

COMPUTER ANALYSIS AND DESCRIPTION
OF POTTERY SHERD PATTERNS

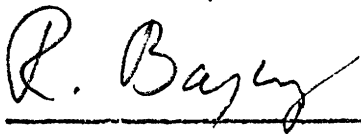
Sandra L. Scidmore

A THESIS IN

COMPUTER AND INFORMATION SCIENCE

Presented to the Faculty of the Moore School of Computer
and Information Science of the University of Pennsylvania
in partial fulfillment of the requirements for the degree
of Master of Science in Engineering.

1977



Supervisor of Thesis

Graduate Group Chairman

**TITLE OF THESIS: Computer Analysis and Description of
Pottery Sherd Patterns**

Abstract: This thesis describes an automated system for the analysis and description of patterns composed of straight line elements. It represents the second stage in the development of a system intended to aid archaeologists in tracing the evolution of decorative patterns found on pottery sherds.

The samples provided by the Anthropology Department for this project are pottery rims excavated from sites at Peñitas, Nayarit, Mexico. The input data are digitized pictures of photographs of the samples.

The first phase of the system is image processing. A set of low level operators is employed to obtain as final output a pictorial and vector description of all line segments in the pattern.

The second phase constructs a series of descriptions of the pattern, in which the successive steps reflect increasing levels of complexity in the interrelationships of the pattern elements. It first finds parallel and connected relationships between pairs of lines. The pairs are examined to obtain groups of equally-spaced parallel lines, and groups of lines connected at mutual endpoints. The groups are analyzed to yield the independent patterns

which compose the picture, and the patterns are named in their left-to-right order.

AUTHOR

SANDRA SCIDMORE



FACULTY SUPERVISOR

DR. RUZENA BAJCSY

TABLE OF CONTENTS

INTRODUCTION	1
CHAPTER 1 REVIEW OF LITERATURE	7
1.1 THE USE OF COMPUTERS IN ARCHAEOLOGY	7
1.2 PATTERN RECOGNITION	11
CHAPTER 2 THE SYSTEM	15
CHAPTER 3 SYSTEM DETAILS	19
3.1 PICTURE PROCESSING	19
3.2 THE DESCRIPTIVE PHASE	21
CHAPTER 4 IMPLEMENTATION	27
4.1 IMAGE PROCESSING	27
4.2 THE DESCRIPTIVE PHASE	49
CHAPTER 5 CONCLUSION	57
5.1 INPUT DATA	57
5.2 MORE COMPLEX PATTERNS	58
5.3 CLASSIFICATION OF PATTERNS	59
5.4 INFERRING WHOLE VESSELS FROM SHERDS	60
BIBLIOGRAPHY	61
APPENDIX: PICTORIAL AND STRUCTURAL DESCRIPTIONS OF THE SAMPLE SHERDS	64

INTRODUCTION

This project was motivated by work in the field of archaeology, specifically by the problems encountered by archaeologists in classifying decorative pattern samples so as to recover the chronological ordering of the sample pottery fragments.

The goal of the anthropological studies is to interpret the culture, evolution, and extent of ancient societies by tracing the design language of the society through time and space. It is believed that artifacts are the fossilized ideas of the culture which produced them, and that a proper study of archaeological data can infer or reconstruct the behavioral patterns of a particular culture (2,5).

It is felt that individual designs have meanings, and tell the story of a culture. For example, the pattern in Picture 1 (APPENDIX) is described as a staircase leading to a temple.

It is believed that evolution of design elements reflects the evolution of the society which created the designs. Each culture possesses mental templates or ideas as to the proper form its products should take. Deetz (5) cites the example of an anthropologist (Lila O'Neale) who investigated the idea of proper form, in studies of the

Yurok Indians in northern California. She showed photographs of baskets to the women of the tribe, and from comments on what was "wrong" with certain baskets, she was able to discover the limits of variation within the rules of proper basketry among those people. A functional attribute of an arrowhead or a decorative pattern on a pottery sherd are also expressions of these mental templates. Then, the "behavior" of a template through time and space reflects the behavior of the society which produced it. Wholesale transfers of templates occur in migrations; diffusion of "female" templates together with the isolation of "male" templates can be seen in societies in which the labor is strictly divided, and in which the women move to neighboring villages through marriage while the men remain in their home villages; circulations of templates occur through trade; and breakdowns of templates occur through conquest of a people.

(5)

Then, to reconstruct the order of events which the artifacts imply, the archaeologist must find the temporal order of the artifacts. To obtain perfect temporal ordering, the archaeologist would need an "ideal site"; that is, one in which the stratigraphic deposits resulting from occupation of the site are found superimposed in their chronological order, without reversal or mixing caused by natural or human disturbance. Under such ideal conditions, the

collection of artifacts would be representative and leave no gap in the record. Finally the archaeologist would introduce no errors that would lead to loss, or mixing of material that was contemporaneous. (1)

In reality, the archaeologist cannot know beforehand what the limits of the true occupation levels are. Moreover, excavation units impinge on neighboring occupation levels, and these boundaries cannot always be exactly determined. Natural or human disturbance to the site prior to the excavation introduce new variables (1). Thus the material cannot be recovered in perfect order. Under these conditions, the archaeologist must examine, compare, and classify each artifact in an attempt to create an order or pattern from the pieces of evidence.

Artifactual pottery offers certain advantages--it is numerous (in one study the number of vessels inferred from fragments, as compared to the numbers of other artifacts ranged from ratios of 6:1 to 15:1 (13)), and therefore offers a great deal of evidence; the design patterns are purely decorative, so do not stop evolving the way that functional attributes do when an upper limit of technological advance is reached. But the problems faced by the archaeologist in processing this information by hand are manifold. To achieve his or her goal, an archaeologist will visit many sites, each of which may yield thousands of sherds for

examination. The work then involves recording each sherd by hand; attempting to determine which fragments belong to the same pottery vessel; noting the differences and similarities among patterns, within a site and between sites; noting the frequency with which individual patterns occur; and finally hypothesizing the evolution of design elements and classifying the patterns accordingly. Thus the task becomes tedious and repetitive while at the same time requiring an intense concentration on details.

The advantages that a successful, automated system has to offer are many. The computer can quickly process the large number of patterns involved in such a study. A descriptive system such as the one described in this thesis could include an automated recording system which would, as it processes series of pictures, count presence/absence qualities of individual patterns, count the frequency with which certain individual patterns appear in combination, and count the presence/absence qualities of similarities between the patterns on the individual sherds. The data could then be quickly stored and retrieved for any additional analysis of the patterns.

The data thus obtained could be used to solve the problem of piecing together fragments of the same pottery vessel. If the calculations determined that the most likely combination was, for example, the stairsteps and temple in

Picture 1 (INDEX), then if the temple were found on the left edge of a sherd, the data could be scanned for a description of a picture which contained a stairstep on the rightmost edge. In the case of the fragmentation of a pattern within a single shard, the procedure could use a maximum-fitting procedure to "guess" a pattern and note the missing parts; then proceed to look for a description of a picture with a pattern that matches the profile of the missing parts.

Finally, and this is the area in which current experiments are being performed, the computer can be used to process the decision data in an attempt to classify the sherds. There are two problems which commonly arise in the present classification procedure. Classifications are largely intuitive, and therefore can vary among archaeologists who use different intuitive models. Current tests are being performed to test the reliability of various statistical methods as a solution to this problem; and the computer has been a useful tool in performing the many sets of calculations needed to compare these methods. The other problem is that the classification model may change during the course of recording the sherds; that is, the archaeologist will begin to see more variation in details of the samples. The size of the data base may not allow backtracking and reclassification, and consistency is difficult in the face of these increasing discriminatory powers (13).

Bordaz's (1) and McPherron's (2) methods in classification by computer implement iterative analyses of the data, in which further clustering of similar sherds or further collapsing of types is dependent on clustering results of the previous iteration.

It is hoped that the work presented in this thesis can contribute to the solution of some of the problems mentioned above. The visual processing stage addresses the problems encountered in processing and recording the large data base involved in such studies. The structural analysis of the design elements performed by the descriptive phase gives several levels of description of the pattern which can be used as a basis for further analyses and comparisons.

CHAPTER ONE

REVIEW OF LITERATURE

1.1 THE USE OF COMPUTERS IN ARCHAEOLOGY. As mentioned in the introduction, computers have been used as a tool to test the reliability of classification techniques. It appears, from the survey of literature, that visual processing and description has not previously been attempted. In the computer classification procedures which were reviewed, the sherds were analyzed for attributes determined to be important in the classification decisions, and data on the attributes was coded for input to the procedure.

Deetz (3,5) and Longacre (3) used computer classification procedures to put forward models as to how attributes of pottery design might be expected to reflect the patterned behavior of the potters. Deetz's samples were excavated from an Arikara Indian site in Northern Dakota. The site was clearly divided into three stratigraphic layers, so that the pottery could be divided into three components. After analyzing the pottery, it was discovered that the pottery from the earliest group could be placed in clearly defined typological groups, but that the last group practically defied classification according to the same attributes used in the earlier sorting. The intermediate group showed intermediate typological clarity. In order to gain a

detailed description of the "patterning" of the attributes of the sherds, a computer was used to perform analyses of the data. A detailed list of 174 attributes was first prepared, and the attributes formed the basis of a code which was used to describe each of the 2000 sherds on cards, one IBM card per sherd. Classes of attributes used in the code were lip profile (square, painted, etc.), lip decoration technique (tool-impressed, finger-impressed, etc.), lip design elements (12 basic designs); identical classes for the collar of the sherd; and sherd-neck angle.

The operation, which involved hundreds of thousands of individual computations, showed that the difficulty in creating typological categories in the later two groups stemmed from a progressive lowering of the degree of association of attributes in the samples. To associate the type of cultural history that would produce such a patterning, historical records on the Arikara Indians were checked. It was discovered that the progressive breakdowns in their family structure which occurred during the time period associated with the sherds would imply the dissolution of passing on instructions in the art of pottery-making from generation to generation.

Longacre's study examined space-trends rather than time-trends, and investigated the dispersion of motif attributes within a Pueblo settlement. His samples were

4,160 sherds. The sherds were coded for 175 motif attributes. He used the computer to calculate statistics as to measurements of interdependence between the variables--motifs, rooms, and locations. The results of the calculations showed that 60% of the 175 attributes occurred throughout the village, while the remaining 40% showed a bipolarization in which motifs showed a frequency either at the north end or the south end of the settlement. The results implied that two localized lineages occupied the site. Burial groupings in the adjacent cemetery, which contained the same three divisions of motifs, seemed to confirm the implications of the groupings of the motifs.

Other archaeologists who have taken advantage of the computer for handling large data bases of artifactual information include Binford and Binford, Sackett, Clarke, Whallen, and Gardin (3).

A comparison of automated classifications of artifacts and intuitive classifications by archaeologists was the object of a study by Hodson, Sneath, and Doran (3). The samples of the study were 100 Swiss brooches, whose chronological order had been determined by the stratification of the cemetery in which they were found. Controls for the study included the withholding of the chronological data. It was found that while the archaeologists, and an anatomist who served as a control, all differed, the intuitive analyses

bore some affinity to the results of the numerical analysis. It was felt that two of the statistical techniques used were similar to intuitive analytical archaeological procedures (3). Then, the techniques used in the automatic analysis compared favorably with traditional procedures, while offering the advantages of high-speed processing.

Clarke (3), and Doran and Hodson (6) are among the proponents of the use of the computer in testing out new statistical techniques for a comparative evaluation of traditional procedures. Experiments have been carried out by Bordaz (1) and McPherron (13), with favorable results.

Deetz suggests that computers be used to automate the printing of maps of the spatial and temporal distribution of any measurable archaeological information, thus taking advantages of the graphics aspect, as well as the numerical efficiency of the computer.

Finally, the computer was put to yet another use in helping with the reconstruction of the Temple of Akhenaten in Egypt. The temple had been completely dismantled after the reign of Akhenaten, and the only evidence of its original structure were the thousands of patterned bricks, or talatats, which were scattered over a wide geographic area. In trying to reconstruct the temple, the talatats had to be compared on the basis of similarity of content and the missing parts of the patterns. Each of the blocks was

numbered and photographed; and pattern details were coded and input to the computer. Most of the temple was able to be reconstructed with the aid of the computer analysis of the details (20).

1.2 PATTERN RECOGNITION. The approach that was taken in designing this system is that of syntactic or structural pattern recognition. Rather than use the classical decision-theoretic approach, in which all significant features are extracted and then compared to stored object descriptions (8,10), the pattern was analyzed for the primitives, and the geometric relationships which define the grammar of an individual pattern.

Proponents of the syntactic, or linguistic, approach include Grief (11), Joshi (12), and Tidhar (21). Grief feels that the method of storing patterns in terms of the relationships of their parts approximates the method which humans use in storing information, and that this hierarchical organization is an efficient way of accessing information.

Tidhar used a structural world model in the recognition of pattern shapes. The lattice-like structure of the model allowed recognition of objects on the basis of partial information because of the greatest lower bound property of the lattice structure. Joshi also cited the advantage of grammars, in the linguistic approach, which allow a factorization in the recognition procedure.

Fu and Rosenfeld (10) feel that the structural approach has advantages when the input patterns are complex and the number of pattern classes is very large. In this case a procedure would consider description and classification of patterns rather than classification only.

The structural approach is pertinent to the problem which motivated this work, that is, the study of the evolution of design elements according to pattern structure; and the hierarchical aspect of this approach allowed the examination of patterns at various levels of structural complexity.

The hierarchical methodology was put forward by Uhr (23) in describing procedures for flexible pattern recognition. He described a system in which descriptions would point to other descriptions, until a description which was adequate to classify the pattern was achieved. He suggests that the characterizers which imply other whole characterizers might also imply where these characterizers would be applied, and thus determine the sequence of processing.

Recent approaches in hierarchical organization were taken by Narasimhan and Reddy (15) in their recognition scheme for hand-printed English letters; and by Rajaskeran and Deekshatulu (16) in their system for the recognition of the Telegu alphabet.

Narasimhan and Reddy made the distinction between

syntax-directed analysis, in which the syntax rules are used only to drive an analyzer to parse the picture, and in which no external contextual information is made use of in the parsing process; and syntax-aided analysis, in which syntax rules are used as an aid in the analysis of the picture and in which the structured description need not be a complete parse of the picture. They used the latter approach. In this project, the input letters were divided into regions within a rectangular window which framed the letter. The existence of particular pattern primitives (vertical lines, D-curves, C-curves, etc.) in particular regions, together with the syntax rules, invoked appropriate branching through the decision tree, to determine the subset of primitive-region combinations to look for next. The second phase of the work involved the addition of a probability table of co-occurrences of letters, in addition to the syntax rules used in the first phase.

The second project involved recognition of the 2000 letters which make up the Telegu alphabet. The solution to the problem used the structural information that the individual characters are composed of two classes of shapes, "build-primitives" and "basic letters." The first pass used a sequential template matching procedure on sequences of directions which would parse the various letter sub-shapes, to recognize and remove the "build-primitives." The second

pass parsed the remaining class of "basic letters," again using the template matching process.

The differences in approach between these methods and the approach presented in this thesis is that, in this project, the recognition of individual patterns follows the expansions of structural analyses. This approach is meaningful in describing the structure of a design so that evolution or change to a design can be traced through the structure.

CHAPTER TWO

THE SYSTEM

The general approach of the recognition scheme consists of applying a series of image processors to find all straight line segments in the input picture, and then building a series of descriptions of the picture which reflect different levels of patterning among the pattern elements.

Descriptions of several of the levels are output so that different levels of this patterning can be illustrated.

The approach is that of filtering out details through successive stages of analysis while building more complex characterizations at each succeeding level. The construction of the characterizers proceeds from the properties of the individual line segments, to the pair-wise relationships between lines, to group relations among lines, and finally to the grammar which describes an individual picture. In this scheme, each level accesses the name of the output at the preceding level for decision details, and the names represent decision nodes that allow parallel-serial processing of separate pattern types.

Details of the image processing phase are left for following chapters, but the general organization of the second phase is outlined below. The system recognizes two

types of patterns: line patterns, which are described by line segments; and region, or solid-shaped, patterns whose outline is composed of line segments. It also uses the information that the arrangement of the patterns is in the form of a horizontal band, and that the band may be delineated at the top or bottom by horizontal zone markers.

The outline of the system is as follows:

I. FOR ALL LINE PATTERNS:

- 1) Get the LINES output by the visual stage.
- 2) Find all PAIR RELATIONSHIPS of the lines:
 - A) Find all CONNECTED PAIRS.
 - B) Find all PARALLEL PAIRS.
- 3) Find all HORIZONTAL ZONE MARKERS.
- 4) Find all GROUP RELATIONS:
 - A) Find all ZIGZAGS, or groups of lines connected at their endpoints.
 - B) Find all PARALLEL GROUPS, or groups of equi-distant parallel lines.
- 5) Find all GROUP SUBSETS:
 - A) Find all ZIGZAG SUBSETS.
 - B) Find all INDEPENDENT PARALLEL GROUPS.

II. FOR ALL REGION PATTERNS:

- 1) Find all CONNECTED PAIRS.
- 2) Find all ZIGZAGS.
- 3) Find all ZIGZAG SUBSETS.

III. FOR ALL PATTERNS:

- 1) Find the left-to-right grammar of the group subsets in the picture space.

The current structure allows generality in the early stages of description, since specific patterns are recognized only at the group-subset stage of analysis. While offering the advantages of structural analyses at several stages of the recognition process, this representation also has the capability of accepting new grammars at the world model level without any major restructuring of the basic procedure.

The organization suits the current data set since there are a small number of superficial classes--parallel groups and linearly-connected patterns--while there are numerous patterns within these classes ("corners," arches, triangles, squares, boxes, waveform zigzags, horizontal zigzags, staircase zigzags, vertical parallel groups, horizontal parallel groups, and staircase/diagonal parallel

groups).

In general, the system can be described by the diagram in figure 1.

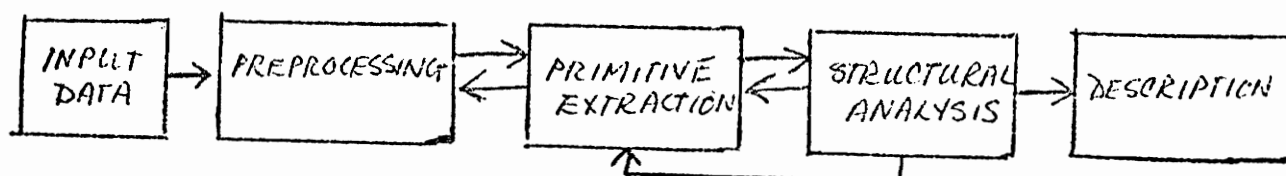


FIGURE 1. BLOCK DIAGRAM OF THE SYSTEM

CHAPTER THREE

SYSTEM DETAILS

This chapter provides an introduction to the data structures, descriptions, and methods of analysis used by the two stages of the system. The pictorial and structural descriptions of the input data which are described in detail in sections 3.1.1, 3.1.3, and 3.2.2-3.2.4 below are contained in the APPENDIX of this thesis.

3.1 PICTURE PROCESSING. The image processing stage receives as input the digitized picture of the potsherd to be analyzed. A series of low-level operators are employed to produce the series of pictures which are intermediate results; the final output is both a binary picture and a descriptive array of line segments which compose the potsherd pattern.

3.1.1 INPUT DATA. To obtain the input data, photographs were taken of the pottery sherds supplied by the Anthropology Department. The photographs were then digitized; that is, transformed to a matrix representation in which each matrix entry(i, j) represents the amount of brightness or reflected light at the corresponding "point" in the photograph.

The digitized pictures thus obtained contain a range of 64 gray levels, or levels of brightness. The two

extrema of the range, 0 and 63, represent the minimum and maximum levels of brightness, black and white. The pictures contain a magnified image of the sherd, so that the line patterns to be analyzed are approximately six matrix points wide.

3.1.2 ASSUMPTIONS ABOUT THE PATTERNS. The patterns contained on the sherds are composed from two types of pattern elements: line elements, which are 'thin' in one direction and 'thick', or 'long', in the other direction; and regions, or solid areas, which are 'thick' in both directions.

The patterns are engraved on the surface of the sherd and retain the light tan color of the clay used to make the pottery. The remaining surface area of the sherd is coated with either a red or gold/tan glaze, and presents a contrast in brightness to the tone of the clay.

The visual processing system uses this information - the expected shapes, and the expected contrast in brightness levels between pattern and background - for extracting the pattern from the background areas. The parameters defining 'thinness' and 'sufficient contrast' are input interactively, since the size and quality of the pictures vary widely.

3.1.3 PICTURES. After the initial step of extracting the pattern from the background from the digitized picture, the pattern is represented during several stages as a ternary picture. The three gray levels signify background, line

patterns, or region patterns. The last stage of analysis produces a binary picture, in which all line segments, both those which approximate line patterns and those which outline region patterns, have the same value.

3.1.4 DESCRIPTIVE ARRAYS. The last low-level processor produces a vector description of each line segment contained in the final picture. The array ALINES contains the descriptions of line-pattern lines; and array REGN contains region-pattern lines. In either array, each row of the array represents one line segment. Columns 1-4 contain (x_1, y_1) and (x_2, y_2) , the endpoints of the line. Column 5 contains the directionality of the line, and column 6 contains either the y-intercept of the line, if it is non-vertical, or the x-intercept, if it is vertical. The directionality, or angularity, of a line was used in the decision procedure of one of the low-level processors. It was chosen as a parameter with a view toward future extensions of this work, in which it could be used for the definition of a curved pattern in terms of its polar coordinates.

The slopes of the lines in ALINES and REGN are stored in the arrays LSLOPE and RSLOPE.

3.2 THE DESCRIPTIVE PHASE. The world model for this stage contains the information that the two relationships 'parallel' and 'connected' define the two superficial classes of patterns. While bottom-up processing is performed

in generating the details of these relationships, the top-down processing directs control to the focusing in on these two pattern types.

This phase is divided into the three steps which find pair descriptions, group descriptions, and subgroup descriptions. Each of these steps, including the descriptions and data structures used by these steps, is described below.

3.2.1 PAIR RELATIONS. The array ALINES is accessed to compare each line with all others in the array. Parallel, non-collinear, pairs are stored in array PARALL. Columns 1-2 contain the labels of the lines and column 3 contains the distance between the lines. All connected pairs are stored in array CNECT, in which columns 1-2 contain the labels of the lines, columns 3-4 contain (x,y), the point of intersection, and column 5 contains the angle of intersection.

3.2.2 GROUP RELATIONS. Array PARALL is accessed to find groups of equally-spaced parallel lines. The pairs which compose the group are removed from PARALL and are stored as adjacent entries in array PARGRP. An index array to PARGRP, IPAREL, contains information on each group: column 1 lists the number of pairs; column 2 lists the slope of the group, either 0, -1, +1, or 999; and column 3 lists the distance of separation. Array CNECT is accessed to find groups of lines connected at their mutual endpoints. The pairs are removed and stored as adjacent pairs in array ZIGZAG. INDZIG is

the index array to ZIGZAG; column 1 contains the number of pairs in the zigzag, and columns 2-3 access the top and bottom rows of ZIGZAG which contain the pattern.

3.2.3 GROUP SUBSETS. Array ZIGZAG is accessed to distinguish between the following subsets: 'L' patterns(, , , , 'V' patterns(, , arches(, triangles, squares, waveform zigzags(, , horizontal zigzags(, , and 'stairstep zigzags(, ). A more complex set of patterns, 'box'() patterns, is built from two of these patterns - arches, and squares.

The information needed to differentiate some of these patterns is the type of corners formed at the connection points, and the left-to-right order of these corners. Each corner in array ZIGZAG is analyzed, and a code number identifying the type of corner is stored in column 6 of each row of ZIGZAG.

Triangles and squares are the only closed shapes in the world model, so closure together with the number of pairs which compose the shape is sufficient to define these shapes. Triangles are described by array INDTRI; columns 1-2 reference the top and bottom rows of the rows of ZIGZAG that contain the triangle, in case future extensions of this work differentiate different types of triangles, and need to access the details of the formation. Squares are stored in array INDSQR; columns 1-2 again contain references

to rows in ZIGZAG. The minimum and maximum x-coordinates are stored in columns 3-4 to serve as information for the recognition of 'box patterns.

Arches are stored in array INARCH. Column 1 of INARCH references the top row of ZIGZAG which contains the pattern, and column 2 references the appropriate row of INDZIG. Columns 3-6 contain information which is needed to recognize the box pattern: columns 3-4 contain the minimum and maximum x-coordinates of the arch, and columns 5-6 contain the lines which are the sides of the arch.

Box patterns are stored in array INDBOX, where column 1 contains the number of concentric arches which compose the box; column 2 contains a code which identifies the presence of a square or isolated vertical line, or the absence of either; in the center of the box; columns 3-4 reference the two vertical isolated lines which flank the set of arches to the left and right. Column 5 contains the label of the isolated line in the center of the box, if one exists.

All connected lines, horizontal zone markers(section 3.2.4), and the isolated lines stored in array INDBOX, are removed from array PARGRP, to determine if there are any parallel groups which are independent, unconnected parallel pattern elements. All such groups are stored in array INDPAR. Columns 1-2 of INDPAR access the top and bottom rows of PARGRP which contain the group; column 3 contains the number of lines in the group; column 4 contains the slope of the

lines; and column 5 contains the slope of the group.

3.2.4 HORIZONTAL PATTERN BAND. The world model also contains the information that the arrangement of patterns on the sherd forms the shape of a horizontal band, and that this horizontal space might be delineated by HORIZONTAL ZONE MARKERS at the top or at the bottom of the space. An array PATTERN references each individual pattern (a box, a triangle) in the picture so that the patterns can be accessed and named in their left-to-right order along the band. Columns 1-2 contain the minimum and maximum x-coordinates of the space which the pattern occupies; column 3 contains the type of pattern (this allows branching in the printout of the description); and column 4 contains additional information about the type of pattern (again for branching and naming)- either direct information, via a code number, or via a pointer to a descriptive array.

Array ZONE stores the labels of the HORIZONTAL ZONE MARKERS, so that the lines may be removed, as described previously, from array PARGRP. HORIZONTAL ZONE MARKERS need to satisfy the condition that they 'sandwich' the rest of the pattern; that is, no pattern line, other than another zone marker can lie between it and the relevant top or bottom of the picture. While the lines need not cover the entire width of the picture, because of the fragmentary nature of the original sherd, collinear lines in the same

horizontal space are not allowed.

CHAPTER FOUR

IMPLEMENTATION

The system was coded in FORTRAN IV so that it would be compatible with routines in the Moore School Vision library. It was implemented on the Univac Spectra Series 90/70 at the Moore School Computing Facility.

The visual processing routines and description procedures are described in detail below.

4.1 IMAGE PROCESSING. The processing system employs a series of low level operators to first extract pattern areas from the original picture; to then sequentially transform the data by smoothing the pattern, and eliminating false pattern candidates; and to finally represent the pattern in terms of the line segments which compose the pattern.

The low level operators thus employed are, in order: 1) THRES, which creates a thresholded version of the input picture; 2) SEARCH, which differentiates between line patterns and region patterns; 3) THINN, which thins pattern areas and smooths pattern edges; 4) SKELTN, which skeletonizes line patterns; 5) REGION, which outlines region patterns; and 6) LINE, which combines the broken segments output by 4) and 5) to produce continuous line segments.

Problems in the quality of the digitized pictures due to high reflection of light off some of the sherds and

the low contrast between the tone of the gold glaze and tone of the tan clay caused the original digitized picture to be processed in one of two ways. If the range of gray levels throughout the picture is relatively homogeneous, or there exists a marked contrast between the pattern and background areas, then program THRES is used to gauge the range of pattern levels and contrasts; and program SEARCH is used to extract the two types of patterns from the background. If the range of gray levels varies dramatically throughout the picture and the contrast between pattern and background does not exceed the other contrasts in the picture, program THRES is used to gauge the pattern levels and to then extract the pattern from the background; and SEARCH is used to differentiate the types of the extracted patterns. In either case, the succeeding steps are the same: low level operators numbered 3)-6) are applied, in order, as needed. This sequence of control is diagrammed in figure 3.

4.1.1 THRES. Program THRES is modeled after the standard threshold operation.(17), but contains the option of examining rectangular subsets, or windows, of the input picture independently.

The standard operation is as follows: Given an input picture PIC1, and input values MIN,MAX,HIGH,LOW; produce an output picture PIC2 where

$$\begin{aligned} \text{PIC2(I,J)} &= \text{HIGH} \text{ if } \text{PIC1(I,J)} \geq \text{MAX} \text{ and} \\ &\text{PIC2(I,J)} = \text{LOW} \text{ if } \text{PIC1(I,J)} \leq \text{MIN}. \end{aligned}$$

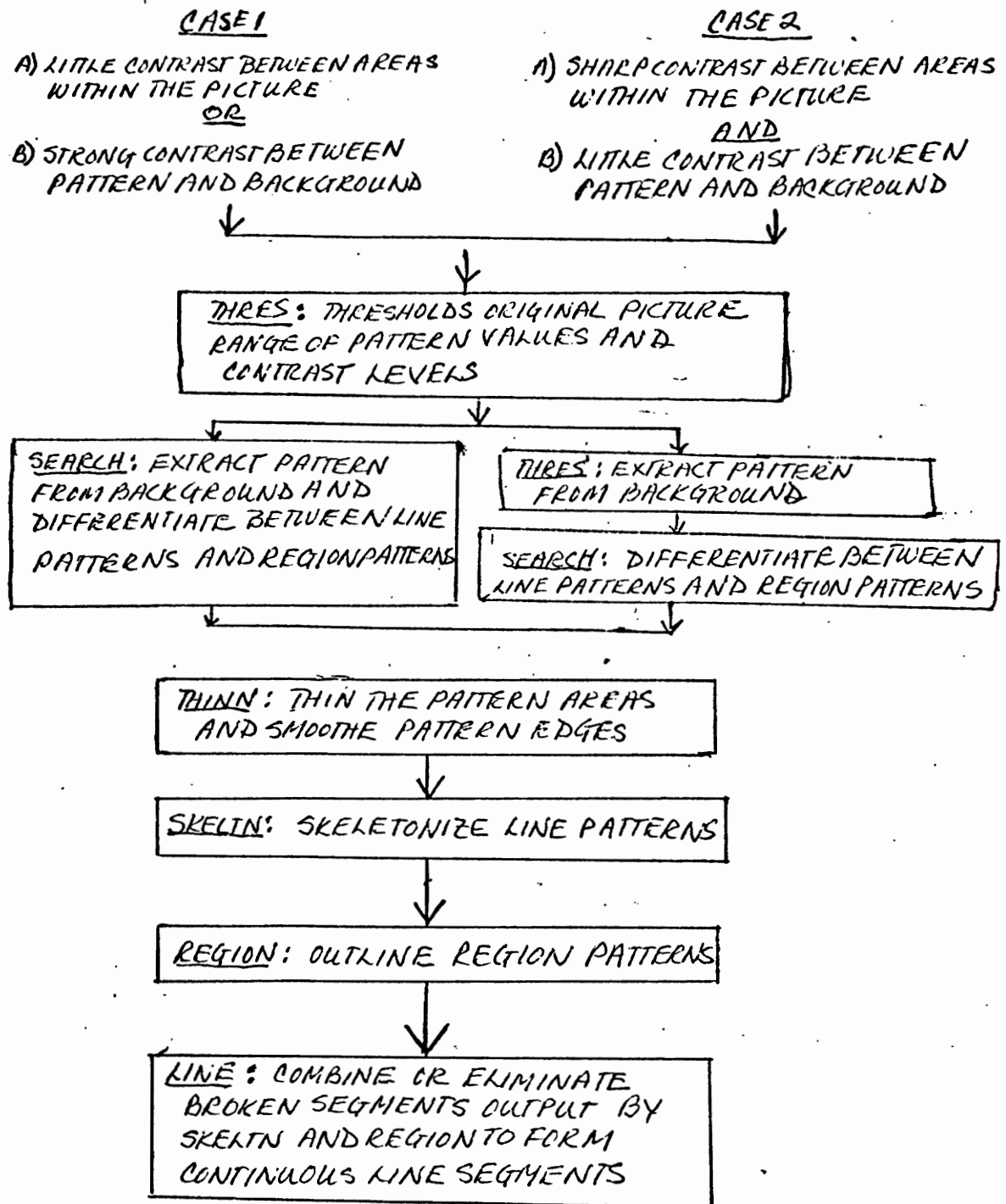


FIGURE 3. PICTURE PROCESSING SEQUENCE

The standard procedure can operate on the digitized picture to produce a binary picture, in which the pattern has value MAX(or MIN) and the background has value MIN(or MAX), if the gray levels are constant throughout the picture. In general, though, the gray levels in the original picture are not constant. The curvature of the pottery and the shiny nature of the glaze cause uneven reflection of light from the surface of the shard. In this case, the operation is used to gauge the range of gray levels which compose the pattern, and the contrast levels between the pattern values and background values.

If it is found that the contrasts in areas of different illumination are as strong as the contrasts between pattern and background, then the option of processing windows separately can be used to extract the pattern. The program will treat each area of homogeneous gray levels as a separate picture with its own input values of HIGH, LOW, MIN, and MAX. To define a rectangular window, the coordinates of the upper-left corner, (x1,y1), and those of the lower-right corner (x2,y2) are input to the program.

4.1.2 SEARCH. The input to program SEARCH is either the original digitized picture, or the thresholded version of this picture, both of which contain a dark(or light) pattern against a light(or dark) background.

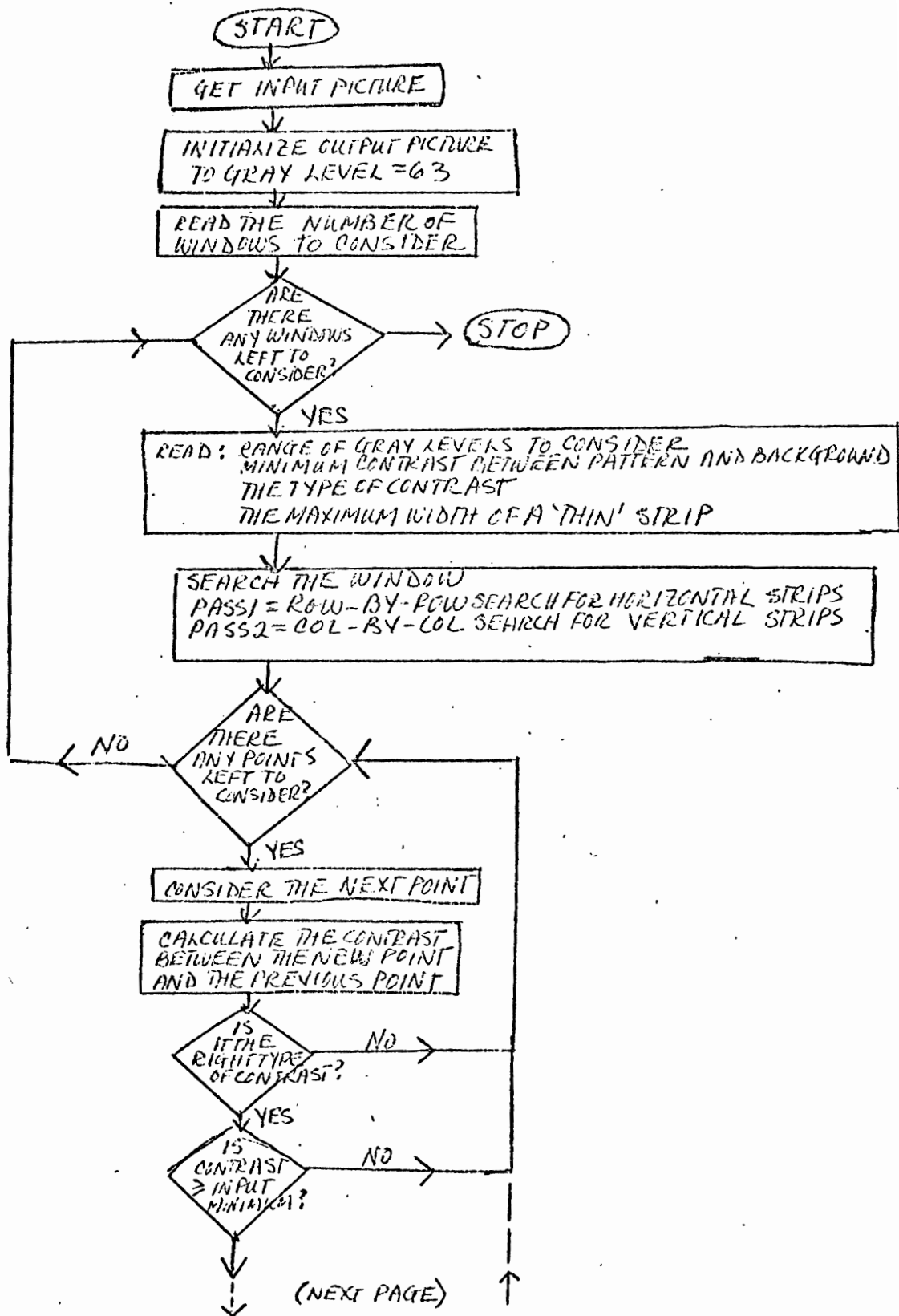


FIGURE 4. FLOWCHART FOR PROGRAM SEARCH

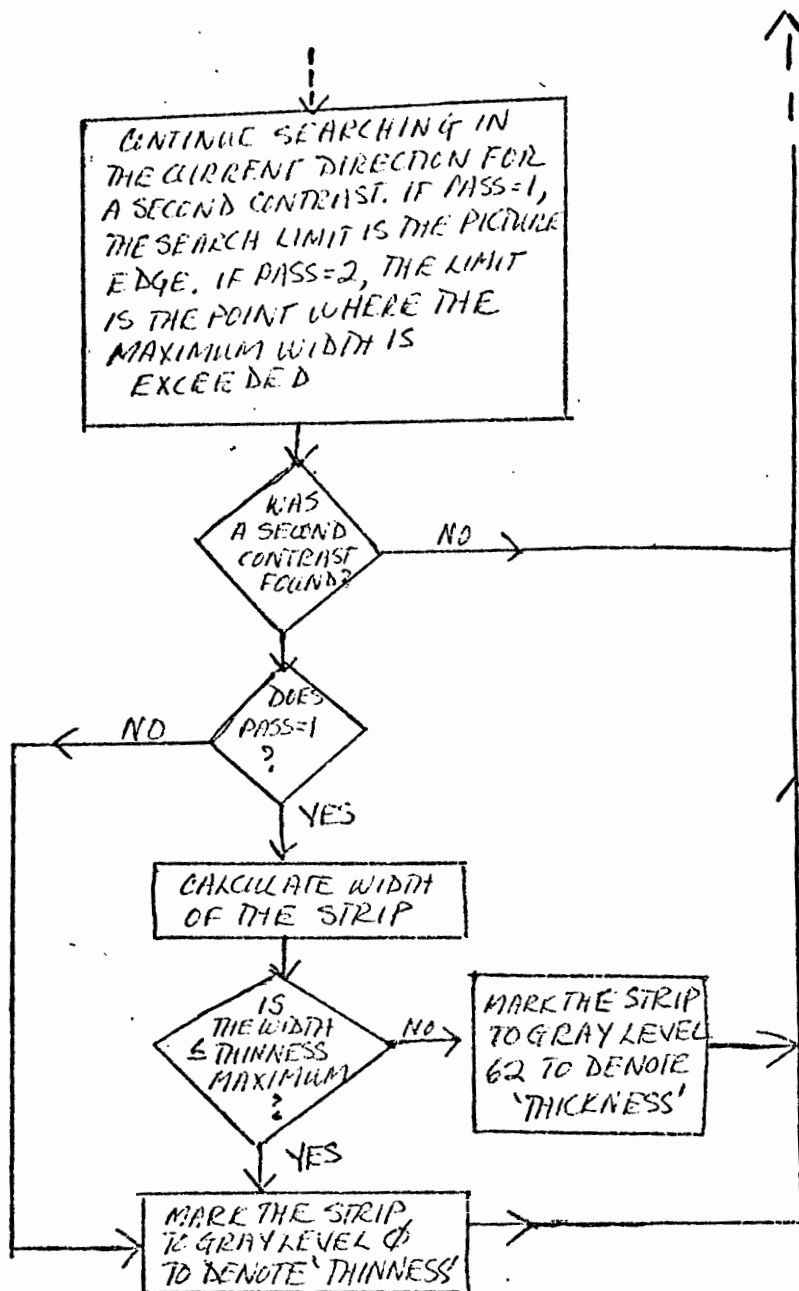


FIGURE 4. continued

The program looks for high-low-high(or low-high-low) contrasts in gray levels as evidence of background-pattern-background transitions (4,21), and when it finds such a transition, marks the potential pattern section, or strip, on the output picture.

The program differentiates between line patterns and regions, and marks line patterns with gray level 0 and region patterns with value 62.

The program uses the following input data for the pattern detection: 1) the upper and lower bounds of the range of gray levels which compose the pattern; 2) the type of contrast to look for - either high-low-high or low-high-low; 3) the minimum contrast which signals a pattern-background or background-pattern transition; and 4) the maximum allowable width for a detected strip to be considered thin. The data is input interactively for each picture, since the size and quality of the pictures vary dramatically.

The distinction between the two types of patterns is accomplished through a two-pass search routine. In the first pass, the program performs a left-to-right, row-by-row, search of the input picture, and detects horizontal pattern strips. If a detected strip is shorter than the value defining 'thinness', then the strip is viewed as a cross-sectional width-slice of a vertical or diagonal line pattern, whose width is 'thin'. If the detected strip is longer than

the input maximum, then the strip is seen as either a length-slice of a horizontal line or a cross-sectional slice of a region pattern. 'Thin' strips are marked 0, and thick strips are marked 62.

The second pass performs a top-to-bottom, column-by-column search of the input picture, and detects vertical strips. This pass will only look for and mark 'thin' strips; that is, when a background-pattern transition is detected, the program will search for a second contrast only until the maximum width for 'thinness' is exceeded. In this manner, regions, which were marked 62 by the first pass, will retain this value, since regions are thick in both directions; while horizontal line patterns, which were considered 'thick' by pass 1, are now marked 0 because their 'thin' width-slices are detected and marked by pass 2.

Program search also retains the option of considering windows of the input picture as separate data structures. In pictures which contain both large and small patterns, the widths of the pattern lines which compose the large patterns appear to be 'thick' in comparison to the widths of the pattern lines that compose the smaller pattern; and they may actually be thicker than the cross-sectional width of a region which appears in the same picture.

4.1.3 THINN. Program THINN smooths pattern edges and thins pattern areas. The routine tests the eight-point neighborhood

of each pattern point in the input picture, and if the connectivity is greater than an input threshold variable, then the point is marked on the output picture.(7) The thinning operation also eliminates isolated points or small isolated clusters of points, and thus reduces 'noise' or areas which appear in the picture due to scratches or areas of high illumination on the surface of the pottery vessel.

The pattern edges on the original shard are irregular due to the porous nature of the pottery, the manner in which the patterns were engraved onto the surface, and most importantly because of the degradation of the sherd over time. These irregularities were magnified in creating the digitized picture, and are quite pronounced in some cases.

If the irregularities are minor, a liberal input variable will cause a minor smoothing of the edges. A strict input variable will cause a smoothing and thinning process.

The program may consider the input picture in terms of window subsets, for reasons similar to those mentioned in the description of the threshold operation. One sherd may contain both large patterns composed of boldly-engraved line segments and small patterns composed of delicate line segments. Thus a strict thinning procedure which would smooth the edges of the bold patterns would degrade the smaller

lines, even causing some of them to disappear.

4.1.4 SKELTN. Program SKELTN operates on line patterns, and represents the first stage in reducing these patterns to line segments which can be analyzed in terms of slopes and other geometric properties.

The program performs two passes of the input picture in search of line patterns, and recognizes such a pattern as a series of connected cross-sectional width-slices (the strips which were output by program SEARCH).

The program first looks for two such connected strips as evidence of the existence of a line pattern. If only one isolated strip is encountered, it is considered a false candidate, and is not analyzed by the program. The program skeletonizes the first two strips by marking the midpoints of the strips on the output picture, and continues in the direction of connection for as long as other strips are detected. The model used by the routine is the information that a line pattern has a consistent width along its length. Thus the routine will stop searching if it finds a strip whose length varies from that of the previously-detected strip by more than three points, on the assumption that a junction with another line was detected.

Continuous line patterns will be disconnected if its width varies sharply at some point, because of this model; and also at points of sharp angularity along its length.

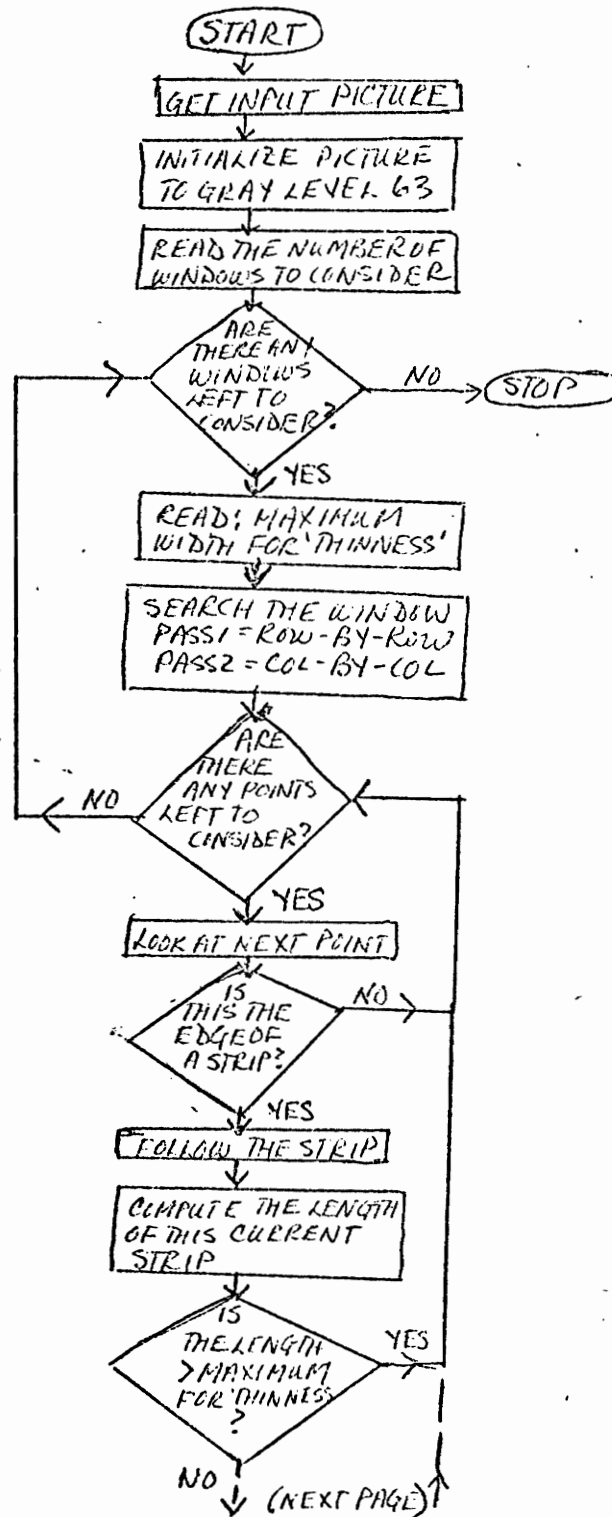


FIGURE 5. FLOWCHART FOR PROGRAM SKELTN

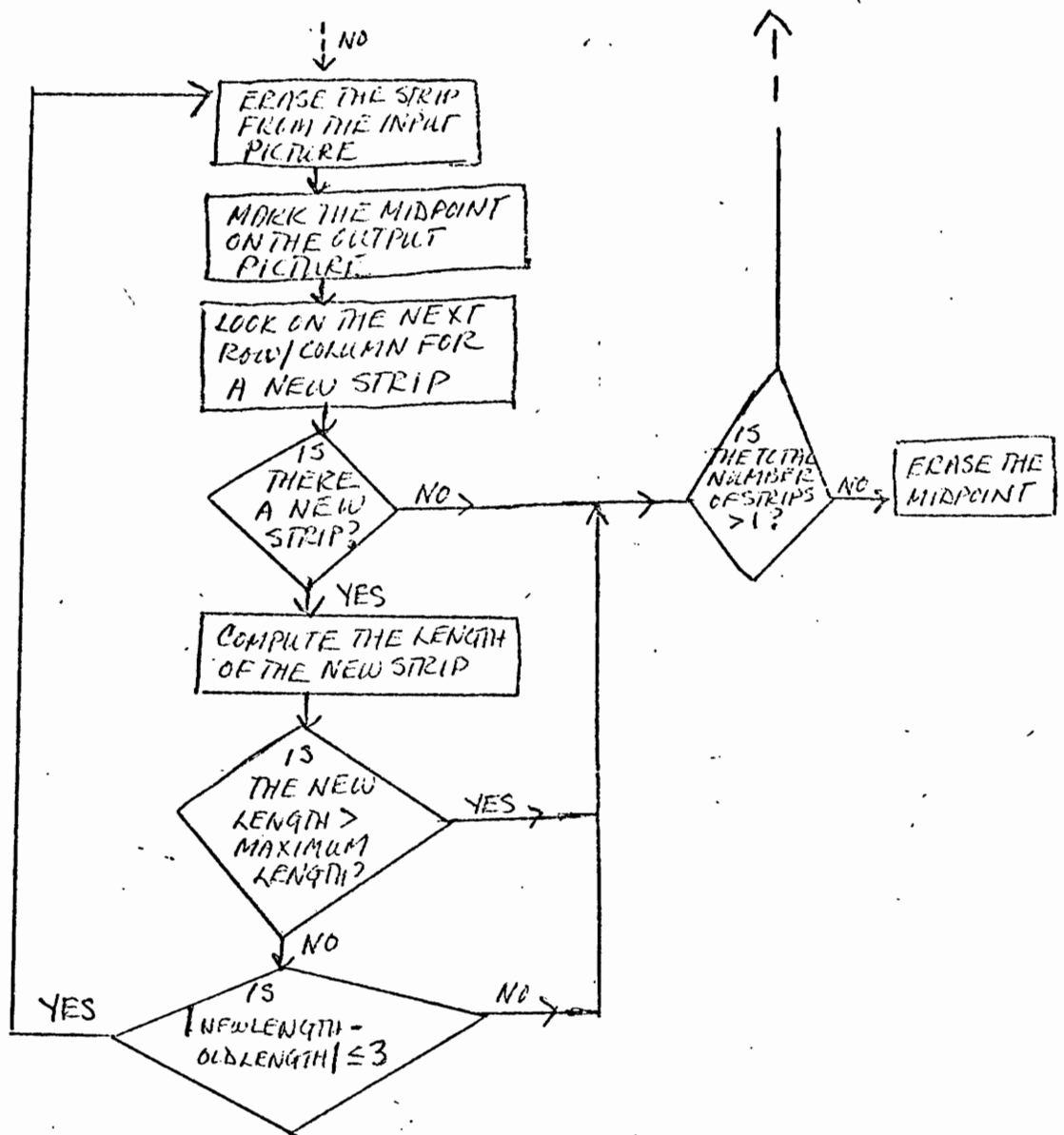


FIGURE 5. CONTINUED

Spurious pattern areas will be further minimized by this procedure since isolated points or lines are not marked on the output picture, and because thin shapes which do not match the model are disconnected.

The program again differs from the standard operation (14), in the ability to treat the input picture, if necessary, as more than one data set. The general structure of program SKELTN is diagrammed in figure 5.

4.1.5 REGION. The region processor operates on solid-shaped patterns, and outlines these patterns by marking the boundary points on the picture. It searches the picture, or window of the picture, until it finds a transition in adjacent points from gray level 63 to gray level 62, and marks the boundary point on the picture. It follows points in the current search direction until a 62-63 transition is found, and again marks the boundary point on the picture. These points have value 61, the code used by program LINE to recognize region points.

The regions in the picture are solid, with no holes, and have boundaries which are straight lines. The output line segments have characteristics similar to those of the segments output by the skeletonization procedure.

The flowchart for program REGION is contained in Figure 6.

4.1.6 LINE. This procedure operates on the broken segments output by SKELTN and REGION to obtain the continuous lines

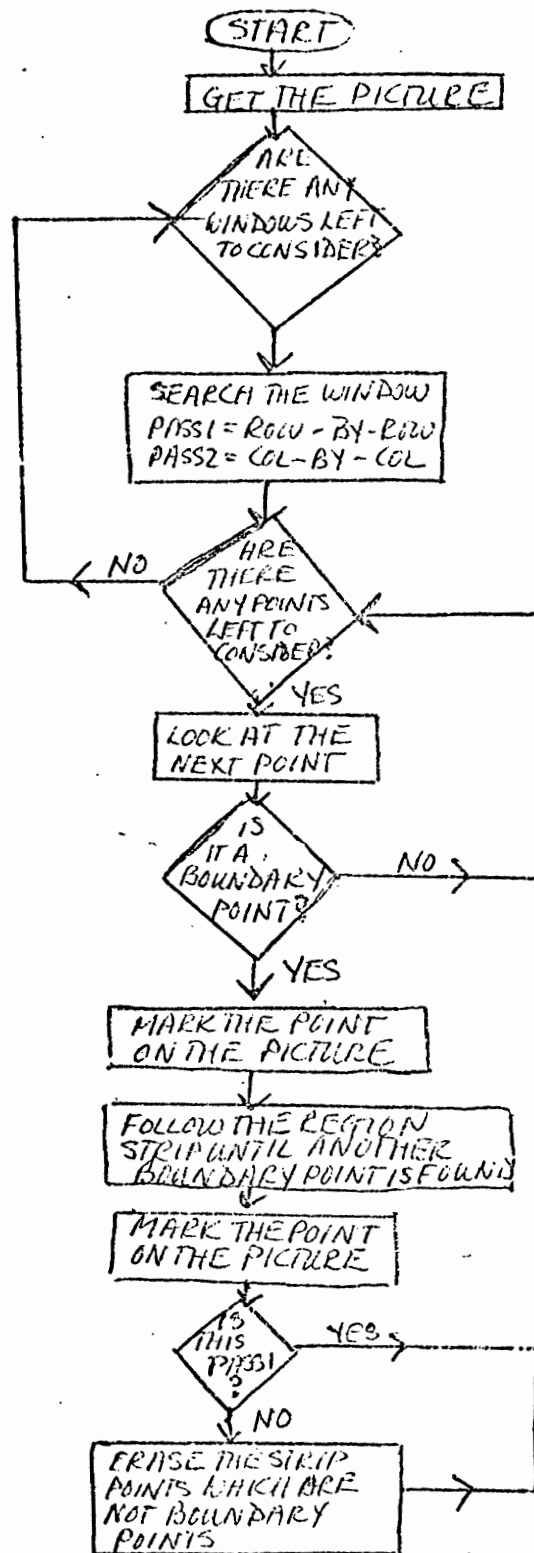


FIGURE 6. FLOWCHART FOR PROGRAM REGION

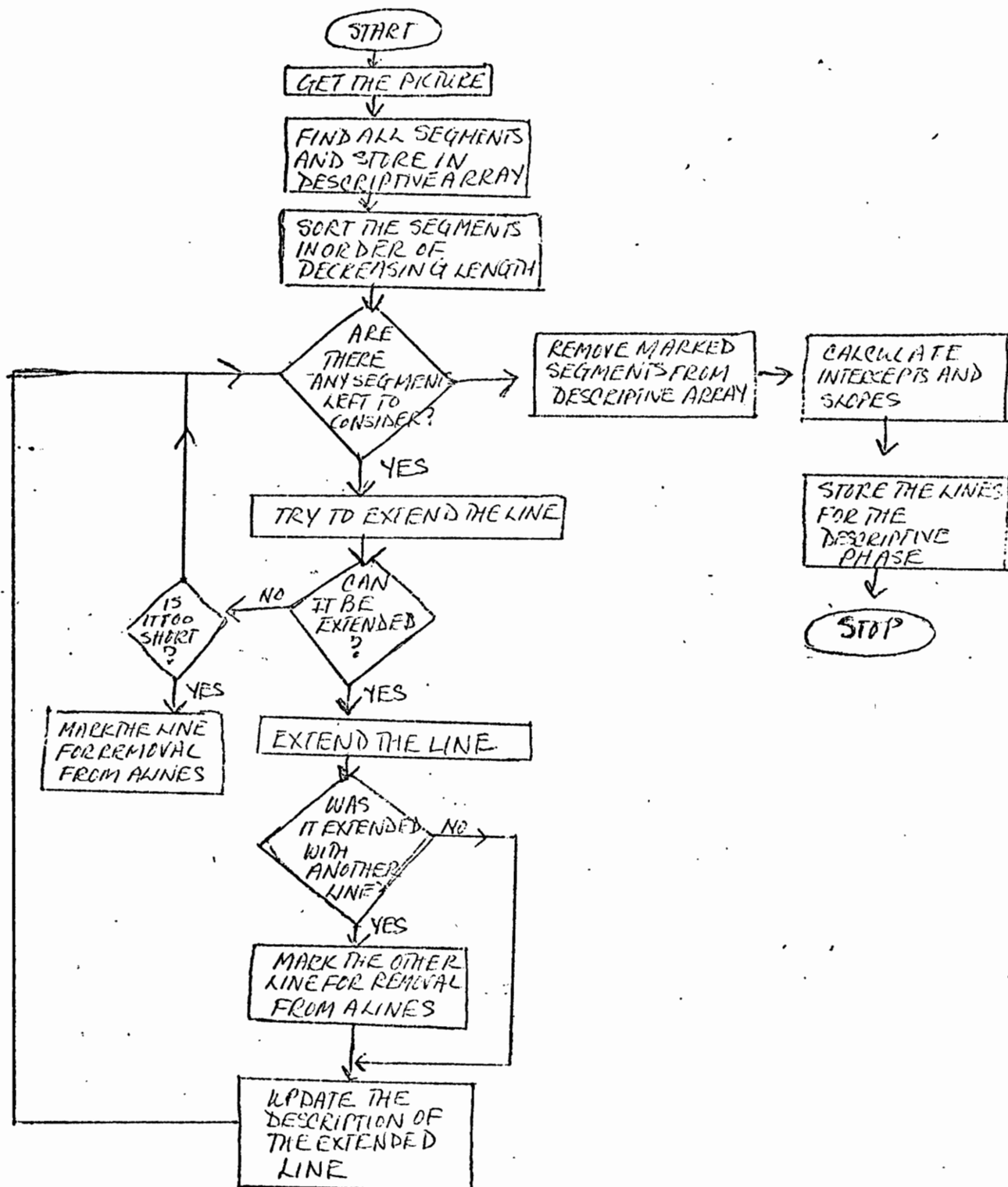
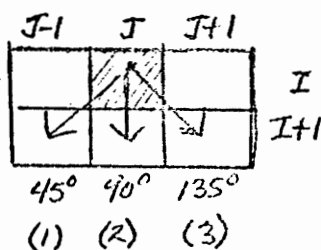


FIGURE 7. GENERAL FLOWCHART FOR PROGRAM LINE

which were disconnected by previous processing.(17,19,20) The first step, accomplished by subroutine SEGMNT, is to find all straight line segments in the input picture. The second step, accomplished by subroutine EXTEND, forms continuous lines by trying to extend each of the segments in each direction from its endpoints. The general flowchart for LINE, and the flowcharts for SEGMNT and EXTEND are illustrated in Figures 7,9, and 11.

1) SEGMNT. The procedure recognizes a straight line segment as a path of connected points which is acceptably long, and maintains a similar directionality along its length. As it traces a path of points, the routine notes the directionality of the newest point with respect to the location of the last-found point. The procedure is a two-raster search and recognizes two sets of directionalities. These sets, with respect to the direction of the search, are illustrated below:

PASS1-TRACES POINTS
ROW-BY-ROW



PASS2-TRACES POINTS
COLUMN-BY-COLUMN

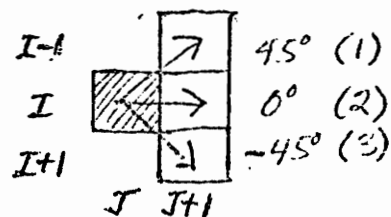


FIGURE 8.DIRECTIONALITIES USED BY SUBROUTINE SEGMNT.

The directionality of a whole line segment is the average of the directionalities along the segment. In tracing a path, the local average directionality is computed at every third point along the path. It is compared to the previous overall average directionality of the segment; if the directionalities are similar, the tracing continues. If at any point along the path, the local average differs from the overall average directionality of the path by 45° or more, processing of the path is ended, and the new points are not considered to be part of the segment.

During the tracing process, the points along the same segment are marked with the same(unique) value, so that each segment can be uniquely identified during the extension process.

At the end of the processing a segment, it is stored in one of the descriptive arrays, ALINES or REGN, if its length is at least as great as the minimum acceptable length. If the segment is too short, its points are retraced and marked with the value which indicates 'noise' or isolated points.

2) EXTEND. This routine operates on a priority system, and tries to extend each segment output by SEGMNT in order of decreasing length. It is assumed that long segments are more likely to represent the true directionality of the original line, and less likely to represent irregularities

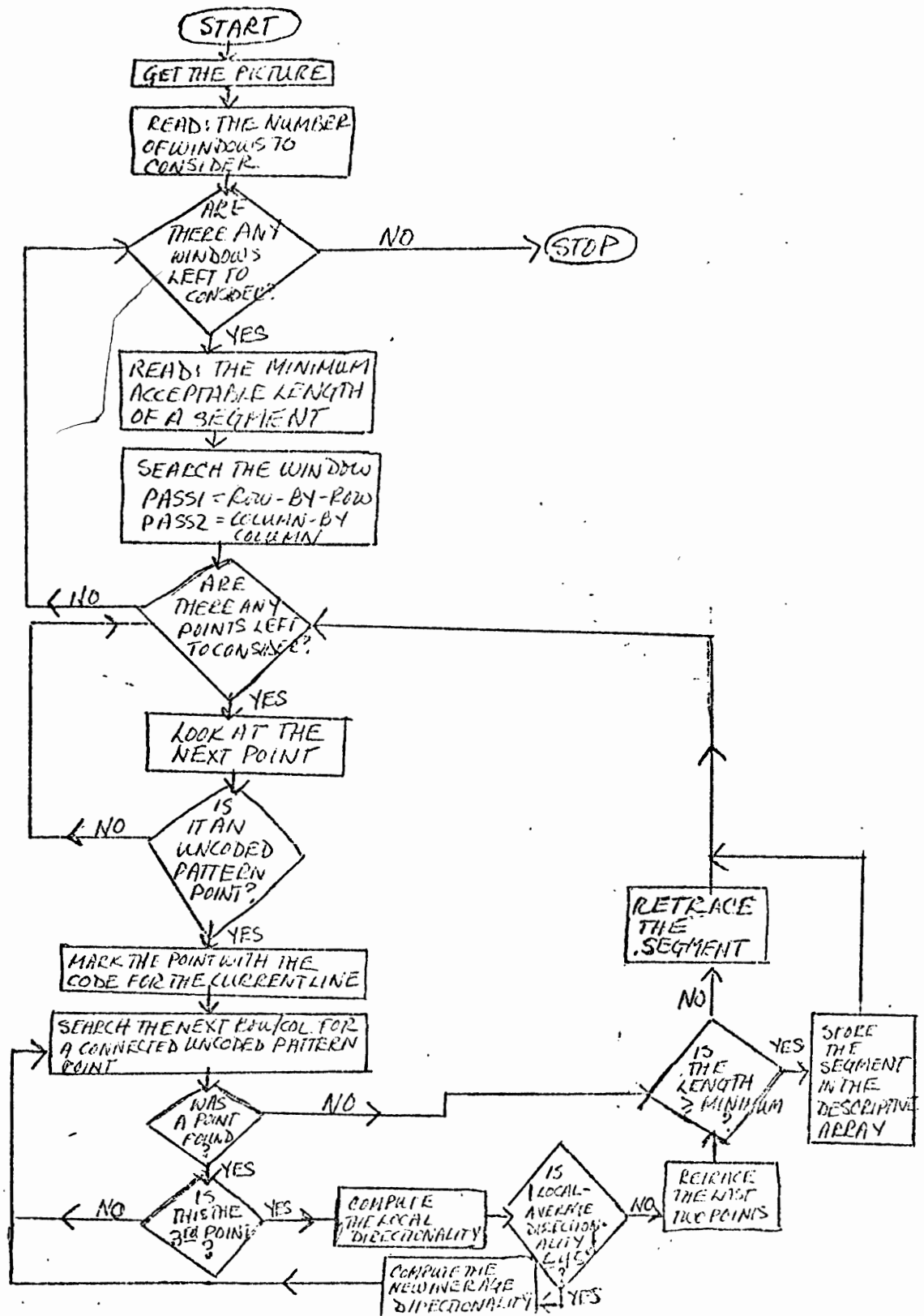


FIGURE 9. FLOWCHART FOR SUBROUTINE SEGMENT

in the width or slope of the line. It receives as input the array of line segments which were ordered in terms of decreasing length by a shell sort procedure.

The procedure tries to extend lines by expanding a series of lookahead grids, from each endpoint, in the direction of the length of the line. Two types of grids are expanded, one for horizontal and vertical lines, and one for diagonal lines. The dimensions of the grid are input interactively, since the length of pattern lines varies greatly from picture to picture. Examples of the grids are illustrated below.

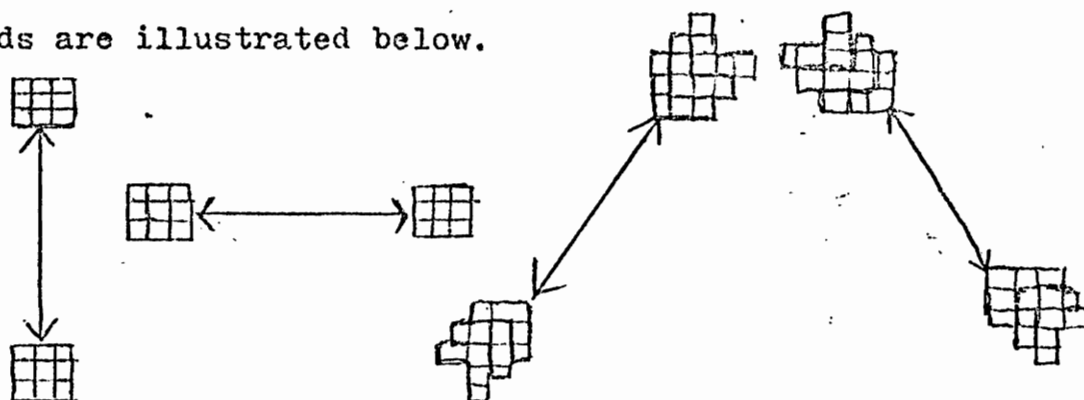
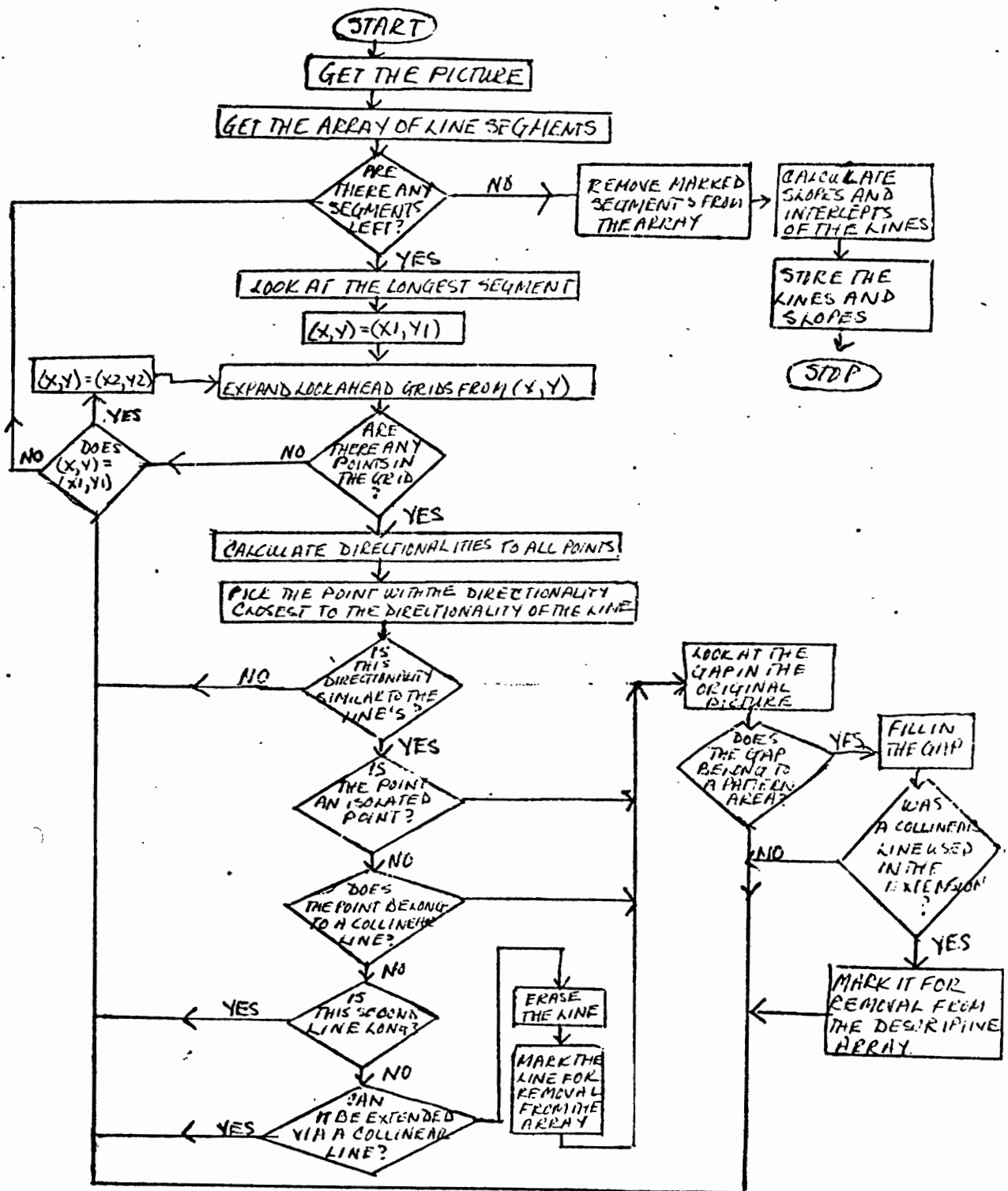


FIGURE 10. GRIDS OF DIMENSION 3 FOR THE FOUR TYPES OF LINES RECOGNIZED BY THE DESCRIPTIVE PHASE.

The number of grids which can be expanded successively in the search direction is input interactively for the same reason as are the grid dimensions. Each time a grid is expanded, the grid points are tested for the existence of pattern points. (The existence of such points in the current grid will stop further expansions of grids.) The directionalities from the endpoint in question to all of the pattern point entries are calculated. The



entry which yields the directionality closest to that of the line in question is picked as the candidate for the extension process. If the directionality to the candidate point is not similar to that of the line, the current extension is halted, and the other endpoint of the line or the next longest line is considered for the next extension.

In general, the procedure tries to extend the line by filling in the gap, or tracing the shortest path, between the endpoint in question and the candidate point. The original digitized picture is referenced for the gray levels of the points along the gap. These values should indicate whether the gap belongs to a pattern line which became disconnected during previous processing (and therefore can be reconnected) or whether it is a background area.

However, before the gap is even tested, certain constraints are placed on the extension procedure. If the candidate point is an isolated point or is a member of a line which is collinear to the line in question, then the gap is automatically tested. If the candidate belongs to a line whose directionality is not similar to that of the current line, then further tests are made. If the line is short, and cannot be extended via a collinear line, it is erased and used in the current extension. It is assumed that such a line is due to a scratch on the original sherd

or due to an irregularity in the original line pattern. But if this second line is long or can be extended, then the extension process of the current line is halted. In the actual gap-filling process, the procedure calculates the slope of the straight line which would join the two points. It approximates this line by traversing the rows, or columns, between the two points and marking the best entry on each row, or column. If the difference in the row-coordinates between the two points is greater than the difference in column-coordinates, then the procedure traverses the rows between the points; otherwise it traverses the columns. It calculates the best entry on each row or column by using the formulas $x_1 = \text{slope} * (y_1 - y) + x$ or $y_1 = \text{slope} * (x_1 - x) + y$.

After each successful extension of a line, the descriptive representation of the line is updated. The new directionality, length, and endpoints are recorded. All segments that were combined with longer collinear segments are marked for removal from the array. For each unsuccessful extension, or erasure, of a short segment, the descriptive entry is also removed.

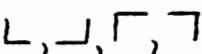
At the end of all extensions of all segments, the descriptive array is shortened by removing all segments which were marked for this purpose. The y- or x-intercepts of the remaining lines are calculated, and replace the

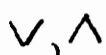
length as a descriptive parameter. At this point the vector descriptions should include all the lines in the picture; the array of segments is stored for use by the descriptive phase.

4.2 THE DESCRIPTIVE PHASE. The early stages of the descriptive phase, including the procedures which find parallel pairs and connected pairs, and parallel groups and zigzags, are not described in detail here. They were written by Judy Epstein and are described in detail in her thesis(9); they were outlined in earlier chapters for the sake of continuity.

This section then describes the procedure PASS4, which takes as input the arrays PARGRP and ZIGZAG, and all of the arrays which are accessed by them, and calls 1) subroutine SUBZIG to find all zigzag subpatterns, 2) subroutine PARAL to find all independent parallel groups, 4) subroutine REGPAT to find all region patterns, and 5) subroutine ORDER to order the individual patterns in left-to-right order.

4.2.1 SUBZIG. This subroutine analyzes all the zigzag patterns and differentiates the following zigzag subsets:

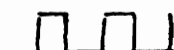
a) 'L' subpatterns - 

b) 'V' subpatterns - 

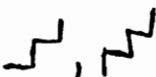
c) arches - 

d) triangles

e) squares

f) waveform zigzags - 

g) horizontal zigzags - 

h) stairstep zigzags - 

The subroutine first calls subroutine CORNER to identify all the types of corners formed by all the connected pairs in array ZIGZAG, and to store a code number in column 6 of ZIGZAG that identifies the corner. The types of corners which are recognized are listed below by their code numbers:

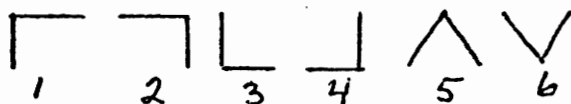


FIGURE 12. TYPES OF CORNERS AND THEIR CODE NUMBERS

The types of corners are determined by knowing the order in which the lines were traced by subroutine SEGMNT, and from this, knowing which columns of ALINES or REGN contain the leftmost endpoint. (For all lines other than diagonal lines of approximately 45° , the leftmost endpoint was the first-detected endpoint, and is stored in the first two columns of the vector description of the line.) To find 'L' corners, the procedure must find a horizontal line and note which endpoint is the point of connection, and find a vertical line and note its endpoint that is the connected endpoint. Type 5 corner is a pair of diagonal lines whose intersection is formed by the first-found endpoints of the lines; Type 6 corner is formed by the intersection of the second-found endpoints of the lines.

The specification of the corners is needed to recognize arches and waveform zigzags, but is calculated for all zigzags in case future extensions of the system

require further analyses of the subsets.

After the corners are coded, subroutine SUBSET differentiates between the various subsets. The first step in the process is to determine whether the zigzag is a closed shape or an open shape. The subroutine accesses the index array to ZIGZAG, to access the adjacent rows of ZIGZAG which contain the current pattern's pairs. If each line which composes the pattern appears twice in the pairs in ZIGZAG, then the pattern is determined to be closed. This is illustrated below for the case of the triangle.

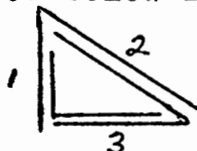


FIGURE 15. PAIR-WISE DESCRIPTION OF A TRIANGLE.
THE PAIRS WHICH COMPOSE THE TRIANGLE ARE
(1,2), (2,3), AND (3,1).

If the shape is not closed, then the lines which appear only once, the end lines of the open shape, are analyzed for their labels, their free, unconnected endpoints, and the label of the pair to which they belong. This information is stored in a temporary, two row array, that is used while the pattern is being processed.

Branching occurs according to the number of pairs which compose the pattern. All patterns containing more than four pairs are analyzed as 'zigzags'. Patterns composed of only one pair are identified as one of the corners described above. A pattern containing two pairs

is tested to see if it is an arch. An arch is recognized if it contains a type 1 corner associated with its leftmost end line and a type 2 corner associated with its rightmost end line. If the pattern is not an arch it is passed to the 'zigzag' analyzer.

If the number of pairs is three, or four, then the pattern is tested for closure. Currently triangles and squares are the only members of the world model which are closed, so they can be identified by this test. If the pattern does not satisfy the property of closure, then it is analyzed as a 'zigzag'.

The process of differentiating among the remaining subsets of zigzags depends on the left-to-right ordering of the types of corners. The array of endlines is accessed to determine which pair of the pattern is associated with the leftmost endline of the pattern. The pairs in ZIGZAG are then reordered by matching contiguous pairs, beginning with the leftmost pair.

The ordered pairs are first analyzed to screen out waveform patterns, since these are uniquely identified by the left-to-right corner description ...12341234... . This left-to-right sequence is illustrated below in figure 14.

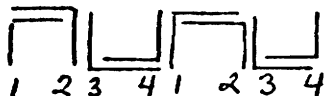


FIGURE 16. 'CORNER' DESCRIPTION OF A WAVEFORM ZIGZAG

If the pattern is not a waveform zigzag, then it is either a horizontal zigzag or a staircase zigzag. The slope of the pattern makes this final distinction. The slope is calculated from the leftmost endpoint to a point at the right extreme of the zigzag which has the same relationship to the zigzag as does the left point. If the left endpoint belongs to a high peak in the zigzag, then the right point should also belong to a high peak; and vice versa. This constraint is needed to obtain an accurate slope. If the number of pairs is odd, then the point of intersection of the last corner is the desired point; if the number of pairs is even, then the right endpoint of the zigzag will satisfy the constraint.

4.2.2 BOX. Subroutine box recognizes patterns which are composed of the following elements: a single arch or set of concentric arches, a set of isolated vertical lines which flank the arch/es to the right and left, and possibly a square or an isolated line in the center of the arch/es. Examples of box patterns are illustrated below:



FIGURE 17. EXAMPLES OF BOX PATTERNS.

The subroutine accesses array INARCH, to compare each arch to all the others, to test for a set of concentric

arches. In order for two arches to be considered concentric, the endpoints of one must lie between the endpoints of the other on the x-axis. This is a sufficient test for 'insideness' for the current world model of patterns. If any concentric arches are found, they are marked in array INARCH so that they won't be considered again. If no other arches were found that are concentric to the current arch, then the current arch is considered to be a set of concentric arches containing one member, in which it is both the largest and smallest arch. The number of arches in the current set is entered in column 1 of array INDBOX as part of the description of the box.

Next, an array of isolated vertical lines is prepared by testing all vertical lines in ALINES for non-membership in the array of connected pairs.

The procedure accesses array INDSQR to test for the existence of a square which lies inside the endpoints of the smallest arch. If this test fails, the array of isolated vertical lines is referenced to determine if any of these lines lies inside the smallest arch. These two conditions, or the condition of the absence of a pattern in the center of the arches, are part of the description of the arch, so one of three code numbers which identify these three conditions is entered in column 2 of array INDBOX.

In testing for the isolated lines which complete the

structure of a box, the procedure places constraints on the candidate lines so that lines belonging to parallel groups or lines that are isolated because of bad input data are not considered. The lines must be approximately the same length or longer than the sides of the largest arch; and the distance from one such line to the nearest side of the largest arch must approximate the average distance between the vertical elements of the incomplete box. Vertical elements include the sides of the arches; an isolated center line, if one exists; or the column coordinates of the midpoint of a square, if a square is one of the elements.

4.2.3 PARAL. Subroutine PARAL finds all unconnected parallel groups which stand alone as independent designs. It tests each parallel pair which is a member of PARGRP against the array of connected pairs (from which HORIZONTAL ZONE MARKERS have been removed), and the array of ZIGZAGS (these pairs had been previously removed from array CNECT), and any isolated lines which were determined to be elements of a box pattern; and marks in PARGRP any pair which belongs to any of these sets of lines.

After these pairs are marked, the index array IPAREL is accessed to determine the location of each separate group stored in PARGRP. Each group is tested, and if a group with unmarked pairs is found, it is output as an independent pattern. The slope of the group is part of the description

of the design, and it is calculated from the leftmost endpoint of the leftmost line to the leftmost endpoint of the rightmost line to obtain the average slope.

4.2.4 REGPAT. The only regions which are contained in the current data set are solid patterns whose outline forms a closed zigzag shape, in particular a triangular shape. The segments which were stored in array REGN by subroutine SEGMNT are processed by program REGPAT through a subset of the programs to analyze line patterns. It first finds all connected segmats, then all zigzag patterns, then calls subroutine SUBZIG to uniquely identify the pattern.

4.2.5 ORDER. The last stage in describing the input picture is to name the individual patterns in their left-to-right order. Each time an individual pattern was recognized, either by SUBZIG, PARAL, or REGPAT, an array entry was created in array PATTERN. Each entry contains the minimum and maximum x-coordinate of the pattern space occupied by the pattern. ORDER sorts the PATTERN entries so that the individual patterns can be described in their left-to-right place in the horizontal pattern band of the picture. PATTERN, as mentioned before, also contains a code as to the type of pattern which is stored, so that branching can take place during the description to obtain the specific details which describe the pattern.

CHAPTER FIVE

CONCLUSION

It is hoped that the work presented in this thesis makes a contribution to the solution of the problems faced by the archaeologist in processing artifactual information. The difficulties inherent in such studies - the size of the data base to be analyzed, the repetitive nature of the recording of pattern details, and the attention to detail required to classify design types - calls for the automation of the work load. It is hoped that the structural analysis, and expansion of several levels of analyses presented here offers one solution to the analysis phase of processing patterns, and that future extensions of this work might aid in the final, classification phase of pattern processing.

Possibilities for future extensions are discussed below.

5.1 INPUT DATA. The input data used in this project were of poor quality, and were difficult to process. The sherds which were photographed fell into two categories: those covered with a veneer of dark red glaze, and those finished with a gold or tan glaze. Each of these categories presented a different set of problems. The red glaze was generally intact, and appeared to have suffered little degradation over time; and the darkness of the glaze presented a strong

contrast to the lightness of the patterns. However, the glossy nature of the veneer caused areas of high illumination at points of curvature in the sherd. While the dull finish of the gold veneer did not result in varied illumination, it offered little contrast with the tone of the engraved patterns. In addition, shadowing which occurred during the photographing of the sherds resulted in shadows in the centers of the engraved lines, so that the contrasts in tone within the pattern line was often greater than the contrast between pattern and background.

The problems thus presented meant that a good deal of time had to be devoted to the pre-processing of the picture. The process of thresholding the picture locally was needed in most cases to extract the pattern from the background areas.

It is hoped that with techniques such as spraying the sherds with a solution to reduce the reflection of light from the surface of the sherd, and with more control of the lighting in photographing the sherds, that most, or all of these problems can be solved.

5.2 MORE COMPLEX PATTERNS. The system, in both stages, was written with the view that it would serve as the basis for further work on the problem. The system was structured so that new patterns could be included in the world model without disrupting the current structure of the system.

Such patterns as spirals, circles, and loops are being considered for future extensions; and they too can be analyzed as zigzags, or series of straight lines connected at mutual endpoints. The basis for a superficial distinction between these shapes and the zigzags already considered is that with respect to closed shapes, circles and scallops have much wider angles at points of intersections of lines than do the previously described zigzags; and with respect to the spiral, it is the only open-shaped zigzag whose corners do not progress along the x-axis in an increasing manner.

5.3 CLASSIFICATION OF PATTERNS. The current system could be expanded to include an automatic recording system which would, as it processed a series of pictures, keep statistics on patterns being processed. It could keep a count of the frequencies with which certain patterns appear, alone or in combination with other particular patterns.

This expanded system might provide data for classification by an archaeologist, or might serve as the basis for an automated classification procedure based on statistical techniques of analysis.

5.4 INFERRING WHOLE VESSELS FROM SHERDS. Finally, the system might be expanded to be able to piece together fragments which are adjacent pieces of the same pottery vessel. If the model had knowledge as to likely

combinations of patterns, or could infer likely candidates according to the size, or gray levels of various shards; or the model could detect the profiles of missing pieces of a pattern, then a fitting procedure might be used to sort and group candidates for common membership to an individual vessel.

Again, it is hoped that this work offers solutions, or ideas for solutions, to the tasks faced by archaeologists.

BIBLIOGRAPHY

- (1) Bordaz, J., and Von Hagen-Bordaz, V., "A Computer Pattern Recognition Method of Classification and Seriation Applied to Archaeological Material", *Archéologie et Calculateurs*, Paris, Editions du Centre National de la Recherche Scientifique, 1970
- (2) Binford, Lewis, "A Consideration of Archaeological Research Design", *American Antiquity*, Volume 29, pp. 425-441.
- (3) Clarke, D.L., *Analytical Archaeology*, London, Methuen, 1970.
- (4) Davis, L.S., and Rosenfeld, A., "Detection of Step Edges in Noisy One Dimensional Data", TR-303, University of Maryland, College Park.
- (5) Deetz, James, *Invitation to Archaeology*, Garden City, The Natural History Press, 1967.
- (6) Doran, J.E., and Hodson, F.R., *Mathematics and Computers in Archaeology*, Edinburgh, Edinburgh University Press, 1975..
- (7) Deutsch, E.S., "Comments on a line thinning scheme", *Computer Journal*, Volume 12, November 1969.
- (8) Duda, Peter O., and Hart, Peter E., *Pattern Classification and Scene Analysis*, New York, John Wiley & Sons, 1973.

- (9) Epstein, Judy, "A Computerized System for the Analysis and Classification of Straight Line Patterns Found on Potsherds", Master's Thesis, Moore School, University of Pennsylvania, Philadelphia, 1975.
- (10) Fu, K.-S., and Rosenfeld, A., "Pattern Recognition and Image Processing," IEEE Transactions on Computers, Volume C-25, December 1976.
- (11) Grief, Stephen, "A Pattern Recognition Program Using Induction", Master's Thesis, Moore School, University of Pennsylvania, Philadelphia, 1972.
- (12) Joshi, A.K., "Remarks on some aspects of language Structure and their relevance to Pattern analysis", Pattern Recognition Journal, Volume 5, Number 4, December, 1973, pp. 365-381.
- (13) Mc Pherron, Alan, "The Jutunen Site and the Late Woodland Prehistory of the Upper Great Lakes Area", Anthropological Papers, Number 30, Museum of Anthropology, University of Michigan, Ann Arbor, 1967.
- (14) Murthy, I., and Udupa, K., "A Search Algorithm for Skeletonization of Thick Patterns", Computer Graphics and Image Processing, Volume 3, September, 1974.
- (15) Narasimhan, R., and Reddy, V.S.N., "A Syntax-aided Scheme for handprinted English letters, Pattern Recognition, Volume 3, 1971, pp. 345-361.

- (16) Rajasekeran,S.N.S., and Deekshatulu,B.L., "Recognition of Printed Telegu Characters", Computer Graphics and Image Processing, Volum3 6, Number 4, August,1977.
- (17) Rosenfeld,A., and Kak,A.C., Digital Picture Processing, New York, Academic Press, 1976.
- (18) Rosenfeld,A. and Weszka,J.S., "Picture Recognition", Computer Science Technical Report Series, University of Maryland, College Park, December 1974.
- (19) Smith, Ray Winfield, "Computer Helps Scholars Recreate An Egyptian Temple", National Geographic, Volume 138, Number 5, November,1970, pp.634-655.
- (20) Tavakoli, Mohamad, "Analysis of Scenes as Seen from the Earth's Satelllites", Doctoral Dissertation, Moore School, University of Pennsylvania, Philadelphia, 1974.
- (21) Tidhar, Amnon, "Using a Structured World Model in Flexible Recognition of Two-Dimensional Patterns", Moore School Report No.75-02, University of Pennsylvania, Philadelphia, 1974.
- (22) Uhr, Leonard, "Flexible Linguistic Pattern Recognition", Pattern Recognition, Volume 3, 1971,pp.363-383.
- (23) Uhr, Leonard, Pattern Recognition, Learning, & Thought, Englewood Cliffs, Prentice-Hall,Inc.,1973,pp.150-162.

APPENDIX

Presented in the APPENDIX are descriptions of the nine samples used for this study. For each sherd in the sample, there are five pictures and a structural description.

The pictures labeled (1) are the digitized pictures. The pictures with label (2) represent the results of the thresholding operation. Those labeled (3) are the results of both the thinning operator, and the strip-finder which marked line patterns black and region areas dark gray. The pictures labeled (4) show skeletonized line patterns and/or outlined region patterns. The (5) pictures show the final results: all line elements in the pattern.

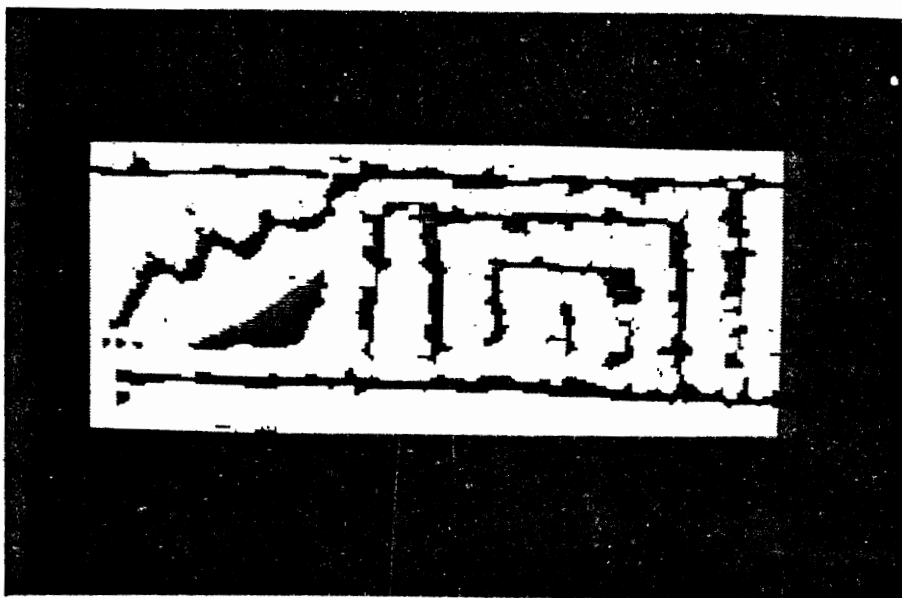


(1)

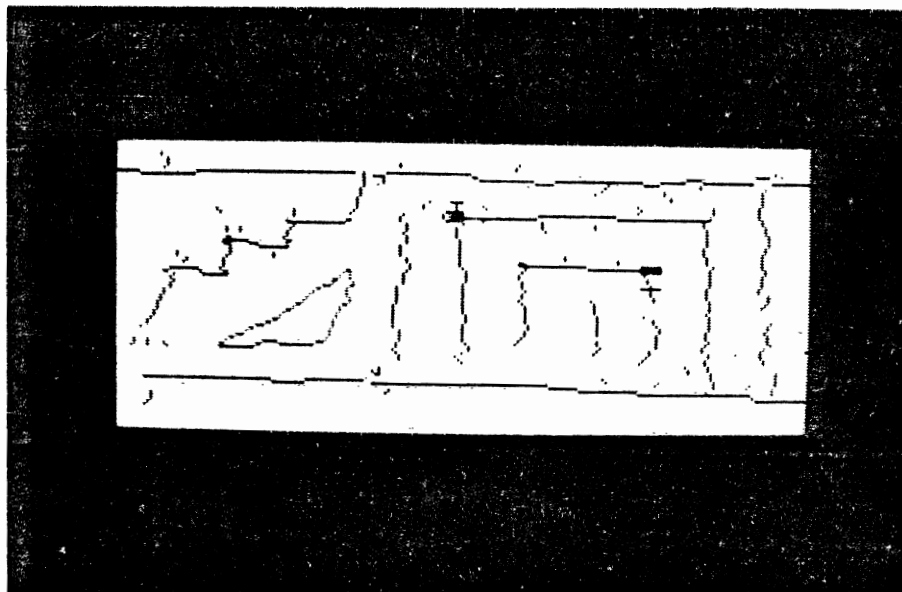


(2)

PICTURE 1.

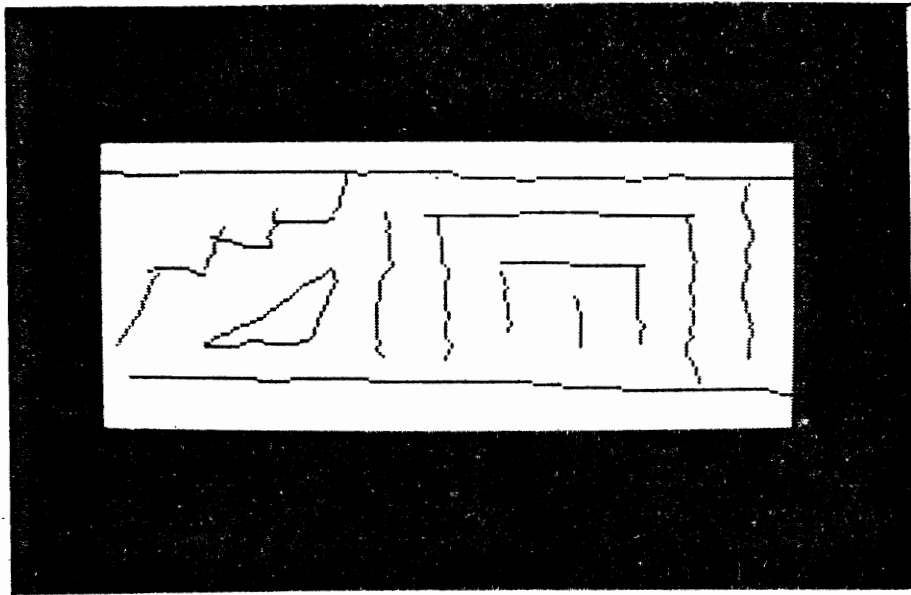


(3)



(4)

PICTURE 1.



(5)

PICTURE 1

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 7
 NUMBER OF VERTICAL LINE SEGMENTS= 7
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 0
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 4

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	2	0	5
2	3	0	5
3	2	-1	10
4	2	-1	15
5	4	999	15
6	3	999	17
7	2	0	66

NUMBER OF PARALLEL HORIZONTAL GROUPS= 3
 NUMBER OF PARALLEL VERTICAL GROUPS= 2
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 0
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 2
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 1
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 1

ZIGZAG PATTERN GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS
1	3
2	7
3	3

NUMBER OF ZIGZAG PATTERNS= 3
 NUMBER OF ZIGZAG "SUBPATTERNS"= 0

LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS:

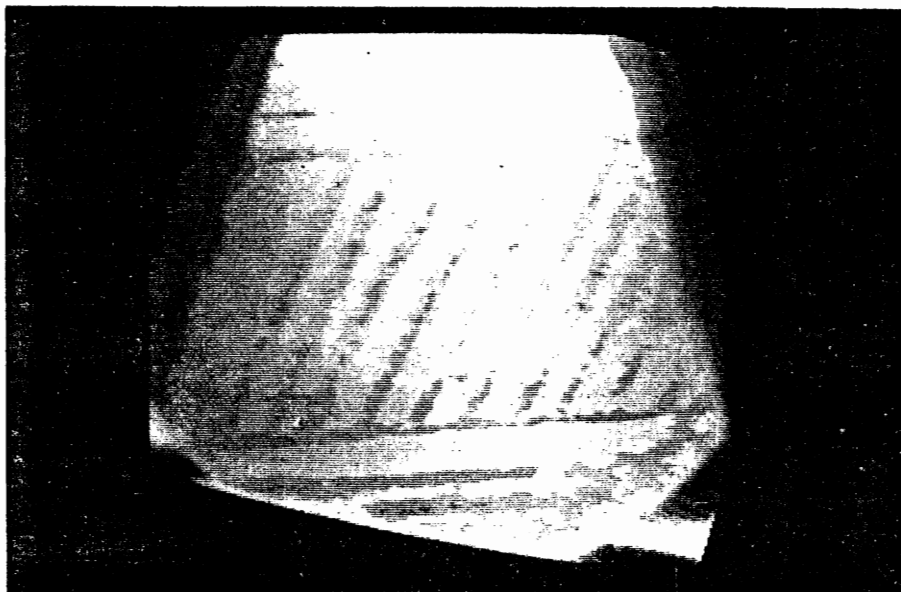
A STAIRSTEP ZIGZAG

A TRIANGULAR REGION

A BOX, WITH:

NUMBER OF ARCHES = 2

AN ISOLATED VERTICAL LINE IN THE CENTER

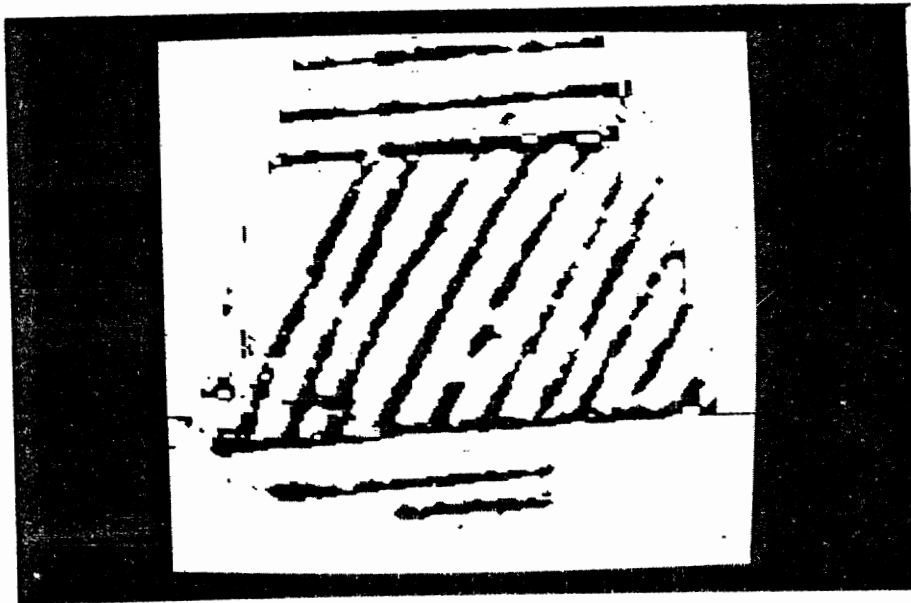


(1)

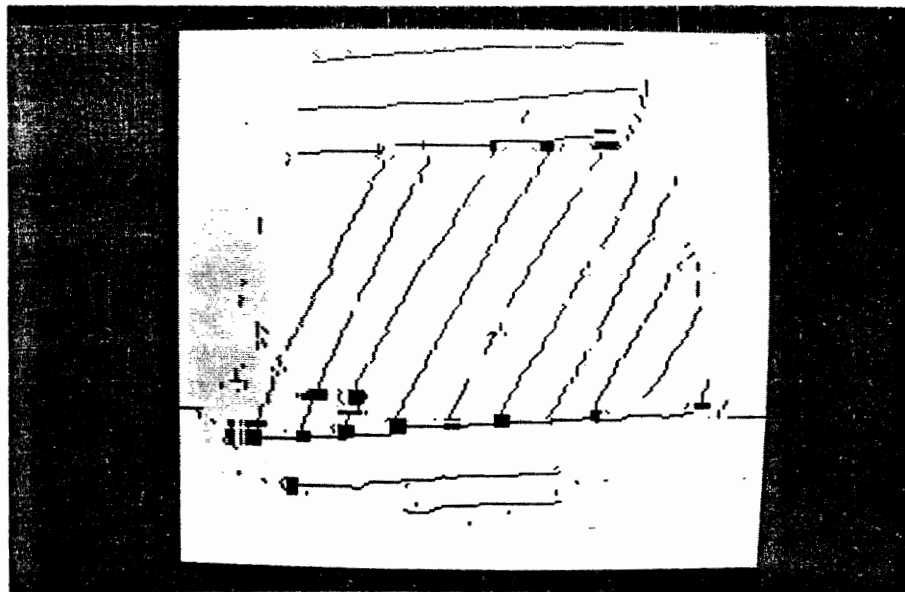


(2)

Picture 2.

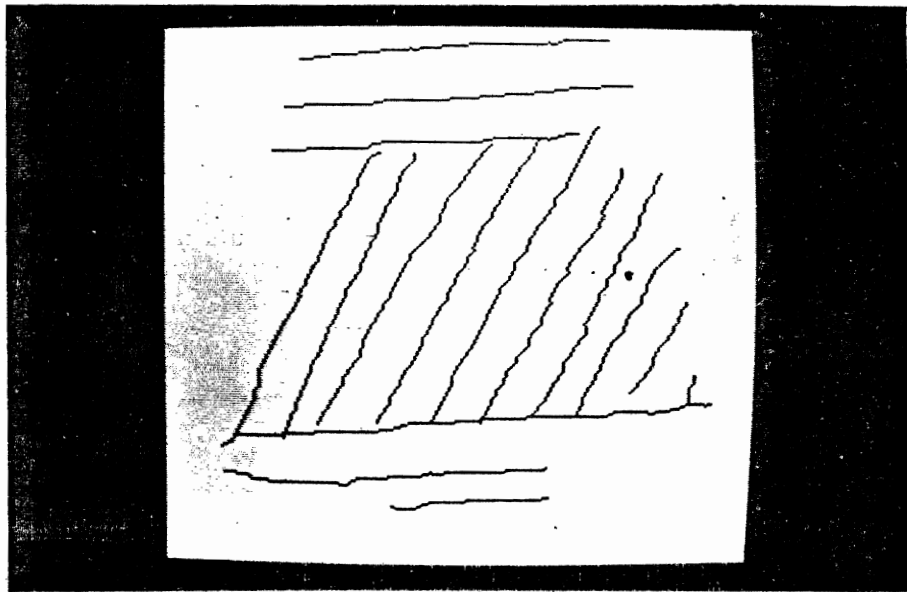


(3)



(4)

PICTURE 2.



(5)

PICTURE 2.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 6
 NUMBER OF VERTICAL LINE SEGMENTS= 0
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 0
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 10

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	3	0	13
2	10	-1	13
3	3	0	18

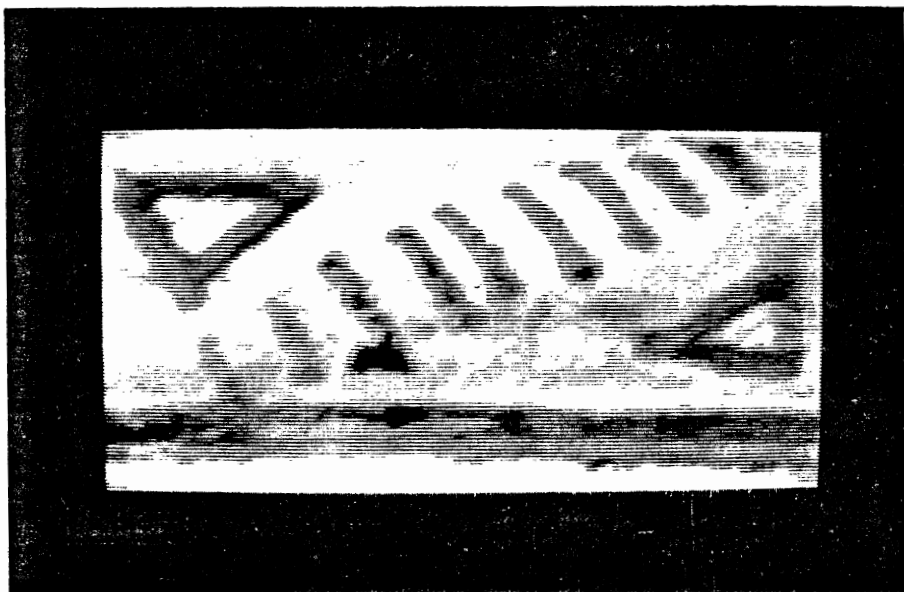
NUMBER OF PARALLEL HORIZONTAL GROUPS= 2
 NUMBER OF PARALLEL VERTICAL GROUPS= 0
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 0
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 1
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 3
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 3

LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS ONE PATTERN:

A PARALLEL GROUP WHERE:

SLOPE OF THE LINES= -1
 SLOPE OF THE GROUP= 0

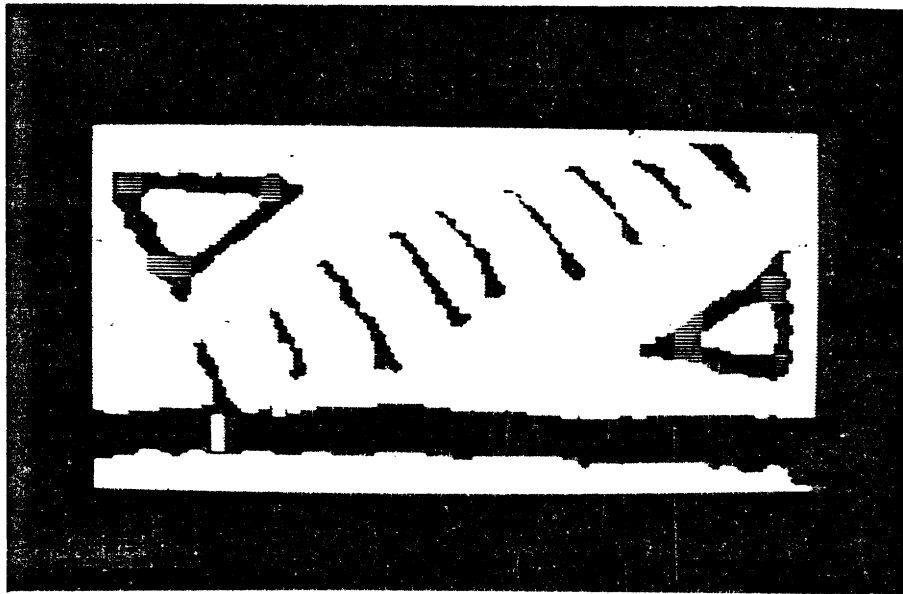


(1)

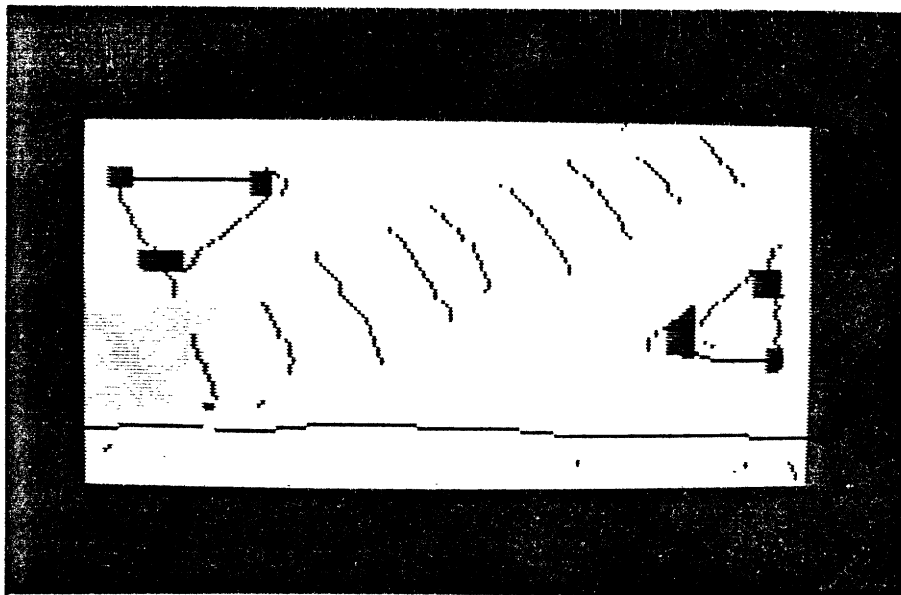


(2)

PICTURE 3.

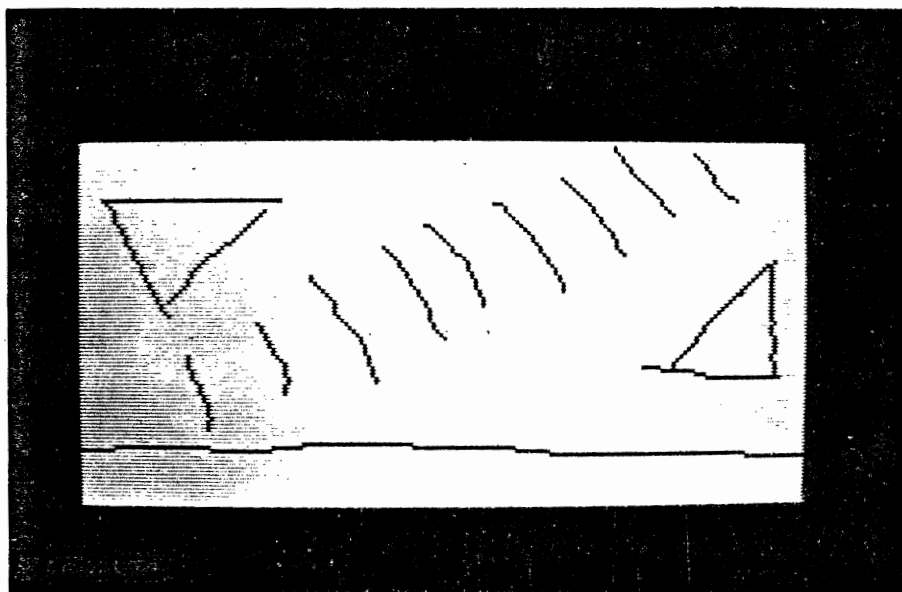


(3)



(4)

PICTURE 3.



(5)

PICTURE 3.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 3
 NUMBER OF VERTICAL LINE SEGMENTS= 1
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 10
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 2

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	9	1	12
2	2	0	22
3	2	-1	109

NUMBER OF PARALLEL HORIZONTAL GROUPS= 1
 NUMBER OF PARALLEL VERTICAL GROUPS= 0
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 1
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 1
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 0
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 1

ZIGZAG PATTERN GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS
1	3
2	3

NUMBER OF ZIGZAG PATTERNS= 2
 NUMBER OF ZIGZAG "SUBPATTERNS"= 0

LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS:

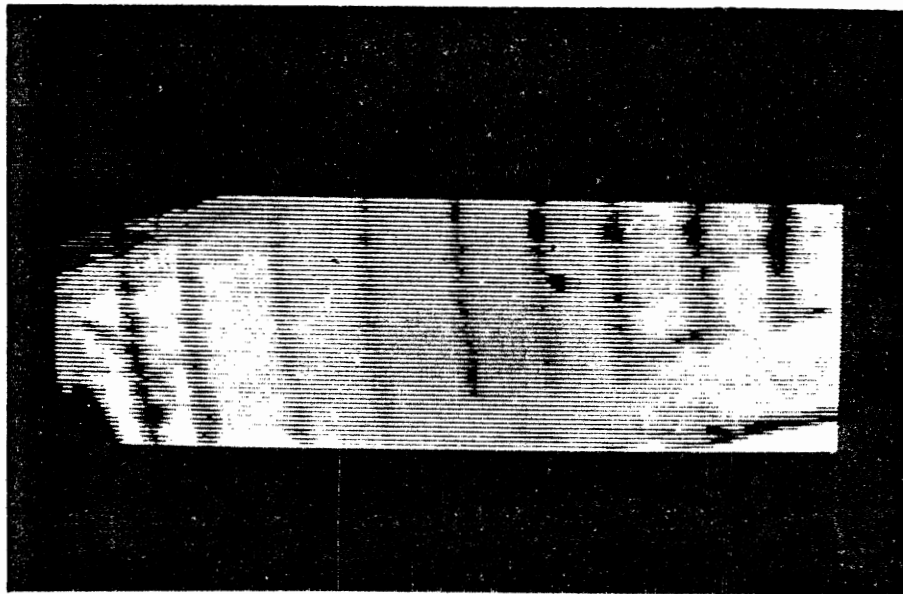
A TRIANGLE

A PARALLEL GROUP WHERE:

SLOPE OF THE LINES= 1

SLOPE OF THE GROUP= -1

A TRIANGLE

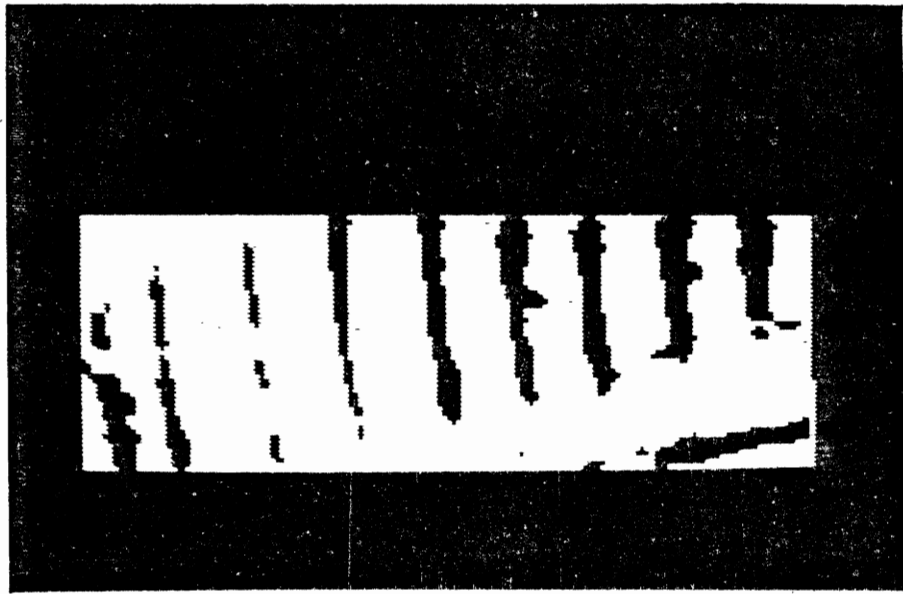


(1)

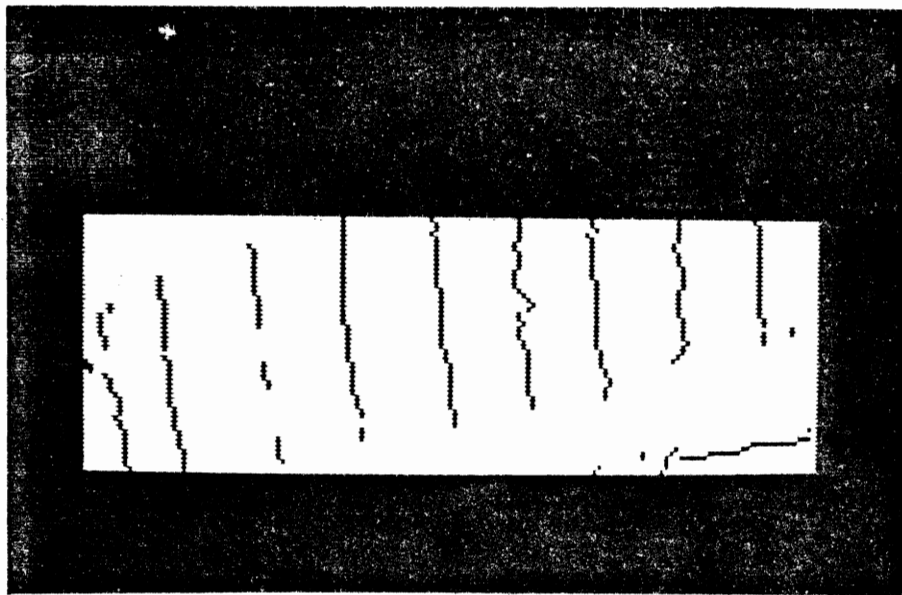


(2)

PICTURE 4.

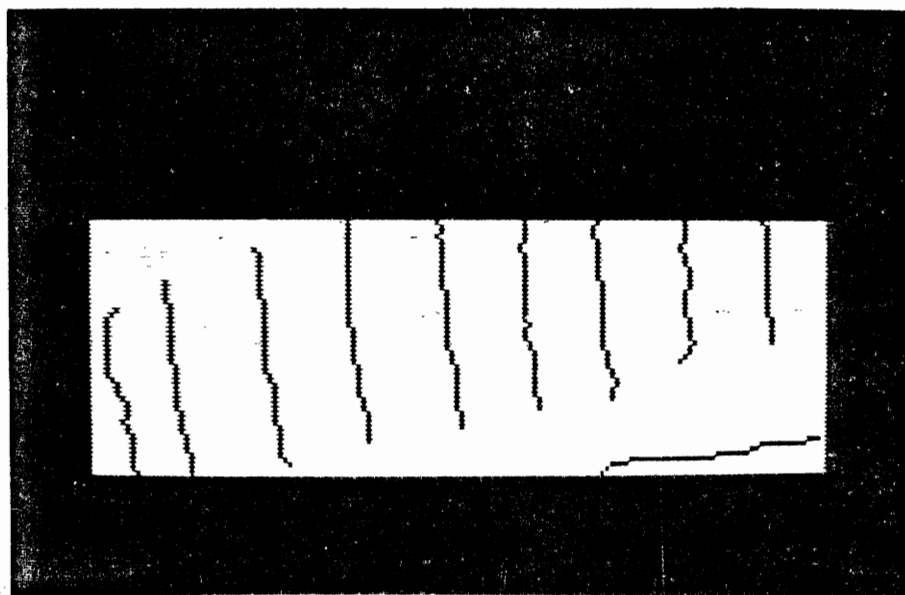


(3)



(4)

PICTURE 4.



(5)

PICTURE 4.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 1
NUMBER OF VERTICAL LINE SEGMENTS= 9
NUMBER OF LINE SEGMENTS WITH SLOPE +1= 0
NUMBER OF LINE SEGMENTS WITH SLOPE -1= 0

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER 1	NUMBER OF SEGMENTS 9	SLOPE OF GROUP 999	DISTANCE BETWEEN 8
NUMBER OF PARALLEL HORIZONTAL GROUPS= 0			
NUMBER OF PARALLEL VERTICAL GROUPS= 1			
NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 0			
NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 0			
NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 0			
NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 1			

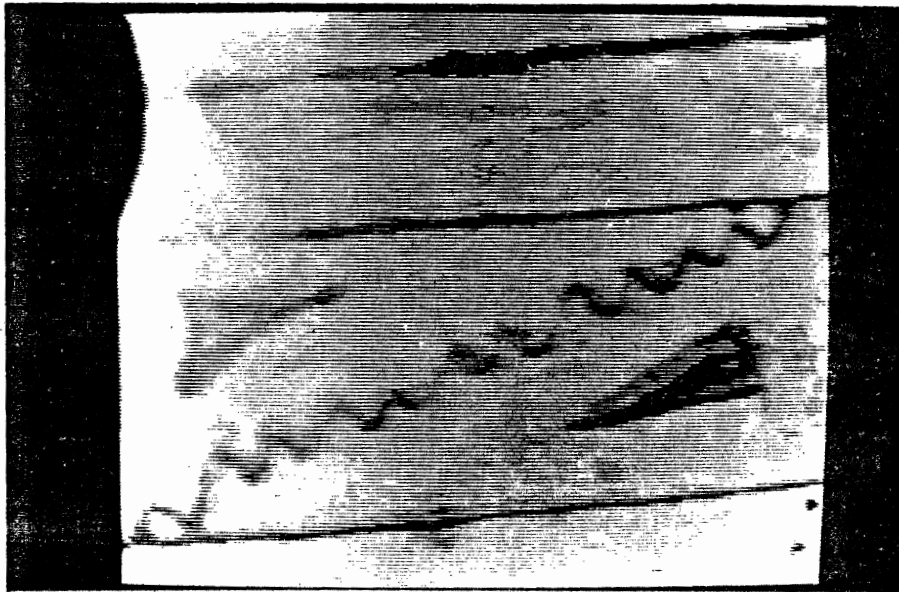
LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS ONE PATTERN:

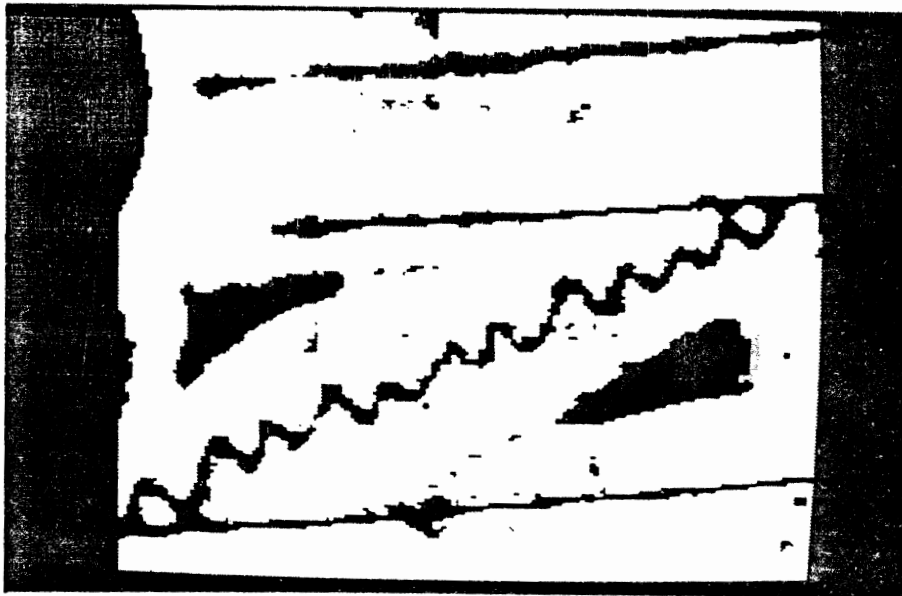
A PARALLEL GROUP WHERE:

SLOPE OF THE LINES= 999

SLOPE OF THE GROUP= 0

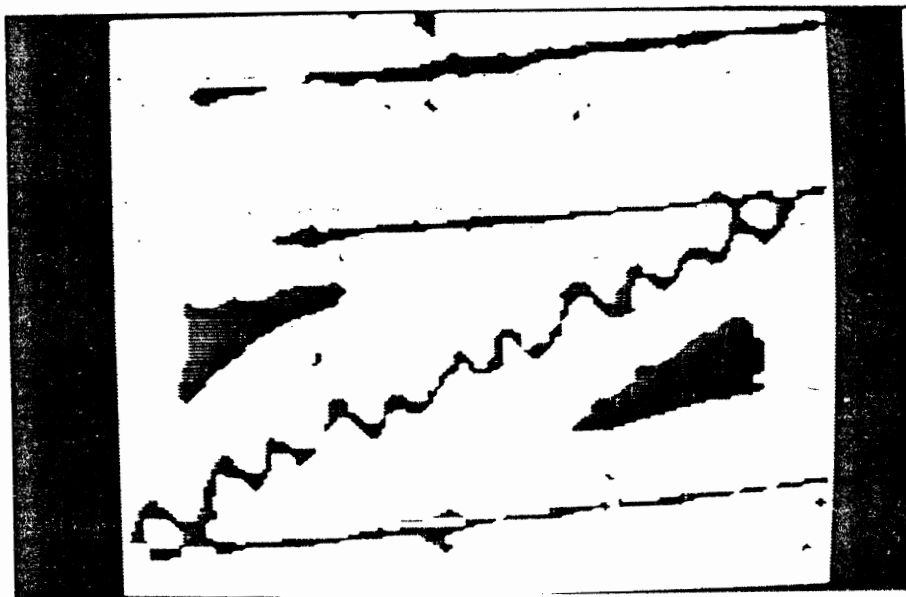


(1)

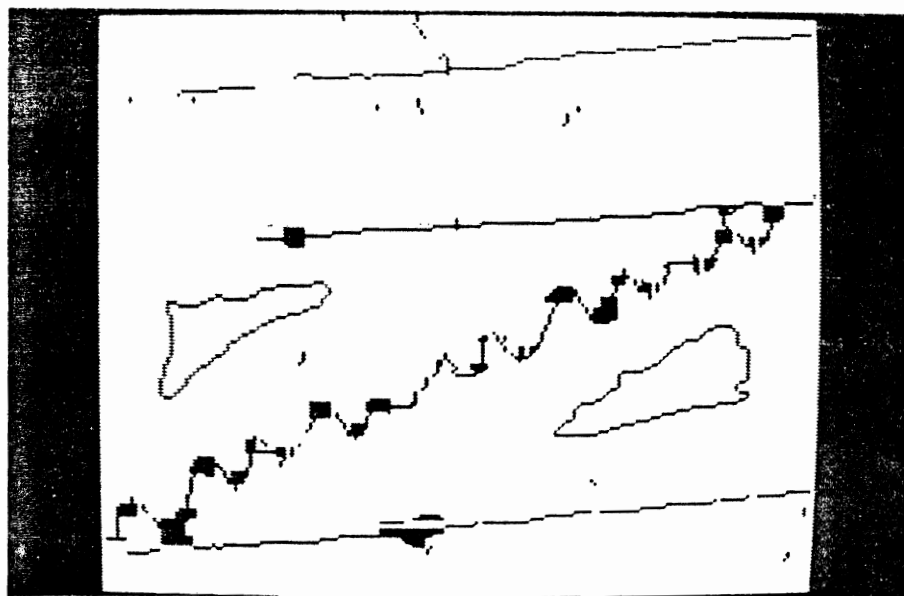


(2)

PICTURE 5.

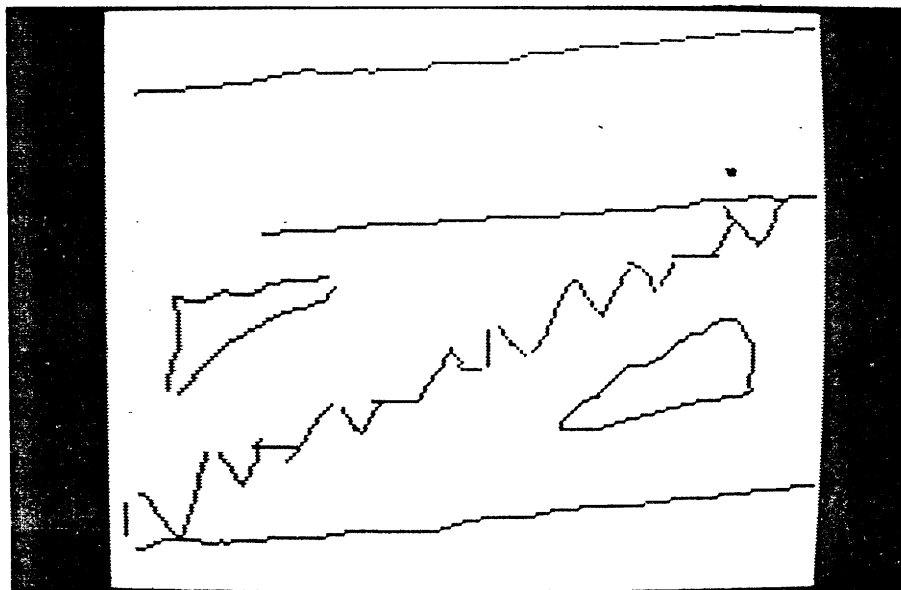


(3)



(4)

PICTURE 5.



(5)

PICTURE 5.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 7
 NUMBER OF VERTICAL LINE SEGMENTS= 2
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 8
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 10

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	3	0	7
2	5	-1	7
3	5	-1	8
4	2	1	14
5	2	1	14
6	2	1	24
7	2	0	44
8	2	0	68
9	2	999	98
10	2	1	117

NUMBER OF PARALLEL HORIZONTAL GROUPS= 3
 NUMBER OF PARALLEL VERTICAL GROUPS= 1
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 4
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 2
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 2
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 1

ZIGZAG PATTERN GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS
1	24

NUMBER OF ZIGZAG PATTERNS= 1
 NUMBER OF ZIGZAG "SUBPATTERNS"= 0

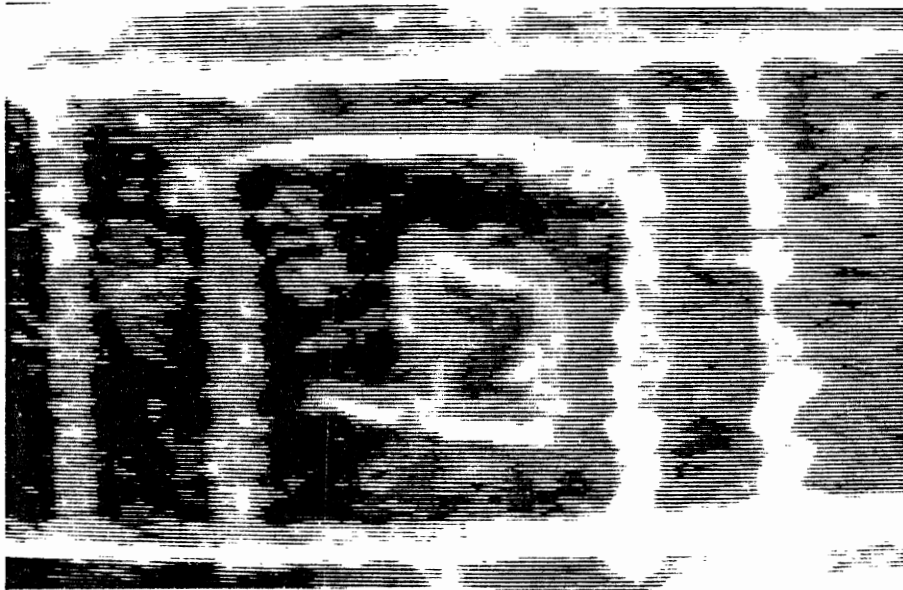
LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS:

A STAIRSTEP ZIGZAG

A TRIANGULAR REGION

A TRIANGULAR REGION

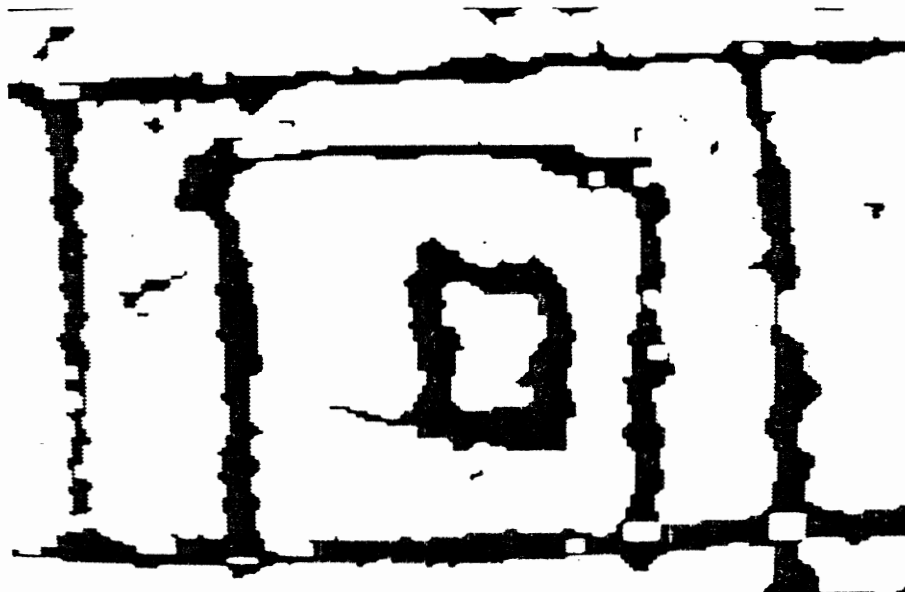


(1)

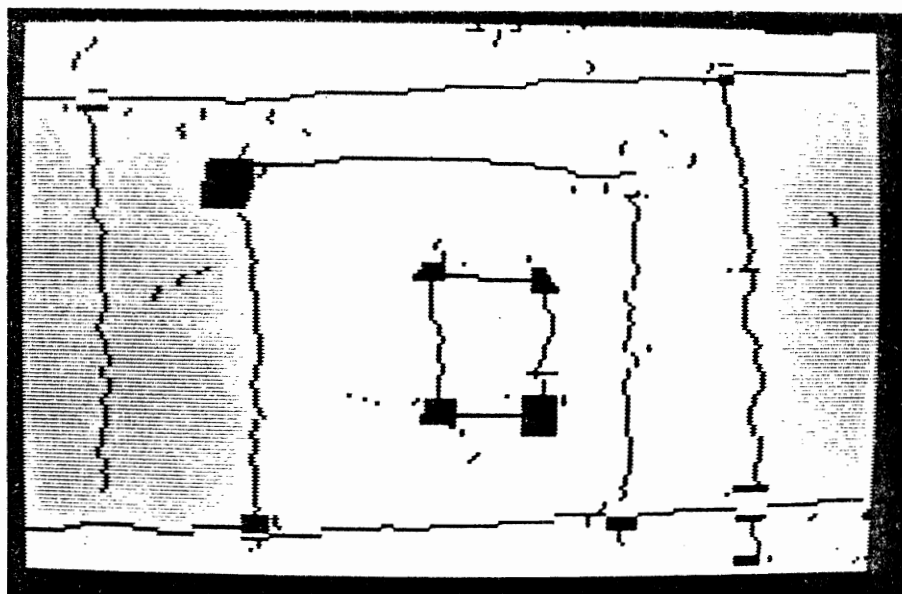


(2)

PICTURE 6.

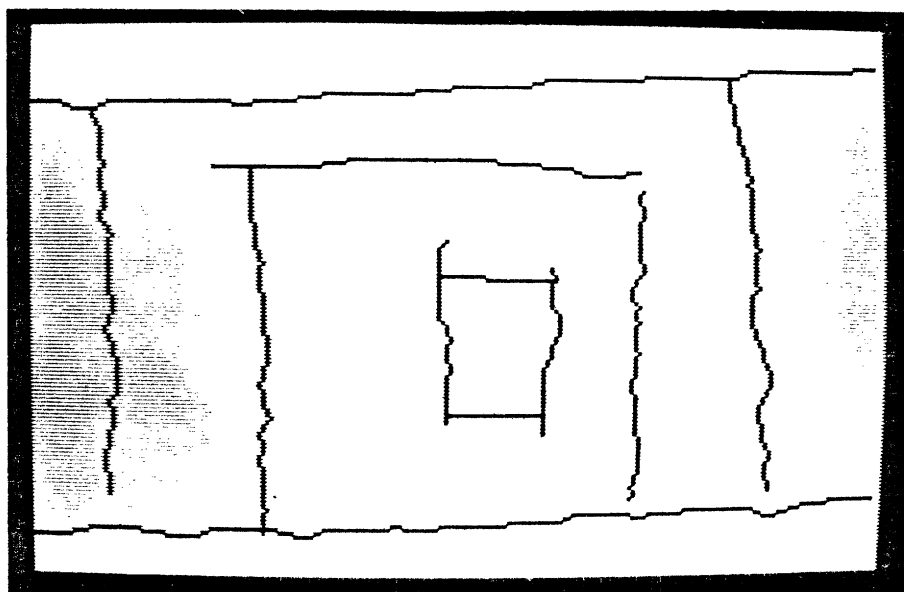


(3)



(4)

PICTURE 6.



(5)

PICTURE 6.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 5
 NUMBER OF VERTICAL LINE SEGMENTS= 6
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 0
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 0

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	4	999	20
2	5	0	20
3	2	999	37

NUMBER OF PARALLEL HORIZONTAL GROUPS= 1
 NUMBER OF PARALLEL VERTICAL GROUPS= 2
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 0
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 0
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 1
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 1

ZIGZAG PATTERN GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS
1	3
2	4

NUMBER OF ZIGZAG PATTERNS= 2
 NUMBER OF ZIGZAG "SUBPATTERNS"= 0

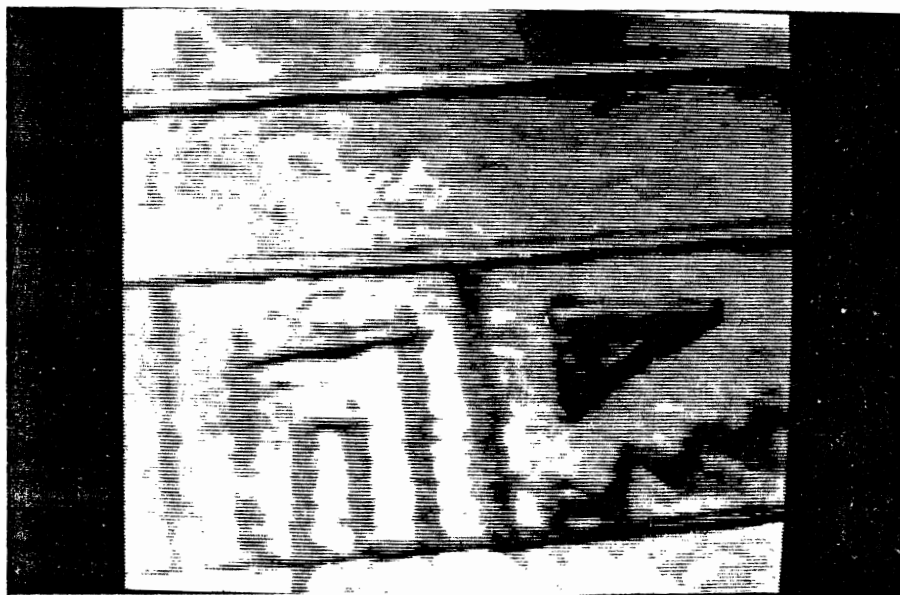
LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS ONE PATTERN:

A BOX, WITH:

NUMBER OF ARCHES= 1

A SQUARE IN THE CENTER

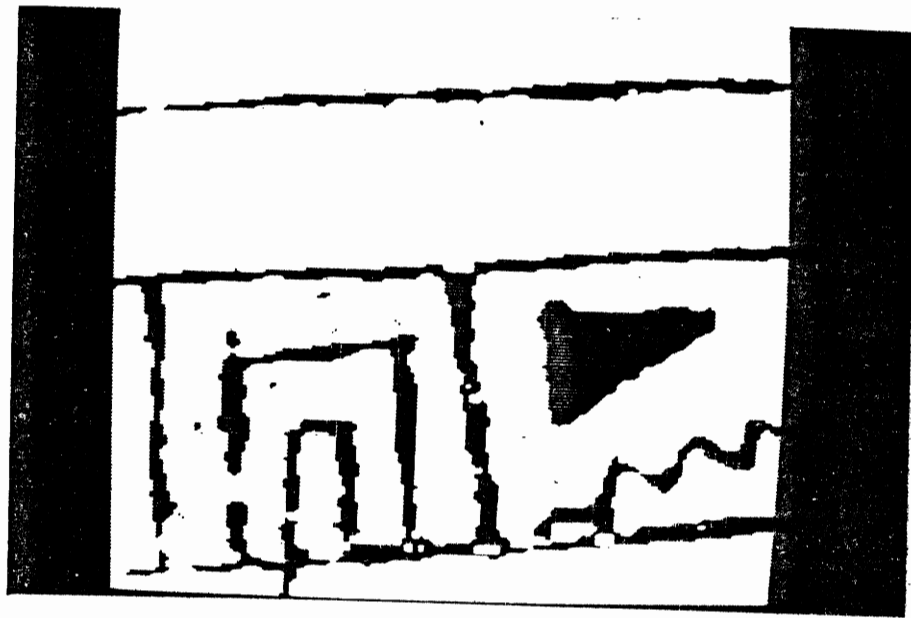


(1)

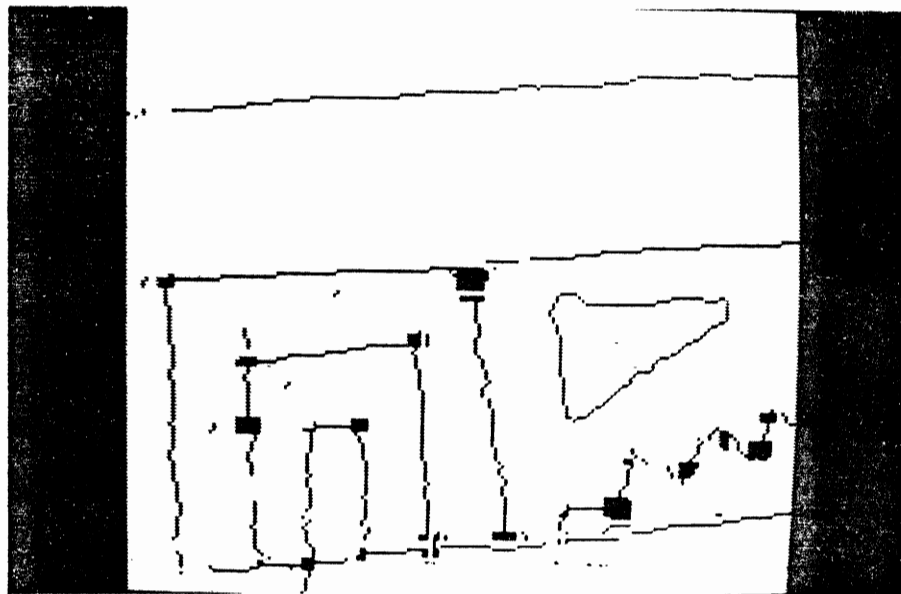


(2)

PICTURE 7.

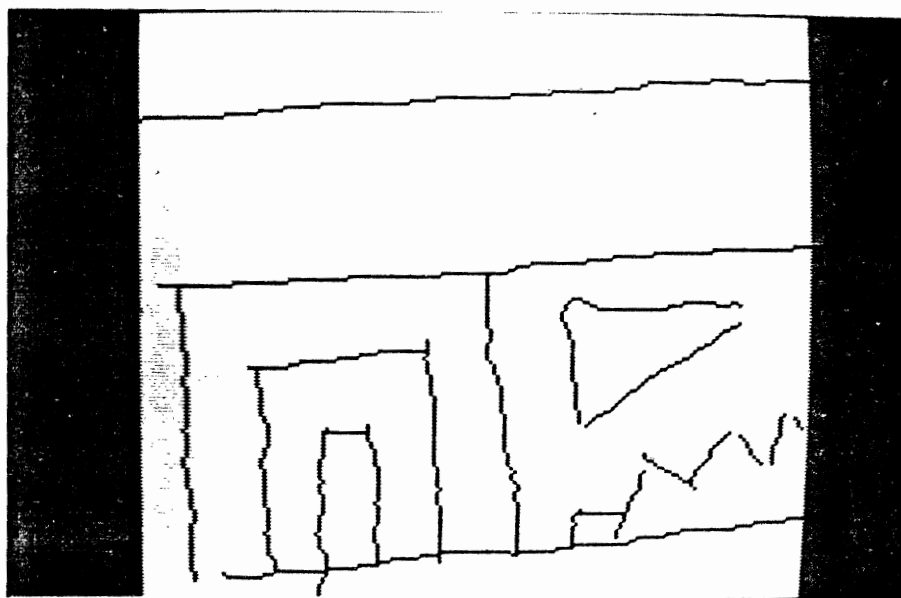


(3)



(4)

PICTURE 7..



(5)

PICTURE 7.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 6
 NUMBER OF VERTICAL LINE SEGMENTS= 6
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 3
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 4

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	2	0	7
2	6	999	10
3	4	-1	10
4	3	1	14
5	3	0	21

NUMBER OF PARALLEL HORIZONTAL GROUPS= 2
 NUMBER OF PARALLEL VERTICAL GROUPS= 1
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 1
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 1
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 2
 NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 1

ZIGZAG PATTERN GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS
1	8
2	3
3	3

NUMBER OF ZIGZAG PATTERNS= 3
 NUMBER OF ZIGZAG "SUBPATTERNS"= 0

LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS:

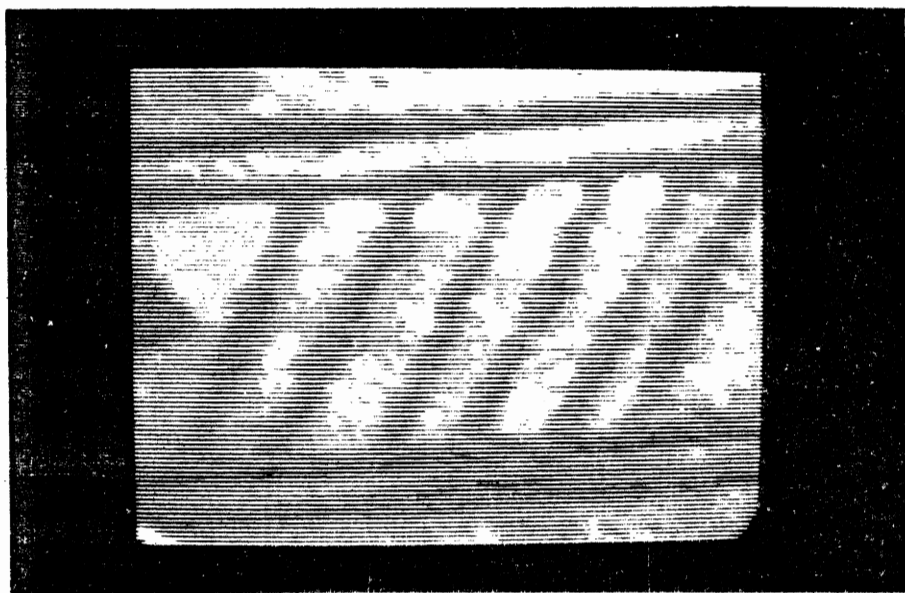
A BOX, WITH:

NUMBER OF ARCHES= 2

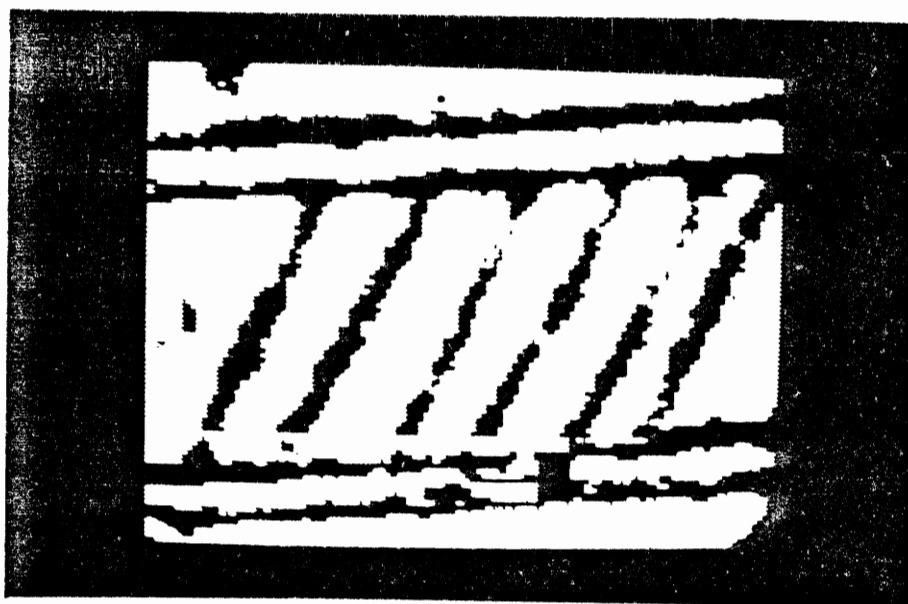
NO PATTERN STRUCTURE IN THE CENTER

A TRIANGULAR REGION

A STAIRSTEP ZIGZAG

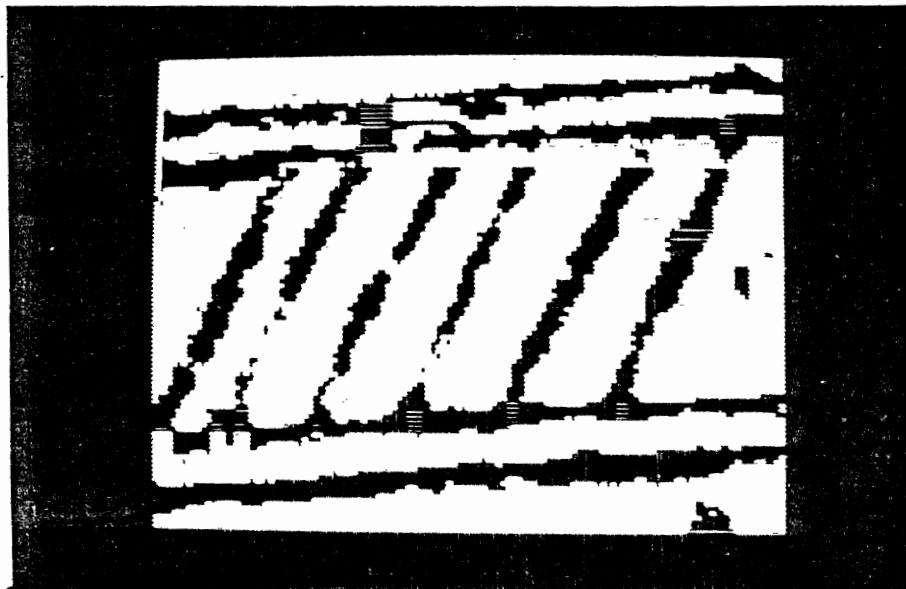


(1)

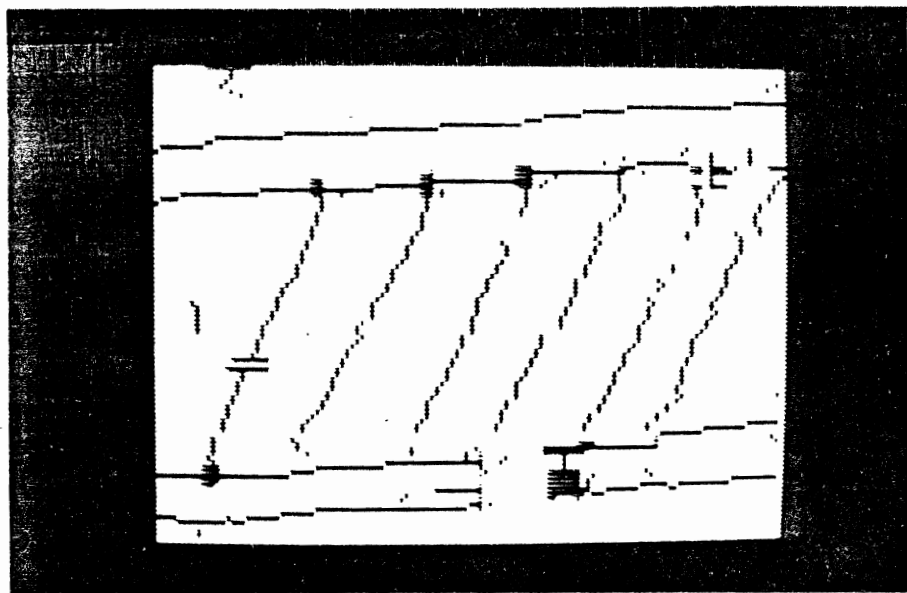


(2)

PICTURE 8.

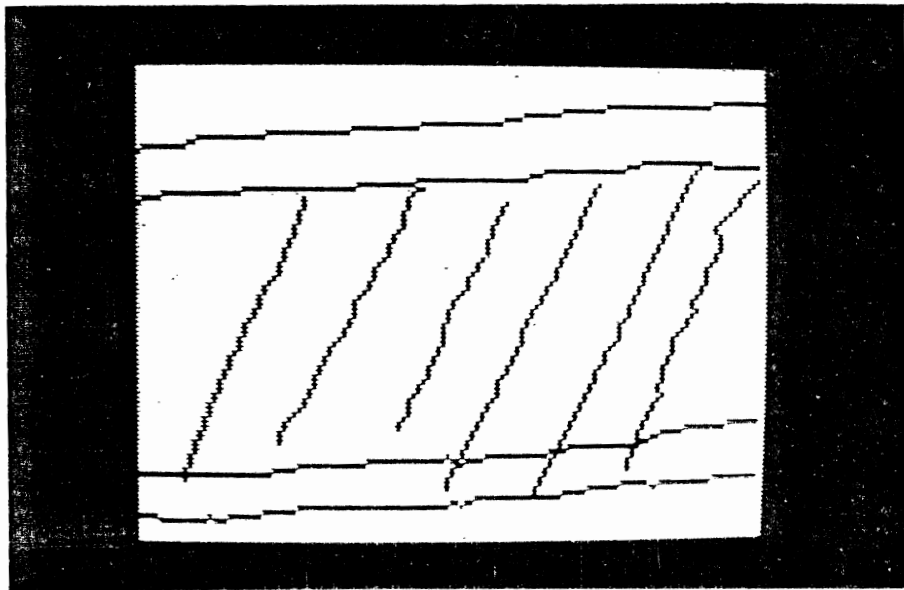


(3)



(4)

PICTURE 8.



(5)

PICTURE 8.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 4
NUMBER OF VERTICAL LINE SEGMENTS= 0
NUMBER OF LINE SEGMENTS WITH SLOPE +1= 0
NUMBER OF LINE SEGMENTS WITH SLOPE -1= 6

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	2	0	9
2	6	-1	12
3	2	0	12

NUMBER OF PARALLEL HORIZONTAL GROUPS= 2
NUMBER OF PARALLEL VERTICAL GROUPS= 0
NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 0
NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 1
NUMBER OF HORIZONTAL ZONE MARKERS AT THE TOP OF THE PATTERN= 2
NUMBER OF HORIZONTAL ZONE MARKERS AT THE BOTTOM OF THE PATTERN= 2

LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS ONE PATTERN:

A PARALLEL GROUP WHERE:

SLOPE OF THE LINES= -1
SLOPE OF THE GROUP= 0

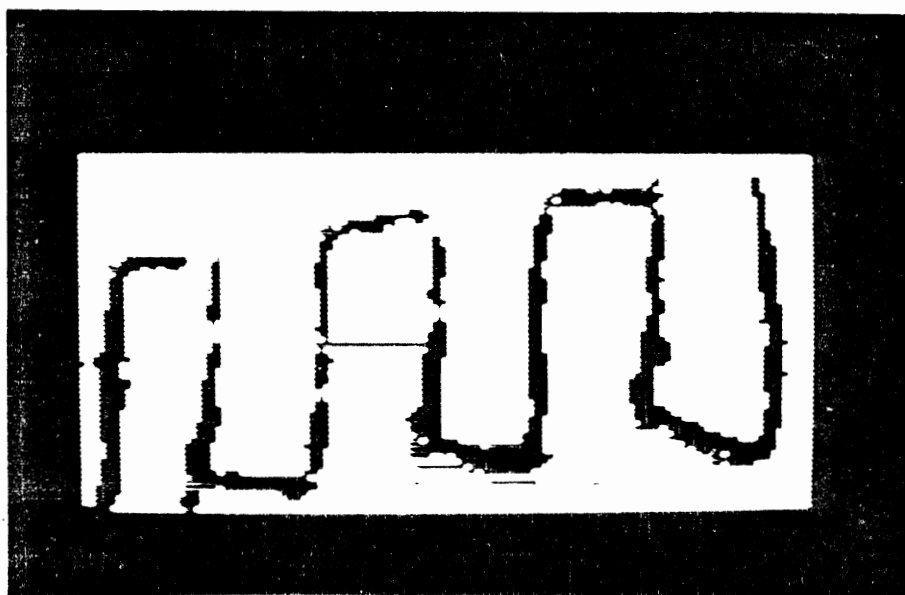


(1)

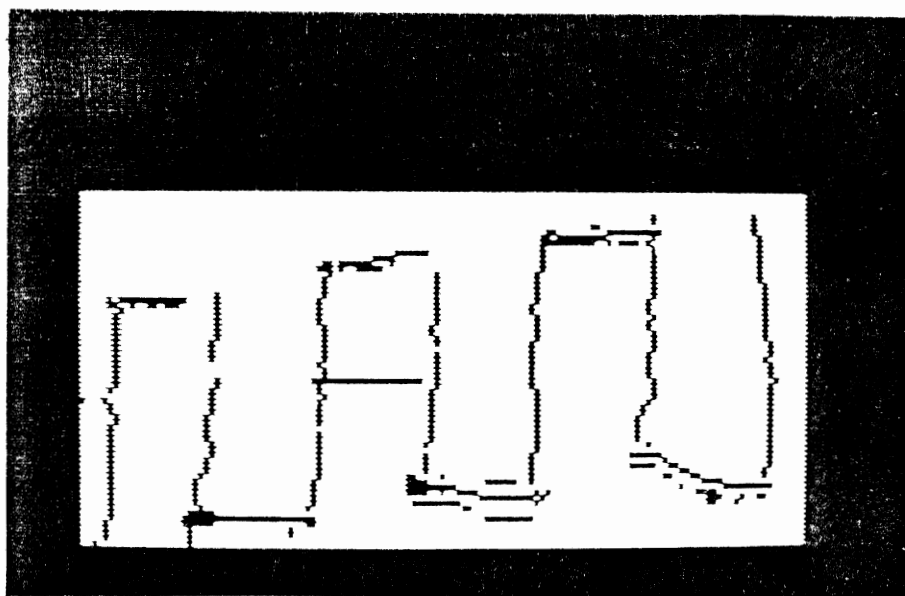


(2)

PICTURE 9.

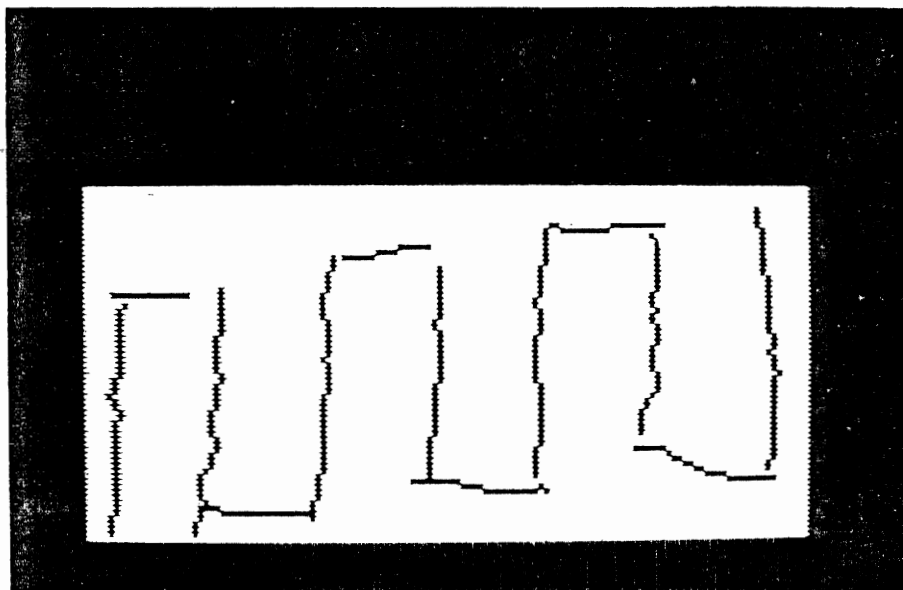


(3)



(4)

PICTURE 9.



(5)

PICTURE 9.

ANALYSIS OF INPUT

NUMBER OF HORIZONTAL LINE SEGMENTS= 6
 NUMBER OF VERTICAL LINE SEGMENTS= 7
 NUMBER OF LINE SEGMENTS WITH SLOPE +1= 0
 NUMBER OF LINE SEGMENTS WITH SLOPE -1= 0

PARALLEL LINE GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS	SLOPE OF GROUP	DISTANCE BETWEEN
1	3	0	6
2	7	999	14

NUMBER OF PARALLEL HORIZONTAL GROUPS= 1
 NUMBER OF PARALLEL VERTICAL GROUPS= 1
 NUMBER OF PARALLEL GROUPS WITH SLOPE +1= 0
 NUMBER OF PARALLEL GROUPS WITH SLOPE -1= 0
 THIS PATTERN DOES NOT HAVE A HORIZONTAL ZONE MARKER

ZIGZAG PATTERN GROUP DIRECTORY

GROUP NUMBER	NUMBER OF SEGMENTS
1	13

NUMBER OF ZIGZAG PATTERNS= 1
 NUMBER OF ZIGZAG "SUBPATTERNS"= 0

LEFT-TO-RIGHT ANALYSIS OF INPUT

THE PICTURE CONTAINS ONE PATTERN:
 A WAVEFORM ZIGZAG