A PARADIGM FOR SEMANTIC PICTURE RECOGNITION

A THESIS

Presented to

The Faculty of the Division of Graduate

Studies and Research

by

Michael L. loyd Baird

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in the School of Information and Computer Science

Georgia Institute of Technology

June, 1973

A PARADIGM FOR SEMANTIC PICTURE RECOGNITION

Approved:

Chairman: Michael D. Kelly

Member: Pranas Zunde

Member: Miroslav Valach

Member: Perry Sprawls
Emory University

External Reader: D. R. Reddy
Carnegie-Mellon University

Date approved by Chairman: May 15, 1973

ACKNOWLEDGMENTS

I would like to first thank my thesis advisor, Dr. Michael D. Kelly, for his enthusiastic support, ideas, and constructive criticism, without which this dissertation would not have been possible. In addition, I am especially appreciative of the TV scan data provided me by Dr. Kelly from his work at Stanford University. A note of thanks goes to Mr. S. M. Basu who permitted me to use his computer program for translating this data from PDP10 code into the required B5500 code.

I would also like to thank Drs. David E. Rogers, Perry Sprawls, Miroslav Valach, and Pranas Zunde for the great effort they contributed in the guidance of the research and in the reading of the thesis. Special appreciation is given for Dr. Zunde's contribution in the refinement of the formalism of Chapter IV.

I also thank Mr. Alton P. Jensen. If it were not for his assistance in placing me in a research position at the Engineering Experiment Station early in my program, this dissertation topic might never have been developed.

Special thanks goes to Dr. Vladimir Slamecka who made this research possible through the financial assistance from graduate teaching and research positions in the School of Information and Computer Science of the Georgia Institute of Technology.

I am especially grateful for Professor D. R. Reddy's willingness to serve as an external reader. His comments on the thesis were most helpful.

To Dr. Karl Murphy I express sincere gratitude for his prompt reading of a draft of the dissertation. His suggestions resulted in a significant enhancement of the style of the dissertation.

Work on this dissertation has been supported in part by National Science Foundation Grant GN-655. This assistance is gratefully acknowledged.

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Concluded)

SUMMARY

This dissertation concerns itself with picture recognition by computer. Existing paradigms for picture recognition are presented, and comments made as to their limitations. It is found that these paradigms are inadequate for many picture recognition tasks. A more powerful paradigm for picture recognition which might overcome some of the inadequacies of existing paradigms has been proposed by several authors, and their insights into what the objectives of such a paradigm might be are presented. Based on these objectives and the knowledge gained from an analysis of the inadequacies of the existing paradigms, a new paradigm for picture recognition is developed. The evaluation of the paradigm is accomplished through its application to several interesting picture recognition problems.

The receptor/categorizer paradigm for picture recognition is found to be of little use when analyzing complex pictures where structure and interrelationships among the picture components are important factors. The syntactic paradigm also is found to have serious limitations. In particular, problems involving (1) ambiguity of shape, (2) non-pictorial paraphrase, (3) non-ideal data, and (4) multistability in perception are, in general, characteristic of those cases in which techniques based on the syntactic paradigm are found to be inadequate. Other techniques for picture recognition which do not fall within the receptor/categorizer or syntactic paradigms are ad hoc and heuristic, and hence of little value

for other than their intended applications.

It is found that the underlying causes for the failure of the syntactic paradigm to apply to certain recognition problems are the use of extensional (versus intensional) class descriptions and an inability to utilize contextual information. The new paradigm, therefore, was developed in such a way that contextual information could be explicitly utilized, and recognition could be performed based on the implicit use of intensional class descriptions. The paradigm provides a general model whereby many problems previously unsolvable by the receptor/categorizer or syntactic paradigms can now be solved. This new paradigm is termed a paradigm for semantic picture recognition, or in short, the semantic paradigm. It was so named because it provides a procedure by which the "semantics" of a picture can be exhibited. That is, the use of the semantic paradigm permits the exhibition of relations and properties of a non-pictorial kind which describe the event depicted by the picture "syntax."

The semantic paradigm can be informally described as the following procedure: first, a primitive description is constructed of the scene depicted by a picture. This description is a function of the information processor's sensory and preprocessing facilities. Secondly, a set of "rules of inference" is constructed which can be thought of as the information processor's body of knowledge of the world. Each of these rules is an independent entity, whose application to a scene description may result in the inference of new properties and relations. Recognition is implicit in the process of applying these rules since the result is a scene description in which objects are described in terms of properties which are found to hold for them, and relations between objects are exhibited.

The contribution that this paradigm makes is fairly straightforward. It presents a very concrete structure for the programmer to follow in approaching a recognition problem. This structure is such that the universal aspects of any picture recognition problem are isolated from the problem-dependent aspects. Thus, the programmer is freed from the task of developing a completely ad hoc program which is difficult to debug and modify. The problem-dependent aspects of the program are succinctly contained in the set of rules of inference which can be rapidly expanded or modified by the programmer to develop a solution for the particular recognition task.

The semantic paradigm is shown to be a valuable contribution to the field through its successful application to problems that heretofore were unsolvable using the existing paradigms. In particular, the following four problems were considered.

(1) For a problem involving ambiguity of shape, a hypothetical world "W" was described in which the recognition task was to distinguish objects having the same shapes but belonging to different classes. For example, world W contains air-filled balloons, gas-filled balloons, solid balls, and egg-shaped lead blobs. Depending upon the context in which an object is found a classification decision can be made which could not be made by analyzing only shape. For example, balloons usually look like balls, but at times, balloons might be supporting a weight which would give them the shape of lead blobs. The explicit use of contextual information through the semantic paradigm provided a solution to this problem.

(2) For a problem involving non-pictorial paraphrase, the problem of recognizing (the function performed by) electrical control circuits was considered. It is shown that an infinite variety of circuits could perform the same function, and hence belong to the same class. Thus, existing paradigms were found to be completely inadequate. The semantic paradigm, however, provided the necessary structure to solve this problem.

(3) For a problem involving non-ideal data, an attempt was made to recognize features such as hair, eyes, eyebrows, nostrils, and the mouth in grey level pictures of human faces. Although this problem can not be solved using existing paradigms, an ad hoc solution has been given (Kelly, 1970). The semantic paradigm, however, proved to provide a more straightforward and flexible solution.

(4) Lastly, a problem involving multistability in perception was investigated, namely, that of attempting to recognize the object in a picture of a reversible (Necker) cube. It was found that the semantic paradigm solution produced the multistable behavior encountered by human beings. That is, first the cube is "seen" from one aspect, and then the figure appears to reverse, or in the case of the semantic paradigm, the previous scene description is replaced by a new scene description, which is then replaced by the old description, etc., indefinitely.

CHAPTER I

INTRODUCTION

This dissertation concerns itself with picture recognition by computer. "Picture recognition" involves a concern with both classification and description.

"Picture classification" is the assignment of labels to objects in a picture so that the labels correspond to the class names of objects designated by the objects (signs) in the picture. Classification is hierarchical if named parts of the picture are repeatedly grouped, and each new group is given a new label, i.e., is further classified.

"Picture description" is a characterization of the picture in terms of relations which exist between classified objects in the picture.

In this chapter, existing paradigms for picture recognition are presented, and comments made as to their limitations. It is found that these paradigms are inadequate for many picture recognition tasks. In Chapter II, the inadequacies are examined in more detail. A more powerful paradigm for picture recognition which might overcome some of the inadequacies of existing paradigms has been suggested by several authors. These suggestions, compiled in Chapter III, provided the motivation for the research results reported in the following chapters.

The primary results of the research consist of (1) the development of a new and more powerful paradigm for picture recognition, called the "semantic paradigm," and (2) an evaluation of the paradigm through its application to several interesting and complex picture recognition problems.

1.1 The Receptor/Categorizer Paradigm

The receptor/categorizer model (RCM) provided the first paradigm for picture recognition. It was first explicitly presented by Marill and Gree (1960). In this paradigm, only picture classification (not description) is possible. The RCM paradigm may be characterized as follows.

1. A picture is first reduced to a "feature" set by the receptor.

2. The feature set is then assigned to one of a finite number of patterns by the categorizer.

3. A "reject" class is often used when the input cannot be assigned to a known pattern.

4. The principal technique for assignment is to treat the feature (measurement) as a point in multidimensional space. The categorizer then partitions the space based on similarity and distance functions.

1.2 Limitations of the Receptor/Categorizer Paradigm

The model is of little use when analyzing complex pictures (scenes) where the structure and interrelationships among the picture components are important factors. This point was well illustrated by Shaw (1968) as follows.

> . . . consider the one-dimensional pattern recognition task required of a programming language translator. . . . One general purpose of the syntax analysis phase of the compiler is to categorize an input program into one of two mutually exclusive classes--the class of syntactically correct programs and its compliment. Theoretically, one could envision a receptor which produces a feature vector from an input program; the categorizer then would determine in which of the two possible subspaces the feature vector lies. While this could be done in principle, it is never considered seriously because of the complexities involved; for example, what is the feature set for a program? Even if this approach were practically feasible for program classification, it would not produce the most important by-product of a successful analysis, i.e., a description of the structure of the input program.

The need for a new approach was established by Narasimhan (1962).

> Categorization, clearly, is only one aspect of the recognition problem; not the whole of it by any means. It is our contention that the aim of any recognition procedure should not be merely to arrive at a "Yes", "No", "Don't know" decision but to produce a structured description of the input picture. Perhaps a good part of this confusion about aims might have been avoided if, historically, the problem had been posed as not one of pattern <u>recognition</u> but of pattern <u>analysis</u> and <u>description</u>.

As a result, a new approach to picture recognition was developed, which is now known as the syntactic approach, as described in the following section.

1.3 The Syntactic Paradigm

The basic idea of syntactic (grammar-based, linguistic) methods is to extend the notions of syntax and semantics to n dimensions and then apply an adaptation of the techniques of natural and artificial language processing. Shaw (1968) provided the paradigm within which syntactic techniques can be described. This paradigm is known as the "linguistic model for picture processing," and is described by Shaw as follows.

The linguistic model for picture processing is comprised of two parts:

1. a general model within which pictures may be described. . . and
2. an approach to the analysis. . .of pictures based directly on their descriptions.

The description D of a picture α will consist of two parts--a primitive or terminal symbol description T and a hierarchic description H; this can be written $D(\alpha) = (T(\alpha),H(\alpha))$. T and H, in turn each have a syntactical or structural component T_s and H_s, and a semantic or value component T_v and H_v. I.e.,

$$T(\alpha) = (T_s(\alpha)\ ,\ T_v(\alpha))$$

$$H(\alpha) = (H_s(\alpha)\ ,\ H_v(\alpha))\ .$$

$T_s(\alpha)$ describes the elementary component classes or primitives in α and their relationship to one another. $T_v(\alpha)$ gives the values or meaning of the components of α. It should be noted that the primitives in $T_s(\alpha)$ denote classes; define $\mathcal{P}(T_s)$ as the set of all pictures with primitive structure T_s.

Example 1:

Let ℓ name the set of all straight line segments. Let c name the set of all circles. ℓ and c are picture primitives. Let 0 denote the geometric relationship of intersection.

Then, if a picture α contains a line segment intersecting a circle, $T_s(\alpha) = \ell\ 0\ c$; $T_v(\alpha)$ could be the list $(v_\ell\ ,\ v_c)$, where v_ℓ is the pair of endpoints of ℓ and v_c is the center coordinates and radius of c. $\mathcal{P}\ (\ell\ 0\ c)$ is the set of all pictures consisting of a line segment intersecting a circle.

Consider a set of rules of grammar $\mathcal{G}$ generating a language $\mathcal{L}(\mathcal{G})$ whose "sentences" are primitive structural descriptions. Then, $\mathcal{G}$ is said to describe the picture class

$$\mathcal{P}_{\mathcal{G}} = \bigcup_{T_s \in \mathcal{L}(\mathcal{G})} \mathcal{P}(T_s).$$

For a given picture $\alpha \in \mathcal{P}_{\mathcal{G}}$, $H_s(\alpha)$ is the ordered set of rules of $\mathcal{G}$ that were used to generate $T_s(\alpha)$; that is $H_s(\alpha)$ is the "linguistic" structure or parse of $T_s(\alpha)$ according to $\mathcal{G}$.

A one-to-one correspondence exists between the elements of a set of semantic or interpretation rules $\mathcal{A}$ and the elements of $\mathcal{G}$. $H_v(\alpha)$ is defined as the result of obeying the corresponding semantic rule for each rule of $\mathcal{G}$ used in $H_s(\alpha)$.

It is important to note that the grammar must be capable of generating primitive structural descriptions of all pictures being considered. No restrictions are made on the form of any of the components of D. A final point is the essential difference between primitive and hierarchic descriptions; the "meaning" of a picture is expressed by both. Thus, several grammars may be used to generate the same class of primitive descriptions, but the hierarchic description of a picture and hence its meaning may be different for different grammars. Even more generally, the same picture class may be described by totally different primitive and hierarchic descriptions; the intended interpretation of the picture dictates its description.

With the description model, the solution to the picture analysis problem can now be formulated:

1. The elementary components or primitives which may appear in a class of pictures are named and given a meaning.
2. The picture class is described by a generative grammar $\mathcal{G}$ and associated semantics $\mathcal{S}$.
3. A given picture α is then analyzed by parsing it according to $\mathcal{G}$ and $\mathcal{S}$ to obtain its description $D(\alpha)$; that is, $\mathcal{G}$ and $\mathcal{S}$ are used explicitly to direct the analysis.

Thus, the spirit of syntactic picture recognition is to produce a structured description of an input picture. By structured description is meant a hierarchical description in which (1) parts of the picture are repeatedly grouped and classified and (2) relations existing between the parts are exhibited.

The syntactic approach to picture recognition can be further characterized by observing that the following situations are generally encountered:

(1) A small set of well-defined primitives is used.

(2) A small number of relations is used, consisting of simple locally observable concatenation or juxtaposition relations.

(3) Any picture parts having the same shape are generally given the same class name, although an analysis of the context in which the part is found would enable a human to give it a less generic class name.

(4) Large and interesting classes of objects are difficult or often impossible to describe because of the necessity of enumerating the many possible hierarchical descriptions.

Since 1962, contributions by Evans (1968, 1969), Kirsch (1964, 1968), Ledley (1966), Shaw (1968), Guzmán (1968, 1971), Anderson (1967), and Narasimhan (1969) have served to firmly establish the techniques and methodology of the field. Recent works by Pavlidis (1972), Carlucci (1972), Simon et al. (1972), Swain and Fu (1972), Narasimhan and Reddy (1971), Uhr (1971), Watanabe (1971), and Clowes (1969, 1971) have extended the basic techniques and suggested new avenues for improvement. An excellent review of the syntactic approach to picture recognition can be found in Miller and Shaw (1968).

1.4 Limitations of the Syntactic Paradigm

Picture recognition techniques based on the syntactic paradigm have several serious limitations. In particular, problems involving (1) ambiguity of shape, (2) non-pictorial paraphrase, (3) non-ideal data, and (4) multistability in perception are, in general, characteristic of those cases in which techniques based on the syntactic paradigm are found to be inadequate. In Chapter II, these problems are examined in some detail, and the underlying causes for the failure of the syntactic paradigm to apply are examined.

1.5 Other Techniques for Picture Recognition

Picture recognition techniques which do not fall within the RCM or syntactic paradigms may be said to be ad hoc and heuristic in nature.

The result is, that for each problem situation to which these techniques are applied, the solution to the problem must be purely ad hoc since no general model has been found for these techniques which could serve as a paradigm for picture recognition.

The obvious disadvantage of these techniques is that they are often difficult to implement, difficult to modify for small changes in the recognition requirements, and almost impossible to modify for application to dissimilar tasks.

The advantage of using these techniques is that they can be applied to certain problems which could not be solved within the RCM or syntactic paradigms.

1.6 Statement of the Problem

The problem is to develop a general model or paradigm for picture recognition which overcomes some of the limitations of existing paradigms. The solution to the problem is approached in the following manner.

1. Determine the inadequacies of syntactic picture recognition techniques. Chapter II discusses in detail these inadequacies and attempts to pinpoint their underlying causes.

2. Establish the objectives of a new paradigm which would overcome some of these inadequacies. Chapter III discusses a new "semantic" approach to picture recognition based on these objectives.

3. Develop a paradigm for semantic picture recognition based upon the knowledge obtained in step one, and the goals established in step two. Chapter IV describes this new paradigm.

4. Evaluate the paradigm through its application to several interesting picture recognition problems that cannot be solved using existing paradigms. Chapters V-VIII describe experiments in which the semantic paradigm is successfully applied to a variety of difficult recognition problems.

CHAPTER II

LIMITATIONS OF SYNTACTIC PICTURE RECOGNITION TECHNIQUES

In this chapter, there are described four related picture recognition problems which characterize some of those cases in which techniques based on the syntactic paradigm are found to be inadequate. The problems deal with the following subjects: (1) ambiguity of shape, (2) non-pictorial paraphrase, (3) non-ideal picture data, and (4) multistability in perception.

The underlying causes for the failure of the syntactic paradigm to apply to these problems are examined, and are found to be related to (1) the use of extensional (versus intensional) class descriptions, and (2) an inability to utilize contextual information.

2.1 Ambiguity of Shape

This problem arises quite often in picture recognition in determining to which of several possible classes an object belongs when different objects in a picture may assume identical shapes. Thus, for example, a circle may represent an eye, the sun, a ball, etc., and its class membership can only be resolved through an analysis of the context (scene) in which the object is found.

In a similar manner, we see that the same problem arises when objects in the same class, having different shapes, cannot be recognized as belonging to that class out of context.

Consider the problem of defining the class of all "hats." Any syntactic approach to describing this class would be limited to the use of some primitive shapes, and a set of local "spatial" concatenation or juxtaposition relations defined on these primitive shapes. It is clear that if the syntactic approach is to be adequate for all recognition purposes, the following must hold: (1) every picture of a hat, given sufficient context, should be recognizable as a hat, and (2) every picture of objects recognizable as other than hats should not be recognized as hats.

To show that existing syntactic techniques are inadequate, let us first consider the object illustrated in Figure 1. Almost everyone would agree that the object represents a hat. Now consider the objects illustrated in Figure 2. These objects might look like hats, but one cannot say for sure. Indeed, it is impossible to make a decision based only on the information given. In Figure 3 it is seen that the objects in Figure 2 may be recognized as objects other than hats. In Figure 4 it is seen that the objects in Figure 2 may be also recognized as hats. In general, syntactic techniques make no use of context, and a decision would be made based upon only the information in Figure 2. Quite obviously, in this case, syntactic techniques would fail, since even man cannot make an error free classification decision without context such as that found in Figures 3 and 4. Thus, the two conditions previously established, namely (1) that every picture of a hat, given sufficient context, should be recognizable as a hat, and (2) that every picture of an object recognizable as other than a hat should not be recognized as a hat, are not met by existing syntactic techniques.

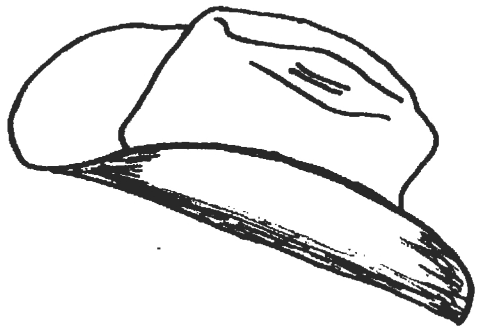


Figure 1. Object Recognizable as "Hat" Because of Its Shape

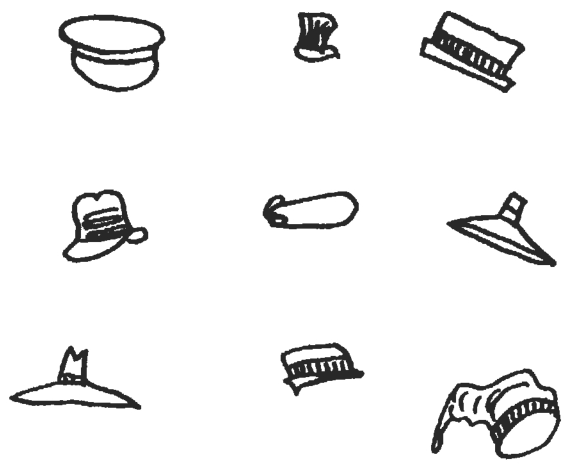


Figure 2. Ambiguous Objects

Figure 3. Ambiguous Objects Recognizable as Other Than "Hats"

Figure 4. Ambiguous Objects Recognizable as "Hats"

Clearly, what is missing from the syntactic approach is the ability to describe classes not only in terms of shape, but also in terms of allowable and compatible context. Indeed, it appears that given sufficient context, an object may be recognized almost regardless of its shape. From this point of view, syntactic techniques are seen as not only inadequate but burdensome as well, in that they may force one to consider information that is irrelevant for many recognition objectives.

Perhaps the point that syntactic techniques are inadequate need not be debated further, since almost everyone is in agreement with Guzmán (1971) when he says that,

> . . . in practice . . . [syntactic] models specify (only) geometrical and topological constraints on the shape, size, and so on, of an object to be classified as hat, while the human constraints refer also to the use of the object (that is, covers the head): for instance, in the tropics big leaves may be used as hats in rain. Not being so smart to take these things into account . . . [syntactic] . . . models . . .

will necessarily continue to have severe limitations on their applicability (brackets added). Thus, researchers must accept the fact that at best, syntactic techniques will be found adequate for recognizing only "most" or "many" objects in some classes.

2.2 Non-pictorial Paraphrase: A Problem of Context Analysis

Clowes (1969) introduced the concept of non-pictorial paraphrase for objects having substantially different structures, but sharing a class name. For example, in Figure 5 each object is a member of the class "five ohm resistance circuit," but there seems to be little common structure by which the class can be described. Indeed, an attempt to describe

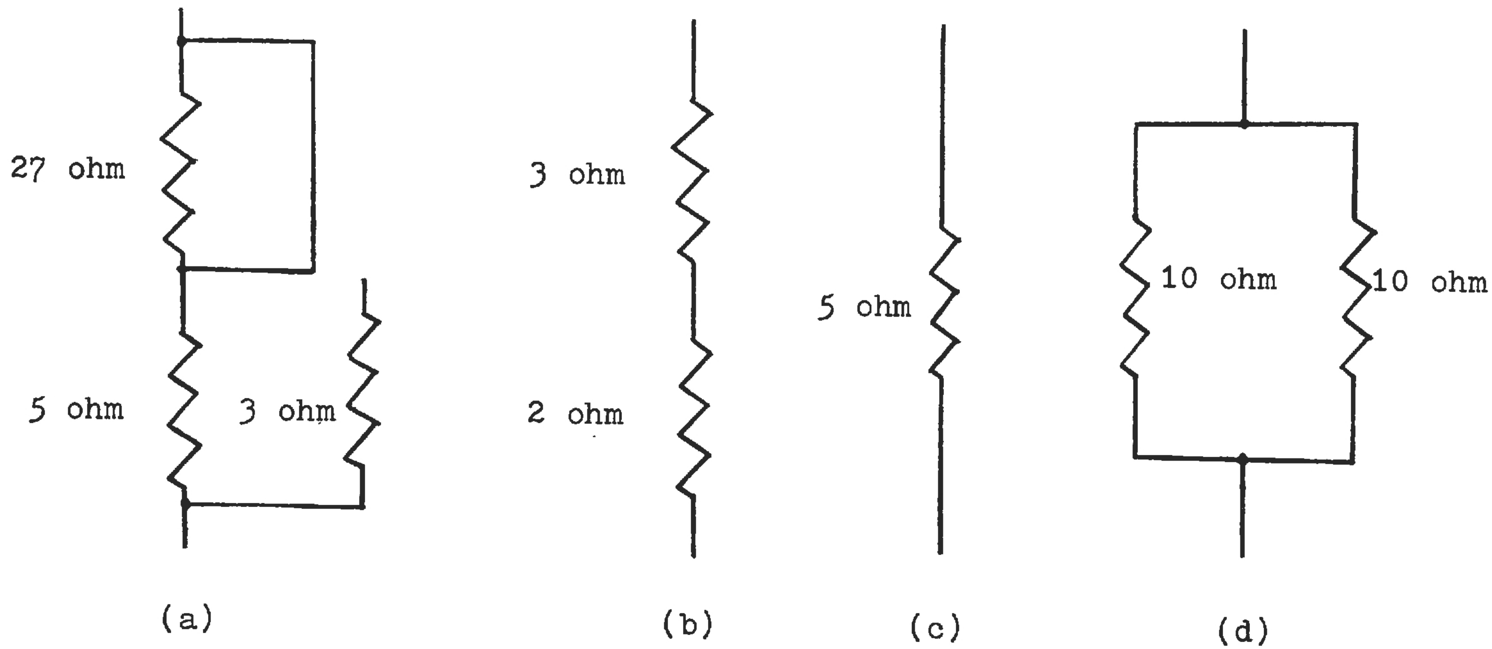


Figure 5. Non-pictorial Paraphrase

the class of "five ohm resistance circuits" using existing syntactic techniques would be an enormous task if at all possible. The "pictorial syntax" of such a class necessarily assumes an infinite number of configurations since any number of resistors having different resistances could be combined in series, parallel, or even in short or open circuits, to produce the required five ohm total resistance.

It is interesting to note that all members of this class could be recognized totally out of context. Contextual information, however, must be thought of as being essential to the analysis of the parts of the circuits. In the above case, an interpreter analyzes each circuit in the class by breaking it down into more and more elementary serial/parallel circuits. The relations between all of the other parts of the circuit to one part thus play the role of context for that part.

It appears obvious that for cases in which non-pictorial paraphrase arises, syntactic techniques will often be found to be inadequate. This is true whether the problem is viewed generally as one of describing the infinite variety of admissible shapes or more specifically as one of capturing the "internal" contextual constraints and requirements.

Existing techniques simply contain no mechanism for representing and using the kinds of knowledge required for analyzing most contextual information in pictures. In the above example, such knowledge might be reflected in the following statements: (1) wires have negligible resistance, (2) resistance in series is additive, and (3) resistance in parallel follows the rule $1/R_{total} = \sum_i 1/R_i$.

For an isolated object such as the hat seen in Figure 1, the notion of non-pictorial paraphrase equally applies. This is true since the object can be recognized completely out of context and shares little common structure with other hats also recognizable out of context. Thus, situations can be visualized in which the only difference between "hats" and "circuits" is that in circuits, the "meaningful" parts of the object are more easily recognized out of context.

2.3 Non-ideal Data

While it is a common practice to approach a difficult problem by first attacking a simplified form of it, it is important to look ahead to evaluate the potential applicability of a research result to the real problem. In the case of picture recognition, the "real" problem involves recognizing objects from a photograph or television scan, for example. The simplified version of the problem is manifested in the utilization of "artificial" pictures, or those pictures in which it is assumed that the real "grey level" picture has been ideally preprocessed into a perfect line drawing. Thus, the problems of noise, missing lines, incomplete lines, preprocessing induced anomalies, shading, segmentation of regions, etc., are minimized. Based on such a simplification, several formal picture grammars have been more or less successful in capturing the structure of pictures from a wide range of artificial problems. Such techniques, unfortunately, consistently fail to carry over naturally to real picture recognition problems. Similarly, the picture processing techniques which have been developed for operating on real pictures have proven to be quite useless for capturing the picture structure which is necessary for description and classification.

Underlying the problem of applying techniques from one of these domains to the other is the question concerning the possible range of generality of a primitive or set of primitives. The small sets of primitives utilized for analyzing artificial pictures are not applicable to real pictures or grey level picture data. Firschein and Fischler (1971) have pointed out this problem quite clearly in their discussion on primitives for grey level pictures.

> The term "grey level" picture is used here to denote a picture having gradations of grey tone or color. An obvious approach for dealing with such pictures is to convert them into a line drawing representation. Although this approach can sometimes meet with success if the conversion procedure is designed for a particular class of objects [see his footnote below], there are several reasons why a single line drawing representation turns out to have very restricted utility. First of all, consider a complex grey level photograph converted to a line drawing using the Shaw primitives (for example). The derived line drawing will be an unresolvable mass of line segments, with no obvious indication as to how the lines are related to one another. Therefore, due to the impractically large number of parsings which must be examined, a syntax-based, simple primitive, linear string representation is not feasible.
>
> However, there is a more basic objection to line drawing reduction: a great many distinct applications-oriented line drawings can often be derived from the same photograph. A practical example . . . taken from a report on ecological surveys from space . . . [shows that] three different line representations . . . [showing] (A) . . . the geology, . . . (B) . . . the sand dunes and ridges, and . . . (c) . . . vegetation and intermittent streams, . . . [were] derived . . . from the [same] photograph. . . . In order to derive such line drawings from the photograph, it is necessary to approach the photograph with different "sets", and the basic primitives looked for by each observer may be quite different.
>
> In summary, the classic concept of a primitive is that of a well-defined entity, detectable by pattern recognition or similar means. However, to deal with grey scale pictures, one may be forced to abandon the concept of primitives defined by shape, and instead consider "primitives" as regions defined by properties such as texture or color . . . [these] . . . methods . . . are not, as yet, far advanced. (Brackets added.)

(his footnote) See Kelly's thesis (1970) dealing with visual identification of people by computer, for an example of how a variety of techniques can be used to analyze a grey level scene if the procedures have been designed for a particular subject area.

2.4 Multistability in Perception

Since a picture serves as a model for a set of scenes, one might expect that more than one interpretation of the picture is possible or even desirable. This is especially evident in those cases in which humans encounter multistability in perception as described below.

The fact that techniques based on the syntactic paradigm output only one interpretation of a picture suggests that they are too deterministic or rigid. As will be seen, this is a result of the use of extensional class descriptions.

The problem of multistability in perception has been described by Attneave (1971) as follows:

> Pictures and geometric figures that spontaneously change in appearance have a peculiar fascination. A classic example is the line drawing of a transparent cube . . . [see Figure 6]. . . . When you first look at the cube, one of its faces seems to be at the front and the other at the back. Then if you look steadily at the drawing for a while, it will suddenly reverse in depth and what was the back face is now the front one. The two orientations will alternate spontaneously; sometimes one is seen, sometimes the other, but never both at once. (Brackets added.)

For the researcher interested in simulating human cognitive processes, an investigation into the psychological aspects of human visual perception (such as multistability) would hopefully lead to the development of an appropriate theory. Unfortunately, there exists no precise methodology for unifying the many disjoint clues and insights gained from such an investigation. The limits of introspection demand the use of a precise empirical methodology for studying human visual perception. Since the research reported herein falls within the domain of artificial

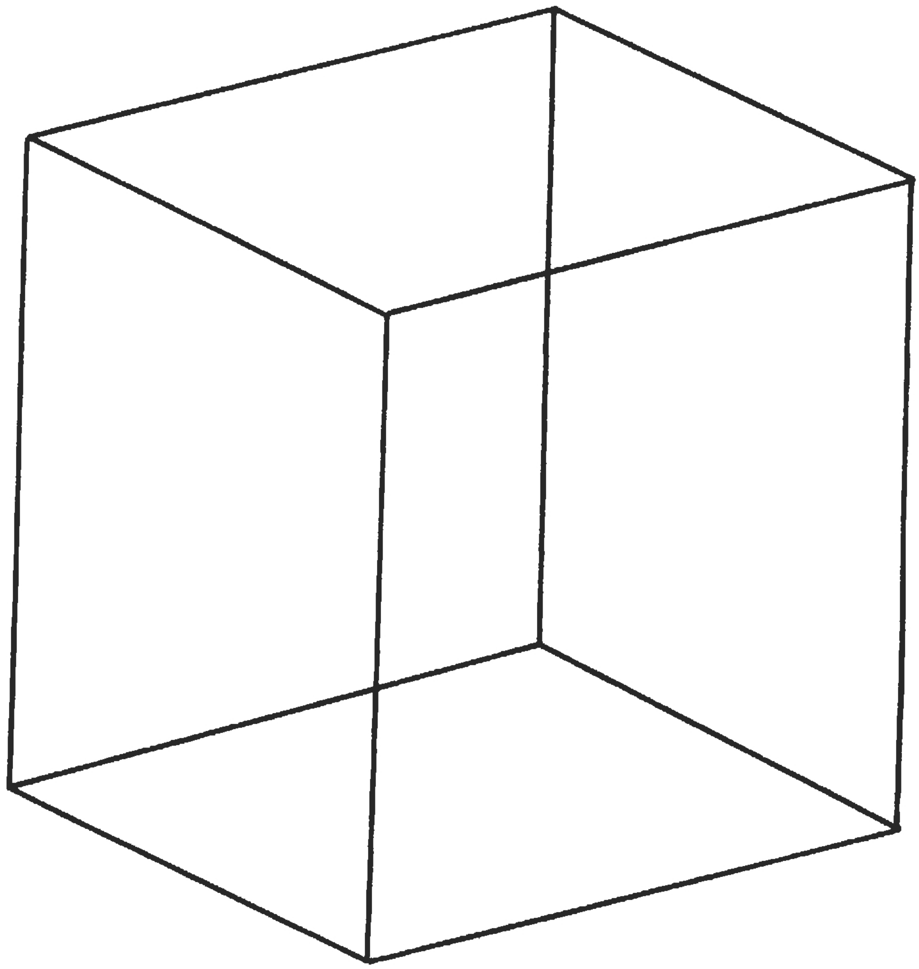


Figure 6. Necker Cube

intelligence (AI), there is no need to be concerned with modeling the human perceptual system. Rather, the concern here is in producing an artificial intelligence with a useful perceptual capability (which may or may not share the characteristics of human perceptual systems). The close relation between these two scientific domains, however, demands that each use whatever results produced by the other to the best advantage.

In the case of AI research, the carry-over is fairly straightforward. If a theory pertaining to human perception is sufficiently well-defined to permit a computer scientist to construct a useful program based on that theory, and such program is found to be superior to existing programs, then the new technique can be added to the AI field.

In the case of simulation of human perceptual behavior, the carry-over is less straightforward. If an AI technique is found to duplicate in some aspect the behavior of a human being, then that technique can be used as a methodological tool for further investigation into the psychological aspects of perception. If the use of the AI technique serves to explain to the psychologist other aspects of human perception, then the possibility of constructing another useful theory for AI research exists.

The problem of multistability in perception is investigated since it will be seen that the semantic paradigm exhibits an analogous kind of behavior which might be of great interest to researchers attempting to simulate human perceptual processes. Chapter VIII reports the results of an experiment which successfully exhibits this interesting aspect of the behavior of the semantic paradigm when applied to the problem of trying to recognize the object in a picture of a reversing cube.

2.5 Underlying Causes of the Limitations of Syntactic Picture Recognition Techniques

In this section the underlying causes for the failure of the syntactic paradigm to apply to the above problems are examined. In particular, it is found that (1) the use of extensional class descriptions, and (2) an inability to utilize contextual information are prominent factors.

2.5.1 The Use of Extensional Class Descriptions

In the following paragraphs, it is explained that because syntactic techniques use extensional class descriptions, the above problems are necessarily encountered. It will first be necessary to discuss approaches to class and object description.

First, let us agree that a description of an object in terms of a set of primitives (such as intensity values for points in an array) and relations (such as juxtaposition relations in an array) does not serve any useful purpose unless the primitives can be grouped in a "meaningful" manner. By "meaningful," it is meant that some primitives and the relations between them can be thought of collectively as forming a "named object." If this process is repeated and a hierarchy of "named objects" can be constructed, the utility of such a description, in general, increases. The assignment of a name in such a description is identical to <u>classifying</u> a part of the object or scene.

Therefore, syntactic techniques do involve classification. (We do not mean to imply, however, that classification always consists of choosing <u>one</u> name from a mutually exclusive set.) Classification of an object can only be made by "matching" a description of the object to a <u>class description</u>. There are two philosophical approaches to class description:

extensional and intensional. Although it is difficult to make a clear distinction between the two, we can make the following observations.

An <u>extensional description</u> of a class is simply a set of descriptions of <u>all</u> individual objects in the class. Assuming that one would wish to admit an infinite number of possible different objects into a class, one might be lead to believe that extensional descriptions must be infinite and hence inadequate for picture recognition tasks. However, it is true that an infinite number of objects can be described in a finite manner if a (e.g., phrase structure) grammar with recursion is utilized for generating descriptions. One accepting the extensional approach to class description must then use this recursive power to describe the infinite set of objects in a class. That this is the approach taken by most researchers in the syntactic picture recognition field is often overlooked. The unfortunate result of using such an approach is that the only way the class description can be expanded to include objects sharing little common structure is through the addition of alternatives in the productions of the grammar. Clearly, as classes of objects are constructed in which non-pictorial paraphrase arises, the number of alternatives in the productions of the grammar approaches infinity.

That extensional descriptions are inadequate for the description of some classes should come as little surprise since intuitively it appears impossible to describe any complex class of objects in terms of its possible arrangement of <u>components</u>.

Any alternative approach to picture recognition which we would hope could overcome the limitations induced by extensional class descriptions, must evidently utilize intensional descriptions of classes. An <u>intensional class description</u> is a (possibly unary) set of essential properties which are peculiar to the objects denoted by the class name. As it will be argued in the following, the <u>nature</u> of these essential properties is of utmost importance for picture recognition tasks. The use of intensional class descriptions somewhat corresponds to the "procedure-based description" approach, as Firschein and Fischler (1971) have explained:

> In contrast to the unique description of a picture obtained using grammar-based [extensional] description, it is possible to use a system which is capable of generating a large number of descriptions for the same picture. This allows one, at least conceptually, to choose from a set of descriptions the one most appropriate for a given <u>context</u>. The situation of single versus multiple parsing is analogous to the unique parsing of a programming expression compared to the multiple parsing possible for a natural language expression.
>
> If the multiple description approach is taken, it is not practical to generate all possible descriptions and choose the most appropriate, due to the large number of possible combinations. Instead, the type and order of procedures are controlled to (hopefully) obtain the most desirable description, and the approach can therefore be called "procedure-based" description [intensional description]. Parameters controlling the type and order of procedures are chosen based on <u>semantic</u> or "high level" considerations, and a description is generated based on the parameters chosen. (Brackets and emphasis added.)

2.5.2 The Inability to Utilize Contextual Information

As it has been shown, some objects can be recognized almost regardless of their shape through an analysis of context. Indeed, some objects can <u>only</u> be distinguished from others through the use of contextual information. Since syntactic techniques make little or no use of contextual information, this is clearly one cause for the limitations encountered with their use. It appears that the inability to utilize contextual information is a natural consequence of the use of extensional class descriptions.

CHAPTER III

SEMANTIC-BASED PICTURE RECOGNITION

The limitations of syntactic picture recognition techniques have been noted by other investigators. A "semantic-based" approach has been suggested as an alternative to the syntactic approach; however, it has only been stated in rather general and obscure terms. A semantic picture recognition technique can be defined as any technique for picture recognition which overcomes some of the limitations of syntactic picture recognition techniques. Obviously, some ad hoc and heuristic techniques fall in this category, notably those of Kelly (1970), Guzman (1968), and Barrow and Popplestone (1971). None of these techniques, however, is based upon a general model. That is to say, there exists no paradigm for semantic picture recognition.

The need for, and the nature of, a paradigm for semantic picture recognition is reflected in the following comments by Minsky, Clowes, and Lipkin et al.*

Minsky (1968), in commenting on a position by Kirsch (1968) that circuit diagrams have a syntax, has stated that,

> It's risky to maintain that there is such a thing as a syntax of circuit diagrams independent of other things. Certainly there are conventions-e.g., if a thing isn't made of lines, it's not a circuit diagram. But this is not a matter of syntax of a formal language with a phrase structure grammar or something like that; this is a matter of knowledge that the program has to have about the subject matter that it's going to deal with.

*Chandrasekaran and Kanal (1971) have compiled a similar survey.

I don't think you can make a distinction between syntactic rules and knowledge about electromagnetic theory. You can say that there's no point in a wire going to a part of the circuit unless at least one other wire comes back. So there must be at least two wires between any two things in certain contexts. You can call that a syntactic fact about the way you draw circuits; or you can call it a fundamental physical fact . . . there's a limitation; if this thing really represents a physical system, then because of the conservation of charge (or something or other) there has to be a return path for current.

I'm not saying: "That's a good distinction, but it doesn't work." In fact, it probably does work well, up to a point. I think it can do some harm, as has happened in the case of mechanical translation. Some of these attempts had the form: "The proper description of language is a set of 'rules'; these are to be written down, and there is to be a parsing program that uses the rules to parse the expressions in the language." That is a good approximation for simple cases, but it's a very dangerous idea to fix on so firmly that you cannot think about it in any other way.

Certainly linguistic problems are hard, but linguists have made it very much harder for themselves. They've got themselves into situations where they can't analyze simple things said in two sentences.

Clowes (1969) has stated that,

The interpretation of pictures involves a further syntactic structure of the event being pictorially depicted which constitutes the semantics of the picture . . .

. . . we should regard the semantics of pictures as concerned with the exhibition of relationships of a non-pictorial kind. These relationships must be regarded as defining the character of the "physical world." Pictures express events in this physical world.

We have alluded earlier to the multiplicity of possible picture-syntactic descriptions assignable to some pictorial expression. Such a multiplicity would present an impossible task for a syntax-directed parser. If however we assume that the parser is directed not only by a picture grammar i.e., by what pictorial relationships are computable by the parser, but also by the necessity to recover well-formed event descriptions in some world (this world being manifested pictorially), then it seems plausible that the variety of potentially assignable descriptions can be dramatically reduced . . .

. . . it seems appropriate to regard English sentences, nuclear equations and bubble chamber photographs all as being mappings of a single nuclear event and to that extent descriptive of it not of one another.

> If this viewpoint is accepted then it follows that the development of a machine capable of relating English sentences or equations of one kind or another, to diagrams or photographs requires the provision of at least three syntax specifications. These would be of English sentence structure, pictorial relationships, and the relational structure of the event (nuclear event, electrical circuit, etc.) being referred to in these various languages. In addition, of course, rules for mapping from structures in one domain to those in another are required. It is these mapping rules together with a syntax specification of some "object world" which are now the most pressing needs in the development of such machines.

Lipkin et al.(1966) envisioned a system consisting of three parts: (1) a linguistic (natural language) grammar, (2) an iconic grammar, and (3) a means for binding the two together. In order to construct a system according to this design, we would need to (1) formalize the notions of syntax and semantics for natural language, (2) formalize the notions of syntax and semantics for an iconic grammar, and (3) define a transformation between the two grammars.

These notions indicate the "direction" toward a new paradigm for semantic picture recognition, but give us little guidance for realizing such. In the following chapter, such a paradigm is developed based upon the above comments, and the limitations of syntactic techniques discussed in Chapter II.

CHAPTER IV

A PARADIGM FOR SEMANTIC PICTURE RECOGNITION

In this chapter, a new paradigm for picture recognition is presented. The development of the paradigm was motivated by the comments presented in Chapter III. The limitations of the syntactic paradigm, outlined in Chapter II, have conceptually been avoided through (1) the explicit use of contextual information and (2) the implicit utilization of intensional class descriptions.

4.1 Glossary of Terms

I an information processor.

Q_i a quality; $Q_i = \{ q_{i_1}, \ldots \}$, where q_{i_j} is the jth property in Q_i.

$\bar{Q}_I$ the quality domain for I.

$\bar{Q}'_I$ set of primitive qualities for I, determined by I's sensory and preprocessing facilities.

R_i a relation.

$\bar{R}_I$ the relation domain for I.

$\bar{R}'_I$ set of primitive relations for I, determined by I's sensory and preprocessing facilities.

U the universe.

E the environment.

S a scene.

T a region.

O_k^t	label assigned to an object occupying region k at time t.
$D(O_k^t)$	I's description of O_k^t.
$\hat{O}(S_t)$	set of descriptions of all primitive objects in a picture of scene S at time t.
$\bar{R}_I'(t)$	set of primitive relations defined by I at time t.
$PSD^t(S)$	primitive state description of scene S at time t.
PD(S)	primitive description of scene S.
$O(S_t)$	set of descriptions of all objects in scene S at time t.
$R_I^*(t)$	set of relations defined by I at time t.
$NPSD^t(S)$	non-primitive state description of scene S at time t.
NPD(S)	non-primitive description of scene S at time t.
Z	the picture recognition process

4.2 Preliminary Definitions and Discussion

4.2.1 Assumption

There exists an object I which has certain properties peculiar to intelligence. In particular, I has the properties of perception, cognition, and comprehension, which capabilities we shall assume to be intuitively understood. We may say that I is an information processor.

4.2.2 Postulate

I comprehends the concepts of property and object.

4.2.3 Definition

A quality Q_i for I is a set of mutually exclusive properties. I.e., $Q_i = \{q_{i_1}, q_{i_2}, \ldots\}$. Note, however, that not every set of properties is a quality. For example, the set of properties {red, happy, hot, small} would not be a quality.

The following sets of properties, however, would be typical qualities for an information processor:

Q_1 = "color" = {blue, green, red, yellow, ...}.

Q_2 = "temperature" = {cold, lukewarm, warm, hot}.

Q_3 = "shape" = {square, triangular, circular, ...}.

Q_4 = "length" = {short, medium, long}.

4.2.4 Definition

A finite set of qualities perceived or cognized by I is said to be the quality domain of I and is denoted by $\overline{Q}_I$. I.e., $\overline{Q}_I = \{Q_1,\ldots,Q_n\}$.

4.2.5 Definition

A relation R_i on the non-empty sets $X_1, X_2, \ldots$ is defined as a subset of the Cartesian product on these sets. I.e., $R_i \subseteq \bigtimes_{j\in J} X_j$.

4.2.6 Definition

A finite set of relations perceived or cognized by I is said to be the relation domain of I and is denoted by $\overline{R}_I$. I.e., $\overline{R}_I = \{R_1,\ldots,R_n\}$.

4.2.7 Proposition

I is an object for I.

4.2.8 Definition

The set U of all objects for I is called the universe of I. Therefore, I is an element of U.

4.2.9 Definition

Let $\{R_1,\ldots,R_n\}$ be a set of relations defined on U, and let $V \in U$ be an object for I. The environment of V in U is the relational system $E = \langle U\text{-}V; \widehat{R}_1,\ldots,\widehat{R}_n \rangle$, where $\widehat{R}_1,\ldots,\widehat{R}_n$ are relations induced by $R_1,\ldots,R_n$ on the set U-V.

4.2.10 Definition

The restriction of the environment E (i.e. of the relational system E) on the set of objects $\hat{S}$ which are simultaneously observable is called the scene, and is denoted by S.

4.2.11 Definition

The restriction of S to some subset $\hat{T} \subset \hat{S}$ is called a region of S. We shall denote it by T.

4.2.12 Discussion

In the following the notions of an object and a description of an object for information processor I are formalized. There will be a deviation from the traditional definition of an object as a relation on some subset of the quality domain, for reasons discussed below.

4.2.13 Assumption

An object occupies only one region and is said to exist at only one time instant.

4.2.14 Discussion

The notion of an object here corresponds closely with what has traditionally been defined as a state of an object. The motivation for dealing with the notion of an object as such comes from the following two observations. (1) It is often difficult to determine when an object ceases to become that object. Consider a pencil. If we move it from one location to another it still appears to be a pencil. However, if we sharpen it until it is very short, it loses its ability to function as a writing implement, is it still a pencil? If it is used to hold a door open is it still a pencil, or is it a door stop? When it is burnt to ashes it certainly is no longer a pencil. The argument is, then, that unless we can say when one state of an object no longer "belongs" to that object, we should treat each state separately as an object. (2) Associated with the above observation is the well known fact that infants and young children often consider the "same" object (such as its mother) at a different time or place to be a different object. Since the magnitude of the recognition problem has evaded previous attempts to formulate a suitable paradigm, it appears to be reasonable to model the recognition capabilities of a small child.

In the above paragraph it was stated that the notion of an object used here will correspond closely with what has traditionally been defined as a <u>state</u> of an object. Thus where traditionally, an object is described as a subset of the Cartesian product of a set of qualities, a state is merely an element in that set. That is, an object state is an <u>ordered</u> n-tuple of properties. Let us consider an example. Assume that Q_1 = temperature = $\{$low, high$\}$, and Q_2 = elevation = $\{$low, high$\}$. If the traditional approach to object description is taken, an object might be defined as a subset of $Q_1 \times Q_2$. Thus, each element of this set corresponds to a state of the object. For example, if an object were at one time low in temperature and high in elevation, one element of the set describing the object might be the ordered couple (low, high). We note that ordering is essential to distinguish this object state from an object low in elevation and high in temperature, for example. Although in general it would not be the case, we find that for reasons of simplifying the programming task, it is convenient to dispense with explicit ordering by assigning a unique label to every property. Thus, the label q_{i_j} is a unique label assigned to the jth property in the quality Q_i. In the above case then, the state of the object low in temperature and high in elevation could be the (unordered) set $\{q_{1_1}, q_{2_2}\}$ or $\{q_{2_2}, q_{1_1}\}$, instead of the ordered couple (low, high). This leads us to the following definition.

4.2.15 Definition

I's description D of object O, occupying region k at time t, is defined to be a set of mutually exclusive and uniquely labeled properties, all of which are perceived by I to hold for the object(s) occupying the region k. I.e., $D(O_k^t) = \{q_{i_j}, \ldots\}$.

4.2.16 Discussion

The community to which this research is directed is concerned with the actual construction of computer systems for performing specified recognition tasks. In the design of such systems, it is very often the case that the scene is not observable by the system. Rather, sensory and preprocessing peripheral devices transmit to the computer, data which represents the scene. For example, a TV camera might scan the scene, and an analog-to-digital converter might then digitize the scan data and transmit to the computer an array of values. Each element of the array corresponds to a region in the scene, and each value of the array corresponds to the property or properties which hold for the object(s) occupying the region, such as average light intensity. In more sophisticated systems, the data received by the computer might be the result of extensive preprocessing, in which case the data transmitted would specify a particular region, and the property or properties which hold. For example, a region might be specified by a set of coordinates, and the property might be "square" or "circular" from the quality labeled shape.

In any case, the scene is represented as a <u>finite</u> number of regions. The data transmitted will be referred to as a <u>picture</u> of the scene.

<u>4.2.17 Assumption</u>

It is assumed that there are data given to I which permit I to determine, for any region in the picture at any time, whether or not any property of a quality holds for the object(s) occupying the region, for some set of qualities. We shall denote this set of qualities by $\bar{Q}'_I$. Clearly, $\bar{Q}'_I \subseteq \bar{Q}_I$.

<u>4.2.18 Definition</u>

Quality Q_i is said to be a <u>primitive quality</u> if and only if $Q_i \in \bar{Q}'_I$. Property q_{i_j} is said to be a <u>primitive property</u> if and only if Q_i is a primitive quality.

<u>4.2.19 Definition</u>

An object is said to be a <u>primitive object</u> at time t if every $q_{i_j} \in D(O_k^t)$ is a primitive property.

<u>4.2.20 Definition</u>

The set of descriptions of all primitive objects in a picture of scene S at time t is denoted by $\hat{O}(S_t)$.

<u>4.2.21 Postulate</u>

Information processor I can perceive (compute) whether or not any n-place relation R_i holds between primitive objects occupying regions $k_1,\ldots,k_n$ for any time for some set of relations $\bar{R}'_I \subseteq \bar{R}_I$.

4.2.22 Discussion

It has been postulated that the information processor has the ability to perceive certain relations between primitive objects, if they hold, at any given time. The implication for computer implementation is that the computer program must have a procedure which, given input parameters specifying a primitive relation and an ordered set of regions (which the primitive objects occupy), can determine whether or not the relation holds over the given primitive objects. It is seen that such relations can be defined by I as follows.

Relation R_i is said to hold over primitive objects O_k^t and O_l^t if and only if the following hold:

(1) $q_{j_m} \in D(O_k^t)$ and $q_{j_n} \in D(O_l^t)$, and

(2) $(q_{j_m}, q_{j_n}) \in R_i$.

We may generalize to say that the n-place relation R_i holds over primitive objects $O_1^t,\ldots,O_n^t$ if and only if the following hold:

(1) $q_{1_x} \in D(O_1^t),\ldots,q_{n_y} \in D(O_n^t)$, and

(2) $(q_{1_x},\ldots,q_{n_y}) \in R_i$.

4.2.23 Definition

Relation R_i is said to be a primitive relation if and only if $R_i \in \bar{R}'_I$.

4.2.24 Discussion

Clearly, $\bar{Q}'_I$, and thus $\bar{R}'_I$ are dependent upon the sensory and preprocessing devices which transmit data to the information processor. In the construction of a computer program to solve a specific recognition task, the specification of $\bar{Q}'_I$ and $\bar{R}'_I$ is usually straightforward. In processing real grey level picture data, $\overline{Q}_I'$ usually consists of only a few simple primitive qualities such as average light intensity, or color, and relative location. $\overline{R}_I'$ usually consists of simple primitive relations which follow from $\overline{Q}_I'$, such as "__ has greater average light intensity than __," "__ is above __," etc.

4.2.25 Definition

A set of primitive relations defined by I over the primitive objects at time t is denoted by $\overline{R}_I'(t)$.

4.2.26 Definition

A primitive state description PSD of scene S at time $t \equiv_{def} PSD^t(S) = \langle \hat{O}(S_t) \; ; \; \overline{R}_I'(t) \rangle$.

As previously mentioned, we consider an object to exist at only one time instant. This rather unusual treatment of the concept of an object arises from our desire to avoid the classification aspect of recognition, and focus on the descriptive aspect. Since an "object" description changes with time, we find it convenient to treat each "object state" as a separate "object."

4.2.27 Definition

A primitive description PD of a scene is defined to be a time sequence of primitive state descriptions. I.e., $PD(S) = \{PSD^{t_1}(S), \ldots, PSD^{t_n}(S)\}$.

4.2.28 Discussion

A primitive state description can be thought of as the input into a computer program for scene analysis and description. This input contains all of the information that the information processor initially has at one time instant about the scene to be analyzed.

From a programming point of view, one can think of each element (primitive object) in the set $\hat{O}(S_t)$ as a list or one-dimensional array. The name of the list corresponds to the region associated with the object at time t. Each element of the list contains the name of a primitive property which holds for that object.

Similarly, each element in the set $\bar{R}_I'(t)$ can be thought of as an n-dimensional Boolean array representing the primitive n-place relation R_i over the set of primitive objects, at time t.

The notion of a primitive description in terms of a sequence of primitive state descriptions may be found useful for "robot" vision in which changes in a scene over time could provide essential information for the analysis of, and interaction with, objects in a scene.

4.2.29 Discussion

At this point, the information processor possesses only a minimum ability to perform scene analysis or picture recognition. I.e., the information processor can communicate only the fact that a few primitive relations hold between a few primitive objects. It is desired that the information processor be able to construct non-primitive descriptions, in order to be able to communicate the fact that not only primitive relations but also non-primitive relations hold between not only primitive objects but also non-primitive objects.

4.2.30 Definition

A set of descriptions of objects, not all of which are primitive, in scene S at time t is denoted by $O(S_t)$.

4.2.31 Definition

A set of relations, not all of which are primitive, is denoted by R_I^*, where $R_I^* \subseteq \bar{R}_I$.

4.2.32 Definition

A set of relations defined by I over the set of objects in a scene at time t is denoted by $R_I^*(t)$.

4.2.33 Definition

A non-primitive state description NPSD of a scene S at time $t \equiv_{def} NPSD^t(S) = \langle O(S_t) \; ; \; R_I^*(t) \rangle$.

4.2.34 Definition

A non-primitive description NPD of a scene S is defined to be a time sequence of non-primitive state descriptions. I.e., NPD(S) = $\{NPSD^{t}1(S), NPSD^{t}2(S), \ldots, NPSD^{t}n(S)\}$.

4.2.35 Definition

Picture recognition is defined to be a process Z of constructing a non-primitive description of a scene from a primitive description (derived from a picture) of a scene. I.e., Z: PD(S) → NPD(S).

4.2.36 Discussion

In the following section the mechanism which underlies the process Z is described. This mechanism involves the utilization of a set of "rules of inference" which allow non-primitive properties and relations to be inferred through an analysis of context (relations holding between objects).

4.3 Rules of Inference

Rules of inference may be thought of as "pieces" of knowledge resident in an information processor. It appears that such rules correspond to what Minsky (1968) was concerned with in his comments (see Chapter III) on the knowledge a program has to have about a subject matter it is going to deal with.

Through the use of inference, we will see that contextual information is explicitly utilized for the construction of scene descriptions.

In the following, definitions are given, and the paradigm for picture recognition is presented. Chapters V-VIII describe experiments where it is shown that the paradigm is easily applied to a variety of interesting and difficult picture recognition problems.

4.3.1 Definition

PROPERTY $(O_k^t\ ;\ q_{i_j})$ is a predicate whose truth value can be determined as follows:

TRUE if $q_{i_j} \in D(O_k^t)$.

FALSE if q_{i_j} is a primitive property, and not $q_{i_j} \in D(O_k^t)$.

FALSE if $q_{i_l} \in D(O_k^t)$, $l \neq j$

UNDETERMINED otherwise.

4.3.2 Discussion

PROPERTY may be thought of as a Boolean procedure which searches for the indicated property. If such a property is found to exist at time t for object(s) occupying region k, a value of TRUE is returned. If the indicated property is primitive and it is not found to hold, then the procedure returns a value of false since, by definition, I can determine if any primitive property holds. If another property in the same quality set is found to hold, then a value of FALSE is returned since the properties in any quality are, by definition, mutually exclusive. We find this necessary in order to return a truth value in those cases in which we wish to evaluate an expression, such as NOT (PROPERTY $(O_k^t ; q_{i_j}))$. If no truth value is found, the procedure, instead of returning a truth value, exits to the interpreter which applies the rules of inference.

4.3.3 Definition

RELATION $(R_i ; O_j^t,\ldots,O_k^t)$ is a predicate whose truth value can be determined as follows:

TRUE if the relation R_i holds between the objects occupying regions $j,\ldots,k$ at time t.

FALSE if the relation R_i is primitive and does not hold between primitive objects occupying regions $j,\ldots,k$ at time t.

UNDETERMINED otherwise.

4.3.4 Discussion

RELATION may be thought of as a Boolean procedure which searches for the indicated relation. If such a relation is found to exist, a value of TRUE is returned. If such a relation is primitive and is not found to hold between primitive objects, a value of FALSE is returned. If no truth value is found, the procedure, instead of returning a truth value, exits to the interpreter which applies the rules of inference.

4.3.5 Definition

An INQUIRY is defined to be a statement in the first-order predicate calculus whose only predicate variables are PROPERTY and RELATION.

4.3.6 Discussion

An INQUIRY is a statement which is used to determine whether or not a scene contains a certain configuration of objects in certain relations to each other. This amounts to being able to determine if an object in a scene exists in a certain context.

4.3.7 Definition

The ADD(q_{i_j} ; O_k^t) operator replaces $D(O_k^t)$ with $D(O_k^t) \cup \{q_{i_j}\}$.

4.3.8 Definition

The DELETE(q_{i_j} ; O_k^t) operator replaces $D(O_k^t)$ with $D(O_k^t) - \{q_{i_j}\}$. (The difference set.)

4.3.9 Definition

The COMBINE(O_m^t ; O_n^t) operator constructs a new object O_{m+n}^t from the objects occupying the (possibly disconnected) regions m and n. $D(O_{m+n}^t)$ is null, initially, until the ADD operator is applied.

4.3.10 Definition

The RELATE(R_i ; $O_j^t,\ldots,O_k^t$) operator replaces R_i with $R_i \cup \{(O_j^t,\ldots,O_k^t)\}$.

4.3.11 Definition

The UNRELATE(R_i ; $O_j^t,\ldots,O_k^t$) operator replaces R_i with $R_i - \{(O_j^t,\ldots,O_k^t)\}$.

4.3.12 Definition

We will denote by $\overline{O}$ the set of operators. I.e., $\overline{O}$ = {ADD, DELETE, COMBINE, RELATE, UNRELATE}.

4.3.13 Discussion

The operators in $\overline{O}$ may be thought of as procedures for operating on the data structures which contain information about the scene description.

4.3.14 Definition

A COMMAND is a statement of the form: DO BEGIN $c_1,\ldots,c_n$ END; where $c_i \in \overline{O}$.

4.3.15 Definition

A RULE OF INFERENCE is a conditional expression of the form: IF X THEN Y, where X is an INQUIRY, and Y is a COMMAND.

4.3.16 Discussion

Rules of inference permit the inference that certain non-primitive properties and non-primitive relations hold. Thus, they contain "knowledge of the world" which permits an information processor to utilize contextual information in the construction of non-primitive scene descriptions. Such rules, it is noted, are formulated in terms of relations and properties (rather than relations and objects) which allows knowledge of the world to be abstracted from particular settings.

In Chapters V-VIII, experiments are described in which the use of rules of inference is illustrated. It is shown that their use results in the satisfactory solution of some very interesting and difficult recognition problems.

4.4 The Paradigm for Semantic Picture Recognition

The paradigm for picture recognition can now be described as the following procedure.

(1) Define a domain $\overline{Q}_I$ of qualities for information processor I.

That is, define $\overline{Q}_I$ such that properties in the qualities denote interesting classes of objects which I wishes to recognize.

(2) Define a subset $\overline{Q}_I'$ of $\overline{Q}_I$ of primitive qualities. That is, based upon I's sensory and preprocessing facilities, determine which qualities in $\overline{Q}_I$ are primitive for I.

(3) Define a domain $\overline{R}_I$ of relations for information processor I. That is, define $\overline{R}_I$ such that relations in the domain are of interest to I in describing a scene.

(4) Define a subset $\overline{R}_I'$ of $\overline{R}_I$ of primitive relations. That is, based upon I's sensory and preprocessing facilities, determine which relations in $\overline{R}_I$ are primitive.

(5) Develop a set of rules of inference which represents I's body of knowledge regarding the environment.

(6) Establish a procedure for applying the rules of inference to a primitive description (derived from a picture) of a scene. The result of successive applications of the rules is a non-primitive description of the scene.

4.5 Remarks

The semantic paradigm is found to conceptually overcome both of the causes of the limitations of syntactic techniques discussed in Chapter II. This follows since (1) explicit use is made of contextual information in the form of rules of inference and (2) the utilization of intensional class descriptions is implicit in the use of the semantic paradigm. This follows since classes are implicitly described with respect to a single essential property which is inferred through the application of the rules of inference. Thus, cases of (1) non-pictorial paraphrase, (2) ambiguity of shape, and (3) non-ideal data, for which extensional descriptions are often inadequate, are now subject to analysis through the use of the semantic paradigm. In so far as syntactic techniques may be characterized in terms of the semantic paradigm, one might make the following observations.

Contextual information, which is exploited by the semantic technique through the application of rules of inference, is not utilized by syntactic techniques. This follows since each production in a typical syntactic phrase structure (context-free) grammar $X \rightarrow Y + Z$ has the effect of combining the objects in regions Y and Z to form a new object occupying region Y + Z. In terms of rules of inference, such a production could be represented as "IF $\exists$ y,zPROPERTY("z",z)AND PROPERTY("y",y)THEN ADD("x", COMBINE(y,z))." Phrase structure rules, however, cannot permit a new property to be associated with a given region based on the context in which that particular region is found. Such a case is illustrated in Figure 7(a), in the following chapter, for example, where the property "floats in air" is to be inferred of the spherical object. That syntactic techniques are found to be inadequate for utilizing contextual information should be of little surprise. An important "exception" to this case is found in Guzmán (1971). In this paper a tree search method is proposed (although not implemented) for using contextual information to solve the following problem.

> Given that a set (a scene) is formed by components that locally (by their shape) are ambiguous, because they can have one of several possible values (a circle = sun, ball, eye, hole) or meanings, can we make use of <u>context information</u> . . . stated in the form of models (a generalized description of an object or a class of objects, with certain parameters left unspecified) in order to assign to each component a value such that the whole set (scene) is consistent or makes global sense?

Guzmán, however, proposes the use of only some of the properties in $\overline{Q}'_I$ and some of the relations in $\overline{R}'_I$ to capture contextual information for further analysis of objects in the scene. Again, we see that there is a reluctance to deal with properties other than "shape," and relations other than "juxtaposition." The point is well illustrated, however, that utilization of contextual information is essential for many recognition objectives.

<u>Class descriptions</u> for syntactic techniques are <u>extensional</u>, in that each class is in effect defined explicitly in terms of all admissible primitive state descriptions. As a result, syntactic techniques, in general, are generative. I.e., the mechanism which permits recognition to be made (e.g., the parsing of a phrase structure grammar) is also used to generate all possible objects belonging to the class. Unfortunately, the generative power of syntactic techniques applies to only "geometrically defined" classes of objects.

While a hierarchy of objects may be derived from a syntactic parse of a picture, it is evident that each object is really nothing more than a set of primitive relations between primitive objects. Hence, each object in the hierarchy may be thought of as a primitive object itself.

The inability of syntactic techniques to construct and utilize true non-primitive objects indicates a fundamental weakness in the syntactic approach since it cannot handle naturally the "architecture of complexity" which Simon (1962) has argued must underlie all complex systems.

CHAPTER V

APPLICATION OF THE PARADIGM FOR A PROBLEM INVOLVING AMBIGUITY OF SHAPE

5.1 The Problem

The recognition problem described in this section is intended to be simple in order that the development and use of rules of inference can be clearly illustrated. It is interesting to note, however, that syntactic techniques do not have the mechanism for handling even this simple problem.

Let us suppose that an information processor I is observing a scene in some world W. W contains some of the objects of the real world, and is subject to certain physical phenomena found in the real world. W, therefore, is a "subset" of the real world as perceived by adults, hence it might be representative of the world as perceived by a small child.

The only physical phenomenon that acts on objects in W is gravity. That is to say, there are no winds, magnetic fields, etc.

Of the objects in W, the following are representative of common "interesting" objects which the information processor must be able to recognize. The singling out of these objects is arbitrary, as is the specification of the "goal," "desire," "duty," etc. of an information processor in W.

(1) helium-filled balloons.

(2) air-filled balloons.

(3) solid balls.

(4) egg-shaped lead blobs.

(5) fixed surfaces.

Figure 7 illustrates several possible events in world W. It is immediately obvious that certain objects cannot be recognized based only on their shape. For example, the circular object in Figure 7(a) is known to be a helium-filled balloon, not just because it is circular (since air-filled balloons and solid balls are also circular), but also because it appears to float in air (and air-filled balloons and solid balls do not). In a similar manner, it is easily seen that contextual information must be taken into consideration in order to resolve ambiguity in shapes for recognition of other interesting objects in world W.

The problem, then, is to apply the semantic paradigm to pictures of events in world W for the recognition of interesting objects.

5.2 The Semantic Paradigm Solution

A typical information processor in W might be expected to have the following definitions for qualities.

$\overline{Q}_I = \{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6\}$, where

Q_1 = shape = {"spherical," "egg-shaped," "flat"}.

Q_2 = buoyancy = {sinks-in-air, floats-in-air}.

Q_3 = weight = {heavy, light}.

Q_4 = deformability = {compressible, rigid}.

Q_5 = internal stress = {significant stress, insignificant stress}.

Q_6 = function = {helium-filled balloon, air-filled balloon, solid ball, egg-shaped lead blob, fixed surface}.

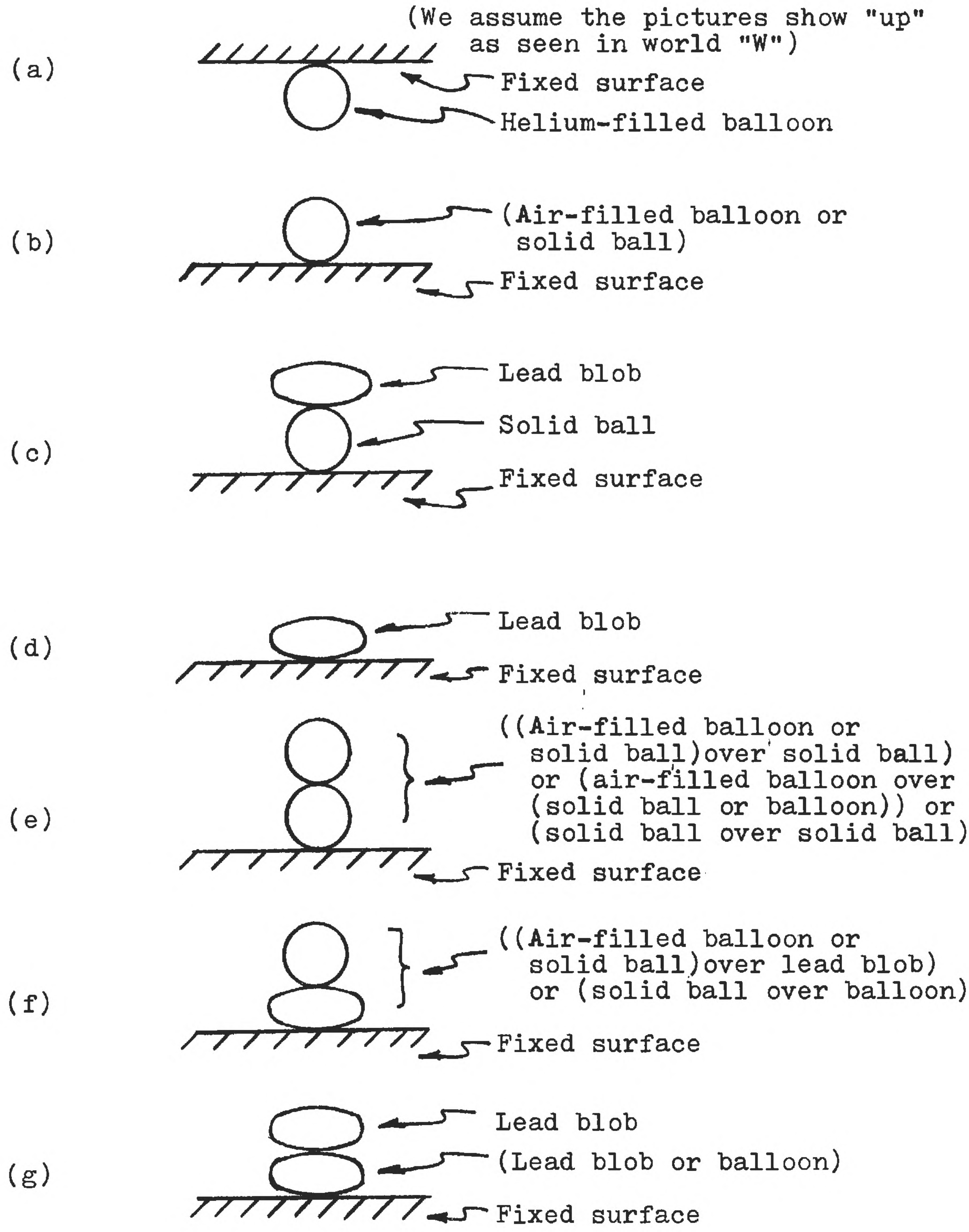


Figure 7. Pictures of World W

We shall assume that I's sensory and preprocessing facilities are such that the following set of qualities can be said to be primitive.

$\overline{Q}'_I = \{Q_1\}$.

We shall assume that I can perceive only the following kinds of relations between objects.

$\overline{R}_I = \{R_1, R_2, R_3, R_4\}$, where

R_1 = "__ is above __."

R_2 = "__ touches __."

R_3 = "__ supports __."

R_4 = "__ restrains __."

We shall also assume that I's sensory and preprocessing facilities are such that the following set of relations can be said to be primitive.

$\overline{R}'_I = \{R_1, R_2\}$.

5.2.1 The Primitive Scene Description

Before developing the rules of inference, we first give an example of a primitive scene description to which the rules would be applied. Consider the event depicted in Figure 7(c). Since the set of primitive qualities consists only of the quality labeled shape, the object descriptions in the primitive scene description would consist of only a single property, i.e., the shape of the object. In addition, we find that since we are dealing with a "snapshot" of an event in W, there will be only one primitive state description in the primitive scene description. If we denote the region occupied by the oblong object by "o," and the region occupied by the circular object by "c," and the region occupied by the flat object by "f," we find the following primitive scene description.

$$PD(S) = \{PSD^{t_1}(S)\}, \text{ where}$$

$$PSD^{t_1}(S) = \langle \hat{O}(S_{t_1}); \bar{R}'_I(t_1) \rangle, \text{ where}$$

$$\hat{O}(S_{t_1}) = \{ \text{"o"} = \{\text{"egg-shaped"}\}, \text{"c"} = \{\text{"spherical"}\}, \text{"f"} = \{\text{"flat"}\} \}, \text{ and}$$

$$\bar{R}'_I(t_1) = \{R_1, R_2\}, \text{ where}$$

$$R_1 = \{(\text{"o"}, \text{"c"}), (\text{"o"}, \text{"f"}), (\text{"c"}, \text{"f"})\}, \text{ and}$$

$$R_2 = \{(\text{"o"}, \text{"c"}), (\text{"c"}, \text{"o"}), (\text{"c"}, \text{"f"}), (\text{"f"}, \text{"c"})\}.$$

5.2.2 Constructing the Rules of Inference

In terms of the properties and relations in $\bar{Q}_I$ and $\bar{R}_I$, one would expect the experienced information processor to have constructed the following (intensional) descriptions of objects which have "interesting" properties.

(1a) {helium-filled balloon, "spherical," floats-in-air, compressible, insignificant stress}

(1b) {helium-filled balloon, "egg-shaped," floats-in-air, compressible, significant stress}

(2a) {air-filled balloon, "spherical," sinks-in-air, light, compressible, insignificant stress}

(2b) {air-filled balloon, "egg-shaped," sinks-in-air, light, compressible, significant stress}

(3a) {solid ball, "spherical," sinks-in-air, heavy, rigid, significant stress}

(3b) {solid ball, "spherical," sinks-in-air, heavy, rigid, insignificant stress}

(4a) {egg-shaped lead blob, "egg-shaped," sinks-in-air, heavy, rigid, significant stress}

(4b) {egg-shaped lead blob, "egg-shaped," sinks-in-air, heavy, rigid, insignificant stress

(5a) {fixed surface, "flat," rigid, significant stress}

(5b) {fixed surface, "flat," rigid, insignificant stress}

We now observe that such object descriptions provide more information than is necessary to <u>distinguish</u> one kind of interesting object from another. For example, anything that has the property "flat" in W is known immediately to be a fixed surface. One's knowledge of the possible objects having the property "fixed surface" would permit the inference immediately that "if x has the property 'fixed-surface' then x also has the property 'rigid'." In the case of fixed surfaces in this world, however, there is no need to infer this information for recognition purposes. For balloons, on the other hand, the shape may be identical to lead blobs, and recognition involves the inference of some other properties that could serve to distinguish the two. For example, if we could infer that an egg-shaped object has insignificant internal stress, we would know that it is a lead blob, since balloons are deformed into egg-like shapes only when they are supporting a significant weight (and hence have significant internal stress).

For information processor I, the following sets of properties (in addition to the interesting properties) are sufficient for distinguishing the interesting objects. (The following sets are determined by merely searching for all smallest subsets of the possible descriptions of the interesting objects which are unique for the class.)

Helium-filled balloons: {floats-in-air}.

Air-filled balloons: {light}.

Solid balls: {"spherical," significant stress}, or

{"spherical," heavy}, or

{"spherical," rigid}.

Egg-shaped lead blobs: {"egg-shaped," heavy}, or

{"egg-shaped," rigid}, or

{"egg-shaped," insignificant stress}.

Fixed surfaces: {"flat"}.

These sets of "sufficient" properties can now be used to formulate the first rules of inference. (Each of these rules permits an interesting property to be added to the description of an object. That object may then be said to be a member of the class denoted by that interesting property.)

The rules of inference that reflect the "sufficient" properties for inferring interesting properties are as follows: (An articulated description precedes the formal rule).

RULE 1: "If an object floats-in-air, then it is a helium-filled balloon."

IF $\exists$ x PROPERTY(x; floats-in-air) THEN ADD(helium-filled balloon; x).

RULE 2: "If an object is light, then it is an air-filled balloon."

IF $\exists$ x PROPERTY(x; light) THEN ADD (air-filled balloon; x).

RULE 3A: "If an object is spherical and under significant stress, then it is a solid ball."

IF $\exists$ x PROPERTY(x; "spherical") AND PROPERTY(x; significant stress) THEN ADD(solid ball; x).

RULE 3B: "If an object is spherical and heavy, then it is a solid ball."

IF $\exists$ x PROPERTY(x; "spherical") AND PROPERTY(x; heavy) THEN ADD(solid ball; x).

RULE 3C: "If an object is spherical and rigid, then it is a solid ball."

IF $\exists$ x PROPERTY(x; "spherical") AND PROPERTY(x; rigid) THEN ADD(solid ball; x).

RULE 4A: "If an object is egg-shaped and heavy, then it is an egg-shaped lead blob."

IF $\exists$ x PROPERTY(x; "egg-shaped") AND PROPERTY(x; heavy) THEN ADD(egg-shaped lead blob; x).

RULE 4B: "If an object is egg-shaped and rigid, then it is an egg-shaped lead blob."

IF $\exists$ x PROPERTY(x; "egg-shaped") AND PROPERTY(x; rigid) THEN ADD(egg-shaped lead blob; x).

RULE 4C: "If an object is egg-shaped and is under insignificant stress, then it is an egg-shaped lead blob."

IF $\exists$ x PROPERTY(x; "egg-shaped") AND PROPERTY(x; insignificant stress) THEN ADD(egg-shaped lead blob; x).

RULE 5: "If an object is flat, then it is a fixed surface."

IF $\exists$ x PROPERTY(x; flat) THEN ADD(fixed surface; x).

Similarly, rules of inference can be formulated to reflect the "invariant" properties of the interesting objects.

RULE 6: "If an object is a helium-filled balloon, then it floats-in-air and is compressible."

IF $\exists$ x PROPERTY(x; helium-filled balloon) THEN BEGIN ADD(floats-in-air; x); ADD(compressible; x); END.

RULE 7: "If an object is an air-filled balloon, then it is light, compressible, and sinks-in-air."

IF $\exists$ x PROPERTY(x; air-filled balloon) THEN BEGIN ADD(light; x); ADD(compressible; x); ADD(sinks-in-air; x); END.

RULE 8: "If an object is a solid ball, then it is spherical, heavy, and rigid."

IF $\exists$ x PROPERTY(x; solid ball) THEN BEGIN ADD("spherical"; x); ADD(heavy; x); ADD(rigid; x); END.

RULE 9: "If an object is an egg-shaped lead blob, then it is egg-shaped, heavy, and rigid."

IF $\exists$ x PROPERTY(x; egg-shaped lead blob) THEN BEGIN ADD("egg-shaped"; x); ADD(heavy; x); ADD(rigid; x); END.

RULE 10: "If an object is a fixed surface, then it is flat and rigid."

IF $\exists$ x PROPERTY(x; fixed surface) THEN BEGIN ADD("flat"; x); ADD(rigid; x); END.

It is seen that, based on the above rules, recognition of balloons is <u>based on context alone</u>, whereas recognition of egg-shaped lead blobs is <u>based on context and shape</u>, and recognition of fixed surfaces is <u>based on shape alone</u>.

5.2.3 Additional Necessary Rules of Inference

We now know what properties must be discovered to infer interesting properties of objects in world W, and we know from what description such properties must be inferred (the primitive scene description with respect to $\overline{Q}'_I$ and $\overline{R}'_I$). Knowing this, we can construct the additional rules of inference necessary to perform recognition. Consider again the pictures of world W in Figure 7.

In Figure 7(a) we know that of all possible objects we might encounter, the spherical object is a helium-filled balloon. This is evident because we can infer that the object floats-in-air (since it is not supported from below, since it does not touch another object that it is above). Recall that objects belonging to the class of helium-filled balloons were adequately distinguished from other objects in W by the set of essential properties {floats-in-air}.

In Figure 7(b) we cannot make a positive decision since it is impossible to infer all of the essential properties which distinguish any interesting class. We might find that the spherical object could only be an air-filled balloon or a solid ball by recognizing (1) that it sinks-in-air and hence does not float-in-air (and hence is not a helium-filled balloon), (2) that it is spherical and hence not egg-shaped (and hence not a lead blob) and (3) that it is spherical and hence not flat (and hence not a fixed surface). Of course it is experience that tells one to construct a rule that infers that all fixed surfaces are flat, and all lead blobs are egg-shaped, etc. An encounter with a novel situation would result in erroneous recognition. If this error is brought to the attention of the information processor, then it would want to modify the rules of inference to reflect this new experience. In a similar manner, all of the other objects shown in Figure 7(c-g) can either be recognized or "narrowed down."

The rules of inference that permit such conclusions to be made are "inherent" in human information processors, however, they may be readily represented for the computer for world W as follows. As before, each rule is preceeded by an articulated description for ease of reading.

RULE 11: "If object x is not flat and it is not above and touching some other object y, then x must float-in-air."

IF $\exists$ x $\forall$ y - (RELATION(R_1;x,y) AND RELATION(R_2;x,y)) AND - PROPERTY(x; flat) THEN ADD(floats-in-air; x).

RULE 12: "If a spherical and compressible object y supports another object x then that other object x must be light."

IF $\exists$ x,y RELATION(R_3;y,x) AND PROPERTY(y; compressible) AND PROPERTY(y; "spherical") THEN ADD(light, x).

RULE 13: "If an object x supports a heavy object y, then x is under significant stress."

IF $\exists$ x,y RELATION(R_3;x,y) AND PROPERTY(y; heavy) THEN ADD(significant stress; x).

RULE 14: "If object x is the only object on top of a compressible and egg-shaped object y, then x must be heavy."

IF $\exists$ x,y RELATION(R_1;x,y) AND RELATION(R_2;x,y) AND $\forall$z $\neq$ x, -(RELATION(R_1;x,z) AND RELATION(R_2;z,x)) AND PROPERTY(y, compressible) AND PROPERTY(y; "egg-shaped") THEN ADD(heavy, x).

RULE 15: "If a spherical object x supports a heavy object y then x is rigid."

IF $\exists$ x,y RELATION(R_3;x,y) AND PROPERTY(y; heavy) AND PROPERTY(x; "spherical") THEN ADD(rigid, x).

RULE 16: "If there is no object above and touching egg-shaped object x then x is heavy, rigid, and under insignificant stress."

IF $\exists$ x PROPERTY(x; "egg-shaped") and $\forall$y -(RELATION (R_1;y,x) AND RELATION(R_2;y,x)) THEN BEGIN ADD(heavy; x); ADD(rigid; x); ADD(insignificant stress; x); END.

RULE 17: "If every object supported by object x is light, x is under insignificant stress."

IF $\exists$ x $\forall$y RELATION(R_3;x,y) IMPLIES PROPERTY(y; light) THEN ADD(insignificant stress; x).

RULE 18: "If an object is heavy or light, then it sinks-in-air."

IF $\exists$ x PROPERTY(x; heavy) OR PROPERTY(x; light) THEN ADD(sinks-in-air; x).

RULE 19: "If object x is above and touching some object y, and there is no object z above and touching x, then x sinks-in-air unless x is flat."

IF $\exists$ x,y RELATION(R_1;x,y) AND RELATION(R_2;x,y) AND -PROPERTY(x; flat) AND $\forall$z -(RELATION(R_1;z,x) AND RELATION(R_2;z,x) THEN ADD(sinks-in-air; x).

In the process of constructing the additional rules of inference, the relation "supports" was utilized, which may be inferred from the following rules of inference.

RULE 20: "If object y is not flat, and it is above and touching object x, then x supports y."

IF $\exists$ x,y RELATION(R_1;y,x) AND RELATION(R_2;y,x) AND -PROPERTY(y; flat) THEN RELATE(R_3;x,y)."

RULE 21: "If object y supports w, and object x supports y, then x supports w."

IF $\exists$ w,x,y RELATION(R_3;y,w) AND RELATION(R_3;x,y) THEN RELATE(R_3;x,w).

The rule of inference for inferring the relation "restrains" appears to be unnecessary, however, it could be formulated as follows:

RULE 22: "If object x floats-in-air and object y is above and touching x, then y restrains x."

IF $\exists$ x,y PROPERTY(x; floats-in-air) AND RELATION(R_1;y,x) AND RELATION(R_2;y,x) THEN RELATE(R_4;y,x).

5.2.4 The Application of the Rules of Inference

The application of rules of inference can in the simplest case consist of merely indexing through the entire set as many times as possible, i.e., until no more relations and properties may be inferred.

More elaborate techniques would involve going directly to rules of inference in which terms exist which were recently inferred (those rules being the most likely candidates for success).

Goal directed recognition techniques would serve to terminate and guide the selection and application of rules of inference as just those properties and relations necessary for determination of the goal property need be inferred.

The program for determining whether any particular quantified statement (i.e., INQUIRY) holds (and hence whether the COMMAND should be executed) could for maximum generality, be a first-order theorem proving program. For special purpose applications, however, ad hoc search programs may be more appropriate.

5.3 Results

The reader can easily verify that the rules of inference, applied to primitive scene descriptions of many events in world W, including those depicted in Figure 7, will produce, whenever possible, non-primitive scene descriptions in which the interesting objects are described.

5.3.1 Discussion

It is observed that rules of inference make explicit use of contextual information in the form of relations and properties rather than relations and objects. This permits an information processor to deal successfully with a large number of novel situations.

It is interesting to note that a variety of properties, relations, and rules of inference could have been developed for adequately representing the information processor's required body of knowledge for recognizing interesting objects in W. It appears that situations could be visualized in W for which a given set of rules would be inadequate, or result in false inferences. Such a situation would represent a novel situation for which the information processor is unprepared. The "consistency" and "completeness" of rules of inference may be a point of theoretical interest for a well-defined environment. From a pragmatic point of view, however, we can think of the "completeness" and "consistency" of a set of rules of inference as corresponding to the "awareness" or "experience" of the information processor that applies them.

A theory of learning based on the acquisition, refinement, and expansion of rules of inference appears to be a natural extension to the work described herein.

If each property and relation in a term in the command portion of a rule of inference is thought of as having attached to it a set of pointers, each pointing to a rule of inference utilizing such property or relation in its inquiry, then we can envision a "network." This network is then a structure for which knowledge of the world can be associated in such a way that (1) the discovery of certain properties and relations leads to the inference of other properties and relations, and (2) newly input rules can be put in relation to large quantities of previously stored information about the same things. A possible correspondence of such a network to Quillian's (1968) Semantic Memory network may be a point of great interest. Of special interest may be the relation between Quillian's (1969) work on the "Teachable Language Comprehender" and a theory of learning based on the acquisition, refinement, and expansion of a network of rules of inference.

CHAPTER VI

APPLICATION OF THE PARADIGM FOR A PROBLEM INVOLVING NON-PICTORIAL PARAPHRASE

6.1 The Problem

The particular problem chosen to represent a problem in non-pictorial paraphrase, for which syntactic techniques are often inadequate, involves the recognition of (the functions performed by) electrical control circuits, given schematic pictures of the circuits.

This problem is of especial interest because it is found to be simple in many ways. For example, these pictures are constructed as line drawings, they may be represented as graphs, there are no three-dimensional views or perspectives to consider, occlusion is explicitly represented, and the pictures are constructed out of a small set of simple well-defined primitives. Even with all of these characteristics, however, syntactic picture recognition techniques are found to be inadequate.

Consider the kinds of circuits illustrated in Figures 8-14. These pictures belong to the class of electrical control circuit schematics, which are composed for the most part of the primitive components illustrated in Figure 15.

In addition to the basic components, we usually find in a schematic near some components, alphanumerics which serve as identification names for the components. These identification names become important in

Figure 8. Normally-open Push Button Operates to Closed Position Energizing Relay

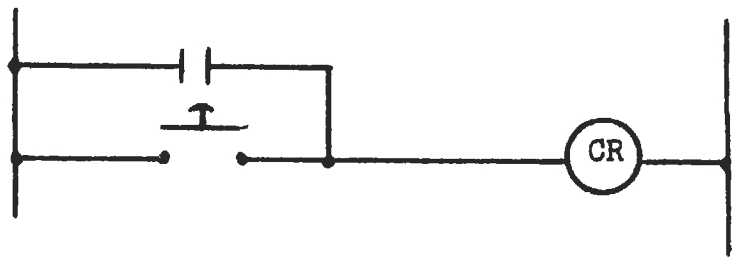


Figure 9. Normally-open Contact of Relay Used to Interlock Around Normally-open Push Button

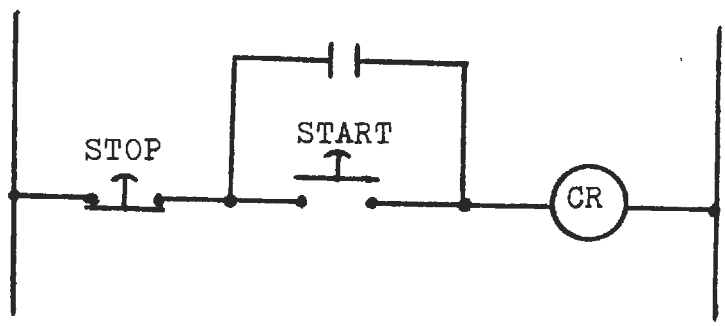


Figure 10. Normally-closed Push Button Used to De-energize Relay

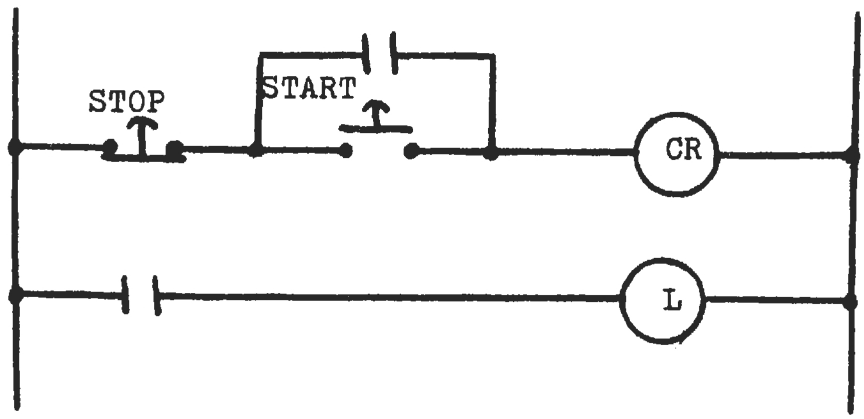


Figure 11. An Additional Normally-open Contact on Relay Used to Energize Load

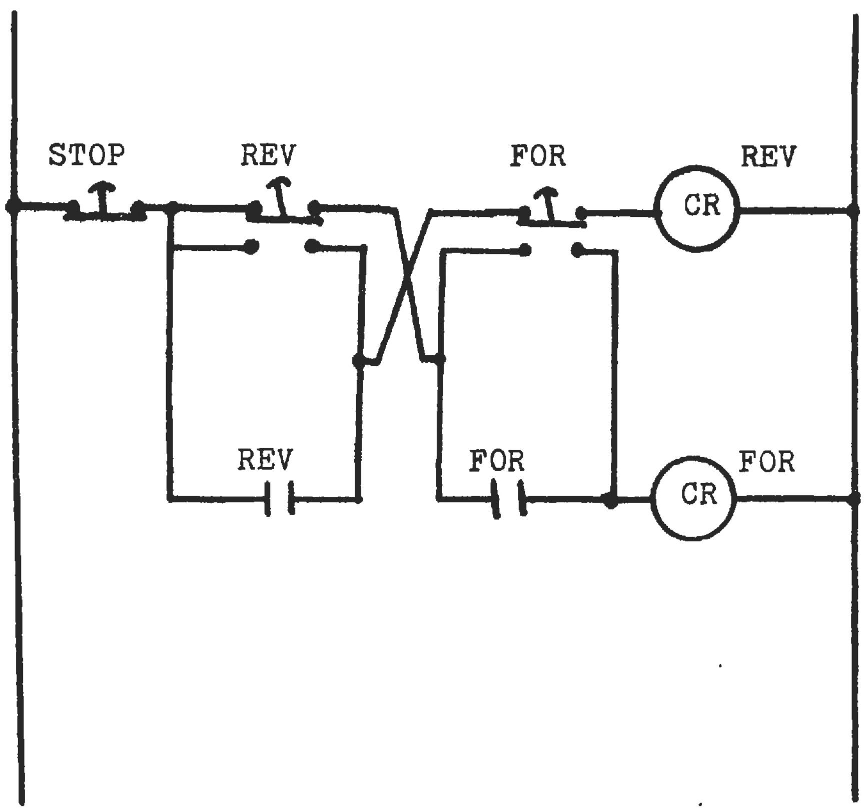


Figure 12. Cycle Start Circuit - Case 1

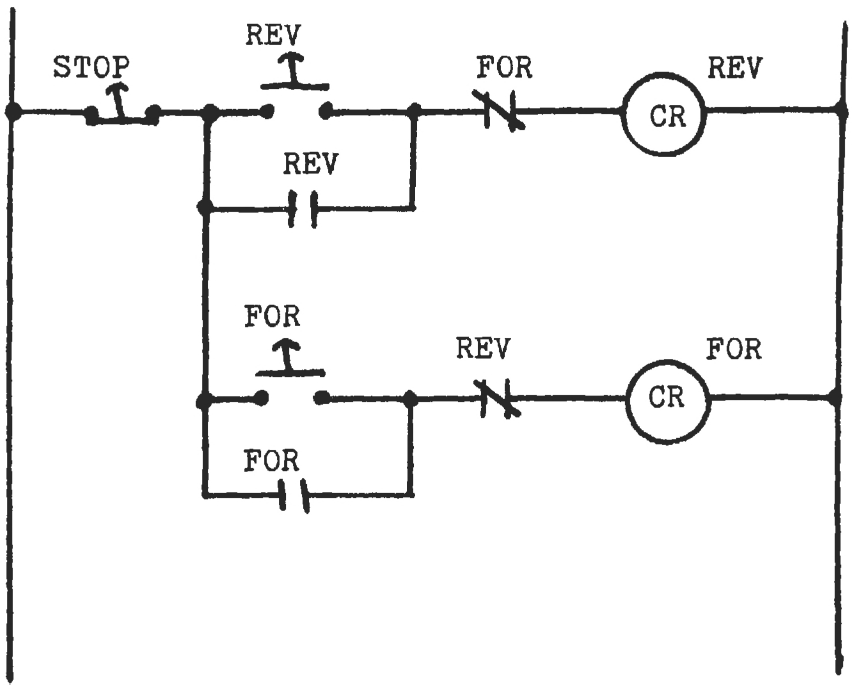


Figure 13. Cycle Start Circuit - Case 2

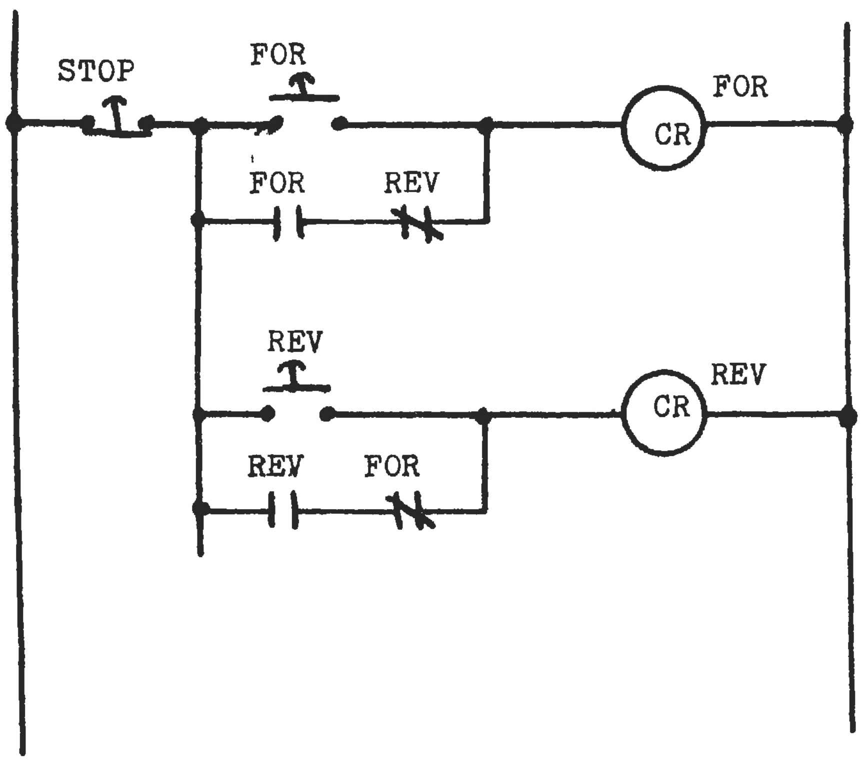


Figure 14. Cycle Start Circuit - Case 3

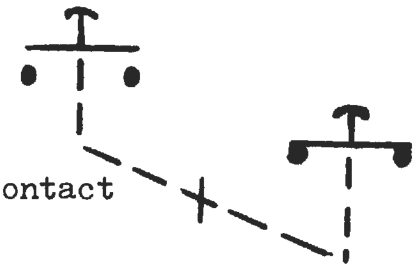
COMPONENT	SYMBOL	ABBREVIATION
Push Buttons		
Normally-open		NOPB
Normally-closed		NCPB
Maintained-contact		MCPB
Contacts		
Normally-open		NOCONT
Normally-closed		NCCONT
Control relay		CR
Timer	"TIMER"	TIMER

Figure 15. Components of Electrical Control Circuits

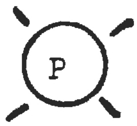
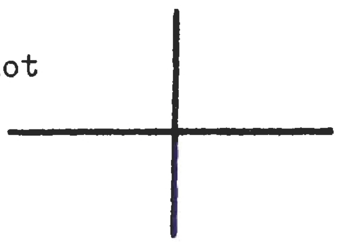
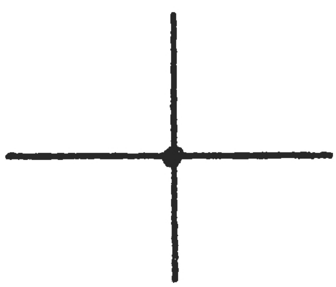
COMPONENTS	SYMBOL	ABBREVIATION
Switches		
Limit-switch, Normally-open		NOLS
Limit-switch, Normally-closed		NCLS
Load		LOAD
Pilot		PILOT
Connections		
Conductors not connected		(Not a component)
Conductors connected		CONNEC

Figure 15. (continued)

distinguishing identical component types.

The operation of the components might not be obvious. In the following paragraphs we indicate how each works.

Push buttons are switches that are held in a normal position by a spring. Thus they are always in a normal position unless a human being is pushing the button, holding it open, or closed as the case may be. A maintained contact push button is the only exception. This button is like a light switch in your house. Each time one operates it, it changes state from off to on, or from on to off.

Contacts are switches much like push buttons, except only control relays operate them. When the control relay associated with the contact is energized, the normally-open contacts are closed, and the normally-closed contacts are opened.

Control relays are coils of wire wound around metal bars. When a control relay is energized by passing current through its coil, the metal bar becomes magnetized, causing a nearby piece of metal to move. This physical movement causes the contacts associated with the control relay to change states. (Not all control relays work this way, but the point is illustrated.) When a control relay becomes de-energized, the magnetic field vanishes, causing the contacts to return to their normal states.

Timers become activated when the control relays to which they belong become energized. The purpose of the timer is to hold the contacts of that relay closed (or open) until a specified period of time has elapsed.

Limit switches, like contacts and push buttons, have a normal state (open or closed). Either springs or gravity may be used to achieve this normal state. Limit switches are opened or closed by physical things, like moving conveyors, falling parts, etc. Thus, while humans operate push buttons, and control relays operate contacts, most other things operate limit switches (Note: we could have included such switching devices as photo-cells, etc. Since they can be represented as limit switches just as well, we have omitted them here.)

Loads do work. They may represent motors, solenoids, etc.

Pilots are usually indicator lights, or very light loads. In general, anything energizable that is operated directly through a push button will be called a pilot. Only those things that draw little current, however, would ever be routinely operated directly by a push button for reasons of safety and economy.

Connections exist where two wires are joined so as to permit current to flow between them

Power sources usually are not explicitly indicated in an electric control circuit schematic, but are implicitly assumed to be connected to the left-most and right-most vertical lines (wires) in the drawing.

Classes of electrical control circuits are usually described in terms of a sequence of inputs and required outputs. Thus, a typical recognition task is to recognize all instances of circuits which perform the same function, such a function being described in terms of inputs (such as pushing buttons, closing limit switches, etc.) and required outputs (such as loads, pilots, etc. coming on or going off in response to the past sequence of inputs).

As a consequence, an infinite number of circuits can be constructed to perform a given function, requiring the syntactic recognition program to represent extensionally an infinite set of unique objects.

Although this problem was discussed in general in Chapter II, it is shown below that for the class of electrical control circuit schematics, the problem of non-pictorial paraphrase is especially evident. Consider the "simplest" class of electrical control circuits, e.g., the class in which "a load is always energized." Examples of circuits which perform this function (i.e., are members of this class) are illustrated in Figure 16.

We make the following observations, and leave it to the reader to convince himself that syntactic techniques, if not completely inadequate, would be so burdensome that a new approach to solving the recognition problem is required.

Obviously, an infinite number of very unique circuits perform the same function, and hence belong to the class.

It appears that the spatial arrangement of the wires in a circuit is quite arbitrary as long as the non-spatial relation "connected electrically" is preserved.

In some circuits, a load is energized only through the control of other parts of the circuit (for example, through the energization of a control relay which causes a contact to close enabling the load to become energized). Thus, we cannot ignore those parts of the circuit not directly "connected" to the load. This amounts to being able to use contextual information.

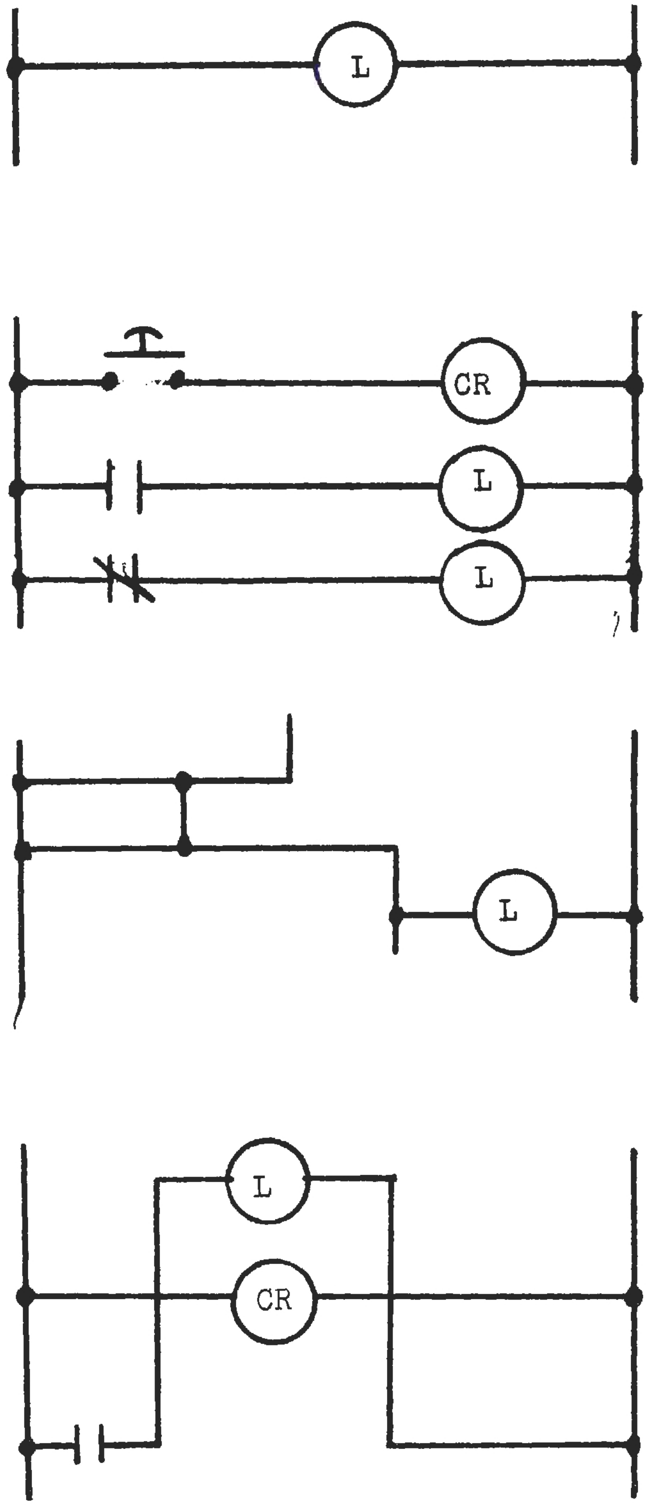


Figure 16. Examples of Circuits Belonging to the Class in Which a Load is Always Energized

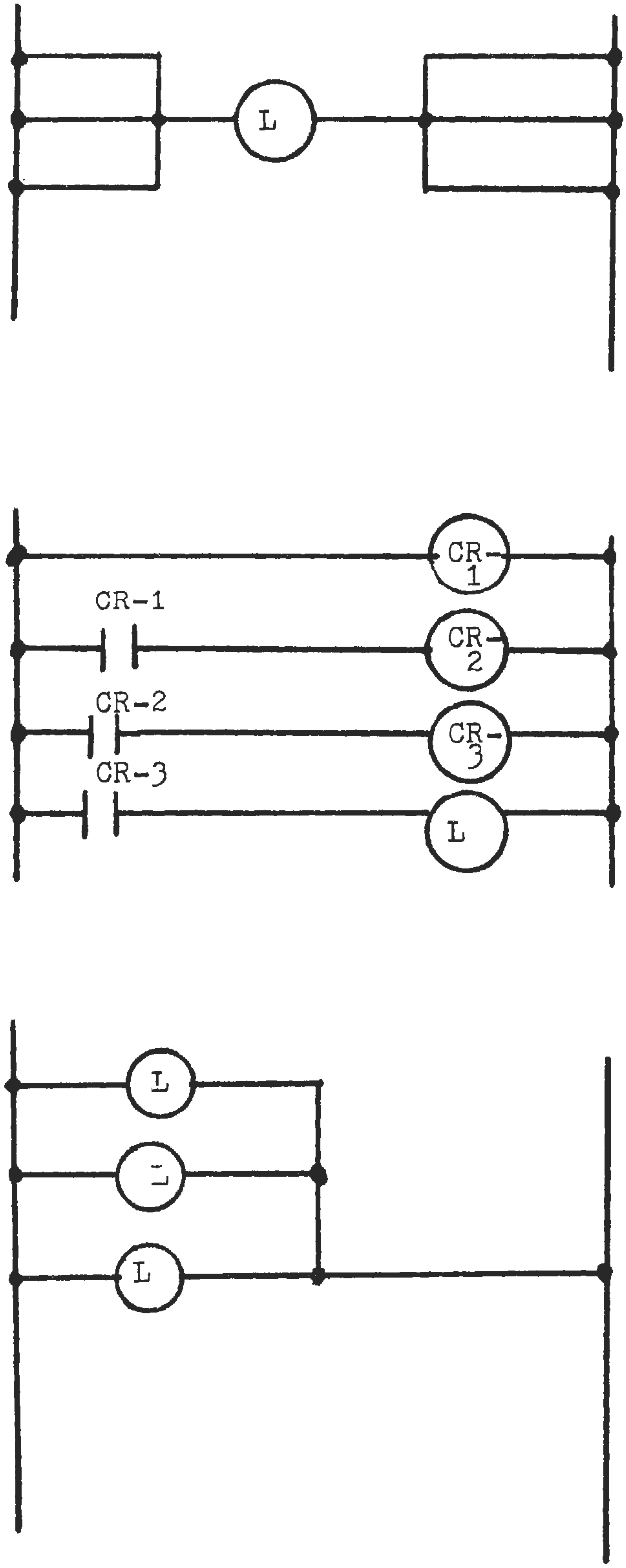


Figure 16. (continued)

Redundant and unnecessary parts of the circuit can be included as part of the object to be recognized without in any way affecting the function of the circuit, and hence its class membership.

We might be tempted to consider only the topology of the circuit in order that we might neglect the specific layout of lines, however, this would destroy our ability to recognize the input/output lines which are always represented as the left-most and right-most vertical lines.

It appears that any set of lines in parallel can be represented as one line.

6.2 The Semantic Paradigm Solution

The set of qualities of interest to an information processor attempting to recognize pictures in the "world" of electrical control circuit schematics was established to be the following.

$\overline{Q}_I = \{Q_1, Q_2, Q_3, Q_4\}$, where

Q_1 = the set of shapes or primitive symbols which are explicit in the picture such as those shown in Figure 15, and the alphanumerics, and implicit components such as the power source and sink = {"NOPB," "NCPB," "MCPB," "NOCONT," "NCCONT," "CR," "TIMER," "NOLS," "NCLS," "LOAD," "PILOT," "CONNEC," "WIRE," "A," "B,"...,"Z," "1," "2,"..., "9," "SOURCE," "SINK"}.

Q_2 = Conductivity = {OPEN, CLOSED}.

Q_3 = Energization = {ON, OFF}.

Q_4 = Function = {P001, P001A, P001B, P002, P003, P003A, P003B, P003C, P003D, P004, P004A, P004B, P004C, P004D} (These are described in detail in the following paragraphs).

P001 is defined to be the class in which the pushing of a normally-open push button (NOPB) causes a load to come on, and the subsequent releasing of the NOPB causes the load to go off and remain off. P001A and P001B are the intermediate properties (corresponding to states) of an object having property P001.

P002 is defined to be the class in which the pushing of one NOPB causes a load to come on, and the releasing of that NOPB results in no change in the state of the load (i.e., it remains on). The pushing of another NOPB or NCPB, however, causes the load to go off and remain off even after the second button is released.

P003 is defined to be the class in which the circuit functions as a P002, and has as its second push button a NOPB. P003A, P003B, P003C, and P003D are the intermediate properties (corresponding to states) of an object having property P003.

P004 is defined to be the class in which the circuit functions as a P002, and has as its second push button a NCPB. P004A, P004B, P004C, and P004D are the intermediate properties (corresponding to states) of an object having property P004.

We shall assume that I's sensory and preprocessing facilities are such that the following set of qualities can be said to be primitive. $\overline{Q}_I' = \{Q_1\}$.

The set of relations of interest to an information processor attempting to recognize pictures in the data base was established to be the following.

$\bar{R}_I = \{R_1, R_2, R_3\}$, where

R_1 = "__ is an identification name for __."

R_2 = "__ is electrically connected (+ to -) to __."

R_3 = "there exists a closed (electrical) path from __ to __."

We shall assume that I's sensory and preprocessing facilities are such that the following set of relations can be said to be primitive.

$\bar{R}'_I = \{R_1, R_2\}$.

6.2.1 The Primitive Scene Description

Before developing the rules of inference, we first comment on the nature of the primitive description for a circuit schematic. Since the set of primitive qualities consists only of the quality labeled shape, the object descriptions in the primitive scene description would normally consist of only a single property, i.e., the shape of the object. We find, however, that in the analysis of a circuit, we must imagine that, for example, push buttons in the circuit are being pushed by a human at various times. From this imagined situation, it is determined whether the required output states (also imagined) are achieved for a given circuit. Therefore, theoretically, a primitive description would have to contain all possible sequences of primitive state descriptions which one could imagine. For reasons of programming efficiency, however, it was found convenient to "generate" the sequence of primitive state descriptions as required in attempting to apply a rule of inference. Thus, everywhere the property OPEN or CLOSED is to be tested for an object having the primitive property NOPB or NCPB, a primitive state description was constructed in which the property OPEN or CLOSED was included in the description of the object in question. This in effect resulted in a "depth-first search" of a set of sequences of primitive state descriptions, which are generated during the search.

6.2.2 Rules of Inference

The rules of inference which follow directly from the descriptions of the classes of circuits are given below. Subscripted variables are used to indicate that they represent objects from different states of a scene description. In this manner, changes (although abstract) over time which are essential for the inference of some interesting properties are explicitly accounted for. Preceeding each formal rule is an articulated description.

RULE A: "If an object x_A in state description A has property P001A, and an object x_B in state description B has property P001B, then both objects have property P001."

IF $\exists$ x_A,x_B PROPERTY(x_A;P001A) AND PROPERTY(x_B;P001B) THEN BEGIN ADD(P001;x_A); ADD(P001;x_B); END.

RULE B: "If there is both a closed normally-open push button and an on load in a state description at a time just before the button becomes open and the load goes off, then the scene in the first state contains an object having property P001A, and the scene in the second state contains an object having property P001B."

IF $\exists$ $x_A,y_A,z_A,w_A,x_B,y_B,z_B,w_B$ PROPERTY(x_A;"NOPB") AND PROPERTY(x_A; closed) AND RELATION(R_1;z_A,x_A) AND PROPERTY(y_A;"LOAD") AND PROPERTY(y_A;on) AND RELATION(R_1;w_A,y_A) AND $t_B = (t_A + 1)$ AND PROPERTY(x_B;open) AND PROPERTY(y_B;off) AND RELATION(R_1;z_B,x_B) AND RELATION(R_1;w_B,y_B) AND $z_A = z_B$

AND $w_A = w_B$ THEN BEGIN ADD(P001A; COMBINE(all x in state desc. A)); ADD(P001B; COMBINE(all x in state desc. B)); END.

RULE C: "If an object has property P003 or P004, then it also has property P002."

IF $\exists$ x PROPERTY(x;P003) OR PROPERTY(x;P004) THEN ADD(x;P002).

RULE D: "If four consecutive state descriptions of a scene contain objects having properties P003A, P003B, P003C, and P003D respectively, then each of those objects also has property P003."

IF $\exists$ x_A, x_B, x_C, x_D PROPERTY(x_A;P003A) AND PROPERTY(x_B;P003B) AND PROPERTY(x_C;P003C) AND PROPERTY(x_D:P003D) THEN BEGIN ADD(x_A;P003); ADD(x_B;P003); ADD(x_C;P003); ADD(x_D;P003); END.

RULE E: "If state description A contains both a closed normally-open push button and an on load, and if in the next state B the push button is opened and the load remains on, and if in the next state description C, another normally-open push button is closed and the load goes off, and if in the next state description D, the second push button is opened and the load remains off, the four state descriptions will be said to contain objects having properties P003A, P003B, P003C, and P003D respectively."

IF $\exists$ $x_A, y_A, z_A, w_A, x_B, y_B, z_B, w_B, x_C, y_C, z_C, w_C, x_D, y_D, z_D, w_D$ PROPERTY(x_A; "NOPB") AND PROPERTY(x_A;closed) AND RELATION(R_1;z_A,x_A) AND PROPERTY(y_A; "LOAD") AND PROPERTY(y_A;on) AND RELATION(R_1;w_A,y_A) AND $t_B = (t_A+1)$ AND PROPERTY(x_B;open) AND PROPERTY(y_B;on) AND RELATION(R_1;z_B,x_B) AND RELATION (R_1;w_B,y_B) AND $z_A = z_B$ AND $w_A = w_B$ AND $t_C = (t_B+1)$ AND PROPERTY(x_C:"NOPB") AND PROPERTY(x_C;closed) AND RELATION(R_1;z_C,x_C) AND PROPERTY(y_C;off) AND

RELATION(R_1;w_C,y_C) AND $w_A = w_C$ AND $t_D = (t_C+1)$ AND PROPERTY(x_D;open) AND PROPERTY(y_D;off) AND RELATION(R_1;z_D,x_D) AND RELATION(R_1;w_D,y_D) AND $z_C = z_D$ AND $w_A = w_D$ THEN BEGIN ADD(P003A; COMBINE(all x in A)); ADD(P003B; COMBINE(all x in B)); ADD(P003C; COMBINE(all x in C)); ADD(P003D; COMBINE(all x in D)); END.

RULE F: "If state description A contains both a closed normally-open push button and an on load, and if in the next state B the push button is opened and the load remains on, and if in the next state description C, a normally-closed push button is opened and the load goes off, and if in the next state description D, the normally-closed push button is closed and the load remains off, the four state descriptions will be said to contain objects having properties P004A, P004B, P004C, and P004D, respectively."

IF $\exists\, x_A,y_A,z_A,w_A,x_B,y_B,z_B,w_B,x_C,y_C,z_C,w_C,x_D,y_D,z_D,w_D$ PROPERTY(x_A; "NOPB") AND PROPERTY(x_A;closed) AND RELATION(R_1;z_A,x_A) AND PROPERTY(y_A; "LOAD") AND PROPERTY(y_A;on) AND RELATION(R_1;w_A,y_A) AND $t_B = (t_A+1)$ AND PROPERTY(x_B;open) AND PROPERTY(y_B;on) AND RELATION(R_1;z_B,x_B) AND RELATION(R_1;w_B,y_B) AND $z_A = z_B$ AND $w_A = w_B$ AND $t_C = (t_B+1)$ AND PROPERTY(x_C; "NCPB") AND PROPERTY(x_C;open) AND RELATION(R_1;z_C,x_C) AND PROPERTY(y_C;off) AND RELATION(R_1;w_C,y_C) AND $w_A = w_C$ AND $t_D = (t_C+1)$ AND PROPERTY(x_D;closed) AND PROPERTY(y_D;off) AND RELATION(R_1;z_D,x_D) AND RELATION(R_1;w_D,y_D) AND $z_C = z_D$ AND $w_A = w_D$ THEN BEGIN ADD(P004A; COMBINE(all x in A)); ADD(P004B; COMBINE(all x in B)); ADD(P004C; COMBINE(all x in C)); ADD(P004D; COMBINE(all x in D)); END.

6.2.3 Additional Necessary Rules of Inference

Because not all of the properties and relations in the previous rules of inference are primitive, they must be inferred through additional rules which reflect knowledge of electrical control circuit theory. These are given below as before, with each rule preceded by an articulated description.

Relation R_3 ("there exists a closed (electrical) path from __ to __") is the only relation in $\bar{R}_I$ that is non-primitive, and hence must be inferred. The two rules for inferring that the relation holds between two points are simply:

RULE 1: "If an object is connected electrically to another object, there exists a closed path between the two objects."

IF $\exists$ x,y RELATION(R_2;x,y) THEN RELATE(R_3;x,y).

RULE 2: "If object x is connected electrically to object w, and w is closed and there exists a closed path from w to an object y, then there exists a closed path from x to y."

IF $\exists$ x,y,w RELATION(R_2;x,w) AND PROPERTY(w;closed) AND RELATION(R_3; w,y) THEN RELATE(R_3;x,y).

The qualities Q_2 = conductivity and Q_3 = energization must also be inferred, because properties in these sets are non-primitive.

Conductivity = {OPEN, CLOSED}. The only elements in a control circuit that are opened or closed internally are contacts (controlled by control relays). Other elements, such as push buttons, are controlled by time, humans, or processes and must be specified in the primitive descriptions (i.e., they cannot be inferred). The rules of inference for assigning the property OPEN or CLOSED take the form:

RULE 3: "If a control relay is on, its normally-open contacts must be closed."

IF $\exists$ x,y,z RELATION(R_1;z,x) AND PROPERTY(x;"CR") AND RELATION(R_1;z,y) AND PROPERTY(y;"NOCONT") AND PROPERTY(x;on) THEN BEGIN ADD(closed;y); DELETE(open;y); END.

RULE 4: "If a control relay is off, its normally-open contacts must be open."

IF $\exists$ x,y,z RELATION(R_1;z,x) AND PROPERTY(x;"CR") AND RELATION(R_1;z,y) AND PROPERTY(y;"NOCONT") AND PROPERTY(x;off) THEN BEGIN ADD(open;y); DELETE(closed;y); END.

RULE 5: "If a control relay is on, its normally-closed contacts must be open."

IF $\exists$ x,y,z RELATION(R_1;z,x) AND PROPERTY(x;"CR") AND RELATION(R_1;z,y) AND PROPERTY(y;"NCCONT") AND PROPERTY(x;on) THEN BEGIN ADD(open;y): DELETE(closed;y); END.

RULE 6: "If a control relay is off, its normally-closed contacts must be closed."

IF $\exists$ x,y,z RELATION(R_1;z,x) AND PROPERTY(x;"CR") AND RELATION(R_1;z,y) AND PROPERTY(y;"NCCONT") AND PROPERTY(x;off) THEN BEGIN ADD(closed;y); DELETE(open;y); END.

Energization = {ON, OFF}. The elements that can be internally energized are control relays, loads, pilots, and timers.

RULE 7: "If the power source is on, any energizable element not across the source must be off."

IF $\exists$ x,y,z (PROPERTY(x;"CR") OR PROPERTY(x;"LOAD") OR PROPERTY(x; "PILOT")) AND PROPERTY(x;on) AND PROPERTY(y;"SOURCE") AND PROPERTY(z; "SINK") AND NOT (RELATION(R_3;y,x) AND RELATION(R_3;x,z)) THEN BEGIN ADD(off; x); DELETE(on;x); END.

RULE 8: "If the power source is on, any energizable element across the source must be on."

IF $\exists$ x,y,z (PROPERTY(x;"CR") OR PROPERTY(x;"LOAD") OR PROPERTY(x;"PILOT")) AND PROPERTY(x;off) AND PROPERTY(y;source) AND PROPERTY(z;sink) AND RELATION(R_3;y,x) AND RELATION(R_3;x,z) THEN BEGIN ADD(on;x); DELETE (off;x); END.

RULE 9: "If a control relay is on, its timer, if it has one, must also be on."

IF $\exists$ x,y,z PROPERTY(x;"CR") AND PROPERTY(y;"TIMER") AND RELATION (R_1;z,x) AND RELATION(R_1;z,y) THEN BEGIN ADD(on;y); DELETE(off;y); END.

The above rules of inference were found to be sufficient for recognizing many circuits in the classes described. In the following section, the results of applying the semantic paradigm solution to the problem are described.

6.3 Results

A computer program was implemented to apply the rules of inference to primitive descriptions of various circuits. In the following, the power of the semantic paradigm is evident, as it is shown that many complicated circuit schematics were easily recognized. In no case was any circuit schematic improperly assigned to a class. Processing times for the recognition of pictures in the data base were quite reasonable considering that efficiency was not strived for. While times varied, about one minute processor (B5500) time represents the time required to determine to which of several interesting classes an arbitrary circuit might belong. A listing of this program is available from the author.

The five circuit schematics tested are illustrated in Figures 17, 19, 21, 23, and 25. Although they are very different in appearance, it was found that four of the five belong to class P001. One circuit belongs to three of the four classes. In Figures 18, 20, 22, 24, and 26 the output of the computer program is given for each test circuit. The reader will find that this problem is typical of those involving non-pictorial paraphrase. Because explicit use of contextual information is utilized and recognition is based on intensional class descriptions, we find that the semantic paradigm provides a simple solution to this problem.

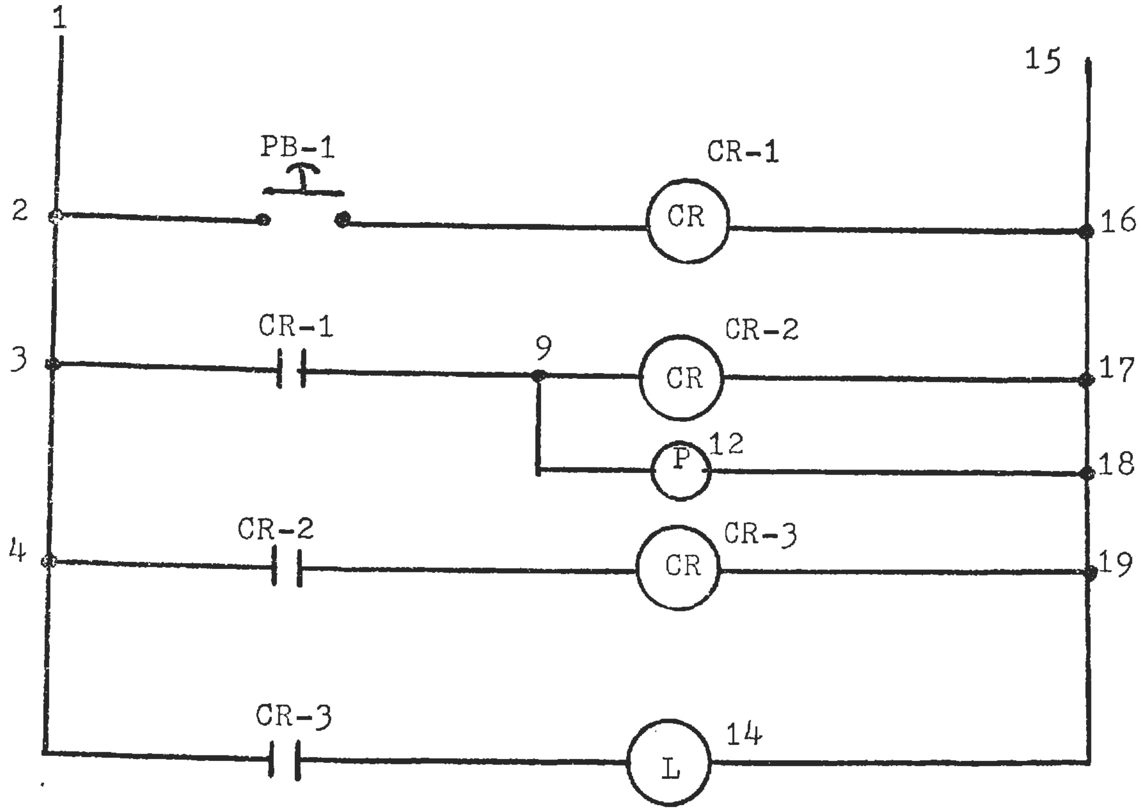


Figure 17. Test Circuit "A"

```
INPUT IS IN CLASS  00P001


P001 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF A NOPB CAUSES A LOAD TO COME ON, AND THE
SUBSEQUENT RELEASING OF THE PB CAUSES THE LOAD TO GO OFF AND REMAIN
OFF


INPUT NOT IN CLASS  00P003

P003... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NOPB


INPUT NOT IN CLASS  00P004

P004... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NCPB

INPUT NOT IN CLASS  00P002

P002 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF ONE NOPB CAUSES A LOAD TO COME ON, AND THE RELEASING
 OF THE BUTTON RESULTS IN NO CHANGE IN THE STATE OF THE LOAD
.     THE PUSHING OF ANOTHER NOPB OR NCPB, HOWEVER, CAUSES THE LOAD
TO GO OFF AND REMAIN OFF EVEN AFTER THE SECOND BUTTON IS RELEASED
```

Figure 18. Results of the Semantic Paradigm Solution
Applied to Test Circuit "A"

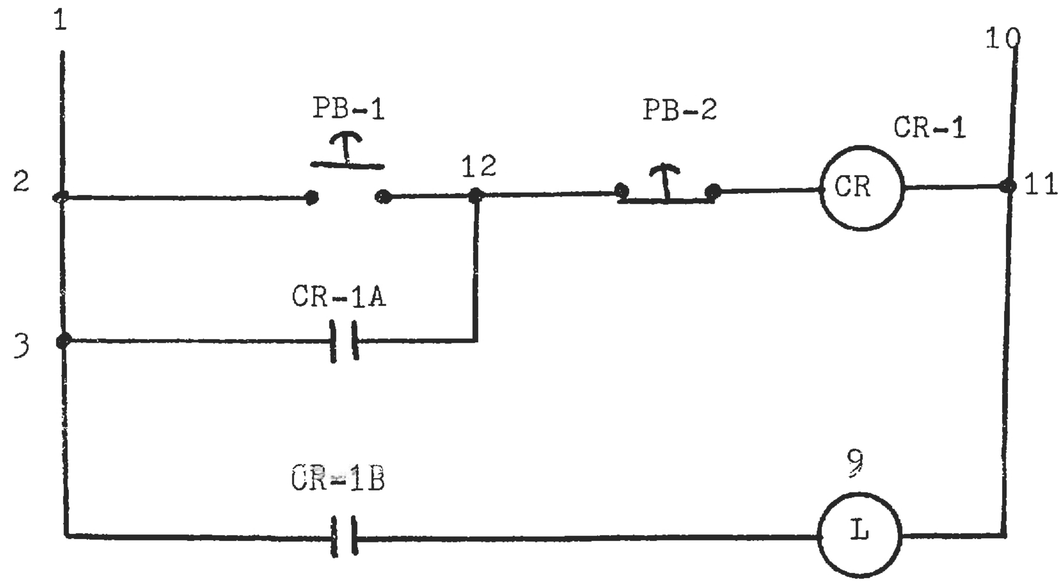


Figure 19. Test Circuit "B"

```
INPUT NOT IN CLASS  00P001

P001 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF A NOPB CAUSES A LOAD TO COME ON, AND THE
SUBSEQUENT RELEASING OF THE PB CAUSES THE LOAD TO GO OFF AND REMAIN
OFF


INPUT NOT IN CLASS  00P003

P003... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NOPB

INPUT IS IN CLASS  00P004


P004... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NCPB

INPUT IS IN CLASS  00P002


P002 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF ONE NOPB CAUSES A LOAD TO COME ON, AND THE RELEASING
 OF THE BUTTON RESULTS IN NO CHANGE IN THE STATE OF THE LOAD
     THE PUSHING OF ANOTHER NOPB OR NCPB, HOWEVER, CAUSES THE LOAD
TO GO OFF AND REMAIN OFF EVEN AFTER THE SECOND BUTTON IS RELEASED
```

Figure 20. Results of the Semantic Paradigm Solution Applied to Test Circuit "B"

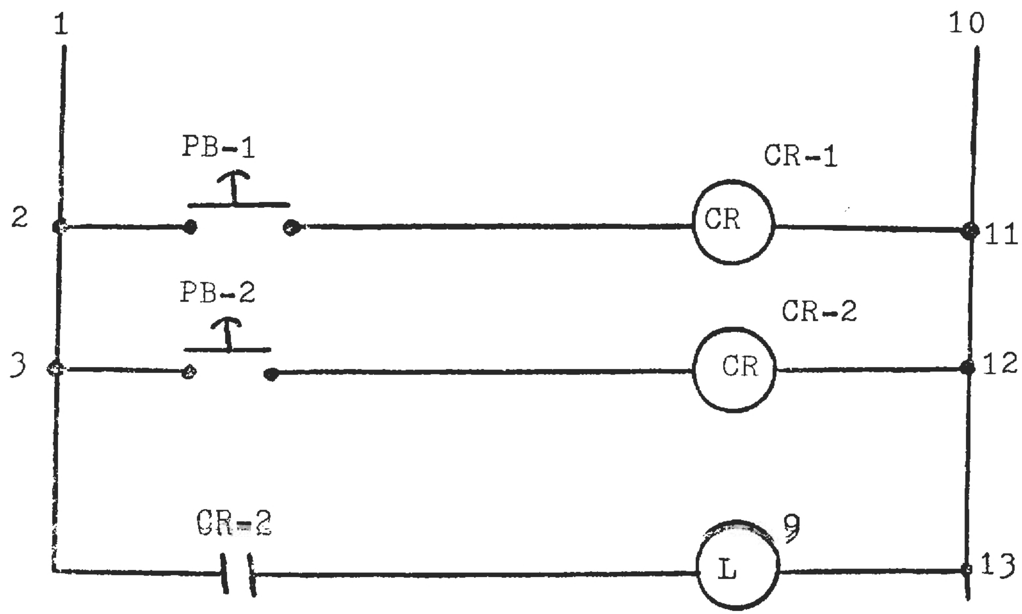


Figure 21. Test Circuit "C"

```
INPUT IS IN CLASS  00P001

P001 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF A NOPB CAUSES A LOAD TO COME ON, AND THE
SUBSEGUENT RELEASING OF THE PB CAUSES THE LOAD TO GO OFF AND REMAIN
OFF

INPUT NOT IN CLASS  00P003

P003... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NOPB

INPUT NOT IN CLASS  00P004

P004... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NCPB

INPUT NOT IN CLASS  00P002

P002 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF ONE NOPB CAUSES A LOAD TO COME ON, AND THE RELEASING
 OF THE BUTTON RESULTS IN NO CHANGE IN THE STATE OF THE LOAD
     THE PUSHING OF ANOTHER NOPB OR NCPB, HOWEVER, CAUSES THE LOAD
TO GO OFF AND REMAIN OFF EVEN AFTER THE SECOND BUTTON IS RFLEASED
```

Figure 22. Results of the Semantic Paradigm Solution Applied to Test Circuit "C"

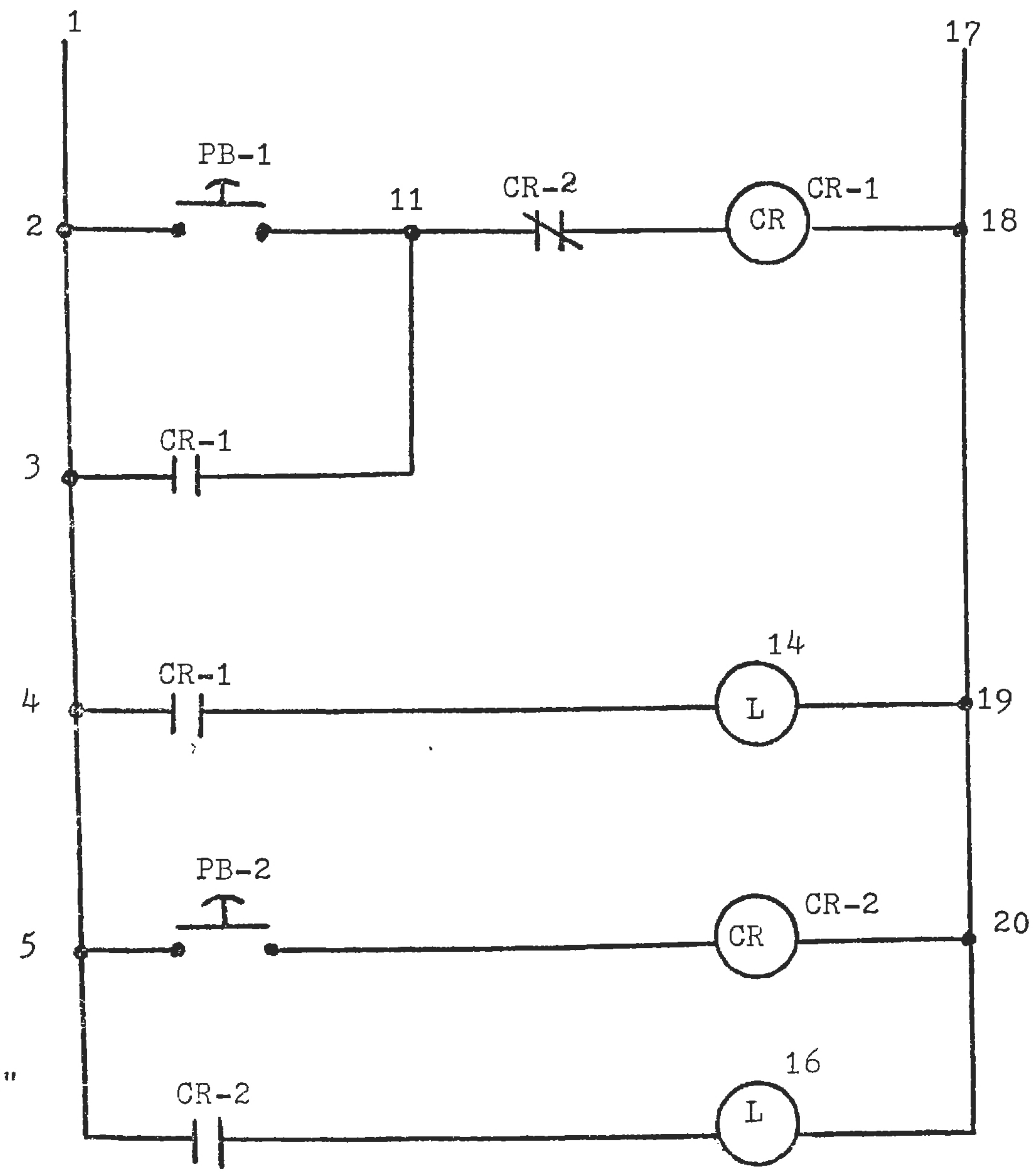


Figure 23.
Test Circuit "D"

```
INPUT IS IN CLASS  00P001

P001 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF A NOPB CAUSES A LOAD TO COME ON, AND THE
SUBSEQUENT RELEASING OF THE PB CAUSES THE LOAD TO GO OFF AND REMAIN
OFF

INPUT IS IN CLASS  00P003

P003... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NOPB

INPUT NOT IN CLASS  00P004

P004... IS DEFINED TO BE ANY CIRCUIT IN WHICH
 THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NCPB

INPUT IS IN CLASS  00P002

P002 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF ONE NOPB CAUSES A LOAD TO COME ON, AND THE RELEASING
 OF THE BUTTON RESULTS IN NO CHANGE IN THE STATE OF THE LOAD
 THE PUSHING OF ANOTHER NOPB OR NCPB, HOWEVER, CAUSES THE LOAD
TO GO OFF AND REMAIN OFF EVEN AFTER THE SECOND BUTTON IS RELEASED
```

Figure 24. Results of the Semantic Paradigm Solution Applied to Test Circuit "D"

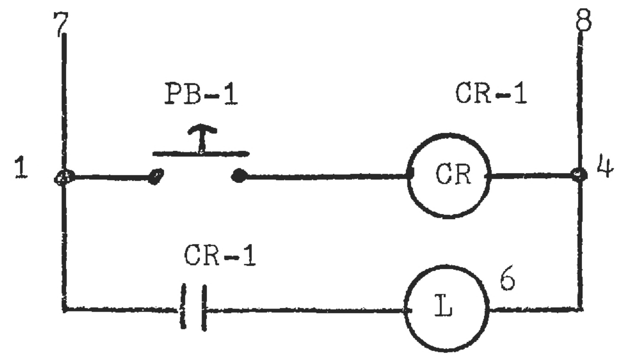


Figure 25. Test Circuit "E"

INPUT IS IN CLASS 00P001

P001 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF A NOPB CAUSES A LOAD TO COME ON, AND THE
SUBSEQUENT RELEASING OF THE PB CAUSES THE LOAD TO GO OFF AND REMAIN
OFF

INPUT NOT IN CLASS 00P003

P003... IS DEFINED TO BE ANY CIRCUIT IN WHICH
THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NOPB

INPUT NOT IN CLASS 00P004

P004... IS DEFINED TO BE ANY CIRCUIT IN WHICH
THE CIRCUIT ACTS LIKE A P002, AND HAS AS ITS SECOND PB A NCPB

INPUT NOT IN CLASS 00P002

P002 ..IS DEFINED TO BE THE CLASS IN WHICH
THE PUSHING OF ONE NOPB CAUSES A LOAD TO COME ON, AND THE RELEASING
OF THE BUTTON RESULTS IN NO CHANGE IN THE STATE OF THE LOAD
. THE PUSHING OF ANOTHER NOPB OR NCPB, HOWEVER, CAUSES THE LOAD
TO GO OFF AND REMAIN OFF EVEN AFTER THE SECOND BUTTON IS RELEASED

Figure 26. Results of the Semantic Paradigm Solution Applied to Test Circuit "E"

CHAPTER VII

APPLICATION OF THE PARADIGM FOR A PROBLEM INVOLVING NON-IDEAL DATA

7.1 The Problem

This problem has been described in detail by Kelly (1970). The task is to choose, from a collection of pictures of people taken by a TV camera, those pictures that depict the same person. Kelly's approach to the problem involved the development of a program to find the location of features such as eyes, nose, or shoulders in the picture. Individuals are classified by measurements between such features. The difficult part of the work is the detection of these features in digital pictures.

In order to solve the problem Kelly employed many specialized picture processing techniques, among them subtraction, template matching, edge detection operators, planning, goal-directed search, dynamic threshold setting followed by smoothing, and line detection operators. As a result, a kind of complex picture processing problem was solved which has never been solved before. The success of the program was due to the heuristic and ad hoc use of context and structure. As Kelly has suggested,

> Another valuable extension of this work would be to attempt to develop a formal structure that would permit high-level specification of the model and the heuristics which are buried in the program. . . . Possible lines of approach could be found in the attempts at linguistic description of pictures and formalization of models. . . . Such a high-level specification, if successful in reproducing the effect of the algorithms . . . could be applied . . . to test its generality.

It was felt that the semantic paradigm might harbor the necessary machinery for doing just this. The specific problem chosen was to take digital pictures from the data base and attempt to locate the features hair, eyebrows, eyes, nose, and mouth using the semantic paradigm.

In this study, the classification aspect of the problem was ignored, in order to focus attention on the more difficult problem of feature detection.

The successful utilization of the semantic paradigm for location of features in real pictures of people would represent a significant achievement in picture processing. This follows since no <u>general</u> picture processing technique has yet been successfully applied to such a problem.

7.2 The Semantic Paradigm Solution

The set of qualities of interest to an information processor attempting to recognize facial features in pictures of people was established to be the following.

$\bar{Q}_I = \{Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7\}$, where

Q_1 = Facial feature = {hair, left eyebrow, right eyebrow, left eye, right eye, left nostril, right nostril, mouth}.

Q_2 = Height-to-width ratio = {flat ($<.375$), medium (≤ 1.2 and $\geq .375$), tall (>1.2)}.

Q_3 = Length = {short (<7), long (≥ 7)}.

Q_4 = Size = {small (>1, <30), big (>400)}.

Q_5 = Connectivity = {connected (4-connectedness)*, non-connected}.

---

*A region R (subset of points in an array) is said to be <u>connected</u> if every point in the region has a binary value of 1, and for any two points s_1 and s_n in the region, there exists a sequence $s_1, s_2, \ldots, s_n$

Q_6 = Average light intensity = $\{0,1,\ldots,14,15\}$.

Q_7 = Binary value = $\{0, 1\}$.

We shall assume that I's sensory and preprocessing facilities are such that the following set of qualities can be said to be primitive.

$\overline{Q}'_I = \{Q_2, Q_3, Q_4, Q_5, Q_6, Q_7\}$.

The set of relations of interest to an information processor attempting to recognize facial features in pictures of people was established to be the following.

$\overline{R}_I = \{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9\}$, where

R_1 = "__ is totally to the right of __."

R_2 = "__ is totally to the left of __."

R_3 = "__ is totally above __."

R_4 = "__ is totally below __."

R_5 = "__ is near from above ($\leq$ 6 units) to __."

R_6 = "__ is close from above ($\leq$ 30 units) to __."

R_7 = "__ is near from the side ($\leq$ 6 units) to __."

R_8 = "__ intersects horizontally with __."

R_9 = "__ is completely or partially to the right of __."

We shall assume that I's sensory and preprocessing facilities are such that the following set of relations can be said to be primitive.

$\overline{R}'_I = \overline{R}_I$.

---

of points in R such that s_i is a neighbor of s_{i-1}. Two points are neighbors if they are immediately adjacent horizontally or vertically.

7.2.1 Representation of a Picture

Each picture in the data base is a one-half sized version (reduced by averaging) of that used by Kelly (1970). The picture is represented as an integer array having a maximum of 105 rows and 110 columns. Each element of the array has an integer value from zero to fifteen, representing the average light intensity at that point. Zero represents the darkest points and fifteen represents the lightest points. The hexadecimal representation of a typical picture is illustrated in Figure 27. Because such a representation is difficult for a human to work with, a procedure for overprinting to achieve a grey tone effect is utilized based on that proposed by Stucki (1969). Figure 28 illustrates the result of three overprints to achieve the darkness or grey tone for the picture represented in Figure 27.

The reader will note that reduction and reproduction of such pictures results in a noticeable degradation of the quality of the picture. If the picture is viewed from a distance, however, the features will be very much more noticeable.

7.2.2 The Primitive Scene Description

The primitive description of a picture in the data base was established to be $PD(P) = \{PSD^1(P),\ldots,PSD^{16}(P)\}$, where PSD^i is derived from a thresholded version of the original picture, in which each element of the array has the binary value 1 if the average light intensity of that element in the original picture is less than i, and 0 otherwise. Figure 28 illustrates a typical picture in the data base, and Figures 29-36 illustrate the thresholded pictures from which the primitive state

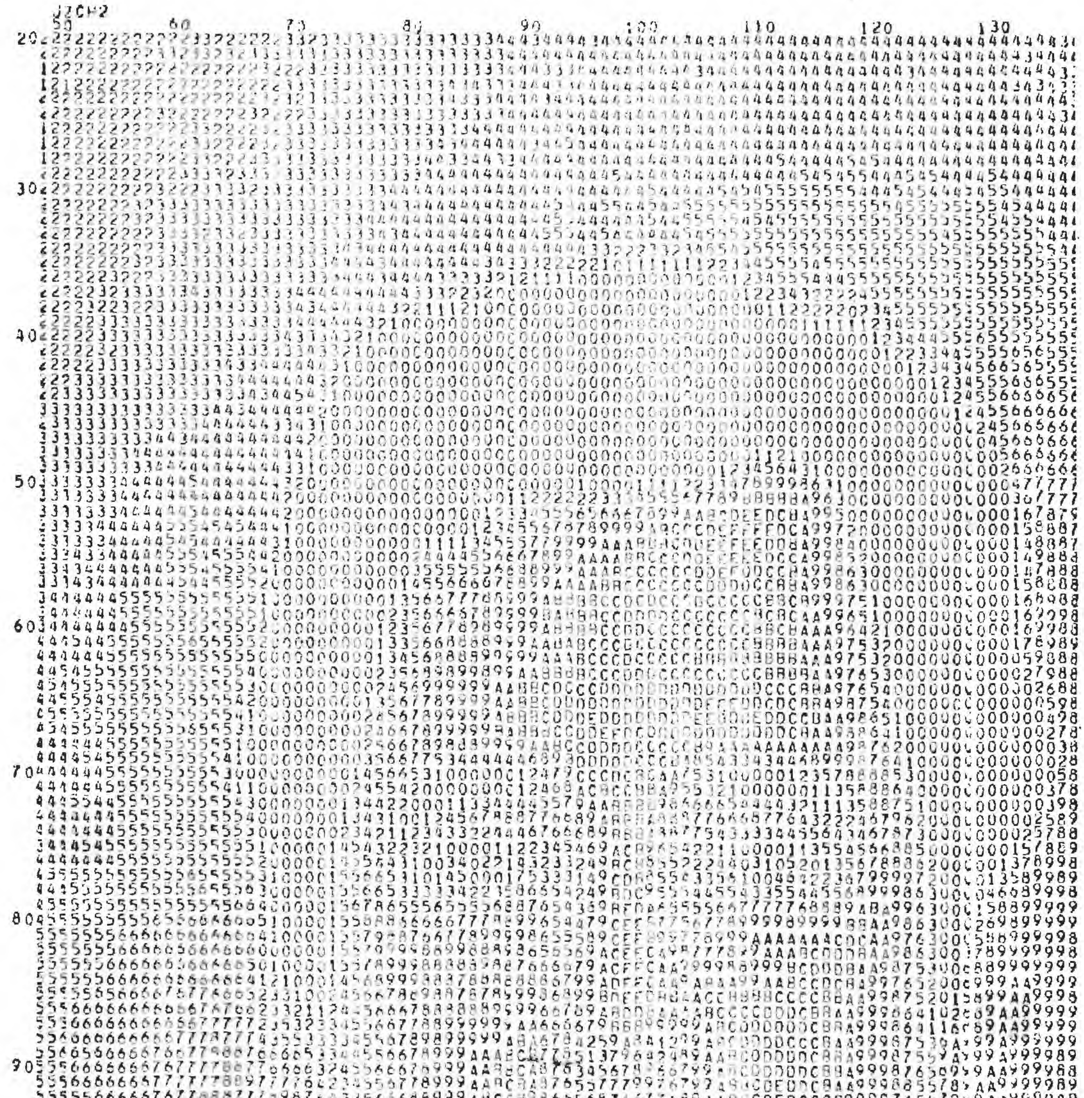


Figure 27. Hexadecimal Representation of Digitized Picture

Figure 28. Overprinting to Achieve Grey
Tone Effect of Picture in Figure 27

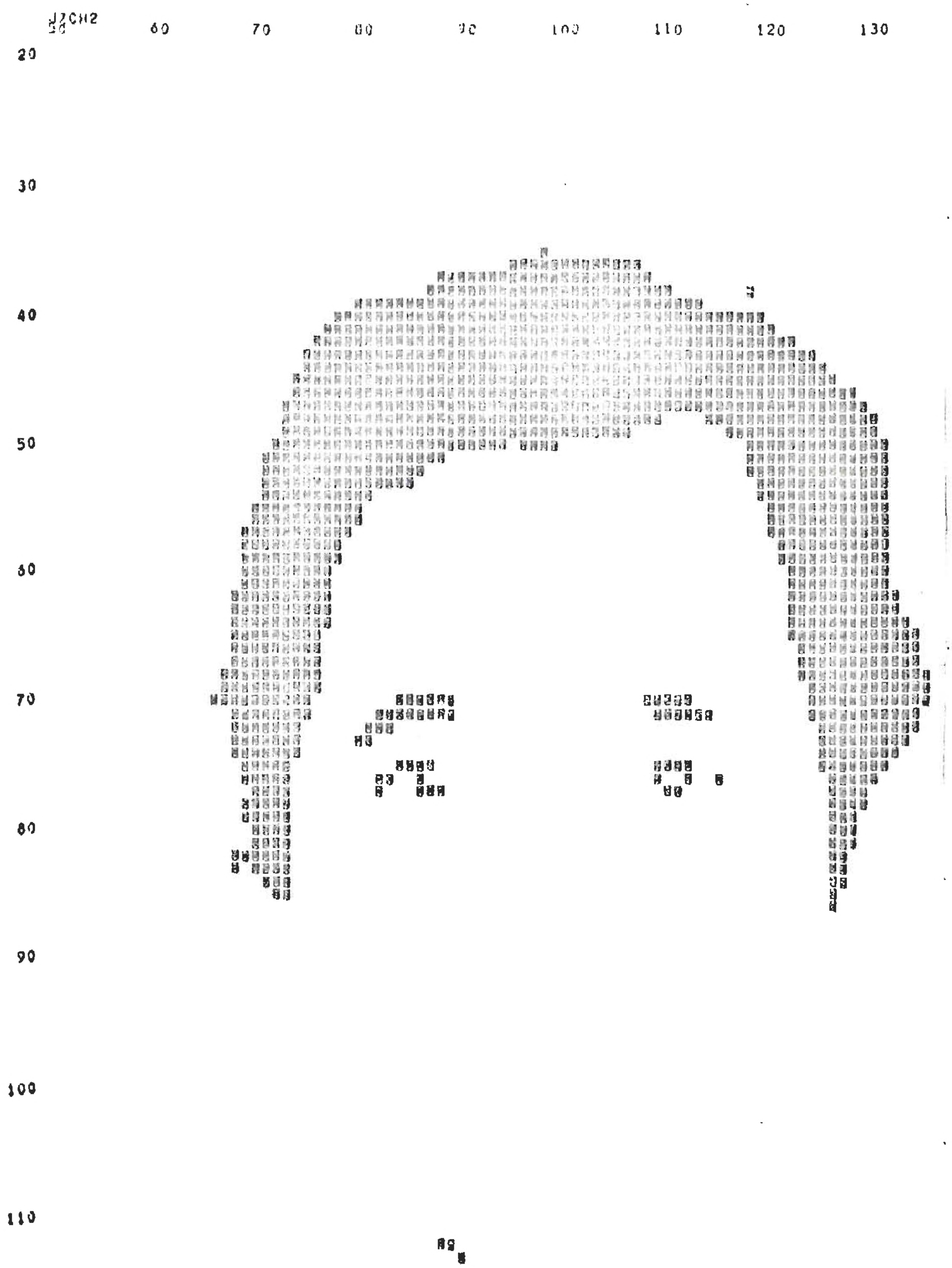


Figure 29. Thresholded Picture-Level 1

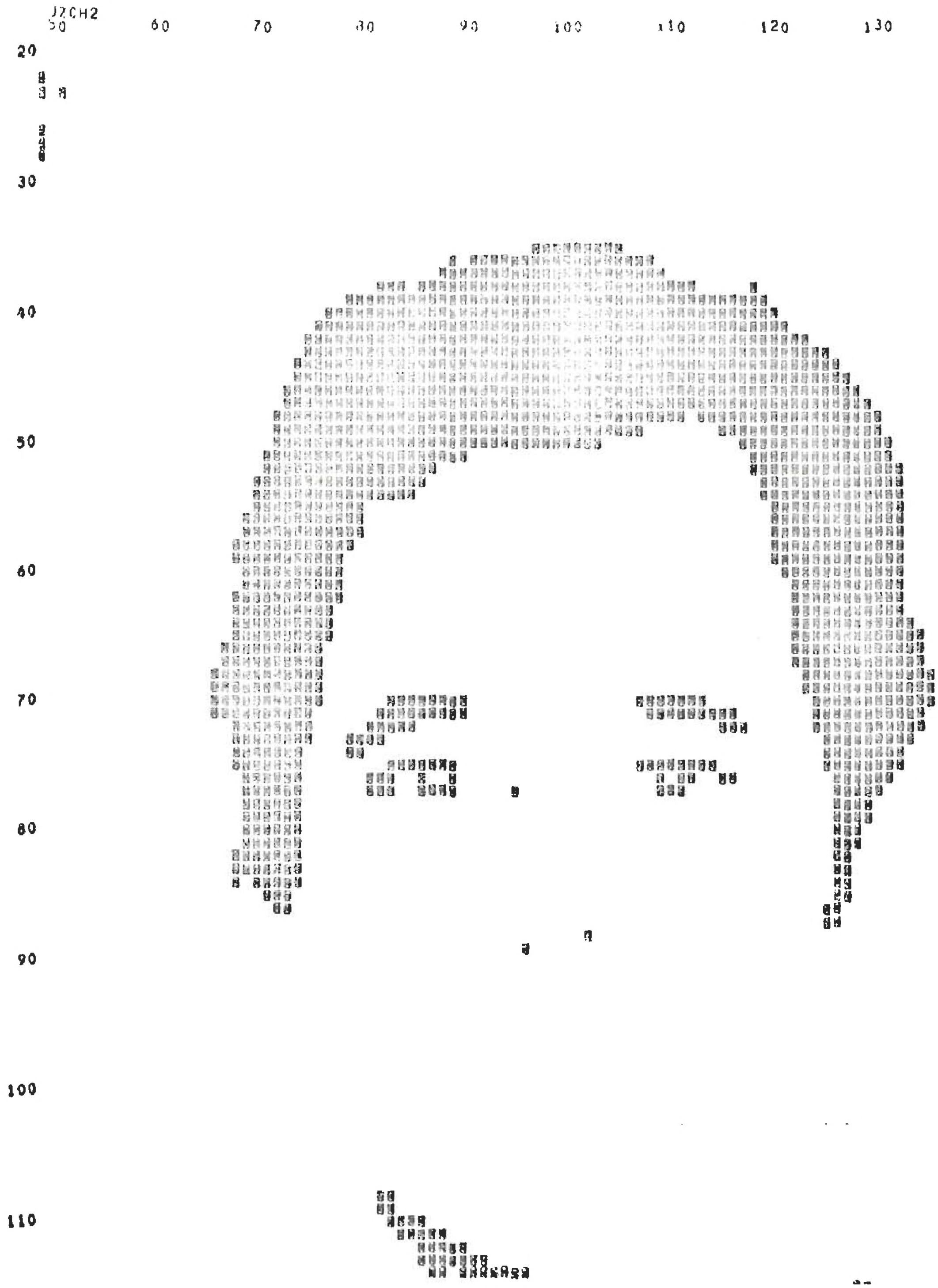


Figure 30. Thresholded Picture-Level 2

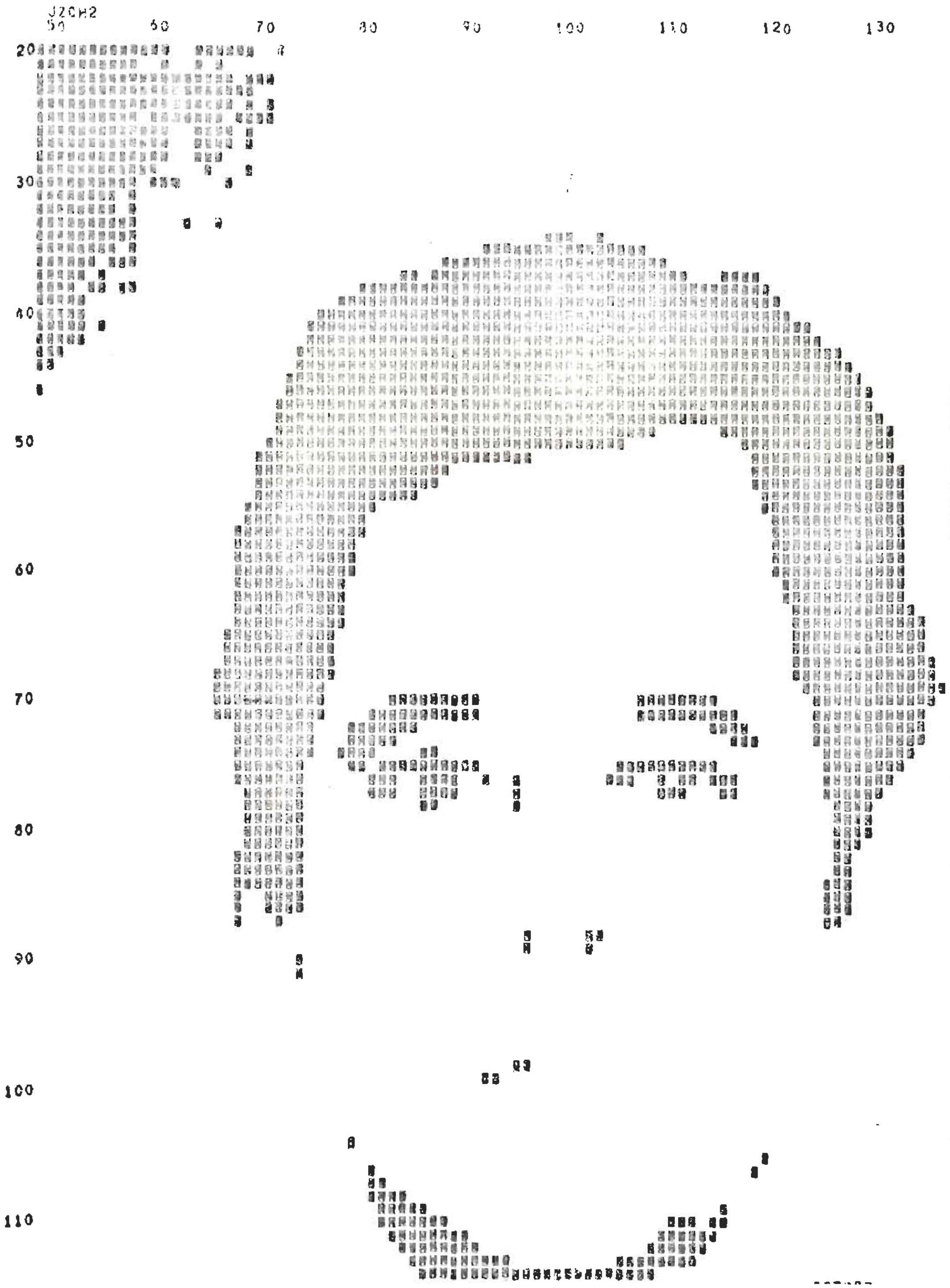


Figure 31. Thresholded Picture-Level 3

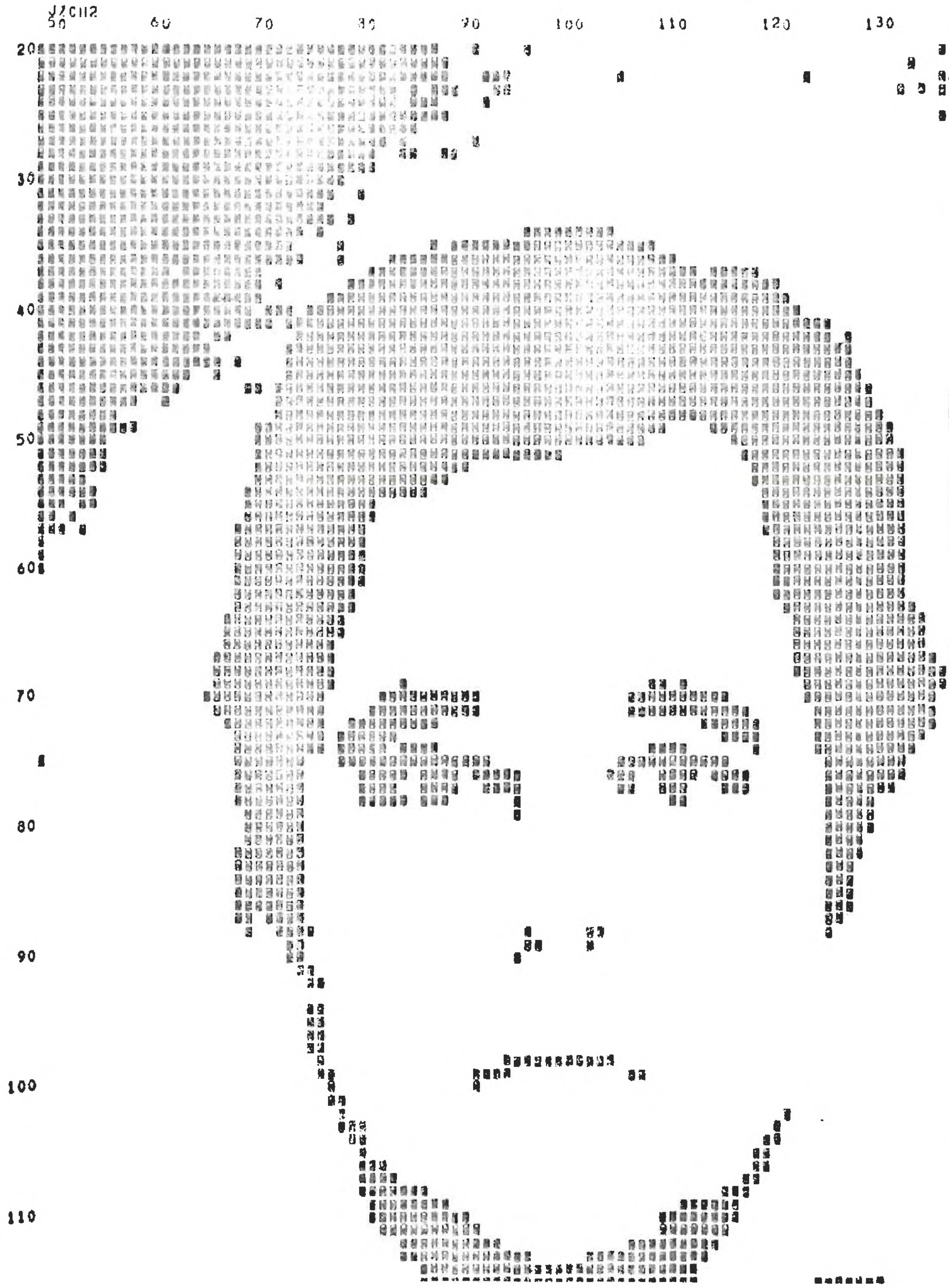


Figure 32. Thresholded Picture-Level 4

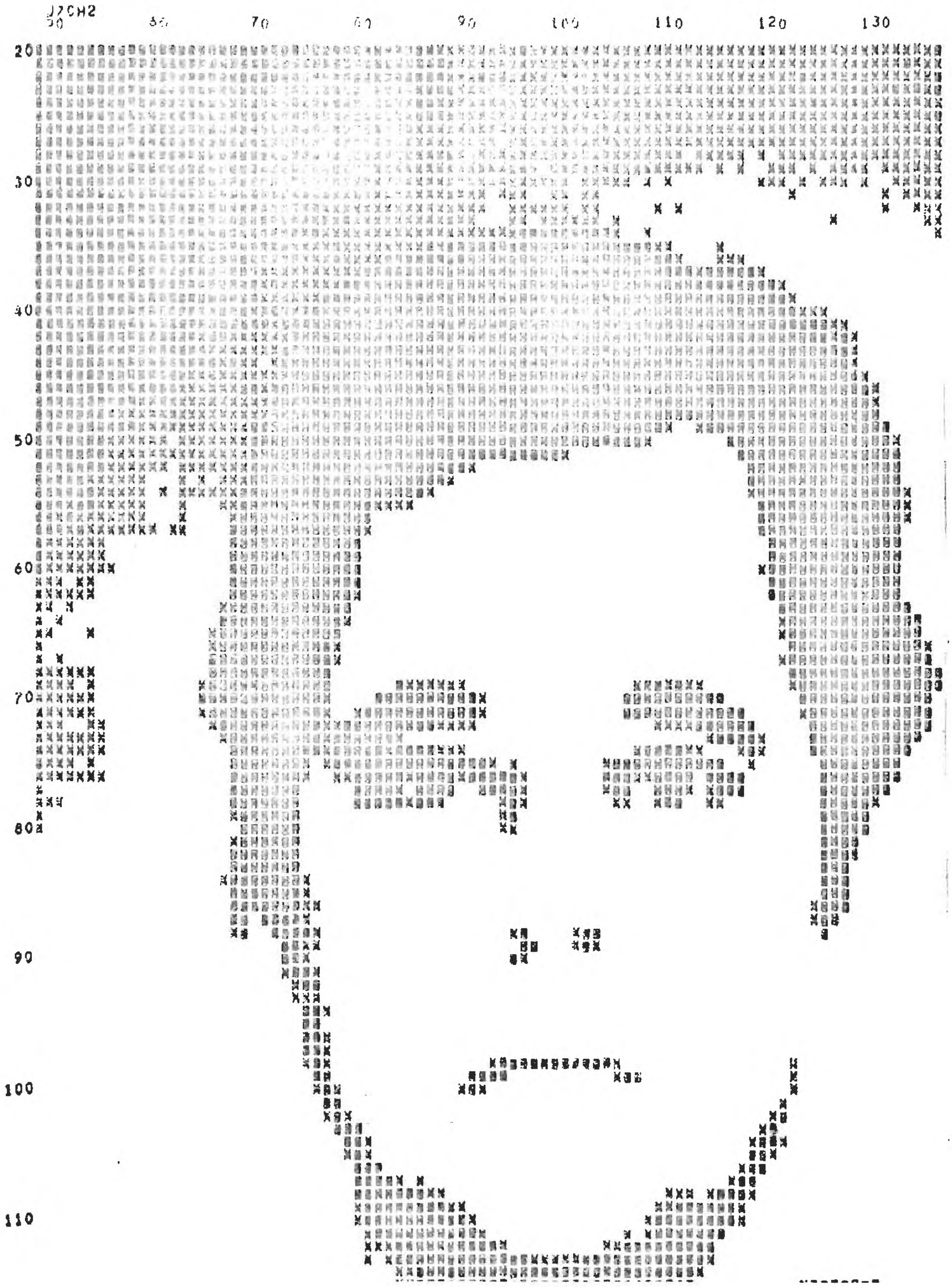


Figure 33. Thresholded Picture-Level 5

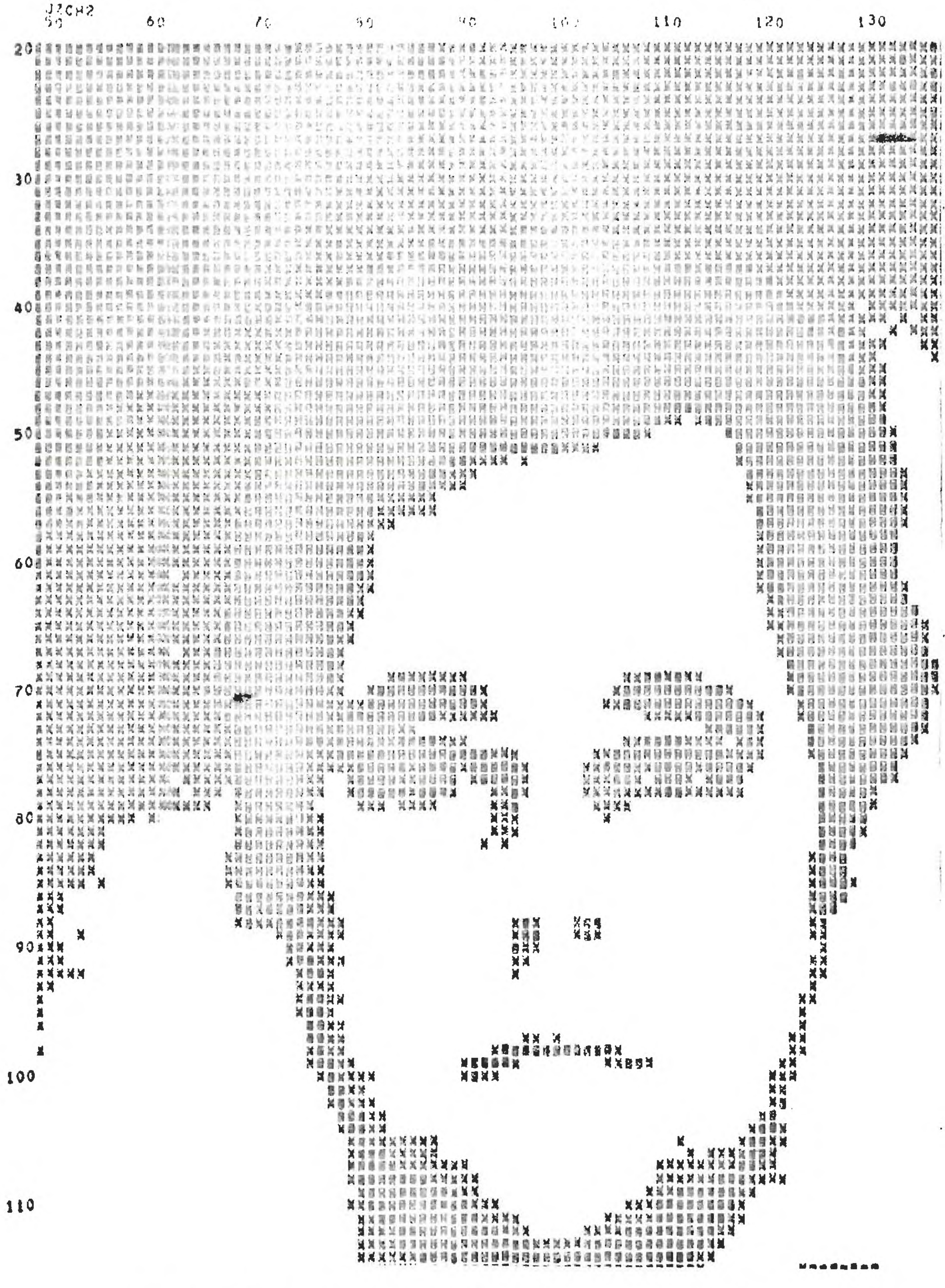


Figure 34. Thresholded Picture-Level 6

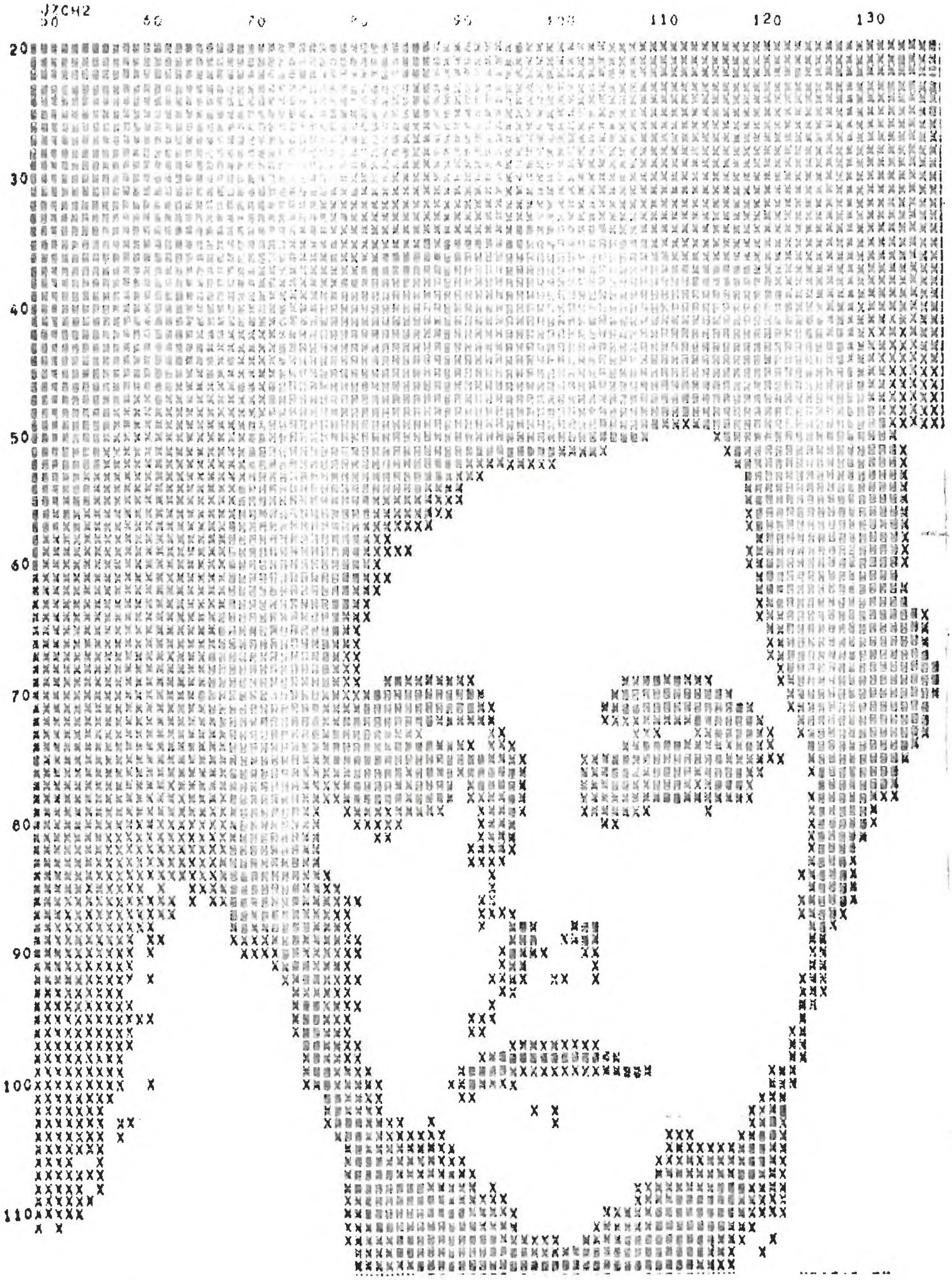


Figure 35. Thresholded Picture-Level 7

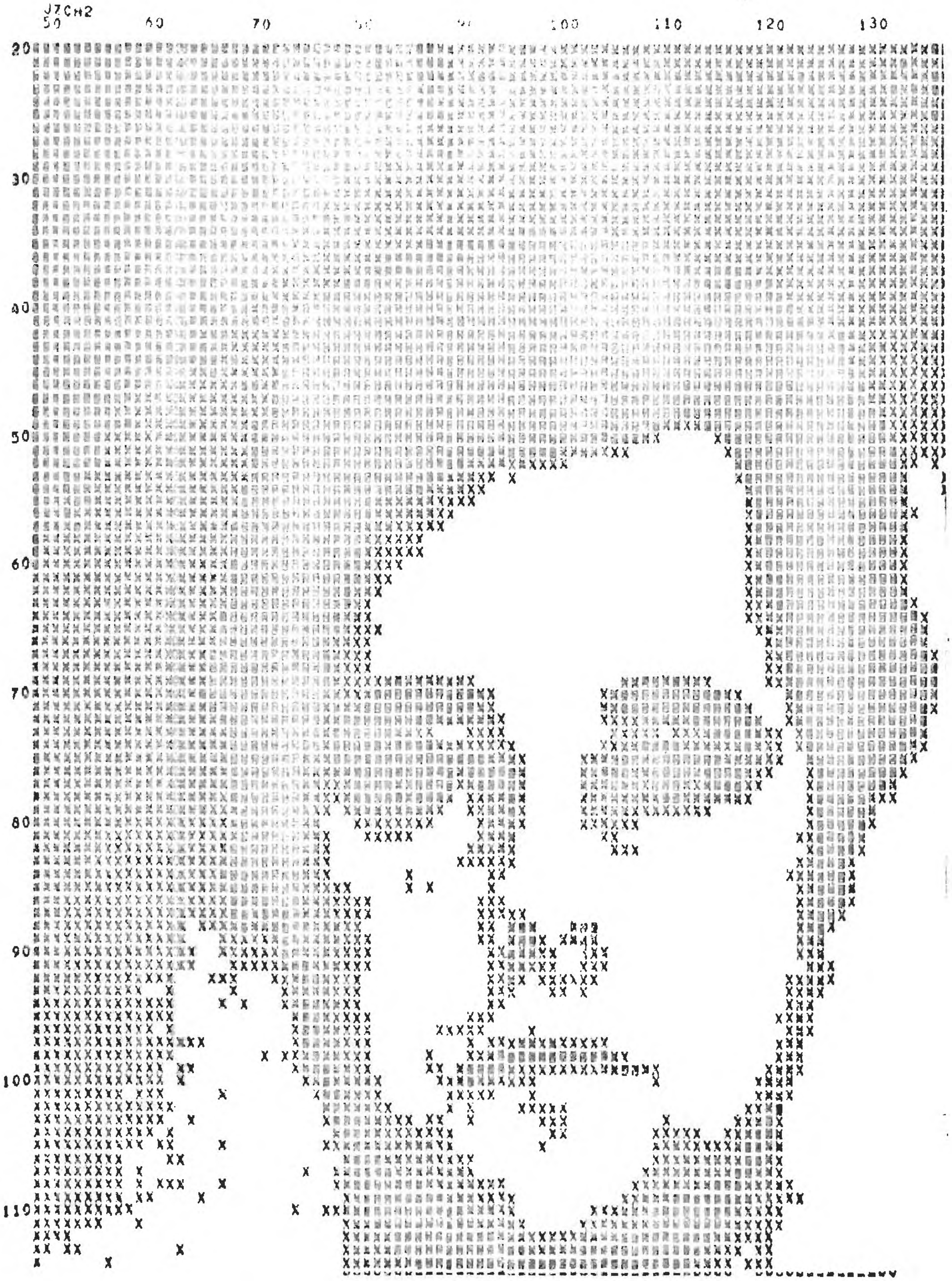


Figure 36. Thresholded Picture-Level 8

descriptions $PSD^1(P)$ - $PSD^8(P)$ respectively, are derived.

The rules of inference were then applied according to the following rule: select the first rule and apply it to $PSD^1(P)$. If it is found not to hold, then apply it to $PSD^2(P)$, and so on, until either the rule is found to hold or the rule is found not to hold for $PSD^{16}(P)$. Apply all of the remaining rules in the same fashion.

7.2.3 Rules of Inference

It was found that four simple rules of inference were adequate for recognizing the facial features. They are given below in natural language, however, the reader can easily construct the formal rules using the properties and relations given above.

RULE 1: If there exists a connected object which is big and has a medium height-to-width ratio, then that object has the property hair.

RULE 2: If there exist four objects O_1, O_2, O_3, and O_4 such that each are connected objects, and they are small, and O_1 and O_3 are near from above and totally above O_2 and O_4, and O_3 and O_4 are totally to the right of O_1 and O_2, then object O_1 has the property left eyebrow (left, that is, from the viewer's viewpoint), object O_2 has the property left eye, O_3 has the property right eyebrow, and O_4 has the property right eye.

RULE 3: If there exists a connected object which is long and flat and it is totally below the left eye, and the left eye is near from above to the object, then that object has the property mouth.

RULE 4: If there exist objects O_1 and O_2 which are small connected objects, and they are totally above the mouth, and are totally below the left eye, and O_1 is completely or partially to the right of the left eyebrow, and O_2 is totally to the right of O_1, and O_1 is near from the side to O_2, and O_1 intersects horizontally with O_2, and the right eyebrow is completely or partially to the right of O_2, then O_1 has the property left nostril, and O_2 has the property right nostril.

7.3 Results

A total of twenty pictures were randomly selected from the data used by Kelly (1970). Each of these were then reduced by averaging to one-half their original size in order to make line printer output more convenient. It was found that this resulted in substantial degradation of the picture, to the extent that a human could not easily identify the locations of certain features such as nostrils, mouth, etc. in many of the pictures. For this reason, the ten "best" pictures (i.e., those in which a human could most easily identify the locations of features) were selected as our data base.

The semantic paradigm solution described in the previous section was applied to these ten pictures.

In order to illustrate the results of applying the semantic paradigm solution to a picture in the data base, a new picture was created depicting the locations (regions) of the objects of interest (the facial features). In this new picture, the following symbols are shown to indicate the regions occupied by the various features:

"H" for hair.

"B" and "R" for the left and right (viewer's point of view) eyebrows, respectively.

"E" and "Y" for the left and right eyes, respectively.

"N" and "O" for the left and right nostrils, respectively.

"M" for the mouth.

Figures 37-56 show the original pictures in the data base as well as the pictures depicting the regions occupied by the facial features. As can be seen, the semantic paradigm solution recognized all of the features in eight of the ten pictures, and recognized all but the nostrils and the mouth in the remaining two.

In the following, we make a comparative analysis of these results with those of Kelly (1970), with regard to the following factors: (1) exactness of location of regions of the features, (2) time required to write the recognition program, (3) time required to execute the program, and (4) length of the program.

7.3.1 Exactness of Location of Regions of Features

Table 1 shows the following measurements, and compares the results produced with those of Kelly: (1) width of head, (2) distance between centers of eyes, (3) distance from center of eyes to top of head, (4) distance from center of eyes to nostrils, and (5) distance from center of eyes to center of mouth. As can be seen, the semantic paradigm produced almost identical measures as those obtained by Kelly. Because of the subjective nature of the determination of the actual measurements, and because the rules of inference were not designed to determine the exact locations of features, but rather just to infer their existence, no statistical analysis of the data is presented.

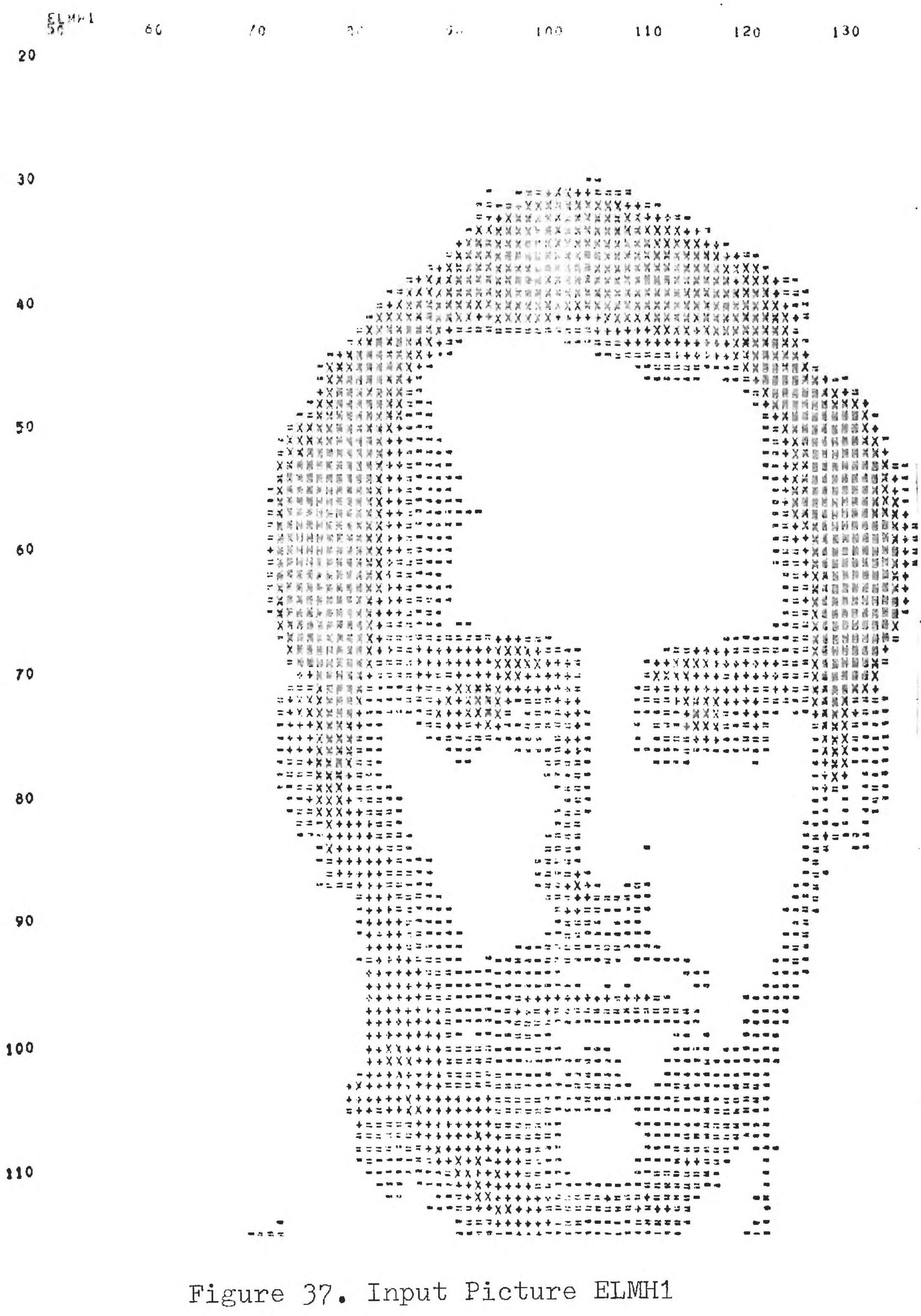


Figure 37. Input Picture ELMH1

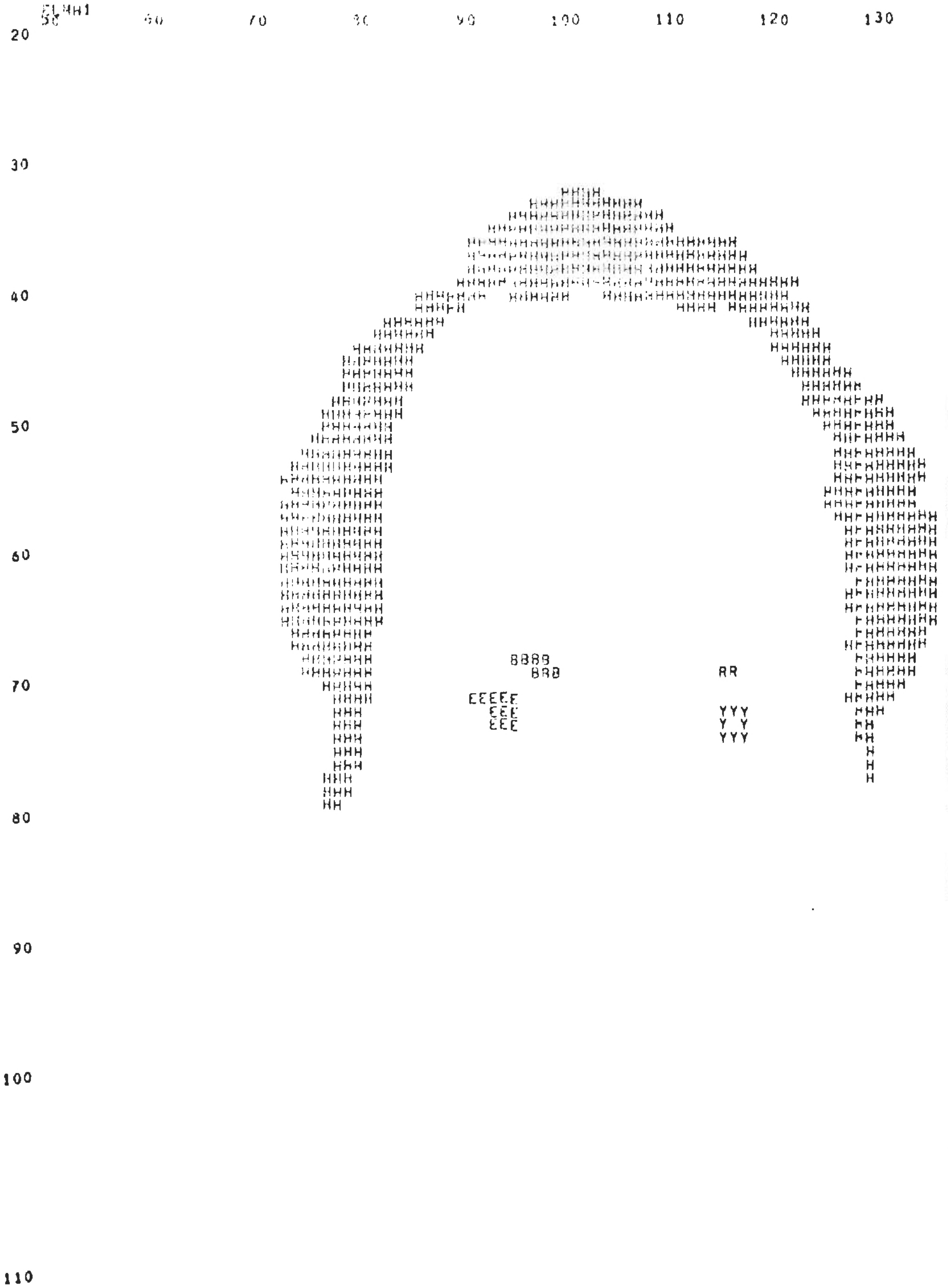


Figure 38. Features Recognized in ELMH1

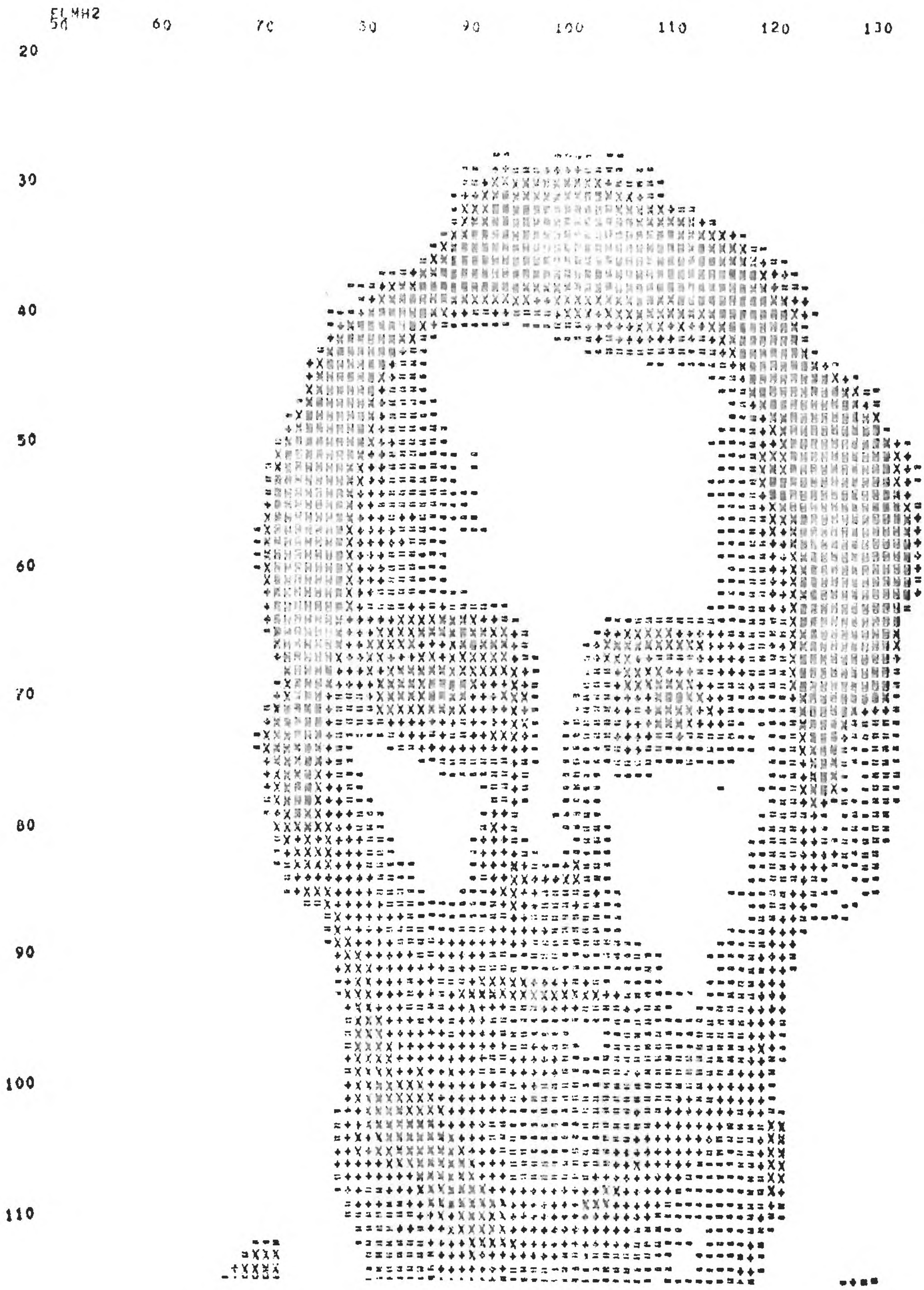


Figure 39. Input Picture ELMH2

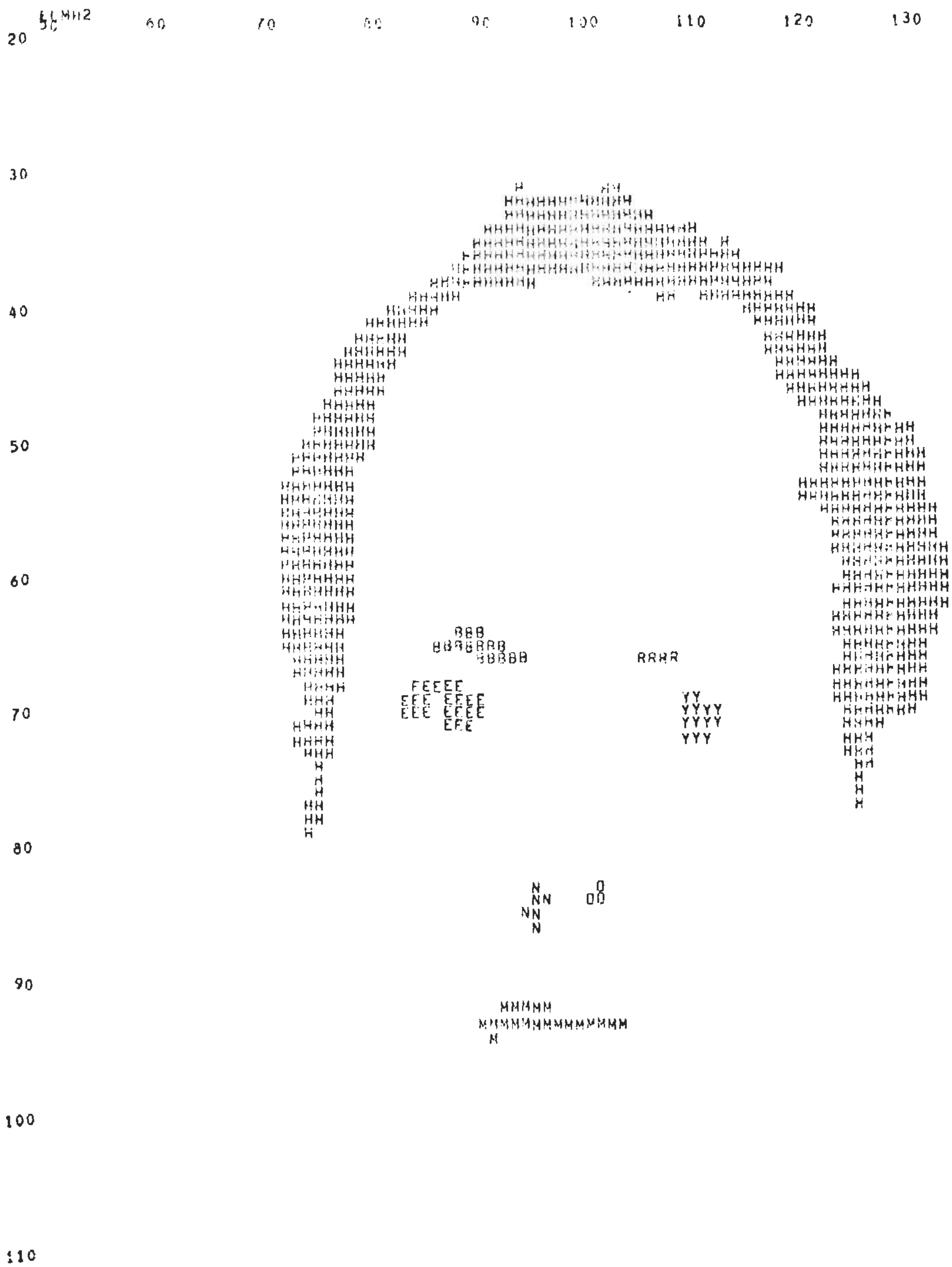


Figure 40. Features Recognized in ELMH2

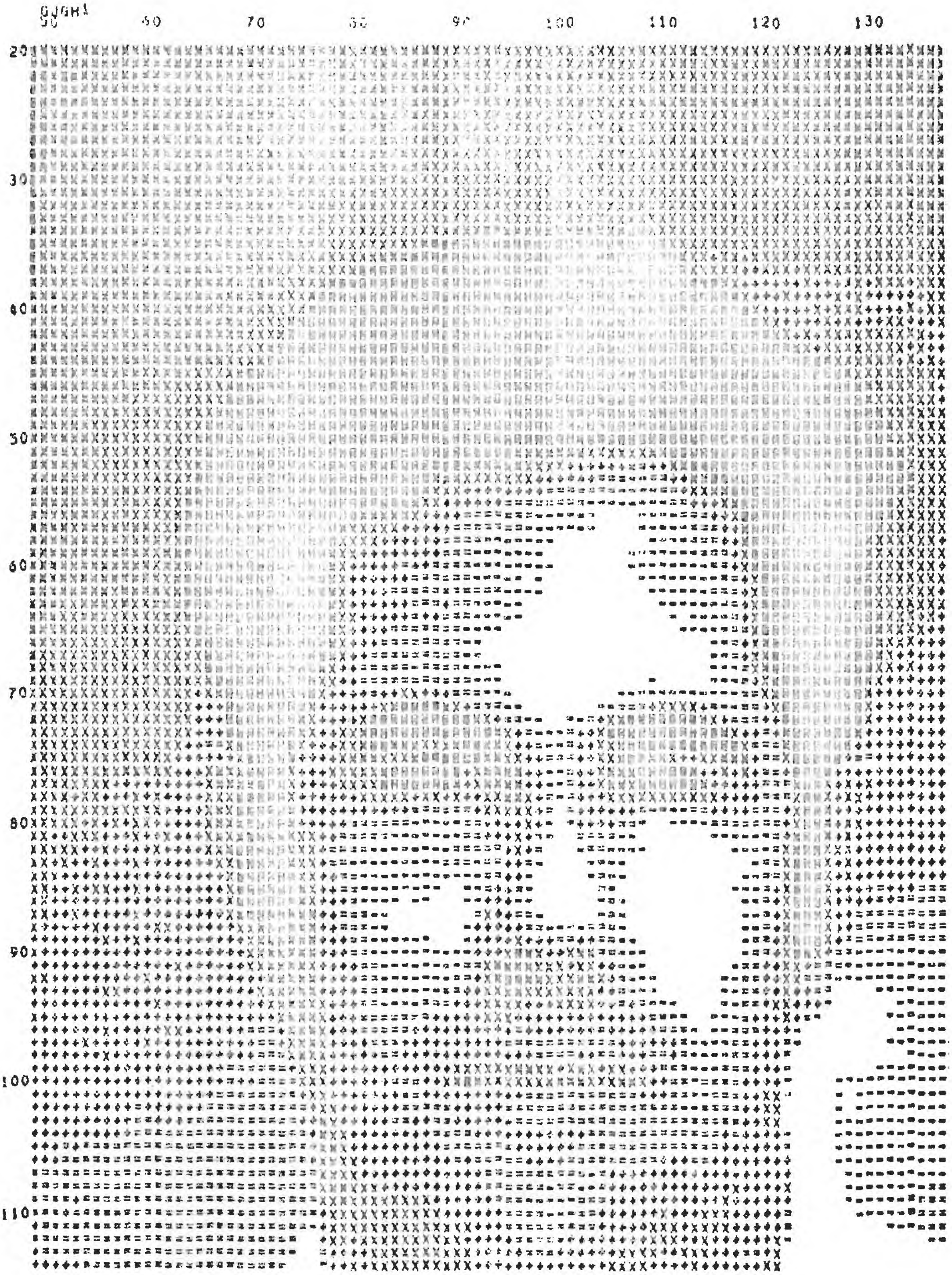


Figure 41. Input Picture GJGH1

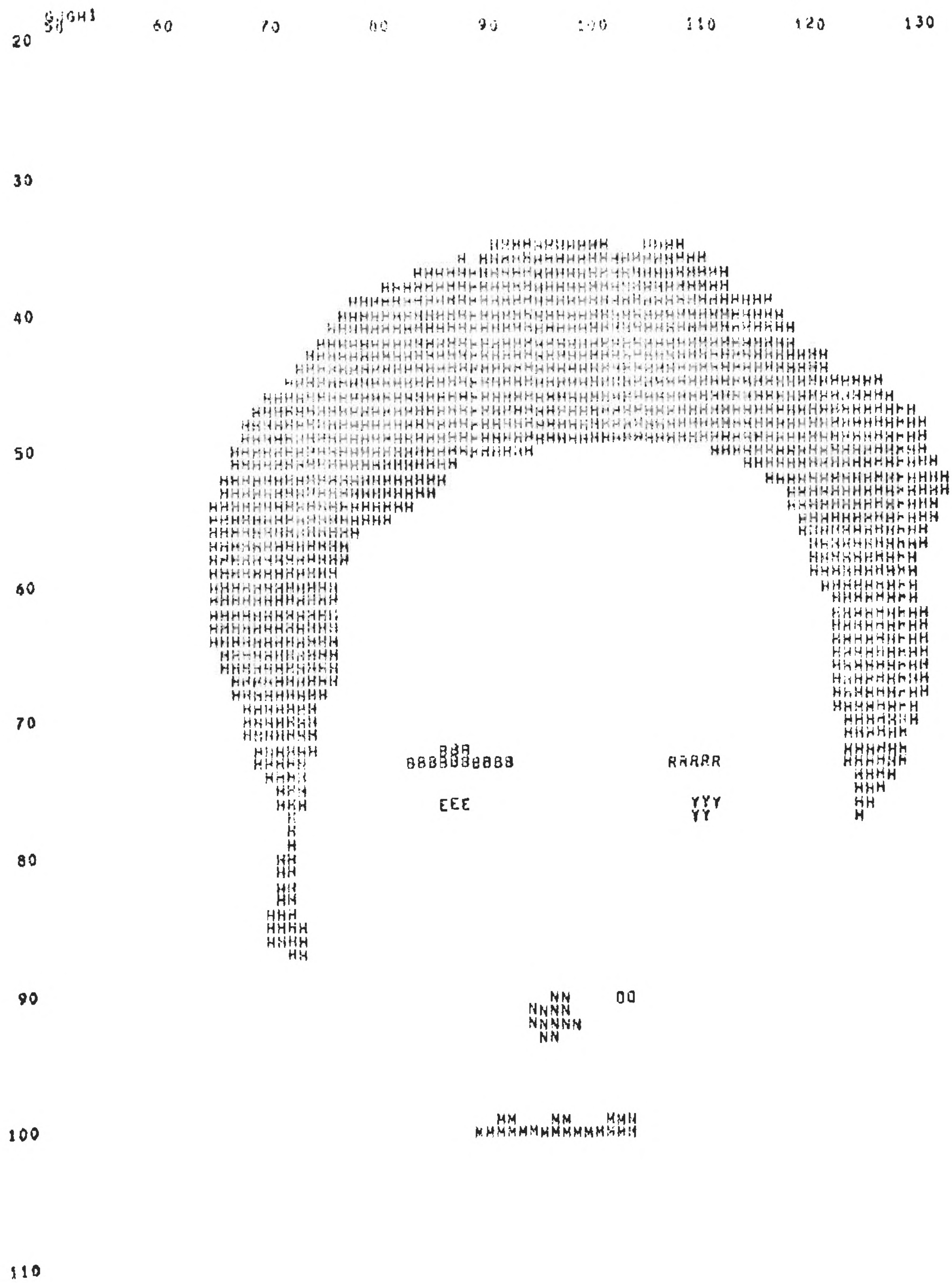


Figure 42. Features Recognized in GJGH1

Figure 43. Input Picture GJGH2

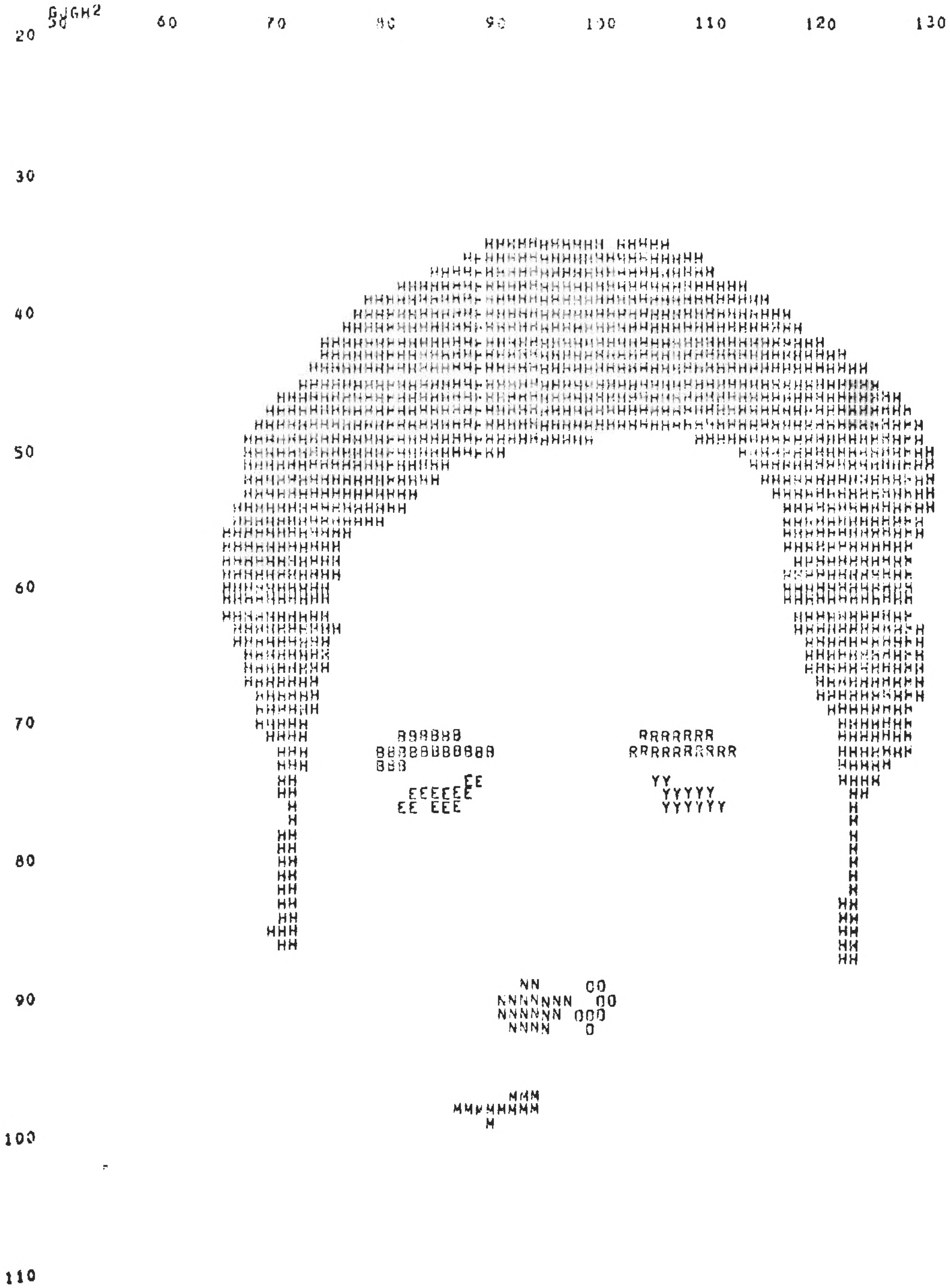


Figure 44. Features Recognized in GJGH2

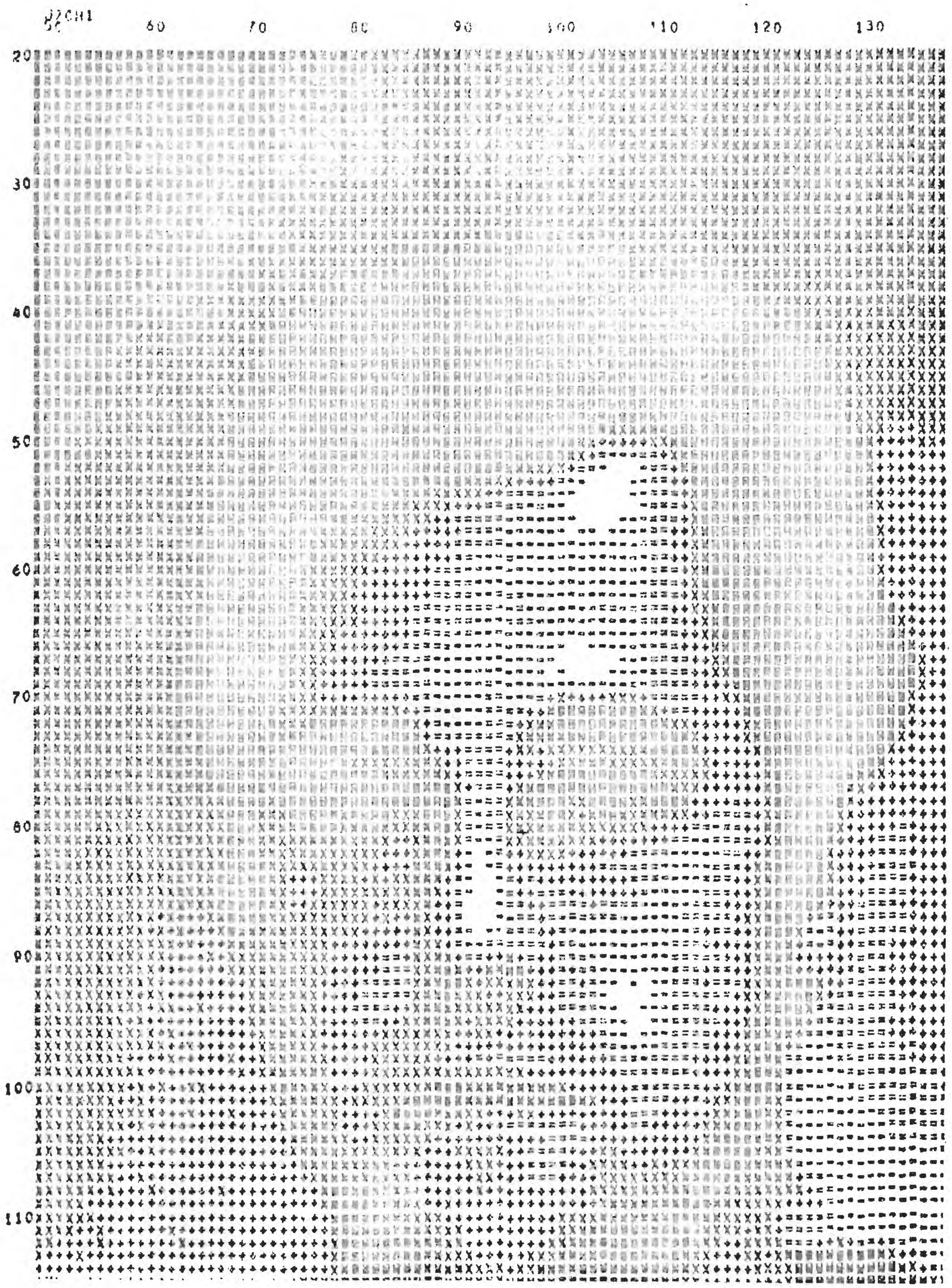


Figure 45. Input Picture JZCH1

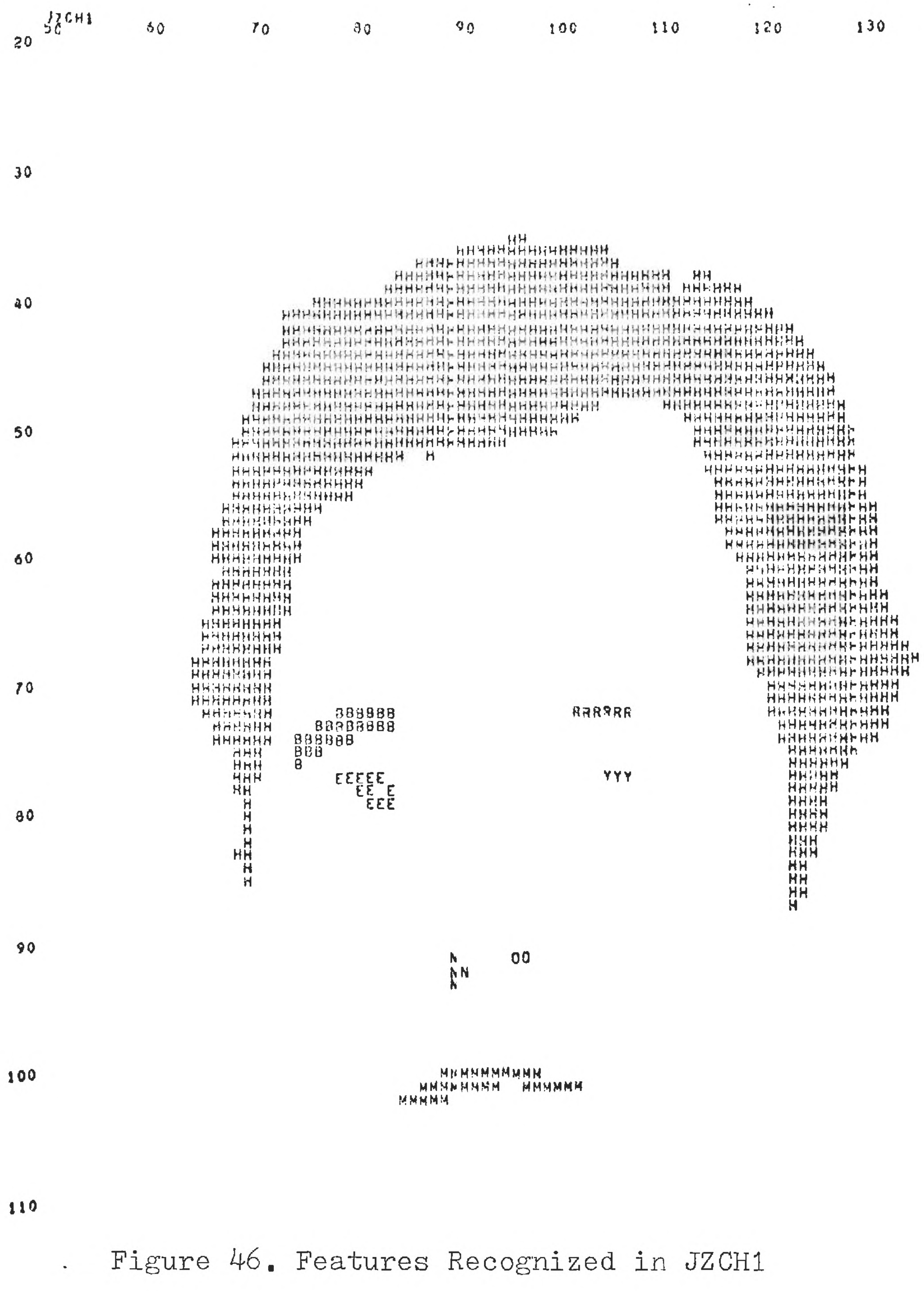


Figure 46. Features Recognized in JZCH1

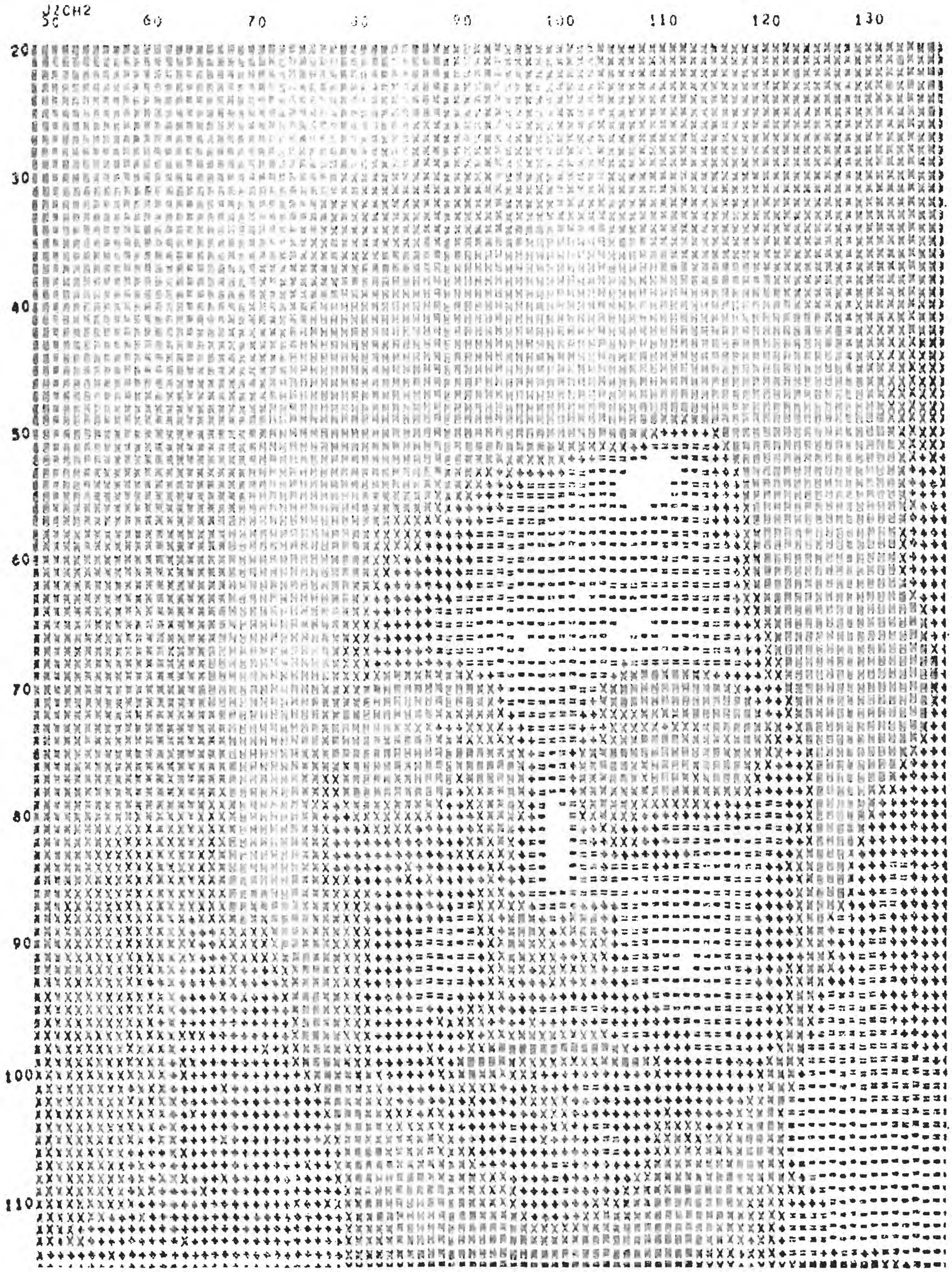


Figure 47. Input Picture JZCH2

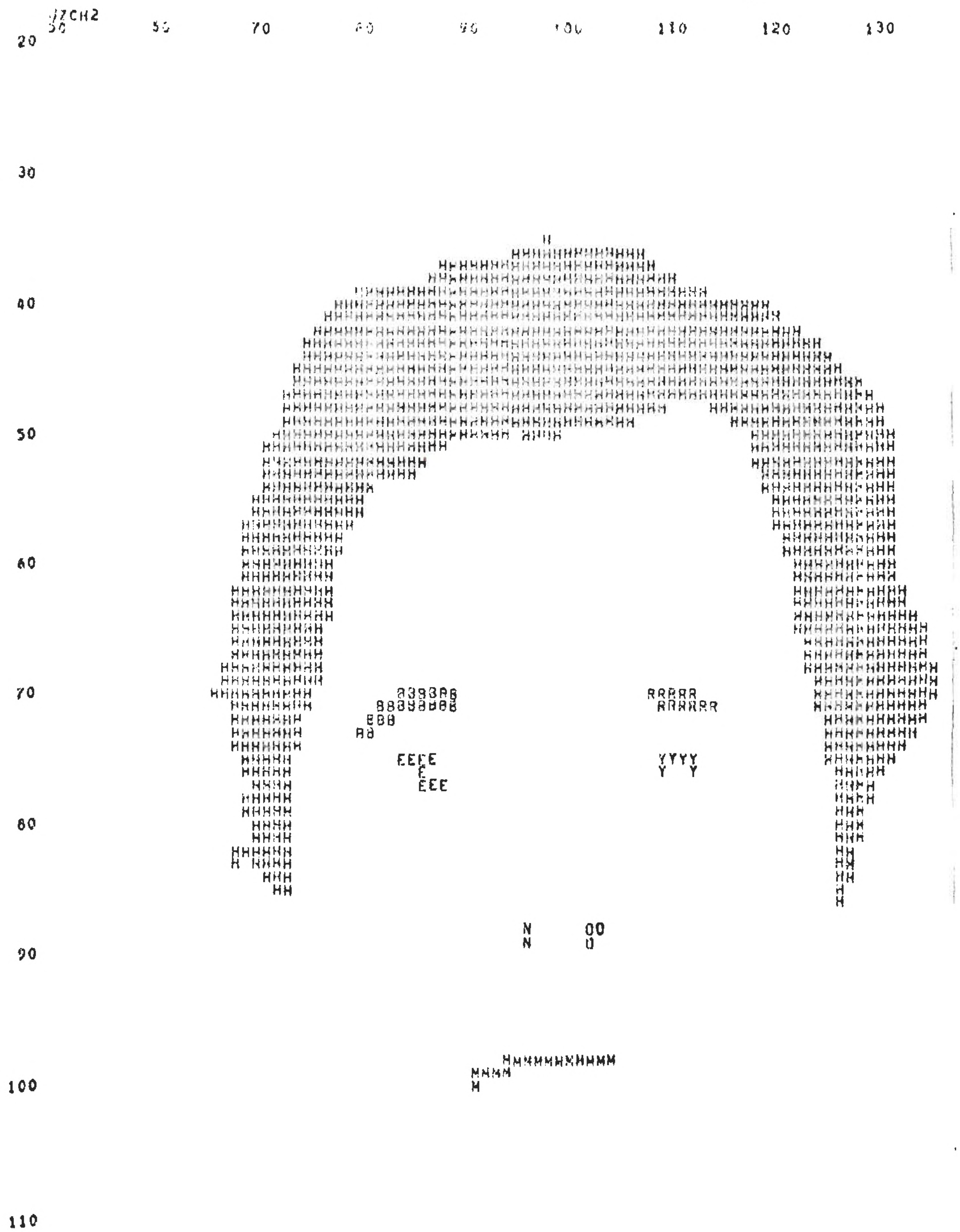


Figure 48. Features Recognized in JZCH2

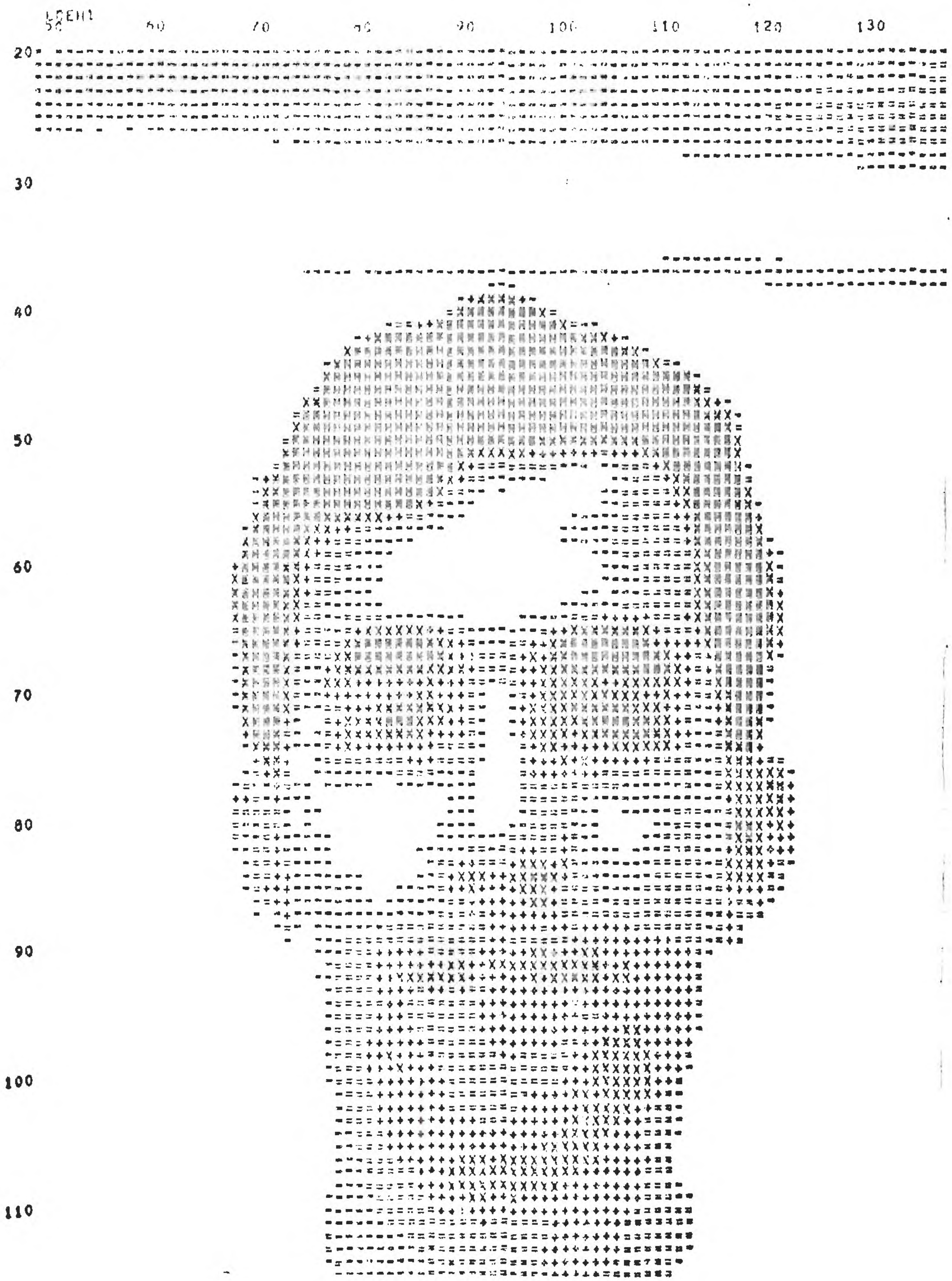


Figure 49. Input Picture LDEH1

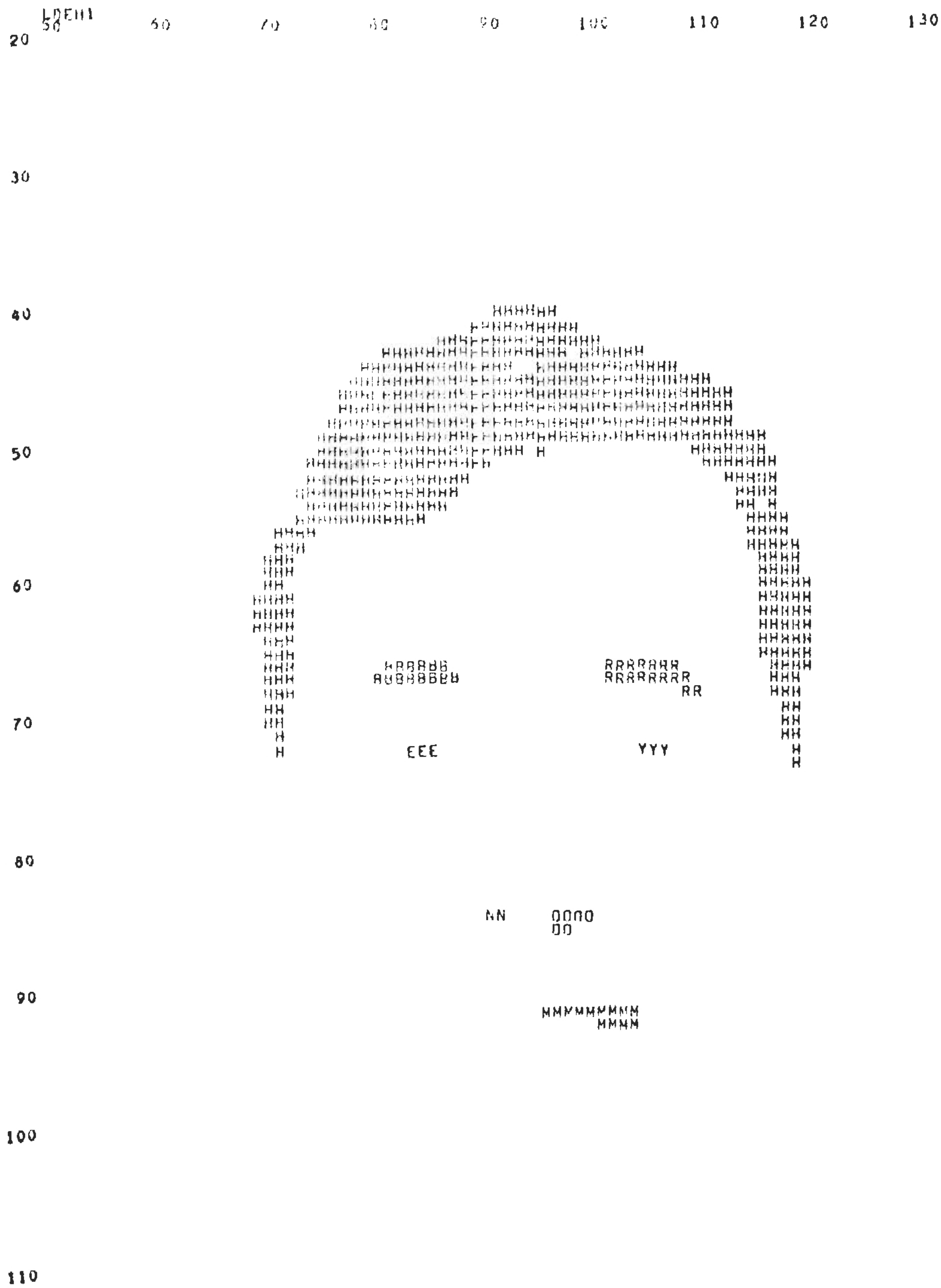


Figure 50. Features Recognized in LDEH1

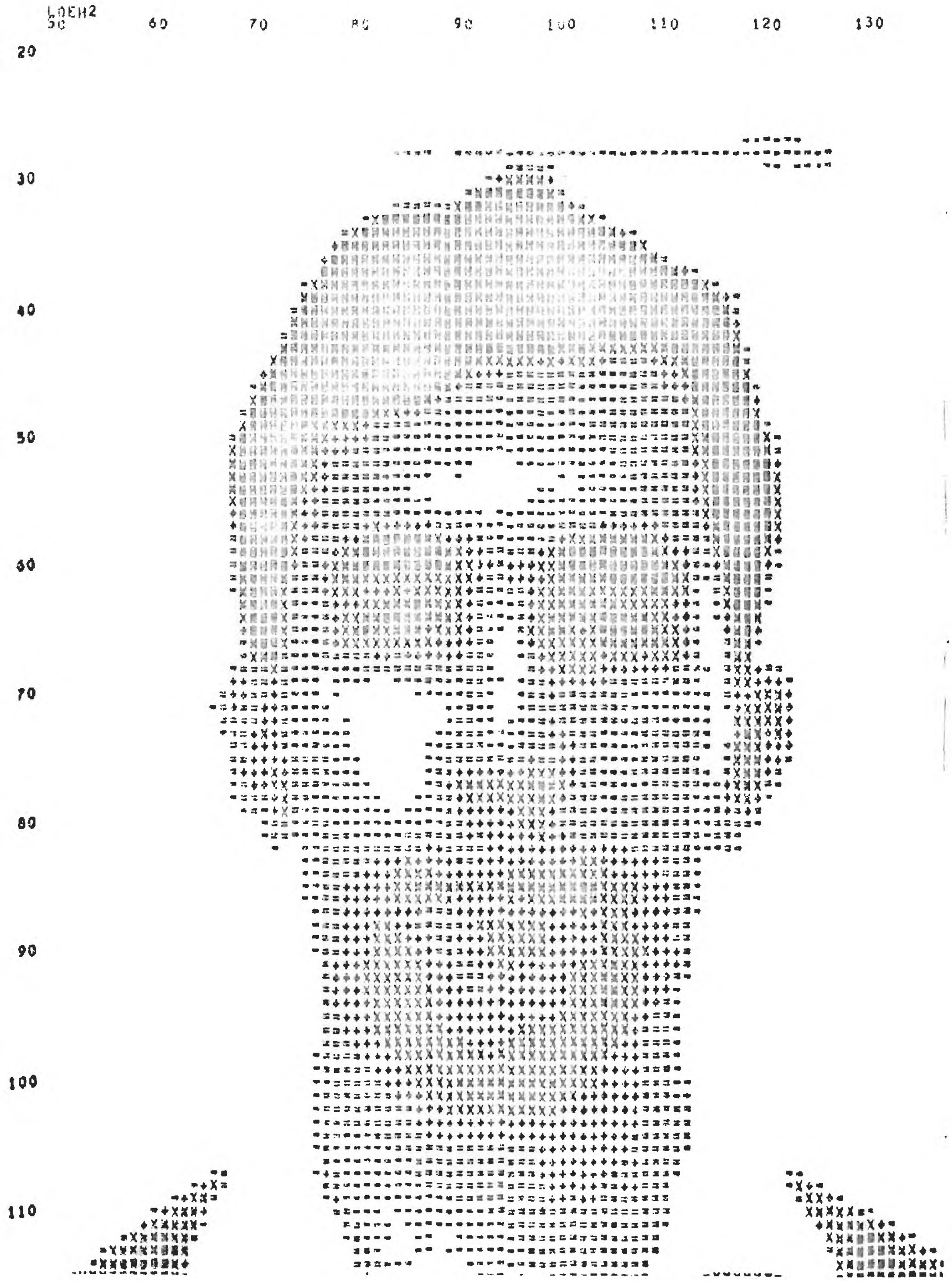


Figure 51. Input Picture LDEH2

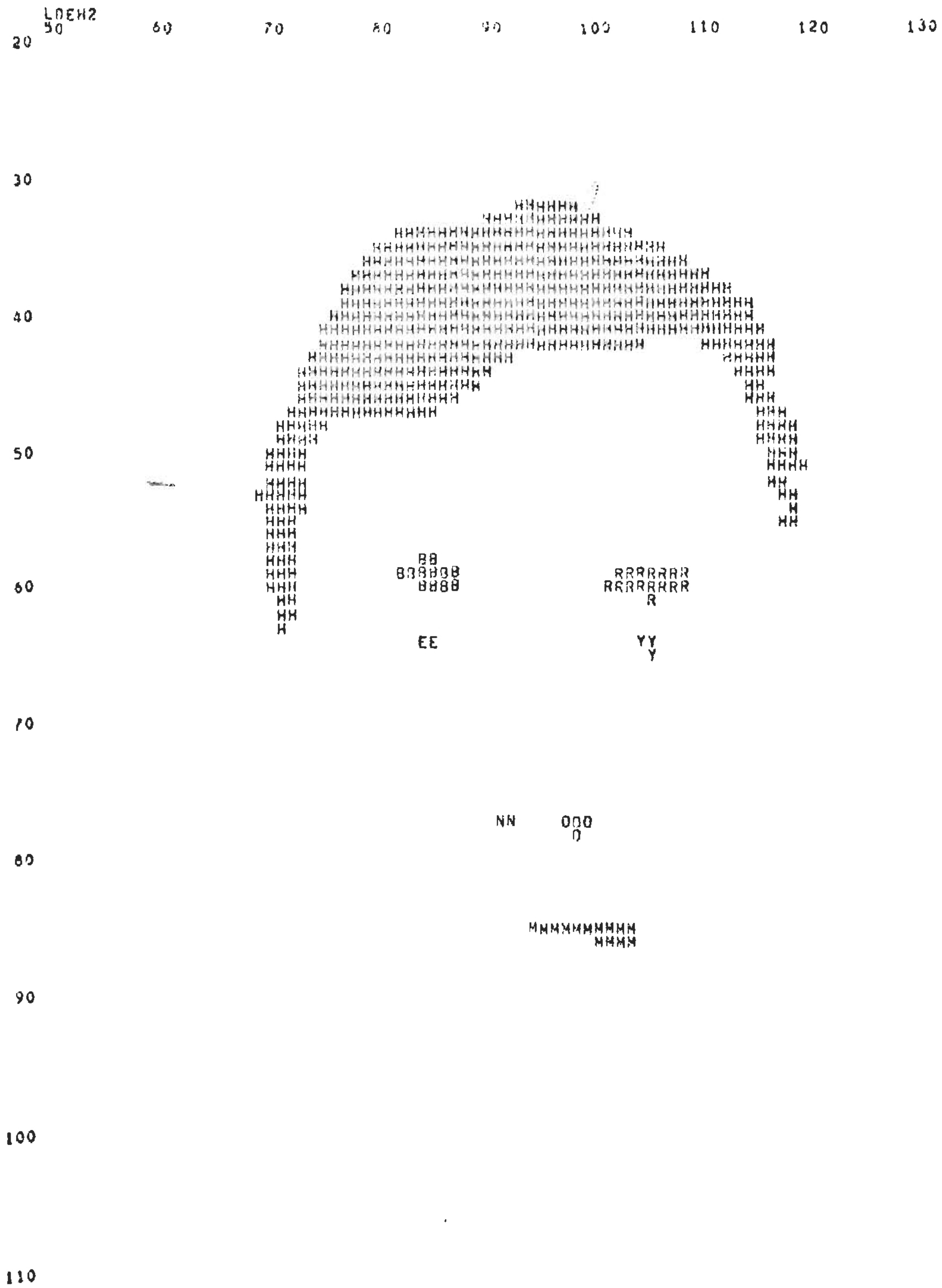


Figure 52. Features Recognized in LDEH2

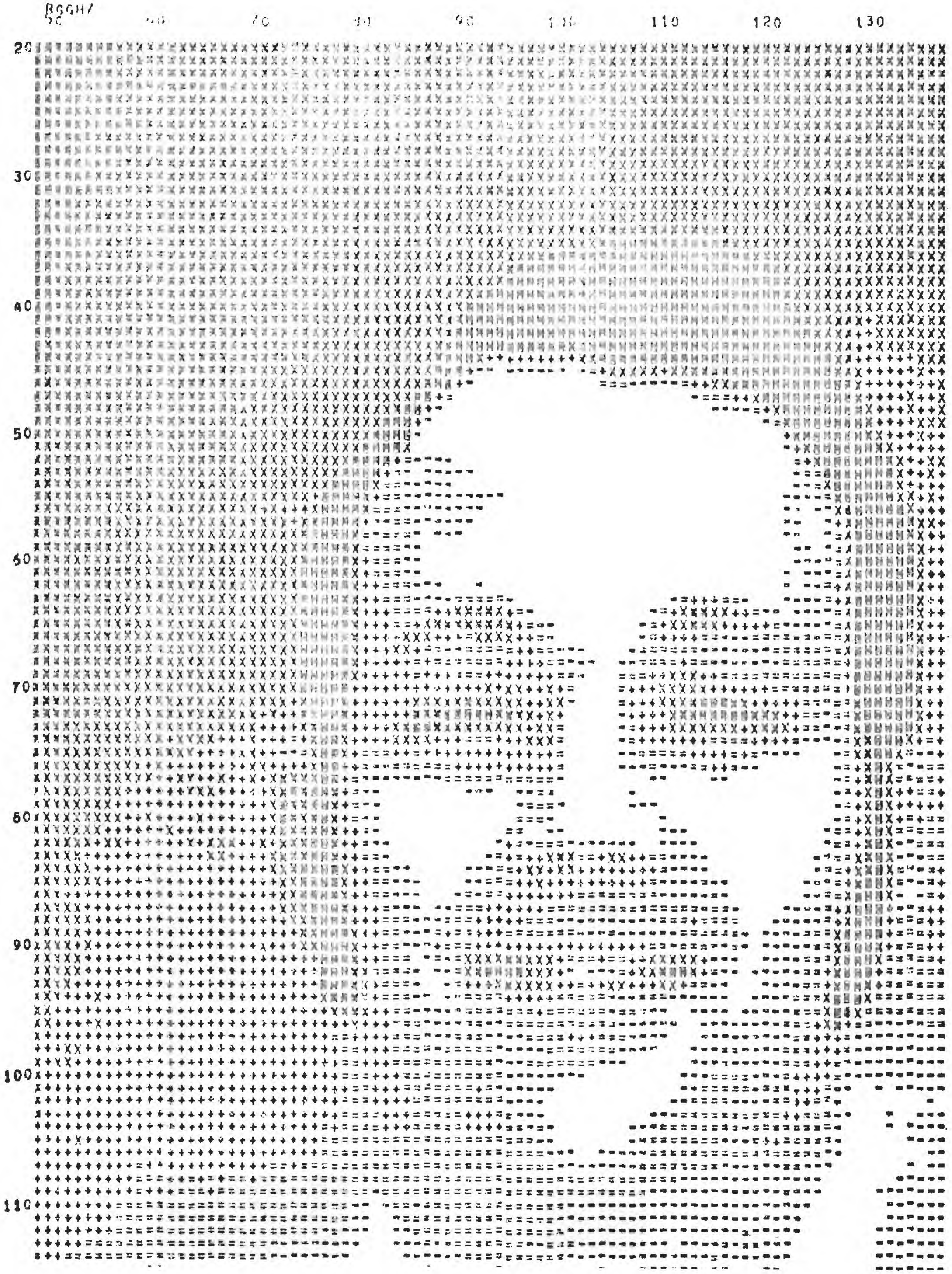


Figure 53. Input Picture RGGH7

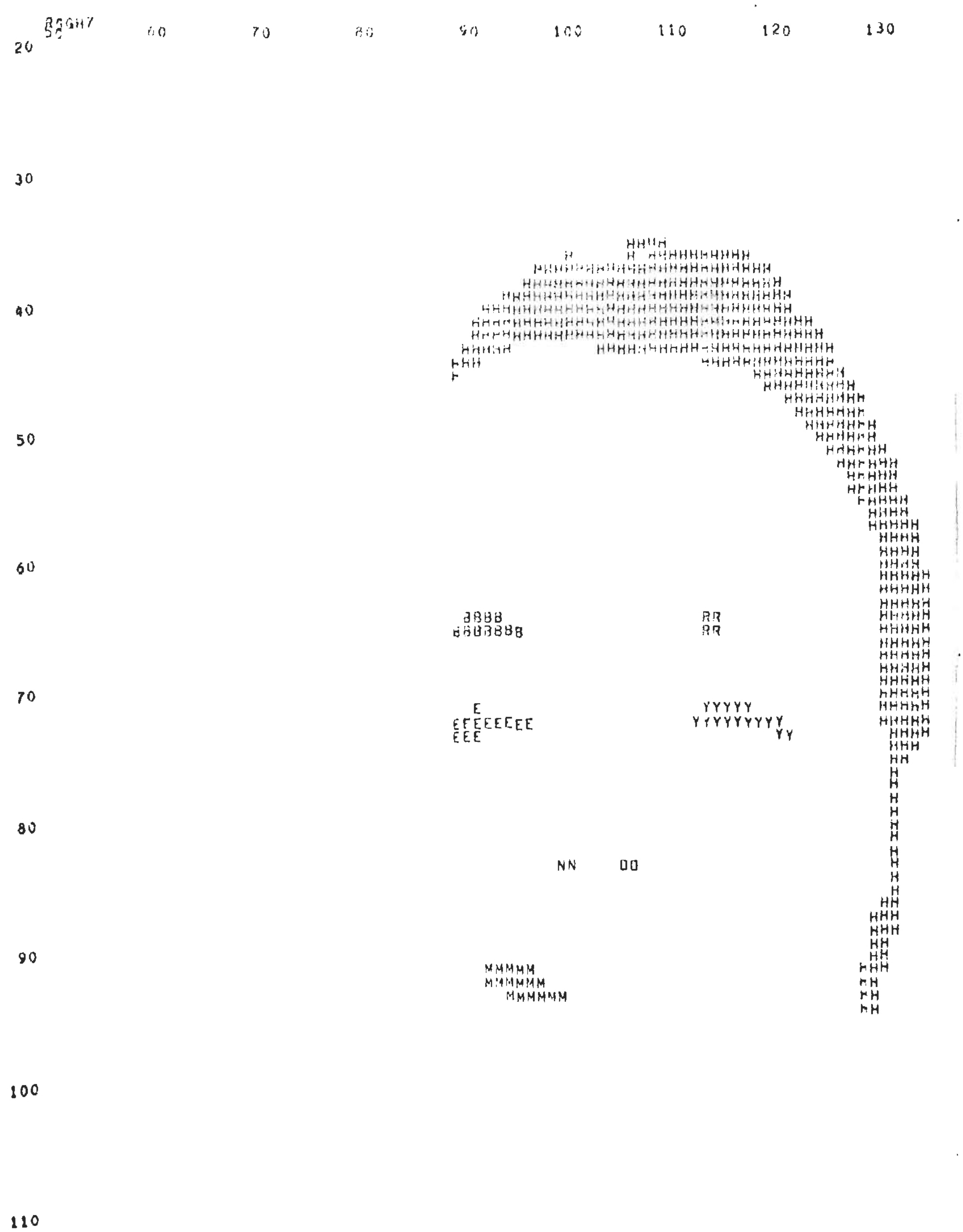


Figure 54. Features Recognized in RGGH7

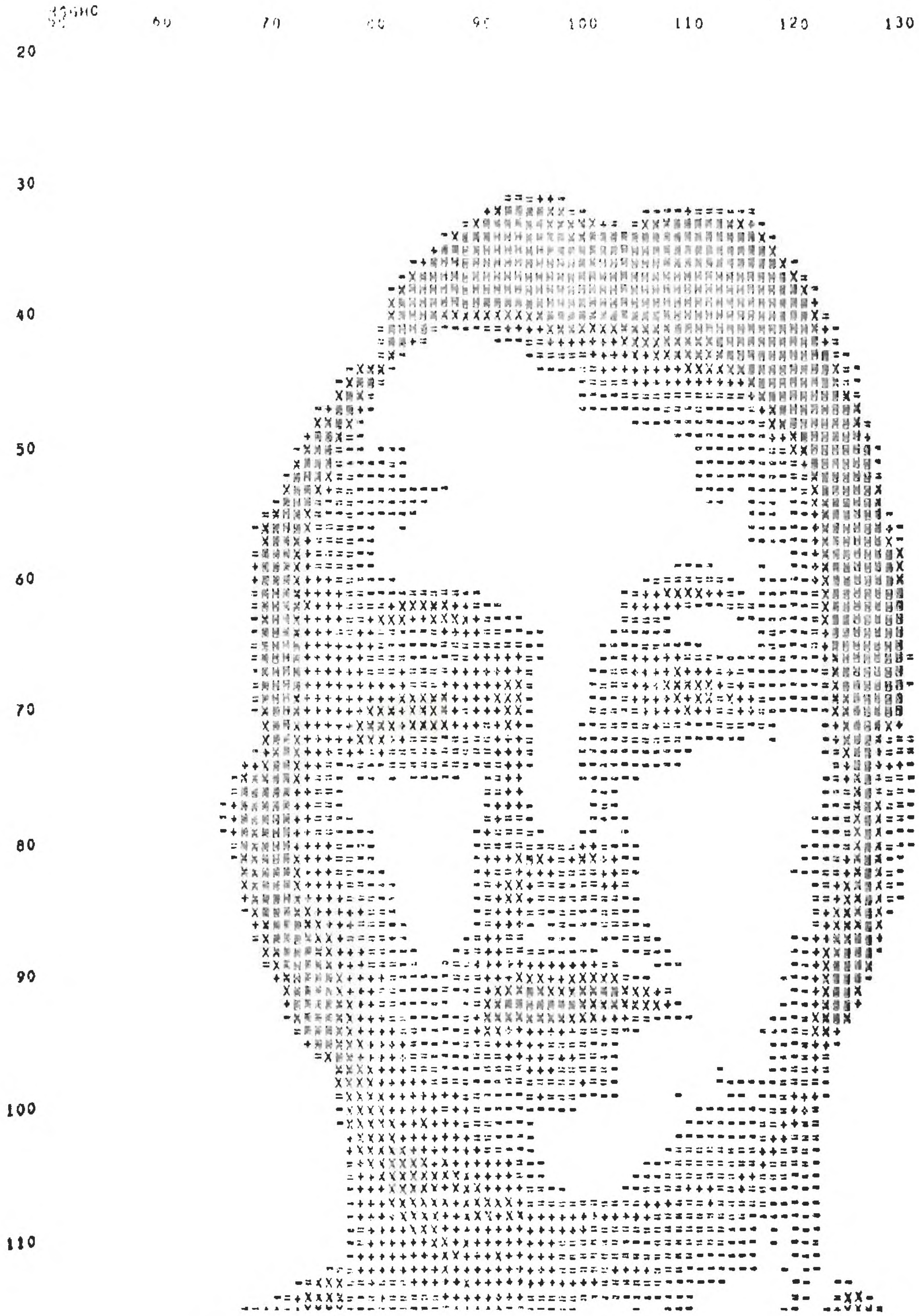


Figure 55. Input Picture RGGHC

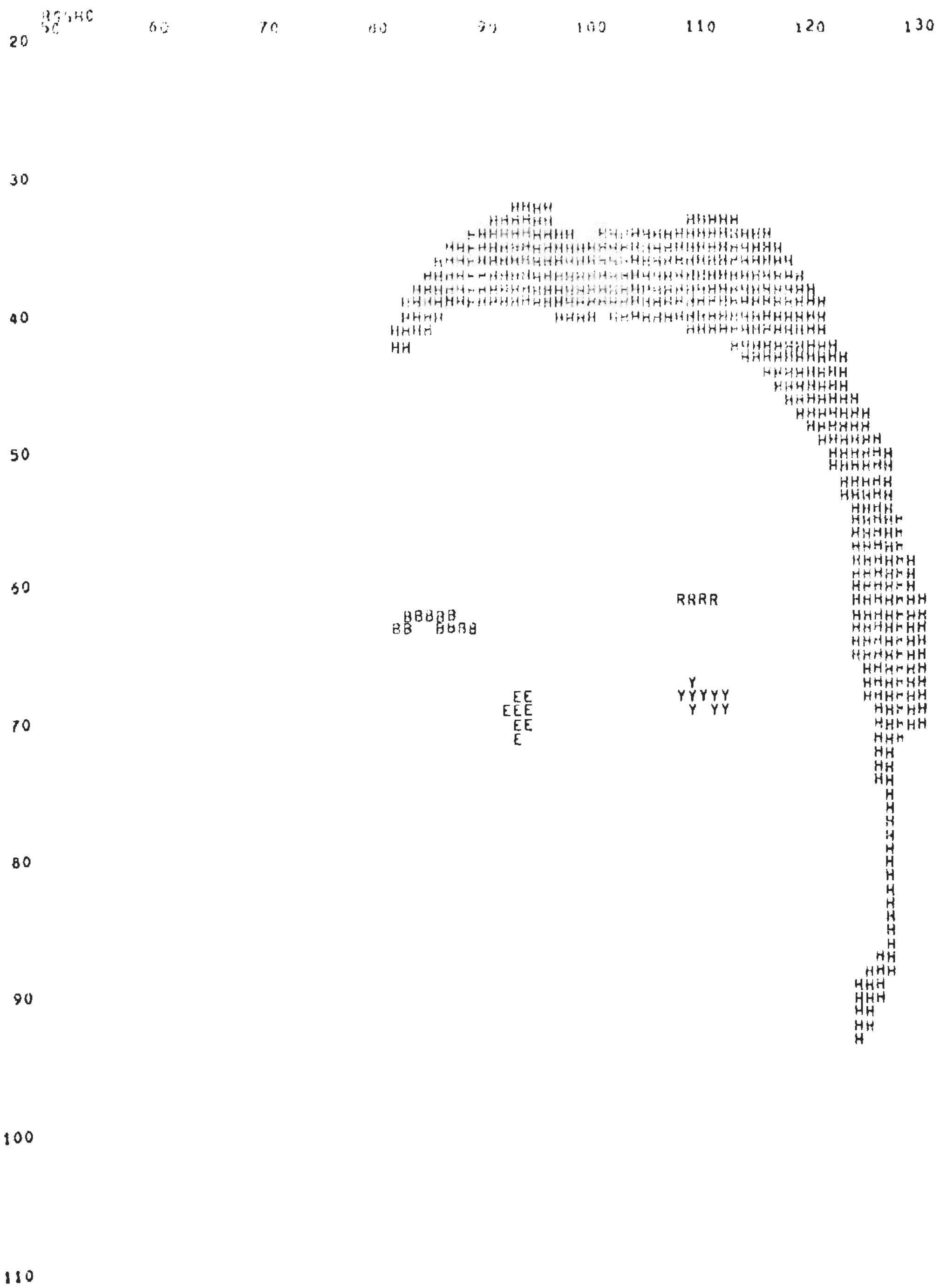


Figure 56. Features Recognized in RGGHC

Table 1. Exactness of Location of Regions of Features

Picture Name	Width of Head			Distance Between Eyes			Distance from Top of Head to Eyes			Distance from Eyes to Nostrils			Distance from Eyes to Mouth			
	A	K	B	A	K	B	A	K	B	A	K	B	A	K	B	LEGEND
EIMH1	63	67	63	21	22	23	40	44	40	15	12	-	23	22	-	A = actual measure
(error)		4	0		1	2		4	0		3	-		1	-	
EIMH2	64	66	62	23	21	25	40	44	39	12	11	15	23	21	24	K = Kelly's measure (1970)
(error)		2	2		2	2		4	1		1	3		2	1	
GJGH1	65	68	67	25	21	23	41	44	41	15	12	15	24	22	24	
(error)		3	2		4	2		3	0		3	0		2	0	B = Author's measure obtained using semantic paradigm solution
GJGH2	64	67	65	22	20	24	43	44	40	14	12	14	21	21	22	
(error)		3	1		2	2		1	3		2	0		0	1	
JZCH1	70	72	70	25	27	25	45	45	43	12	11	14	22	21	23	
(error)		2	0		2	0		0	2		1	2		1	1	
JZCH2	69	75	70	25	26	25	43	44	41	11	10	11	22	20	23	
(error)		6	1		1	0		1	2		1	0		2	1	
LDEH1	53	56	51	21	19	20	34	34	33	12	12	13	20	19	20	
(error)		3	2		2	1		0	1		0	1		1	0	
LDEH2	55	57	51	20	20	20	34	38	32	12	12	13	20	20	21	
(error)		2	4		0	0		4	2		0	1		0	1	
RGGH7	60	65	47	24	23	23	37	40	38	8	9	11	21	18	20	
(error)		5	13		1	1		3	1		1	3		3	1	
RGGHC	61	66	50	23	24	17	38	41	38	10	9	-	20	20	-	
(error)		5	11		1	6		3	0		1	-		0	-	
Average error		3.5	3.6		1.6	1.6		2.3	1.2		1.3	-		1.2		
Range of error		4	13		4	6		4	3		3			3		

7.3.2 Time Required to Write the Recognition Program

The semantic paradigm solution was developed and applied to the ten pictures in the data base within a matter of weeks. Since the semantic paradigm solution utilized none of the specialized picture processing techniques required by Kelly (e.g., template matching, edge detection operators, planning, dynamic threshold setting, line detection operators, etc.) this short development time was not due to a carry-over of any insight into the problem solution. Kelly's program for the solution of this problem required a matter of years rather than weeks of development time. Even though Kelly was concerned with obtaining more exact location information for features, and also had to deal with the classification aspect of the problem as well as cope with the problem of obtaining a data base, the extreme differences in development times makes it obvious that the semantic paradigm solution is preferable over an ad hoc, heuristic program.

7.3.3 Time Required to Execute the Program

In the development of the semantic paradigm solution, little effort was made to attain efficient coding. Nevertheless, only about three to five minutes was required to apply the program to a picture in the data base and obtain a printout of the locations of the features. Since this program was implemented on a B5500 computer, it would be difficult to compare execution time with Kelly's program implemented on a much faster PDP10 computer. In any case, it was found that without any effort at optimization, the semantic paradigm solution could be executed in a reasonable amount of time.

7.3.4 Length of the Program

The semantic paradigm program, including many comment cards, was about 700 cards in length, or nine pages. In contrast, we find that one of Kelly's subroutines for finding the outline of a head consists of over 30 pages of coding. It is obvious that the shorter program was easier to write and debug.

7.3.5 Conclusions

The successful application of the semantic paradigm to this problem involving non-ideal data represents a significant achievement in picture processing. This follows since no other general model or paradigm is found to be useful for handling problems of this type.

CHAPTER VIII

APPLICATION OF THE PARADIGM FOR A PROBLEM INVOLVING MULTISTABILITY IN PERCEPTION

8.1 The Problem

For this problem, an effort is made to illustrate an analogy between multistability as it applies to the reversing cube, and the dynamics of the semantic paradigm. The problem is to show that the application of the rules of inference results in the construction of a non-primitive scene description which contains one "principal aspect" description of the cube. Following this, continued application of the rules are to result in the inference of the alternative "principal aspect," thus resulting in a contradiction and the deletion of the previous description.

8.2 The Semantic Paradigm Solution

The set of qualities of interest to an information processor attempting to recognize the reversing cube was established to be the following.

$\overline{Q}_I = \{Q_1, Q_2, Q_3, Q_4\}$, where

Q_1 = Feature = {Necker cube ("NECKER"), Face of cube ("FACE")}.

Q_2 = Vertex type = {Shaped like a Y (abbreviated "YY"), Shaped like an inverted Y (abbreviated "NEGYY")}.

Q_3 = Type of cube perceived = {Seen from the bottom (abbreviated "CUBEB"), Seen from the top (abbreviated "CUBEA")}.

Q_4 = Face type = {Face belonging to cube of type "CUBEA", Face belonging to cube of type "CUBEB"}.

The reader will note that we have established no properties such as light intensity values, etc., usually found in $\overline{Q}_I$. The reason for this is that, for clarity, we assume that the information processor has certain preprocessing abilities not normally encountered.

We shall assume that I's sensory and preprocessing facilities are such that the following set of qualities can be said to be primitive.

$\overline{Q}'_I = \{Q_1, Q_2\}$.

Thus, because we assume that Q_1 is a primitive quality, it follows that an object having the shape of a Necker cube is recognizable by I as a primitive object, through, for example, a template matching preprocessing operation.

The set of relations of interest to an information processor attempting to recognize the reversing cube was established to be the following.

$\overline{R}_I = \{R_1, R_2, R_3, R_4\}$, where

R_1 = "__ is above __,"

R_2 = "__ is below __,"

R_3 = "__ is a vertex of face __," and

R_4 = "__ is to the right of __."

We shall assume that I's sensory and preprocessing facilities are such that the following set of relations can be said to be primitive.

$\overline{R}'_I = \overline{R}_I$.

8.2.1 The Primitive Scene Description

The primitive scene description for the reversible cube illustrated in Figure 57 is as follows, where the regions occupied by the objects are indicated by the labels in the picture.

$PD(S) = \{PSD^t(S)\}$, where

$PSD^t(S) = \langle \hat{O}(S_t); \bar{R}'_I(t) \rangle$, where

$\hat{O}(S_t)$ = {L1-L12 = {"NECKER"}, V1 = {"YY"}, V2 = {NEGYY"}, L1,L2,L3,L4 = {"FACE"}, L1,L8,L9,L5 = {"FACE"}, L3,L5,L7,L10 = {"FACE"}, L9,L10,L11,L12 = {"FACE"}, L2,L6,L8,L11 = {"FACE"}, L4,L6,L7, L12 = {"FACE"}}.

$\bar{R}'_I(t) = \{R_1, R_2, R_3, R_4\}$, where

R_1 = {(L3,L5,L7,L10 ; L1,L8,L9,L5), (L4,L6,L7,L12 ; L2,L6,L8,L11)},

R_2 = {(L1,L8,L9,L5 ; L9,L10,L11,L12), (L3,L5,L7,L10 ; L9,L10,L11, L12), (L1,L2,L3,L4 ; L2,L6,L8,L11), (L1,L2,L3,L4 ; L4,L6,L7, L12)},

R_3 = {(L9,L10,L11,L12 ; L1,L8,L9,L5), (L9,L10,L11,L12 ; L3,L5,L7, L10), (L2,L6,L8,L11 ; L1,L2,L3,L4), (L4,L6,L7,L12 ; L1,L2, L3,L4)}, and

R_4 = {(V1 ; L1,L2,L3,L4), (V1 ; L2,L6,L8,L11), (V1 ; L4,L6,L7,L12), (V2 ; L1,L8,L9,L5), (V2 ; L3,L5,L7,L10), (V2 ; L9,L10,L11, L12)}.

8.2.2 Rules of Inference

It was found that four simple rules of inference were adequate for exhibiting multistability in the construction of non-primitive scene descriptions from a primitive scene description of a necker cube. They are given below in natural language for ease of reading. The reader is directed to the Appendix for a listing of the recognition program which implemented this semantic paradigm solution.

RULE 0: If there exist objects v,w,x,y,z, such that (1) v is a necker cube, (2) w is a vertex of type inverted Y, (3) w is a vertex of x,y, and z, (4) x is below y and z, and (5) y is to the right of z, then (1) v is a cube of type "CUBEB," and (2)x,y, and z are faces of a cube of type "CUBEB."

RULE 1: If there exist objects w,x,y,z such that (1) w is a cube of type "CUBEA" and "CUBEB," and (2) x,y, and z are faces of a cube of type "CUBEA" then (1) w is no longer a cube of type "CUBEA," and (2) x,y, and z are no longer faces of a cube of type "CUBEA." The reader will note that this rule follows since no two properties in any one quality set may apply at the same time to any one object.

RULE 2: If there exist objects v,w,x,y,z such that (1) v is a necker cube, (2) w is a vertex of type Y, (3) w is a vertex of x,y, and z, (4) x is above y and z, and (5) y is to the right of z, then (1) v is a cube of type "CUBEA," and (2) x,y, and z are faces of a cube of type "CUBEA."

RULE 3: If there exist objects w,x,y,z such that (1) w is a cube of type "CUBEA" and "CUBEB," and (2) x,y, and z are faces of a cube of type "CUBEB," then (1) w is no longer a cube of type "CUBEB," and (2) x,y, and z are no longer faces of a cube of type "CUBEB." The reader will note that this rule (like rule 1) follows since no two properties in any one quality set may apply at the same time to any one object.

8.3 Results

The result of applying the semantic paradigm solution to this problem is illustrated in Figure 57. Shown first is an illustration of the picture from which the primitive scene description was obtained in terms of the primitive properties and relations enumerated in the previous section. Figure 57 also illustrates the sequence of non-primitive scene descriptions inferred through the application of the rules of inference. As can be seen, first one view of the cube is "perceived," then the picture reverses, resulting in the other view being "perceived" and the previous view lost. This reversal continues indefinitely, as it does with a human information processor. The program terminates after an "excess time" message clues the program that it is in a loop (i.e., in a state of multi-stability).

The program for the solution to this problem is included in the Appendix for the reader interested in determining further how the semantic paradigm solution is implemented in a programming language (ALGOL). The reader will note that the proof of the inquiry portion of a rule of inference is achieved through depth-first search of the objects in the scene description. Thus, this problem solution is completely general.

```
IN THIS PROBLEM, THE SEMANTIC PARADIGM IS APPLIED TO A PROBLEM
INVOLVING MULTISTABILITY IN PERCEPTION.  SPECIFICALLY, THE
PROBLEM INVOLVES THE REVERSAL EFFECT ENCOUNTERED BY HUMAN
INFORMATION PROCESSORS IN ATTEMPTING TO RECOGNIZE A TRANSPARENT
CUBE (NECKER CUBE).  IT IS SHOWN THAT THE SEMANTIC PARADIGM
EXHIBITS AN ANALOGOUS KIND OF BEHAVIOR IN WHICH FIRST ONE
DESCRIPTION OF THE SCENE IS CONSTRUCTED WHICH CONTAINS A
DESCRIPTION OF THE CUBE VIEWED FROM ONE PRINCIPAL ASPECT (A)
AND THEN CONTINUED APPLICATION OF THE RULES OF INFERENCE RESULTS
IN THE DELETION OF THE PREVIOUS DESCRIPTION OF THE CUBE AND
REPLACES IT WITH ANOTHER  VIEWED FROM ANOTHER PRINCIPAL ASPECT
(B).  THIS REVERSAL CYCLE CONTINUES INDEFINITELY, AS IT DOES IN A
HUMAN INFORMATION PROCESSOR.
     THE INPUTED PRIMITIVE SCENE DESCRIPTION IS IN TERMS OF
THE FOLLOWING 9 OBJECTS:
OBJECT1 (F1),...,OBJECT6 (F6) ARE FACES OF OBJECT9 (NECKERCUBE)
OBJECT 7 (V1) AND OBJECT8 (V2) ARE VERTICES OF TYPE Y (YY) AND
INVERTED Y (NEGYY) RESPECTIVELY.
THE SINGLE QUALITY IN Q BAR PRIME IS Q1 = SHAPE=
(NECKER CUBE,FACE OF A CUBE, TYPE Y VERTEX,TYPE INVERTED Y VERTEX
VERTEX).   WHERE Q BAR = (Q1, Q2) AND
Q2 = CUBE TYPE = (CUBEA, CUBEB).  I.E., THE TWO VIEWS OF THE CUBE
TO BE RECOGNIZED
     THE RELATIONS INVOLVED ARE R = R PRIME = (R1,R3,R4,R5)
WHERE
R1 = --IS TO THE RIGHT OF, DISJOINT, AND BOUNDARY-CONNECTED TO--
R3 = --IS ABOVE          , DISJOINT, AND BOUNDARY-CONNECTED TO--
R4 = --IS BELOW          , DISJOINT, AND BOUNDARY-CONNECTED-TO--
R5 = --IS A VERTEX ON THE BOUNDARY OF FACE--
```

Figure 57. Results of Applying the Semantic Paradigm Solution to a Problem Involving Multistability in Perception

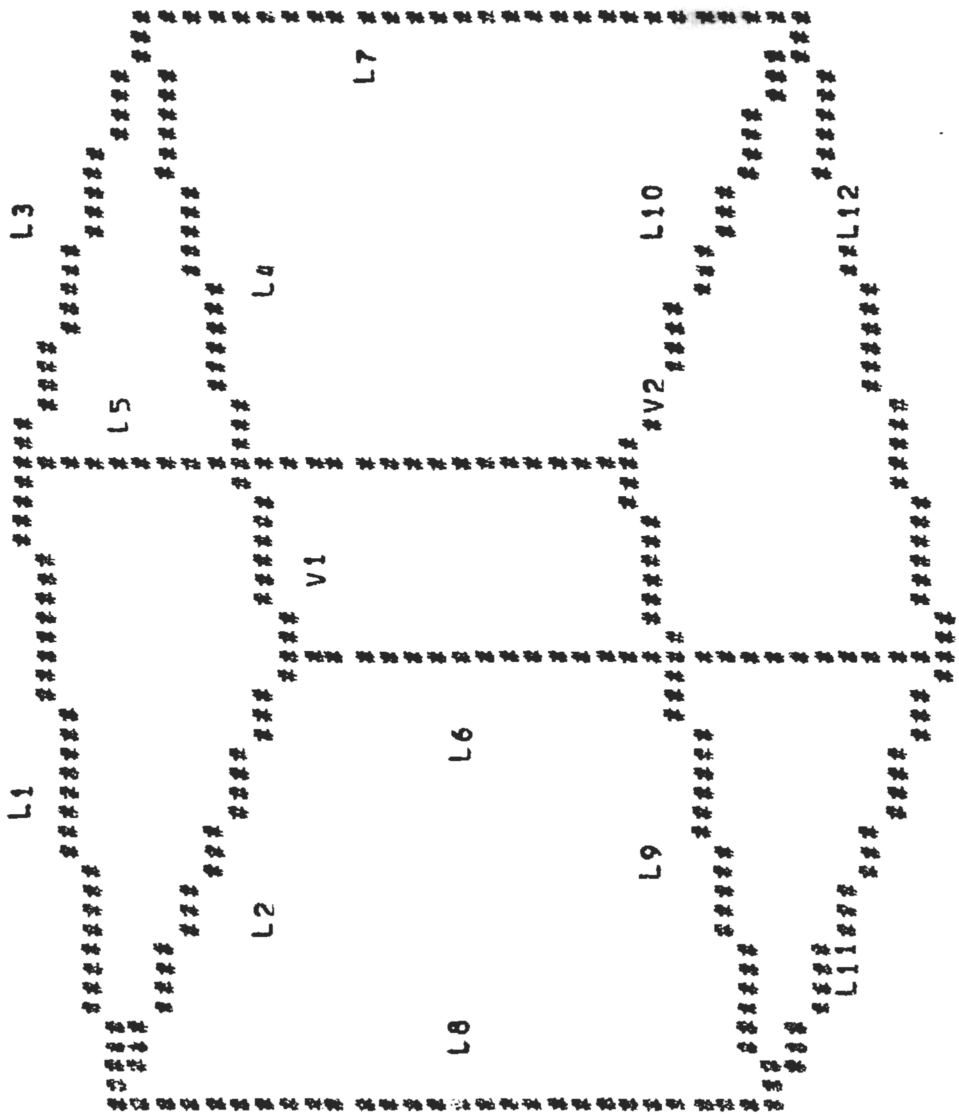
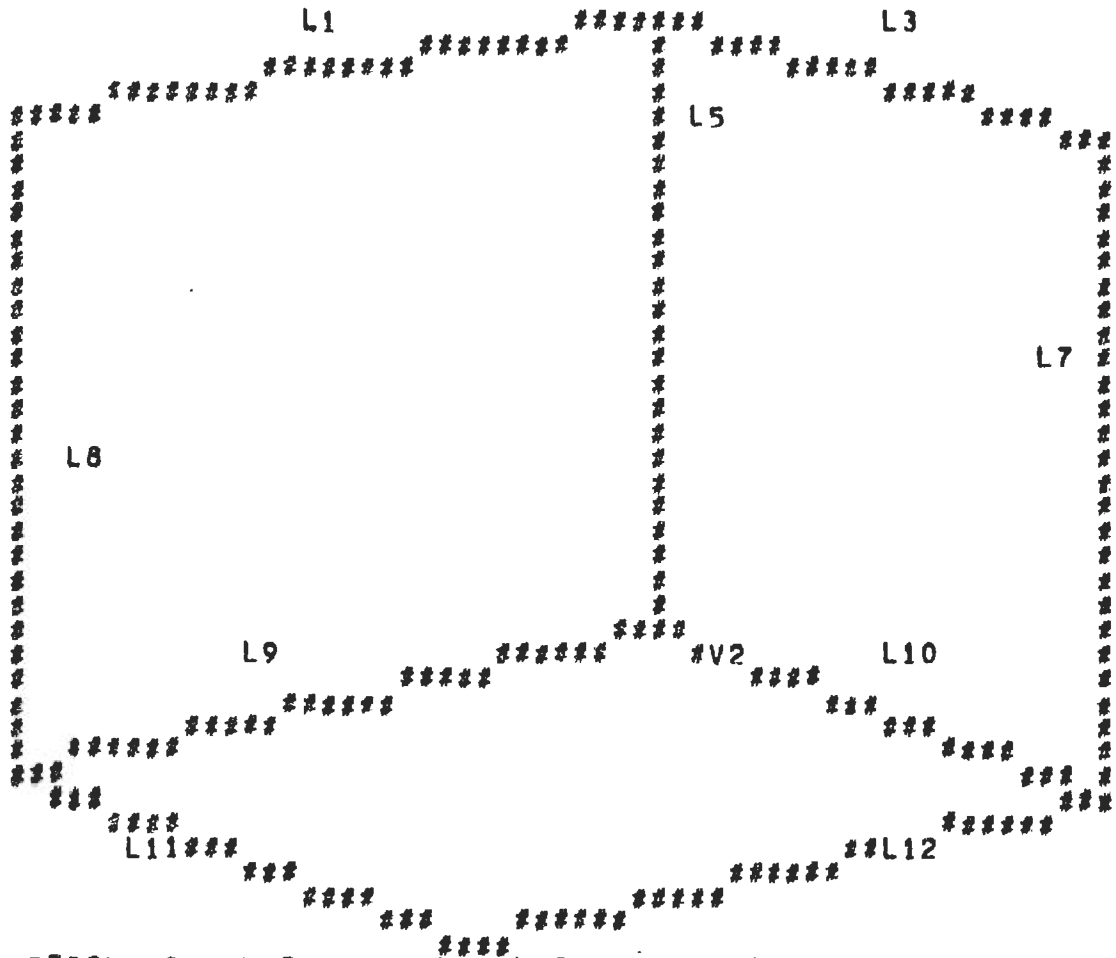


Figure 57. (continued)

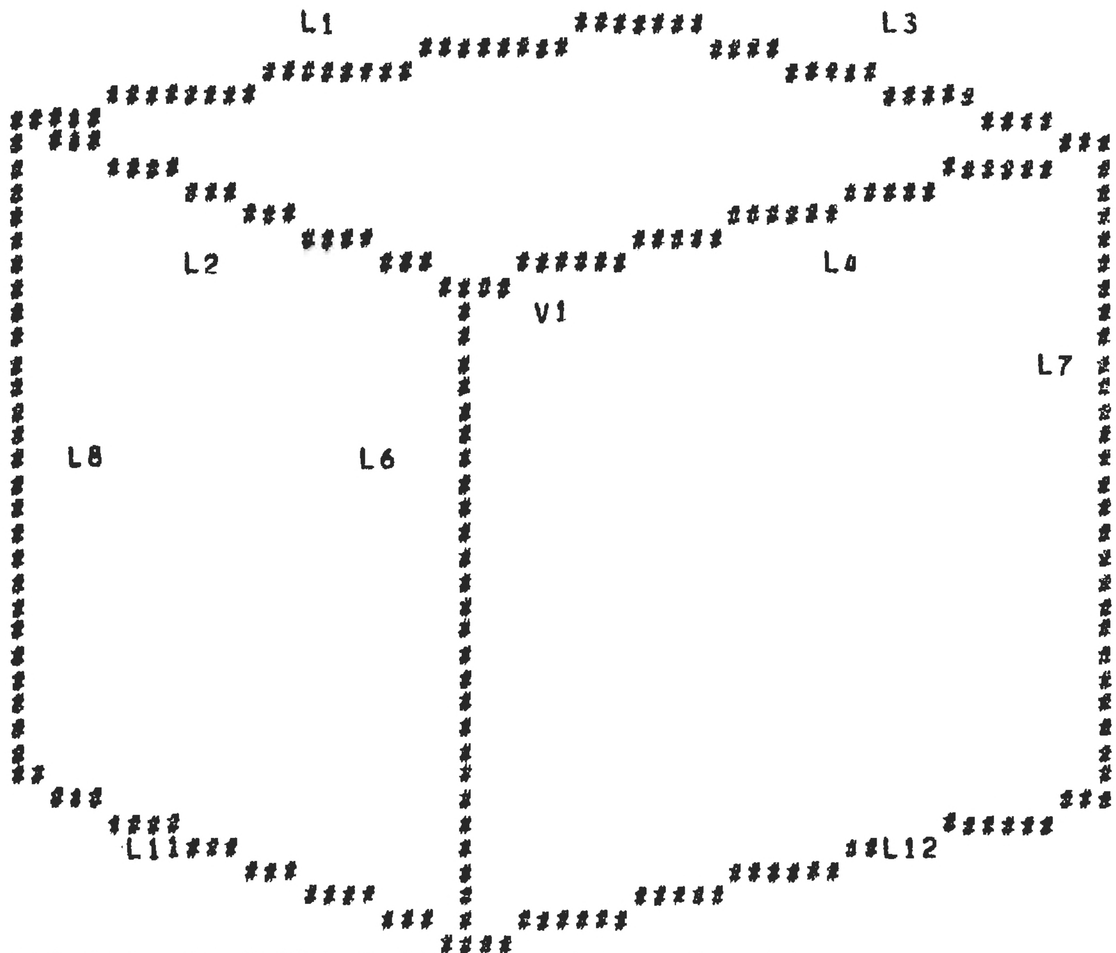
Figure 57. (continued)



BOTTOM IS BOUNDED BY L9 L10 L12 L11

RIGHTSIDE IS BOUNDFD BY L3 L5 L10 L7

LEFTSIDE IS BOUNDED BY L1 L8 L9 L5



TOP SIDE IS BOUNDED BY L1 L2 L4 L3

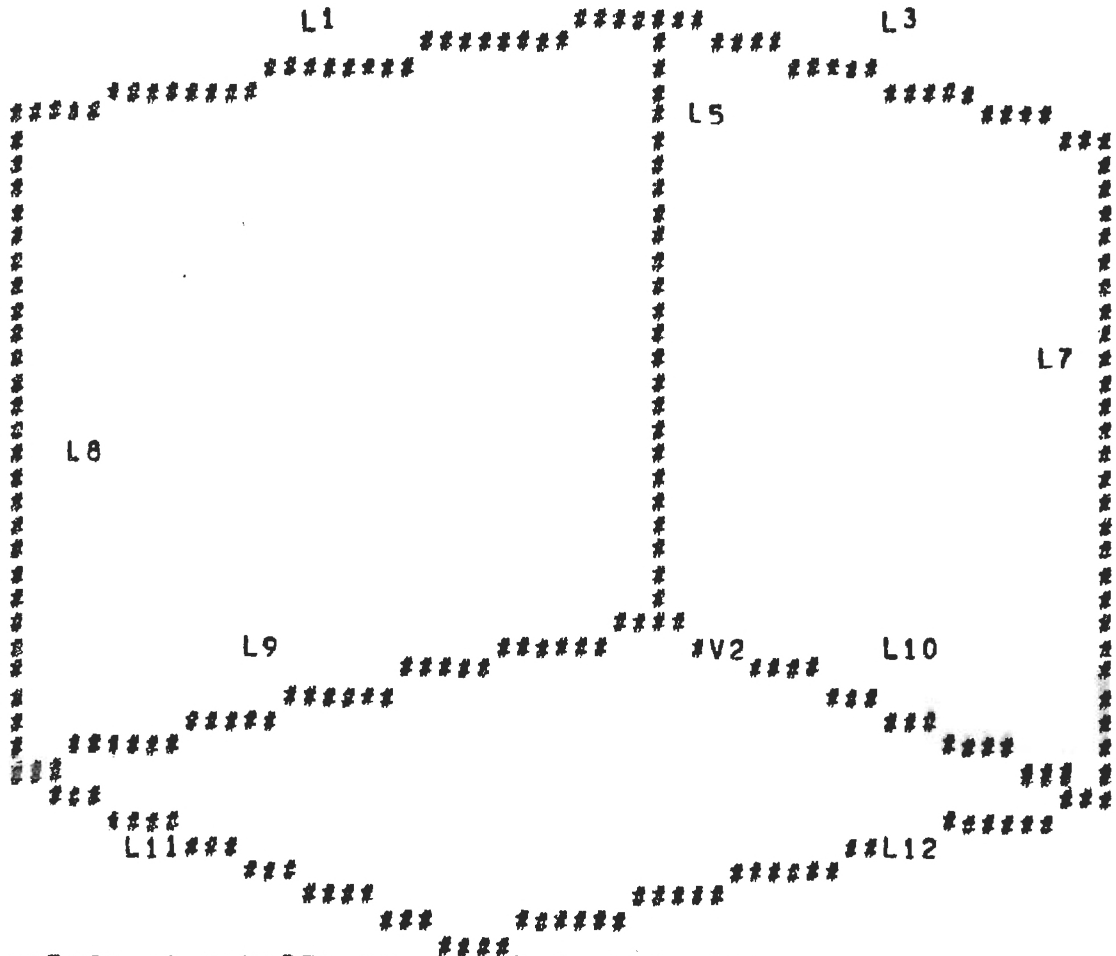
RIGHTSIDE IS BOUNDED BY L4 L7 L12 L6

LEFTSIDE IS BOUNDED BY L2 L8 L11 L6

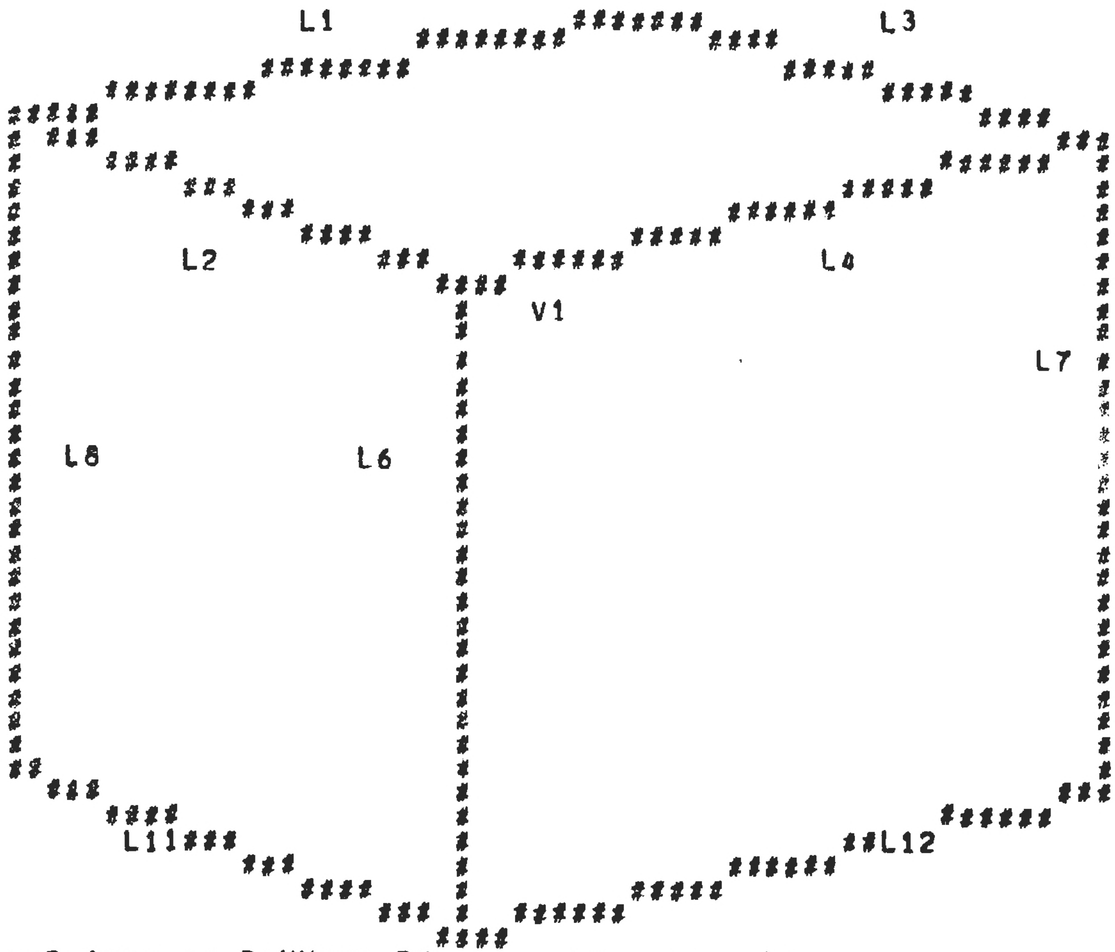
THE PROGRAM HAS DELETED THE PREVIOUS DESCRIPTION (B)

Figure 57. (continued)

Figure 57. (continued)




```
BOTTOM IS BOUNDED BY  L9  L10  L12  L11
RIGHTSIDE IS BOUNDED BY  L3   L5  L10   L7
LEFTSIDE IS BOUNDED BY  L1   L8   L9   L5
THE PROGRAM HAS DELETED THE PREVIOUS DESCRIPTION (A)
```



TOP SIDE IS BOUNDED BY L1 L2 L4 L3

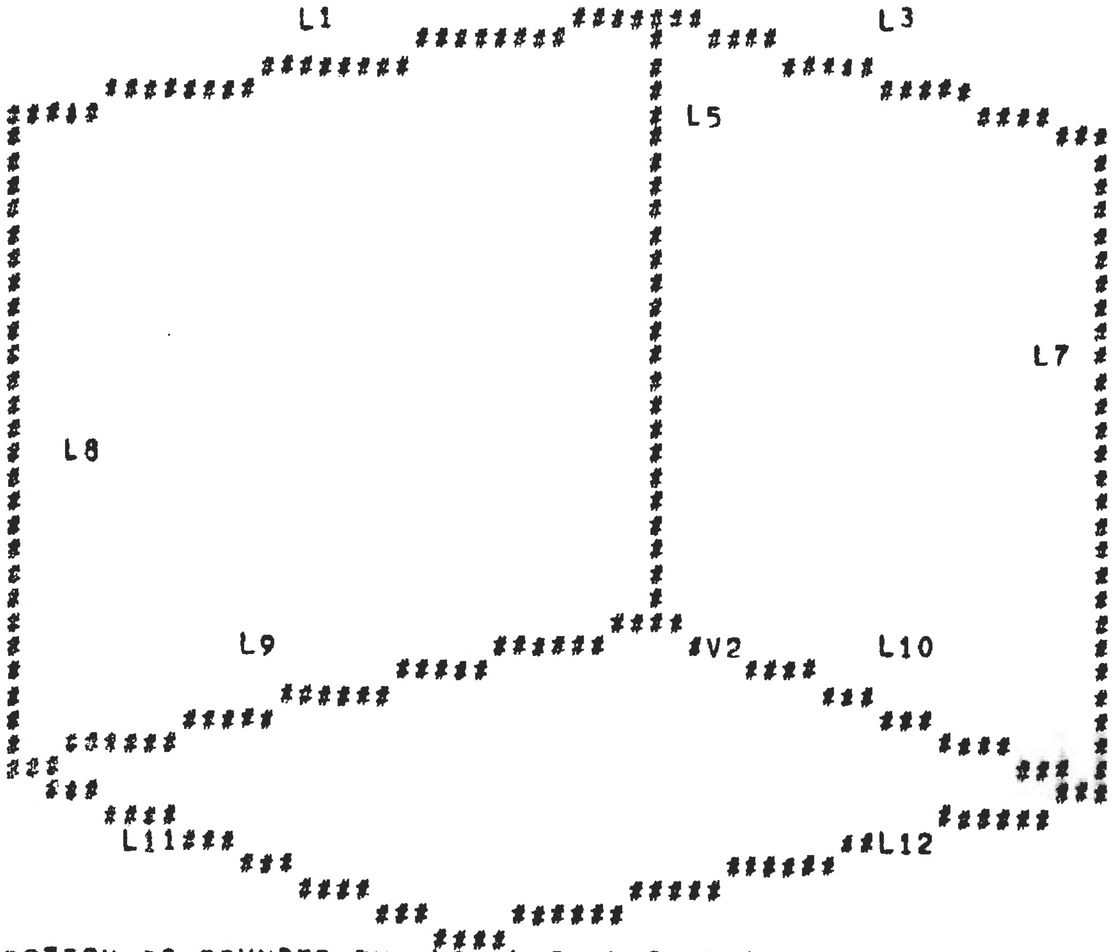
RIGHTSIDE IS BOUNDED BY L4 L7 L12 L6

LEFTSIDE IS BOUNDED BY L2 L8 L11 L6

THE PROGRAM HAS DELETED THE PREVIOUS DESCRIPTION (B)

Figure 57. (continued)

Figure 57. (continued)



BOTTOM IS BOUNDED BY L9 L10 L12 L11

RIGHTSIDE IS BOUNDED BY L3 L5 L10 L7

LEFTSIDE IS BOUNDED BY L1 L8 L9 L5

THE PROGRAM IS IN A STATE OF MULTISTABILITY (LOOP)
AND HAS TERMINATED ITSELF SINCE NO NEW PROPERTIES OR RELATIONS ARE
BEING INFERRED THAT HAVE NOT BEEN INFERRED BEFORE.

CHAPTER IX

CONCLUSIONS AND RECOMMENDATIONS

The receptor/categorizer paradigm and the syntactic paradigm for picture recognition have been shown to be inadequate for many recognition tasks. In the past, problem solutions that could not be formulated using one of these paradigms were necessarily ad hoc and heuristic in nature. The need for a new more powerful paradigm for picture recognition has been widely recognized. In this thesis, a new more powerful paradigm for picture recognition has been developed, and it is termed the "semantic paradigm," or the "paradigm for semantic picture recognition." The development of the paradigm came about through a detailed analysis of the underlying causes of the limitations of syntactic picture recognition techniques.

Evidence that the semantic paradigm is more powerful than existing paradigms is found in the solution of several picture recognition problems that heretofore were solvable only through ad hoc and heuristic programming.

As with most research results, more problems are discovered than solved. This research has brought out the need for dealing in a systematic way with problems of "consistency" and "completeness" of a set of rules of inference and the quality and relation domains. The largely unsolved problem of learning is especially evident in the use of the semantic paradigm. It appears that the semantic paradigm offers a concrete structure within which a learning strategy might be incorporated, although the magnitude of this problem must not be underestimated.

A thorough investigation into the use of automatic theorem proving techniques for testing rules of inference is warranted since "brute force" search often is too slow for on-line applications, and ad hoc search procedures detract from the appeal of the generality of the paradigm.

A more immediate suggestion for further research is to attempt to apply the semantic paradigm for the solution of recognition problems that are economically justifiable. Typical of such research would be the design of industrial "robots" for assembly and inspection of parts. The author plans on participating in this type of research, which will hopefully give more insight into the solution of the longer-range problems mentioned above.

APPENDIX

LISTING OF ALGOL PROGRAM FOR THE EVALUATION OF THE PARADIGM FOR A PROBLEM INVOLVING MULTISTABILITY IN PERCEPTION

BURROUGHS B-5700 ALGOL COMPILER MARK XIV.000   TUESDAY, 12/12/72,

BIN1001/CUBE

```
*%COMPILE BIN1001/CUBE    *ALGOL           .32C12049    *1001 MIKE BAIRD
*%PROCESS=   5;IO=   5.
*%PROGRAM.
*%
*%
*BEGIN
*COMMENT -------------------------------------------------------------;
*COMMENT -------------------------------------------------------------;
*FILE OUT LP 16(2,15);
*COMMENT     NECKER CUBE PROGRAM
*            MICHAEL L. BAIRD
*            GA. TECH. BOX 30776
*            OFFICE EE E-475
*            PHONE  894-3150  ;
*
*
*COMMENT -----------DECLARATIONS--------------------------------------;
*COMMENT -----------DECLARATIONS--------------------------------------;
*COMMENT -----------DECLARATIONS--------------------------------------;
*
*
*COMMENT -----------PROPERTIES----------------------------------------;
*COMMENT -----------PROPERTIES----------------------------------------;
*
*
*COMMENT THE FOLLOWING ARRAY STORES PROPERTY INFORMATION.
*        NOTE THAT UP TO THREE PROPERTIES CAN BE USED TO DESCRIBE
*        AN OBJECT.  THERE ARE 9 OBJECTS (REGIONS) LABELED 1-9;
*ALPHA ARRAY OBJECT[1:9,1:3];
*
*
*COMMENT -----------RELATIONS-----------------------------------------;
*COMMENT -----------RELATIONS-----------------------------------------;
*
```

147

```
*
*COMMENT THE FOLLOWING ARRAYS (GRAPHS) STORE RELATION INFORMATION
*        BETWEEN THE OBJECTS.   E.G., A  1  IN ROW  3, COLUMN 4
*        MEANS THAT THE RELATION HOLDS BETWEEN OBJECTS  3  AND  4,
*;
*INTEGER ARRAY RIGHTOF,ABOVE,BELOW,VERTEXOF [1:9,1:9];
*
*
*COMMENT -----------OTHER DECLARATIONS----------------------;
*COMMENT -----------OTHER DECLARATIONS----------------------;
*
*
*INTEGER V,W,Y,Z;
*INTEGER I,J,K,L,M,X,POINTER;
*LABEL INTERPRETER,EXIT,RULE0,RULE1,RULE2,RULE3;
*SWITCH RULE←RULE0,RULE1,RULE2,RULE3;
*
*COMMENT -----------INFORMATIVE PRINTOUT INFO HERE-----------;
*
*FORMAT EXPLANATIONOFPROGRAM(
*"IN THIS PROBLEM, THE SEMANTIC PARADIGM IS APPLIED TO A PROBLEM",/,
*"INVOLVING MULTISTABILITY IN PERCEPTION.  SPECIFICALLY, THE ",/,
*"PROBLEM INVOLVES THE REVERSAL EFFECT ENCOUNTERED BY HUMAN ",/,
*"INFORMATION PROCESSORS IN ATTEMPTING TO RECOGNIZE A TRANSPARENT",/,
*"CUBE (NECKER CUBE).  IT IS SHOWN THAT THE SEMANTIC PARADIGM",/,
*"EXHIBITS AN ANALOGOUS KIND OF BEHAVIOR IN WHICH FIRST ONE",/,
*"DESCRIPTION OF THE SCENE IS CONSTRUCTED WHICH CONTAINS A",/,
*"DESCRIPTION OF THE CUBE VIEWED FROM ONE PRINCIPAL ASPECT (A)",/,
*"AND THEN CONTINUED APPLICATION  OF THE RULES OF INFERENCE RESULTS",/,
*"IN THE DELETION OF THE PREVIOUS DESCRIPTION OF THE CUBE AND ",/,
*"REPLACES IT WITH ANOTHER  VIEWED FROM ANOTHER PRINCIPAL ASPECT",/,
*"(B).  THIS REVERSAL CYCLE CONTINUES INDEFINITELY, AS IT DOES IN A",/,
*"HUMAN INFORMATION PROCESSOR.",/,
*"      THE INPUTED PRIMITIVE SCENE DESCRIPTION IS IN TERMS OF",/,
*"THE FOLLOWING 9 OBJECTS:",/,
*"OBJECT1 (F1),...,OBJECT6 (F6) ARE FACES OF OBJECT9 (NECKERCUBE)",/,
*"OBJECT 7 (V1) AND OBJECT8 (V2) ARE VERTICES OF TYPE Y (YY) AND ",/,
*"INVERTED Y (NEGYY) RESPECTIVELY.",/,
*"THE SINGLE QUALITY IN Q BAR PRIME IS Q1 = SHAPE=",/,
*"(NECKER CUBE,FACE OF A CUBE, TYPE Y VERTEX,TYPE INVERTED Y VERTEX",/,
*"VERTEX).   WHERE Q BAR = (Q1, Q2) AND",/,
*"Q2 = CUBE TYPE = (CUBEA, CUBEB).  I.E., THE TWO VIEWS OF THE CUBE",/,
*"TO BE RECOGNIZED",/,
```

```
*"      THE RELATIONS INVOLVED ARE R = R PRIME = (R1,R3,R4,R5)",/,
*"WHERE",/,
*"R1 = --IS TO THE RIGHT OF, DISJOINT, AND BOUNDARY-CONNECTED TO--",/,
*"R3 = --IS ABOVE           , DISJOINT, AND BOUNDARY-CONNECTED TO--",/,
*"R4 = --IS BELOW           , DISJOINT, AND BOUNDARY-CONNECTED-TO--",/,
*"R5 = --IS A VERTEX ON THE BOUNDARY OF FACE--",////////);
*
*
*FORMAT GIVEITUP("THE PROGRAM IS IN A STATE OF MULTISTABILITY (LOOP)",/,
*"AND HAS TERMINATED ITSELF SINCE NO NEW PROPERTIES OR RELATIONS ARE",/,
*"BEING INFERRED THAT HAVE NOT BEEN INFERRED BEFORE.",///);
*FORMAT PROCES("PROCESS TIME USED = ", I5, " SECONDS...",///);
*FORMAT INPOUTPU("INPUT/OUTPUT TIME USED = ", I5, " SECONDS....",///
*);
*FORMAT DELETEA("THE PROGRAM HAS DELETED THE PREVIOUS DESCRIPTION (A)",//
*);
*FORMAT DELETEB("THE PROGRAM HAS DELETED THE PREVIOUS DESCRIPTION (B)",//
*);
*
*
*COMMENT -----------DEFINES-----------------------------------;
*COMMENT -----------DEFINES-----------------------------------;
*
*
*COMMENT -----------DEFINES USED TO SHORTEN THE RULES OF INFERENCE;
*DEFINE FORV = FOR V ← 1 STEP 1 UNTIL 9 DO #
*;
*DEFINE FORW = FOR W ← 1 STEP 1 UNTIL 9 DO #
*;
*DEFINE FORX = FOR X ← 1 STEP 1 UNTIL 9 DO #
*;
*DEFINE FORY = FOR Y ← 1 STEP 1 UNTIL 9 DO #
*;
*DEFINE FORZ = FOR Z ← 1 STEP 1 UNTIL 9 DO #
*;
*
*
*COMMENT -----------PROCEDURES--------------------------------;
*COMMENT -----------PROCEDURES--------------------------------;
*COMMENT -----------PROCEDURES--------------------------------;
*COMMENT -----------PROCEDURES--------------------------------;
*
*
*BOOLEAN PROCEDURE PROPERTY(VAR,PROP);
*ALPHA PROP; INTEGER VAR;
*BEGIN
```

```
*LABEL OU ;
*FOR I← 1 STEP 1 UNTIL 3 DO IF OBJECT[VAR,I]=PROP THEN
*       BEGIN PROPERTY ← TRUE; GO TO OU ;
*       END;
*COMMENT IF PROP IS PRIMITIVE AND IS NOT IN THE LIST, VALUE IS FALSE;
*IF (PROP="NECKER" OR PROP="YY" OR PROP="NEGYY") THEN
*       BEGIN
*       PROPERTY ← FALSE; GO TO OU ;
*       END;
*COMMENT SINCE WE CAN NOT SAY FOR SURE THAT THE PROPERTY ...
*MIGHT HOLD, WE CONTINUE THE SEARCH BY SETTING PROPERTY=FALSE...
*SINCE THIS HAS THE EFFECT OF REJECTING THE RULE UNLESS IT IS
*SHOWN TO HOLD......(NOTE:THIS HOLDS ONLY FOR RULES NOT USING
*NEGATIONS OF PROPERTIES IN THE INQUIRY FIELD...AND THAT IS THE
*CASE IN THIS PROGRAM).....;
*PROPERTY←FALSE;
*OU:
*END;
*
*COMMENT ----------------------------------------------------;
*
*
*BOOLEAN PROCEDURE RELATION(R,VAR1,VAR2);
*ALPHA ARRAY R[*,*]; INTEGER VAR1, VAR2;
*BEGIN
*LABEL OU;
*IF R[VAR1,VAR2] = 1 THEN
*       BEGIN
*       RELATION ← TRUE; GO TO OU ;
*       END;
*COMMENT SINCE ALL RELATIONS ARE PRIMITIVE, IF NO TRUTH VALUE IS

*OBTAINED, WE RETURE A VALUE OF FALSE;
*RELATION ← FALSE;
*OU:
*END;
*
*COMMENT ----------------------------------------------------;
*
*
```

```
*PROCEDURE ADD(PROP,VAR);
*ALPHA PROP; INTEGER VAR;
*BEGIN
*LABEL OU;
*FORMAT ERROR("NO EMPTY SPACES IN PROPERTY LIST...ERROR...",///);
*FOR I ← 1 STEP 1 UNTIL  3 DO
*IF OBJECT[VAR,I] = " " THEN
*       BEGIN
*       OBJECT[VAR,I] ← PROP;
*       GO TO OU;
*       END;
*WRITE(LP,ERROR);
*OU;
*END;
*
*COMMENT -----------------------------------------------------;
*
*
*PROCEDURE DELETE(PROP,VAR);
*ALPHA PROP; INTEGER VAR;
*BEGIN
*FOR I←1 STEP 1 UNTIL 3 DO IF OBJECT[VAR,I]=PROP THEN OBJECT[VAR,I]←" ";
*END;
*
*COMMENT -----------------------------------------------------;
*
*
*PROCEDURE CUBEPRINTOUT(Q);
*ALPHA Q;
*BEGIN
*COMMENT FORMATS USED TO PRINT OUT FACE INFO.;
*FORMAT TOP("TOP SIDE IS BOUNDED BY ", 4(A3,X2),/);
*FORMAT RIGHTSIDE("RIGHTSIDE IS BOUNDED BY ", 4(A3,X2),/);
*FORMAT LEFTSIDE("LEFTSIDE IS BOUNDED BY ", 4(A3,X2),/);
*FORMAT BOTTOM("BOTTOM IS BOUNDED BY ", 4(A3,X2),/);
*FORMAT RASTER(39(57A1));
*ALPHA ARRAY  A[0:56, 0:38];
*FOR I←0 STEP 1 UNTIL 56 DO FOR J←0 STEP 1 UNTIL 38 DO A[I,J]←" ";
*COMMENT NOW THAT THE ARRAY IN WHICH THE PICTURE IS REPRESENTED
*HAS BEEN INITIALIZED (BLANKED) WE INSERT DATA FOR LINES
*L1,L3,L7,L8,L11,L12 (IN THAT ORDER) WHICH ARE COMMON TO ALL
*VIEWS OF THE CUBE;
*FOR X←0 STEP 1 UNTIL 33 DO   A[X,ENTIER(4/33×X + 34.5)]←"#";
*FOR X←33 STEP 1 UNTIL 56 DO A[X,ENTIER(-5/23×X + 45.673)]←"#";
*FOR Y← 6 STEP 1 UNTIL 33 DO A[56,Y]←"#";
*FOR Y←7 STEP 1 UNTIL 34 DO A[0,Y]←"#";
*FOR X←0 STEP 1 UNTIL 23 DO A[X,ENTIER(-7/23×X + 7.5)]←"#";
*FOR X←23 STEP 1 UNTIL 56 DO A[X,ENTIER(6/33×X -4.1818 + 0.5)]←"#";
```

```
*COMMENT NOW WE LABEL THE LINES (SAME ORDER);
*A[15,38]←"L"; A[16,38]←"1";
*A[45,38]←"L"; A[46,38]←"3";
*A[53,24]←"L";  A[54,24]←"7";
*A[3,20]←"L";  A[4,20]←"8";
*A[6,4]←"L";  A[7,4]←"1";  A[8,4]←"1";
*A[45,4]←"L";  A[46,4]←"1";  A[47,4]←"2";
*COMMENT ONLY FOR VIEW A OR THE NECKER CUBE WE DEFINE THE LINES L2,L4,
*L6;
*IF (Q="A" OR Q="NECKER") THEN
*BEGIN
*FOR X← 0 STEP 1 UNTIL 23 DO A[X,ENTIER(-7/23×X + 34.5)]←"#";
*FOR X ← 23 STEP 1 UNTIL 56 DO A[X,ENTIER(6/33×X  + 22.8182  + 0.5)]←
*"#";
*FOR Y← 0 STEP 1 UNTIL 27 DO A[23,Y]←"#";
*A[9,28]←"L";  A[10,28]←"2";
*A[42,28]←"L";  A[43,28]←"4";
*A[18,20]←"L";  A[19,20]←"6";
*A[27,26]←"V";  A[28,26]←"1";
*END;
*COMMENT ONLY FOR VIEW B OR THE NECKER CUBE DO WE NEED THE FOLLOWING
*LINES : L5,L9,L10;
*IF (Q="B" OR Q="NECKER") THEN
*BEGIN
*FOR Y←13 STEP 1 UNTIL 38 DO
*A[33,Y]←"#";
*FOR X← 0 STEP 1 UNTIL 33 DO

*A[X,ENTIER(6/33× X  + 7.5)]←"#";
*FOR X← 33 STEP 1 UNTIL 56 DO
*A[X,ENTIER(-7/23×X  + 23.54)]←"#";
*A[35,34]←"L"; A[36,34]←"5";
*A[12,12]←"L";  A[13,12]←"9";
*A[45,12]←"L";  A[46,12]←"1";  A[47,12]←"0";
*A[36,12]←"V"; A[37,12]←"2";
*END;
*WRITE(LP[PAGE]);
*FOR I←38 STEP -1 UNTIL 0 DO
*WRITE(LP,RASTER, FOR J←0 STEP 1 UNTIL 56 DO A[J,I]);
*COMMENT PRINT OUT INFORMATION REGARDING EXPOSED FACES (DEPENDS ON
*THE VIEW OF COURSE);
*IF Q="A" THEN
*      BEGIN
*WRITE(LP,TOP, " L1", " L2", " L4", " L3");
*WRITE(LP,RIGHTSIDE," L4"," L7","L12"," L6");
*WRITE(LP,LEFTSIDE," L2", " L8", "L11", " L6");
*      END;
```

```
*IF Q="B" THEN
*      BEGIN
*WRITE(LP,BOTTOM," L9", "L10", "L12", "L11");
*WRITE(LP,RIGHTSIDE," L3", " L5", "L10", " L7");
*WRITE(LP,LEFTSIDE," L1", " L8", " L9", " L5");
*      END;
*END;
*
*COMMENT -----------------------------------------------------;
*
*
*PROCEDURE GETDATA;
*BEGIN
*COMMENT  IN THE FOLLOWING, PROPERTIES AND RELATIONS ARE
*ASSIGNED WHICH CORRESPOND TO THE INPUTED PRIMITIVE SCENE
*DESCRIPTION.     ;
*FOR I+1 STEP 1 UNTIL 9 DO FOR J+1 STEP 1 UNTIL 3 DO OBJECT[I,J]+" ";
*OBJECT[9,1]+"NECKER"; OBJECT[7,1]+"YY"; OBJECT[8,1]+"NEGYY";
*RIGHTOF[1,2]+RIGHTOF[3,2]+RIGHTOF[4,5]+RIGHTOF[6,5]+1;
*ABOVE[1,2]+ABOVE[1,3]+ABOVE[5,4]+ABOVE[6,4]+1;
*BELOW[2,1]+BELOW[3,1]+BELOW[4,5]+BELOW[4,6]+1;
*VERTEXOF[7,1]+VERTEXOF[7,2]+VERTEXOF[7,3]+VERTEXOF[8,4]+
*VERTEXOF[8,5]+VERTEXOF[8,6]+1;
*END;
*
*
*COMMENT -----------------------------------------------------;
*
*
*COMMENT -----------BEGIN PROGRAM---END OF PROCEDURES--------;
*COMMENT -----------BEGIN PROGRAM---END OF PROCEDURES--------;
*COMMENT -----------BEGIN PROGRAM---END OF PROCEDURES--------;
*COMMENT -----------BEGIN PROGRAM---END OF PROCEDURES--------;
*COMMENT -----------BEGIN PROGRAM---END OF PROCEDURES--------;
*
*
*GETDATA; WRITE(LP,EXPLANATIONOFPROGRAM); POINTER+-1;
*CUBEPRINTOUT("NECKER");
*
*
*COMMENT -----------INTERPRETER-------------------------------;
*COMMENT -----------INTERPRETER-------------------------------;
*COMMENT -----------INTERPRETER-------------------------------;
*
```

```
*
*INTERPRFTER:
*COMMENT THIS SECTION APPLIES THE 4 RULES OF INFERENCE
*(RULE0, RULF1, RULF2, RULE3) IN SFQUFNTIAL ORDER.
*THF VARIABLF NAMFD "POINTER" IS AN INDEX INDICATING THE
*LAST RULF REFERFNCFD;
*IF TIME(2)/60 ≥
*120
*THFN GO TO FXIT;
*POINTER←(POINTER + 1)MOD 4;
*GO TO RULE[POINTFR+1];
*
*
*COMMENT -----------RULES OF INFERFNCE---------------------;
*COMMENT -----------RULES OF INFERFNCE---------------------;
*COMMENT -----------RULES OF INFERFNCE---------------------;
*COMMENT -----------RULES OF INFERFNCE---------------------;
*
*
*COMMENT  QUANTIFICATION IS RESOLVFD IN THIS CASE THROUGH DFPTH FIRST
*SEARCH....SFE DEFINE STATEMFNTS FOR FXPLAINATION OF FORZ, FORY, ETC.;
*;
*
*
*RULF0:
*
*FORZ IF PROPERTY(Z,"CUBEB") THEN GO TO INTERPRETER;
*FORZ FORY FORX FORW FORV
*IF PROPFRTY(V,"NFCKFR") THEN
*IF PROPFRTY(W,"NFGYY") THEN
*IF RELATION(VERTFXOF, W, X ) THEN
*IF RELATION(VERTFXOF, W, Y ) THEN
*IF RELATION(VERTFXOF, W, Z ) THEN
*IF RELATION(BELOW, X, Y ) THEN
*IF RELATION(BELOW, X, Z ) THEN
*IF RELATION(RIGHTOF, Y, Z ) THEN
*BFGIN
*ADD("CUBFB",V);
*ADD("FACFB",X);
*ADD("FACFB",Y);
*ADD("FACFB",Z);
*CUBEPRINTOUT("B"); GO TO INTERPRETER;
*END;
*GO TO INTERPRETER;
*
```

```
*
*RULE1;
*
*COMMENT THIS RULE FOLLOWS SINCE ALL PROPERTIES IN A QUALITY SET
*ARE MUTUALLY EXCLUSIVE;
*FORZ FORY FORX FORW
*IF PROPERTY(W,"CUBEA") THEN
*IF PROPERTY(W,"CUBEB") THEN
*IF PROPERTY(X,"FACEA") THEN
*IF PROPERTY(Y,"FACEA") THEN
*IF PROPERTY(Z,"FACEA") THEN
*BEGIN
*DELETE("CUBEA",W);
*DELETE("FACEA",X);
*DELETE("FACEA",Y);
*DELETE("FACEA",Z);
*WRITE(LP,DELETEA);
*GO TO INTERPRETER;
*END;
*GO TO INTERPRETER;
*
*
*RULE2;
*
*FORZ IF PROPERTY(Z,"CUBEA") THEN GO TO INTERPRETER;
*FORZ FORY FORX FORW FORV
*IF PROPERTY(V,"NECKER") THEN
*IF PROPERTY(W,"YY") THEN
*IF RELATION(VERTEXOF,W,X) THEN
*IF RELATION(VERTEXOF,W,Y) THEN
*IF RELATION(VERTEXOF,W,Z) THEN
*IF RELATION(ABOVE, X,Y) THEN
*IF RELATION(ABOVE, X,Z) THEN
*IF RELATION(RIGHTOF, Y, Z) THEN
*BEGIN
*ADD("CUBEA",V);
*ADD("FACEA",X);
*ADD("FACEA",Y);
*ADD("FACEA",Z);
*CUBEPRINTOUT("A");  GO TO INTERPRETER;
*END;
*GO TO INTERPRETER;
*
```

```
*
*RULE3:
*
*COMMENT THIS RULE FOLLOWS SINCE ALL PROPERTIES IN A QUALITY SET
*ARE MUTUALLY EXCLUSIVE;
*FORZ FORY FORX FORW
*IF PROPERTY(W,"CUBEA") THEN
*IF PROPERTY(W,"CUBEB") THEN
*IF PROPERTY(X,"FACEB") THEN
*IF PROPERTY(Y,"FACEB") THEN
*IF PROPERTY(Z,"FACEB") THEN
*BEGIN
*DELETE("CUBEB",W);
*DELETE("FACEB",X);
*DELETE("FACEB",Y);
*DELETE("FACEB",Z);
*WRITE(LP,DELETER);
*GO TO INTERPRETER;
*END;
*GO TO INTERPRETER;
*
*
*
*EXIT:
*WRITE(LP,GIVEITUP);
*WRITE(LP,PROCES,TIME(2)/60); WRITE(LP,INPUOUTPU,TIME(3)/60);
*END.
```

BIBLIOGRAPHY

1. R. H. Anderson, Syntax-directed recognition of hand-printed two-dimensional mathematics. Ph.D. Dissertation, Applied Mathematics, Harvard University, Cambridge, Mass. Also in Proc. of the ACM Symposium on Interactive Systems for Experimental Applied Mathematics. (1967).

2. F. Attneave, Multistability in perception, Scientific American, 62-71 (Dec., 1971).

3. H. G. Barrow and R. J. Popplestone, Relational descriptions in picture processing, Machine Intelligence 6, 377-396 (1971).

4. L. Carlucci, A formal system for texture languages, Pat. Recog. 4,1, 53-72 (1972).

5. B. Chandrasekaran and L. Kanal, On Linguistic, Statistical, and Mixed Models for Pattern Recognition, Computer and Information Science Research Center, Technical Report Series (OSU-CISRC-TR-71-3) (1971).

6. M. Clowes, Transformational grammars and the organization of pictures, Automatic Interpretation and Classification of Images, A. Grasselli (Ed.), Academic Press, N.Y., 43-78 (1969).

7. M. Clowes, On seeing things, Artificial Intelligence 2, 1 (Spring, 1971).

8. T. Evans, A program for the solution of a class of geometric-analogy intelligence test questions. Ph.D. Dissertation, Department of Mathematics, M.I.T. (May, 1963). Also in Semantic Information Processing, M. Minsky (Ed.), M.I.T. Press, Cambridge, Mass., 271-353 (1968).

9. T. Evans, Descriptive pattern analysis techniques, Automatic Interpretation and Classification of Images, A. Grasselli (Ed.), Academic Press, N.Y., 79-95 (1969).

10. O. Firschein and M. A. Fischler, Describing and abstracting pictorial structures, Pat. Recog. 3,4, 421-443 (1971).

11. A. Guzmán, Decomposition of a visual scene into three-dimensional bodies, AFIPS, FJCC, 291-304 (1968).

BIBLIOGRAPHY (Continued)

12. A. Guzmán, Analysis of curved line drawings using context and global information, Machine Intelligence 6, 325-375 (1971).

13. M. D. Kelly, Visual identification of people by computer. Ph.D. Dissertation, Computer Science Department, Stanford University (July, 1970). Also Stanford A.I. Project memo AI-130.

14. R. A. Kirsch, Computer interpretation of English text and picture patterns, IEEE Trans. on Electronic Computers EC-13, 4, 363-376 (August, 1964).

15. R. A. Kirsch, Picture syntax, Pattern Recognition, L. Kanal (Ed.), Thompson Book Company, Washington, D.C., 183-184 (1968).

16. R. S. Ledley, Programming and Utilizing Digital Computers, Chapter 8. McGraw-Hill (1962).

17. L. E. Lipkin, W. C. Watt, and R. A. Kirsch, The analysis, synthesis, and description of biological images, Annals New York Academy of Sciences 128, 984-1012 (Jan. 31, 1969).

18. T. Marill and D. M. Green, Statistical recognition functions and the design of pattern recognizers, IRE Transactions of Electronic Computers EC-9,4, 472-477 (Dec., 1960).

19. W. F. Miller and A. C. Shaw, Linguistic methods in picture processing: a survey, AFIPS, FJCC, Part I, 279-290 (1968).

20. M. Minsky, Discussion of Kirsch's talk, Pattern Recognition, L. Kanal (Ed.), Thompson Book Company, Washington, D.C., 185 (1968).

21. R. Narasimhan, A Linguistic Approach to Pattern Recognition, Digital Computer Laboratory Report No. 121, University of Illinois, Urbana, Illinois (1962).

22. R. Narasimhan, On the description, generation, and recognition of classes of pictures, Automatic Interpretation and Classification of Images, A. Grasselli (Ed.), Academic Press, N.Y., 1-42 (1969).

23. R. Narasimhan and V. S. N. Reddy, A syntax-aided recognition scheme for handprinted English letters, Pat. Recog. 3,4, 345-362 (1971).

24. T. Pavlidis, Representation of figures by labeled graphs, Pat. Recog. 4,1, 5-18 (1972).

BIBLIOGRAPHY (Concluded)

25. M. R. Quillian, Semantic memory, Semantic Information Processing, M. Minsky (Ed.), M.I.T. Press, Cambridge, Mass., 227-270 (1968).

26. M. R. Quillian, Teachable language comprehender: a simulation program and theory of language, CACM 12, 8, 459-476 (1969).

27. A. C. Shaw, The formal description and parsing of pictures. Ph.D. Dissertation, Computer Science Department, Stanford University (1968).

28. H. A. Simon, The architecture of complexity, Proc. American Philos. Soc. 106, 467-482 (1962).

29. J. C. Simon, A. Checroun, and C. Roche, A method of comparing two patterns independent of possible transformations and small distortions, Pat. Recog. 4,1, 73-82 (1972).

30. P. Stucki, Generation of grey tones by computer for simulation of visual information systems, IEEE Trans. on Electronic Computers, 642 (July, 1969).

31. P. H. Swain and K. S. Fu, Stochastic programmed grammars for syntactic pattern recognition, Pat. Recog. 4,1, 83-100 (1972).

32. L. Uhr, Flexible pattern recognition, Pat. Recog. 3,4, 363-384 (1971).

33. S. Watanabe, Ungrammatical grammar in pattern recognition, Pat. Recog. 3,4, 385-408 (1971).

VITA

Michael Lloyd Baird was born in Washington, D.C. on December 17, 1945. He graduated from Sparrows Point High School in Baltimore, Maryland in 1964 and attended the General Motors Institute of Technology from 1964 to 1969, earning the Bachelor of Industrial Engineering degree. Mr. Baird's undergraduate education was based on a co-operative work-study plan. He was admitted to the "academic fifth year plan," during which time he produced a thesis entitled "Attenuation of Noise." Mr. Baird served as a Peace Corps volunteer in Lima, Peru before entering the Georgia Institute of Technology in 1970. He received the Master's degree in Information Science in 1971, and the Doctor of Philosophy degree in Information and Computer Science in 1973.

While at Georgia Tech, Mr. Baird was employed as a research assistant at the Engineering Experiment Station, and as a Graduate Research and Teaching Assistant in the School of Information and Computer Science.

Mr. Baird has co-authored papers and reports in picture processing by computer. He is a member of the IEEE, the Pattern Recognition Society, and has served as chairman of the Georgia Tech Student Chapter of the Association for Computing Machinery.

Mr. Baird expects to continue his work on picture recognition as a member of the machine perception group in the Computer Science Department at the General Motors Research Laboratories in Warren, Michigan.