gamma transformation procedure would require local processing capabilities which are available in a parallel machine. Types of transformations which could be implemented are upward or downward linear shifts, non-linear shifts, or divided shifts in which part is shifted upward and part downward. The placement of a logarithmic gray scale wedge on a margin of the object image would provide a reference standard to the gamma transformation procedure. Local areas could be selectively transformed by specifying a window or masking plane in the procedure call.

Parameters: The minimum parameter requirements for a gamma transformation procedure would require an input plane index, output plane index and the (g) or gamma value. (g) would typically range from $.1 \leq g \leq 2$. Additional parameters might include the selection of an upward, downward or divided shift mode and specify the modification function. A third plane specifying a mask area could also be included.

Parameter Call:

Call Gamma (IPI, IPO, g, S, F, IW)

where:

IPI	-Index of plane in
IPO	-Index of plane out
g	-gamma value $.1 \leq g \leq 2$
S	$\left\{ \begin{array}{l} \text{+up shift} \\ \text{-down shift} \\ \text{0 divided shift} \end{array} \right.$
F	-shifting function or rule
IW	-index of mask or window

NOTE: S, F, IW may be left blank.

References:

William K. Pratt, "Hadamard Transform Image Coding," Proceeding of IEEE, Vol. 57, No. 1, January 1969.

Alan Oppenheim, et al, "Nonlinear Filtering of Multiplied and Convalved Signals, Proceeding of IEEE, Vol. 56, No. 8, August 1968.

SKELETONIZATION

Skeletonization is the process of transforming a closed pattern, for which the edge has been clearly marked, into a locus, or skeleton, which represents and describes the exact shape of the pattern. This skeleton is called the medial axis of the pattern. The transform is generated by causing the pattern (outside border) to shrink down in size till it forms the skeleton of this figure. The formation of this axis corresponds to a wavefront propagating inward from each point on the curve with uniform isotropic velocity normal to the wavefront. The locus of points at the intersection of propagating wavefronts is the medial axis transform locus. The distances from the points of intersection to the pattern boundary, which is equivalent to the times of intersection, are necessary for description of the pattern. Each point on the skeleton is described by three coordinates, two specifying its position, and the third its time of appearance.

In the AFCRL system, the transformation is carried out on a rectangular grid of 256×256 elements. Presumably, if a whole pattern lies partly outside this grid, the size of the pattern could be transformed into a grid of the above size. Likewise too small a pattern could be enlarged within the above grid. Time units are quantized so that the pattern propagates a distance of two units at each step. The new pattern is examined for corners after each time unit, and points detected are listed as belonging to the skeleton.

An octagonal approximation to a circle is used for propagation. A circle of radius 2 units is constructed with center at the point under examination. The points on the circle are examined, and the central point is labelled as a point of the skeleton if either

- 1) More than eight of the sixteen points on the circle are inside the pattern. This condition detects points which are on corners of angle less than 150° . Angles greater than 150° contribute very little shape information.

2) As a point moves around the circle, it enters and leaves the pattern more than once. This condition obtains a continuous line for the skeleton in cases in which the angle is small.

Parameters:

Input: Exact points for edge of figure, normalized to 256 x 256 grid.

Output: Each skeleton point has 3 coordinates: x, y, and time of appearance.

Reference: Otis Philbrick, "Shape Description with Medial Axis Transformation," Pictorial Pattern Recognition, Change, Ledley, et al. Thompson Book Co. 1968.

EDGE ENHANCEMENT

Edge enhancement is directed at the problem of boundary definition for objects in an image of multiple gray levels. The result of applying the edge enhancement algorithm to a digitized image should be a line-like drawing or cartoon of the initial scene. The basis of the procedure is the premise that meaningful information usually exists in regions of drastically changing gray levels.

This information is local in nature and is readily adaptable to parallel processing algorithms. This procedure assumes an algorithm, most likely of the spatial differentiation type, which operates on each element of the picture matrix and its 8 neighboring cells, simultaneously. The rate of change of optical density is calculated in each of the 8 neighbor cell directions and the largest value (steepest descent) and its direction are stored for each picture matrix cell. A second algorithm using statistical weighting functions selects the threshold values and extracts line segments representing boundaries and contours. A line extension algorithm is also included to provide line extensions based on weighting information developed from the gradient values.

Parameters: There are three global parameters used in the Edge Enhancement procedure that are defined in previous blocks in the pattern recognition model. These are the input plane, output plane and number of gray scale values. In addition several local parameters are developed internally by the procedure on the basis of statistics generated during the procedure operations. Since these local parameters are automatically specified the procedure call should only specify global parameters.

Procedure Call:

Call EDGE EN (IPI, IPO, N)

where:

IPI	-Index of Input plane
IPO	-Index of Output plane
N	-Number of gray scale values

References:

T. Sakai, et al., "Line Extraction and Pattern Detection in a Photograph," Pattern Recognition, Vol. 1, pp. 233-248.

M. L. Mendelsohn, "Digital Transformation and Computer Analysis of Microscopic Images," Advances in Electron and Optical Microscopy, 1968.

LINE ENHANCEMENT

Line enhancement is concerned with the problem of noise filtering or the thinning of line segments in 2-tone line drawings and in character recognition applications. This is a pre-processing step which may be applied after the edge enhancement procedure has reduced the gray scale image to a 2-tone line drawing and before the feature extraction algorithms are applied. The end result of applying this procedure is an idealized line drawing to which the template matching or feature extraction procedures can be applied. Together with the edge enhancement procedure, line enhancement could be joined to create an outline transformation procedure which would produce the cartoon of a gray scale image. However in character recognition for example, edge detection may not be required so there is some advantage in having two separate procedures.

Line thinning is a local operation which can be performed by a series of local boolean operations. Wide lines are gradually reduced in an iterative process to lines of a single cell width. After the thinning process, a line regeneration or gap filling algorithm is applied to reconnect lines which may have been broken in the thinning procedure or were broken due to noise, etc.

The regeneration or reconnection of lines in the line enhancement procedure should not be confused with the line extension algorithm in the edge enhancement procedure which make use of information local to that procedure such as gradient values and directions.

Parameters: The parameters required by the Line Enhancement procedure are the index of an input plane and the index of an output plane, and are similar to those used by the Edge Enhancement procedure.

Parameter Call:

Call Line EN (IPI, IPO)

where:

IPI	-Index of Input plane
IPO	-Index of Output plane

References:

C. Shinn, R. Kline, "A Computer-Directed System for Measuring Distance Between Edges in Optical Images," Tech Report #10, Computer Systems Laboratory, Washington University, St. Louis, Missouri.

S. Yamada, J. P. Fornango, "Experimental Results for Local Filtering of Digitized Pictures," Digital Computer Lab Report 184, University of Illinois.

LINE AND ARC SEGMENT EXTRACTION

After an image has been reduced to a line drawing or its "cartoon" by the Edge and Line Enhancement procedures, the process of decomposing the line drawing and naming the constituents is begun. This process is performed by the LINE and ARC SEGMENT EXTRACTION procedure. One approach to implementing this procedure is through the "bug" and "pointer" technique.

A bug is made to scan across the image plane line by line beginning at the origin of the coordinate axis. When the bug encounters a black cell, indicating the presence of a line, object or noise element, a tracing process is begun. The bug traces the line and marks curvatures, intersections, terminations, etc., with identifying pointers. The bug always traces the line so that the enclosed area is kept to the right. After reaching the point where it began tracing, the bug proceeds to an intersection and traces a branch or continues its scanning of the image until all lines have been traced.

The output of the procedure is a list structure giving coordinates and pointer type for all cells that the bug has marked.

This procedure is the first step in decomposing the cartoon using a Name - Independent Parse.

Parameters: The parameters for this procedure, with the exception of the index of the input plane, are all concerned with the output of the list structure of coordinates and their associated pointers. It is assumed that some provision for dynamic storage allocation is available to accommodate the varying string lengths output from the procedure.

Parameter Call:

Call L ARC EX (IPI, A, k)

where:

IPI	-Index of the input plane
A	-List structure pointer
k	-Number of elements in the list

Reference:

Robert S. Ledley, "Pattern Recognition Studies in the Biomedical Sciences," Proceedings - Spring Joint Computer Conference, 1966.

NODAL EXTRACTION

Nodal Extraction is a process which utilizes the list structure information processed by the Line and Arc SEGMENT EXTRACTION procedure in addition to the image as processed by the LINE and EDGE ENHANCEMENT procedures. The dual input allows the Nodal Extraction procedure to operate on the list structure and rapidly locate intersections, terminations, vertices, etc. and then refer to the image for more critical analysis. This would provide a parallel processing approach such as template matching. Knowing where to look and approximately what to look for provides an advantage over procedures which scan the entire image in several directions with each template searching for a match. Templates can be 3 x 3 matrices or larger depending on the application.

Parameters: The parameters of the Nodal Extraction procedure include the input plane index and a list structure of nodes, descriptors and coordinates.

Parameter Call:

Call NODE EX (IPI, A, k, B, l)

where:

IPI	-Index of input plane
A	-list pointer from L ARC EX procedure
k	-number of entries in A list
B	-list pointer output from NODE EX
l	-number of entries in B list

Reference:

Adolfo Guzman, "Decomposition of a Visual Scene Into Three Dimensional Bodies," Proceedings Fall Joint Computer Conference, 1968.

LABELLING

Labelling is a higher level process than the pointer assignment routine in the LINE and ARC SEGMENT EXTRACTION procedure and the Nodal Extraction procedure. Where these procedures assigned pointers to all significant points on the image the LABELLING procedure combines segments and applies names such as ARC, T, Y, etc. This procedure is a list processing procedure with the input being the output of the previous two procedures and the output of this procedure also a list structure.

Parameter: The parameters associated with this procedure are list designators and parameters telling the number of elements in the list.

Parameter Call:

Call LABEL (A, k, B, , C, M)

where:

A	-list designator from L ARC EX procedure
k	-number of elements in A
B	-list designator from NODE EX procedure
l	-number of entries in B list
C	-list designator output from LABEL
m	-number of elements in C

References:

Hiroshi Genchi, "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting," Proceedings of the IEEE, Vol. 56, No. 8, August 1968.

Claude Fennema, "A Region Oriented Data Structure," Artificial Intelligence Group Report No. 7, Stanford Research Institute, May 1969.

CONNECTIVITY ANALYSIS

The CONNECTIVITY ANALYSIS procedure operates on the labelled elements output from the LABELLING procedure. It reassembles the component parts into objects forming a tree description of each object in the scene. This procedure is also a test processing procedure with the input being the output of the LABELLING procedure. As a result of applying the scene contents is now in a lattice form to which a Name-Dependent or Name-Independent recognition process can be applied.

Parameters: The parameters of this procedure are list structures with the input coming from the LABELLING procedure.

Procedure Call:

Call CONNECT (C, m, D, n)

where:	C	-input list designator from LABEL
	m	-number of elements in C
	D	-list designator output from CONNECT
	n	-number of elements in D

References:

John C. Schwebel, Bruce H. McCormick, "Analysis of Simple Line Drawings,"
Department of Computer Science File No. 796, April 11, 1969.

Rosenfield, "Sequential Operations in Digital Picture Processing,"
Journal of the Association for Computing Machinery, Vol. 13, No. 4,
October, 1966, pp. 471-494.

SKELETONAL EXTRACTION

Before the skeleton can be analyzed, the collection of points from the skeletonization process are sorted into a set of nodes corresponding to appearance and disappearance points of the skeleton, and a set of branches corresponding to lines connecting the nodes in the skeleton. Points at the intersection or junction of two branches are also regarded as nodes. This process is carried out by a program which examines adjacencies between points and assigns to each point a branch number or a node number.

A program now extracts the following information about each branch:

- 1) The nodes to which the branch is connected.
- 2) The time duration of the branch, defined as the difference in times of appearance of the two nodes at its ends.
- 3) The length of the branch, defined as the straight-line distance between end nodes.
- 4) The orientation of the branch, defined as the angle between the branch and a vertical line.
- 5) The average velocity of propagation along the branch, defined as the length divided by time duration.
- 6) The velocity as a function of time obtained at a finite number of points (8 points usually) along the branch.

The following information is then tabulated from each node:

- 1) The location of the node.
- 2) The number of branches connected to the node.
- 3) The time of appearance of the node.
- 4) The sum of the lengths of all branches connected to that node. This provides a measure of how closely the node is connected with the rest of the network.

After the above properties have been extracted, the description of the pattern consists of a list of numbers representing these properties. It is now possible for a program to perform a syntactic analysis on the above data to classify a pattern or make particular measurements on this pattern.

Reference: Otis Philbrick, "Shape Description With the Medial Axis Transformation," Pictorial Pattern Recognition, Cheng, Ledley, et al. Thompson Book Co., 1968.

CONTAINMENT RELATIONS

A means of analyzing patterns on a global basis is to categorize relationships of the various subsets of the picture and from these to form a global description of the picture. Relationships which are normally considered include "contains", "is contained in", "adjacent", "above" and "below", and other geometric type relations.

Three papers are described which utilize relation-finding algorithms in problem-solving and pattern analysis. The first of these, MANS, is a prototype Map Analysis System developed as an experimental general picture description system. It is discussed in "Local and Global Picture Processing by Computer", by John Pfaltz, et al.

An entity in a picture is defined to be any subset of the picture. Let L be a language whose set of well-formed sentences include statements about entities. A property of the entity E is a true sentence in L about E . A relationship between entities E_1 and E_2 is a sentence in L which contains the names of two entities, and is true if they are E_1 and E_2 respectively.

The problem of global picture description is as follows: Given a finite set of entities in a picture, and a finite set of properties of and relationships among the given entities, and a set of transformations within the language L . The objective is to generate a further set of true statements about the picture, involving the given entities as well as new ones which can be constructed from them.

For any given entity E of a picture, values are determined for the top, bottom, leftmost, and rightmost points of the entity. The listing of these points, with appropriate descriptions of each entity, allow the relations "contains" and "is contained in" to be analyzed among the various entities of the picture.

D. L. Londe and R. F. Simmons, in "Namer: A Pattern-Recognition System for Generating Sentences About Relations Between Line Drawings", describe a series of programs which are used to recognize line drawings and the spatial relationships between them. At the pattern recognition level the programs learn to associate a name with the pattern representing

a line drawing. At the relation-learning level the programs abstract characteristics which relate to such spatial concepts as "above", "left", "thicker than", etc. The names which have been learned in association with a drawing, and relation names which the program selects as true for the spatial relationships between two drawings, are substituted into a simple generation grammar; new sentences which are true for the picture are generated.

"Namer" is a series of programs which learn the names of line drawings and the spatial relations between pairs of drawings. "Namer" is divided into two main routines: "Trip", which recognizes and learns names for line drawings or figures, and "Name", which recognizes and learns names of spatial relationships between such drawings.

"Trip" accepts card inputs representing sketches on a matrix of 96 x 96 bits. It first expands and outlines the drawing. It then uses a set of algorithms to calculate 96 characteristics such as turn points, relative locations, projections, and indentations. During learning trials, the system is given a name for each type of input. "Trip" uses a probability learning technique to determine which subset of the 96 characteristics is most closely correlated with each name.

"Name" operates after "Trip" has recognized figures or line drawings. "Name" learns the names of spatial relationships which hold between two figures in a 96 x 96 matrix.

"Name" first characterizes each of two figures in the matrix by the lists of their x-y coordinates. The two sets of coordinates are then processed to compute means, average deviations, maxima, and minima for x and y values. Differences between these values for the two sets of coordinates are used to produce a set of 96 characteristics. These characteristics are derived as comparisons between variously sized and shaped parameters of the two pictures and general enough to be independent of the actual pictures themselves. The resulting characterization is called an RAV (relation attribute vector). Relation names, like pattern names, are inserted by human input. An adaptive program identifies which combination of vector bits pertain to any given single relationship.

Thomas Evans, in "A Heuristic Program to Solve Geometric-Analogy Problems", describes a relation analysis program which solves a wide class of so-called "geometric-analogy" problems frequently

encountered on intelligence tests. Each member of this class of problems consists of a set of labeled line drawings. The task performed is described by the question: "Figure A is to Figure B as Figure C is to which of the given answer figures?"

The program "Analogy" proceeds to solve the problem as follows: First the input description of the figures are read. Currently these descriptions are given as LISP expressions: they could be generated from scanners by programs using line-tracing techniques. The description represents the figures in terms of straight line segments and arcs of circles.

The first step is to decompose each problem figure into objects or subfigures. The decomposition program separates a problem figure into its connected subfigures. Next, the objects generated from scene decomposition are presented to a routine which calculates a specified set of properties of these objects and relations among them. This set can be changed. As a sample of a relation-calculating subroutine, one is presented which calculates that one given object lies inside another object. The method used involves calculating all intersections with the outer object of a line segment drawn from a point on the inner object to the edge of the field. Since the number of such intersections is odd, namely one, and the outer object is a simple closed curve, the first object is inside the second object. A repertoire of analytic geometry routines is required to determine intersections of straight line segments and arcs of circles in all cases and combinations. Other relation routines calculate the relations above, below, left, right, etc.

This program also determines a set of 'similarity' calculations. For each appropriate pair of objects, all members from a certain class of transformations which carry one object of the pair into the other are determined. The elements of this set are compositions of Euclidean similarity transformations (rotation and uniform scale change), with horizontal and vertical reflections. Given descriptions of virtually any pair of arbitrary line drawings x and y , these routines calculate the parameters of all transformations that map x into y .

This routine is a pattern recognition program with built-in invariance under scale changes, rotations, and certain types of reflections.

After the similarity information is computed for every required pair of objects, this information is used as input to the analysis portion of the program, to determine the correct answer to the analogy problem. The first step is to generate a rule with alternates, transforming figure A into figure B. Such a rule specifies how the objects of figure A are removed, added to, or altered in their properties and relations to other objects to generate figure B. The next task is to generalize each rule so that the resulting rules take figure A into figure B and now take figure C into one of the answer figures.

Reference

J. L. Pfaltz, J. W. Snively, Jr. and A. Rosenfeld, "Local and Global Picture Processing by Computer", in George Cheng, R. S. Ledley, Donald K. Pollock, and A. Rosenfeld, Pictorial Pattern Recognition, Thompson Book Co., Wash. D.C., 1968.

D. L. Londe and R. F. Simmons, "Namer: A Pattern-Recognition System for Generating Sentences About Relations Between Line Drawings", Proceedings of the 20th National Conference, ACM, August, 1965.

T. G. Evans, "A Heuristic Program to Solve Geometric-Analogy Problems", AFIPS Conference Proceedings, Vol. 25, 1964 SJCC

Other Reference

Adolfo Guzman, "Decomposition of a Visual Scene into Three-Dimensional Bodies", AFIPS Conference Proceedings, Vol. 33, Part 1, 1968 EJCC.

NEIGHBORHOOD RELATIONS

Neighborhood relations represent an alternative technique to the linguistic approach to cognition, as well as an aid to pattern classification as an intermediate step in pattern processing. The linguistic approach to pattern recognition involves the representation and reduction of a pattern into a linear string and the manipulation of this string utilizing linguistic techniques in order to perform cognition of the pattern. Three reports by John C. Schwebel and Bruce McCormick, and John C. Schwebel discuss the properties of relations which lead directly to a graphic representation of a pattern in an algebraic form.

The first report, "Consistent Formal Properties of Binary Relations," defines 13 formal properties of binary relations, 9 of which are independent. The properties listed are those of reflexive properties, symmetry properties, cyclic properties, and transitive properties. A listing is presented of the 40 logically consistent combinations of these nine properties, and the theorems which lead to the discarding of the inconsistent combinations. In addition illustrations of image processing relations which are logically consistent are presented, such as above, below, completely outside of, far from, subset of, etc.

The second report, "Consistent Properties of Composite Formation Under a Binary Relation," defines properties of composite formation for binary relations and categorizes relations by which of the set-theoretic properties these relations preserve. Composite elements are formed by containment relationships among other set elements, and depend on the structure on the set.

The third report, "Graph Transformations for Composite Formation," includes an example which utilizes the previous report to construct an abstract graph of a picture from a basic set of initial relations, and to find relations among composite objects.

The four relations used in this report are: above, intersects, proper subset of, and left of. The input to this relationship manipulation program consists of an initial abstract graph whose nodes represent elements of the picture and whose branches represent the four image processing binary relations given above.

The input graph could be constructed by an algorithm which examines local features of individual picture elements to determine relations between adjacent items. Using these input relations, and the logic and set-theoretic properties which these relations preserve, it should be possible to infer additional and hopefully global relations on the input picture.

The first goal of this example is to form composites of the individual nodes, and to determine the relationships between these composites. Successive combination produces more global representations for the original picture, leading to a linear representation of the picture. Such a linear representation could be considered an end result of this process and would be equivalent to a syntactic representation as discussed in the linguistic analysis previously.

References:

Bruce McCormick and John Schwebel, "Consistent Formal Properties of Binary Relations," File No. 762, Dept. of Computer Science, University of Illinois, Urbana, Ill., June 6, 1968.

John C. Schwebel and Bruce H. McCormick, "Consistent Properties of Composite Formation Under a Binary Relation," Report No. 348, Department of Computer Science, University of Illinois, Urbana, Illinois, August 12, 1969.

John C. Schwebel, "Graph Transformations for Composite Formation," Report No. 368, Department of Computer Science, University of Illinois, Urbana, Illinois, December 12, 1969.

TEXTURE ANALYSIS

Texture is an attribute of a visual field comprised of very small but discriminable spatial variations in hue and brightness. Another way of defining texture is that texture refers to the patterns of information or arrangement of the structure found in a picture. These characteristics are relatively simple for a human to distinguish at a glance, but are relatively difficult in automatic pattern recognition.

Most of the work in texture analysis is involved with analyzing the techniques used by humans in distinguishing patterns based on texture, so that pattern recognition can utilize these techniques.

Bela Julesz, in "Visual Pattern Discrimination", constructed experiments to investigate the discriminability of unfamiliar visual patterns. Two visual fields are generated either side-by-side or one is contained in the other. The first arrangement is used when the fields differ along the horizontal dimension; the second display whenever the fields differ in two dimensions. The problem studied was: If two visual fields are presented simultaneously, in what properties must they differ in order to be discriminated?

Discrimination in this application requires a high degree of spontaneity. One way of obtaining patterns devoid of familiarity cues was to generate them from a stochastic process. Such processes are specified by their n 'th order joint probability distribution. In the first two experiments tried, the random fields were described by their first order probability distributions. Slight differences in first-order statistics resulted in visual discrimination.

The next experiment was in fields of Format B (contained one within the other) with identical first order distributions (black, gray, and white dots of equal probability) but differing in their second order distributions. Any two points in the smaller field are statistically independent of each other in brightness value, while the larger field was a Markhov process having a second-order probability distribution. Because the second-order distribution determines the power spectrum of the stochastic process, this experiment compares a field with a flat power spectrum with a field which has a low-pass spectrum along the

horizontal direction. The two patterns are easily discriminated and the effect is regarded as a difference in granularity.

Efforts are made to see if patterns with higher order statistics could be differentiated visually. The experiments indicated that in visual discrimination, clusters or lines formed by local areas of uniform brightness play a decisive role. Cluster and line detection are called connectivity detection.

In addition to connectivity, other properties of regions, such as size, average length and width, orientation, density are used to perform visual discrimination.

A. Rosenfeld, in "Automatic Recognition of Basic Terrain Types from Aerial Photographs", describes a study of the feasibility of automatically identifying various types of terrain on the basis of texture analysis. Typical specimens of terrain types such as hydrographic features, cultivated and uncultivated land, and types of urban areas, can be recognized at a glance by an observer with a minimum of training. The hypothesis tested is that each type scene has a characteristic visual texture.

The visual texture of a portion of aerial photograph seems to be determined by

- a) overall average density of the portion
- b) mean frequency with which contrasts (local departures of density from the average) of varying degrees occur within the portion.

To measure the contrast frequency, the photograph is scanned and the density information converted into a time varying video signal. Visual texture is computed by differentiating, level slicing, and averaging. Differentiation converts contrasts of varying degrees into peaks of varying heights. Level slicing and averaging are applied to determine the average number of peaks of approximately a given height per unit area of the photograph.

The analysis was carried out on assorted portions of aerial photographs showing essentially uniformly textured regions of terrain. Two specimens each of urban residential, industrial, woods, and cultivated

fields were analyzed.

Results showed

- a) Urban residential areas have characteristically high-contrast frequencies.
- b) Cultivated fields have characteristically low frequencies and relatively low contrasts.
- c) Industrial areas and woods have very similar contrast frequency spectra.

Other examples of experimental studies in texture identification and analysis and applied studies in texture analysis are referenced below.

References

- Bela Julesz, "Visual Pattern Discrimination", IEEE Transactions in Information Theory, IT-8, Feb. 1962.
- A. Rosenfeld, "Automatic Recognition of Basic Terrain Types from Aerial Photographs", in Photogrammetric Engineering, v. 28, March, 1962.
- A. Rosenfeld and Goldstein, "Optical Correlation for Terrain Type Discrimination", in Photogrammetric Engineering, v. 30, June, 1964.
- R. M. Pickett, "The Perception of Random Visual Texture", in Models for the Perception of Speech and Visual Form, Wather-Dunn, Editor, M.I.T. Press, Cambridge, Mass., 1967.
- Steiner and Haefner, "Tone Distortion for Automated Interpretation", in Photogrammetric Engineering, v. 31, March, 1965.

SHAPE ANALYSIS

Shape analysis, also called contour analysis, is involved with those techniques which analyze the line and arc and node features extracted in pattern articulation, and categorize these features into an identification of the original pattern or into a form for further analysis. Several examples will be described of the use of shape analysis in character recognition, automatic fingerprint interpretation, and in the solution of "apictorial" jigsaw puzzles.

E. C. Greanias, P. F. Meagher, R. J. Norman, and P. Essinger, in "The Recognition of Handwritten Numerals by Contour Analysis" describe the use of contour analysis. Handwritten characters are scanned on a stationary document by a flying spot scanner. These characters are scanned along a variable path that depends on the information obtained as scanning proceeds. Scanning starts with a search pattern that looks for a character on the document. When a character is detected, the search pattern stops and the beam follows the edge of the character with an exploratory circular motion. The signals that are passed on to the categorizer for analysis are essentially the beam deflection signals, with the circular pattern filtered out.

To accomplish recognition, the contours of each character are traced two or more times. During the first tracing, the extreme positive and negative excursions of the character's outer outline in x and y are stored. On subsequent passes around the character, the beam is located relative to the character extremities and is not dependent on character location or initial contact point. The stored peaks of the character contours provide a measure of character size and is used to normalize the vertical and horizontal distance measures of the system.

The analysis of the beam deflection signals (input data to shape analysis routines) is based on two properties of line edges they represent:

- a) relative location of the line edge within the character
- b) approximate direction of the line edge.

The vertical deflection range is divided into four equal increments and the horizontal deflection range into three equal increments by comparing the filtered deflection signals on subsequent passes with voltage thresholds defined by the resistor divider networks. These thresholds define twelve possible zones in a three by four array that the deflection beam enters as it travels around the characters.

Line directions are determined by generating the time derivative of the filtered horizontal and vertical deflection signals. A comparison is made of these derivatives to determine into which of eight 45 degree sectors the beam is directed. Recognition is based upon selected sequences of line directions and locations detected in the second pass around the character.

Significant features (line direction and location values) were identified for each type character by the use of simulation on known characters. Shape feature tests were defined and recognition logic assembled which satisfied 2752 of the 3000 characters in the sample. Shape features emphasized for each character are given below.

Shape Features for each Character

Significant Feature	Location	Use
long N, NE, or NW strokes	right side	0,1,7
long S, SW, or SE strokes	left side	0,1,7
horizontal line end	top left	2,3,7
horizontal line end	top right	5,6
horizontal line end	bottom left	3,5
horizontal line end	bottom right	2
loop	bottom	2
west bay	bottom left	3,5
north bay	top center	4
northeast bay	top right	9,8
short arc	top left	9
NE stroke	top left	4
horizontal line end	right center	4
east bay	top right	6,5
notches	left	8
notches	right	8
southwest bay	bottom left	4,9

Significant Feature	Location	Use
short arc	bottom right	6,5
large lake	center	0
small lake	top center	8,9
small lake	bottom center	8,6

Determination of the particular shape features listed above from the input excursion data can be considered the shape analysis portion of the program. Character recognition consists of the sequential recognition of the above shapes with an appropriate time interval for each shape; this feature would correspond to discriminant function analysis. The characters 2,3,4,5, and 7 are recognized by the above techniques. When the possible occurrence of 0,1,6,8, or 9 is detected, scanning of the interior contour is performed. The information derived from the inner contour scan is used to define shape feature tests similar to those listed above, and final recognition takes place when appropriate inside shape features are detected.

H. Freeman and L. Garder, in "A Pictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition" uses shape analysis for the solution of a pictorial jigsaw puzzles, puzzles whose pieces supply no pictorial information, only shape. The techniques used in solving problems of this type have great application of pattern recognition.

The objective of any jigsaw puzzle is the arrangement of a set of given pieces into a single, well-fitting structure, with no gaps left between adjacent pieces. Any puzzle is completely characterized by knowledge of all piece boundaries plus the mating relationships between the boundaries. Piece boundaries are given; the mating relationships are to be determined.

The mating relationships of a particular jigsaw puzzle can be represented by means of a mating diagram. Each puzzle piece is replaced by a node, and mating boundaries between pieces become branches connecting the corresponding nodes. An extra node is provided for the exterior; branches to this node correspond to the exterior boundary contributions from each piece.

Radiality refers to the kinds of interior and exterior junctions in the assembled puzzle. An n-radial junction is the junction of n-boundary lines at a junction point. At every interior n-radial junction ($n \geq 3$) of a jigsaw puzzle, the boundaries of at least $n-2$ of the pieces must possess a slope discontinuity. The knowledge that there must exist a minimum of one slope discontinuity at every interior junction is of great importance in the required solution.

In a given puzzle, a large number of the pieces must possess one or more slope discontinuities. Since the extent of mating between two pieces is defined by two junctions, different types of matings are distinguished on the basis of the presence or absence of slope discontinuities at the junctions. Six possible types of matings are defined.

The first step of the solution is the encoding of the individual pieces. The chain-encoding technique is utilized. In this scheme the individual piece is placed on a square grid and the intersection of boundary with the grid marked. The grid node closest to each intersection is chosen as a curve point. Successive points are connected by means of straight lines to form a chain. With eight possible directions in going from one curve point to an adjacent one, the succession of straight line segments can be represented as a sequence of octal digits. All contours are traced in a clockwise sense.

Since mating between two pieces usually occurs over only a small fraction of their respective boundaries, each complete chain is separated into a set of chainlets, each of which is likely to mate with one and only one chainlet from another piece. The best guide for forming chainlets is provided by the slope discontinuities. Points at which a chain is separated into chainlets are called critical points, and can be determined by computer program.

When two pieces mate on a type IV, V, or VI basis, a slope discontinuity appears in the composite boundary at a point at which neither of the individual pieces had a critical point. The points at which such discontinuities are found are called secondary critical points, and are used to form additional chainlets. It was not stated how such secondary critical points could be determined.

Segmentation into chainlets changes the solution of puzzles into that of mating chainlets (or portions of chainlets) subject to certain independence constraints. The number of chainlets is many times the number of chains. Instead of making an exhaustive search for mating chainlets, the chainlets are first classified according to some of their measurable, orientation-invariant features. Typical features are the length of the chainlet, the length of the straight line between the end points (distance vector) of the chainlet, the number of times the chainlet crosses its distance vector, the ratio of areas enclosed by the chainlet to either side of the distance vector, and the perpendicular distance between chainlet and distance vector.

The set of m features of a chainlet can be expressed as a vector in m -dimensional feature space. Since similarity of features is a necessary condition for type I mating, the feature vectors of two chainlets which mate on a type I basis must lie in close proximity. Conversely, if no feature vector lies in close proximity to the feature vector of a given chainlet, that chainlet either forms part of the exterior boundary or mates on a basis other than type I. The generation of the feature vector for each chainlet and the comparison techniques to other feature vectors can be considered the shape analysis portion of this problem.

Let the feature vector of chainlet i be denoted by F_i , and its components by F_{ik} ($k=1,2,\dots,m$) where m is the total number of features. Two chainlets i and j are considered similar if

$$|f_{ik} - f_{jk}| < T_k$$

is satisfied for all k , $k=1,\dots,m$. T_k is a positive threshold assigned to the k th feature.

The feature separation S_{ij} between two vectors F_i, F_j is given by

$$S_{ij} = \sum_{k=1}^m [f_{ik} - f_{jk}]^2 w_k$$

where w_k is a nonzero weighting factor assigned to the k th vector.

The thresholds are assigned on the basis of the precision with which various features can be determined. The weighting coefficients

are assigned on the basis of the probability distributions associated with the corresponding feature.

The procedure for finding a mate to a given chainlet on a type I basis is as follows: 1) Threshold tests are applied to the feature vectors of all chainlets. Those chainlets similar to the given chainlet are listed as mating candidates.

2) Feature separations between the chainlet and each candidate are calculated. Similar chainlets are listed in order of increasing feature separation.

3) The candidate chainlets in (2) are compared until a mating chainlet is found.

If no mating chainlet is found among the candidates, it is assumed that the chainlet either mates on a basis other than type I, or that it is on the exterior boundary.

In type I mating, paired chainlets mate over their full length. In type II or IV mating, paired chainlets mate only over a portion of their respective ranges, starting from one common point. To analyze type II or IV mating, subfeature vectors are introduced. These vectors are analogous to the feature vector except that they refer to specific fixed length initial and terminal portions of a chainlet. The length of the initial or terminal portions is taken as some small fraction of the average chainlet length of the particular puzzle.

The solution of apictorial jigsaw puzzles consists of a succession of boundary matings and assemblies of matching pieces. Initially, mating is attempted on a type I basis. When this fails subfeature vectors are formed and type II matings are sought. Matings of type III through VI may require the use of secondary critical points.

The contour comparison procedure consists of three steps, and is used to determine the mate of a given chainlet after the candidate chainlets have been ordered according to their feature separation from the given chainlet.

In the first step, the direction of candidate chainlet is reversed. In order to permit direct comparison between a given chainlet and a mating candidate, the latter must be inverted. In the second step the candidate is rotated to bring it into alignment with the given chainlet. After inversion and rotation, the two chainlets are compared on an element by element basis. The squared geometric distances between successive points along the two chainlets are calculated. The sum of these squared distances is a measure of the resemblance between the two chainlets.

The assembly of the puzzle is based on an outward spiraling growth centered around one piece. This technique is based on the concept of junction closure. A junction is any critical point located on the exterior boundary of a cluster of pieces. The junctions on the cluster are examined sequentially in a counterclockwise sense and an attempt is made to close them, surrounding them completely with mutually mating pieces. As each junction is closed, a new open junction is selected and the process repeated until all pieces are included.

The probability of occurrence of an invalid mating around a junction is decreased by investigation of alternate possibilities for junction closures. Whenever there are several seemingly acceptable choices, a junction figure of merit is determined for each assembly combination. The combination with the highest figure of merit is selected; alternate assembly combinations are retained. When an assembly error is detected, the alternate combinations are examined in search for the correct assembly.

Comparison of the chainlets on the basis of squared distances between points on the chain, closure tests on the chainlets and puzzle pieces, and global junction relationships could be considered the discriminant function analysis portion of this program.

W. J. Hankley and J. T. Tou, in "Automatic Fingerprint Interpretation and Classification Via Contextual Analysis and Topological Coding," utilize contour analysis to categorize fingerprint characteristics.

Following the conventions of the Henry system of classifying fingerprints, there are four basic fingerprint pattern classes: loops,

whorls, arches, and accidentals. Loops are subclassified as radial, loops open toward the thumb side of the hand, and ulnar, loops which open toward the other side of the hand. Arches are subclassified as regular and tented, characterized by upswept inner ridges. Accidentals are composites of the three other basic forms. A loop ridge is characterized by both branches tending to run off the same side of a fingerprint, as compared to an arch ridge.

The Henry system further subclassifies whorls and each of the two loop types. Whorls are classified I,O,M (inner, outer, or meet) based upon a line running from one delta and above, below, or toward the other delta. Loops are classified S,M,L (small, medium, or large) based upon a ridge count. An imaginary line is drawn from the core to the delta, where the core is either the top of the centermost rod, or, in the absence of rods, the centermost loop shoulder which is furthest from the delta. The ridge count is the number of ridges intersected by this imaginary line, excluding the core and the delta.

The Henry system is based entirely upon the topological structure of the print. In identification of single prints, the objective is to compare and possibly match two given fingerprints. Both identification and classification of single prints are based upon the pattern of print minutiae. These include ridge ends, forks (bifurcations) and dots.

The proposed automatic fingerprint processing system consists of three basic subsystems--input, interpretation, and topological coding. The input accepts an actual fingerprint and yields a partially filtered digitized fingerprint matrix. The interpretation subsystem searches the digitized input along the ridges and interprets and corrects ridge gaps and contiguous ridges, and detects print topological features. The topological coding subsystem reads the processed line pattern and generates a code sentence which describes sequentially the topological structure of the line pattern.

The interpretation program operates on an input pattern matrix and analyzes the contour pattern imbedded in an array of points. The objective of the program is to generate the directed line pattern representing the print. All lines are directed downward from locally uppermost points

called top nodes. These top nodes are points of locally maximum curvature. Additional types of nodes are labeled A through F. A, B, and C are line intersection nodes, and point D is a line end node. E, F are radial nodes which are intersections of the line with a circular boundary with center at the print core. The degree of a node is the number of downward lines which emanate from the node.

The interpretation program generates an ordered list of top nodes, a succession of line element directions emanating from each node, with associated node intersections of lines, and a list of terminating nodes which represent end points for each line.

The fundamental interpretation program operation is ridge tracing together with line coding. This is performed by a subroutine, TRACE, which allows for choices of starting and stopping conditions and modes of ridge following, where the normal mode is to trace along the convex edge of a ridge. The interpretation program is a line generation program composed of recurrent applications of TRACE, path scanning and transfer operations. Ordering of these operations and setting TRACE conditions is controlled by using such information as the state of various input data words, line angles and curvature, and length of test scans, where the length of a path or line is the number of points which constitute the path or line.

The interpretation program, then, is involved with the shape analysis of the ridges, generation of the overall line pattern, identification of the print type, and determination of the fingerprint core.

The topological coding program operates upon the processed line pattern and the list of tops and generates a code sentence which describes the topological structure of the central part of the fingerprint. Such description is under control of a syntactic-like language.

References:

- E. C. Greanias, P. F. Meagher, R. J. Norman, P. Essinger, "The Recognition of Handwritten Numerals by Contour Analysis", IBM Journal of Research and Development, Vol. 7, No. 1, Jan. 1963.
- H. Freeman, L. Garder, "Apictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition", IEEE Transactions on Electronic Computers, EC-13, No. 2, April, 1964.
- W. J. Hankley and J. T. Tou, "Automatic Fingerprint Interpretation and Classification via Contextual Analysis and Topological Coding", in Pictorial Pattern Recognition, by G. Cheng, R. Ledley, D. Pollock, and A. Rosenfeld, Thompson Book Company, 1968.

SYNTACTIC AND SEMANTIC ANALYSIS

Syntactic analysis is involved with the analysis and identification of patterns, based on the application of linguistic processes to encoded line patterns, usually in string form. Semantic analysis is usually involved with the calculation of parameters for the various sections of the syntactic analysis, combined with a comparison of the values of the parameters calculated to parameter limits. Thus an object is determined, i.e., recognized, by a listing of the syntactic properties with an evaluation of values within set parameter limits.

Two reports detail the application of the above techniques to various problems of pattern recognition. The first of these is the "Syntax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics," by Robert H. Anderson, and the second is "Linguistic Specification and Analysis of Classes of Line Patterns," by Jerome Feder. These are discussed in detail below. Other examples of syntax-semantic techniques of pattern recognition are also presented.

"Syntax-Direct Recognition of Hand-Printed Two Dimensional Mathematics" is involved with automatic recognition of two-dimensional mathematical formulae with the printout of an English-like description of the recognized expression. The input consists of each recognized symbol, i.e., mathematical symbol, or letter or number, with its corresponding coordinates in an x-y coordinate system. These coordinates consist of each symbol's x-min, x-center, x-max, y-min, y-center, and y-max. The syntax-directed parsing algorithm attempts to recognize the string of symbols as being a valid terminal sentence of a grammar by determining how the goal (initial group, totality) of the grammar can produce that symbol string. If such a production is exhibited, it is called a parse of that string.

The parsing algorithm for two-dimensional character recognition is top-down. The syntax rules for two-dimensional analysis specify replacements of syntactic categories contingent upon their correct relative placement, that is, contingent upon an examination of their relative coordinates. The process consists of identifying the original expression, or goal, and classifying its coordinate parameters. The algorithm then attempts to separate the expression into two or more

subexpressions satisfying a syntactical relationship according to a syntactical rule and returning the coordinate parameters of each of the sub-expressions. The semantic phase of each parse step is to test the placement of each individual character of the original expression within the coordinate set of the respective subnetworks; if this placement or position test fails then the particular syntactical rule is non-applicable, and another rule is tried on the original expression. The output of each step of the parsing algorithm is the set of coordinates of each sub-expression, the listing of characters belonging to each sub-expression, and the string representation of the particular type decomposition successfully recognized in each step. When all expressions and sub-expressions have been successfully parsed to terminal characters, the final string representation represents a successful recognition of the original mathematical expression. An example of a step of this recognition procedure is given in Figure 1.

"Linguistic Specification and Analysis of Classes of Line Patterns" is involved with the application of a two stage table-driven pattern analyzer to partition a pattern into segments and then to reconstruct the characteristics of the pattern structure from these elementary segments. The output, then, is the extraction of desired information from the constructed list model of this pattern. As such, this is an example of a bottom-up parsing strategy.

The parsing of the table driven pattern analyzer is performed in two stages, with a separate metagrammar in each stage. The input data consists of chains of characters C_i , where the C_i are the octal digits that are the elements of the chain. Each chain is a line or boundary of the original picture. It is assumed that preprocessing has reduced the pattern to the sets of chains of the above form. In the first parsing stage, chains are parsed into strings of NAPES, each NAPE representing a portion of the chain. A NAPE is the representation of symbol, say a line, with its associated attaching points, a left one and a right, at which it joins to or associates with other symbols. The more general representation of a NAPE is that symbol which has an arbitrary number n of attaching points. This representation is used in this described report.

The first stage parsing algorithm, then, processes the chains of the input structure one at a time. Attempts are made to parse each chain to correspond to some language product in the string table. Parse 1, first stage parse, starts with the first linguistic entry and advances to a succeeding entry when an attempt to match fails. A chain is scanned in two directions, from the first element to the last, and from the last to the first. Thus the algorithm is not dependent on the directed nature of the chains.

The list structure that results from the successful operation of the first stage parse is similar to the input structure except that each chain is represented by a number of lists called terminal blocks which are linked together to form a model of the chain. Each terminal block corresponds to one component of the product language into which the chain has been parsed. The terminal block can be considered as representing a NAPE with two attaching points and a number of associated parameters.

The second-stage parsing algorithm regards the first stage structure as a terminal plex (structure); the terminal blocks are considered as terminal NAPES. Parse 2 looks for the terminal blocks in the first-stage structure which correspond to the components of a linguistic definition. In effect terminal NAPES are linked together to form higher level structures according to various parameter tests. Parameters are generated for these non-terminal NAPES which are then used in building up higher structures. The parameter tests could be considered the semantic tests of the second-stage analysis. The result of a successful parse is a list model that describes the hierarchical structure of the input according to the second-stage grammar.

Ledley, in "Pattern Recognition Studies in the Biomedical Sciences," uses a syntax directed pattern recognition technique to build up "derived parts" of chromosomes. This procedure results in the recognition of the ends of the arms and the centromere position.

When a chromosome is located, the boundary of that chromosome is traced. The boundary is characterized by a sequence of boundary segments.

(a)

term		
S1: expression	P1: $c_{01} > c_{21}$ and $c_{03} < c_{23}$	C1: c_{21}
	and $c_{04} > c_{26}$	C2: c_{22}
S2: horizline	P2: $\emptyset$	C3: c_{23}
S3: expression	P3: $c_{01} > c_{21}$ and $c_{03} < c_{23}$	C4: c_{34}
R: $\emptyset$	and $c_{06} < c_{24}$	C5: c_{25}
M: $(m_1)/(m_3)$		C6: c_{16}

(b)

Figure 1: Graphical and tabular form of replacement rule

C_{01} = $X_{\min}$ of first character, or arbitrary character

C_{21} = $X_{\min}$ of terminal character horizline

C_{03} = $X_{\max}$ of arbitrary character

C_{23} = $X_{\max}$ of terminal character horizline

C_{04} = $Y_{\min}$ of arbitrary character

C_{26} = $Y_{\max}$ of terminal character horizline, etc.

Each segment is characterized by the coordinates of its center point, components of a leading vector and components of a trailing vector. The number of points selected is supplied by the user, and are sufficient such that the angle between the leading and trailing vectors is approximately a measure of the curvature of the segment.

The vector sum of the leading and trailing vectors is approximately the tangent to the segment at the center point and gives a measure of the direction of the segment. A boundary-characterization list is constructed as each boundary segment is analyzed successively, until the original boundary entry point is reached again.

The recognition of the object by the "FILACSYS" programming system is accomplished by means of the boundary-characterization list. A syntax directed pattern-recognition technique is utilized.

The following description of this technique is taken from Ledley, Use of Computers in Biology and Medicine.

Five characteristic parts of the boundary contour of the object are identified by the program. The first is a clockwise curve, the second a straight line, the third a counterclockwise curve, the fourth is a notch, and the fifth a "wide" clockwise curve. This can be considered the shape analysis portion of the program. The five curves are known as types A, B, C, D, and E respectively.

The syntax phase of the analysis utilizes the sequence of parts generated in the shape analysis phase to identify the type of chromosome. An arm must have a type A part at its end, and a pair of arms must have a type C part in between.

Representations for submedian and teleocentric chromosomes are given on the following figures, along with the syntactic definition for these kinds of chromosomes. If a pattern is parsed and does not satisfy the syntactic description, it is removed (erased) as not being a chromosome. Following the syntactic analysis of a chromosome, measurements of its are and centromere position are performed.

In "A Picture Calculus," by W. F. Miller and A. C. Shaw, a picture description language is described which describes the concatenation of the primitive elements of a picture. Picture primitives are specified

Fig. 10-2. Submedian and telocentric chromosomes, with their parts list below each.

The following illustrates "syntactic definitions" for these kinds of chromosomes:

$$\begin{aligned}
 \langle \text{arm} \rangle &::= A|B \langle \text{arm} \rangle | \langle \text{arm} \rangle B \\
 \langle \text{side} \rangle &::= D|B|B \langle \text{side} \rangle | \langle \text{side} \rangle B \\
 \langle \text{bottom} \rangle &::= E|E \langle \text{bottom} \rangle | \langle \text{bottom} \rangle B \\
 \langle \text{pair} \rangle &::= \langle \text{arm} \rangle C \langle \text{arm} \rangle \mid \langle \text{side} \rangle \langle \text{arm} \rangle \mid \langle \text{arm} \rangle \langle \text{side} \rangle \\
 \langle \text{submedian chromosome} \rangle &::= \langle \text{arm} \rangle \langle \text{arm} \rangle \mid \langle \text{pair} \rangle \langle \text{pair} \rangle \\
 \langle \text{telocentric chromosome} \rangle &::= \langle \text{arm} \rangle \langle \text{bottom} \rangle \mid \langle \text{pair} \rangle \langle \text{bottom} \rangle
 \end{aligned}$$

Figure 2: Syntactic Definition for Submedian and Telocentric Chromosomes.

in terms of primitive classes which in turn are specified by an attribute list. The attribute list contains the class name, a tail and head specification, and an arbitrary list of additional attributes. Primitives are concatenated together only at their heads and/or tails. A picture described in the picture description language is represented as a labeled, directed graph with the primitives as directed edges pointing from head to tail.

A picture parse (syntactic analysis) is directed toward recognizing those primitives which are grouped together in allowable structures according to the grammar. A top down parse of a picture commences from a starting symbol and generates from left to right sentences S in the language according to the grammar. When the goal is a primitive class name, a search routine is called to determine whether a primitive of the class sought is located at the specified location. If there is a primitive present, its value is returned to the value list.

The output of the picture description language consists of the values of the primitives found, the string description of the picture, and the parsing tree.

References:

- Robert H. Anderson, Syntax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics, Ph.D. Thesis, Harvard University, Jan. 1968.
- Jerome Feder, Linguistic Specification and Analysis of Classes of Line Patterns, April, 1969, Technical Report 403-2, N.Y. University School of Engineering and Science, Dept. of Electrical Engineering, Lab. for Electrosience Research.
- Robert Ledley, Use of Computers in Biology and Medicine, McGraw-Hill, Inc., 1965.
- Robert Ledley, Louis Rotolo, et al, "Pattern Recognition Studies in the Biomedical Sciences," Proceedings of the S.J.C.C., 1966.
- W. F. Miller and A. C. Shaw, "Linguistic Methods in Picture Processing - A Survey," Fall Joint Computer Conference, 1968.
- W. F. Miller and A. C. Shaw, "A Picture Calculus," in Don Secrest and Jurg Nievergelt, Emerging Concepts in Computer Graphics, W. A. Benjamin, Inc., N.Y., 1968.

DISCRIMINANT FUNCTION ANALYSIS

Discriminant function analysis is involved with the classification of patterns into several separate classes by the discrimination for particular features or with respect to statistical parameters. Several examples of applications utilizing discriminant function and analysis have already been given and several others will be described below.

In "An Experimental Investigation of Video Pattern Recognition", by E. M. Darling, Jr., and R. D. Joseph, discriminant function analysis is used with six different decision mechanisms to differentiate between the following pattern classes: lunar craters with flat floors, craters with central peaks, rima, wrinkle ridges, noncumulus clouds, cumulus polygonal cells, and cumulus solid cells. Five two-class recognition tasks were defined on these patterns.

A recognition system was designed for each task. Subspaces for property filters were selected randomly, switching surfaces for property filters established by discriminant analysis, decision function on property filters by loss function minimization, and sequential selection of property filters by loss function reduction.

A set of 28 pattern discriminators was designed. These discriminators were patterned after the aspects which a human might use for recognition. The first 15 discriminators were general in nature, the remaining 13 chosen for cloud recognition. These sets of discriminators were called known property filters. Linear decision functions were applied to all 28 filters to discriminate the cloud recognition tasks.

Hiroshi Genchi, Ken-Ichi Mori, S. Watanabe, and S. Katsuragi, in "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting", describe the recognition of characters by means of sequential recognition logic. This logic can be thought of as an application of discriminant function analysis.

The first procedure in character recognition is to detect the direction of each stroke segment while smoothing the raggedness of the stroke edge. A 3 x 3 matrix detects the 512 possible kinds of partial patterns and classifies them into seven stroke segment directions. A character pattern expressed by 32 x 32 mesh points is scanned by the 3 x 3 submatrix, each bit horizontally from left to right, and every second bit vertically from top to bottom. Thus a character field is divided into 16 horizontal zones each of which consists of a string of 32 stroke segment directions.

The second step of the recognition procedure determines a feature from the horizontal sequence of 32 stroke segment directions. A string of stroke segment directions is classified into one of 16 horizontal features.

Along with the 16 feature descriptions of the horizontal relations with the zone, a feature which describes the vertical connective relation of the strokes is extracted. A flip-flop is set if the left end of the stroke in one zone is connected within + 1 bit to that in the upper adjacent zone.

The string obtained by feature extraction is then supplied to the sequential recognition logic. Each figure has a description diagram which describes the figure in terms of the above sixteen features. Potential variations of each figure are included in this description logic. The input sequence of features is compared with every transition diagram to determine the best match. A penalty count system is employed each time an input improper to the diagram occurs in following the transition diagram. The minimum value of penalty, if less than two, determines the unique numeral.

T. Sakai, M. Nagao, and S. Fujibayashi, in "Line Extraction and Pattern Detection in a Photograph", perform a pattern recognition for front-facing human faces.

The recognition principle in this report is based on template matching. Picture patterns extracted are usually strongly deformed, so that simple pattern matching does not produce good results. If there is a portion of a pattern with a specific characteristic corresponding to a portion of a standard reference, then with adequate reliability, it is

recognized as such, even if the other portions of the figure have no definite correspondence. The pattern is divided into several subpatterns, for each of which the degree of matching is calculated. A complex template for the human face is adopted, which is subdivided into eyes, nose, mouth, contour of the face, and a non-existence feature.

The method of recognition is as follows. A picture is scanned and positions detected which match very roughly to the two subtemplates associated with the contour of the face. Then the matching with the other subtemplates is performed. The matching is to count the existence of points in a standard subtemplate and to standardize the count in percentage by the total meshes of the standard subtemplate. In order to identify a face, each of the individual subtemplates, individually and in groups, must be matched to certain minimum percentages.

References:

- E. M. Darling, Jr. and R. D. Joseph, "An Experimental Investigation of Video Pattern Recognition", in Pictorial Pattern Recognition, by George Cheng, R. Ledley, D. Pollock, A. Rosenfeld, editors, Thompson Book Co., 1968.
- Hiroshi Genchi, Ken-Ichi Mori, Sadakazu Watanabe, Sumio Katsuragi, "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting", Proceedings of the IEEE, Vol. 56, No. 8, August, 1968.
- T. Sakai, M. Nagao and S. Fujibayashi, "Line Extraction and Pattern Detection in a Photograph", Pattern Recognition, Vol. 1, pp. 233-248, 1969.

Other References:

- E. C. Greanias, P. F. Meagher, R. J. Norman, and P. Essinger, "The Recognition of Handwritten Numerals by Contour Analysis", IBM Journal of Research and Development, Vol. 7, No. 1, Jan. 1963.
- John H. Munson, "Experiments in the Recognition of Hand-Printed Text: Part 1 - Character Recognition," Proceedings of the Fall Joint Computer Conference, 1968.

CONTEXT ANALYSIS

Context Analysis can be considered a higher level content extraction process, which takes the raw output of a pattern recognizing or character recognizing program and associates groups of individually recognized components, according to some regular (linguistic) characteristic in order to correct for previous errors in recognition, as well as generating larger order classifications.

A description of context analysis is given in Richard Duda and Peter Hart, "Experiments in the Recognition of Hand-Printed Text: Part II - Context Analysis," Proceedings of the FJCC, 1968. Individual hand-printed characters, as well as a list of possible alternative classifications for each character, are presented as input to the context-analysis phase. Associated with each character and its alternates is a number which measures the confidence that the presented letter is correct. The task of the analyzer is to achieve as low as possible an overall error rate by making appropriate choices from among the alternatives.

The techniques used in this report involve application of the syntax and semantics of FORTRAN to the input FORTRAN statements, and the application of clustering techniques to the appropriate identification of variable names. The context-directed analyzer is organized as a two-pass program. During the first pass each statement is identified by type and a table of identifier names assembled and clustered. During the second pass each statement is resolved and the final classification of the FORTRAN text is made.

The identification of statement type is facilitated by the appearance of a control word, such as DIMENSION, IF, etc., at the beginning of all statements except the arithmetic assignment statement. The procedure for identifying statement type is to compute the confidence of each FORTRAN control word from the leading segment of each statement. If the highest confidence completed exceeds a threshold, the statement is of the appropriate type, otherwise it is an arithmetic assignment.

A clustering algorithm groups together all segments in an identifier table having at least one common alternative for each character. If these common alternatives have sufficiently high average confidence, then all entries are replaced by a single list having only the common alternatives.

The final phase of the identifier analysis is concerned with using the identifier table to resolve ambiguities in text. Two steps are used: finding a candidate segment in an input statement and matching this segment against the identifier table.

Following identifier analysis, an expression analyzer is applied to each expression. The techniques of the expression analyzer are:

- (1) Identifier match with the table of identifiers,
- (2) compression operation to reduce combinatorial complexity,
- (3) procedure to increase local consistency,
- (4) partition of each expression into segments which might represent subexpressions,
- (5) resolution of the subexpressions,
- (6) reconstruction to uncompress the final answer.

These six procedures, with identifier analysis, constitute the context analysis for FORTRAN statements.

Application of context analysis reduced a sample program error rate from 9.3% to 2.4%.

References:

Richard Duda, et al, "Experiments in the Recognition of Hand-Printed Texts: Part II - Context Analysis", Proceedings of the Fall Joint Computer Conference, 1968.

4.0 BIBLIOGRAPHY

1. Robert H. Anderson, "Syntax-Directed Recognition of Handprinted Two-Dimensional Mathematics," Ph.D. thesis, Harvard University, January, 1968.
2. William D. Bond and Roger Wiegel, "Bubble Scan I - Part 1: Main", File No. 590 - DCL University of Illinois, Urbana, Illinois, February, 1964.
3. Richard T. Borovec, editor, "The Pax II Picture Processing System at the University of Illinois - Programming Manual," Report No. 314-DCL, University of Illinois, Urbana, Illinois, March, 1969.
4. G. C. Cheng, "Pictorial Pattern Recognition," Pattern Recognition, Vol. 1, pp. 187-194, 1969.
5. G. C. Cheng, editor, "Pictorial Pattern Recognition," Proceedings of the Symposium on Automatic Photointerpretation, 1968.
6. Richard O. Duda, et al, "Experiments in the Recognition of Hand-Printed Texts: Part II - Context Analysis," Proceedings of the Fall Joint Computer Conference, 1968.
7. Lawrence A. Dunn, et al, "Parametric Description of a Scan-Display System," Report No. 308 Digital Computer Lab, University of Illinois, Urbana, Illinois, February, 1969.
8. Hiroshi Genchi, et al, "Recognition of Handwritten Numerical Characters for Automatic Letter Sorting," Proceedings of the IEEE, Vol. 56, No. 8, August, 1968.
9. J. L. DiGiuseppi, "A Survey of Pictorial Data - Compression Techniques," Tech. Report No. 16 - University of Michigan, 1969.
10. Adolfo Guzman, "Decomposition of a Visual Scene into Three-Dimensional Bodies," Proceedings - Fall Joint Computer Conference, 1968.
11. Jerome Feder, "Linguistic Specifications and Analysis of Classes of Line Patterns," Tech. Report 403-2, New York University, Bronx, New York, 1969.
12. Claude Fennema and Claude Brice, "A Region-Oriented Data Structure," Technical Note No. 7 - Artificial Intelligence Group, Stanford Research Institute, 1969.
13. Kimio Ibuki, "The Taxicrinic Unit of Illiac III: A Tentative Design," Report 150 - Digital Computer Lab - University of Illinois, Urbana, Illinois, August 1963.
14. E. R. Ide, et al, "Some Conclusions on the Use of Adaptive Linear Decision Functions," Proceedings of the Fall Joint Computer Conference, p. 1117, 1968.

15. Robert S. Ledley, et al, "Pattern Recognition in the Biomedical Sciences," Proceedings of the Spring Joint Computer Conference, 1966.
16. Bruce H. McCormick and John C. Schwebel, "Consistent Formal Properties of Binary Relations", File No. 762 - Digital Computer Lab., University of Illinois, Urbana, Illinois, June, 1968.
17. McCormick, B. H., "The Illinois Pattern Recognition Computer - Illiac III," IEEE Trans. Electronic Computers, EC-12; No. 5, pp. 791-813, 1963.
18. Bruce H. McCormick and John C. Schwebel, "Consistent Properties of Composite Formation Under a Binary Relation", Report No. 348 - Digital Computer Lab, University of Illinois, Urbana, Illinois, July, 1968.
19. Bruce H. McCormick, "The Illinois Pattern Recognition Computer," Report No. 148, Digital Computer Lab - University of Illinois, Urbana, Illinois, August, 1963.
20. Bruce H. McCormick, Advances in the Development of Image Processing Hardware, University of California Press, 1968.
21. M. L. Mendelsohn, et al, "Computer-Oriented Analysis of Human Chromosomes", Cytogenetics 5: pp. 223-242, 1966.
22. M. L. Mendelsohn, et al, "Approaches to the Automation of Chromosome Analysis," "Conference on Biological Image Processing," UCLA Forum in Medical Sciences, No. 9.
23. M. L. Mendelsohn, et al, "Computer-Oriented Analysis of Human Chromosomes II." Annals of the New York Academy of Sciences, 1967.
24. M. L. Mendelsohn, et al, "Digital Transformation and Computer Analysis of Microscopic Images," (source unknown).
25. W. F. Miller and A. C. Shaw, "Linguistic Methods in Picture Processing - A Survey," Proceedings - Fall Joint Computer Conference, 1968.
26. John H. Munson, "Experiments in the Recognition of Hand-Printed Text: Part I - Character Recognition," Proceedings of the Fall Joint Computer Conference, 1968.
27. George Nagy, "State of the Art in Pattern Recognition," Proceedings of the IEEE, Vol. 56, No. 5, May 1968.
28. R. Narasimhan, "Bubble Scan I - Program," Report No. 167 - Digital Computer Lab, University of Illinois, Urbana, Illinois, 1964.
29. R. Narasimhan, James Fornango, "A Preprocessing Routine for Digitized Bubble - Chamber Pictures," File No. 558 - DCL - University of Illinois, Urbana, Illinois, July, 1963.
30. R. Narasimhan, "Labeling Schemata and Syntactic Description of Pictures," Information and Control, Vol. 7, 1964.

31. Alan V. Oppenheim, et al, "Nonlinear Filtering of Multiplied and Convolved Signals," Proceedings of the IEEE, Vol. 56, No. 8, August, 1968.
32. T. Pavlidis, "On the Syntactic Analysis of Figures," Proceedings of 1968 ACM National Conference.
33. William K. Pratt, et al, "Hadamard Transform Image Coding," Proceedings of the IEEE, Vol. 57, No. 1, January, 1969.
34. R. Kevin Rice, "Bubble Scan I - Part 2: Search", File No. 594, Digital Computer Lab, University of Illinois, Urbana, Illinois, April, 1964.
35. William S. Rohland, et al, "The Design of an OCR System for Reading Handwritten Numerals," Proceedings of the Fall Joint Computer Conference, 1968.
36. Azriel Rosenfeld, et al, "Sequential Operations in Digital Picture Processing," Journal of the ACM, Vol. 13, No. 4, October, 1966.
37. T. Sakai, "Line Extraction and Pattern Detection in a Photograph," Pattern Recognition, Vol. 1, pp. 233-248, 1969.
38. Carl M. Shinn Jr. and R. M. Kline, "A Computer-Directed System for Measuring Distance Between Edges in Optical Images," Tech. Report No. 10, Computer Systems Laboratory, Washington University, St. Louis, Missouri, October, 1968.
39. John C. Schwebel and Bruce H. McCormick, "Analysis of Simple Line Drawings," File No. 796 - DCL - University of Illinois, Urbana, Illinois, 1969.
40. John C. Schwebel, "Use of Graph Transformations to Characterize an Image: An Illustrative Example," File No. 770-DCL - University of Illinois, Urbana, Illinois, 1968.
41. Unger, S. H., "Pattern Detection and Recognition," Proc. IRE, Vol. 47, pp. 1737-1752, October, 1959.
42. Andries Van Dam and David Evans, "A Compact Data Structure for Storing, Retrieving and Manipulating Line Drawings," Proceedings Spring Joint Computer Conference, 1967.
43. Roger Wiegel, "Bubble Scan I - Part 3: List Memory Macros," File No. 585, Digital Computer Laboratory, Urbana, Illinois, 1964.
44. Yamada, J. P. Farnanago, "Experimental Results for Local Filtering of Digitized Pictures," Report No. 184 - DCL - University of Illinois, Urbana, Illinois, June 1965.
45. "Illiac 3 Programming Manual", Digital Computer Lab - University of Illinois, Urbana, Illinois.
46. E. M. Darling, Jr. and R. D. Joseph, "An Experimental Investigation of Video Pattern Recognition", in Pictorial Pattern Recognition, G. C. Cheng, et al, editors, Thompson Book Co., 1968.

47. T. G. Evans, "A Heuristic Program to Solve Geometric-Analogy Problems", AFIPS Conference Proceedings, Vol. 25, 1964, SJCC.
48. H. Freeman and L. Garder, "A Pictorial Jigsaw Puzzles: The Computer Solution of a Problem in Pattern Recognition," IEEE Transactions in Electronic Computers, EC-13, No. 2, April, 1964.
49. E. C. Greanias, P. F. Meagher, et al, "The Recognition of Handwritten Numerals by Contour Analysis", IBM Journal of Research and Development, Vol. 7, No. 1, June, 1963.
50. W. J. Hankley and J. T. Tou, "Automatic Fingerprint Interpretation and Classification Via Contextual Analysis and Topological Coding", in Pictorial Pattern Recognition, G. C. Cheng, et al, editors, Thompson Book Co., 1968.
51. Charles Heider, "Studies in Image Processing: A System Viewpoint", report for Computer Science 490, Part I of this report, University of Illinois, Sept. 1, 1969.
52. Bela Julesz, "Visual Pattern Discrimination", IEEE Transactions in Information Theory, IT-8, Feb. 1962.
53. Robert Ledley, Use of Computers in Biology and Medicine, McGraw-Hill, Inc., 1965.
54. Martin Levine, "Feature Extraction: A Survey", Proceedings of the IEEE, Vol. 57, No. 8, August, 1969.
55. D. L. Londe and R. F. Simmons, "Namer: A Pattern-Recognition System for Generating Sentences About Relations Between Line Drawings", Proceedings of the 20th National Conference, ACM, August, 1965.
56. W. F. Miller and A. C. Shaw, "A Picture Calculus", in D. Secrest and J. Nievergelt, editors, Emerging Concepts in Computer Graphics, W. A. Benjamin, Inc., N. Y., 1968.
57. J. L. Pfaltz, et al, "Local and Global Picture Processing by Computer", in Pictorial Pattern Recognition, G. C. Cheng, et al, editors, Thompson Book Co., 1968.
58. Otis Philbrick, "Shape Description with Medial Axis Transformation," in Pictorial Pattern Recognition, G. C. Cheng, et al, editors, Thompson Book Co., 1968.
59. R. M. Pickett, "The Perception of Random Visual Texture", Models for the Perception of Speech and Visual Form, Wather-Dunn, editor, MIT Press, Cambridge, Mass., 1967.
60. A. Rosenfeld, "Automatic Recognition of Basic Terrain Types from Aerial Photographs", Photogrammetric Engineering, v 28, March 1962.
61. A. Rosenfeld and Goldstein, "Optical Correlation for Terrain Type Discrimination; Photogrammetric Engineering, v 30, June 1964.

62. John Schwebel, "Toward the Specification of a New Image Processing Language", File No. 788, D.C.L., University of Illinois, Feb. 6, 1969.

63. Steiner and Haefner, "Tone Distortion for Automated Interpretation", Photogrammetric Engineering, v 31, March 1965.

64. Wather-Dunn, editor, Models for the Perception of Speech and Visual Form, MIT Press, Cambridge, Mass., 1967.