This is a reproduction of a library book that was digitized by Google as part of an ongoing effort to preserve the information in books and make it universally accessible.

Google™ books

https://books.google.com

Stanford University Libraries
3 6105 025 637 757

Digitized by Google

STANFORD UNIVERSITY LIBRARIES

Digitized by Google

Digitized by Google

Digitized by Google

Digitized by Google

Digitized by Google

VISUAL IDENTIFICATION OF PEOPLE BY COMPUTER

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON THE GRADUATE DIVISION
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By

Michael David Kelly

August 1970 +R

Digitized by Google

Digitized by Google

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Reddy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

John McCarthy

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

W. F. Miller

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

[illegible]

(Stanford Research Institute)

Approved for the University Committee
on the Graduate Division:

[illegible]

Dean of the Graduate Division

Digitized by Google

Digitized by Google

ACKNOWLEDGEMENTS

I would like to express my appreciation to my thesis advisor, Professor D. R. Reddy, for his guidance, advice, and encouragement. I am grateful to Dr. Peter E. Hart for his helpful suggestions and to Dr. Hart and Professors John McCarthy and William F. Miller for their constructive readings of this thesis.

I am also indebted to R. Gary Goodman and J.M. Tenenbaum for many enlightening discussions, to Gilbert Falk for the use of Figures 3-1 and 3-2, to Karl Pingle and Richard Neely for making available to me some of their programs, and to Joan Kelly for her outstanding typing and drawing and for her constant patient encouragement.

I dedicate this work to my wife Joan.

This research was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (SD-183).

Digitized by Google

Digitized by Google

TABLE OF CONTENTS

Digitized by Google

Digitized by Google

CONTENTS (continued)

Digitized by Google

Digitized by Google

LIST OF TABLES

Digitized by Google

Digitized by Google

LIST OF ILLUSTRATIONS

Digitized by Google

Digitized by Google

ILLUSTRATIONS (CONTINUED)

Digitized by Google

Digitized by Google

ILLUSTRATIONS (CONTINUED)

Digitized by Google

Digitized by Google

CHAPTER 1

INTRODUCTION

Most present day picture processing programs are unsuitable for the analysis of naturally occuring visual scenes. When presented with objects such as trees, cars, and people, these programs would become hopelessly confused. What is more, after looking at the structure of these programs, it becomes obvious that there are no simple generalizations of these techniques which will succeed. This thesis describes an attempt to develop a computer program which performs a complex picture processing task. The task is to choose, from a collection of pictures of people, those pictures that depict the same person.

In brief, the program works by finding the location of features such as eyes, nose, or shoulders in the pictures. Individuals are classified by measurements between such features. The interesting and difficult part of the work reported in this thesis is the detection of these features in digital pictures.

The methods of the program as well as the goals of the research are summarized in the next chapter. In general, the primary purpose of this research has been directed toward the development of new techniques for picture processing. The identification of people is only an interesting problem that can be used for this purpose. It

Digitized by Google

Digitized by Google

is an interesting problem because pictures of people are so complex, and there is great variation between pictures. This is in sharp contrast to the simple geometrical solids used so often in research in picture processing.

No one has previously dealt successfully with complex objects such as people in computer picture processing. The success of the program is due to and illustrates the heuristic use of context and structure. A new, widely useful, technique called planning has been applied to picture processing. Planning is a term which is drawn from artificial intelligence research in problem solving.

The bulk of the work of this thesis has been devoted to methods of finding low level features such as eyes or shoulders in digital pictures. Many of the methods developed by previous researchers have been applied to different parts of the problem. The large number of methods that are used can be indicated by summarizing the techniques used in recognizing each part of the picture. Each method is discussed more fully in the remainder of the thesis. The position of the body is located by _subtraction_ of the background. The top of the head and the feet are identified by _template matching_. The edges of the head, neck, shoulders, and hips are found by _edge detection operators_. The outline of the head is obtained using _planning_ to guide _goal directed search_ after using edge detectors. The eyes are found by _dynamic threshold setting_ followed by _smoothing_

Digitized by Google

and template matching. The nose is located by dynamic threshold setting. The mouth is found with a <u>line detection operator.</u> All of these methods are applied heuristically in a goal directed manner which is based on an implicit model of the structure of the human body.

A large part of the success of this research is due to the availability of powerful interactive computer facilities for experimentation with the picture processing algorithms. Most often in devising algorithms for identifying particular features, it was found that intuition was faulty. The experimental procedure used to develop the final methods went somewhat as follows. For a particular feature a method for identification would be postulated. This method would be programmed and tried. Usually the method would fail. Using the display terminals available on the time-sharing system it was possible to watch the progress of the algorithm while it was running. Parameters could be varied interactively. This led to insight as to why the method failed and would suggest new methods or modifications. This "postulate-try-fail" loop usually had to be repeated many times before a satisfactory method was found. Once the method was found, "fine-tuning" was made feasible by the interactive facilities.

Digitized by Google

Digitized by Google

HISTORY.

My first interest in this problem began in a class project in the winter of 1967. A small amount of effort was devoted to this problem until the spring of 1968 when serious work began. The past two years have been devoted to this thesis research and to a study of basic picture processing procedures.

At first one of the goals was to recognize people on-line as they were standing in front of the TV camera. Eventually it turned out that the characteristics of the TV input system available prevented this. The chief problem is the inability of the system to provide detail over a wide range of light intensity values. If the entire video signal is digitized, very little detail is present. In order to get facial details, the video signal must be clipped top and bottom and the resulting narrow window digitized. This leads, however, to the necessity of adjusting external lighting, TV sensitivity, and clip levels for each person. This is time consuming and so error-prone as to be impractical. It demands too much of the patience of a volunteer subject. The work of J. M. Tenenbaum [1970], on automatic accommodation of visual parameters, promises improvement in this area.

As a consequence of the limitations on the quality of TV input, the following scheme was adopted for obtaining acceptable pictures. A number of pictures of people would

Digitized by Google

be taken in a picture taking session. Later these pictures would be examined visually by printing them on a line-printer using an approximate grey-scale (about 5 minutes per picture) or displaying them on a storage tube (about 20 minutes per picture). Those pictures which contained sufficient detail would be retained for processing. Minor faults did not cause a picture to be excluded; this was not a selection to fit the peculiarities of the recognition program.

ORGANIZATION.

This thesis is structured to discuss first the problem, next the solutions, and then the results. The next chapter will outline the goals of this research and relevant past work. Chapters 3 to 5 discuss the solutions in broad terms and in compairison with alternative approaches. Chapter 3 surveys picture analysis and description. Chapter 4 discusses goal-directed picture processing and the use of models. Chapter 5 covers the new technique, planning. Chapters 6, 7, and 8 examine the recognition algorithms in detail. Chapter 9 outlines the use of the nearest neighbor classification algorithm. The final chapter summarizes results and offers suggestions for further work.

Digitized by Google

Digitized by Google

CHAPTER 2

THE PROBLEM - RECOGNITION OF PEOPLE

This chapter will describe the overall goals and methods of the research described in this thesis. The method used for recognition will be outlined briefly to provide an overview of the structure of the program. In addition, past work on the recognition of people by computer will be summmarized.

GOALS.

There is great potential for enhancing the usefulness of computers by the addition of visual input capabilities. The primary goal of the research reported here has been the development and improvement of techniques for computer picture processing.

The term "picture processing" is used in this thesis to mean the processing of a picture obtained from the outside world. It includes areas often called "pictorial pattern recognition" and "picture analysis and description". It excludes generation of pictures by computer, so-called "computer graphics". This corresponds to Rosenfeld's usage of "picture processing" [1969a]. (Miller and Shaw [1968] ascribe a wider meaning to the term.)

An effort has been made to apply heuristic methods, drawn from artificial intelligence research in problem

Digitized by Google

Digitized by Google

solving, to computer picture processing. "A heuristic is a rule of thumb, strategy, trick, simplification, or any other kind of device which drastically limits search for solutions in large problem spaces" (Feigenbaum and Feldman [1963], p. 6). The term artificial intelligence has been used with a wide variety of meanings by specialists in various fields. In this thesis artificial intelligence research is considered to be complex problem solving by computer (of problems which may require intelligence for their solution) using heuristic techniques. Examples of such research include the General Problem Solver of Newell, Shaw, and Simon [1959]; programs that play checkers and chess (Samuel [1967], Greenblatt et al. [1967]); and the Dendral program of Lederberg and Feigenbaum [1968] for inference in organic chemistry.

Effective computer picture processing will probably come about via the incorporation of heuristic methods. The problems to be solved are large, complex, and not well understood. Adaptation to picture processing of generally effective heuristics used by artificial intelligence workers appears to be the best way to attack these problems. Surprisingly, in the past there has been relatively little interaction between these two research areas.

Advancement of computer picture processing can come from work on a specific problem. Past experience is meager and generally useful concepts are few. The primitive state

Digitized by Google

Digitized by Google

of knowledge makes it appropriate to attempt a particular problem with hopes of generalizing the results. This is what has been done in the work reported here.

The problem which has been chosen is the following: develop a program which will identify people from pictures taken by a TV camera attached to a computer.

OUTLINE OF THE METHOD.

The general scheme of operation of the program can be summarized as follows. Two pictures of the individual to be recognized are read into the computer. One is a picture of the entire body, head to feet. The other is a close-up of the head. The program processes the pictures to locate feature points such as the irises of the eyes, nostrils, the top of the head. Once these points are found, measurements are derived from them such as height, distance between eyes, width of head, etc. These measurements are then used in a pattern classification algorithm to extract the identity of the person from a dictionary containing known individuals and their measurements.

The method outlined above appears quite straight-forward: locate features, obtain measurements, classify pattern. Obviously, obtaining measurements once feature points are located is a trivial operation. The final stage of the method, pattern classification, has been well studied. Given a good set of features, there are standard

Digitized by Google

Digitized by Google

classification algorithms which may be used. However, the first step, locating the features in a digital picture, is a process about which very little is known.

This then is the main effort of this thesis research: accurately locating desired specific points on pictures of people. Such low level picture processing, called variously feature extraction, pre-processing, or characterization of the picture, is generally recognized as the most difficult part and the principal problem in pattern recognition. (See, for example, Ho and Agrawala [1968], p. 2102.) It is worthwhile emphasizing, in view of the fact that so much previous work in pattern recognition has been concerned with mathematical techniques for classification, that in this work classification has received only minor attention.

Measurements from the face and body should provide a good means of identifying people. The reason for this expectation is that physical measurements were the basis of the Bertillon system for identification of people, which had wide use in police work prior to the discovery of the usefulness of fingerprints (Thorwald [1965]). This expectation is confirmed by the results obtained by Bledsoe which are described later in this chapter.

The measurements which have been selected for use in this thesis are given in Figures 2-1 and 2-2. Each measurement is normalized by dividing it by the measured

Digitized by Google

Digitized by Google

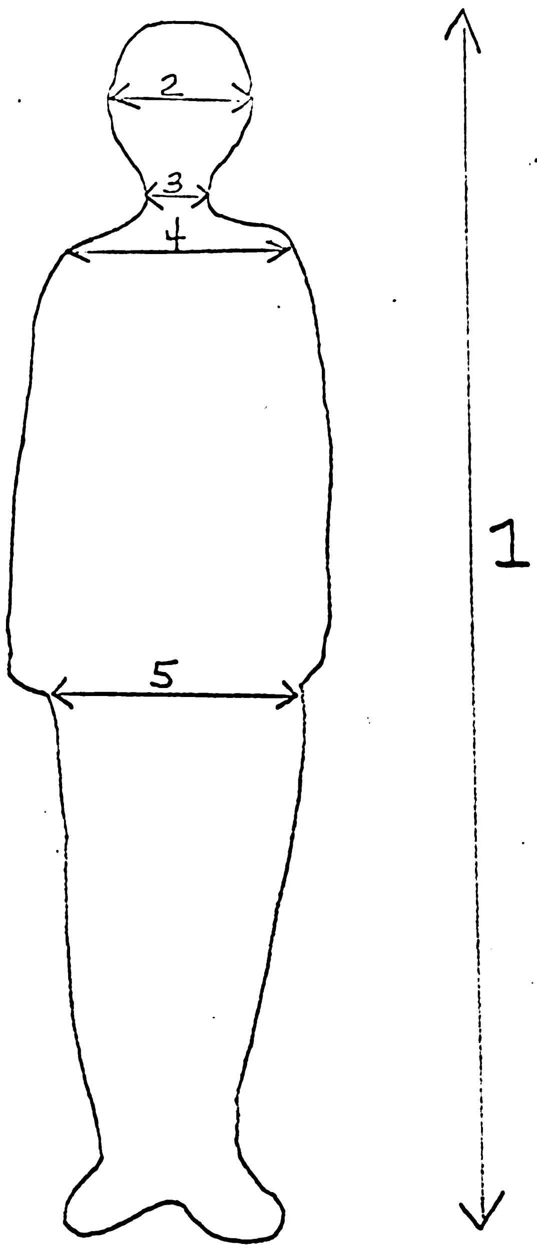


Figure 2-1. Measurements from the body.

1. Height.
2. Width of head.
3. Width of neck.
4. Width of shoulders.
5. Width of hips.

Digitized by Google

Digitized by Google

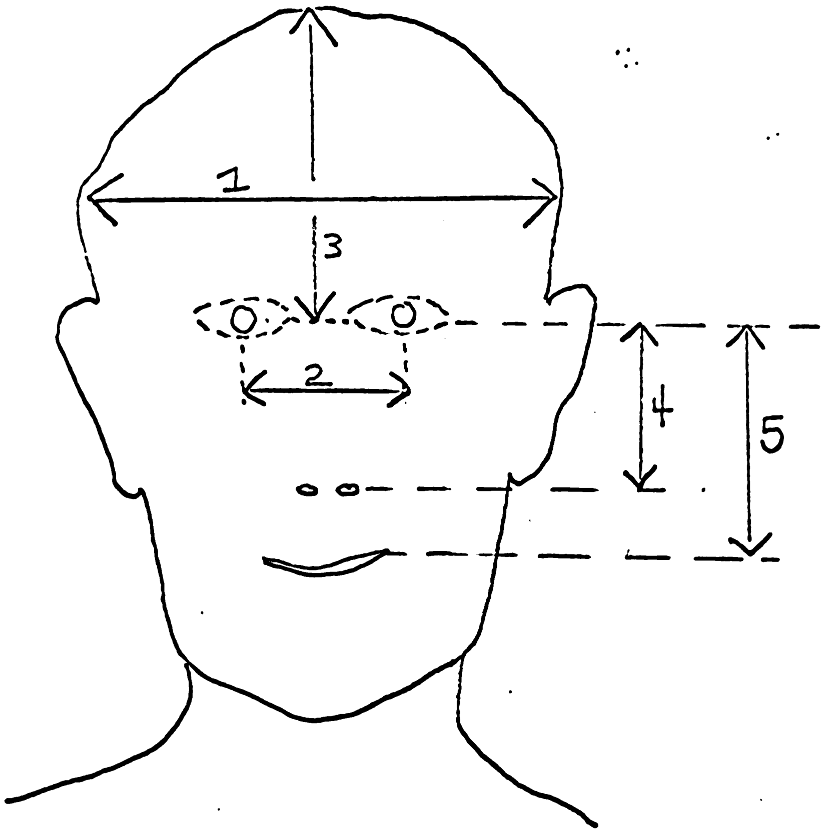


Figure 2-2. Measurements from the head.

1. Width of the head.
2. Distance between eyes.
3. Distance from top of head to eyes.
4. Distance from eyes to nose.
5. Distance from eyes to mouth.

Digitized by Google

Digitized by Google

height.

PREVIOUS WORK.

A number of papers have appeared in which pictures of faces have been partially processed by computer and the results displayed. Examples are the papers by Narasimhan and Forango [1964] and Hueckel [1969]. Both of these papers present computer produced line drawings of faces which were derived from grey scale input pictures. Such operations are fine for presenting faces to the human eye but represent only a very small step toward computer description and recognition.

The principal prior work on recognition of people by computer has been done by W.W. Bledsoe. This work was begun at Panoramic Research, Inc. and continued with P.E. Hart at Stanford Research Institute. (See Bledsoe [1964] and [1966]).

There are differences and similarities between the work of Bledsoe and the work reported here. The chief difference is that Bledsoe created a man-machine system in which a human operator, working with a face projected on a Grafacon or "Rand tablet", located the feature points on the face and manually pointed out their position for the computer to record. In contrast, my work consists primarily of an attempt to automate this feature location step. Bledsoe was concerned with recognition of photographs of faces; I

Digitized by Google

Digitized by Google

consider both body and face. Bledsoe permitted a wide variation in head rotation, tilt, lean, photograph quality, and light contrast; I require much more standardization of pose and can obtain it since I control the picture taking environment. In spite of these differences, the work reported here follows the basic idea for visual identification of people first laid out by Bledsoe: find the measurements and use them for identification. Another way of summarizing this is: automate the identification techniques of Bertillon.

Bledsoe's results verified that facial measurements made on photographs could be used effectively for facial recognition. In his work, measurements were obtained from 2000 photographs, 2 photographs for each person in the sample. Given a set of measurements for an unknown person, the classification system attempted to supply a name or a small list of names which included the unknown individual. Using various classification methods Bledsoe found that an average reduction in uncertainty of about 1/100 could be obtained.

Bledsoe's group was also concerned to a limited extent with finding features automatically (Bisson [1965a], [1965b]). The results of this were inconclusive; many problems were encountered trying to determine the location of feature points. Bledsoe's success with facial classification using measurements while leaving open the

Digitized by Google

Digitized by Google

problem of automatic feature location has been a stimulus for the work reported in this thesis.

Sakai, Nagao, and Fujibayashi [1969] have reported their work on finding faces in photographs. Their goal is to detect if a face or faces are present in a picture. They first produce a picture which contains the edges of the input picture. A large oval template corresponding to the head outline is then matched with the edge picture. All reasonable positions and sizes of the template are tried. In those positions where the oval template receives a high response, the head hypothesis is checked by further template matching that expects many edges in the eyes, nose, and mouth and few edges on the forehead. This method appears to be time consuming, and the result is only an approximate location for the head in the picture.

Three Russians, El'bur [1967], Yurans [1967], and Rastrigan [1967], have presented methods for identifying faces from photographs. The methods assume that a representation of a face as a set of points is available. The papers are mostly on projective geometry; there is very little mention of application. Hart [1969] has prepared a summary of the content of these papers.

Digitized by Google

Digitized by Google

CHAPTER 3

PICTURE ANALYSIS AND DESCRIPTION BY COMPUTER

This chapter will discuss past work in automatic picture analysis and description. Applications will be summarized briefly. Useful techniques which have been developed will then be described in some detail.

APPLICATIONS.

Applications of computer picture processing can be summarized as an area of much past work with little solid success. Many experiments have been performed which have used computers to process and identify visual images. In only one area has this work passed from the experimental to the practical. This area is optical character recognition (OCR). The methods used for OCR will be discussed. Following this some of the other application areas will be surveyed.

Character reading machines are a practical success. To quote a long time leader in the development of the field, "It is now possible to read with any desired accuracy any reasonably good printing whether done by typewriter, high-speed printer, or typesetter" (Rabinow [1968], p.24). In many ways, the problems encountered in reading characters are different from and simpler than the problems found in more general picture processing. Nevertheless, because of

Digitized by Google

Digitized by Google

the great success which has been achieved by OCR, it is appropriate to examine the methods which have been used to achieve this success so that their potential for other types of picture processing may be considered.

One method which has been used for successful OCR is special input, designed for recognition. The many character sets intended for machine reading are examples of this. Rabinow reports building a high resolution machine to read alphanumeric characters specifically designed for machine reading. "It read without error, that is, it has read several billion characters without a single error and the reject rate is something of the order of one character in 2 million" ([1968], p.7). Control of the input with regognition in mind must be considered in any general picture processing task.

Special character sets are not the only characters which can be read by reading machines. In general however it is the fact that printed characters are black against a contrasting background, with more or less sharp edges, that permits the use of straightforward recognition methods with success. A good discussion of the recognition methods used, the subject of the next several paragraphs, may be found in Character Recognition 1967 (British Computer Society [1967]).

Character readers generally recognize characters by template matching. The character "A" for instance is

Digitized by Google

Digitized by Google

compared with idealized specimens of "A", "B", ... , "Z" to determine the best match. The matching is not however a straightforward cross-correlation, which ignores the fact that many characters have common areas and that some areas are nearly unique. Rather, a weighted mask is correlated with the unknown character. In particular, the unknown character is usually represented as grey values on a rectangular matrix A_{ij}. For each character, C, to be recognized there is a weight matrix W^{c}_{ij}. The score for character C is then $S^{c} = \sum W^{c}_{ij} A_{ij}$ (ignoring character positioning). The weights for particular characters are most often obtained heuristically and intuitively. This sort of weighted template matching can be useful in picture processing when the object to be found can be reliably characterized as to shape and relative grey values.

Another recognition method used in OCR is stroke analysis. In this method a character is classified by a group of characteristic properties and features. For instance a "T" might be required to consist of a long vertical stroke with a horizontal stroke at the top. The individual strokes may be found using weighted templates as before. This method is currently used on highly stylized characters. However, it is an example of the sort of two-stage method that is necessary when the input has wide variability.

In contrast to the success of optical character

Digitized by Google

recognition, applications in other areas are only partially successful or are experimental.

Recognition of blocks and simple geometrical solids has been the goal of much interesting recent research. The first work of this nature was done by Roberts in his work "Machine perception of three-dimensional solids" [1963]. This was a computer program which processed and recognized pictures of blocks and wedges. From a photograph, a line drawing of the scene was extracted. The line drawing was processed and a list of the three-dimensional objects in the scene was produced. Once the object list was obtained, various two-dimensional projections of the objects could be dispalyed. This work was particularly noteworthy in two respects. First, solutions were developed for the difficult problems encountered in going from the representation of a picture as a matrix of light intensities to the representation of a picture as a line drawing. Roberts' edge detection operator will be discussed in detail later in this chapter. Secondly, Roberts introduced the use of a four-dimensional, homogeneous coordinate system to handle perspective transformations.

The Stanford University Artificial Intelligence Project is attempting to develop visual and motor capabilities for computers. Building on the work of Roberts, several programs have been written which can recognize blocks on a tabletop. Work is currently in progress directed toward

Digitized by Google

Digitized by Google

recognizing more complicated geometrical solids, processing pictures of roads for a computer controlled car, accommodation of vision system parameters for enhanced visual perception, using texture to distinguish areas of interest in pictures, and developing models and data structures for the representation of scenes. (See McCarthy et al. [1968], Feldman et al. [1969], Falk [1969], and Paul et al. [1969])

At Stanford Research Institute a computer controlled robot is being developed (Raphael [1968], Nilsson [1969]). Vision programs are being supplied which can cope with the robot's environment, a room with large blocks, wedges, and platforms.

Guzman [1968], working at the M.I.T. Artificial Intelligence Project, has developed a program which decomposes a list of lines into a list of objects. Guzman's program can handle scenes of far greater complexity than the programs mentioned above. However, the input to this program is a symbolic and error-free list of the edges in the scene.

From the preceding paragraphs it may be seen that there is much work directed toward computer recognition of blocks and solids. In examining this work, however, the fact stands out that even for simple scenes the computer processing of visual images is difficult and not clearly understood.

Digitized by Google

Digitized by Google

Other application areas where computer picture processing techniques have been used with some success include the following of particle tracks in bubble, cloud, and spark chamber photographs; the processing of photomicrographs, particularly those of chromosomes; the processing of aerial photographs to obtain information on cloud types and terrain; and the visual processing of fingerprints.

EDGE DETECTION.

In picture processing one of the most important problems is edge detection.

Much work on edge detection in digital pictures has been reported. An "edge" is the boundary between two objects or between an object and the background. It is contrasted with "line" which denotes a thin stroke against a uniform background. (The lines in Figure 3-2 represents edges in Figure 3-1.)

Finding the edges in a picture is important, if the picture is to be analyzed and described by a computer. Much of the important information in a picture is contained in the edges. This may be seen in Figure 7-4 which shows the edges present in Figure 7-3. It is evident that most of the information of Figure 7-3 is retained.

Edge detection has received considerable attention as a part of computer picture processing. Roberts [1963], In

Digitized by Google

Digitized by Google

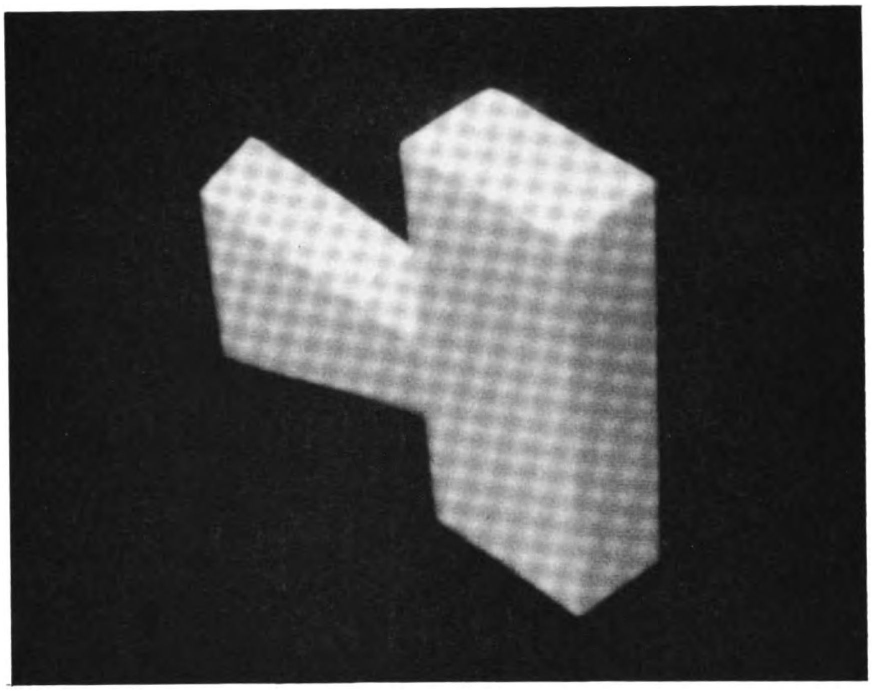


Figure 3-1. Grey scale picture of blocks.

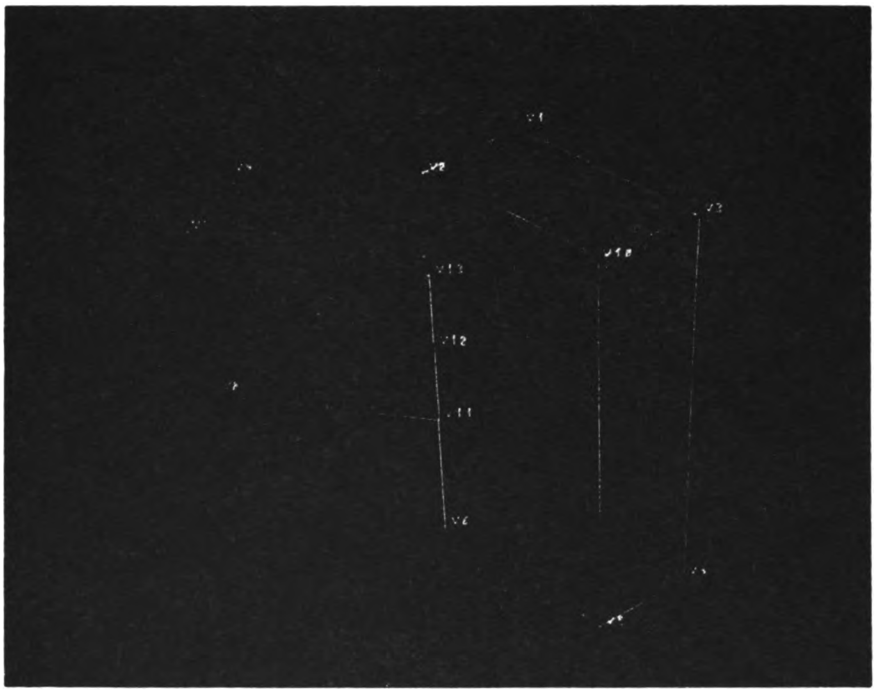


Figure 3-2. Edges from Figure 3-1.

Digitized by Google

Digitized by Google

his pioneering work in machine perception, detected edges in the picture as the first step of processing, Narasimhan and Fornango [1964] emphasized the importance of edges in experiments with pictures of human faces. Guzman [1968], in his work on the analysis and description of scenes, assumed that all edges in the picture had been found accurately. Other research which illustrates the importance of edges in pictures includes the Stanford University Artificial Intelligence Project (Feldman et al. [1969]), the S.R.I. Robot project (Forsen [1968]), and Rosenfeld et al. [1969b]

Edge detection in pictures may be considered a high pass filtering operation. If the high spatial frequencies in a picture are emphasized, and the low spatial frequencies are suppressed, the result approximates a line drawing of the scene. This technique has been used to enhance contrast in lunar photographs (Billingsley [1967]).

Edges in digital pictures are usually detected, however, by the use of local operators. Such operators examine and compare intensity values within a small region of the picture. Most often this operator is some variant of the gradient, the derivative in the direction of the maximum change of intensity. Figure 7-4 is an example of the results of the application of a gradient operator to the picture of Figure 7-3. Mathematically the gradient of a function f(x,y) of two variables is the vector given by

$$\overrightarrow{\mathrm{grad}(x)} = \frac{\partial f}{\partial x}\vec{i} + \frac{\partial f}{\partial y}\vec{j}$$

Digitized by Google

Digitized by Google

where $\vec{i}$ and $\vec{j}$ are unit vectors along the x and y axes. Many different approximations to the gradient have been used. Some of these are discussed below.

$z_{i,j}$	$z_{i+1,j}$	$z_{i+2,j}$	A	B	C
$z_{i,j+1}$	$z_{i+1,j+1}$	$z_{i+2,j+1}$	D	E	F
$z_{i,j+2}$	$z_{i+1,j+2}$	$z_{i+2,j+2}$	G	H	I

Figure 3-3. Notation for points used in describing gradient approximations.

Roberts [1963] used the following discrete approximation to the gradient. Let z be the picture matrix and denote the intensity value at elements $z_{i,j}$, $z_{i+1,j}$, ..., $z_{i+2,j+2}$ with the letters A, B, ..., I as in Figure 3-3. Then the magnitude of the gradient at point $z_{i,j}$ is given by

$$|\vec{G}| = \sqrt{(E - A)^2 + (B - D)^2}.$$

Robert's gradient operator is quite sensitive to noise. To reduce this problem Sobel and Feldman developed a gradient operator which is described by Pingle [1969]. Their approach was to consider the nine points in a 3x3 square. The difference between the intensity at each outer point and the center is weighted by an appropriate vector. Using the notation of Figure 3-3, the gradient at point

Digitized by Google

Digitized by Google

$z_{i+1,j+1}$ is given by

$$\vec{G} = (A-E)[-1,1] + (B-E)[0,2] + (C-E)[1,1]$$
$$+ (D-E)[-2,0] + (F-E)[2,0]$$
$$+ (G-E)[-1,-1] + (H-E)[0,-2] + (I-E)[1,-1].$$

This can be reduced to

$$\vec{G} = U[1,0] + V[0,1]$$

where

$$\frac{\partial f}{\partial x} \approx U = \frac{C + 2F + I - A - 2D - G}{8}$$

and

$$\frac{\partial f}{\partial y} \approx V = \frac{A + 2B + C - G - 2H - I.}{8}$$

It is interesting to note that the lack of sensitivity to noise of this operator is due to the fact that it is equivalent to smoothing and then differencing. Let the picture be smoothed by replacing the intensity at each point with the average intensity in a 2x2 square containing the point, e.g.

$$A' = (A + B + D + E) / 4$$
$$B' = (B + C + E + F) / 4$$
$$D' = (D + E + G + H) / 4$$
$$E' = (E + F + H + I) / 4 \quad \text{etc.}$$

Now let $\frac{\partial f}{\partial x}$ be approximated simply by

$$\frac{\partial f}{\partial x} \approx U' = \frac{B' + E' - A' - D'}{2},$$

If the definitions of the primed variables are substituted

Digitized by Google

Digitized by Google

In this formula, the result

$$\frac{\partial f}{\partial x} \approx U' = \frac{C + 2F + I - A - 2D - G}{8}$$

is exactly the same as Sobel and Feldman's approximation.

Sakai et al. [1969] also use the nine points of a 3x3 square for their approximation to the gradient. Referring again to Figure 3-3, the magnitude of the gradient at point $z_{i+1,j+1}$ is given by

$$|\vec{G}| = E - \min(A,B,C,D,E,F,G,H,I).$$

Rosenfeld and his associates (Rosenfeld et al. [1969b], Abbamonte et al. [1970]) noticed that an edge detector which considers only a few points is sensitive to noise. On the other hand, if an edge detector considers many points over a wider area, it will smooth out noise but it detects major edges in a wide range of positions around the actual position. They suggest that the product of the output of several small and large operators will combine the advantages of both.

The output of the product operator will only be high if all of the constituent operators have high output. Thus at an edge there will be a sharp peak along with noise suppression. To implement this idea, they suggest the use of the following approximation to the gradient. "Let H_k denote the absolute difference between averages taken over two vertically adjacent, non-overlapping k-by-k squares on opposite sides of a given point" [1970, p. 17]. Then at

Digitized by Google

Digitized by Google

that point

$$\frac{\partial f}{\partial y} \approx \prod H_{k_i} \qquad k_i = 1,2,4,8,16$$

and

$$|\vec{G}| \approx \max(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}).$$

Hueckel [1969] has developed an elaborate edge detection operator which considers all points in small circles of the picture. These circles contain from 32 to 177 picture points. Within these circles a set of functions F(x,y) representing ideal edges are defined. The intensities found in an actual circle represent a function E(x,y). Finding the edge in a circle then becomes the problem of finding the particular function F(x,y) which minimizes the distance

$$L = [\ E(x,y) - F(x,y)\ ]^2.$$

The size of L is a measure of how edge-like the edge is. By constraining the problem suitably, Hueckel has found a straight-forward algorithm for implementing this minimization. Hueckel's operator gives good results on circles which contain one edge or at most two edges. His algorithm is, of course, much slower than the operators mentioned earlier that use fewer points.

In addition to the gradient, it has been suggested that higher derivatives (Williams [1965]) or the Laplacian (Rosenfeld [1969a], Bell [1968]) should be used for finding edges. These methods seem unsatisfactory because of their tendency to amplify noise.

Digitized by Google

Digitized by Google

Edge followers are used to combine the edge information produced by a local operator into a line that represents the edge. An edge follower is an algorithm that searches for and follows an edge. It uses the direction and curvature of edges to connect adjacent short segments. Edge following is discussed by Rosenfeld [1969a, p. 136]. Descriptions of working edge followers are given by Pingle [1966], by Pingle et al. [1968], and by Greenblatt et al. [1966]

Besides edge detection one of the most common techniques encountered in picture processing is template matching. This method was mentioned briefly in the discussion of optical character recognition. In OCR this technique has been very useful. However, in more general picture processing problems the technique usually loses its effectiveness. The reason for this is that, except for very simple objects in constrained situations, it becomes almost impossible to develop an idealized prototype.

Another picture processing technique that is generally useful is smoothing. Smoothing is the modification of the grey level at a point in such a way as to make its value more similar to the values of its neighbors. Typically this is an averaging operation. The greatest value of smoothing is that it reduces the noise that is present in almost all pictorial input. It is also helpful in reducing minor surface variations caused by texture and shadow.

Digitized by Google

Digitized by Google

CHAPTER 4

GOAL-DIRECTED PICTURE PROCESSING

TOP-DOWN ANALYSIS.

The program described in this thesis uses top-down picture analysis. It is believed that the explicit use of this approach is one of the major reasons for the success of this program.

The use of the terms "top-down" and "bottom-up" to describe picture analysis methods was originated by Shaw [1968]. The names are derived from a loose analogy which can be drawn to methods with similar names which are used for syntax-directed analysis of programming languages. Such analysis or "parsing" is based on the structure of the programming language. The structure of the program is expressed by a set of "productions" or rewriting rules. The principal methods used are discussed in a paper by Feldman and Gries [1968]. A bottom-up parse of a program starts with the characters and symbols of the program and attempts to combine them together using the productions in reverse order until the whole assemblage represents a correct program. In contrast, a top-down parse works in the opposite direction. Starting from the highest level goal (a program) a search is made among the alternative constituents of each level of production until eventually a series is found which includes the characters of the input.

Digitized by Google

Digitized by Google

In discussing analysis of pictures, "top-down" and "bottom-up" are not used to describe formal mechanisms such as are used in programming language translators. Rather, bottom-up analysis will be used to describe schemes which first process the picture exhaustively at low levels. Top-down analysis will refer to goal oriented processing which searches for the constituent parts of the objects searched for in the program.

Bottom-up analysis has been used widely in picture analysis. It consists of a series of more or less independent processing steps which are applied sequentially in an attempt to transform an input picture into a description of the picture. Each stage of processing reduces the amount of information that will be passed on to the following stage. The idea is to first concentrate on the lowest level of detail and then to consider higher levels one at a time until finally the entire picture has been analyzed and described.

The schematic diagram given in figure 4-1 is an example of the typical organization of a bottom-up processor. The first stage is called region analysis. This usually consists of low level operations such as detecting edges or homogeneous areas. The output of this first stage would be a list of all of the edges or all of the regions. The second stage is called region description. This part of the program collects subsets of its input into recognizable

Digitized by Google

Digitized by Google

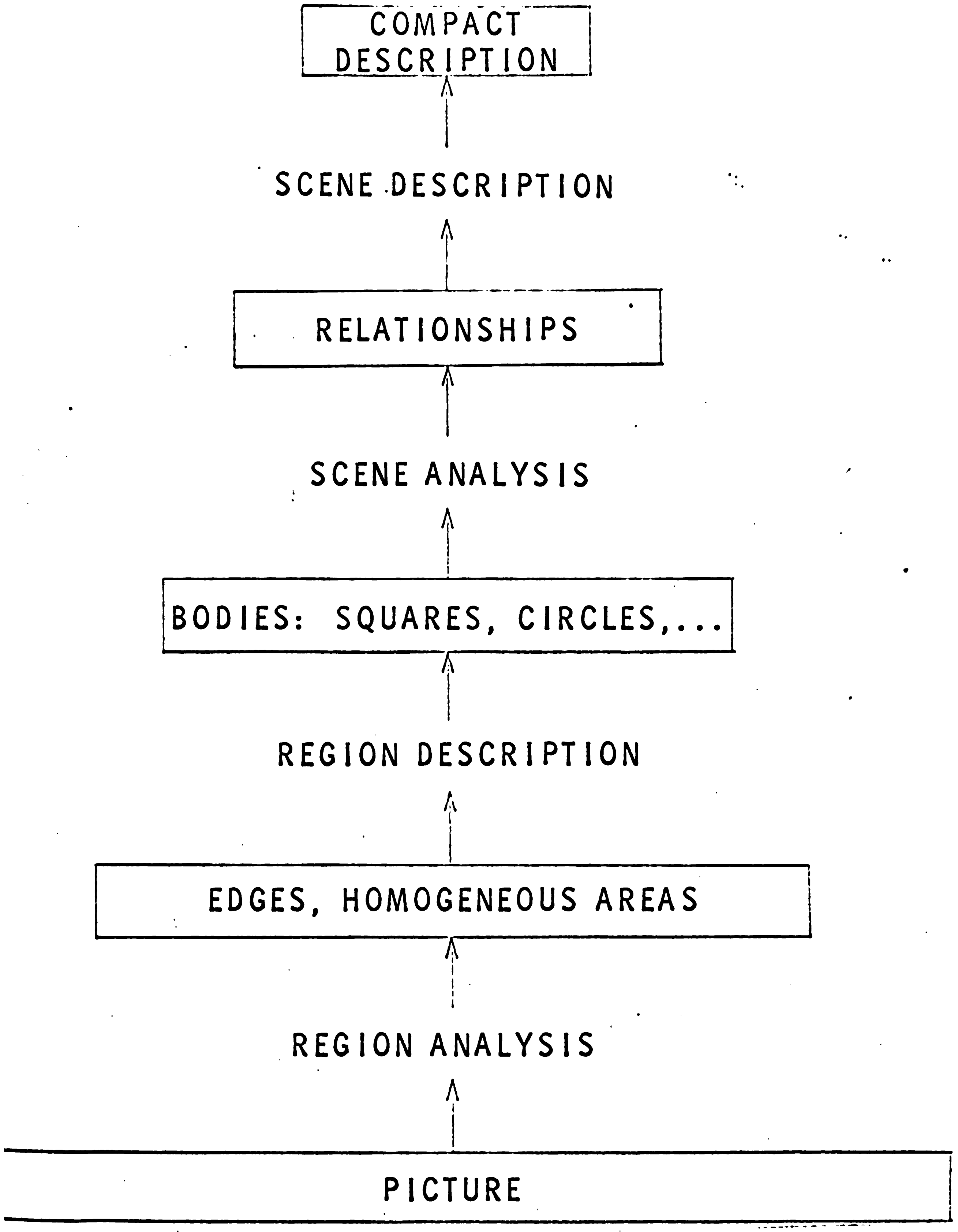


Figure 4-1. Logical flow of a bottom-up picture analyzer.

Digitized by Google

Digitized by Google

small pieces of the picture. For instance, the output of this stage might be a list of rectangles or ellipses or blobs of a certain shape along with their coordinates. The third stage is called scene analysis. In this stage relationships between regions are considered. Examples of such relationships are "next to", "above", "close to", or "within". Thus the output of this stage contains the relationships between the regions in the picture. The last stage is called scene description. Here the final description is obtained. For instance, a square adjacent to two parallelograms may represent a cube, or an oval which contains two circles, two dots, and a line may represent a face.

Most picture processing programs are organized along the lines of the bottom-up model described above (see Feldman et al. [1969]). However, for complex pictures the bottom-up method fails. The reason for this failure is that in cluttered pictures the program must know what it is looking for in order to find it. The bottom-up strategy would work if pictures were perfect and noiseless. However, real pictures contain much noise and irregularity. Accordingly, at each stage of processing the input contains false information as well as missing information.

To avoid the problems inherent in bottom-up organization, a top-down, goal directed approach should be taken. In such an approach the program knows of certain

Digitized by Google

Digitized by Google

classes of objects which may be present in a picture. The goal of the program is to try and find these objects if they are present. The constituent parts of each top level goal are known, as well as the parts of the parts down to the lowest level. The program searches for some of these basic parts, namely those that can be most reliably detected. If a particular low level part is found, then, because the structure of the object is known, some other known part should be present at an adjacent location. A specific routine can examine the area in question in detail to determine if such a part is present. In this way, for instance, the edge of a block might be detected once a corner is found, even though the edge might be too weak to be passed through the threshold of a gradient operation. As more and more adjacent constituent parts of a goal object are found, the confidence that an object has been found increases. Prediction of the location of the remaining parts of the object becomes easier and easier. Fairly straightforward methods can be used to verify whether these parts are indeed present. Thus, as recognition proceeds the amount of work to be done diminishes.

Top-down analysis is needed because it considers global information about the structures and interrelationships of the objects in the picture. It would be desirable for that part of the program which directs the search for the constituents of a goal object to be general purpose. To

Digitized by Google

Digitized by Google

such a general purpose program one could specify in some higher level notation or "language" the structure of objects. For instance, a "head" could be described as:

- round on top.
- somewhat flat on the sides.
- curving inward for the neck below the sides.
- etc.

The same program could then be used to identify cows or battleships (for instance) using a different structural description. Such high level descriptions are the goal of picture description languages.

Picture description languages have much promise as a tool for picture analysis. A number of descriptive notations and processing algorithms have been suggested for this purpose (Narasimhan [1966]; Lipkin, Watt, and Kirsch [1966]; Ledley [1965]; Miller and Shaw [1967]; Clowes [1969]; Pfaltz and Rosenfeld [1969]). Excellent discussions of these linguistic picture processing techniques can be found in the paper by Miller and Shaw [1968] and in Rosenfeld's book [1969a, Chapter 10].

In general, picture analysis using picture description languages proceeds as follows. A set P of "pieces" or "primitives" is defined; these are the basic elements from which a picture description can be built up. A set C of composition operations is defined. A set R of rules or "productions" specifies allowable combinations of elements

Digitized by Google

Digitized by Google

of P and C. Then F becomes a set of pictures that can be built up using R, P, and C. If x is an unknown picture belonging to F, then F can be recognized if we can recognize the elements of P in it and the way these primitives are combined.

This sort of analysis works for simple pictures but as the pictures become complex the method breaks down. For complex pictures a choice must be made. Very simple primitives may be chosen which are easy to recognize. In this case R, the allowable rules of combination, becomes so complex and unwieldy that recognition of these combinations becomes impossible. If instead, complex primitives are chosen, these become as difficult to recognize as the picture itself.

A possible solution to this problem is to provide hierarchies of structural descriptions. Straight-forward application of hierarchical decomposition leads back, however, to a bottom-up approach to processing where one loses sight at the lower levels of the top-level structure.

Analysis of complex pictures using picture description languages will not be possible unless powerful and systematic algorithms for processing elaborate and complicated picture descriptions can be developed.

The problems discussed above can be summarized as follows. Picture description languages are not sufficiently well developed to permit the specification of the numerous

Digitized by Google

Digitized by Google

heuristic tests and considerations of interrelated structure necessary to find objects in cluttered, grey-level pictures. Because of this a general purpose analysis program based on a picture description was not used.

The methods that are used to implement goal-oriented processing are "planning" and "models". Planning will be discussed in the next chapter. The use of models is discussed briefly below.

MODELS.

A model is a description which contains some of the structure of the thing which is represented. The idea of using models of objects to be recognized by computer seems to be implicitly present in most picture recognition programs. The idealized "A" represented by a resistor matrix in a character reader certainly is a model of an "A". It is only in recent years, however, that it has become evident that the model should be used to direct all levels of processing, not just a top level decision. Reddy [1969] discusses the explicit use of such models in his recent paper. Feldman and his associates at the Stanford Hand-Eye project are working on developing data structures and representations for models of geometrical objects. Bledsoe [1964] suggested using models of the face as a method of attacking the problem of locating facial features. If an object has been described in a picture description language,

Digitized by Google

the description is, of course, a model of the object.

Models of the structure of human bodies and faces are used throughout the processing programs discussed in later chapters and guide these programs. The form of these models varies widely. The matrix of average light intensities models the background of pictures of people. A table of average ratios between facial features models a face. A flow chart of the subroutine which distinguishes nostrils from noise models a "reasonable" nose. Because of the wide variety of uses of and needs for these models, no standard modeling formalism was used. Nevertheless, at every stage a conscious attempt has been made to incorporate goal-oriented processing directed by a model of the object desired.

Digitized by Google

Digitized by Google

CHAPTER 5

PICTURE PROCESSING USING "PLANNING"

This chapter discusses a method for performing goal-directed picture analysis. The method is an improvement on the most commonly used procedures because it considers global information more efficiently than traditional local techniques. The method is called "planning" in this paper, a term drawn from artificial intelligence research, for the method precisely fits the description of planning given by Minsky [1961]. It is believed that planning can be useful in many projects involving computer vision. In addition, it is helpful to relate the search for objects or features in pictures to the work on reduction of a given search space in the field of artificial intelligence.

Much of picture processing can be viewed as search. The space to be searched is the matrix of light intensities which represents the picture. The goal of scene analysis and description is locating the important objects in the picture. The intensity matrix must be searched to find an eye, a chromosome, or the corner of a cube. Many of the difficulties encountered in picture analysis are due to the large size of the search space. Truly, one cannot see the forest for the trees. If a processing algorithm attempts to detect features by considering local areas, as most

Digitized by Google

Digitized by Google

algorithms do, the great detail which is present in the picture obsoures the larger features, A global strategy is needed.

In the field of artificial intelligence, global search is one of the central problems. Much research has been done on developing heuristics for reducing search in such fields as problem solving and game playing. A generally useful concept which has emerged is planning, first used by Newell, Shaw, and Simon in their General Problem Solver [1959].

Planning is discussed by Minsky in his paper "Steps toward Artificial Intelligence" [1961]. Artificial Intelligence is considered to be mechanization of the problem solving process. Basic techniques for problem solving include search, pattern recognition, learning, planning, and induction. The following quotations from Minsky's paper define planning.

Planning is the analysis of problem structure in the large. "For really difficult problems, ... step-by-step heuristics ... will fail, and the machine must have resources for analyzing the problem structure in the large - in short, for 'planning' " [p. 21].

> "Perhaps the most straightforward concept of planning is that of using a <u>simplified</u> <u>model</u> of the problem situation. Suppose that there is available, for a given problem, some other problem of essentially the same character with less detail and

Digitized by Google

Digitized by Google

complexity. Then we could proceed first to solve the simpler problem. Suppose, also, that this is done using a second set of methods, which are also simpler, but in some correspondence with those for the original. _The solution of the simpler problem can then be used as a 'plan' for the harder one_." [p. 25]

The steps outlined in the preceding paragraph can be applied almost directly to the problem of finding desired objects in a picture. This represents an entirely new application for planning. It has not been used previously in picture processing.

In brief then, picture search using planning consists of three simple steps. A new digital picture is prepared from the original; the new picture is smaller and has less detail. Objects are tentatively identified in the reduced picture. The tentative analysis of the reduced picture, a list of the objects found and their locations, is used as a plan to verify the presence of edges in the original picture.

In a sense, the idea of using planning in picture analysis is not new. Kirsch et al. [1957] gave suggestions for possible use in processing pictorial information with a digital computer. One of the suggestions is that a defocused preliminary scan of a picture should be made. The averaging and size reduction in my implementation of

Digitized by Google

Digitized by Google

planning is very similar to defocusing and sampling. In spite of this early recommendation for a defocused scan, in the years since then the idea has been forgotton and ignored.

AN EXAMPLE: EDGE DETECTION USING PLANNING.

It was pointed out in Chapter 3 that edge detection is necessary for picture analysis and description. Many methods for detecting edges by computer were presented. In spite of much effort, the accurate selection by computer of "important" edges in fairly cluttered scenes has not been particularly sucessful. The methods given for edge detection are fine for presenting information to the human eye (as Figure 3-4 shows), but what is desired is scene analysis and description within the computer. In reaching for this latter goal problems of noise, disappearing edges, and false edges intrude. Brief comments about the limitations of these methods encountered at the Stanford Artificial Intelligence Project have appeared (Pingle et al. [1968], McCarthy et al. [1968], Paul et al.[1969]). Nilsson [1969] states the problems with local methods of edge detection from experience at the S.R.I. Robot project:

> "The line drawing often contains flaws that seriously complicate its analysis. Some of these flaws could be corrected by more elaborate local processing. However, there is a limit to how well

Digitized by Google

Digitized by Google

local processing can perform, and when significant edges cannot be told from insignificant edges on the basis of local criteria, the goal of producing a perfect line drawing in this way must be abandoned." [p. 513]

Global information about the edges in the picture and the structure of the objects they represent is needed. This is the idea behind the application of linguistic techniques which was mentioned earlier. Another approach to incorporating global structure into edge detection is the decision tree in use at S.R.I. (Nilsson [1969]).

The global methods for edge detection also encounter problems. Here the program looks for specific edges in specific relationships. The search space is very large, tens of thousands of points in a picture, and the combinatorics of the problem force unacceptably long search times. In the picture there are just too many lines and there is too much looking to do.

The difficult problem of locating important edges in pictures may be approached by using the three steps of planning.

1. Extract a simplified problem from the given problem. To simplify the problem, a much smaller picture is prepared with the intensity at each point equal to the average intensity over an area of the original picture. The new problem: find the important edges in the new small

Digitized by Google

picture.

The new problem is much easier than the original problem. Many of the small features in the original picture are no longer present. The picture has been greatly reduced in size. Only the important features of the picture remain. Less detail means fewer edges, a smaller search space, and much faster processing. Because of the smoothing done while averaging intensities, the small picture will contain little noise. Faint and blurred edges of large objects will be enhanced.

2. Solve the simpler problem. The techniques for finding edges discussed in Chapter 3 can be applied to find the desired edges in the small picture. Since only the principal objects from the original picture now remain, only major edges will be detected. As short edges are collected to form shapes they may be tested as to their acceptability. This testing should incorporate knowledge of acceptable shapes for the objects to be recognized. Since there are few edges to consider, false paths are found relatively quickly. Backtracking is far less of a problem since the data structures which must be erased are much smaller.

3. Use the solution to the simpler problem as a guide (a plan) for solving the actual problem. Within the small picture, certain edges have been found. Now it is a fairly simple matter to return to the larger picture and find the desired edges accurately. For example, a straight line may

Digitized by Google

Digitized by Google

connect points P′ and Q′ in the small picture. P and Q are the corresponding points in the large picture. Therefore we know that in the large picture there is an approximately straight edge which runs from the vicinity of point P to the vicinity of point Q. Since the approximate location and direction of this edge is known, it is possible to detect it quite easily and accurately. The search for this edge can be confined to a narrow band between P and Q. It will be easy to detect a false path for it will soon diverge from this narrow band.

The remainder of this section will discuss briefly some observations about edge detection using planning.

Is "planning" merely the application of a large edge detection operator? Certainly the search for edges made in the small picture could be a search of the large picture using an operator which examines a 24 x 24 point square. In fact, the two steps, large picture to small picture and small picture to small edge matrix, could be one step, the application of a 24 x 24 point operator. Alternatively, the entire large picture could be heavily smoothed and the search for edges could be done in this smoothed picture. Planning, however, has two advantages over either of the approaches listed above. First, it is much easier to design and debug the program which searches for significant edges when using planning. This is because the reduced picture is so much smaller. The use of planning aids insight. While

Digitized by Google

Digitized by Google

designing a program to find the edges in a picture of 625 points (for instance), the designer can comprehend easily both the overall structure and the detailed structure of the picture. It is very difficult to obtain the same level of understanding when the picture contains 40,000 points. The second advantage of planning is speed. Planning is 40 to 80 times faster than the same search without planning. (This factor was obtained during tests described in detail in Chapter 7.) This increase of speed is particularly important when designing search programs on an interactive computer system.

If a program incorporated more elaborate structural knowledge of the object to be recognized, could planning in a picture of reduced size be eliminated? No, such a program must be less effective for the following reasons. The combinatorics of the large search space will require excessive time. The effect of noise and irregularities in the picture will be even worse. This is emphasized by the elaborate provisions for search that had to be built into the plan follower. Even when the direction and approximate location of an edge is known, its detection can be hard. For such edges, planning is essential.

An improvement on the planning technique presented in this paper would be recursive application of planning at varying size reductions. For instance, the original picture could be reduced in size twice, four times, and eight times.

Digitized by Google

The plan found in the 1/8 size picture could be used to find edges in the 1/4 size picture, This could then be used as a plan to find edges in the 1/2 size picture. Finally the accurate true edge would be found in the original picture. A hint of such a method was mentioned in the description of the plan follower, The local edge finding operator examined only alternate points, It could be considered to be using a 1/2 size picture. The program could decide itself how much size reduction and averaging to do, At each level, if there is too much detail, a smaller picture could be called for.

Digitized by Google

Digitized by Google

CHAPTER 6

LOCATING AND MEASURING THE BODY

This chapter describes the experimental environment in which this work was done, and explains the methods used to locate measurement points in a full length picture of the person to be identified.

EQUIPMENT, PROGRAMS, AND INPUT SPECIFICATIONS.

This work was performed using the facilities of the Stanford Artificial Intelligence Project. The system consists of two computers, a PDP-10 and a PDP-6, connected as dual-processors sharing 130,000 words of core memory; a very fast, head-per-track Librascope disk for swapping; and an IBM 2314 disk storage unit for program and data storage. Several types of display consoles as well as teletypes are available for use as terminals. A more detailed description of the computer system can be found in McCarthy et al. [1968] All of the programs are written to run under the Stanford time-sharing monitor (Moorer [1969]).

Picture input is obtained from a standard vidicon TV camera. The video signal is digitized by a analog-to-digital converter and sent to memory via a high speed data channel. During one TV reading operation up to 333 samples can be taken from each video scan line and up to 256 of the alternating scan lines can be read. Thus the

Digitized by Google

Digitized by Google

maximum size picture which can be read is 256 x 333 points. Such a picture covers the entire field of view of the TV camera. Each point consists of a four bit light intensity value so that sixteen levels of grey are available. Zero indicates the darkest points; fifteen, the brightest.

The program is written almost entirely in Fortran. A few subroutines are coded in assembly language, such as those that handle input-output, list processing functions, dynamic storage allocation for picture buffers, and partial word operations. The programs occupy about 30,000 words of storage. Data storage, assigned dynamically, typically requires another 20,000 words, primarily for buffers for processed and unprocessed pictures.

CHARACTERISTICS OF THE PICTURE.

The person to be recognized stands in a standardized position against a normal room background. The person is told, "Stand in a relaxed position of 'attention', feet together, arms at your sides. Look directly at the TV camera." Figure 6-1, an input photograph of a person to be recognized, gives an example. Minor variations in position are permitted. It would, of course, be impossible to prevent such variations.

The background may vary. It normally consists of the objects in the computer room where these experiments were performed. There are several reasons for not requiring a

Digitized by Google

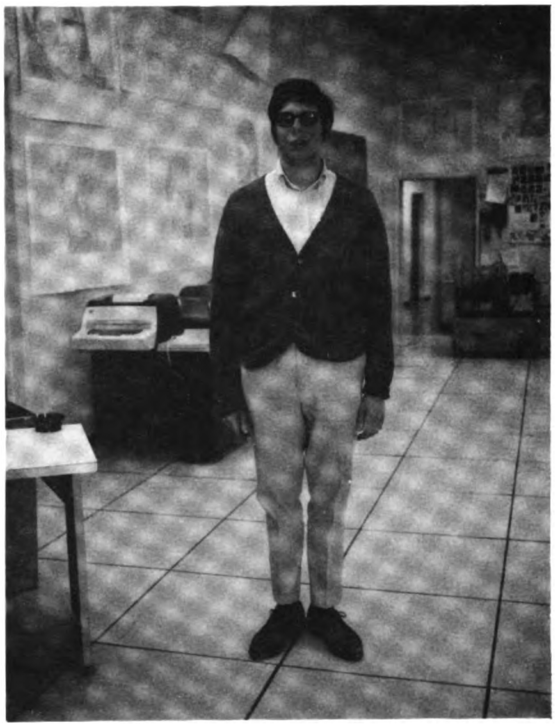


Figure 6-1.
Input photograph.
Subject in standard
recognition position.

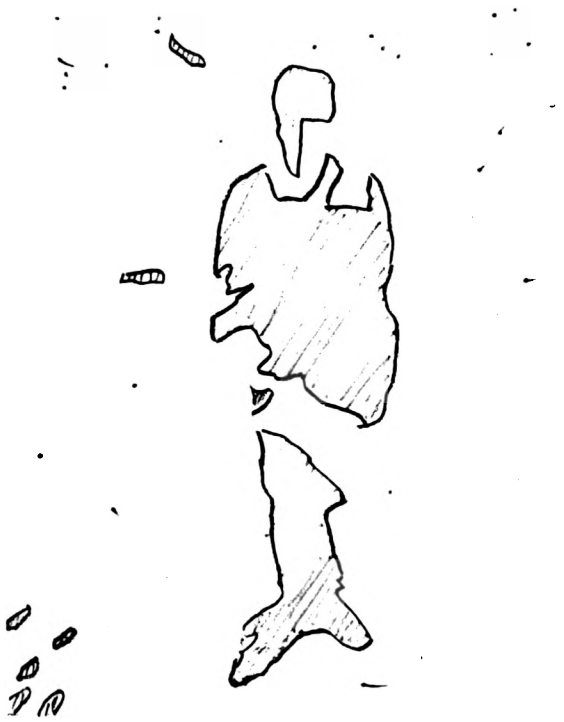


Figure 6-2.
S'. Binary valued,
smoothed, reduced size
picture used for
locating the body.

Digitized by Google

Digitized by Google

standard background. When this work was begun a door-size standard background was built. However, it was still difficult to find the outline of the person. Shadows from the individual being recognized and non-uniform lighting on the background presented new problems. The solution developed for these problems permits a wider variety of backgrounds. Since the more general case is easy to handle, it appeared worthwhile to do so. Another factor which helps in permitting various backgrounds is the small depth-of-field which is characteristic of the TV lenses which were used. Objects that are very far behind the person are out of focus, hence blurred and washed out.

APPROXIMATE LOCATION OF THE BODY.

The first step in processing is to determine the approximate location of the body. A flow chart of this process is given in Figure 6-3. Each of the steps is described in more detail below.

Before beginning identification, a picture M (model) is taken of the background with no person in the picture. This picture is saved until the picture B (body) of the person is taken. Thus when processing of the picture of the body begins there are two pictures available to the program; both have been taken from identical camera angles and are the same size. One is a picture of the background, the other contains a man in front of the background.

Digitized by Google

Digitized by Google

```
                    -------------
                    |   read M   |
                    -------------
                          ↓
                          ↓
                    -------------
                    |   read B   |
                    -------------
                          ↓
                          ↓
                 ------------------
                 |   S ← | B-M |   |
                 ------------------
                          ↓
                          ↓
       ------------------------------------------
       |   S' ← S reduced in size by averaging   |
       ------------------------------------------
                          ↓
                          ↓
       ------------------------------------------
       |   Convert S' to binary-valued picture   |
       ------------------------------------------
                          ↓
                          ↓
              -----------------------------
              |   Fill in isolated holes   |
              -----------------------------
                          ↓
                          ↓
           ----------------------------------
           |   Find all connected regions,   |
           |      Count points in each.      |
           ----------------------------------
                          ↓
                          ↓
             ------------------------------
             |   Delete "small" regions.   |
             ------------------------------
                          ↓
                          ↓
    ------------------------------------------------
    |   Delete regions that intersect the border.   |
    ------------------------------------------------
                          ↓
                          ↓
     -----------------------------------------------
     |   Draw rectangle around remaining regions.   |
     -----------------------------------------------
```

Figure 6-3.
FLOW CHART - APPROXIMATE LOCATION OF THE BODY.

Digitized by Google

Digitized by Google

In a very simple, yet very real sense, the picture M is a model of the background of picture B. If M and B are compared, those areas of B that are similar to M should be background. Those areas of B that do not match the model M are areas where a disturbance has entered the picture. These are the areas where the man is located.

The two pictures are subtracted. That is, a new picture S is formed with the intensity at each point of S equal to the absolute value of the difference of the intensities at corresponding points in M and B.

$$S_{ij} = | B_{ij} - M_{ij} |$$

In a simple world, S would be non-zero only at points on the man and at all of these points. In reality there are shadows and noise which produce false readings. In addition, some areas of the individual's clothes may be similar in intensity to the background and give no difference after subtraction. This may cause the non-zero areas of S to be subdivided into several parts or to contain holes. Therefore the first thing that must be done is to distinguish the blobs that make up the man from stray background blobs. This can be done using considerations of size and central location.

The subtracted picture S is reduced in size n times by averaging. The new picture is designated S''. The intensity at each point S''_{ij} is the average intensity over an nxn square of S. The averaging reduces noise; the size

Digitized by Google

Digitized by Google

reduction speeds up subsequent processing. Various values of n were tried. A large n speeds up processing but leads to considerable uncertainty in the location of the blobs that are identified. An n of 6 or 8 seems to be the most satisfactory compromise.

S'' is now converted to a binary valued picture S' by thresholding.

$$S'_{ij} = 1 \quad \text{if } S''_{ij} \text{ is greater than } 1.$$
$$= 0 \quad \text{otherwise.}$$

The threshold value which is used, 1, insures that most points set to zero are truly background. An example of S' as it appears at this stage is given in Figure 6-2.

Isolated holes in S' are filled. An isolated hole is a point with value 0 that is surrounded on 4 sides with 1's. The value of such a point is set to 1.

In the picture S', connected regions (subsets of the picture) are identified and labeled. Two points, S'_1 and S'_n are <u>connected</u> if there exists a sequence S'_1, S'_2, S'_3, ... , S'_n such that S'_i is a neighbor of S'_{i-1}. Two points are <u>neighbors</u> if they are immediately adjacent horizontally or vertically; i.e., the neighbors of S'_{ij} are $S'_{i,j-1}$, $S'_{i,j+1}$, $S'_{i+1,j}$, and $S'_{i-1,j}$. A subset of the points of S' is a <u>connected region</u> if all points in that subset are <u>connected</u>. Connectivity, as defined above, is often called 4-connectivity (Rosenfeld [1970]).

The program that determines the connectivity of the

Digitized by Google

Digitized by Google

reduction speeds up subsequent processing. Various values of n were tried. A large n speeds up processing but leads to considerable uncertainty in the location of the blobs that are identified. An n of 6 or 8 seems to be the most satisfactory compromise.

S'' is now converted to a binary valued picture S' by thresholding.

$$S'_{ij} = 1 \quad \text{if } S''_{ij} \text{ is greater than 1.}$$
$$= 0 \quad \text{otherwise.}$$

The threshold value which is used, 1, insures that most points set to zero are truly background. An example of S' as it appears at this stage is given in Figure 6-2.

Isolated holes in S' are filled. An isolated hole is a point with value Ø that is surrounded on 4 sides with 1's. The value of such a point is set to 1.

In the picture S', connected regions (subsets of the picture) are identified and labeled. Two points, S'_1 and S'_n are <u>connected</u> if there exists a sequence S'_1, S'_2, S'_3, ... , S'_n such that S'_i is a neighbor of S'_{i-1}. Two points are <u>neighbors</u> if they are immediately adjacent horizontally or vertically; i.e., the neighbors of S'_{ij} are $S'_{i,j-1}$, $S'_{i,j+1}$, $S'_{i+1,j}$, and $S'_{i-1,j}$. A subset of the points of S' is a <u>connected region</u> if all points in that subset are <u>connected</u>. Connectivity, as defined above, is often called 4-connectivity (Rosenfeld [197Ø]).

The program that determines the connectivity of the

Digitized by Google

picture is a procedure that is used several times in the processing described in subsequent chapters. It is an efficient procedure that determines the connectivity during one pass over the picture array. The labelling is given by a matrix of region numbers for each point and a table of equivalences. The algorithm is given in the form of an Algol procedure in Figure 6-4.

This algorithm was developed by the author. Subsequently, it was discovered that the underlying idea was first published in 1957 and has been re-invented several times since. (See the references in Rosenfeld [1969, p.138] as well as Lourie [1969]). However, none of these references give a detailed algorithm.

Now that connected regions in the picture are identified, it is easy to delete small regions and regions which intersect the border. A parameter, SM, determines what constitutes a small region. Four was found to be a satisfactory value for SM for the pictures in this thesis. This value of course depends on the resolution of the video input device.

At this point all that should remain in S' are those connected regions which make up the body of the man. The extremes of these regions, horizontally and vertically, define a rectangle in which the man is located. All further processing is done only within this rectangle. The rectangle gives a first approximation to the size of the

Digitized by Google

Digitized by Google

```
procedure CONNECTED (A, REGNR, REGTBL, NREG, M, N, P);
     value M, N, P;
     Boolean array A[Ø:M,Ø:N];
     integer array REGNR[Ø:M,Ø:N];
     integer array REGTBL[Ø:P];
     integer NREG, M, N, P;
     comment
          This procedure labels connected regions in a
          binary valued picture, Four-connectedness is
          used. A is the picture matrix. To each "true"
          element of A a positive integer will be assigned
          such that all elements in a given connected region
          receive the same value, while elements in different
          connected regions receive different values. This
          positive integer is called the "region number". On
          exit from this procedure the region number
          for point A[I,J] is in REGTBL[REGNR[I,J]]. The
          region number for any "false" point will be zero.

          Also on exit, NREG will contain the number of
          distinct connected regions in the picture.

          P, the length of  REGTBL, must be at least as great
          as NREG.

          For clarity, some of the administration required
          is omitted from this routine. There is no
          checking to see if REGTBL overflows and the
          "garbage collection" necessary to handle this
          overflow is omitted. Special case treatment of
          the picture borders is not included. Instead, it
          is assumed that for all I,
                  A[Ø,I] = A[I,Ø] = false
          and     REGNR[Ø,I] = REGNR[I,Ø] = Ø;
```

Figure 6-4.
PROCEDURE CONNECTED.

Digitized by Google

Digitized by Google

```
begin
    integer I, J, K, L, N1, N2, SMALL, LARGE;
    comment
        I, J, K, are running indices.
        L is the index of the last element of REGTBL
        in use.
        N1 and N2 are used to contain the current
        region numbers of the picture elements
        immediately above and to the left of A[I,J].
        SMALL and LARGE are temporaries;
    L := Ø;
    REGTBL[Ø] := Ø;
    for J := 1 step 1 until N do
    for I := 1 step 1 until M do
    if A[I,J] then
    begin
        N1 := REGTBL[REGNR[I,J-1]];
        N2 := REGTBL[REGNR[I-1,J]];
        if N1 = Ø ∧ N2 = Ø then
            begin
                comment Both neighbors are zero so
                assign a new region number;
                L := L+1;
                REGTBL[L] := L;
                REGNR[I,J] := L;
            end
    else if N1 = Ø then REGNR[I,J] := N2
    else if N2 = Ø v N1 = N2 then REGNR[I,J] := N1
        else
            begin
                comment Both neighbors are non-zero.
                Set up an equivalence between their
                region numbers;
                SMALL := if N1 < N2 then N1 else N2;
                LARGE := if N1 < N2 then N2 else N1;
                for K := LARGE step 1 until L do
                if REGTBL[K] = LARGE then
                    REGTBL[K] := SMALL;
                REGNR[I,J] := SMALL;
            end;
    end;
    comment Count the regions;
    NREG := Ø;
    for K := 1 step 1 until L do
    if REGTBL[K] = K then NREG := NREG + 1;
end;
```

Figure 6-4. (continued from previous page)

Digitized by Google

Digitized by Google

man. This size information indicates such things as the size of the head to be searched for.

MEASURING HEIGHT.

The height of the person in the original picture is now measured. This is done by locating the top of the head and the feet and measuring the distance between them.

The top of the head is found first. The approximate location of the head is known from the previous processing of the reduced picture. It is still necessary to examine closely an area of the original picture to pinpoint the location of the top of the head.

The top of the head is found by a template matching operation. The reason that template matching works at this point is because the search region has been narrowed down to a small area. If an attempt was made to pass a head template over the entire picture many false responses would be encountered. This problem is minimized when searching a limited area.

A curved template has been made by considering a number of pictures of heads. Figure 6-5 is an example of such an "ideal" top-of-the-head. The size of the template is varied according to the approximate height of the individual which was obtained during the approximate location step.

The template is represented in the computer as a matrix of Ø's, +1's, and -1's. A typical template is shown in

Digitized by Google

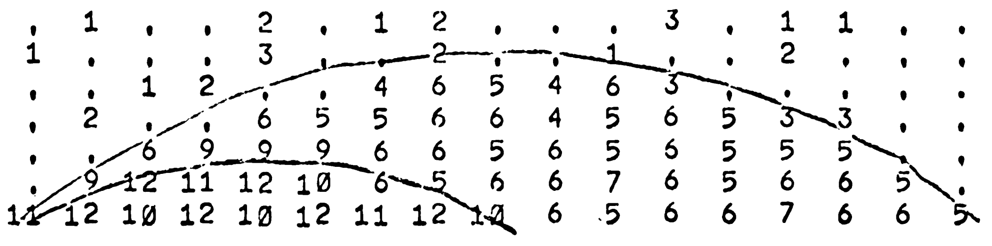
Digitized by Google

```
 ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,
 ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,
 ,   ,   ,   ,   ,   ,   6   6   6   6   6   ,   ,   ,   ,   ,   ,
 ,   ,   ,   ,   6   6   6   6   6   6   6   6   6   ,   ,   ,   ,
 ,   ,   6   6   6   6   6   6   6   6   6   6   6   6   6   ,   ,
 ,   6   6   6   6   6   6   6   6   6   6   6   6   6   6   6   ,
 6   6   6   6   6   6   6   6   6   6   6   6   6   6   6   6   6
```

Figure 6-5. Ideal top of head.
("," stands for "Ø" in the figures on this page.)

```
-1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1  -1   ,   ,   ,  -1  -1  -1  -1  -1  -1  -1
-1  -1  -1  -1   ,   ,   ,  +1  +1  +1   ,   ,   ,  -1  -1  -1  -1
-1  -1  -1   ,  +1  +1  +1   ,   ,   ,  +1  +1  +1   ,  -1  -1  -1
-1  -1   ,  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1   ,  -1  -1
-1   ,  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1   ,  -1
 ,  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1  +1   ,
```

Figure 6-6, Template for top of head.



```
 ,   1   ,   ,   2   ,   1   2   ,   ,   ,   3   ,   1   1   ,   ,
 1   ,   ,   ,   3   ,   ,   2   ,   ,   1   ,   ,   2   ,   ,   ,
 ,   ,   1   2   ,   ,   4   6   5   4   6   3   ,   ,   ,   ,   ,
 ,   2   ,   ,   6   5   5   6   6   4   5   6   5   3   3   ,   ,
 ,   ,   6   9   9   9   6   6   5   6   5   6   5   5   5   ,   ,
 ,   9  12  11  12  1Ø   6   5   6   6   7   6   5   6   6   5   ,
11  12  1Ø  12  1Ø  12  11  12  1Ø   6   5   6   6   7   6   6   5
```

Figure 6-7. Actual top of head.

Digitized by Google

Digitized by Google

Figure 6-6. This template is cross-correlated, in the area where the head is located, with the picture S (mentioned earlier) which contains the absolute value of the difference in intensity between the picture of the man and the picture of the background. In S it is expected that the top of the head will be somewhat like Figure 6-5. Above the head the values should be zero or close to it. Within the head the values should be positive.

As in all picture processing, noise and imperfections make the simple become difficult. If the picture were noiseless one could look for the step from zero background to non-zero head. However, the background is noisy and the interior of the head has steps of intensity within it that may be greater than the steps from background to head. Figure 6-7 gives a stylized example of these problems. The background has an average intensity of about 1. Most of the head has an average intensity of about 5. Within the head there is a curved area with average intensity near 11. In a straight-forward cross-correlation the step from 5 to 11 would give a higher response than the step from 1 to 5. This would be undesireable.

What is desired is to find the step from "low" background values to higher head values. In practice, it is not effective to define "low" background with a threshold for there is no natural way to set this threshold. Instead, the cross-correlation is computed using logarithms

Digitized by Google

Digitized by Google

to emphasize low steps of intensity.

The template T is matched with a rectangular subset of S as follows.

$$V_{i,j} = \sum_{p,q} T_{p,q} \ln(S_{i+p,j+q} + 1)$$

This is easily implemented efficiently because S is limited to sixteen grey levels, 0 to 15. The logarithms can be computed once and stored in a table for easy access during cross-correlation.

$V_{i,j}$ is the value of the cross-correlation at a particular point. The maximum $V_{i,j}$ found gives the location of the top of the head and the coordinates of the point are recorded.

Finding the feet is quite similar to finding the top of the head. An examination of Figure 6-1 shows that in the "standard" position the subject stands with heels together and toes apart. In this position, the outer sides of the toes form a characteristic pattern on the floor. Thus the feet can be found by template matching in the area where they are known to be approximately located. The logarithmic cross-correlation is used as with the head. The coordinates of the feet are recorded.

The height of the person is now known in picture units (raster units). This value is stored as the first measurement which characterizes the person.

Digitized by Google

Digitized by Google

MEASURING WIDTHS.

The width of the head, the neck, the shoulders, and the hips are now measured. Figure 6-8 shows where each of these measurements is taken.

The determination of the best place to take these measurements (e.g., the widest part of the head, the narrowest part of the neck) can be difficult. This difficulty is avoided by measuring at standard positions.

Let the measurements for width of head, neck, shoulders, and hips be designated by m1, m2, m3, and m4. Assume that the height of the subject is h raster units. Assume further that the feet are located at (0,0) and the head at (0,h) as in Figure 6-9. Then on a "typical" person the "best" place for measuring m would be at $(-x_i, y_i)$ and (x_i, y_i) where $x_i = w_i\,h$ and $y_i = r_i\,h$. The values used for w_i and r_i are given in Table 6-1. These values were obtained by hand measurements on a number of pictures.

Unfortunately, often the position of the body is not vertical. The TV camera or the vidicon tube may be displaced in such a way that the body appears at an angle. Accordingly, the search positions $(-x_i, y_i)$ and (x_i, y_i) must be mapped into correct positions (XL,YL) and (XR,YR). The straight-forward calculations needed for this mapping are given in Figure 6-10.

Now, at each end of the four widths to be measured an edge detection operator is used to find the vertical edge.

Digitized by Google

Digitized by Google

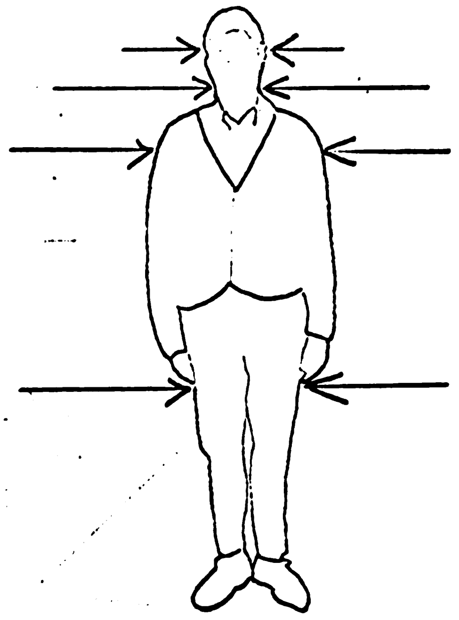


Figure 6-8. Measurements of width.

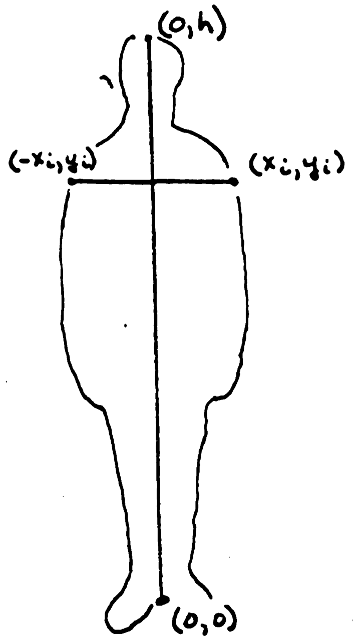


Figure 6-9. Schematic diagram of placement of width measurements.

Digitized by Google

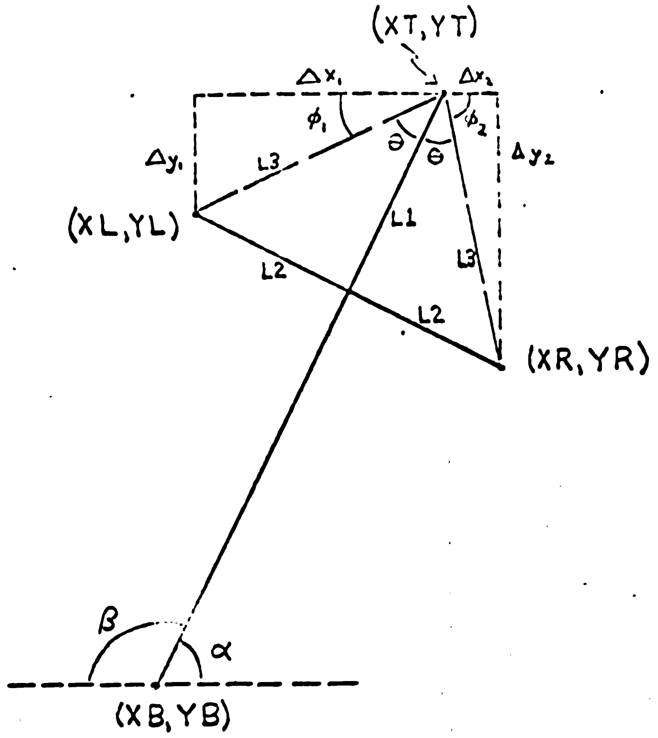
Digitized by Google

TABLE 6-1.
Standard positions for width measurements.

i	w = x/h	r = y/h	operator height
1	.11	.93	.028
2	.067	.87	.017
3	.28	.76	.028
4	.21	.38	.028

Digitized by Google

Digitized by Google



$$\beta = \arctan\left((YB-YT)/(XB-XT) \right)$$

$$\alpha = \pi - \beta$$

$$\theta = \arctan\left(L2/L1 \right)$$

$$L3 = \sqrt{L2^2 + L1^2}$$

$$\phi_1 = \alpha - \theta \qquad \phi_2 = \beta - \partial$$

$$\Delta x_1 = -L3\cos\phi_1 \qquad \Delta x_2 = L3\cos\phi_2$$

$$\Delta y_1 = L3\sin\phi_1 \qquad \Delta y_2 = L3\sin\phi_2$$

$$XL = XT + \Delta x_1 \qquad XR = XT + \Delta x_2$$

$$YL = YT + \Delta y_1 \qquad YR = YT + \Delta y_2$$

Figure 6-10
Allowance for the variance from vertical.

Digitized by Google

Digitized by Google

The height of the operator is given in Table 6-1 as a fraction of the height of the subject. When these edges are found they provide the width in raster units of each feature.

The measurements are now recorded for use in the classification step. This concludes the processing of the picture of the entire body.

Digitized by Google

Digitized by Google

CHAPTER 7

FINDING THE OUTLINE OF THE HEAD

This chapter describes the program which has been developed for extracting an accurate outline of a man's head from a digital picture. A typical input picture is shown in Figure 7-1. When the methods described in this chapter are applied to the input represented by Figure 7-1, the result is the outline of the man's head as shown in Figure 7-2.

In the preceding chapter the methods used to extract measurements from the picture of the body were described. The resolution of that picture is not adequate for measuring the position of facial features. Therefore a second picture is taken, a close-up of the head of the subject. This picture, like that taken of the body, contains the head of the subject in front view, looking at the TV camera, against a background of normal room objects.

It is necessary to find an accurate outline of the head so that dependable reference points can be found from which the width of the head can be measured. The location of the top of the head is also obtained from the outline. The position of the outline of the head is used in subsequent processing to determine where to search for smaller features such as eyes.

The details of this processing are described in the remainder of this chapter. The picture presented in Figure

Digitized by Google

Digitized by Google

Figure 7-1.
Unprocessed input picture of a man's head.

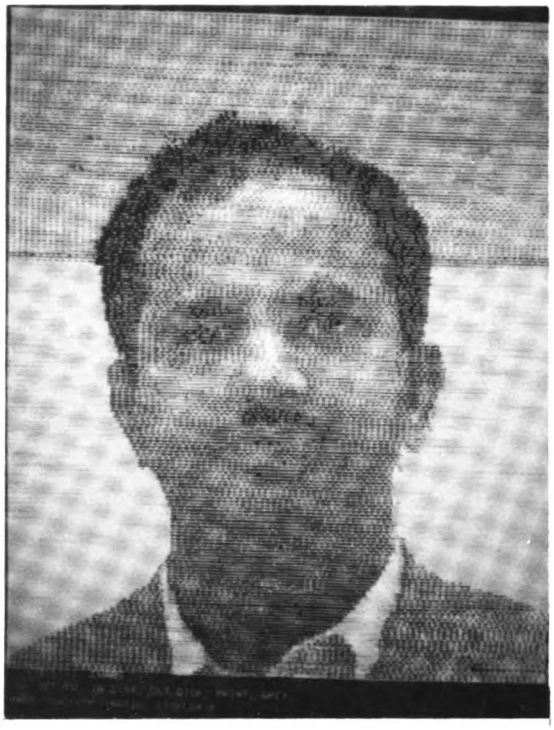
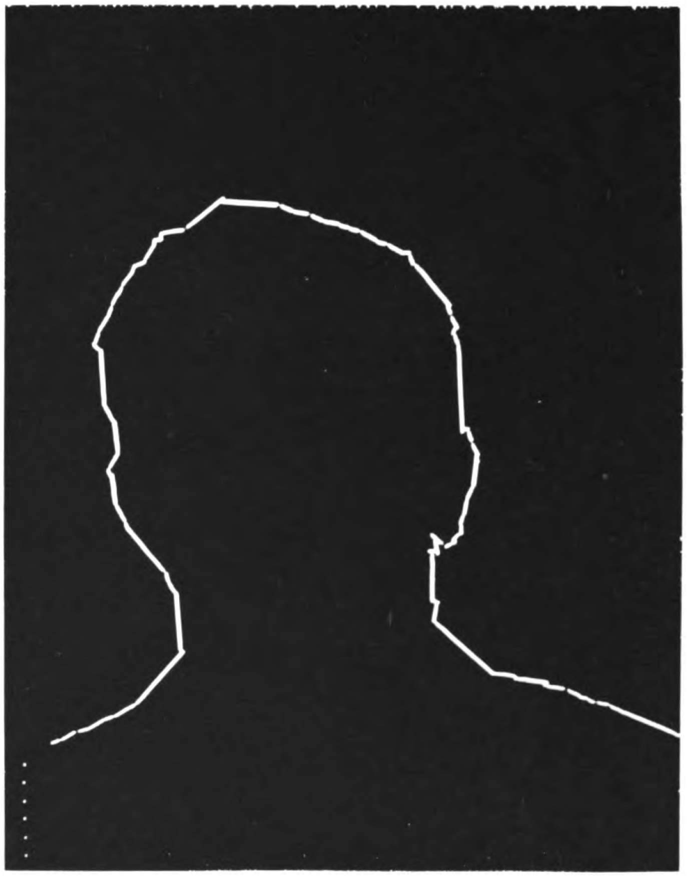


Figure 7-2.
The outline of a head, the result of processing Figure 7-1



Digitized by Google

Digitized by Google

7-3 will be used as an example throughout the chapter. The size of this picture is 226 x 325 points. In general the size of the pictures of the head is variable. Since the programs which search for and find heads is complex and only described in general terms in this chapter, a complete listing of these programs is included as Appendix 3.

REDUCE IN SIZE.

A new small picture is produced from the original. Each point in the small picture is the average value of the 64 points in an 8 x 8 square in the original picture. Figure 7-5 is an example of a small picture. Only the shape of the head is clearly visible. Other details have been smoothed out. The size of Figure 7-5 is 28 x 40 points. The decision to reduce in size by the factor eight was somewhat arbitrary. In smaller pictures the head tended to disappear; in larger pictures some unwanted details were still present.

FIND ALL EDGES.

The small picture obtained in the preceding step is processed to produce a new matrix which contains information on the edges in the small picture. This matrix will be called the <u>small edge matrix</u>. Figure 7-6 is an example of the small edge matrix derived from Figure 7-5. Each element in the matrix contains four bits, representing the four

Digitized by Google

Digitized by Google

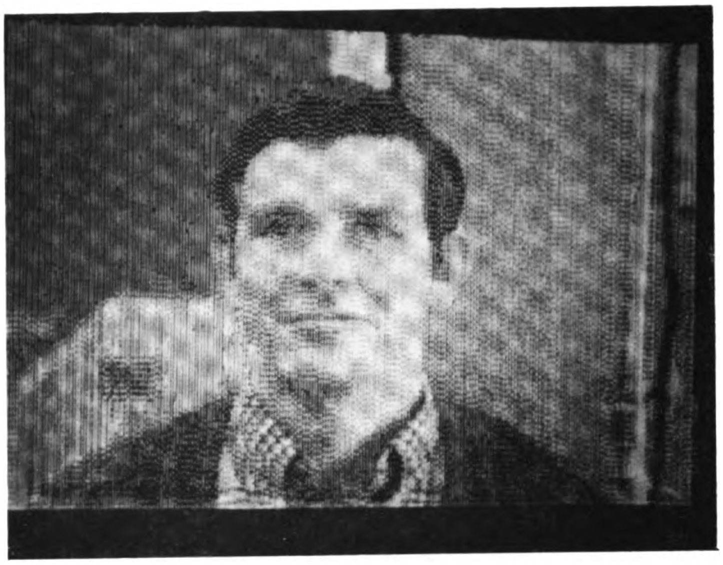


Figure 7-3. Unprocessed input picture used to obtain Figures 7-4 through 7-8

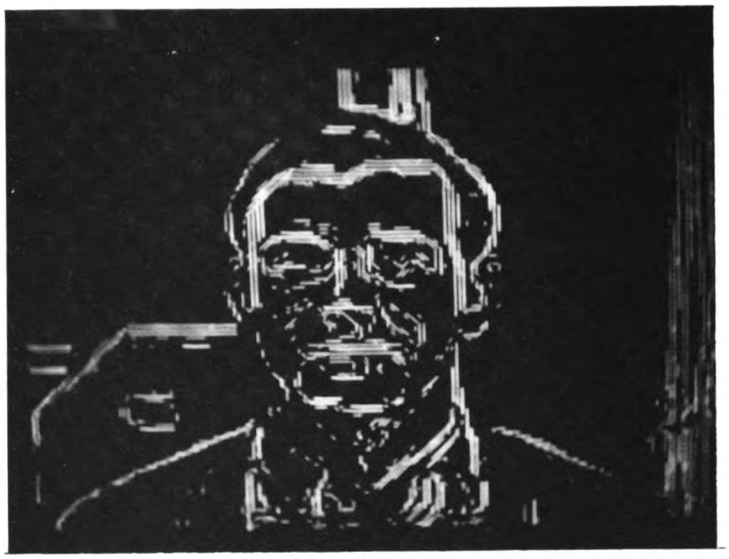


Figure 7-4. The results of applying a gradient operator to Figure 7-3

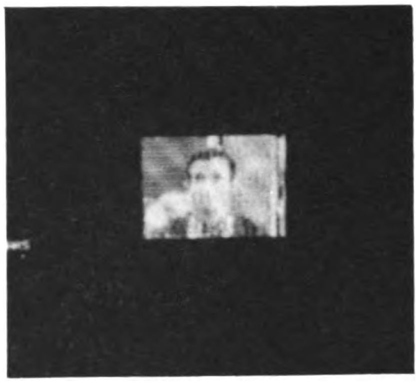


Figure 7-5. The results of reducing Figure 7-3 in size by averaging.

Digitized by Google

Digitized by Google

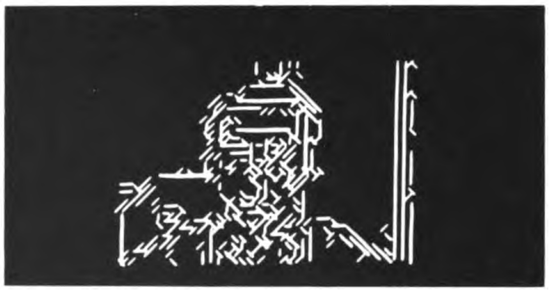


Figure 7-6
The results of applying and edge-finding operator to Figure 7-5.

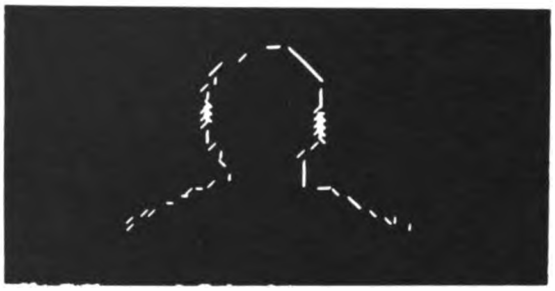


Figure 7-7.
The "plan" extracted from Figure 7-6 which will be used to follow the head outline.

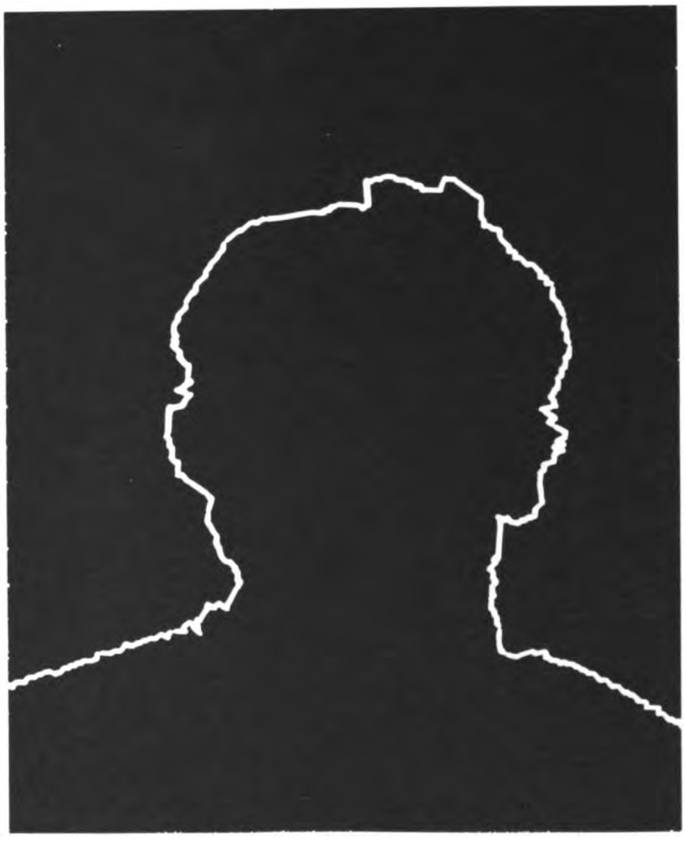


Figure 7-8.
The outline of the head in Figure 7-3, obtained using the plan of Figure 7-7.

Digitized by Google

Digitized by Google

directions (horizontal, vertical, diagonal right, diagonal left) in which edges may be detected. All bits are zero if no edge was found at a point. A one in any bit indicates that an edge in that direction was found.

The local operator which is used to detect edges is described below. Many operators were candidates for this purpose, including those mentioned earlier in the section on edge detection. The choice of the particular operator used was made because of high confidence in its ability to reject false edges. Some edges may be missed but this is preferable to reporting false edges. A more straight-forward gradient operator was not used because they sometimes produce an edge indication in an uneven transition area of grey shades. In addition there is no natural cut-off threshold for the gradient operator. Hueckel's operator [1969], which can give very good results, was rejected because it was too large to be effectively used in the small picture. Most of the smoothing utilized by Hueckel's operator has already been done during averaging.

Digitized by Google

Digitized by Google

The edge operator used is applied to 3 x 3 squares of the picture. Let the points in this square be labeled as follows:

```
A B C
D E F
G H I.
```

The algorithm for detecting a horizontal edge is:

```
if B>H then
        if [min(A,B,C)-max(G,H,I)] > 1 then EDGE
                                        else NO-EDGE
else if B<H then
        if [max(A,B,C)-min(G,H,I)] > 1 then EDGE
                                        else NO-EDGE
else NO-EDGE.
```

The EDGE or NO-edge results apply to point E. This algorithm is also applied in the vertical direction (using points A,D,G and C,F,I) and along both diagonals (points A,B,D and F,H,I for one diagonal; points B,C,F and D,G,H for the other). Although this algorithm may appear to be time consuming, it may be programmed efficiently.

The small edge matrix is searched as described in the next step to find the small head outline. Because of

Digitized by Google

Digitized by Google

repeated search and backup during the search for the head, it is best to perform the local edge identification operation only once for each point, Since the picture is small, this does not consume excessive time,

FIND HEAD IN REDUCED PICTURE,

The search for the outline of the head is done in the small edge matrix, Various heuristics are included in the search program which define an acceptable head shape, The final result of the search is a list structure containing the coordinates of the points which constitute the outline of the head, Each entry in the list structure designates a point where an edge is present in a direction which is reasonable for that part of the head, An example of an unreasonable edge direction would be a vertical edge among a row of horizontal edges which have been labeled the top of the head, Figure 7-7 shows the edges of the head found in this step, The representation of acceptable head shapes is incorporated into program statements, There are searches, branches, and possible back-up as the outline of the head is built up, This process is similar in some ways to the S.R.I. decision tree search (Nilsson [1969]), If no head is present in the original picture, it is detected at this stage,

The details of the search for the head are as follows, Three short line segments are found which are candidates for

Digitized by Google

Digitized by Google

being part of the top and sides of the head. The spatial relationship between these lines must be reasonable (e.g., the top of the head must be above the sides). An attempt is made to connect these line segments to form a "head" shape. The program searches the region between them for edges which are part of the somewhat semicircular top of the head. If this cannot be done, other possible short line segments are tried as sides or tops. If the top half of the head is found, a search is made below it for the inward curves of the neck and then the curves outward toward the shoulders. When all of these requirements have been met, the head has been found. The sides of the head are then examined for indentations where the ears should be since the ears sometimes merge with the background. If indentations exist, these are filled in.

USE PLAN TO FIND OUTLINE IN ORIGINAL.

This part of the program is a <u>plan follower</u>. Its input is the full size intensity picture and the list structure containing the small head outline. The output of the plan follower will be a new list structure containing the coordinates of an accurate outline of the head. Between successive points in the plan, the plan follower searches a narrow band for an edge which connects the points. This band is sixteen points wide, since the plan was reduced in size by a factor of eight. Although this band is narrow

Digitized by Google

Digitized by Google

compared to the size of the picture, it is still wide enough to contain several edges. The proper edge is chosen primarily by direction. If there are two parallel edges running in the desired direction within the search band, that edge farthest from the center of the head is chosen. The reason for this choice is the assumption that edges within the head (in hair, ears, etc.) are more likely to follow the directions of the head outline than edges found on background objects.

The operator used to detect edges in the full sized picture is almost the same as that used in the small picture. In the small picture the operator was applied to the nine points of a 3 x 3 square. In the large picture the same operator is applied to the nine points at the corners, the center of the sides, and the center of a 5 x 5 square, e.g.

```
A - B - C
- - - - -
D - E - F
- - - - -
G - H - I
```

The reason for this change was to allow detection of faint edges in the large picture.

Digitized by Google

Digitized by Google

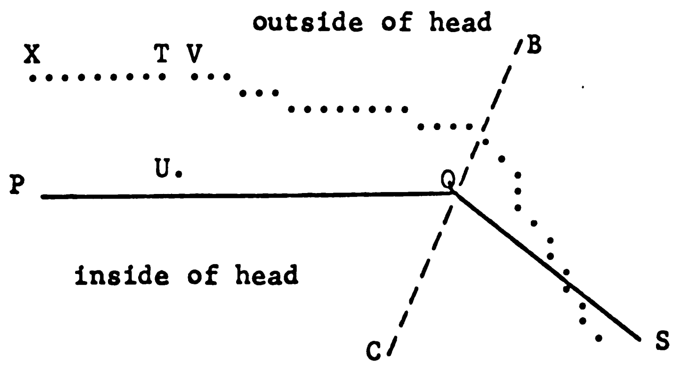


Figure 7-9. Operation of plan follower.

The details of the plan follower are given below. Reference is made to Figure 7-9 which is a schematic diagram of the operation of the plan follower in a small region. Two non-parallel straight lines are identified in the plan, lines PQ and QS. Point X is the last previous point that has been found on the edge. The direction of search is generally from P to Q to S. The line BC is found which bisects the angle PQS. Corresponding to the line PQ in the plan, the plan follower will search from point X until the edge crosses line BC. Within this region the edge will be accepted if its direction is roughly parallel to PQ or to QS. The edge search is done by moving one unit in the direction of PQ from the last point found and applying the

Digitized by Google

Digitized by Google

edge detection operator along a line perpendicular to PQ. Normally the edge detection operator is only applied to five points immediately in front of the last point found. Under certain conditions however the edge detection operator is applied over all 16 points across the search band. These conditions are:

a. The edge has just taken a sharp turn.

b. The edge is lost.

c. The intensity outside the head outline has changed abruptly.

d. The edge zig-zags.

These measures are necessary to correctly track the edge when it is sharply curving. The three points most recently found are filtered to remove a single point which is far off the edge. For example, if U were the only edge point found between points T and V, point U would be rejected.

SPEED COMPARISON.

The search for the outline of the head could have been done without using planning, as pointed out in Chapter 5. Tests were made to determine the speed improvement obtained by using planning. The results of these tests are given in Table 7-1.

These results indicate that using planning is about 40 times faster than the same search without planning. The tests were performed as follows. First the program just

Digitized by Google

Digitized by Google

TABLE 7-1.

Planning speed comparison.

PLANNING

STEP		TIME
1.	Reduce in size.	2.5 sec.
2.	Find all edges.	.8 sec.
3.	Find plan (small outline).	.3 sec.
4.	Use plan (large outline).	2.6 sec.
	Total	6.2 sec.

WITHOUT PLANNING

STEP		TIME
1.	Smooth picture.	160. sec.
2.	Find all edges.	34. sec.
3.	Find head outline.	40. sec.
	Total	234. sec.

Digitized by Google

Digitized by Google

described which finds the head outline using planning was timed. The time to process a head picture from input to completion of the large outline was about 6.1 seconds. This time, as well as all others mentioned, was measured on the PDP-10 computer and is accurate to ±20%. Then the program was modified to perform without planning.

The first step was to eliminate the fine detail of the picture by smoothing in a fashion similar to the averaging used in planning. The entire picture was smoothed using averages over 8 x 8 squares. This took 420 seconds. By optimizing the program, this time could be brought down to 64 times the speed without planning, namely 160 seconds. This latter time is used in Table 7-1.

All edges were found in the smoothed picture. This took 34 seconds.

Next the search program was used on the full sized smoothed picture to search for the outline of the head. Here is where planning really showed it speed. This search, in a reduced size picture, takes about 0.3 seconds. In the large picture, the search took about 40 seconds. The reason for this is that quite a sizeable incorrect data structure is built up when following false paths and many points must be processed before the error is detected.

In summary, the program without planning took 494 seconds, 80 times slower than the time using planning. By optimizing the program, this could be improved to about 234

Digitized by Google

Digitized by Google

seconds. This would still be 40 times slower than planning. This comparison cannot be complete without again stating the primary advantage that planning gives to the designer of a program which searches for edges or objects: a small picture that he can comprehend both locally and globally.

LOCATE MEASUREMENT POINTS.

Now that the outline of the head has been found it is necessary to locate the top of the head and the points on the sides from which the width of the head is measured. This is done using a sliding average to avoid single point errors which may be present in the outline.

The outline of the head is represented in the computer by a list structure. Each element of the list contains the two coordinates of a point in the outline as well as pointers to the list elements representing the adjacent points on both sides. Thus it is possible to traverse this list in either direction. There are also a number of pointers external to the list which mark approximate locations of key elements.

These key element pointers have been obtained in the following way. Originally a plan for the head was formed, as already described. This plan was created using the known structure of the head. So in creating the plan certain areas were identified as top of head, extremes of approximately vertical side of head, etc. This information

Digitized by Google

Digitized by Google

is carried forward to the plan following stage. When the plan follower passes one of these areas while creating the list structure for the true head outline it sets an appropriate pointer into the list structure.

Knowing the approximate location for the top of the head, it is straight-forward to search for the highest point in the outline in this area. A sliding average of five points is used. The highest value gives the coordinates of the top of the head.

In a similar way the widest part of the head is identified on both sides of the head. The difference between these values is recorded as the width of the head. The coordinates of the points which have been identified on the outline of the head are now used to guide the search for the features interior to the head.

Digitized by Google

Digitized by Google

CHAPTER 8

FINDING THE FEATURES OF THE FACE.

The previous chapter described how the outline of the head was located. Once this outline is found, approximate locations for the eyes, the nose, and the mouth can be predicted. In spite of the fact that the approximate location of these features is known, there is still considerable processing that must be done to locate them precisely so that measurements of position may be made. This chapter discusses the methods used for finding each feature.

It is appropriate here to describe the coordinate system used in processing pictures in this program, since much of the discussion which follows makes use of this coordinate system. A left-handed coordinate system is used with the origin at the upper left of the pictures. (See Figure 8-1.) There are two reasons for this. First, this is the method used by the system when setting the limits on a window to be read by the TV camera. Secondly, it provides a convenient notation for interchanging coordinates (x,y) of a point with matrix indices (I,J) for that point.

EYES.

The measurement that is desired from each eye is the location of the center of the iris, the dark circle in the

Digitized by Google

Digitized by Google

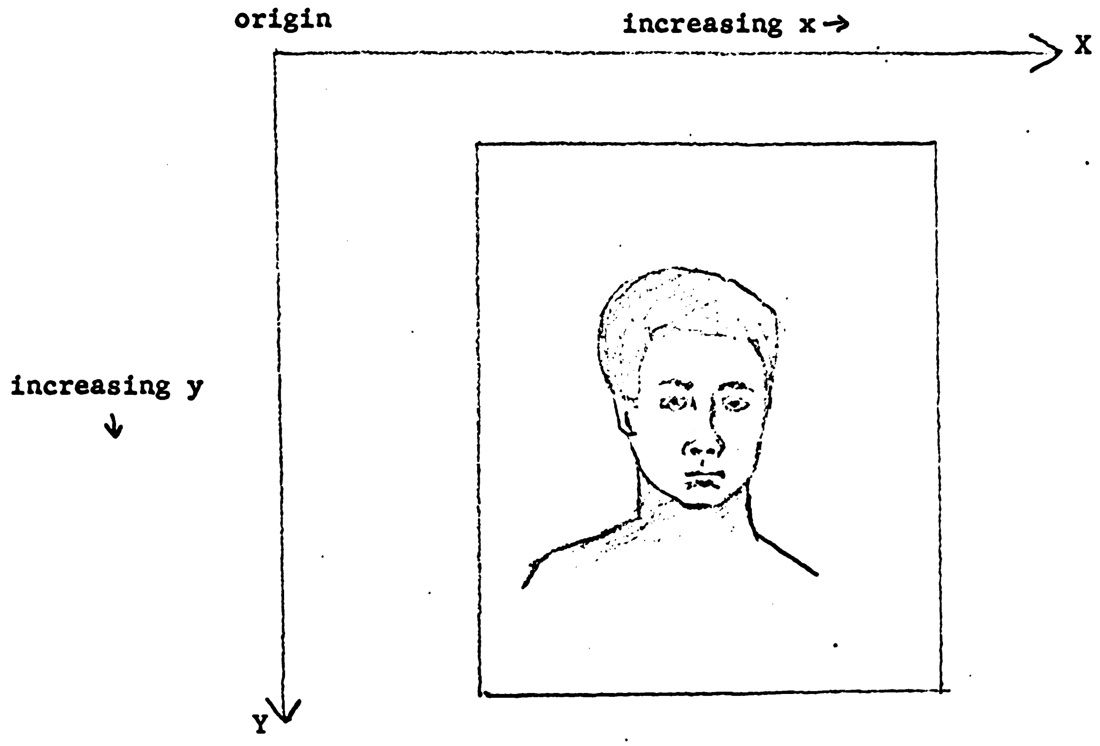


Figure 8-1.

Left-hand coordinate system used.

Digitized by Google

Digitized by Google

center of the eye. The iris is located by finding horizontal cross-sections which exhibit the characteristic shape shown in Figure 8-2. In essence, this is a template matching operation. The shape of Figure 8-2, when plotted as light intensity vs. x coordinate, possesses the following features. Outside the eye the skin has an irregular intensity of medium value. The white of the eye on both sides of the iris forms high peaks of light intensity. The iris itself is a dark valley between these two peaks.

The characteristic cross-section of the eye is elusive and difficult to find. The peaks and valley that are sought can be nothing more than anthills and a depression. To find them, one must know where to look. Pinning down just where to look for the iris and the whites of the eyes constitutes the bulk of the processing used on the eyes. The method used is called dynamic threshold setting.

Approximately locating the eye is also difficult. The detail in this area, as well as shadows around the eyes and nearby hair, can confuse the program. The one thing that can be found is a characteristic dark blob formed by the dark iris, eyebrow, and shadows under the overhanging brow. However, there is no natural threshold which defines the dark region. What is meant by dynamic threshold setting is the process of searching for a threshold that clearly shows this dark region.

The first thing that is done is to predict the location

Digitized by Google

Digitized by Google

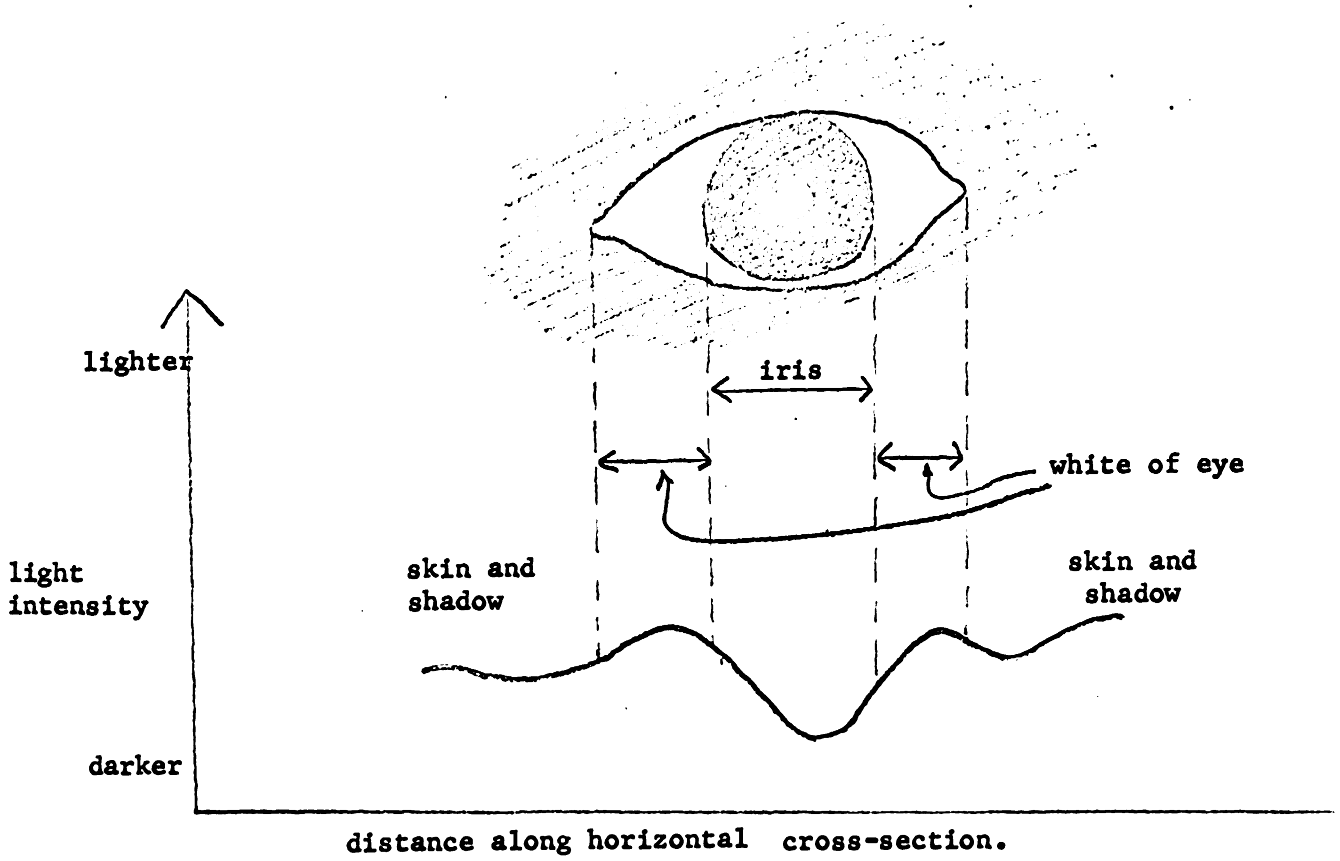


Figure 8-2. Horizontal cross-section which characterizes the eye.

Digitized by Google

Digitized by Google

of the eyes. This prediction is based on the average position of the eyes within the head. The predicted eye location will be used as the center of the area to be searched for the eye.

Figure 8-3 shows the model of the head and eyes on which the prediction is based. According to this model, if the top of the head is on the line y=0 and the sides are on the lines x=0 and x=W (for width), then the eyes are located at

(FX(1)*W, FY(1)*HWRAT*W) and

(FX(2)*W, FY(1)*HWRAT*W).

The experimentally obtained values of the parameters of this model are given in Figure 8-4.

Figure 8-5 shows how this model is used to predict eye location in an actual picture. From the previous processing it is known that the left and right sides of the head are at X=XL and X=XR. The width of the head is therefore W = XR - XL + 1. Using the model, the x coordinates of the two eyes should be

X = FX(1)*W + XL for left eye

and X = FX(2)*W + XL for right eye.

The top of the head is known to be at Y=YT. So from the model, the y coordinate of the eyes can be predicted as

Y = FY(1)*HWRAT*W + YT.

The predicted eye location is only a rough first approximation to the actual location of the eyes. This

Digitized by Google

Digitized by Google

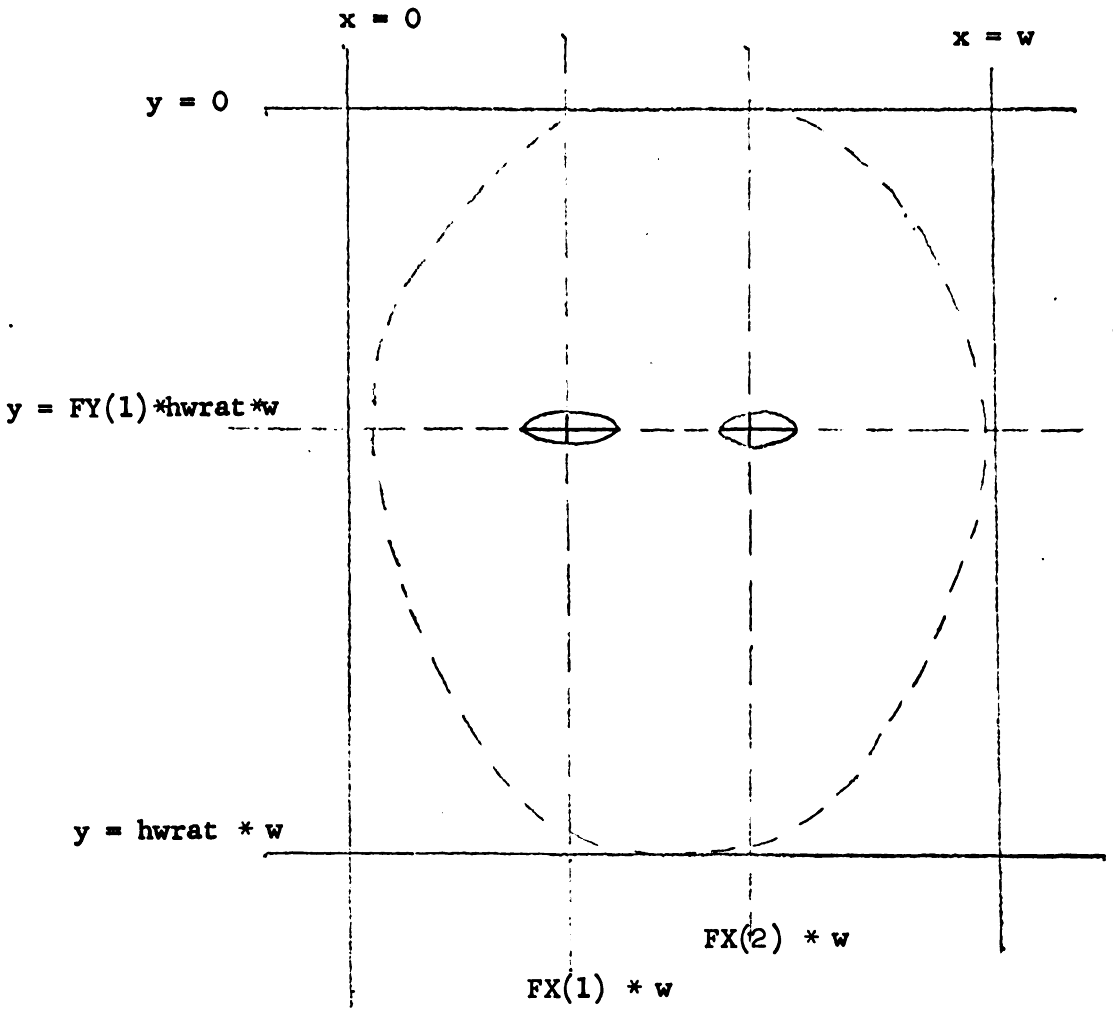


Figure 8-3. The model for eye location on an average head.

FX(1)	.329
FX(2)	.671
FY(1)	.494
HWRAT	1.2

Figure 8-4. Constants used in predicting the location of the eyes.

Digitized by Google

Digitized by Google

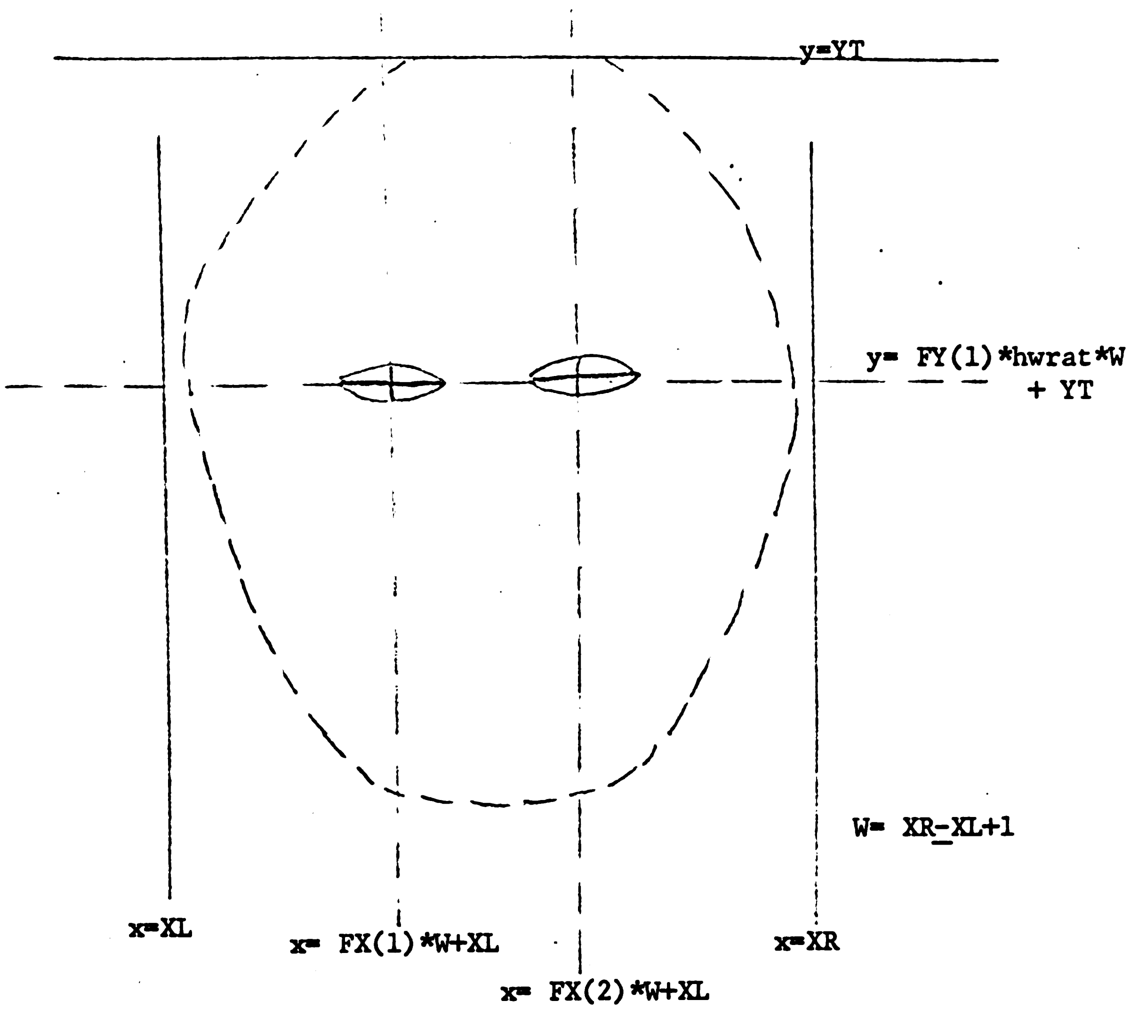


Figure 8-5. Predicting the location of the eyes on a given head.

Digitized by Google

Digitized by Google

approximation must be refined considerably before the search for the iris can be expected to work.

The area around the eye is found by looking for a dark blob. The iris, the lashes, the eyebrows, and especially the shadows under the overhanging eyebrow form an area that is darker than the surrounding skin. When this dark area is located, it will be possible to look within it for the iris. Locating this dark area is the next step in eye processing.

A search procedure that examines first a large area, then a small area, is used to home in on the dark blob that is the eye. The reasons for this two part search are given in the discussion of step A4 which follows. The flow chart of this search procedure is given in Figure 8-6. The steps of this procedure are discussed in detail in the following paragraphs.

A1. A square region to be searched for the eye is selected. This square is centered on the predicted eye location. It is large enough so that it will quite surely contain the eye even though the eye may be fairly far away from the predicted eye location. A suitable size for this square was determined experimentally. The size used is based on the measured width of the head W and is .383 W.

A2. Within this square, points which are dark and within large, centrally located, connected regions are identified by dynamic threshold setting. The procedure that identifies these points is fairly complex. It is used

Digitized by Google

Digitized by Google

```
     ------------------------------------------------
A1,  | Center large square on predicted eye location |
     ------------------------------------------------
                          |
                          v
          ----------------------------------------
A2,       |  Identify centrally located dark area |
          |         within large square,          |
          ----------------------------------------
                          |
                          v
          ----------------------------------------
A3,       |  Find center of gravity of dark area, |
          ----------------------------------------
                          |
                          v
        ---------------------------------------------
A4,     |  Center small square on center of mass     |
        |     of previously found dark area,         |
        ---------------------------------------------
                          |
                          v
        ---------------------------------------------
A5,     |   Identify centrally located dark area     |
        |          within small square,              |
        ---------------------------------------------
                          |
                          v
    ----------------------------------------------------
A6, | Characterize dark area by its center of gravity |
    |                 and mean radius,                 |
    ----------------------------------------------------
```

Figure 8-6, Procedure for finding
the dark area around the eye,

Digitized by Google

at steps A2 and A5. It will be described in detail following the completion of the discussion of the flow chart given in Figure 8-6. In general, however, the procedure looks for a dark area that surrounds the eye. What is meant by "dark"? It is not an absolute measure. Rather, when examining the square under consideration, some points are darker; some are lighter. Some fraction of the points must be heuristically defined as "dark". This fraction is a parameter that was determined experimentally. The value used is .107. The procedure omits dark points which are isolated; it assumes these represent noise. It also omits all dark points which are part of connected regions which intersect the border of the square. These are assumed to be hair, the other eye, or shadows on the nose. Because the search square is large, these other features are sometimes present.

A3. The dark points found in the preceding step should be the points that make up the dark area around the eye. The center of this area is defined to be the centroid of these points. Let S be the set of "dark" points found. Let n_S be the number of points in S. For $\alpha \in S$ let the coordinates of α be x_α, y_α. Then the centroid $(\bar{x}_1, \bar{y}_1)$ is computed from

$$\bar{x}_1 = \frac{\sum_{\alpha \in S} x_\alpha}{n_s}$$

$$\bar{y}_1 = \frac{\sum_{\alpha \in S} y_\alpha}{n_s} .$$

Digitized by Google

Digitized by Google

A4. It might be expected that the dark area found above is an acceptable approximation to the dark area around the eye. This is not the case. The extent of the area found is strongly dependent on the fraction used to define "dark" points. Since the square used must be quite large so that the eye will not be missed, the uncertainty in the value of this fraction is large. Another iteration of the dynamic threshold setting procedure is used. This time a smaller square will be searched so that the confidence in the value of the fraction is higher. The square will be centered on the centroid of the dark area which was found previously. The size of the square used is again based on a proportion of the width W of the head. The side of the square is .287 W.

A5. The centrally located dark area within the small square is identified. The procedure is the same as that used in step A2. The value of the fraction used to define dark points is .06.

A6. The dark points found in step A5 are characterized by their centroid as was done in step A3. The new centroid is denoted by $\overline{x}_2, \overline{y}_2$. An additional parameter is measured which characterizes the dark region. This is the "mean radius", $\overline{r}$, of the dark region. Let S, n_s, α, x_α, and y_α be defined as in A3. Then the mean radius is given by

$$\overline{r} = \frac{\sum_{\alpha \in S} [(x_\alpha - \overline{x}_2)^2 + (y_\alpha - \overline{y}_2)^2]^{1/2}}{n_s}.$$

Digitized by Google

Digitized by Google

This completes the discussion of the general flow given in Figure 8-6. The dark area around the eye has been found and characterized by its centroid and its average radius. A picture of this characterization is given in Figure 8-8.

Before going on to discuss locating the iris, the details of the dynamic threshold setting procedure used in steps A2 and A5 will be given. A generalized flow chart of this procedure is given in Figure 8-7.

The procedure attempts to find a dark area that is central to a square subset of the picture. The dimensions of the square are input parameters to the procedure. Another input parameter is NDARK, the number of dark points expected. The values of these parameters were specified in the section which described the calls on these procedures.

An inspection of Figure 8-7 shows that there are several loops within this procedure. The looping may not be necessary in pictures where the eye region is clearly distinct from its surroundings. First, the flow of the procedure will be discussed assuming a straight, non-looping flow through the program. Following this, the reasons for allowing iteration and the loops will be discussed.

Let A_{ij} represent the grey-valued picture matrix which includes only the points in the square under consideration. The first step is to form a histogram of the intensities of S. The histogram is formed in the array h[Ø:15]. The value of the elements of h are

Digitized by Google

Digitized by Google

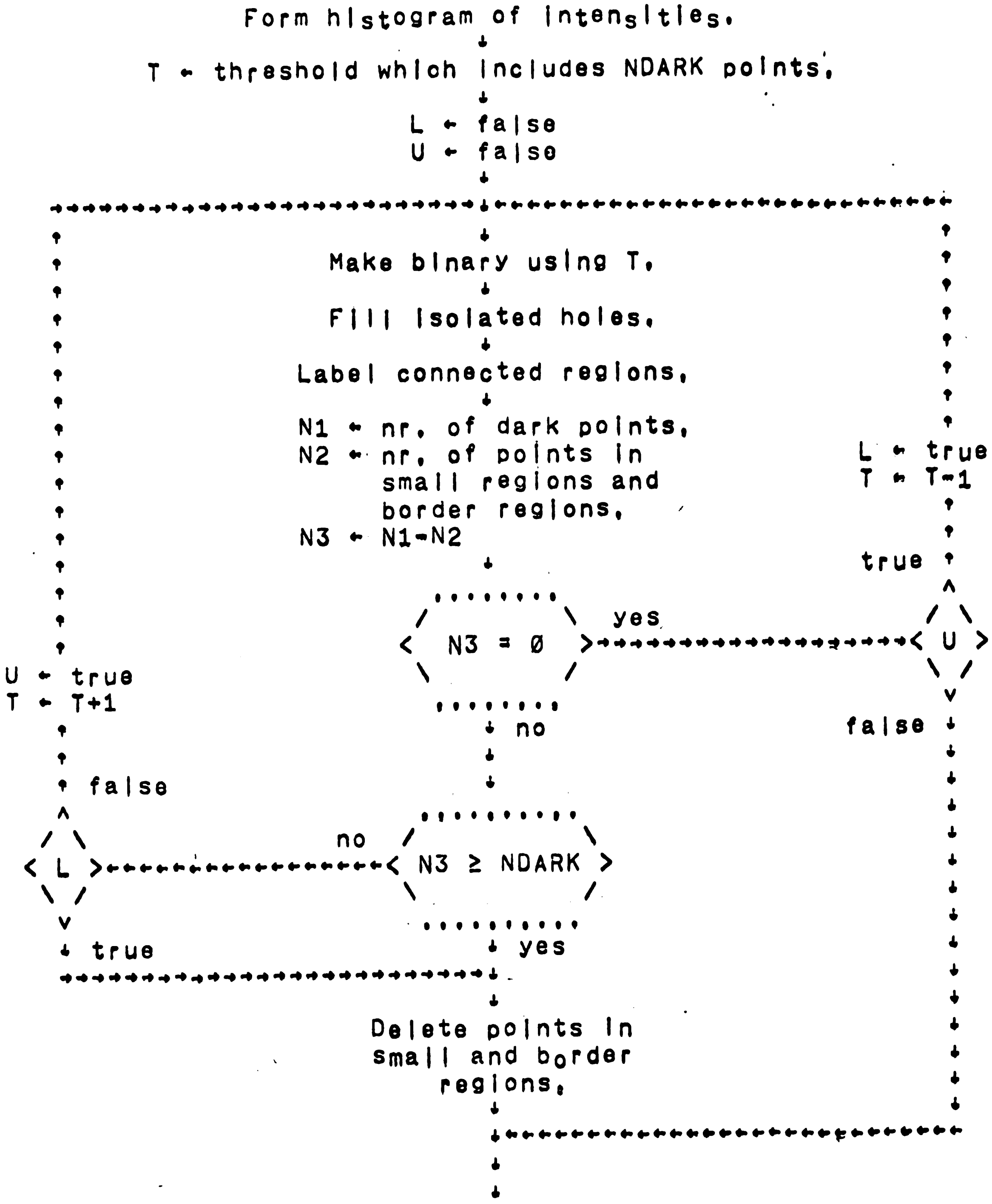


Figure 8-7. Dynamic threshold setting. (Procedure for finding central dark areas.)

Digitized by Google

Digitized by Google

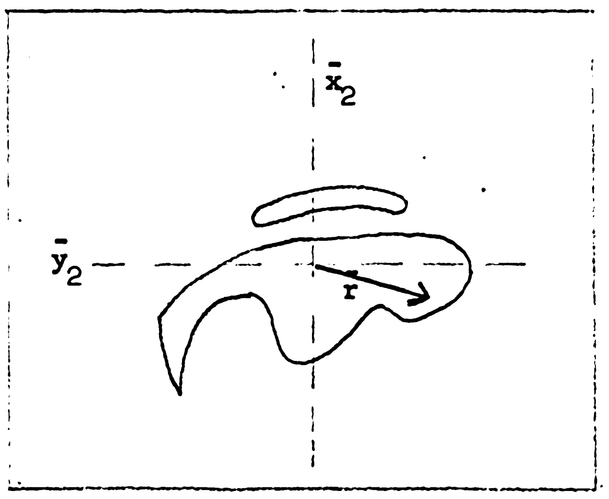


Figure 8-8. The dark region around the eye and characterizing parameters.

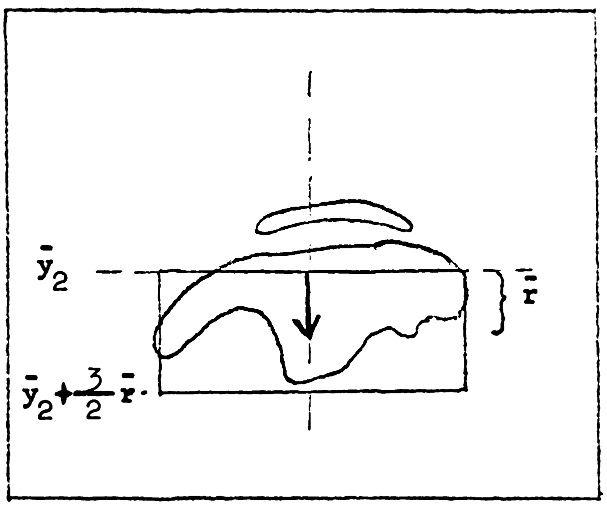


Figure 8-9. The area searched for the iris.

Digitized by Google

Digitized by Google

$$h_k = \sum_{ij} [\text{if } A_{ij} = k \text{ then } 1 \text{ else } \emptyset].$$

The variable T is set to that threshold that includes NDARK points. More precisely, T is the minimum value such that

$$\sum_{k=0}^{T} h \geq \text{NDARK}.$$

It is clear that T divides the picture A into two sets of points: the dark points with $A_{ij} \leq T$ and the lighter points with $A_{ij} > T$.

Now two Boolean variables, L and U, are set to "false". These are only used if looping is necessary. This completes the initialization of the procedure.

The main body of the procedure begins with construction of a binary-valued picture B from A. The threshold T is used in this construction as follows:

$$\begin{aligned} B_{ij} &= 1 \text{ if } A_{ij} \leq T \\ &= \emptyset \text{ if } A_{ij} > T. \end{aligned}$$

Thus the 1's in B represent the dark points of A.

The picture B is smoothed somewhat by filling in isolated holes. Points with value Ø are set to 1 if they are surrounded with 1's.

Connected regions of 1's in B are identified and labeled. This is done using the procedure "CONNECTED" which was discussed in detail in Chapter 6 (see Figure 6-4). Since each dark point of B is labeled with the region number of the connected region to which it belongs, it is easy to count the number of points in each connected dark region and

Digitized by Google

Digitized by Google

to identify those dark regions which intersect the border.

N1 is set equal to the total number of dark points in B. N2 is set equal to the number of points in B that are in small regions (< 4 points) or in regions that intersect the border. N3 is set to N1-N2. N3, then, is the number of points in connected dark regions that are reasonably large and somewhat central.

N3 is tested to ensure that it is greater than NDARK. If it is, all is fine. Those points previously identified as being members of small regions or border regions are deleted from the binary picture B. When a point is deleted, its value is set to zero.

The procedure has completed its work. The binary picture B is available with 1's only in points that are central and dark. The centroid and mean radius can be computed using B.

Now, we must return to consider the cases where iteration is necessary in the procedure. This is necessary if fewer than NDARK points would be left after deleting small regions and border regions. Recall that N3 is the variable that is a measure of these points. There are four cases to consider.

Case 1. Ø < N3 < NDARK and L is false. In this case the assumption is made that the threshold T was not set high enough. It was desired that NDARK points should be present in the eye area. When much dark hair from the forehead or

Digitized by Google

temples enters the square under consideration, the threshold T, which was obtained from the histogram during the initialization of the procedure, is unreliable. The dark points from the hair are present in the histogram but then are deleted as border points. The solution is to raise the threshold T. Although more hair may be included it will be ignored since it always extends to the boundary of the area under consideration. T is increased by 1. The Boolean variable U is set to true to indicate that the threshold has been moved up. A transfer is made to the beginning of the body of the procedure.

Case 2. N3 = Ø and U is false. If U is false, this is the first iteration of this procedure. At the first threshold setting, all points appear to be border points or in small regions. When this condition is reached, it indicates that the eye region possesses a strange connectivity in the picture under consideration. All dark points are kept in the matrix B and the program is allowed to proceed onward to the next stage of processing. Hopefully, the tolerance of the remaining routines will be able to handle this strange picture. (This case is included in the program only for completeness. It has never been encountered when processing normal pictures.)

Case 3. N3 = Ø and U is true. In this case we have already encountered case 1 and have increased the threshold. In doing so the entire set of dark points has become

Digitized by Google

Digitized by Google

connected to the borders. This undesirable situation is remedied by lowering the threshold one notch and using those results as the best available. T is decreased by one. L is set to true. The program transfers to the beginning of the body of the procedure. Case 4 will occur on the next iteration.

Case 4. $\emptyset < N3 < NDARK$ and L is true. The previous iteration encountered Case 3. T was raised too high. Accept the value of N3. Delete the small regions and border regions. The resulting matrix B is suitable for further processing.

This completes the description of the dynamic threshold setting procedure. We will now consider the routines which detect the iris.

Figure 8-8 shows the dark region around the eye and the parameters, $\bar{x}_2, \bar{y}_2$, and $\bar{r}$, which characterize the region. Now it is necessary to find the iris. In particular, we wish to find the characteristic horizontal cross-section that was shown in Figure 8-2. An effective procedure for doing this is the following.

First the area where the iris is expected to be found is heavily smoothed. This area is bounded above and below by the lines $y = \bar{y}_2$ and $y = \bar{y}_2 + \frac{3}{2}\bar{r}$ and on the ends by the maximum extent of the dark region. This area is illustrated in Figure 8-9. Within this area smoothing is done by replacing the intensity at each point with the average

Digitized by Google

intensity over a pxq rectangle centered at the point. The height of the rectangle, p, is equal to the mean radius $\bar{r}$. The width, q, is $\frac{1}{3}\bar{r}$. A typical value for $\bar{r}$ is 7. Thus a typical size for the smoothing rectangle is 7x2.

Now, a horizontal cross-section is taken along each line in the rectangle in Figure 8-9. The characteristic shape (two mountains of eye white with a valley between for the iris) can be identified. The center of the iris is located, and its coordinates are recorded.

The positions of the eyes have been located. Therefore, processing of the eyes is complete.

NOSE.

The measurements that are obtained from the nose are the coordinates of the center of the two nostrils. In a front view of the face the nostrils are the only part of the nose that is reliably and consistantly present.

The nostrils appear in the digital pictures as two small dark areas of roughly elliptical shape. A typical nose region may be seen in Figure 8-10.

The processing needed to locate the nostrils is similar in concept to that used in processing the eyes. First, a location for the nostrils is predicted, based on a model of the relationship of the eyes and nose. Then the area surrounding the predicted location is searched to determine the actual location.

Digitized by Google

Digitized by Google

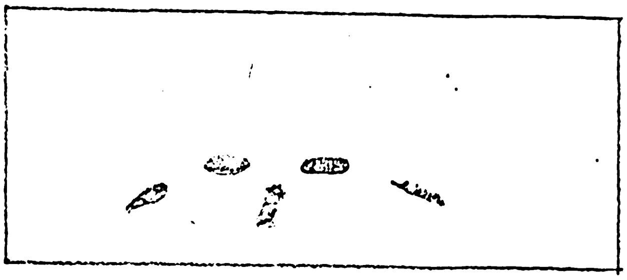


Figure 8-10.
Nose region in
unprocessed picture.

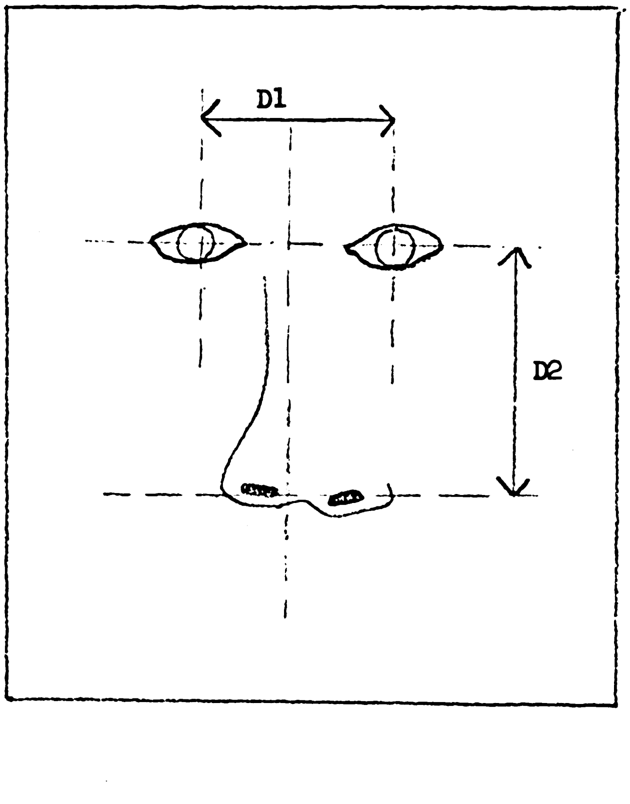


Figure 8-11.
Essentials of the
model of the eye-nose
relationship.

Digitized by Google

Digitized by Google

Figure 8-11 shows the essentials of the model that relates the positions of the eyes and the nose. The distance between the eyes is denoted by D1. The distance between parallel horizontal lines intersecting the eyes and intersecting the nostrils is denoted by D2. In the model D2/D1 is assumed to be a constant ratio. The experimental value used is .75. The model is used as follows. Let XL,YL and XR,YR be the previously measured eye locations. Then the predicted nostril locations XP,YP are given by

$$XP = \frac{1}{2}(XL + XR)$$
$$YP = \frac{1}{2}(YL + YR) + \left(\frac{D2}{D1}\right)(XR - XL).$$

These coordinates give a first approximation to the location of the nostrils. A square centered on XP,YP will be searched for the nose. The sides of the square are .287 times the measured width of the head W.

A key point in locating the nose, as in finding almost every other feature, is preliminary smoothing. This tends to make prominent things prominent and to make the minor local variations have a smaller effect. The smoothing is done on the nose region by replacing the average intensity at each point in the square under consideration by the average intensity of the four points in a 2x2 square.

At the same time that this smoothing takes place, a histogram of intensity is made. This histogram is used for subsequent threshold setting.

The nostrils are located by a procedure which does

Digitized by Google

Digitized by Google

dynamic threshold setting. The broad outlines of the search for the nostrils is given by the flow chart in Figure 8-12. This flow chart is discussed in detail below. The general idea is as follows. A threshold T is gradually raised from lower to higher grey values. At the lowest value only the few darkest points are below the threshold. As the threshold is raised a little, several small dark areas will be below the threshold. These areas should include the two nostrils. There may also be dark areas for shadows under the nose or from the sides of the nose. A heuristic decision procedure is used to identify those dark areas which represent the nostrils.

The details of the procedure of Figure 8-12 are as follows.

C1. The first value of the threshold T is chosen as the minimum intensity in the nose region. This value is obtained from the previously prepared histogram.

C2. Dark regions are identified. A dark point is a point with intensity less than or equal to T. A binary valued picture is formed with dark points having value 1, and light points having value Ø. This is similar to the procedure used in locating the dark eye area. The procedure for identifying connected regions (CONNECTED, Figure 6-4) is applied to the binary valued picture; its methods were discussed previously in eye and body processing. Each dark point is associated with the region number of the connected

Digitized by Google

Digitized by Google

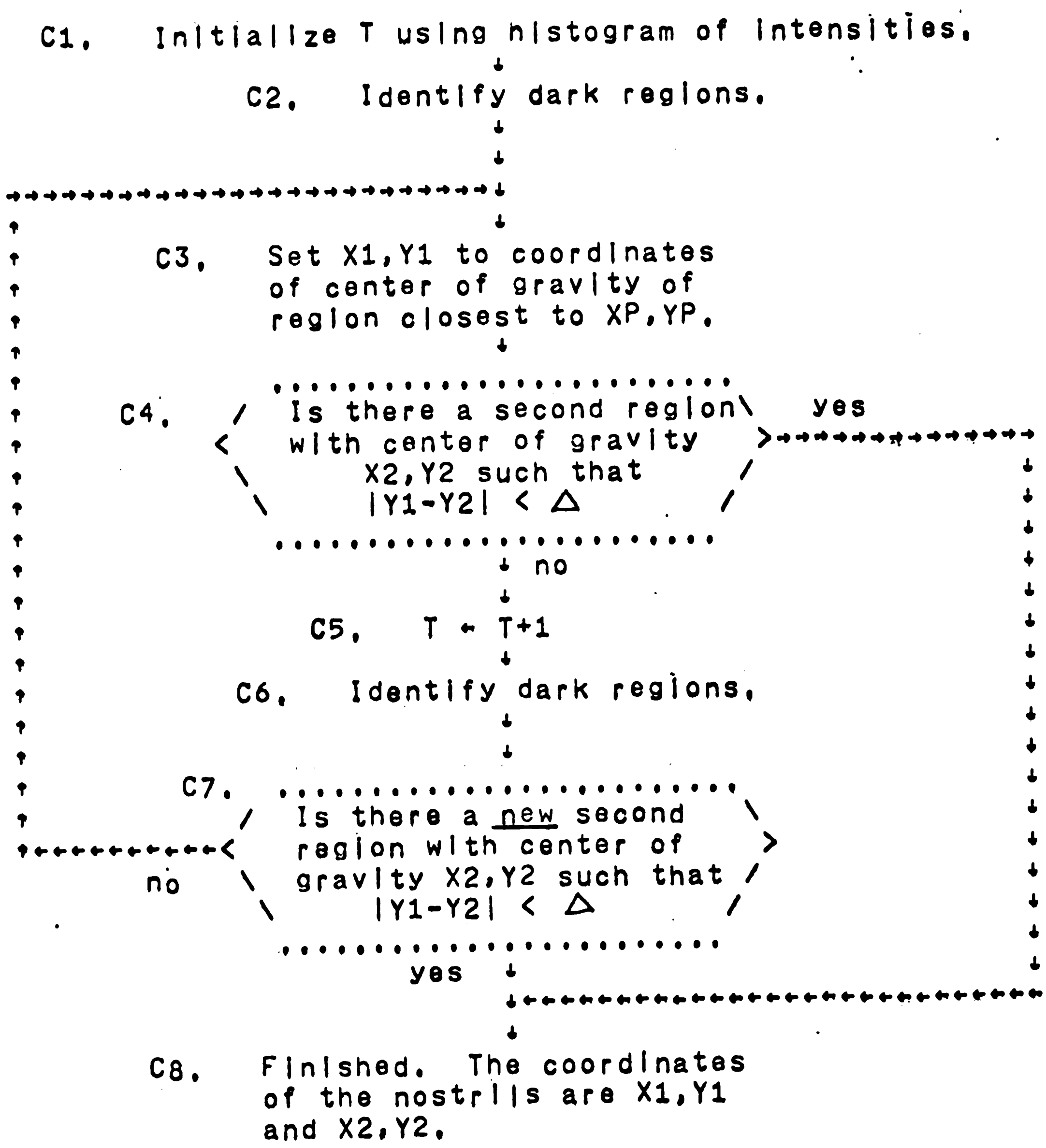

C1. Initialize T using histogram of intensities.
C2. Identify dark regions.
C3. Set X1,Y1 to coordinates of center of gravity of region closest to XP,YP.
C4. Is there a second region with center of gravity X2,Y2 such that |Y1-Y2| < Δ
yes
no
C5. T ← T+1
C6. Identify dark regions.
C7. Is there a new second region with center of gravity X2,Y2 such that |Y1-Y2| < Δ
no
yes
C8. Finished. The coordinates of the nostrils are X1,Y1 and X2,Y2.

Figure 8-12. Locating the position of the nostrils.

Digitized by Google

region to which it belongs. From this information the centroid of each dark, connected region is computed.

C3. XP,YP are the coordinates of the point predicted to be between the nostrils. The closest region to XP,YP is found. Here the distance from a point to a region is defined as the Euclidian distance from the point to the centroid of the region. X1 and Y1 are set equal to the centroid of the closest region.

C4. A check is made to see if there is a second region in the picture at approximately the same height as X1,Y1. In particular, what is sought is a centroid X2,Y2 such that

$$|Y1 - Y2| < \lambda 1 \quad \text{and} \quad |X1 - X2| > \lambda 2.$$

Satisfactory values for the parameters are $\lambda 1=4$ and $\lambda 2=8$. If such a region is found, the search for the nostrils is successful; the program transfers to C8. If such a region is not found, more regions are searched for at C5.

C5. The threshold T is increased by one. This will include more dark points in the next iteration of the procedure.

C6. Dark regions are identified in the area of the nose using the new value of T. This step is identical with step C2.

C7. Hopefully, new regions have been found in the preceding step. The coordinates X1,Y1 of the old closest

Digitized by Google

Digitized by Google

region (found in step C3) are still known. A check is made to see if a new region is present at approximately the same height as X1,Y1. The details of the test are the same as in step C4. If there is a new region which meets this criteria, the search for the nostrils is successful, and the program continues to step C8. If not, another iteration of the procedure is needed and a transfer is made to step C3.

C8. The procedure is finished. X1,Y1 and X2,Y2 represent the coordinates of the two nostrils.

This completes the processing of the nose. The locations of the nostrils are recorded for eventual use in the classification step.

MOUTH.

The mouth was a difficult feature to locate consistently. When work was first begun, it was anticipated that points of the mouth and lips could be found by edge detection operators. This did not work because of the great variability present in the mouth region. In a light intensity representation of the mouth and its surroundings, the things that stand out are the highlights and shadows caused by light falling on the recesses and protruding curves of the mouth. The lips and the mouth itself are much less prominent. These lights and shadows change greatly with various lighting arrangements.

Another problem in recognizing mouths is the

Digitized by Google

Digitized by Google

variability of position. There is no "standard" position that a subject can be instructed to assume with his mouth. Therefore, pictures of the same mouth look different from picture to picture.

The one characteristic of the mouth that is uniformly detectable is the dark thin horizontal line which the mouth forms. The program was written to detect the presence of this line and its location.

The principal component of this process is a line-finding operator. A great deal of experimentation was performed in an attempt to devise a good operator. The following procedure is the result.

The operator detects horizontal dark lines. It examines points in a 7x7 square of the picture. Let B_i be the average intensity value in row i of the picture.

If

$$B_1 \geq B_2 \geq B_3 \geq B_4 \leq B_5 \leq B_6 \leq B_7$$

and if

$$B_1 > B_4 < B_7$$

then the point at the center of the square is on a dark line. If the two relations above do not hold, the central point is not on a dark line.

Now the procedure for locating the mouth can be described.

Step 1. The area to be searched for the mouth is delimited. This area extends vertically from below the nose

Digitized by Google

Digitized by Google

to near the predicted chin, Horizontally, it extends from left. to right just outside of the previously located eye centers,

Step 2, The mouth region is smoothed using 2x2 smoothing, Here as elsewhere, smoothing improves performance,

Step 3, Within the mouth region all points that are on dark horizontal lines are identified, The 7x7 line finding operator described above is used,

Step 4, A threshold is determined such that half of the points which are on dark horizontal lines are darker than this threshold, Points with values above this threshold are eliminated.

Step 5, The rows of the region under consideration are examined to find that row which contains the greatest number of the points remaining, At this point the mouth is easily identified as the string of remaining dark points on adjacent rows,

This completes the processing of the mouth, the final feature on the face to be determined, All measurements are now available, The classification algorithm can be applied to the measurements to identify the individual.

Digitized by Google

Digitized by Google

CHAPTER 9

PATTERN CLASSIFICATION

This chapter will describe the methods used to identify a subject once a set of measurements has been obtained from the picture. It is worthwhile to re-emphasize that the study of pattern classification methods does not represent a major part of this thesis. The location and identification of features in grey scale pictures is the concern of this thesis. Once the features are located and measurements obtained, it is necessary to use a classification algorithm for identification of the subject. For this purpose a standard and easily used classification method is used.

The pattern classification problem is an example of decision-making when there is uncertainty in the data on which the decision is based. It may be defined as follows. Given a real valued vector $\vec{x}$, which has been selected from one of the classes C1, C2,..., Cm, we wish to find which of the classes that $\vec{x}$ represents. That is, we wish to find a decision function of the feature vector $\vec{x}$ such that

$$f(\vec{x}) = i \quad \text{if } x \in Ci.$$

In this thesis, the components of $\vec{x}$ are the measurements obtained from the picture, normalized by dividing each measurement by the measured height. The classes Ci each represent one person whose measurements are in the recognition dictionary.

Digitized by Google

Digitized by Google

This pattern classification problem may be separated from experimental pattern recognition and treated as a problem in mathematical statistics. There exists an extensive literature concerning this problem. Many classification algorithms have been described. Various algorithms are appropriate depending on the knowledge which is available about the probabilities of the classes and the probability distributions of the features for members of each class. The many methods which have been proposed and used are broadly surveyed and described in the recent papers by Nagy [1968] and Ho and Agrawala [1968]. Both papers contain lengthy bibliographies of the important literature dealing with pattern classification.

Identification of people, as Bledsoe and Hart have observed, is characterized by a small number of samples from each of a large number of classes. This contrasts with traditional pattern recognition problems where one has many samples from a few classes. Thus it is difficult to estimate probability distributions.

Accordingly, from the many classification schemes that are available, the <u>nearest neighbor</u> method (Cover and Hart [1967]) was chosen for this work in identification of people. The reason for this choice is that the nearest neighbor method is simple, easy to use, intuitively appealing, and requires no prior knowledge of underlying probability distributions.

Digitized by Google

Digitized by Google

The nearest neighbor method may be described as follows. For each person whose measurements are in the recognition dictionary, one or several n-component measurement vectors have been obtained. Each measurement vector may be considered as specifying a point in an n-dimensional space. A distance function $D(\vec{x},\vec{y})$ is defined between any two points $\vec{x}$ and $\vec{y}$ in the space. When it is necessary to classify a new measurement vector $\vec{x}$, the distances between $\vec{x}$ and each point in the dictionary are computed. The closest point to $\vec{x}$ is that $\vec{y}_i$ for which $D(\vec{x},\vec{y}_i)$ is minimum, and $\vec{x}$ is identified as representing the same person as $\vec{y}_i$.

With relatively few vectors for each person in the dictionary, this sort of classification will work well if the feature vectors for each person are nicely clustered in n-space. The success of the Bertillion system and Bledsoe's work with identification from measurements indicates that this is indeed the case for accurate measurements. In the recognition system studied here, the measurements contain measurement errors, but the nature of these errors is difficult to estimate. Nevertheless, the nearest neighbor classification method provides a simple way to test the feature detection that forms the principal effort in this thesis.

As stated before, the nearest neighbor procedure is simple and intuitively appealing. Its chief disadvantage,

Digitized by Google

Digitized by Google

the large memory required to store every previously encountered measurement, is not bothersome with the relatively small data sets used here. There is another feature of the nearest neighbor procedure which is worth mentioning. It is well known that when all the underlying probability structure of a classification problem is known and used, the optimum decision rule may be obtained using Bayes analysis and choosing the class Ci which maximizes

$$P(Ci|\vec{x}) = \frac{P(\vec{x}|Ci)\ P(Ci)}{\sum_{j=1}^{m} P(\vec{x}|Cj)\ P(Cj)} ,$$

Cover and Hart [1967] have shown that for large samples, the probability of error of the nearest neighbor rule is bounded above by twice the Bayes probability of error. (Minsky and Papert [1969, p. 197] give a clear example of a special case of this result.)

In choosing the distance function for use in the nearest neighbor algorithm, the metric which was used successfully by Bledsoe [1966] was selected. The distance from $\vec{x}$ to $\vec{y}$ is defined as

$$D(\vec{x},\vec{y}) = \sum_{j=1}^{n} \frac{1}{\sigma_j^2}(x_j - y_j)^2$$

where σ_j^2 is the intra-person difference variance for measurement j; in other words, σ_j^2 is the variance of differences in measurement j between pictures of the same person. More precisely, let $\vec{a}$ and $\vec{b}$ be two measurement vectors for person i. Form the difference vector $\vec{di}$

Digitized by Google

Digitized by Google

for person i as

$$ai_j = a_j - b_j \ .$$

Do the same for each person until a complete set of difference vectors $\vec{a1}, \vec{a2}, \ldots, \vec{am}$ has been obtained. Define σ_j^2 as the variance of the jth component of these difference vectors:

$$\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}(ai_j)^2 - \left[\frac{1}{m}\sum_{i=1}^{m} ai_j\right]^2 \ .$$

Clearly this distance measure is a Euclidean distance on a space in which each component has been weighted by a confidence level $\frac{1}{\sigma_j^2}$. This weight will be large on a measurement that is reliably repeatable. It will be small on an unreliable measurement.

It was mentioned above that this is the distance measure used by Bledsoe. To be precise, it should be mentioned that Bledsoe had many more measurements in his manually obtained data. These measurements were grouped according to the features. For instance, one group might be all the measurements on points of the mouth. The distance measure presented above is the measure used by Bledsoe for the distance between centroids of groups.

Digitized by Google

Digitized by Google

CHAPTER 10
RESULTS AND CONCLUSIONS

The principal positive result of this research is the use of goal-directed techniques to successfully locate features in cluttered digital pictures. This success has been verified by displaying the results of the feature finding algorithms and comparing these locations with the locations obtained by hand from digital printouts of the pictures. Successful performance in the task of identification of people provides further verification for the feature finding algorithms. The test of the performance of the program on identification of people is described below.

A collection of 72 digital pictures was used in this test. This collection comprised 24 sets of pictures. Each set consisted of 3 pictures for one individual: a picture of the body, a picture of the background which corresponded to the body picture, and a close-up of the head. Ten individuals were represented in the collection. For most, there were 2 sets of pictures. Three individuals had 3 or 4 pictures each in the collection. The pictures of the same individual were taken at various times over a two year period.

A large amount of information will be included in this chapter about the collection of pictures which was used in

Digitized by Google

Digitized by Google

the performance test. This is done in order to provide a clear indication of the quality of the input and the characteristics of the results. In addition this inforation could be used in compairison of alternate methods with the methods of this thesis. A full set of the data used to test this program is available from the author.

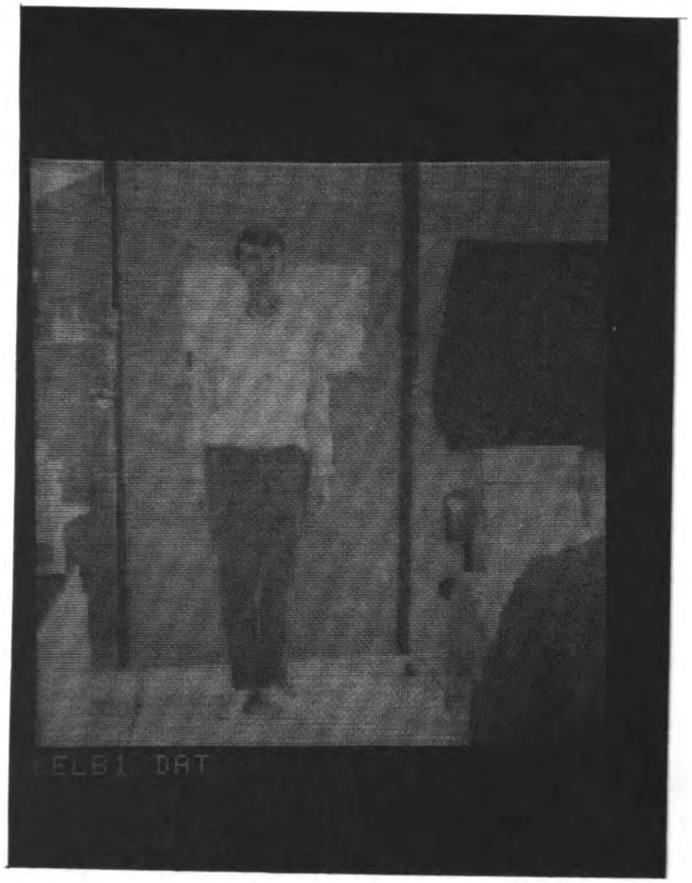
Detailed information on the pictures of the entire body will be provided for 4 pictures, 2 pictures each of 2 individuals.

The complete input data for 2 of these pictures is included as Appendix 1. For each picture several pages of computer printout are given. Taken together these pages completely specify the input picture. In the printouts in Appendix 1 (and also in Appendix 2, to be described later) the sixteen levels of light intensity are represented as follows. Each point contains a light intensity value which ranges from zero (the darkest points) to fifteen (the lightest points). Intensity values Ø through 9 are represented by the digits "Ø" through "9". Intensity value 1Ø is represented by the letter "A"; 11 by "B"; and 12, 13, and 14 by the letters "C", "D", and "E" respectively. Intensity value 15 is printed as a period.

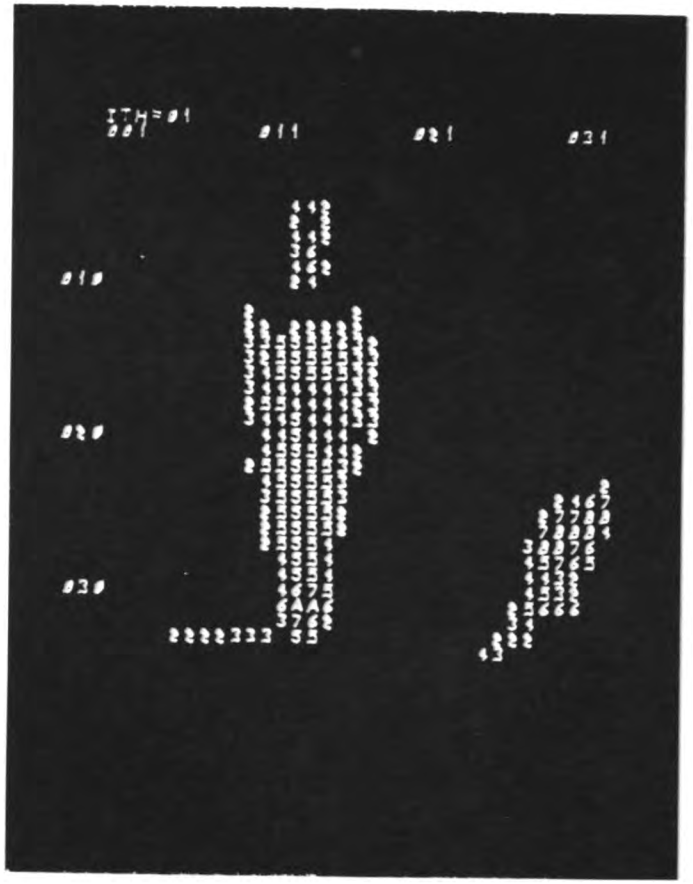
Each of Figures 1Ø-1 through 1Ø-4 gives 4 photographs derived from the 4 example pictures of bodies. Part (a) gives a grey scale representation which was obtained by photographing a computer display terminal. Part (b) shows

Digitized by Google

Digitized by Google



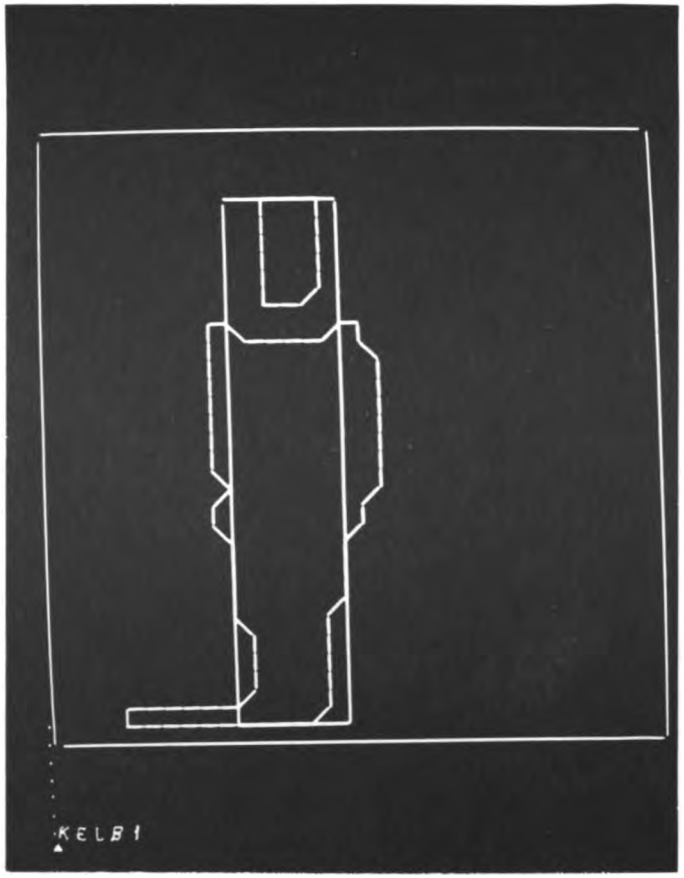
(a)



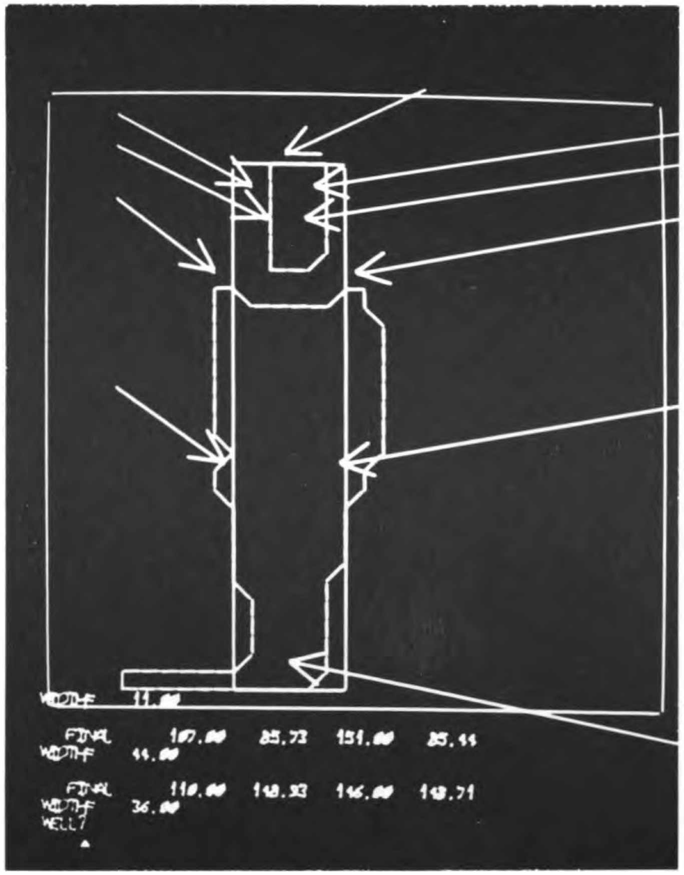
(b)

Figure 10-1. Picture KELB1

Digitized by Google



(c)

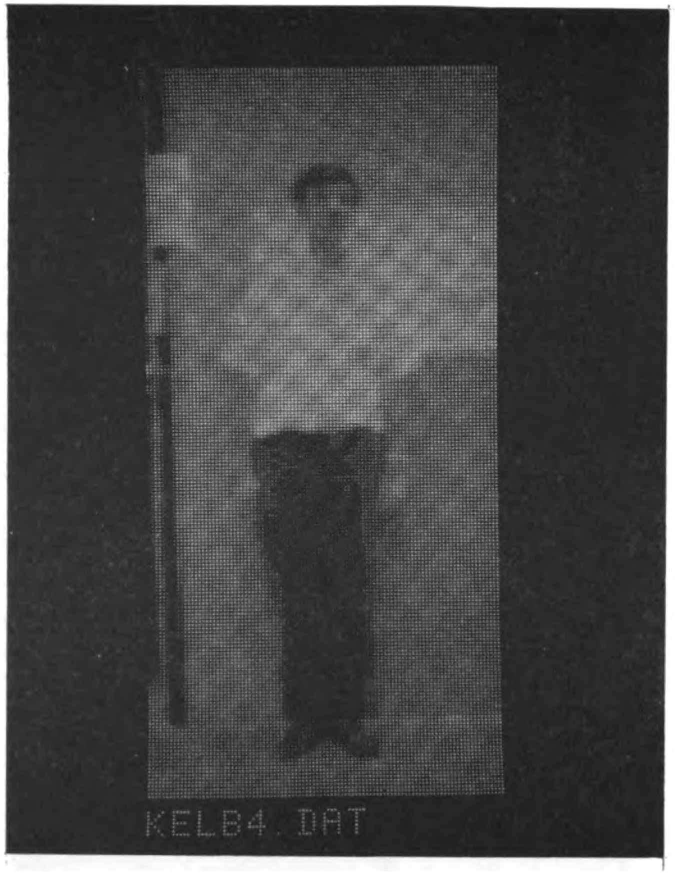


```
WIDTHF      11.00
   FINAL        107.00    85.73    151.00    85.44
WIDTHF      44.00
   FINAL        110.00   148.93    146.00   148.71
WIDTHF      36.00
WELL7
```

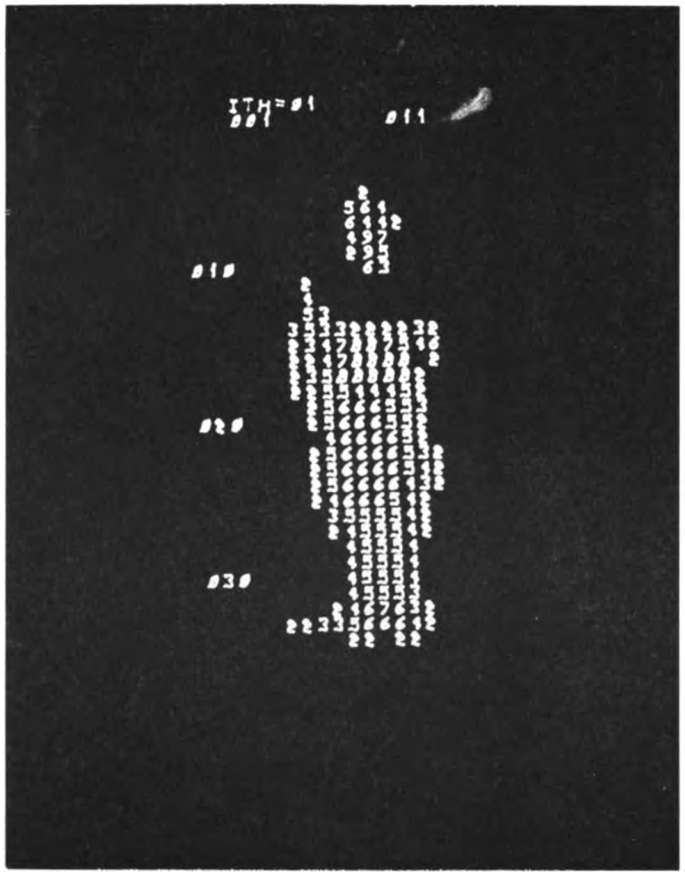
(d)

Figure 10-1 (continued)

Digitized by Google



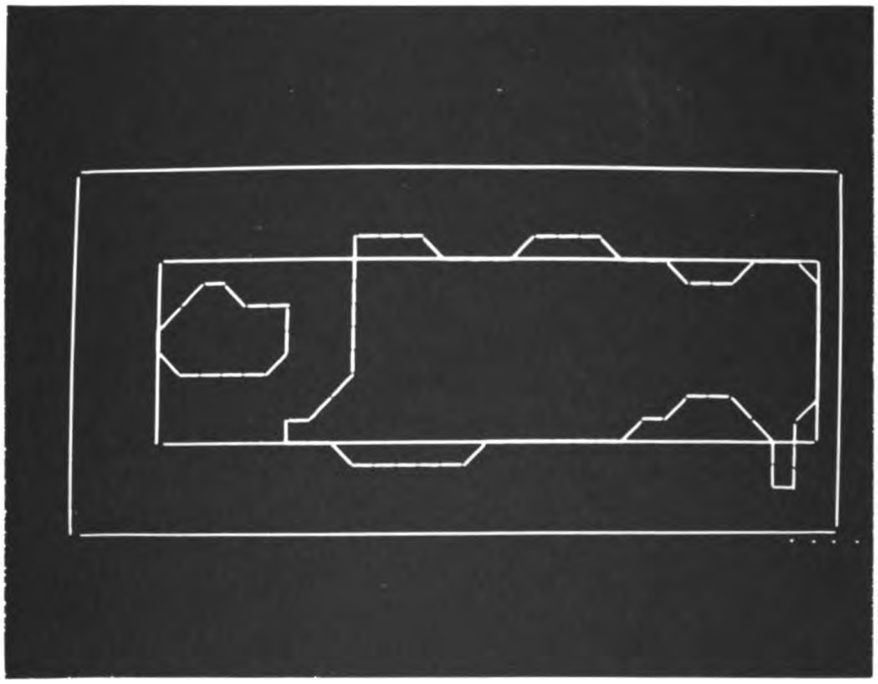
(a)



(b)

Figure 10-2. Picture KELB4

Digitized by Google


(c)

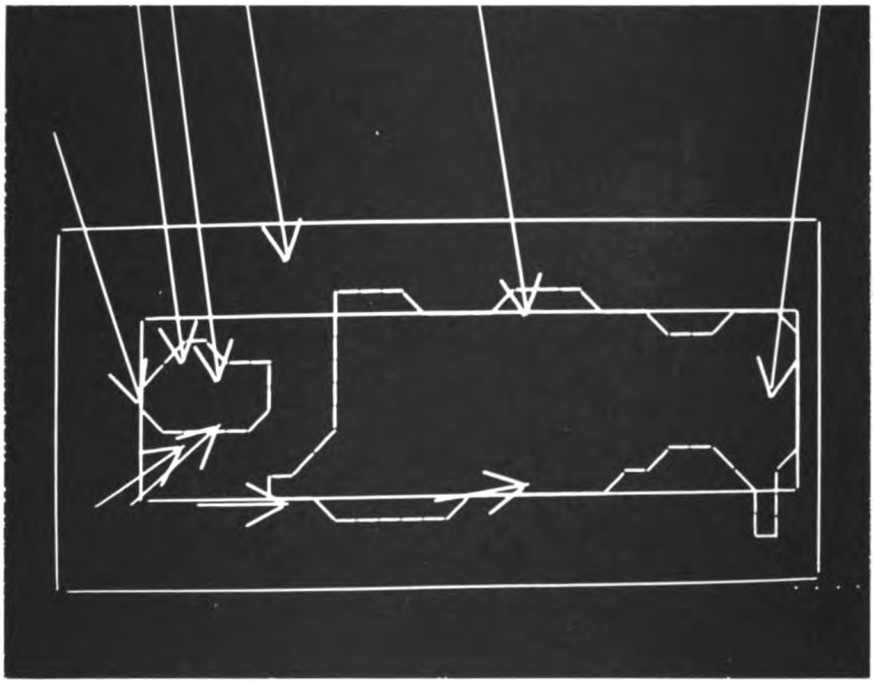
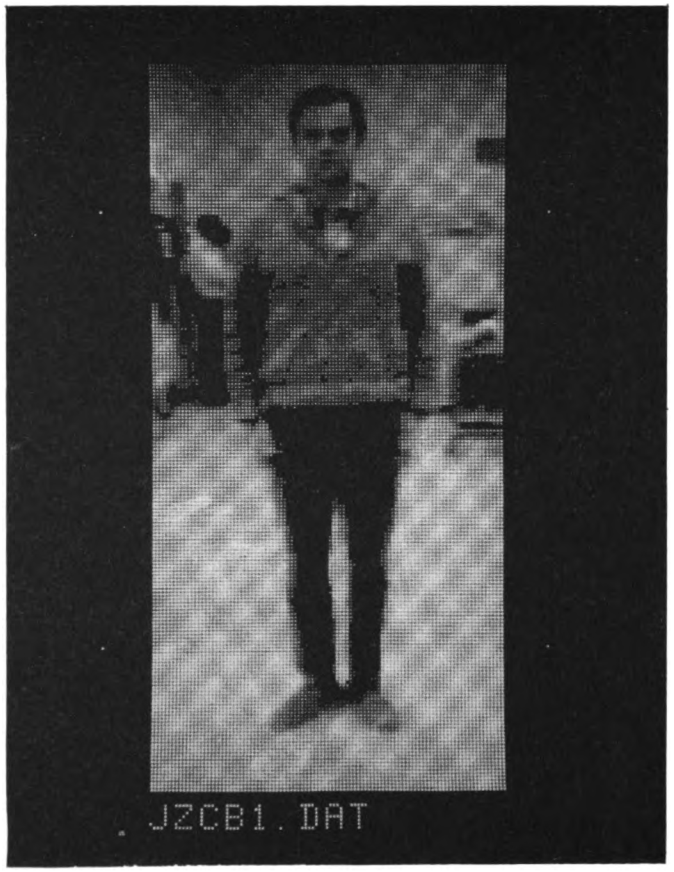

(d)

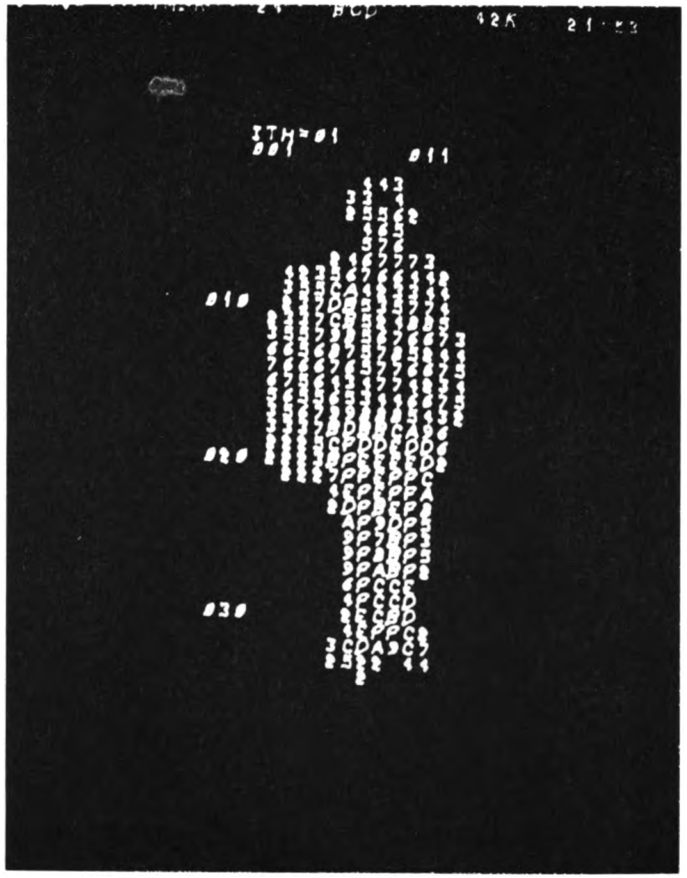
Figure 10-2 (continued)

Digitized by Google

Digitized by Google



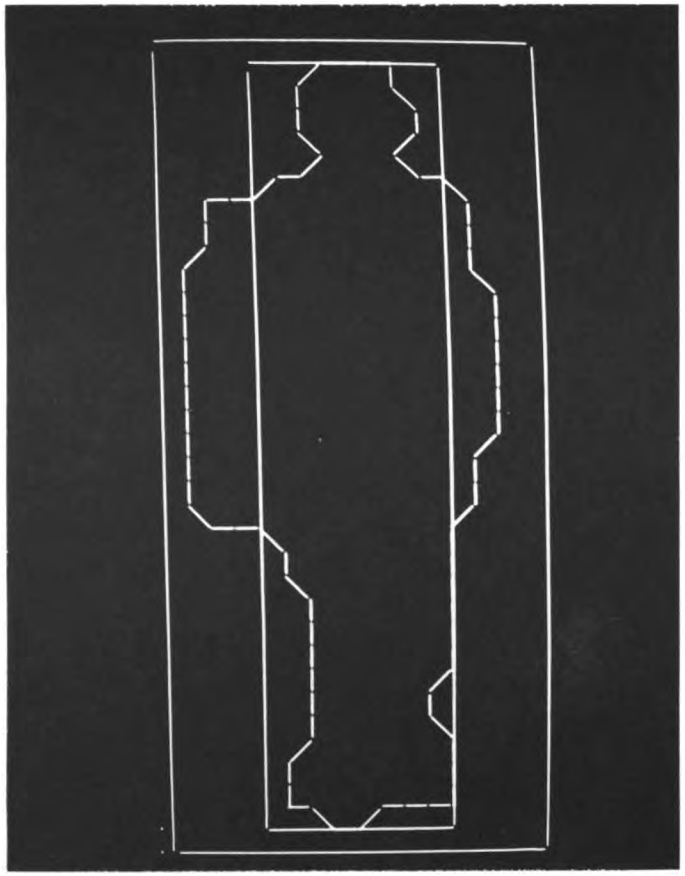
(a)



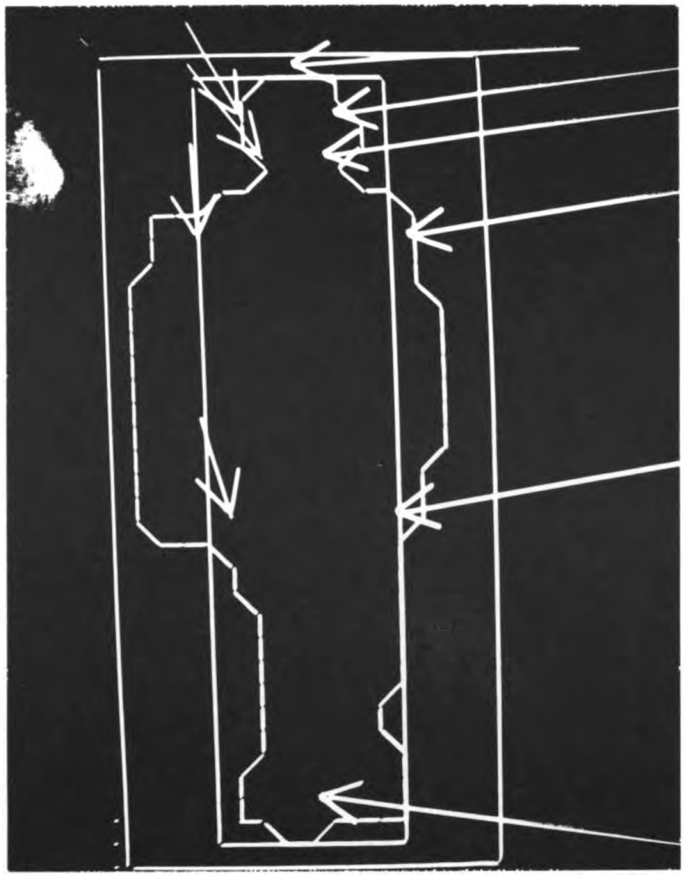
(b)

Figure 10-3. Picture JZCB1

Digitized by Google



(c)

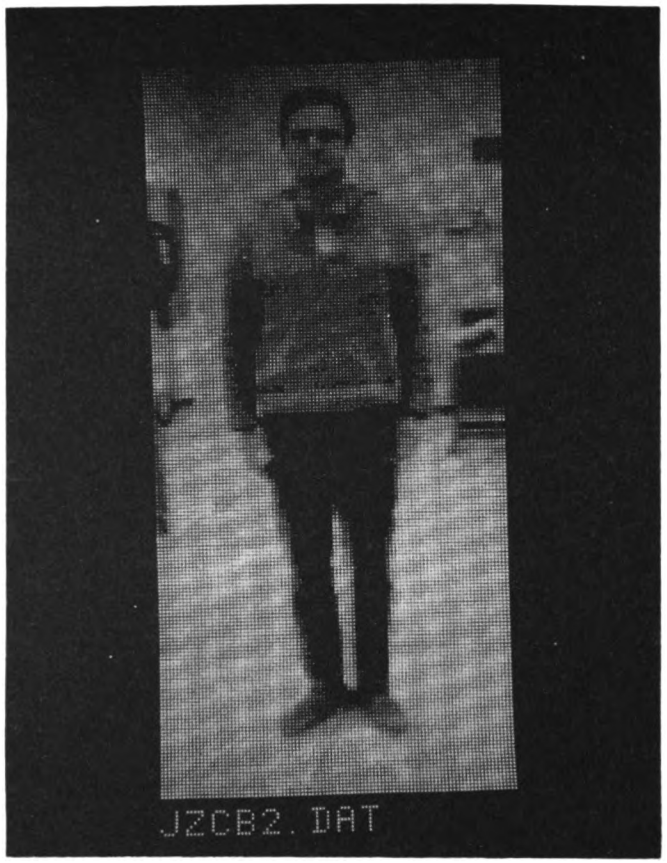


(d)

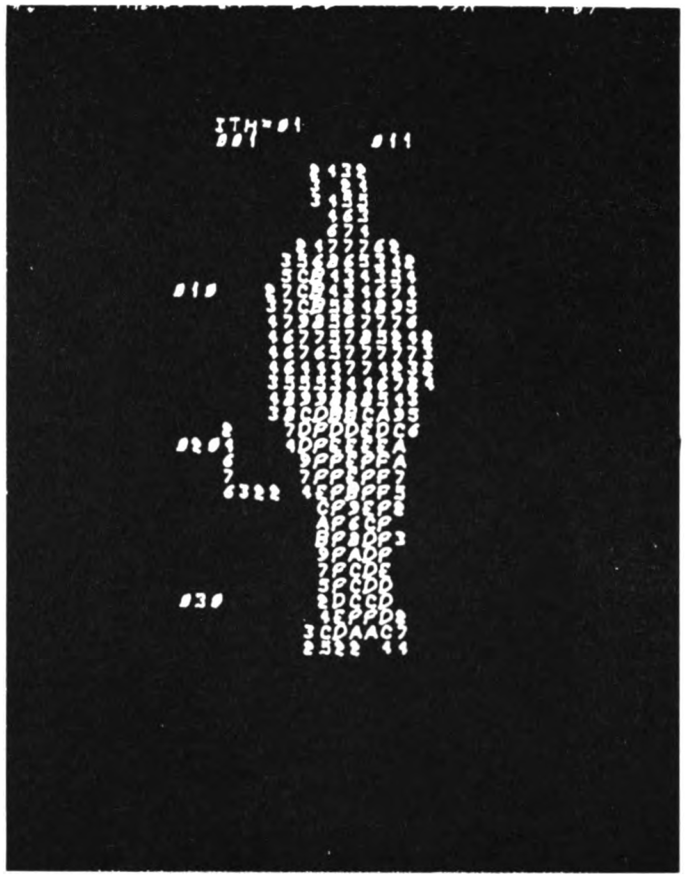
Figure 10-3 (continued)

Digitized by Google

Digitized by Google



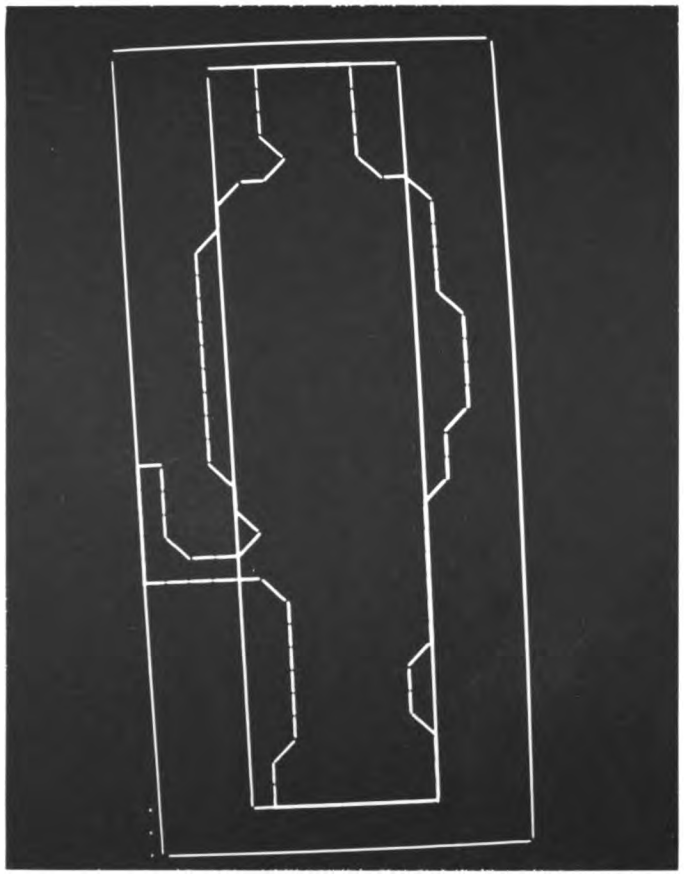
(a)



(b)

Figure 10-4. Picture JZCB2

Digitized by Google


(c)

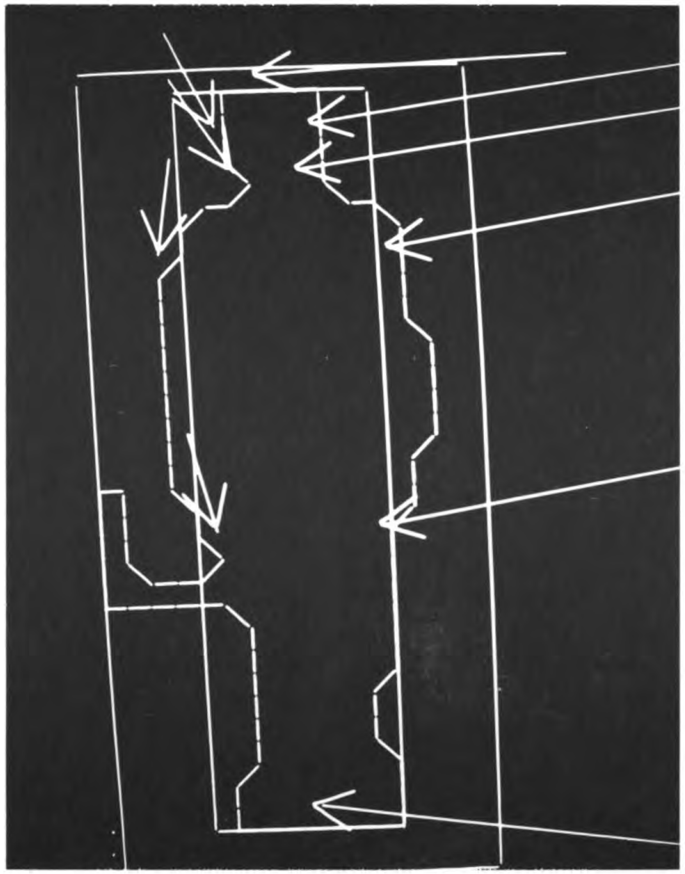

(d)

Figure 10-4 (continued)

Digitized by Google

Digitized by Google

the results of processing the pictures after subtraction and reduction in size. The representation is the same as that used in Appendix 1 which was described above. Part (c) shows the outlines of the regions which remain after smoothing, noise elimination, and removal of the border regions. Part (d) shows the same outlines as part (c) with arrows superimposed which point to the locations of the measurement points.

Table 10-1 gives the coordinates of all of the measurement points located in the 4 example body pictures.

Appendix 2 consists of complete input data for 2 pictures of heads, both representing the same individual. The data representation is the same as that used in Appendix 1 and described above.

Figures 10-6 through 10-29 contain pictures of each of the 24 heads in the test collection. In each figure part (a) is a grey scale photograph from a computer display terminal. Part (b) shows the outline of the head, the features found, and coordinates of features located. The eyes are represented by an outline of the dark areas around the eyes. Similar outlines are given for the nostrils. The mouth is indicated roughly by horizontal lines. Around the nose and mouth some noise points have not been suppressed. On the right of the outline of the head the feature coordinates that are used in subsequent measurements are given. Figure 10-5 explains which numbers belong with

Digitized by Google

Digitized by Google

TABLE 10-1.

Coordinate measurements from 4 body pictures.

Picture KELB1

				x coord.	y coord.
Top of Head				129	45
Feet				130	213
Sides of head	left			119	56
	right			138	56
		Width=	19		
Sides of neck	left			124	67
	right			135	67
		Width=	11		
Sides of shoulders	left			107	86
	right			151	85
		Width=	44		
Sides of hips	left			110	149
	right			146	149
		Width=	36		

Picture KELB4

				x coord.	y coord.
Top of Head				143	46
Feet				142	219
Sides of head	left			132	58
	right			153	58
		Width=	21		
Sides of neck	left			137	68
	right			148	68
		Width=	11		
Sides of shoulders	left			116	88
	right			179	88
		Width=	63		
Sides of hips	left			120	153
	right			164	153
		Width=	44		

Digitized by Google

Digitized by Google

TABLE 10-1. (Continued)

Coordinate measurements from 4 body pictures.

Picture JZCB1

				x coord.	y coord.
Top of Head				142	25
Feet				143	214
Sides of head	left			129	38
	right			152	38
		Width=	23		
Sides of neck	left			134	49
	right			149	49
		Width=	15		
Sides of shoulders	left			117	71
	right			169	71
		Width=	52		
Sides of hips	left			123	142
	right			164	142
		Width=	41		

Picture JZCB2

				x coord.	y coord.
Top of Head				137	24
Feet				142	214
Sides of head	left			127	37
	right			150	37
		Width=	23		
Sides of neck	left			131	49
	right			146	48
		Width=	15		
Sides of shoulders	left			112	71
	right			167	69
		Width=	55		
Sides of hips	left			122	142
	right			162	141
		Width=	40		

Digitized by Google

Digitized by Google

Figure 1Ø-5.

Explanation of measurements in
Figures 1Ø-6 through 1Ø-29.

On the right side of each figure are ten values. The meaning of these values is defined by their position as follows:

```
<NAME OF PICTURE>

                                <TOP OF HEAD, Y COORD.>

<HEAD, LEFT SIDE X COORD.>

<HEAD, RIGHT SIDE X COORD.>

<LEFT EYE X COORD.>             <LEFT EYE Y COORD.>

<RIGHT EYE X COORD.>            <RIGHT EYE Y COORD.>

                                <NOSTRILS, Y COORD.>

                                <MOUTH, Y COORD.>
```

Digitized by Google

Digitized by Google

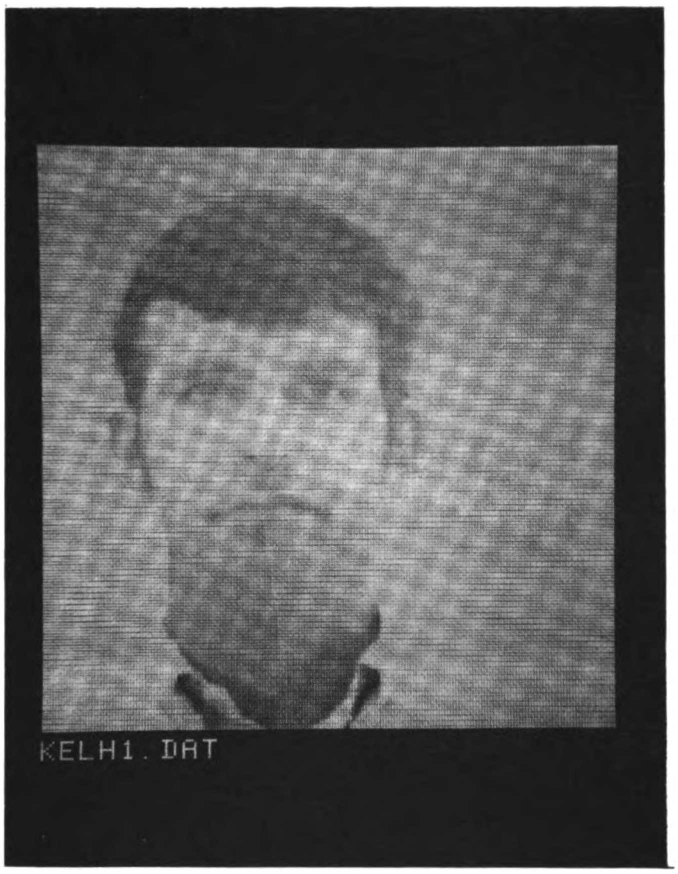
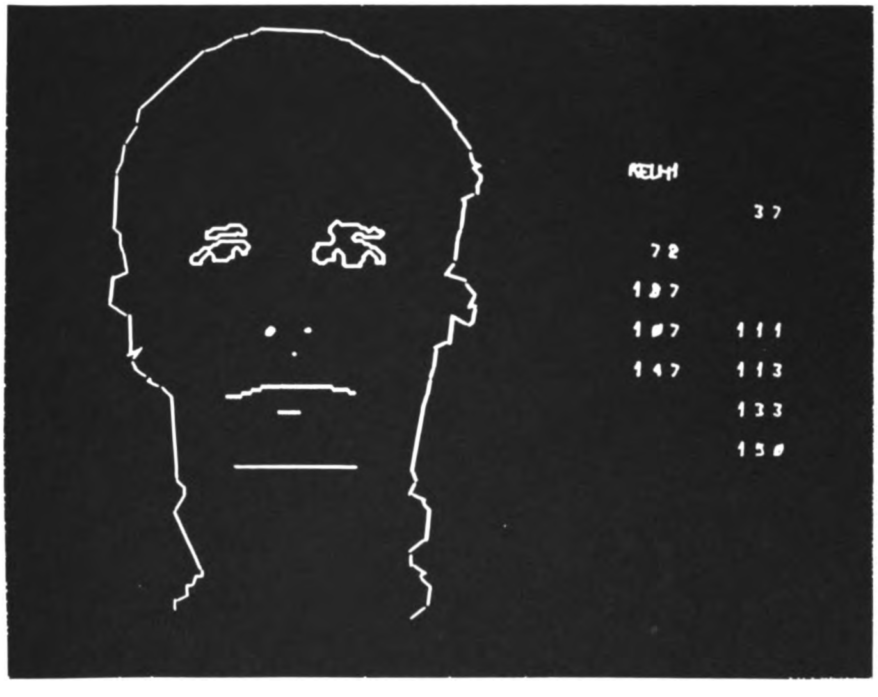


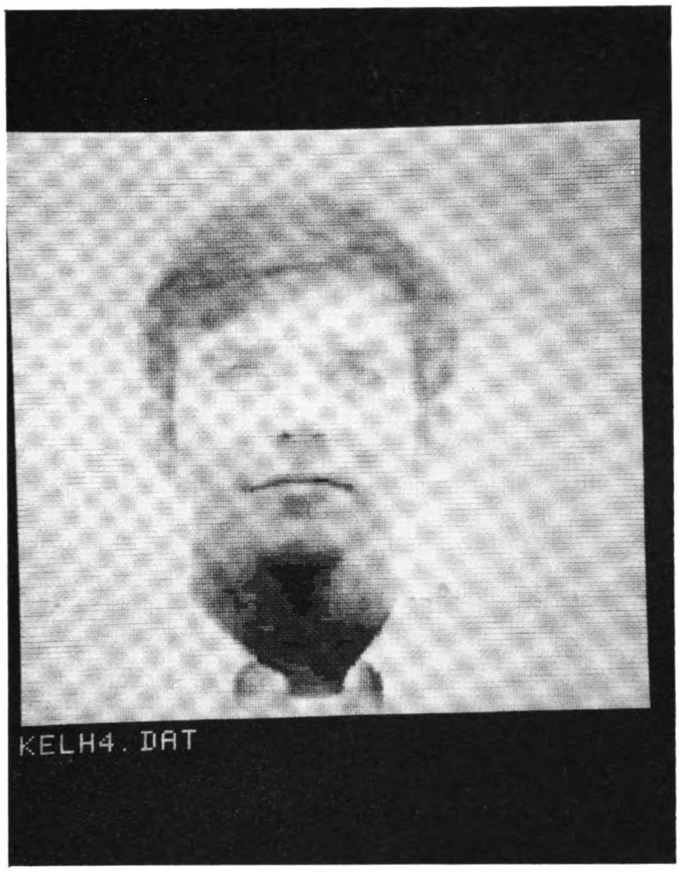
Figure 10-6



(b)

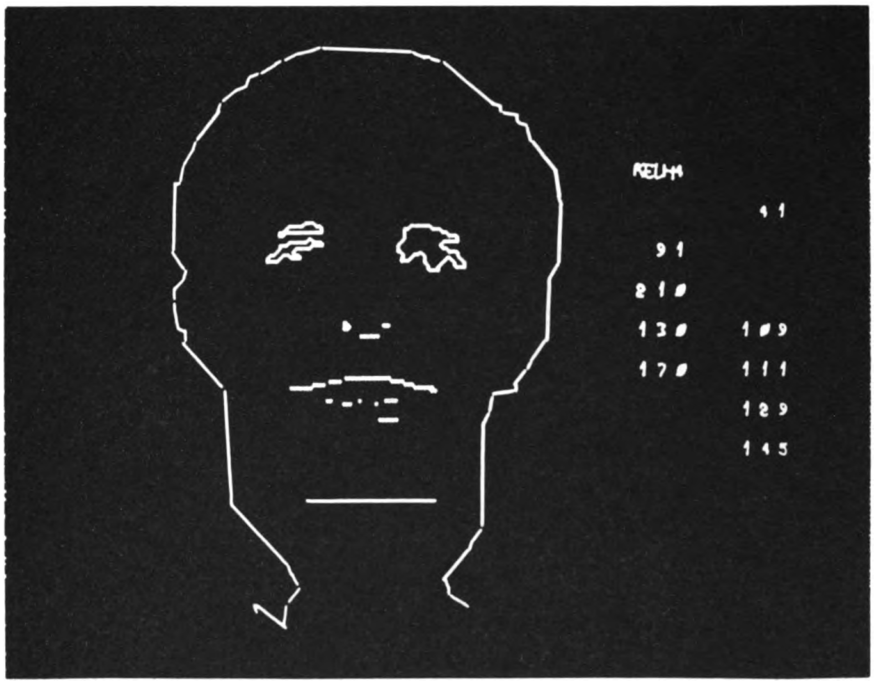
Digitized by Google

Digitized by Google



(a)

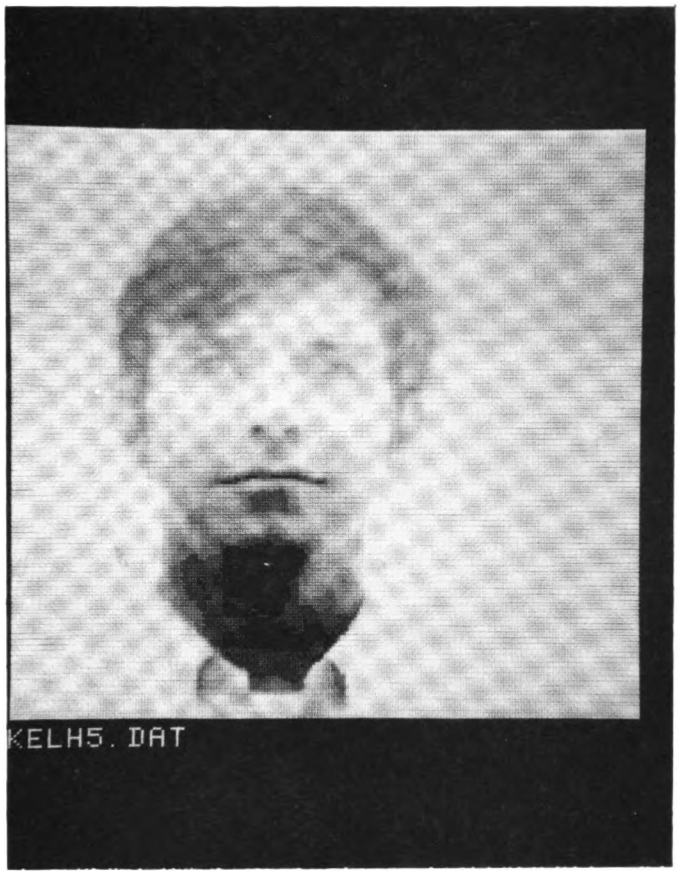
Figure 10-7



(b)

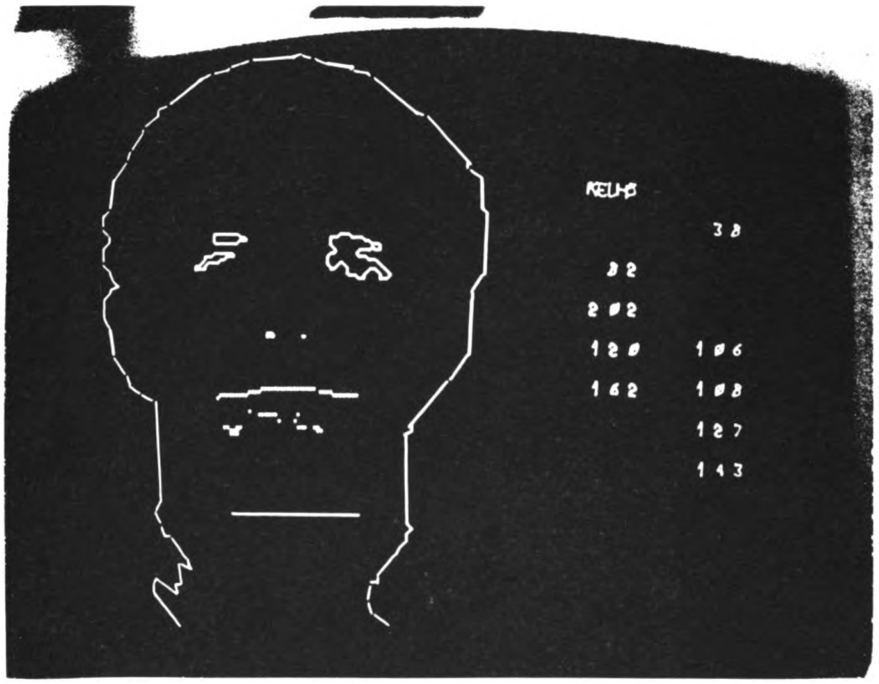
Digitized by Google

Digitized by Google



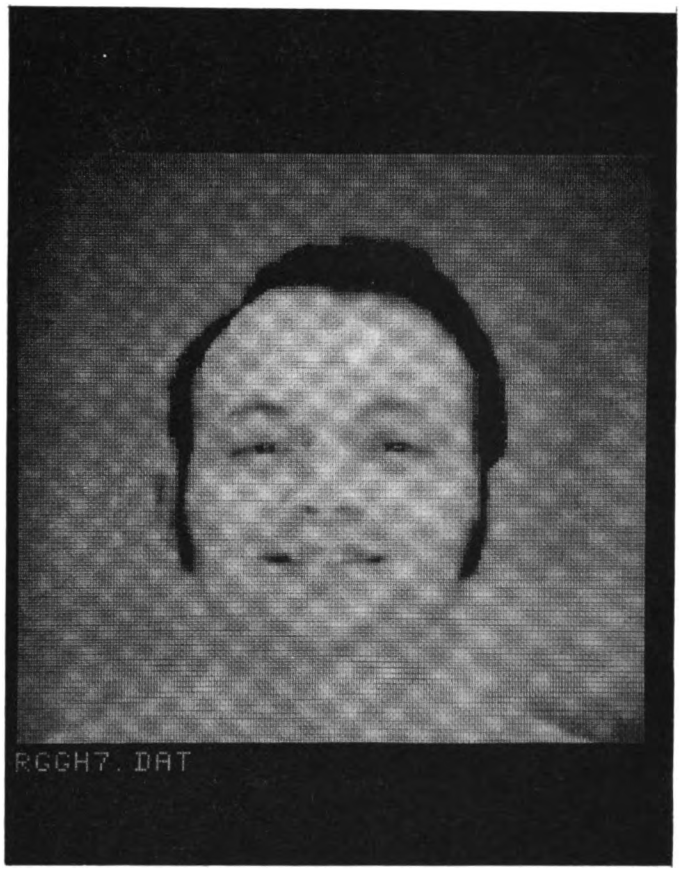
(a)

Figure 10-8



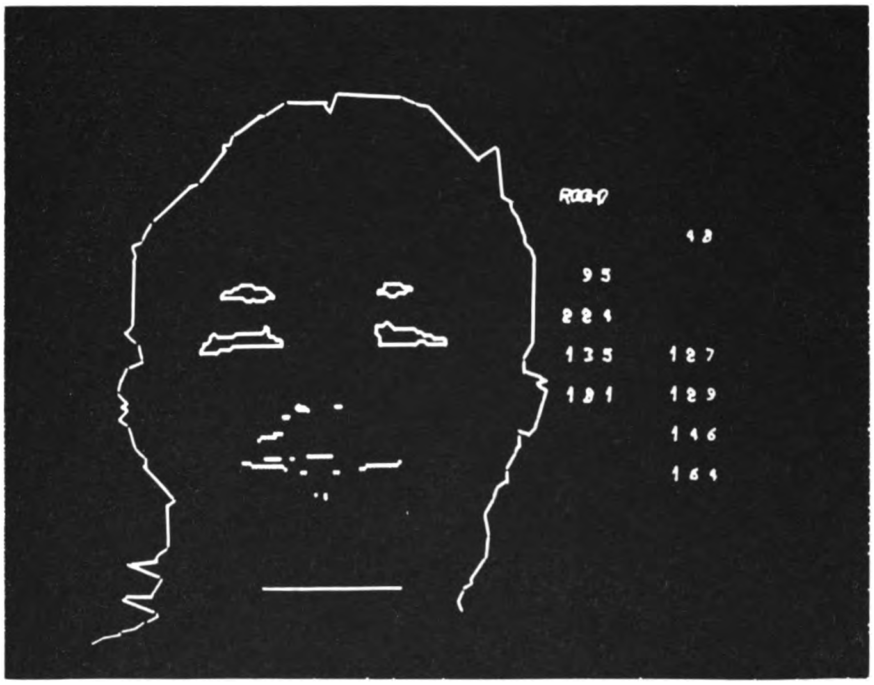
(b)

Digitized by Google



(a)

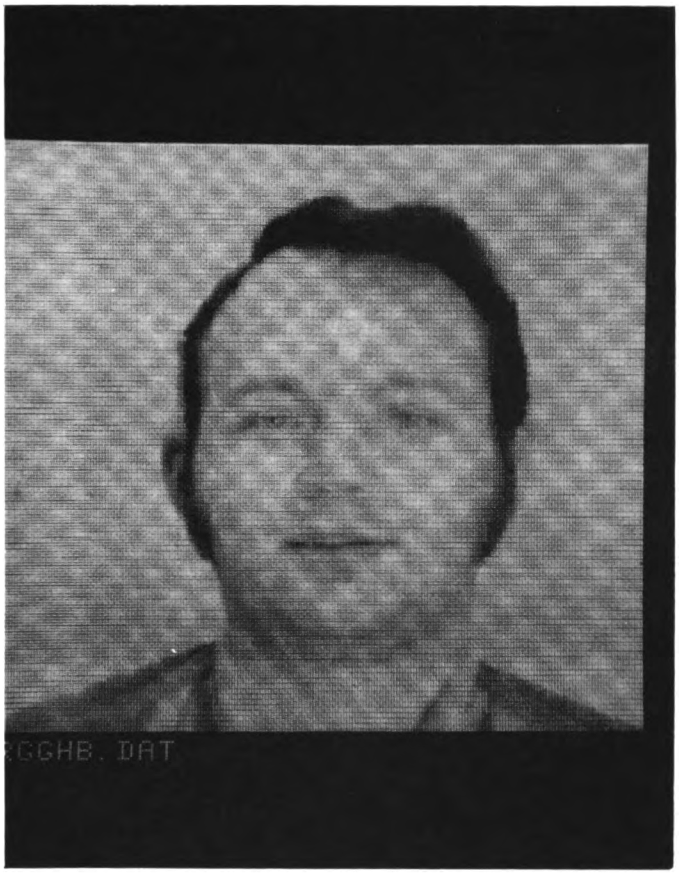
Figure 10-9



(b)

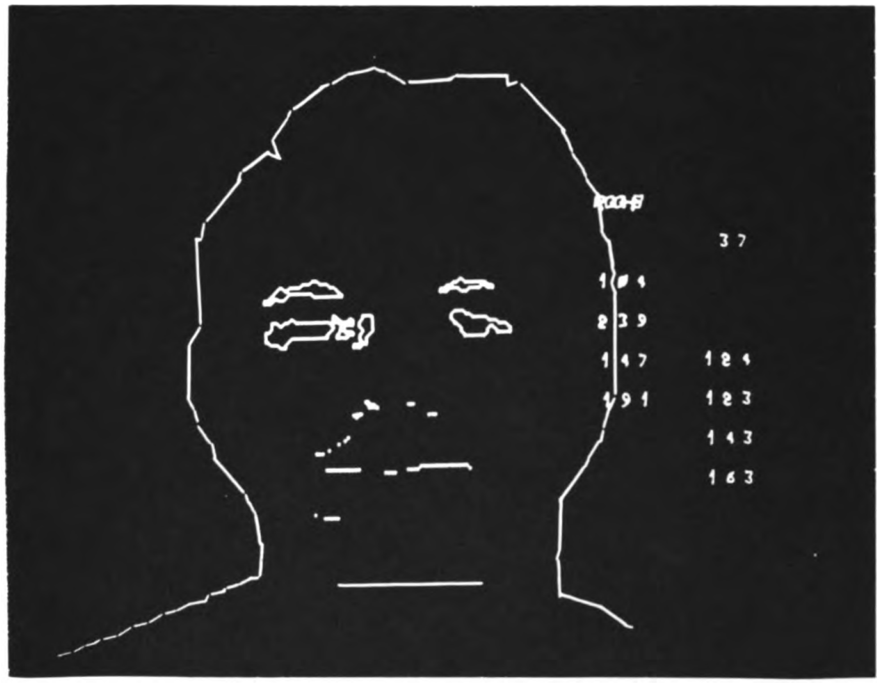
Digitized by Google

Digitized by Google



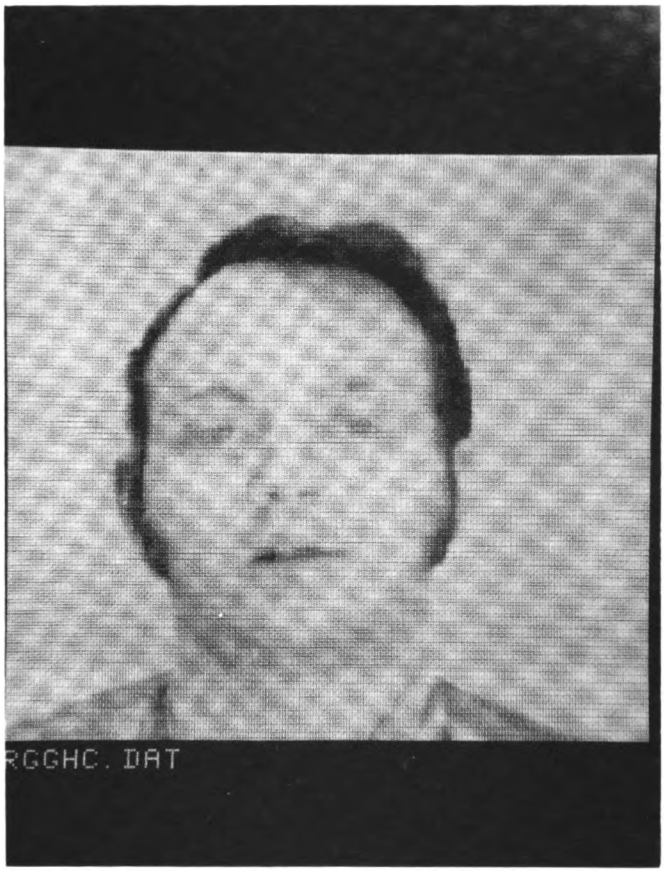
(a)

Figure 10-10



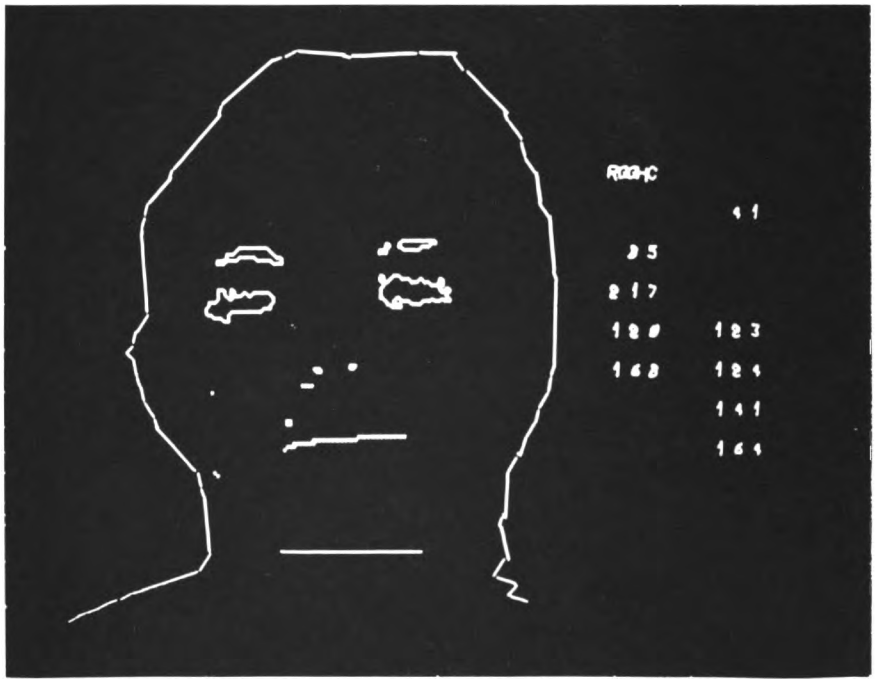
(b)

Digitized by Google



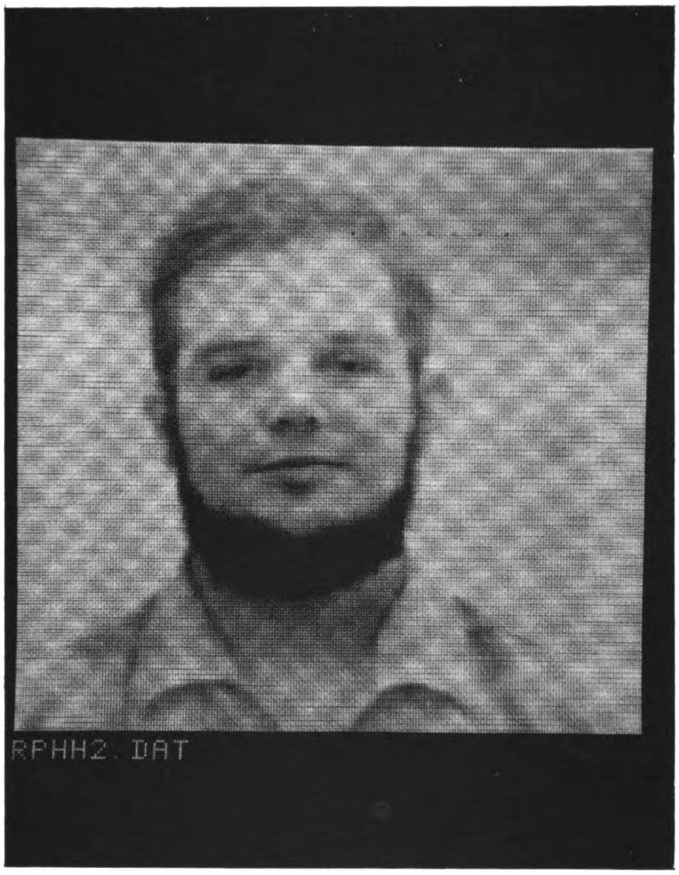
(a)

Figure 10-11



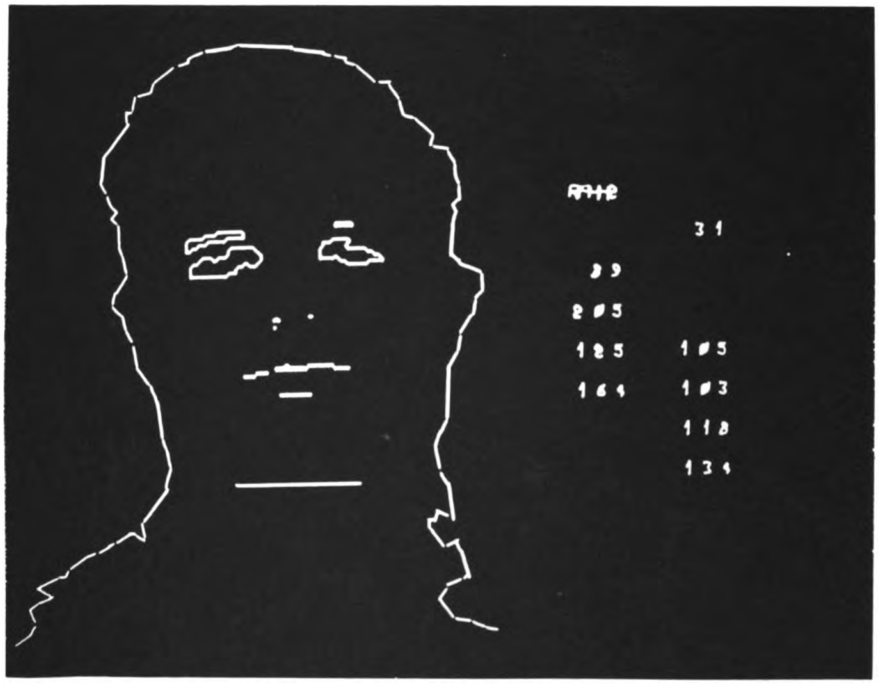
(b)

Digitized by Google



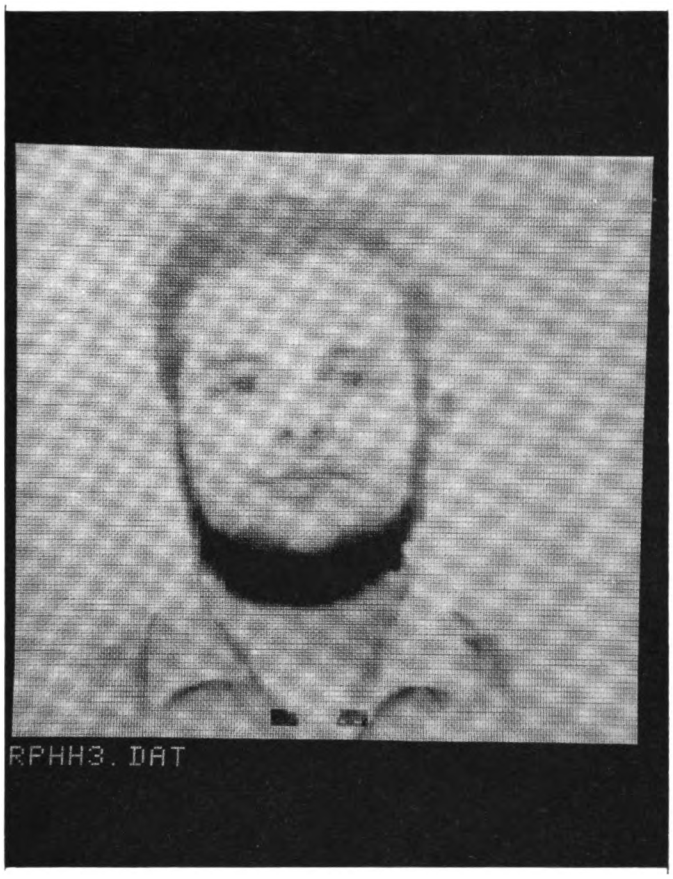
(a)

Figure 10-12



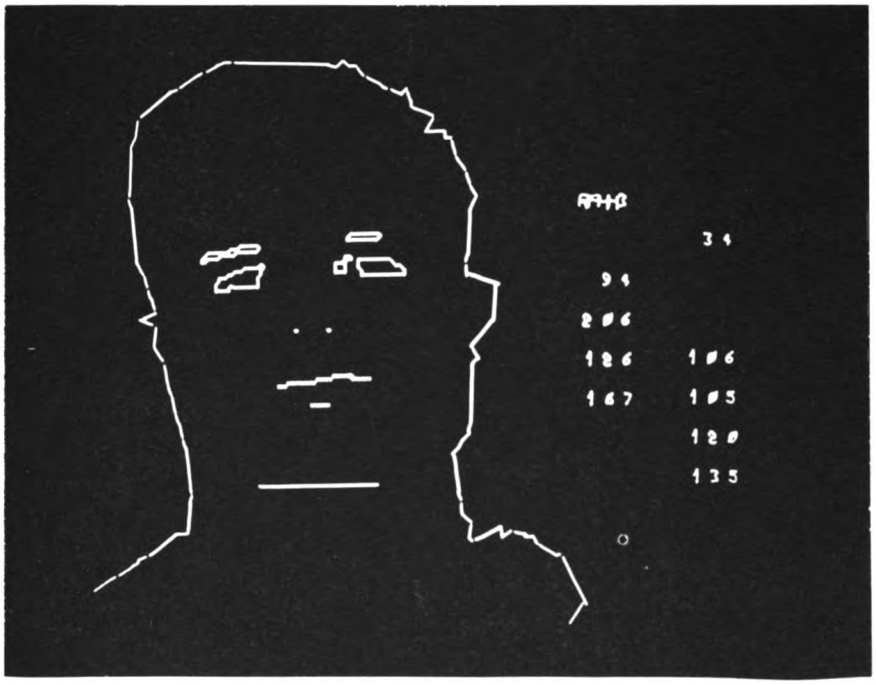
(b)

Digitized by Google



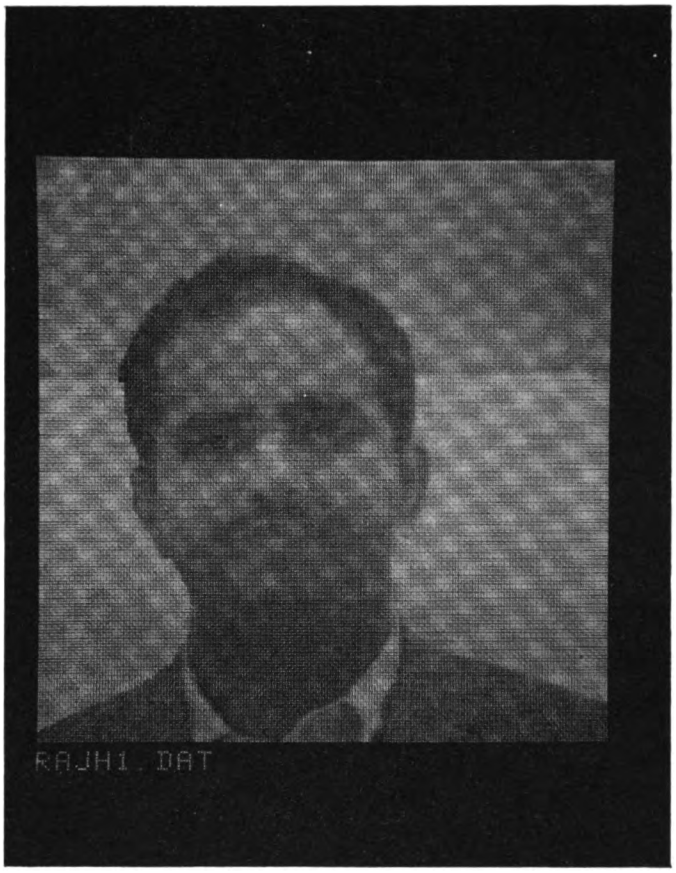
(a)

Figure 10-13



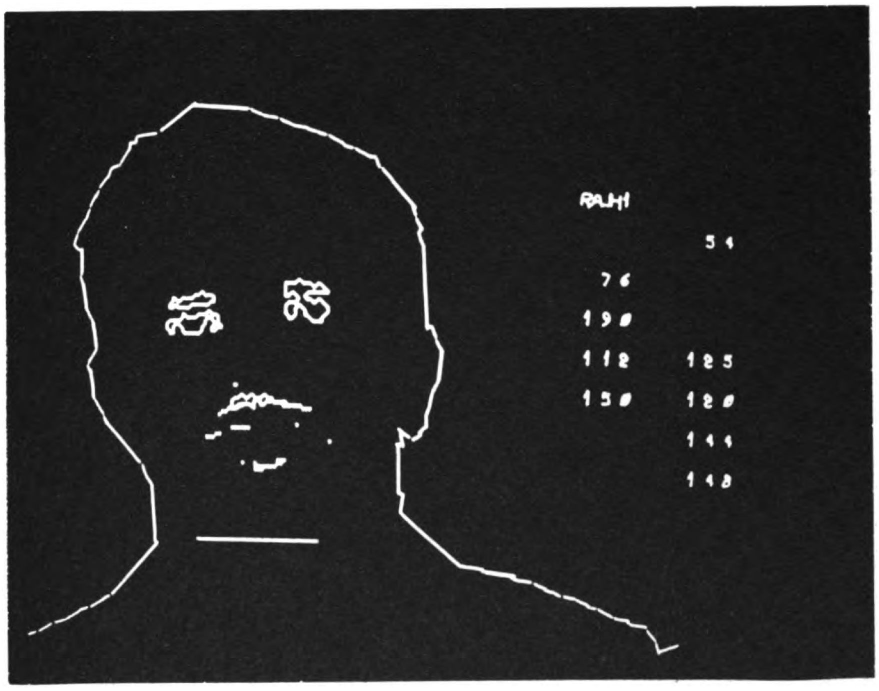
(b)

Digitized by Google



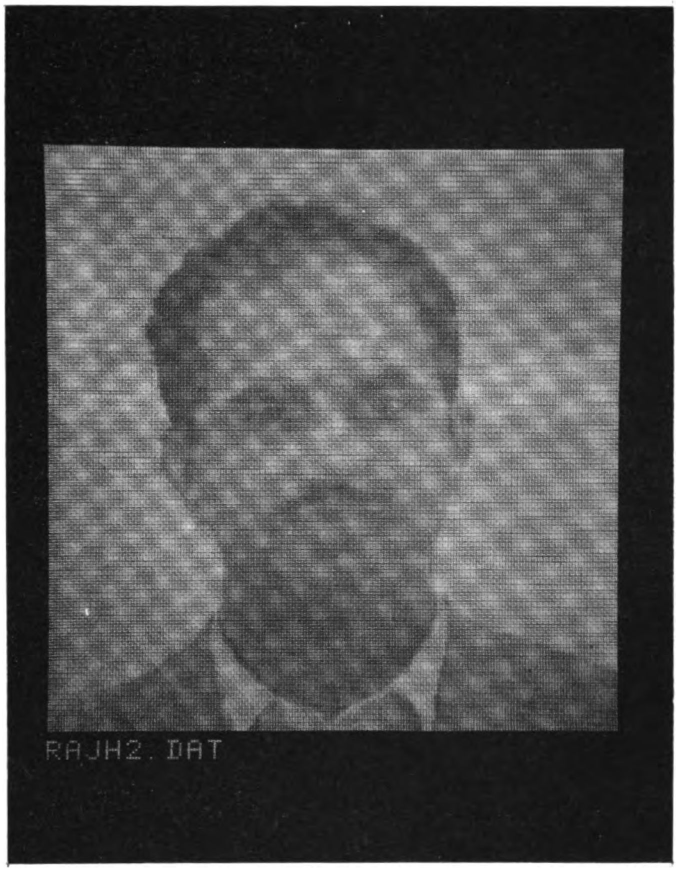
(a)

Figure 10-14



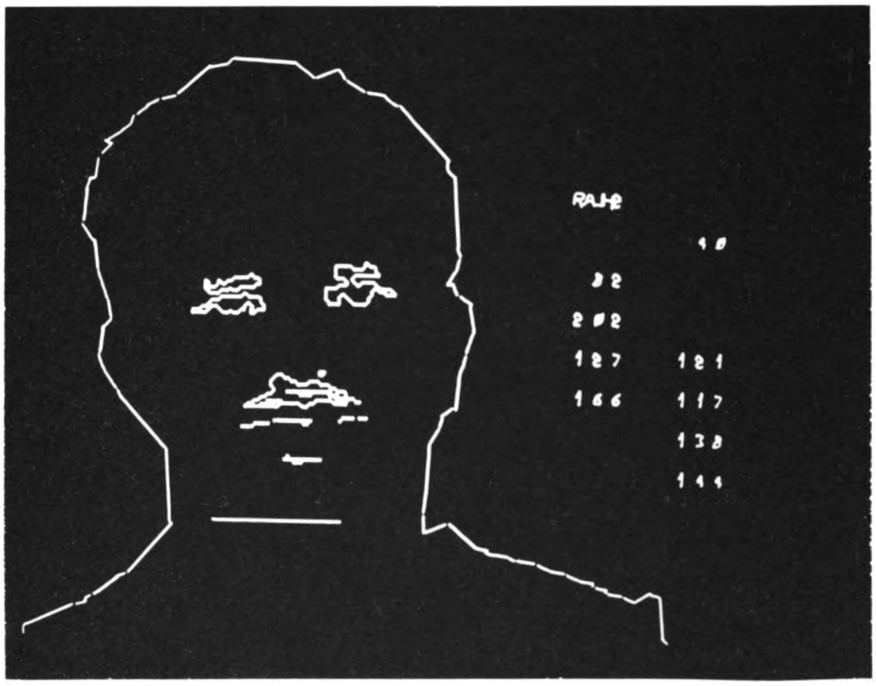
(b)

Digitized by Google



(a)

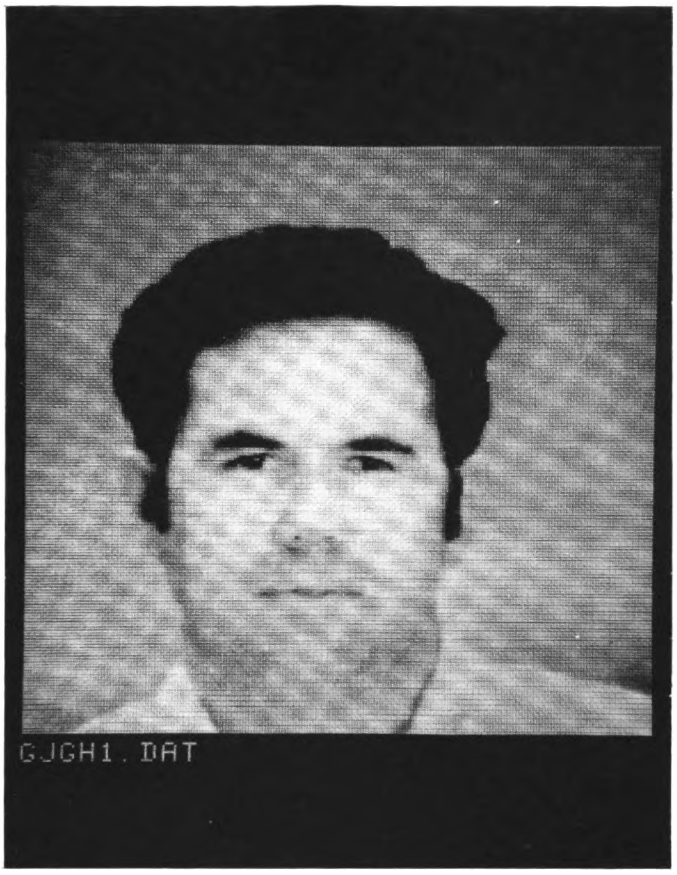
Figure 10-15



(b)

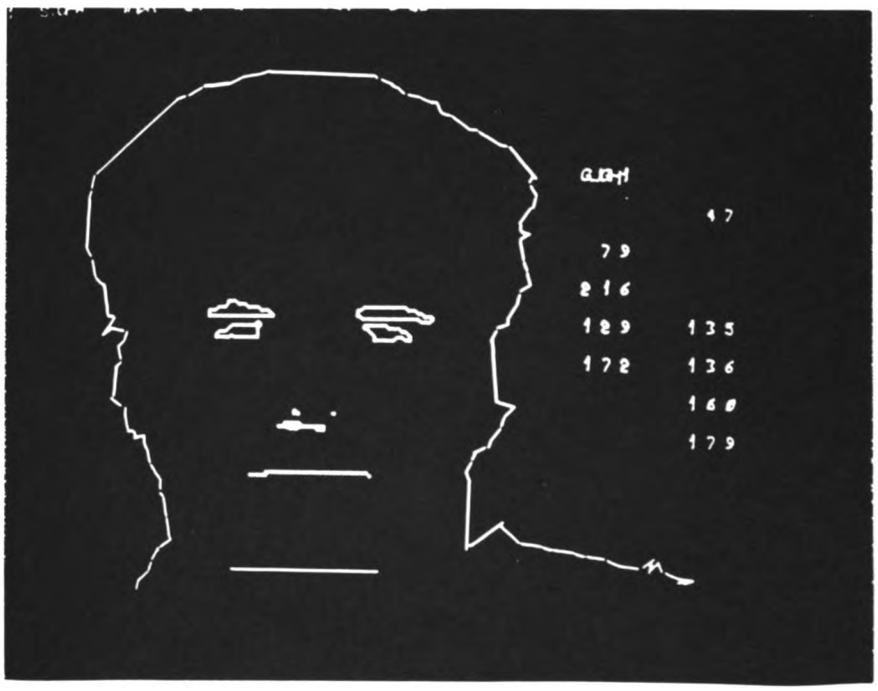
Digitized by Google

Digitized by Google



(a)

Figure 10-16

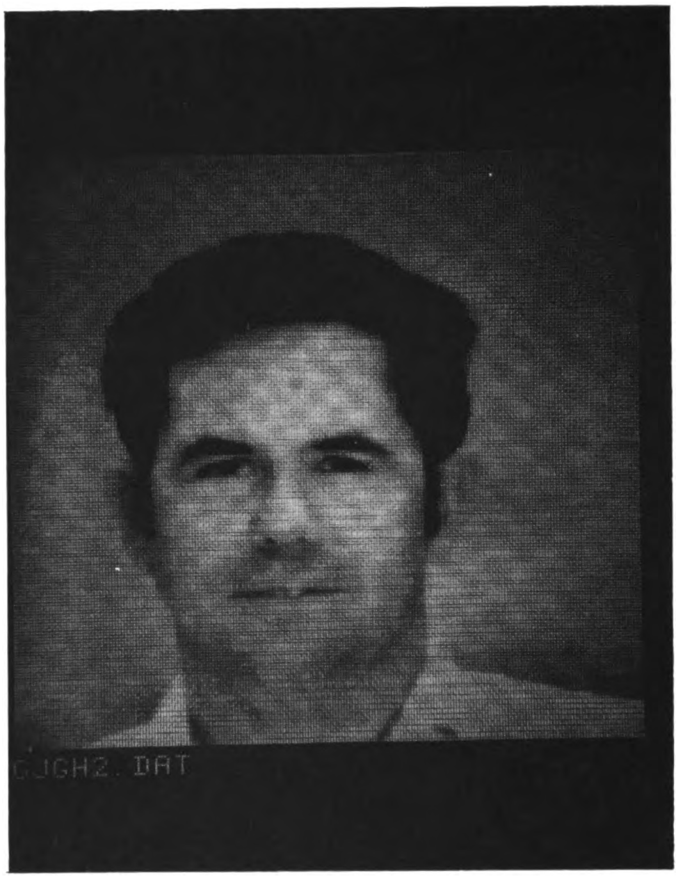


```
GJGH1
                 47
      79
     216
     129        135
     172        136
                160
                179
```

(b)

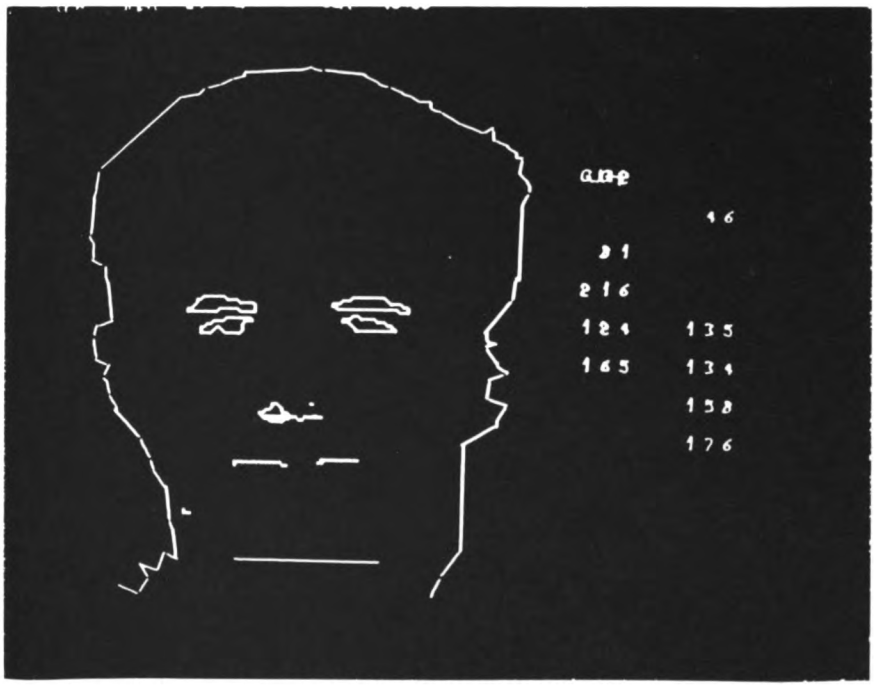
Digitized by Google

Digitized by Google



(a)

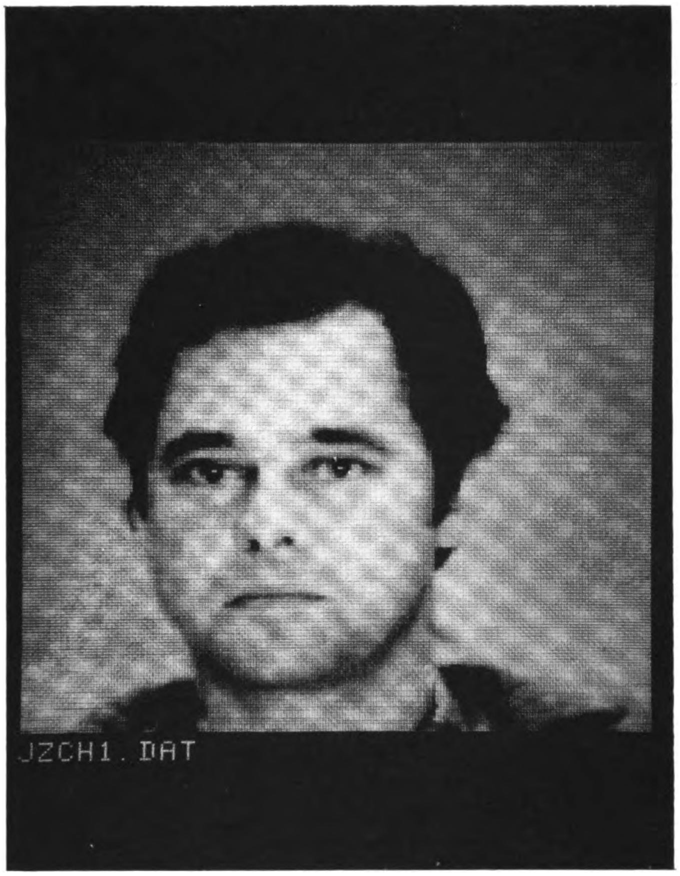
Figure 10-17



(b)

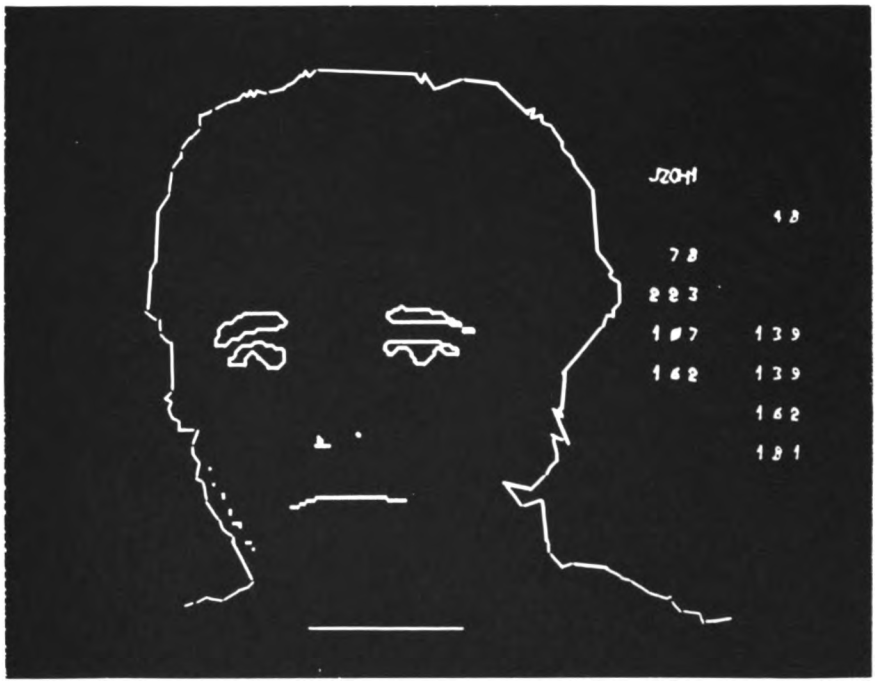
Digitized by Google

Digitized by Google



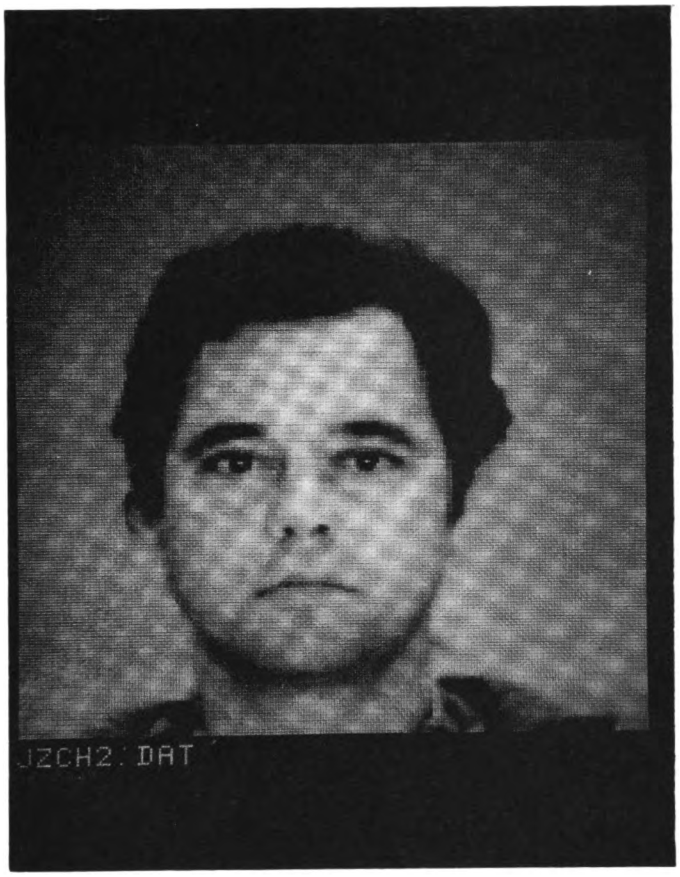
(a)

Figure 10-18



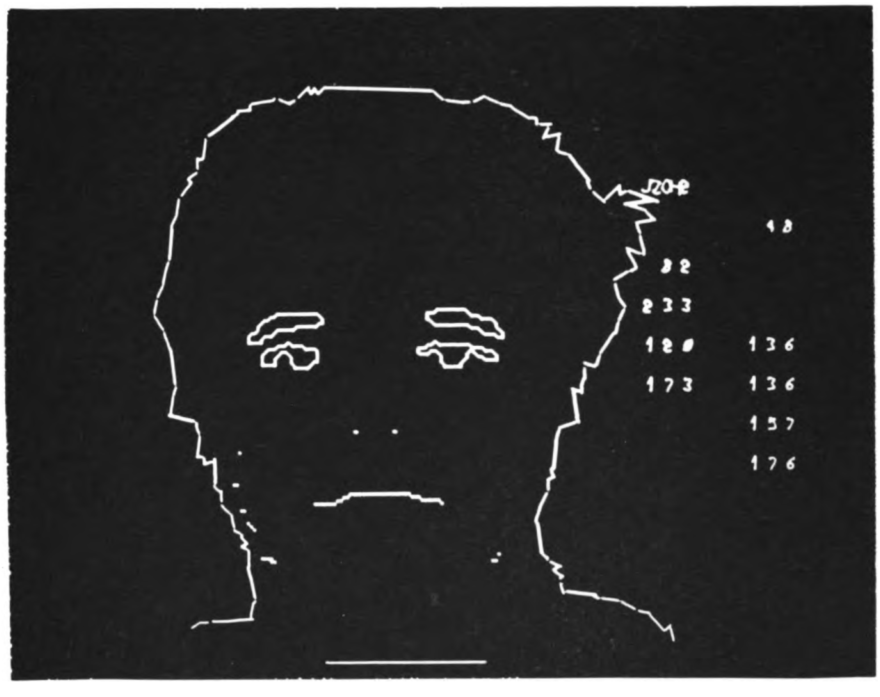
(b)

Digitized by Google



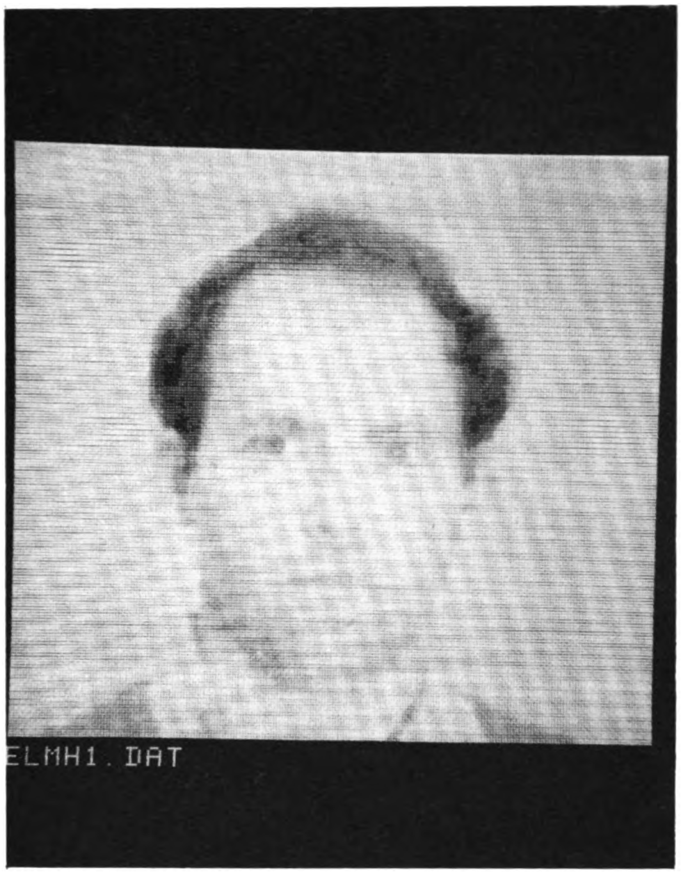
(a)

Figure 10-19



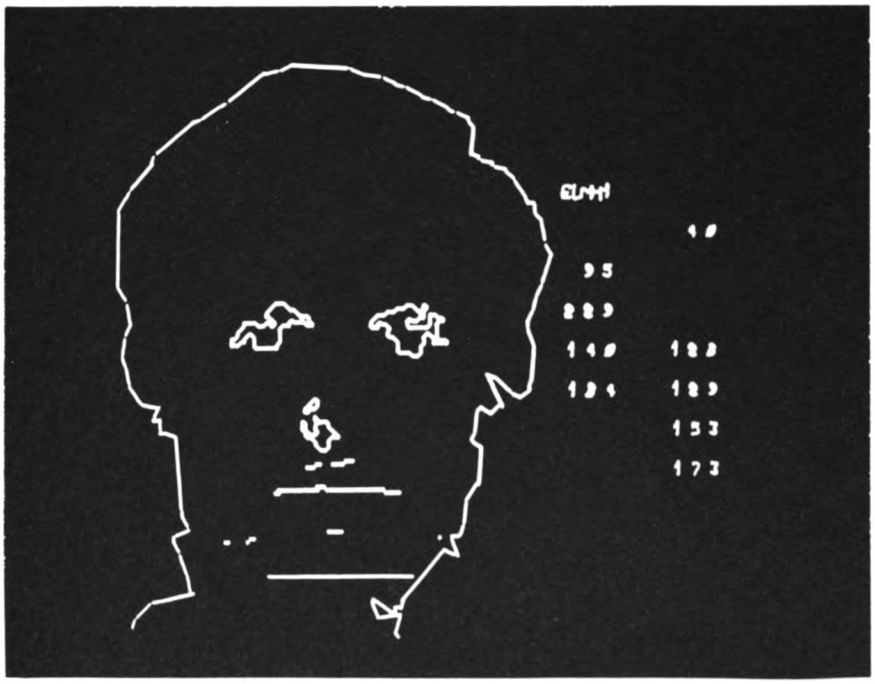
(b)

Digitized by Google



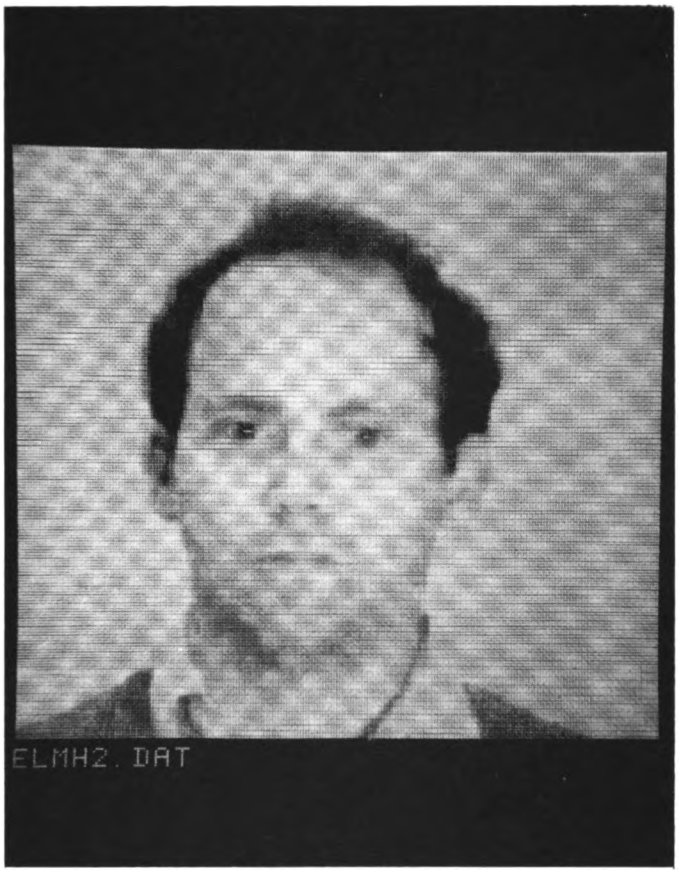
(a)

Figure 10-20



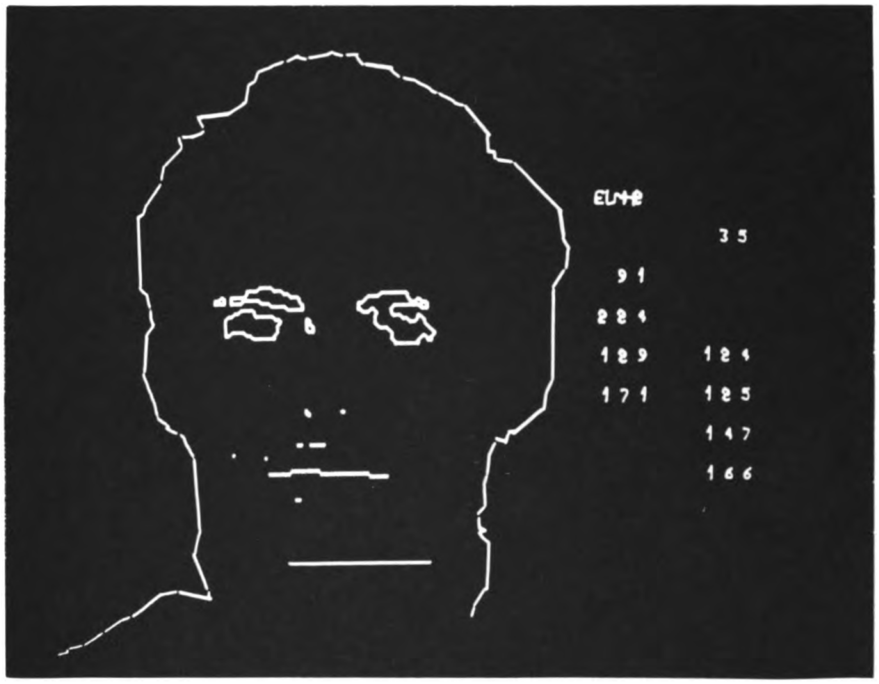
(b)

Digitized by Google



(a)

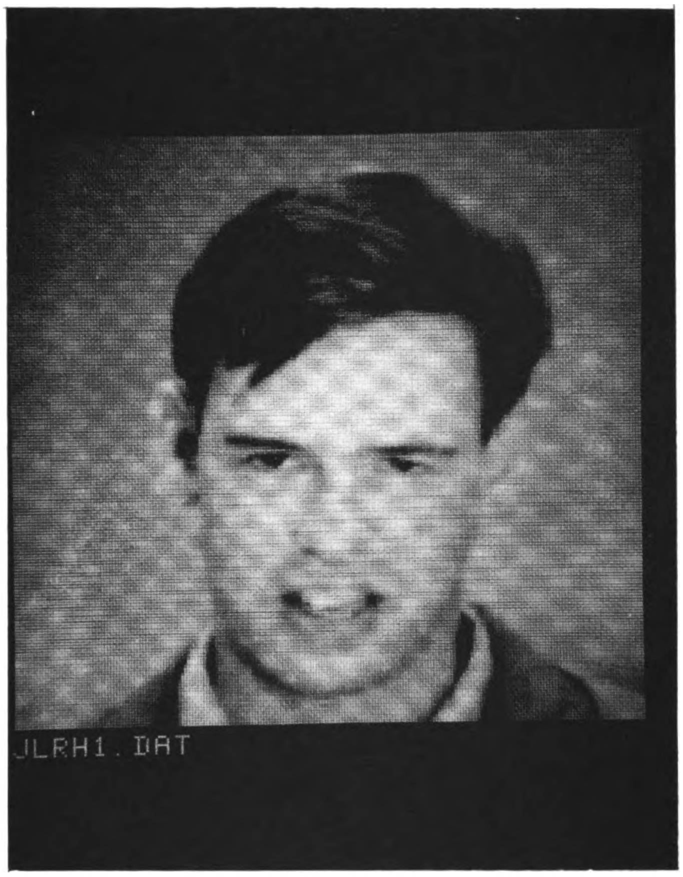
Figure 10-21



(b)

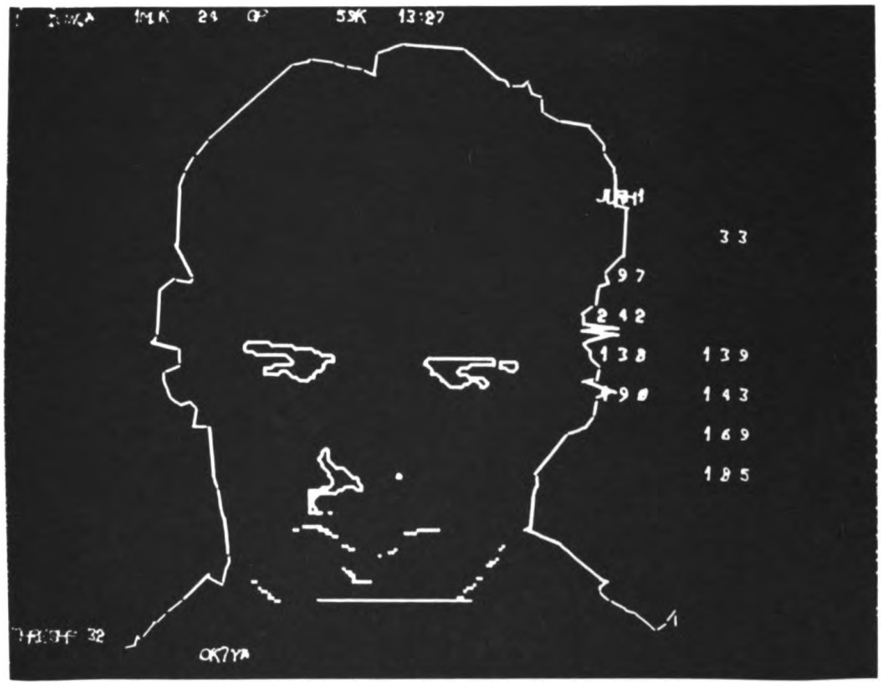
Digitized by Google

Digitized by Google



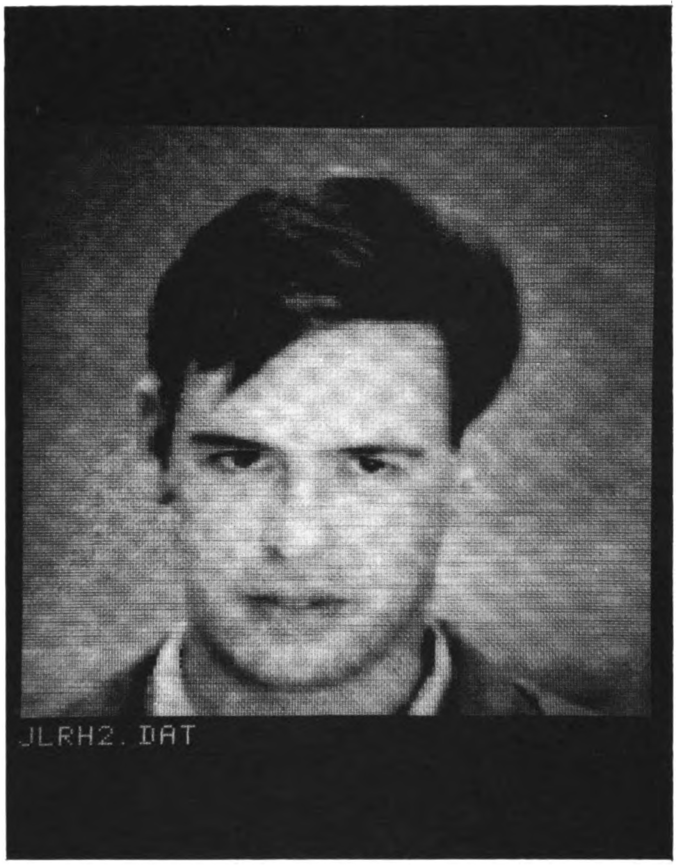
(a)

Figure 10-22



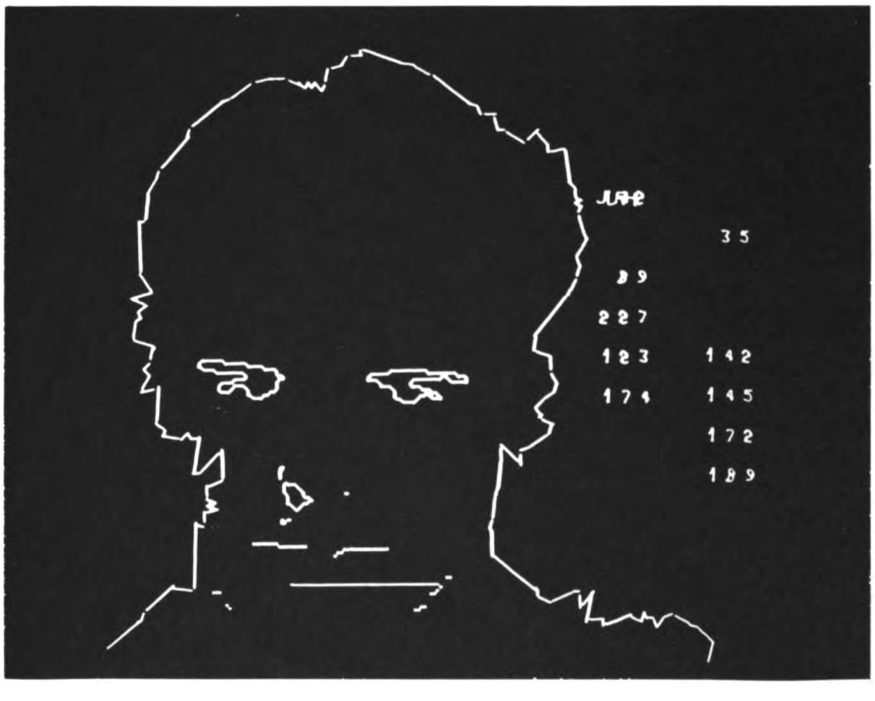
(b)

Digitized by Google



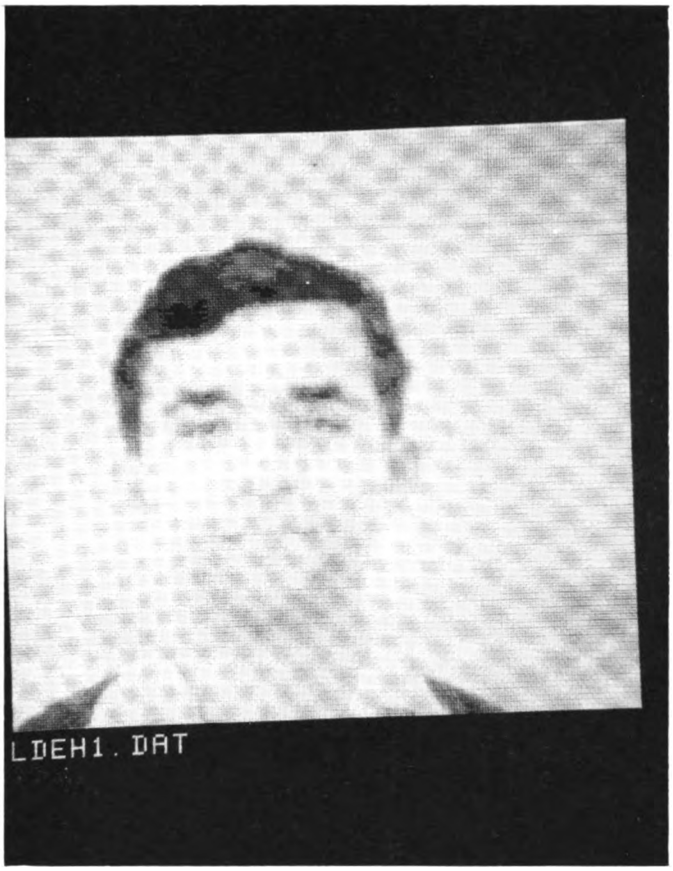
(a)

Figure 10-23



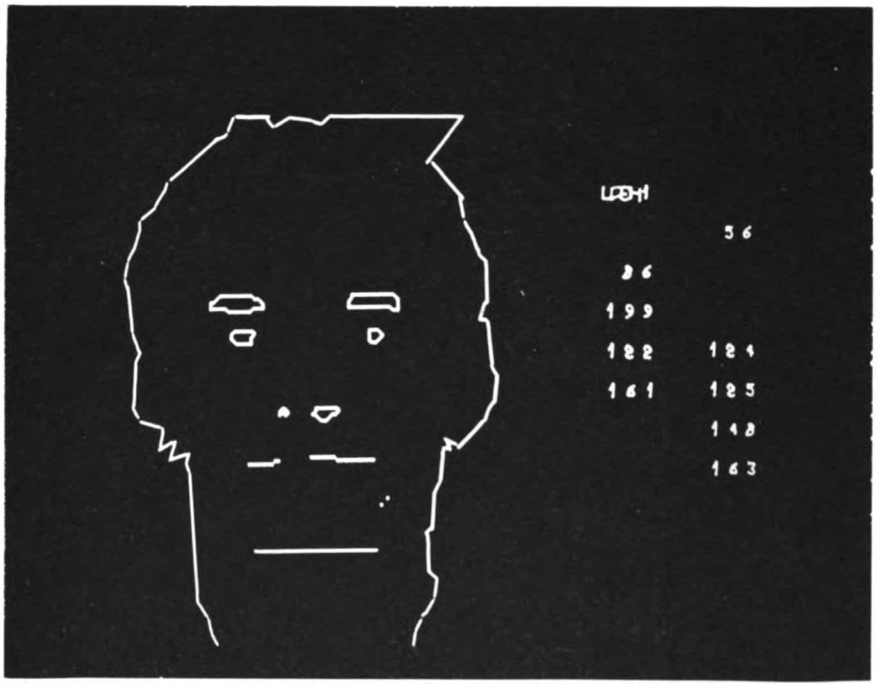
(b)

Digitized by Google



(a)

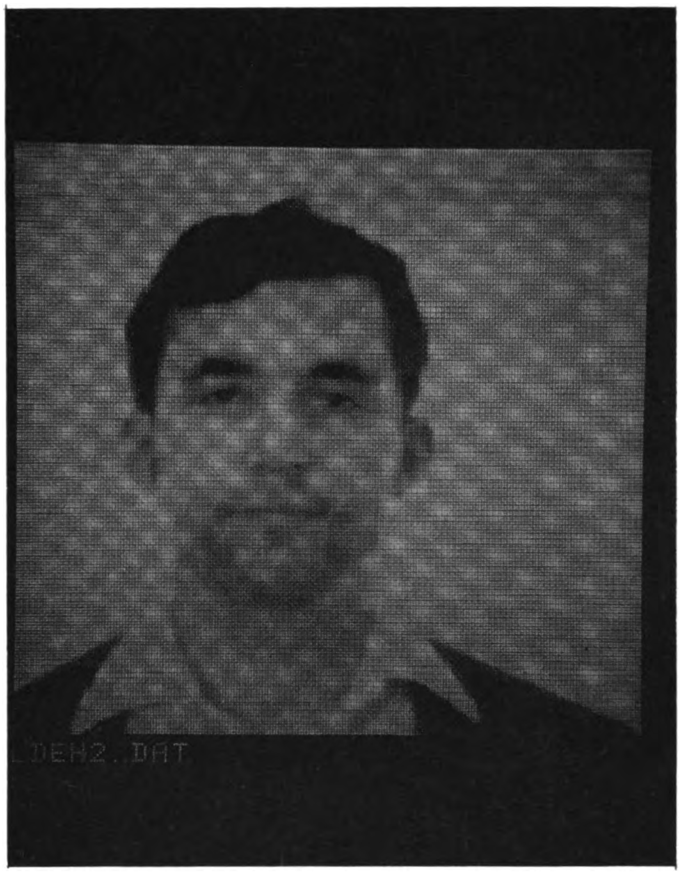
Figure 10-24



(b)

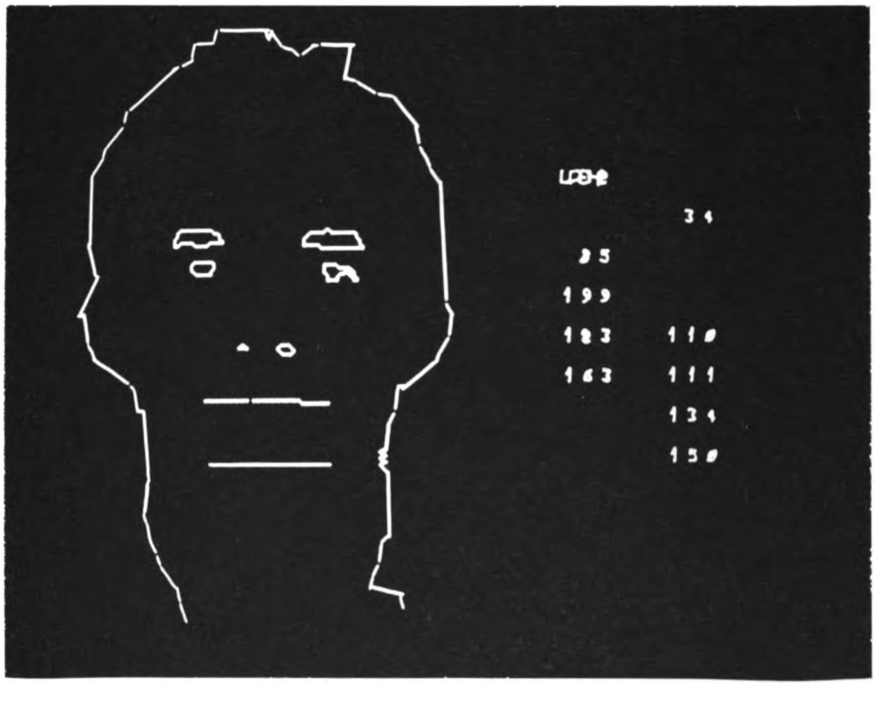
Digitized by Google

Digitized by Google



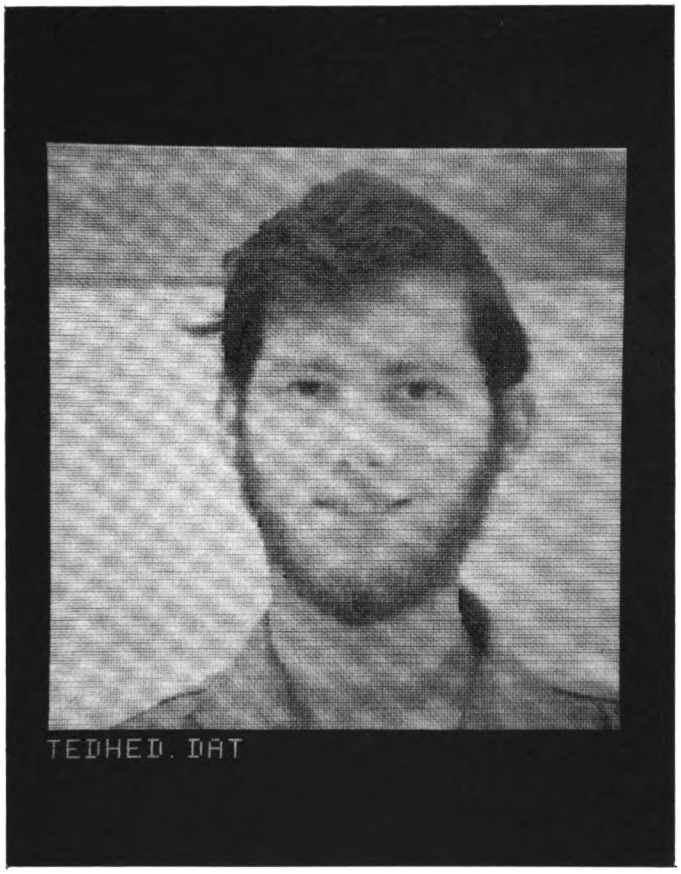
(a)

Figure 10-25



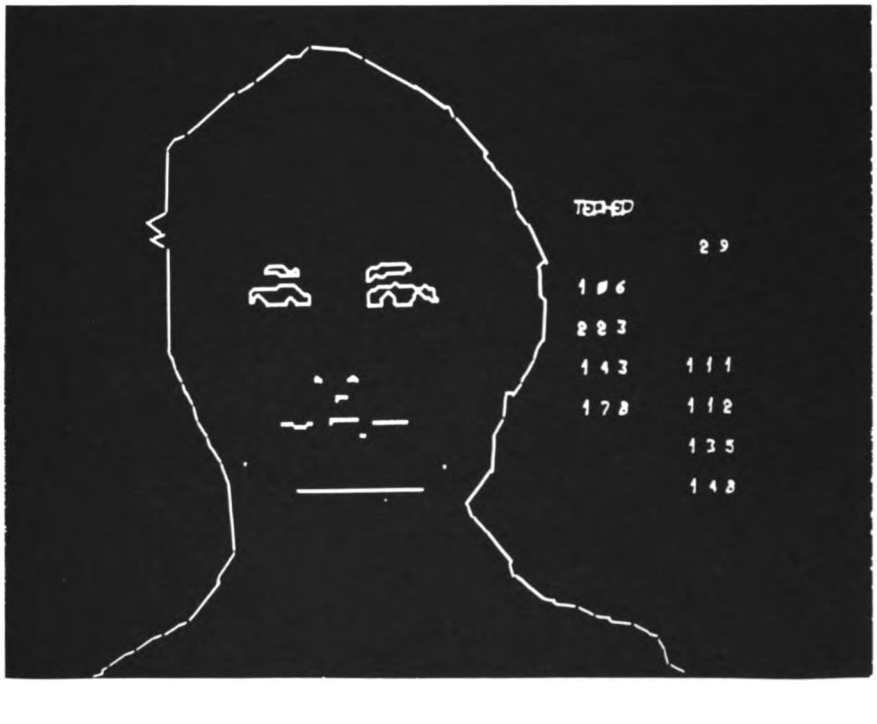
(b)

Digitized by Google



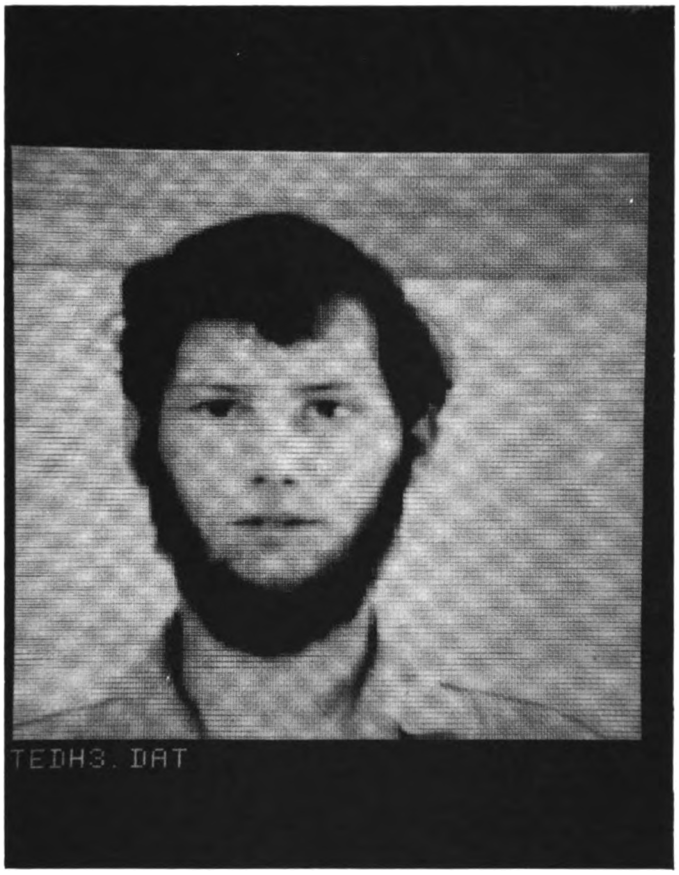
(a)

Figure 10-26



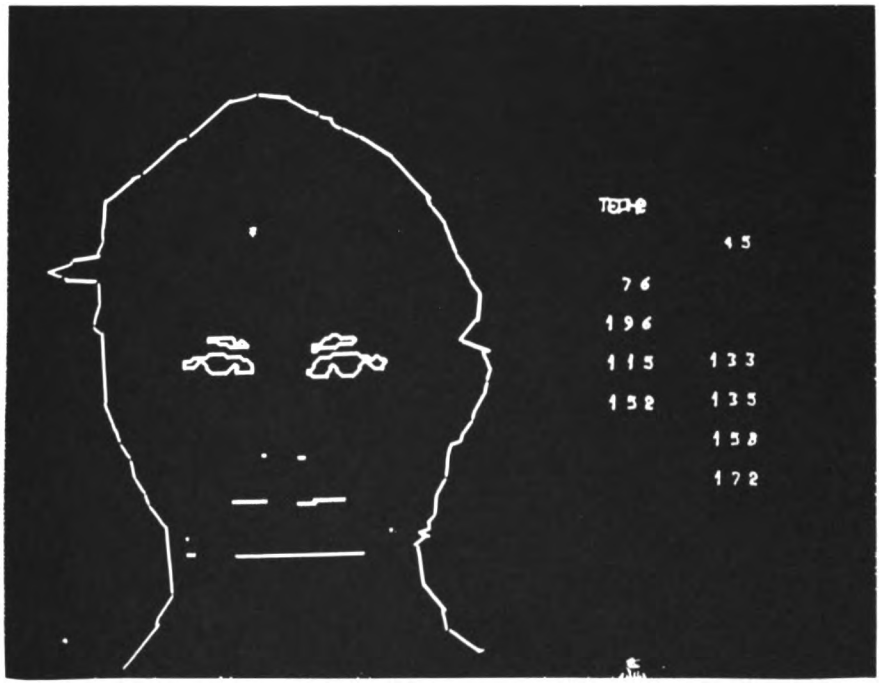
(b)

Digitized by Google



(a)

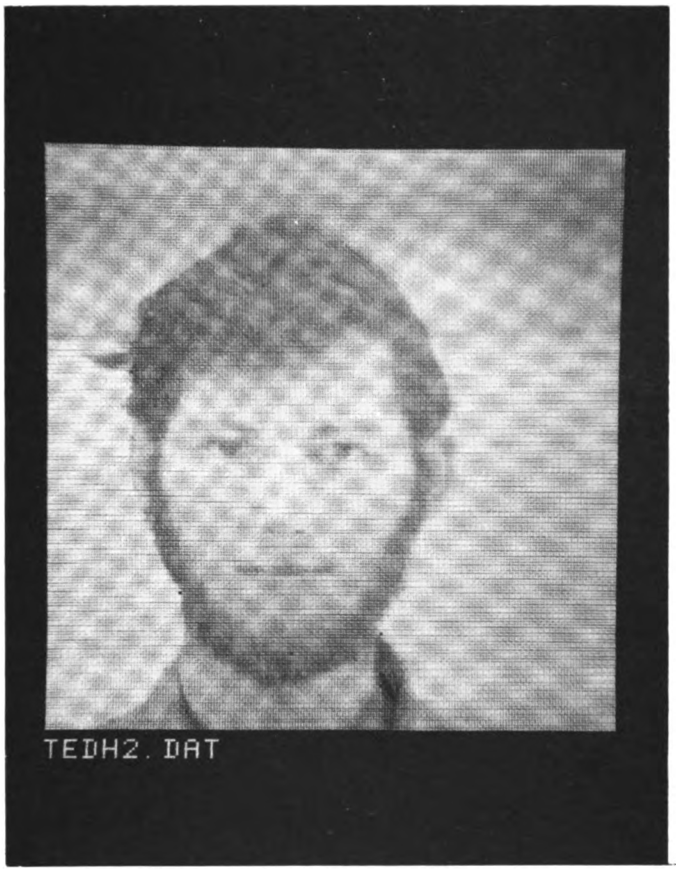
Figure 10-27



(b)

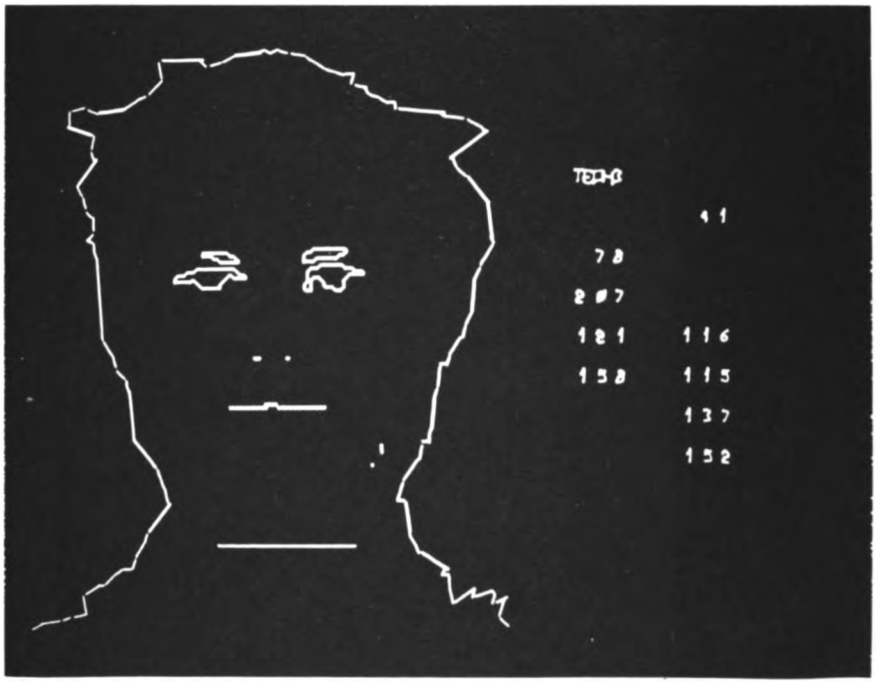
Digitized by Google

Digitized by Google



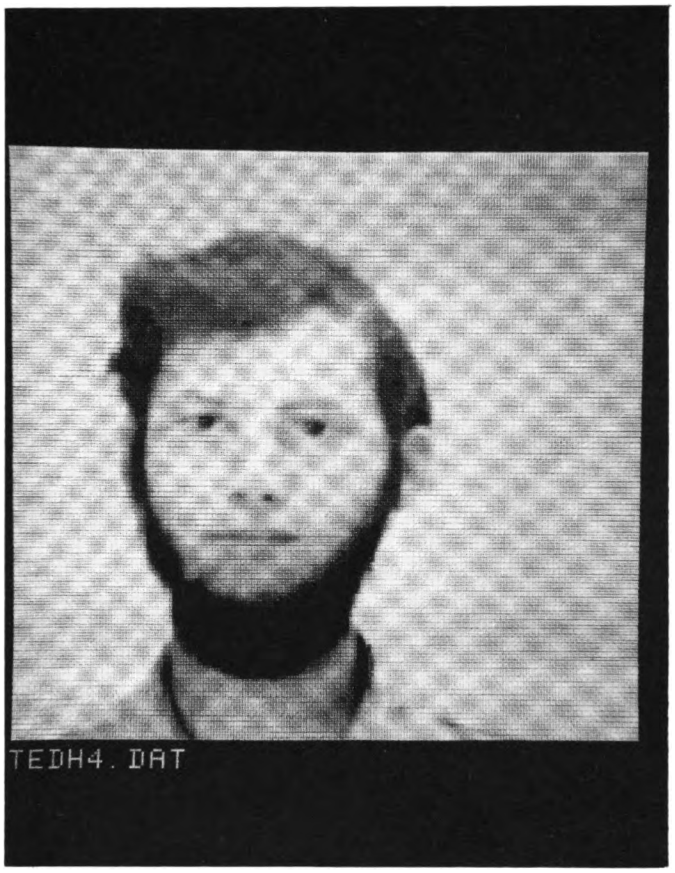
(a)

Figure 10-28



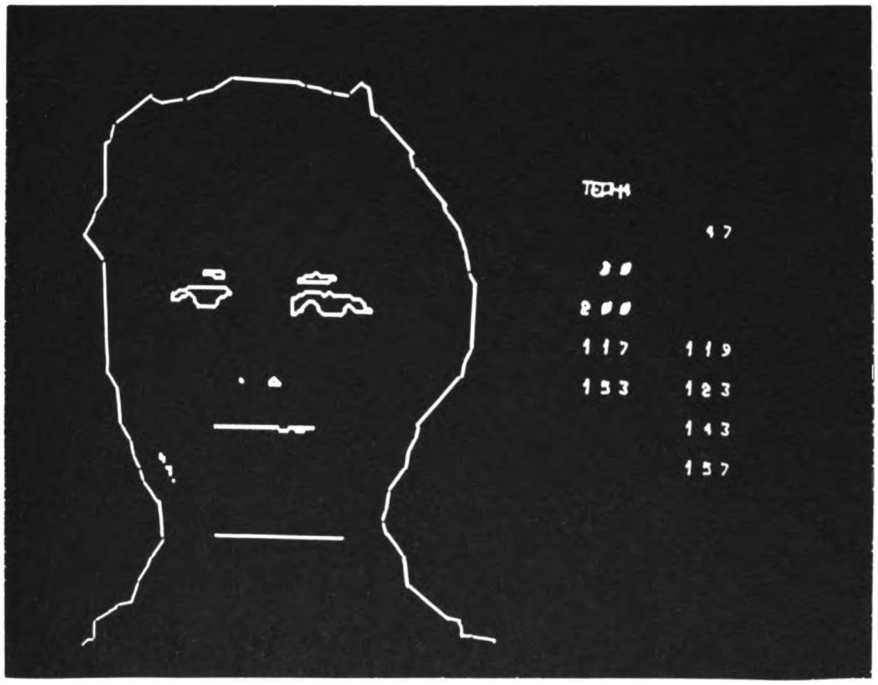
(b)

Digitized by Google



(a)

Figure 10-29



(b)

Digitized by Google

Digitized by Google

which features.

The test of identification of people was performed as follows. First, the complete collection of pictures was processed, and all of the measurements used for recognition were recorded. This produced 24 sets of measurements describing the 10 people. These measurements are given in Table 10-2. Recognition was attempted on these sets of measurements.

All of the measurements were normalized to compensate for the varying distances of the subjects from the camera as described earlier. This normalization was performed by dividing all of the measurements in each picture by the height measured in that picture. Normalized measurements are given in Table 10-3.

Weights for use in the classification procedure were obtained as described in chapter 9. These weights are listed in Table 10-4.

For each set of measurements, the remainder of the collection of measurements was considered to be a dictionary of known individuals. Thus, the following experiment was repeated 24 times. An unknown measurement vector was selected, and an attempt was made to identify the person to whom these measurements belonged from the dictionary which contained 23 entries.

In this recognition experiment only two errors were made. The correct individual was identified in 91% of the

Digitized by Google

Digitized by Google

TABLE 10-2.

Measurements for identification test.

Each entry consists of a 3 letter code for the individual followed by an identifying letter or digit. The ten measurements which follow are the ten measurements listed in Figures 2-1 and 2-2.

ELM1					
185.	21.0	14.0	52.0	46.0	134.
44.0	88.0	25.0	45.0		
ELM2					
185.	20.0	14.0	51.0	38.0	133.
42.0	89.0	23.0	42.0		
GJG1					
184.	21.0	16.0	59.1	41.0	137.
43.0	88.0	25.0	44.0		
GJG2					
182.	24.0	15.0	41.0	40.0	135.
41.0	88.0	24.0	42.0		
JLR1					
195.	23.0	14.0	58.0	42.0	145.
52.2	108.	28.0	44.0		
JLR2					
194.	23.0	16.0	57.0	42.0	138.
51.1	108.	29.0	46.0		
JZC1					
189.	23.0	15.0	52.0	41.0	145.
55.0	91.0	23.0	42.0		
JZC2					
190.	23.0	15.0	55.0	40.0	151.
53.0	88.0	21.0	40.0		
KEL1					
168.	19.0	11.0	44.0	36.0	115.
40.0	75.0	21.0	38.0		
KEL4					
173.	21.0	11.0	63.0	44.0	119.
40.0	69.0	19.0	35.0		
KEL5					
172.	20.0	11.0	47.0	43.0	120.
42.0	69.0	20.0	36.0		
LDE1					
154.	17.0	10.0	47.0	50.0	113.
39.0	68.0	24.0	39.0		
LDE2					
156.	17.0	11.0	47.0	45.0	114.
40.0	76.0	24.0	40.0		

Digitized by Google

Digitized by Google

TABLE 10-2. (Continued)

Measurements for identification test.

RAJ1					
174.	18.0	13.0	50.0	27.0	114.
38.3	68.0	22.0	26.0		
RAJ2					
177.	18.0	12.0	50.0	28.0	120.
39.2	79.0	19.0	25.0		
RGG7					
183.	22.0	16.0	45.0	39.0	129.
46.0	80.0	18.0	36.0		
RGGB					
190.	21.0	16.0	58.0	40.0	135.
44.0	86.0	20.0	40.0		
RGGC					
190.	21.0	15.0	55.0	40.0	132.
48.0	82.0	18.0	41.0		
RPH2					
174.	20.0	13.0	46.0	47.0	116.
39.1	73.0	14.0	30.0		
RPH3					
172.	19.0	14.0	52.0	48.0	112.
41.0	71.0	15.0	30.0		
TED1					
162.	20.0	13.0	42.0	34.0	117.
35.0	82.0	24.0	37.0		
TED2					
163.	20.0	13.0	43.1	33.1	120.
37.1	89.0	24.0	38.0		
TED3					
163.	19.0	13.0	39.0	50.0	129.
37.0	74.0	22.0	37.0		
TED5					
154.	18.0	12.0	39.0	31.0	120.
36.2	74.0	22.0	36.0		

Digitized by Google

Digitized by Google

TABLE 1Ø-3,

Normalized measurements,

The format is the same as Table 1Ø-2,

```
 ELM1
  1,ØØ       Ø,114       Ø,757E-Ø1 Ø,281       Ø,249       Ø,724
 Ø,238       Ø,476       Ø,135     Ø,243
 ELM2
  1,ØØ       Ø,1Ø8       Ø,757E-Ø1 Ø,276       Ø,2Ø5       Ø,719
 Ø,227       Ø,481       Ø,124     Ø,227
 GJG1
  1,ØØ       Ø,114       Ø,87ØE-Ø1 Ø,321       Ø,223       Ø,745
 Ø,234       Ø,478       Ø,136     Ø,239
 GJG2
  1,ØØ       Ø,132       Ø,824E-Ø1 Ø,225       Ø,22Ø       Ø,742
 Ø,225       Ø,484       Ø,132     Ø,231
 JLR1
  1,ØØ       Ø,118       Ø,718E-Ø1 Ø,297       Ø,215       Ø,744
 Ø,268       Ø,554       Ø,144     Ø,226
 JLR2
  1,ØØ       Ø,119       Ø,825E-Ø1 Ø,294       Ø,216       Ø,711
 Ø,263       Ø,557       Ø,149     Ø,237
 JZC1
  1,ØØ       Ø,122       Ø,794E-Ø1 Ø,275       Ø,217       Ø,767
 Ø,291       Ø,481       Ø,122     Ø,222
 JZC2
  1,ØØ       Ø,121       Ø,789E-Ø1 Ø,289       Ø,211       Ø,795
 Ø,279       Ø,463       Ø,111     Ø,211
 KEL1
  1,ØØ       Ø,113       Ø,655E-Ø1 Ø,262       Ø,214       Ø,685
 Ø,238       Ø,446       Ø,125     Ø,226
 KEL4
  1,ØØ       Ø,121       Ø,636E-Ø1 Ø,364       Ø,254       Ø,688
 Ø,231       Ø,399       Ø,11Ø     Ø,2Ø2
 KEL5
  1,ØØ       Ø,116       Ø,64ØE-Ø1 Ø,273       Ø,25Ø       Ø,698
 Ø,244       Ø,4Ø1       Ø,116     Ø,2Ø9
 LDE1
  1,ØØ       Ø,11Ø       Ø,649E-Ø1 Ø,3Ø5       Ø,325       Ø,734
 Ø,253       Ø,442       Ø,156     Ø,253
 LDE2
  1,ØØ       Ø,1Ø9       Ø,7Ø5E-Ø1 Ø,3Ø1       Ø,288       Ø,731
 Ø,256       Ø,487       Ø,154     Ø,256
```

Digitized by Google

Digitized by Google

TABLE 1Ø-3. (Continued)

Normalized measurements.

```
RAJ1
 1.ØØ       Ø.1Ø3       Ø.747E-Ø1 Ø.287       Ø.155       Ø.655
Ø.22Ø       Ø.391       Ø.126     Ø.149
RAJ2
 1.ØØ       Ø.1Ø2       Ø.678E-Ø1 Ø.282       Ø.158       Ø.678
Ø.221       Ø.446       Ø.1Ø7     Ø.141
RGG7
 1.ØØ       Ø.12Ø       Ø.874E-Ø1 Ø.246       Ø.213       Ø.7Ø5
Ø.251       Ø.437       Ø.984E-Ø1 Ø.197
RGGB
 1.ØØ       Ø.111       Ø.842E-Ø1 Ø.3Ø5       Ø.211       Ø.711
Ø.232       Ø.453       Ø.1Ø5     Ø.211
RGGC
 1.ØØ       Ø.111       Ø.789E-Ø1 Ø.289       Ø.211       Ø.695
Ø.253       Ø.432       Ø.947E-Ø1 Ø.216
RPH2
 1.ØØ       Ø.115       Ø.747E-Ø1 Ø.264       Ø.27Ø       Ø.667
Ø.225       Ø.42Ø       Ø.8Ø5E-Ø1 Ø.172
RPH3
 1.ØØ       Ø.11Ø       Ø.814E-Ø1 Ø.3Ø2       Ø.279       Ø.651
Ø.238       Ø.413       Ø.872E-Ø1 Ø.174
TED1
 1.ØØ       Ø.123       Ø.8Ø2E-Ø1 Ø.259       Ø.21Ø       Ø.722
Ø.216       Ø.5Ø6       Ø.148     Ø.228
TED2
 1.ØØ       Ø.123       Ø.798E-Ø1 Ø.264       Ø.2Ø3       Ø.736
Ø.228       Ø.546       Ø.147     Ø.233
TED3
 1.ØØ       Ø.117       Ø.798E-Ø1 Ø.239       Ø.3Ø7       Ø.791
Ø.227       Ø.454       Ø.135     Ø.227
TED5
 1.ØØ       Ø.117       Ø.779E-Ø1 Ø.253       Ø.2Ø1       Ø.779
Ø.235       Ø.481       Ø.143     Ø.234
```

Digitized by Google

Digitized by Google

TABLE 10-4. Weights.

Measurement	σ^2	σ
From body picture:		
Head width	0.0000402	0.0063405
Neck width	0.0000197	0.0044354
Shoulder width	0.0019402	0.0440476
Hip width	0.0019666	0.0443458
From head picture:		
Head width	0.0007326	0.0270665
Distance between eyes	0.0001170	0.0108170
Top of head to eyes	0.0014393	0.0379384
Eyes to nose	0.0000663	0.0081423
Eyes to mouth	0.0001208	0.0109927

Digitized by Google

Digitized by Google

cases involved. The distance in the recognition space between each pair of measurements was tabulated and is given in Table 10-5. This test indicates that the identification of people is possible relying solely on measurements obtained by a computer.

The two errors in the recognition tests occurred when one individual was misidentified twice. The reason for this mistake was faulty measurements of the width of the shoulders and neck in the picture of the body. The faulty measurements were caused by low contrast with the background.

An example of this sort of problem can be seen in Figure 10-1(a) where there is very little contrast between the subjects shirt and the window behind him.

FUTURE WORK.

The recognition procedures described in this thesis are elaborate and complex. It would be worthwhile to examine these procedures in detail and attempt to determine the key heuristics on which success or failure of the algorithms really depends. In isolation they could be examined more effectively to determine the range of their usefulness and applicability. Statistical data could be obtained on the success rates of particular heuristics. This could then be used to obtain meaningful evaluations. Unfortunately, such testing and evaluation is a problem which is comparable. In

Digitized by Google

Digitized by Google

TABLE 10-5. RECOGNITION DISTANCES.

	ELM1	ELM2	GJG1	GJG2	JLR1	JLR2	JZC1	JZC2	KEL1	KEL4	KEL5	LDE1
ELM1	0.0	6.7	8.5	16.0	18.0	17.4	36.0	42.0	12.8	42.4	27.3	19.6
ELM2	6.7	0.0	13.1	19.5	27.7	31.0	43.8	41.1	9.5	31.5	20.8	41.6
GJG1	8.5	13.1	0.0	15.1	28.6	18.1	39.6	42.6	34.3	60.8	49.8	42.4
GJG2	16.0	19.5	15.1	0.0	34.4	29.3	44.3	44.7	31.8	54.6	42.9	56.7
JLR1	18.0	27.7	28.6	34.4	0.0	9.0	19.9	31.6	28.7	60.8	41.4	29.2
JLR2	17.4	31.0	18.1	29.3	9.0	0.0	29.1	47.2	40.9	82.3	61.6	36.8
JZC1	36.0	43.8	39.6	44.3	19.9	29.1	0.0	5.7	46.1	66.7	45.0	60.5
JZC2	42.0	41.1	42.6	44.7	31.6	47.2	5.7	0.0	47.4	54.3	39.3	76.7
KEL1	12.8	9.5	34.3	31.8	28.7	40.9	46.1	47.4	0.0	18.3	6.6	33.1
KEL4	42.4	31.5	60.8	54.6	60.8	82.3	66.7	54.3	18.3	0.0	7.5	69.1
KEL5	27.3	20.8	49.8	42.9	41.4	61.6	45.0	39.3	6.6	7.5	0.0	47.5
LDE1	19.6	41.6	42.4	56.7	29.2	36.8	60.5	76.7	33.1	69.1	47.5	0.0
LDE2	12.7	33.1	28.9	46.8	18.6	19.9	48.3	66.5	32.4	75.7	52.2	4.0
RAJ1	95.3	63.5	99.1	98.9	108.6	121.8	121.4	104.7	63.7	52.2	53.4	142.7
RAJ2	114.4	74.0	125.0	119.2	120.1	147.7	129.0	104.0	72.2	52.2	56.3	168.9
RGG7	50.8	35.6	46.0	40.6	65.5	66.7	37.5	26.4	45.9	44.1	37.0	113.7
RGGB	28.2	12.9	23.9	31.2	53.0	53.4	45.1	33.6	28.2	30.9	28.2	84.0
RGGC	36.8	23.2	42.2	48.1	56.4	63.5	36.0	27.0	26.6	28.4	21.8	86.8
RPH2	95.3	63.5	105.0	90.8	122.7	140.4	103.9	78.5	62.3	34.3	41.6	161.0
RPH3	86.5	59.2	89.1	88.8	110.6	119.4	87.9	68.5	62.0	37.9	43.3	149.1
TED1	13.6	17.1	13.6	8.3	30.5	23.2	62.3	67.0	30.5	60.2	49.3	44.9
TED2	11.7	17.8	12.2	9.8	18.9	13.4	49.6	57.0	32.0	67.1	53.0	41.0
TED3	13.4	17.9	14.8	14.4	34.6	37.6	44.6	40.9	33.5	54.8	39.2	37.3
TED5	7.9	13.5	9.6	11.8	18.2	20.0	35.9	38.4	26.8	59.5	41.1	31.1

	LDE2	RAJ1	RAJ2	RGG7	RGGB	RGGC	RPH2	RPH3	TED1	TED2	TED3	TED5
ELM1	12.7	95.3	114.4	50.8	28.2	36.8	95.3	86.5	13.6	11.7	13.4	7.9
ELM2	33.1	63.5	74.0	35.6	12.9	23.2	63.5	59.2	17.1	17.8	17.9	13.5
GJG1	28.9	99.1	125.0	46.0	23.9	42.2	105.0	89.1	13.6	12.2	14.8	9.6
GJG2	46.8	98.9	119.2	40.6	31.2	48.1	90.8	88.8	8.3	9.8	14.4	11.8
JLR1	18.6	108.6	120.1	65.5	53.0	56.4	122.7	110.6	30.5	18.9	34.6	18.2
JLR2	19.9	121.8	147.7	66.7	53.4	63.5	140.4	119.4	23.2	13.4	37.6	20.0
JZC1	48.3	121.4	129.0	37.5	45.1	36.0	103.9	87.9	62.3	49.6	44.6	35.9
JZC2	66.5	104.7	104.0	26.4	33.6	27.0	78.5	68.5	67.0	57.0	40.9	38.4
KEL1	32.4	63.7	72.2	45.9	28.2	26.6	62.3	62.0	30.5	32.0	33.5	26.8
KEL4	75.7	52.2	52.2	44.1	30.9	28.4	34.3	37.9	60.2	67.1	54.8	59.5
KEL5	52.2	53.4	56.3	37.0	28.2	21.8	41.6	43.3	49.3	53.0	39.2	41.1
LDE1	4.0	142.7	168.9	113.7	84.0	86.8	161.0	149.1	44.9	41.0	37.3	31.1
LDE2	0.0	142.4	168.3	100.9	72.4	77.2	159.7	144.0	35.3	28.1	33.6	22.6
RAJ1	142.4	0.0	11.4	61.4	53.2	67.7	47.5	43.0	87.7	102.7	97.9	98.2
RAJ2	168.3	11.4	0.0	65.7	59.4	66.9	33.0	38.4	114.7	125.2	115.2	118.6
RGG7	100.9	61.4	65.7	0.0	10.5	10.4	28.7	19.9	63.0	64.8	51.1	57.3
RGGB	72.4	53.2	59.4	10.5	0.0	7.9	32.8	24.8	40.7	43.5	33.5	37.3
RGGC	77.2	67.7	66.9	10.4	7.9	0.0	30.0	22.4	65.4	64.9	51.1	53.4
RPH2	159.7	47.5	33.0	28.7	32.8	30.0	0.0	6.2	110.3	120.6	93.9	113.8
RPH3	144.0	43.0	38.4	19.9	24.8	22.4	6.2	0.0	105.0	113.6	89.9	107.5
TED1	35.3	87.7	114.7	63.0	40.7	65.4	110.3	105.0	0.0	2.8	18.3	10.0
TED2	28.1	102.7	125.2	64.8	43.5	64.9	120.6	113.6	2.8	0.0	19.4	7.4
TED3	33.6	97.9	115.2	51.1	33.5	51.1	93.9	89.9	18.3	19.4	0.0	8.5
TED5	22.6	98.2	118.6	57.3	37.3	53.4	113.8	107.5	10.0	7.4	8.5	0.0

Digitized by Google

Digitized by Google

the effort required, to the original design and development of the system.

The work reported in this thesis could be extended to deal with larger sets of people. One way in which this could be done would be to use absolute measurements of people rather than measurements which have been normalized by height. Absolute measurements can be obtained from the pictures in a straight-forward way if the distance from camera to subject is known. Alternatively, if the geometry of the camera position is known, the distance from the camera to the individual may be calculated using the ground plane assumption (Sobel [1970]). In the pictures that were processed, the height of the people could be correctly measured to within 2 raster units. This corresponds to about a half of an inch when considered as a fraction of a person's height. Thus, a preliminary reduction in the fraction of the dictionary to be searched could be done using the absolute height of the person. The rest of the measurements obtained from the pictures could be used to distinguish people of almost the same height. In this way the number of people that a recognition system could handle could be greatly expanded.

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 that has been described.

Digitized by Google

Digitized by Google

Possible lines of approach could be found in the attempts at linguistic description of pictures and formalization of models which were discussed in Chapter 4. Such a high-level specification, if successful in reproducing the effect of the algorithms so far described, could be applied to other areas of picture processing to test its generality.

It would be useful to attempt to overcome the obstacles that prevented this program from performing on-line recognition. What is needed is a picture input system with the ability to digitize a wider range of light intensities. Also needed in an effective on-line system is a quick-response picture output device for digital pictures that can provide visual feedback to the operator while setting input parameters.

It is clear that this heuristic program is nowhere near the final solution to the problem of visual identification of people. Nevertheless, the success that has been achieved indicates that the methods used are worthwhile. The research which has been done here provides useful procedures for further work in processing very complex scenes.

Digitized by Google

Digitized by Google

References

Abbamonte, M., Johnston, E.G., Lee, Y.H., Nagel, R., Rosenfeld, A., and Thurston, M. [1970] Edge and curve enhancement in digital pictures, 2. Report 70-103, Computer Science Center, University of Maryland, College Park, Maryland (January).

Bell, D.A. [1968] Computer aided design of image processing techniques. Proc. of I.E.E./N.P.L. Conference on Pattern Recognition, Institute of Electrical Engineers, London.

Billingsley, F.C. [1967] Digital image processing at JPL. Proc. Computer Imaging Techniques Seminar, Society of Photo-optical Instrumentation Engineers, Washington, D.C.

Bisson, C.L. [1965a] Preliminary investigation on measurements by computer of the distances on and about the eyes. Report PRI:18, Panoramic Research Inc., Palo Alto, California (April).

Bisson, C.L. [1965b) Location of some facial features by computer. Report PRI:20, Panoramic Research Inc., Palo Alto, California (June).

Bledsoe, W.W. [1964] The model method in facial recognition. Report PRI:15, Panoramic Research Inc., Palo Alto, California (August).

Bledsoe, W.W. [1966] Man-machine facial recognition. Report PRI:22, Panoramic Research Inc., Palo Alto, California (August).

British Computer Society. [1967] Character Recognition 1967. British Computer Society, London.

Clowes, M.B. [1969] Pictorial relationships - a syntactic approach. In Machine Intelligence 4, Meltzer, B. and Michie, D. (Ed.), American Elsevier, New York, pp. 361-383.

Cover, T.M. and Hart, P.E. [1967] Nearest neighbor pattern classification. IEEE Transactions on Information Theory IT-13 (January), pp. 21-27.

El'bur, R.E. [1967] Utilization of the apparatus of projective geometry in the process of the identification of individuals by their photographs. In Problems of

Digitized by Google

Cybernetics and Law, Kudryavtsev, V.N. (Ed.), Nauka Publishing House, Moscow, pp. 321-348. Available through U.S. Department of Commerce, Joint Publications Research Service, No. JPRS: 43,954 (January 10, 1968).

Falk, G. [1969] Some implications of planarity for machine perception. Artificial Intelligence Memo No. 107, Stanford University, Stanford, California (December).

Feigenbaum, E.A. and Feldman, J. (Ed.) [1963] Computers and Thought. McGraw-Hill, New York.

Feldman, J.A., Feldman, G.M., Falk, G., Grape, G., Pearlman, J., Sobel, I., Tenenbaum, J.M. [1969] The Stanford hand-eye project. Proc. International Joint Conference on Artificial Intelligence (May), pp. 521-526.

Feldman, J.A. and Gries, D. [1968] Translator writing systems. Communications of the ACM 11 (February), pp. 77-113.

Forsen, G.E. [1968] Processing visual data with an automaton eye. In Pictorial Pattern Recognition, Cheng, G.C. et al (Ed.), Thompson Book Co., Washington, D.C.

Greenblatt, R.D., Eastlake, D.E., Crocker, S.D. [1967] The Greenblatt chess program. Proc. AFIPS 1967 Fall Joint Computer Conference, Thompson Book Co., Washington, D.C., pp.801-810.

Greenblatt, R.D., Holloway, J., Sordillo, D.A. [1966] Sides 21. Artif. Intell. Memo No. 104-2 (MAC-M-320), M.I.T., Cambridge, Mass. (Aug).

Guzman, A. [1968] Decomposition of a visual scene into three-dimensional bodies. Proc. AFIPS 1968 Fall Joint Computer Conference, Vol. 33, Thompson Book Co., Washington, D.C., pp. 291-304.

Hart, P.E. [1969] Three object recognition methods using projective invariants. Report, Stanford Research Institute, Menlo Park, California (January).

Hueckel, M. [1969] An operator which locates edges in digitized pictures. Artificial Intelligence Memo No. 105, Stanford Univ., Stanford, Calif. (October).

Ho, Y. and Agrawala, A.K. [1968] On pattern classification algorithms, introduction and survey. Proc. IEEE 56 (December), pp. 2101-2114.

Kelly, M.D. [1970] Edge detection in pictures by computer

Digitized by Google

using planning. Artificial Intelligence Memo No. 108, Stanford University, Stanford, California (January).

Kirsch, R.A., Cahn, L., Ray, C., and Urban, G.H. [1957] Experiments in processing pictorial information with a digital computer. Proc. 1957 Eastern Joint Computer Conference, Spartan Books, New York, pp. 221-229.

Lederberg, J. and Feigenbaum, E.A. [1968] Mechanization of inductive inference in organic chemistry. In Formal Representation for Human Judgement, Kleinmuntz, B. (Ed.), Wiley, New York.

Ledley, R.S., Rotolo, L.S., Golab, T.J., Jacobsen, J.D., Ginsberg, M.D., and Wilson, J.B. [1965] FIDAC: film input to digital automatic computer and associated syntax-directed pattern recognition programming system. In Optical and Electro-Optical Information Processing, Tippett, J.T. et al (Ed.), MIT Press, Cambridge, Mass.

Lipkin, L.E., Watt, W.C., and Kirsch, R.A. [1966] The analysis, synthesis, and description of biological images. Annals of the New York Academy of Sciences 128 (January), pp. 984-1012.

Lourie, J.R. [1969] The computation of connected regions in interactive graphics. Proc. 1969 A.C.M. National Conference, Association for Computing Machinery, New York, pp. 369-377.

McCarthy, J., Earnest, L.D., Reddy, D.R., and Vicens, P.J. [1968] A computer with hands, eyes, and ears. Proc. AFIPS 1968 Fall Joint Computer Conference, Vol. 33, Thompson Book Co., Washington, D.C., pp. 329-338.

Miller, W.F. and Shaw, A.C. [1967] A picture calculus. GSG Memo 40, Computation Group, Stanford Linear Accelerator Center, Stanford, California (June).

Miller, W.F. and Shaw, A.C. [1968] Linguistic methods in picture processing: a survey. Proc. AFIPS 1968 Fall Joint Computer Conference, Vol. 33, Thompson Book Co., Washington, D.C., pp. 279-290.

Minsky, M. [1961] Steps toward artificial intelligence. Proc. of the IRE 49 (January), pp. 8-30. Also in Computers and Thought, Feigenbaum, E.A. and Feldman, J. (Ed.), McGraw-Hill, New York, 1963, pp. 406-450.

Minsky, M. and Papert, S. [1969] Perceptrons. MIT Press, Cambridge, Massachusetts.

Digitized by Google

Digitized by Google

Moorer, A. [1969] Stanford A-I Project monitor manual, Artificial Intelligence Operating Notes Nos. 54 and 55, Stanford University, Stanford, California (September).

Nagy, G. [1968] State of the art in pattern recognition. Proc. IEEE 56 (May), pp. 836-862.

Narasimhan, R. [1966] Syntax-directed interpretation of classes of pictures. Communications of the ACM 9 (March), pp. 166-173.

Narasimhan, R. and Forango, J.P. [1964] Some further experiments in the parallel processing of pictures. IEEE Transactions on Eloctronic Computers 13 (December), pp. 748-750.

Newell, A., Shaw, J.C., and Simon, H.A. [1959] Report on a general-problem solving program. Proc. International Conference on Information Processing, UNESCO House, Paris, pp. 256-264.

Nilsson, N.J. [1969] A mobile automaton: an application of artificial intelligence techniques. Proc. International Joint Conference on Artificial Intelligence (May), pp. 509-515.

Paul, R., Falk, G., and Feldman, J.A. [1969] The computer representation of simply described scenes. Artificial Intelligence Memo No. 101, Stanford Univ., Stanford, Calif. (October).

Pfaltz, J.L. and Rosenfeld, A. [1969] Web grammers. TR-69-84, Computer Science Center, University of Maryland, College Park, Maryland (January).

Pingle, K.K. [1966] A program to find objects in a picture. Artificial Intelligence Memo No. 39, Stanford Univ., Stanford, Calif. (January).

Pingle, K.K. [1969] Visual perception by a computer. In Automatic Interpretation and Classification of Images, Grasselli, A. (Ed.), Academic Press, New York, pp.277-284.

Pingle, K.K., Singer, J.A., Wichman, W.M. [1968] Computer control of a mechanical arm through visual input. Proc. IFIP Conference, Edinburgh, pp. H140-H146.

Rabinow, J. [1968] The present state of the art in reading machines. In Pattern Recognition, Kanal, L.N. (Ed.), Thompson Book Co., Washington, D.C., pp 3-29.

Digitized by Google

Digitized by Google

Raphael, B. [1968] Programming a robot. Proc. IFIP Conference, Edinburgh, pp. H135-H139.

Rastrigin, L.A. [1967] About the identification of images of spatial objects. In Problems of Cybernetics and Law, Kudryavtsev, V.N. (Ed.), Nauka Publishing House, Moscow, pp. 361-368. Available through U.S. Department of Commerce, Joint Publications Research Service, No. JPRS: 43,954 (January 10, 1968).

Reddy, D.R. [1969] On the use of environmental, syntactic, and probabilistic constraints in vision and speech. Artificial Intelligence Memo No. 78, Stanford University, Stanford, California (January).

Roberts, L.G. [1963] Machine perception of three-dimensional solids. Technical Report No. 315, Lincoln Laboratory, M.I.T. (May). Also in Optical and Electro-Optical Information Processing, Tippett, J.T. et al (Ed.), MIT Press, Cambridge, Mass., 1965, pp. 159-197.

Rosenfeld, A. [1969a] Picture Processing by Computer. Academic Press, New York.

Rosenfeld, A., Thomas, R.B., and Lee, Y.H. [1969b] Edge and curve enhancement in digital pictures. Univ. of Maryland Computer Science Center Technical Report 69-93 (May).

Rosenfeld, A. [1970] Connectivity in digital pictures. Journal of the ACM 17 (January), pp. 146-160.

Sakai, T., Nagao, M., and Fujibayashi, S. [1969] Line extraction and pattern detection in a photograph. Pattern Recognition 1 (March), pp. 233-248.

Samuel, A.L. [1967] Some studies in maching learning using the game of checkers II - recent progress. IBM Journal of Research and Development 11 (November), pp. 601-617.

Shaw, A.C. [1968] The formal description and parsing of pictures. PhD Thesis, Report CS 94, Computer Science Department, Stanford Univ., Stanford, Calif.

Sobel, I. [1970] Camera models and machine perception. PhD Thesis, Artificial Intelligence Memo No. 121, Stanford University, Stanford, California (May).

Tenenbaum, J.M. [1970] Accomodation in computer vision. Forthcoming Ph.D. Thesis, Electrical Engineering Dept., Stanford University.

Digitized by Google

Digitized by Google

Thorwald, J. [1965] The Century of the Detective. Harcourt, Brace, & World, New York.

Williams, C.M. [1965] Isolation of important features of a multi-toned picture. Artificial Intelligence Memo No. 29, Stanford University, Stanford, California (January).

Yurans, V. [1967] Certain questions concerning the theory of identification of objects with the use of the apparatus of projective geometry. In Problems of Cybernetics and Law, Kudryavtsev, V.N. (Ed.), Nauka Publishing House, Moscow, pp. 349-360. Available through U.S. Department of Commerce, Joint Publications Research Service, No. JPRS: 43,954 (January 10, 1968).

Digitized by Google

Digitized by Google

APPENDIX A

Two pictures of bodies: Input data

Digitized by Google

Digitized by Google

```
KELB1                                  16-JUL-7Ø Ø8:46
   ITH=15
   Ø5Ø       Ø6Ø       Ø7Ø       Ø8Ø       Ø9Ø       1ØØ       11Ø       12Ø       13Ø       14Ø       15Ø       16Ø
   BCEE....EE...............A6656ACCCCBBBBCBCCCCCCCCBCCCBBCCCCCCBCCBCDCCCCCCCCCCCCCCBCCCBCCCCCCCCCCCCCCCBCCCCCCCCCCDCBC
   BDEE....EE...............C6666BCBCBBBBBBBCCBCCCBCCBCCCCCCCCPBCCCCBCCCCDCCCCCCCBCCCCCCRCCCBBCDRCBCCCCCBCCBBCBBCCCCBCCB
   DEE.EE..EE..E....EE......B6655BCBHBCCABCBUBBCCCBCCBCBCCCCCCCBCBBCCBCCBCCDACBCCBCBBCPDBBCCCCCCCCCBCCCCCBBCCBCCRCCCCCBC
   DEE..E.E.................B6655ABBBBBBCBBBHBBCCBBCBCBCCCCCPCCBCCCBCBCCCCHBCCCCCBCCCCRCCBCCCCBBCBBCCCCCBCCBCCBBBBBCBCBC
   E..E..E....E..........E..DB6656APBBBBBBCBBBRBBCCBBCBBBCBBCBCCBCCBBBCCBCBCCCCBCCCBBCCCRBCCCBBBCCBBBCBBRCCCCBCBCCBCCCBBCB
   E...EE.E.....E.......EBAAC6656ABBBBBBCBBBRBBCCBCBRCBRBBBBBRCBBBBCCBCBCCBCCBBCCCBBBRBCCRCCCRBBCCBBCCCBCCBCBBBCBBBCBBB
   E.E..E.EEE...E.....EBAAA9C66559BBABBBRBBCBRBBCBBBBBBBCBCBCCBBBBCCCBBBCCBCBBCBCCCBRBBHCBCCBBBBCBHCCBCBCBCBCCCBBBRBBB
   E..E.EE........EEDDBAAAAAB66569BBRBBBBBBBBBBRBRBBCRBHBRBRBCBCCBBBCRBBBBBBBCBCBBBBBBBBBRBBHCHBABBBBBBBBBBBCBCBBBBBBBBBBB
Ø3ØEFEEEE..EEEE..EDAAAA9AAAAB6655ABBABBBBBBCBBRBBBBBRBBBRBBBPBBBBBBCBBBBBBCCBBBBCBBBRCRBCBBBBBCBBBBBBBBBBBCHBCBRBRBBBBBB
   EEEEEEEE..EEEDAA9AAAAAAAAAB6655AABBBAABBRBBBBPBBBBBBBPBRBBBBBBBAABBBBBBBCCBBBBBBBBBARBBBBBBBBBBBBBBBBABBBBBBCCBBBBBBB
   EEEEEFEE..ECB9A9AAAAAAA9AB6655ABABBBAABABBBBBRBBBBBBBBBBBBBBBBBBBBBBBBAHBBBBBBBBBBBBBBBABBBBBBBABBBBAABBABABBAABBBBABBB
   EFEEE.EEEECAAAAAAAAA99AAAC66559BBBBAABBBBRBBRBBABBABBBBABBBBBBBBBBBBABBBBBBBBBBBBBBBBBBARBBABARBBBBBBBBBBBBBRAABBBABBA
   EEEEEEEBAAAAAAAAAAAAAAAA9B6655ABABBAAAABBBBBBBABABBBBBBBBBBBBBBBBBBBABBBBBBBBBBBBBBBBBABBBAABABBBBBBBBBBBBBABRBBBBBBARBBB
   EEEEECAAAAAAAAAAAAAAA9AAB66459BABGAABBBBABABBBABBBBBBBBBABBBBAABBABABBAABBABBBBBBBBABABABBBBAABBABBBABABBBBBAABBABBBAB
   EEDBAAAAAAAAAAAAAAAAAAA9AAAA559BABAAABBBBABABRABBABBBBBBABBRBBBBBBBBBBBBBBBBBBBBABBBBBBBBBAABABBBABBBABAAAABBABBBAABBABABAA
   EEAAAAAAAAAAAAAAAAAAAAAAAB99559BABAAABABBBABAAAAARBBBRBBABBABBAABBABBBBBBBBAABBBABBBRBBBABAABBBAABBAABABBBBABBBABABBAB
   EDAAAAAAAAAAAAAAAAAAAAAAAAA99559BAPBAABABBABBBBBBBBBBBBABBABBBBBABBAABBBBBBAAABBBBBBABBABAAABABBBABBAABAAABAABBABBAAB
   EE9AAAAAAAAABAAAA9AAAAAAAAAA459BAAAAAAAAABABBBBABABABRBBABBBABAAAABABBAABAABABBAABBBAABBBAAAAAABBAPAAAAABABBAABABABAAA
Ø4ØEDAAAAAAAAAAAAAAAAAAAAAAAAB3459BAAAAABBAABAABBABBAABAAABAABABBAABBABABAAABBBAABBAAABAABBBAAABAABBBABBABABAABAAAAAAA
   EDAAAAAAAAAAAABAAAAAAAAAAB99458BAAAAAAAAABAAAAAABAAAAPBAAAAABBAABBAAAAAAAAABBAAAAABAAAAABABAAAABAAAAAAAAAAAAAAAAAA
   EDAAAAAAAAAAAAAAAAAAAAAAAAAA458BAAAAAAAAABAAAAABAABAAAAABBAABBABRBRABBBABBBAAAAAABAAABABBAAAAABAABABAAAAABAAAABBAAABAAAA
   EDAAAAAAAAAAAAAAAAAAAAAAAAAA458BAAAAAAAAAABABBAABAABABBBBAAABABBBAABABABBBABAABABBAABAABAAABABAAAAAAAAAABAAAABBAAAAAAA
   EDAAAAAAAAAAAAAAAAAAAAAAAAPAA458ABAAAAAAABAABABBBAABAAAAAABBAABAAAABBAABBBAAABAABAAAAAAABABRABBAAAAAAAABBAAAABAAAAAAAAAAA
   DDAAAAAAAAAAAAAAAAAAAAAABAAAAA458AAAAAAAAAAAAAAAAABAAAAAAABAAAABAAAAAABAABAABA9888779AABAAAAAAAAAAAAAABAAAAAAAABAAAAAA
   EEAAAAAAAAAAAAAAAAAAAAAABAAAA458AAAAAAAAAABABAAABAAAAAABBAAAAABBAAAAAAAABAAB87777666 7BAAAABAAABAAAAAABBAAAAAAAAAAAAAAABAA
   DCAAAAAAAAAAAAAAAAAAAAAABABAA458BAAAAABAABBAABBAAAAABAAAABBAAABAABAAAAAABA56676677766 89BBAAAAAABABAAAAAAAAAAAAAABAAAAAA
   89AAAAAAAAAAAAAAAABBAAAAAAAAAA459BAAAAAABABAABAAAABAAAAAAAAAABAABAAABAAABABA6555666666665ABAAAAAAAAAAAAAABBAAAAAAAAAABAA
   55ABAAAAAAAAAAAAAAAAAAAAAAAAAA45ABBAAAA9AAAAAAAAAAAAAAAABBAAAAAAAAAAABAAAA9A455555555555655ABBBAAAAAAAAAAAAAAAAAAAAAAABAAA
Ø5Ø66AAAAAAAABAAAAAAAAAAAAAABAAA449BAAAAABAAAAAAAAAABAAAAAAAAAABAABAAAABAABAA654555555555556568AAAAAAAAAAAAAAAAAAAAAAAAAAA
   66AAAAAAAAAAAAAAAAAAAAAAAAAAA448AAAAAAAAAAAAAAAAAAAAABAAAAAAAAAAAAAAAAAAAA55A855555555566666AAAAAAAAAAAAAAAAAAAAA9BAAAAAA
   67AAAAAAAAAA9AAAAAAAAAAAAAAA348AAAAAAAAAAAAAAAAAAAAAAAABAAAAAAAAAABAAA65A.CA68866669A8559AAABAAAAAAAAAAAAAAAAAA9AAAAAA
   66AAAAAAAAAAA9AAAAAAAAAAAAAA449AA9AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA65ABCCCBBA999ACB659AABAAAAAAAAAAAAAAAAAAAAAAAAAA
   76AAAAAAAAAAAAAAAAAAAAAA9BAAA45AAAAAAAAAAAAAAAAAAAAAAAAAAAAABABBAAAAAAAAA655ADCAAAAAAABCC65AAAAAAAAAAAAAAAAAAAAAAAAAAAAA
   67AAAAAAAAAAAAAA9AAAAAAAAAAA349BAAAAAAAAAAAAAAAAAAAAABAAAAAABABAAAAAAAAAAAAA645AB876899997BB69AAAAAAAAAAAAAAAA9AAAAAAAAAAA9
   77AAAAAAAAAAAAAAAAAAAAAAAAAAA448AAAAAAAAAAAAAAAAAAAAAAAAABAAAAAABAAAA65AA865579BB857969ABBAAAAAAAAAAAAAAABAAAABBAAAAB
   779AAAAAAAAAAAAAAAAAAAAAAAAAA448AAAAAAAAAAAAAAAAAAAAAAAAPB3ABBABBAAAABBBBA5A996778ABB968AA9BAABBBBBBBAABBBABBBBBBBARABA
   67AA9AAAAAAAA9AAA999AAAAAAAA359AAAAAAAAA9AAAAAAAAAAA9BBBBBBBBBBBBBBBBBBBA7998BA99AABBB9ABA7BBBBBBABBBBBBABBBBBAABABBAABB
   769AAAA9AAAAAAAAAAAAAAAAAAAAA449AAAAAAAAAAAAAAAAAAAAABBBBBGBBBBBBBBABBBA999ECCAAAACCCABCA8ACCCBCCDCCCCDDDDDDDDDEEDEED9A
Ø6Ø8BAAAAAAAAAAAA9A9AAAAA9AAAAA34AABAAAAAAAAAAAAAAAAA9BABDDCEDEEE.EE.....E8699BCBAAAAABABCA8........................
   CBAA9AAA99AAA9AAA9AAAAAAA9AA34AAAAAAAAAAAAAAAAAAAA9ABBD......................96BBA996998ACDAA........................
   DEA9AAAAAAAAAAAA9AAAA9AAAAAA348AAAAAAAAAAAA9AAAA9ABAE......................96AA99986699BCAD........................
   EDA99A99AAAAAAAAA9AAA9AA99AA345AAAAAAAAAAAAAAAAA9AAAAD.......................8AA99A9889ABCB.........................
   DDA99AAA99AA9AAA9AAA9AA9AAAA445AAAAAAAAAAAAAAAAAA9ABBD........................999778889BB.........................
   DCA99999AAAAAA9AA999AA999AAA34AAA9AAAAAAAAAAAAAA99AABD........................99A9877AABD.........................
   8799AA9AAA9A9AA9A9999AAA79AA35AAAAAAA9AAAAAAAAAAA99AAD........................A9A98999ABE.........................
   668AAAAAAAA9A99AAAABAAAA79AA44AAAAAAAAAAAAAAAAAAA9AAABD........................99AAA88AACE.........................
   669AAAAA9999AAAAAAAA987999AA44AAAA9AAAAAAAAAAAAAA99ABD........................999998BABE.........................
   768AA9AAAAAAAAAA998888999B9AA34AAAA9AAAAAAAAAAAA999AABE........................89988999BBE.........................
Ø7Ø678A9AAAAAAA89898999AAA79AA355AAAA999AA9AAAAAA999ABD........................9989899AABAE........................
   769AA9A9998999999999999A79BB44AAAAAAA9AA9AAAAAA999BAD........................9989999AAA7E........................
   768A9998AA899999999999A979AA348AAAA99999A9A9AA999ABAD.......................899989AA988A.........................
```

1⁶8

Digitized by Google

```
       668AA9A9CC99999AA998999A79AA345AAA99AAAAA9AAAA9A99BAD...................A798899987A.............................
       679AA9AACC8999AE98889AA979BB445AAA999AAAAAAAA9A9AAAD...................A8799988879B............................
       678AA9AADD8999AE9998899A79BBAA5AA99AAAAAAA9A99A99AAAD...................5AD888998999............................
       668999ABDD8999AE9889899989AAAB59A99AA9AA9AAA99A999ABD...................A7AA78999AAD............................
       668A99AAEE8999A.9999878989AAAC5AAAAAAAA9A9AAAAA99AAAE...................CA8AA8999ED............................8
       768AA9AAEE8999A.9888767899BBBD5AA9A9999AA9AA99A99AAAD...................D.BABBBB.9E............................9
       679A99ADDD999AABA99998896988D85AA999AAAA999999A999BAD..................EE.CDDDDDDEBD...........................9
087668AA9ABEEDEEDDEDDDDCCDA68CCDD49999999999A99AA9999AAD...................EEEDD.EE9D............................9
       769999ABEEDDADDDCBBDAACA78BBEA59999999999A9999999AAD....................EEEEEAACE.............E...........9
       667A99ABEEDDCDCEDBBCCADA7899AA5999A9999999999AA999AAD.....E.............E..EEEBBDE.............E..E.........A
       663999988EEDECCCCDDDB9997866A5599999999A99A999999AAAD.....EE.............EEEDEEEE...............EEED.........9
       677999AHEEFCECCCDDCCBA9A8776685599A9999999A99999999AAAD....DE.E..........EE...E.EEEE..............E..DDD.........9
       668A99AAEED.DEEEEEDDDHCA69AAA559A9999999A99999989AAD....CDEDEE..........EE.EEEEE.E...........E..CCD.........9
       799ABAAADDBBAA99AAA999999ADDB559A9999999A99AA9999BAD....BDDBDDE.........E.EEEEEEF.E..............BBD.........9
       AABA999888888988988888888889DDB55998A999999A99999999ABD....BDECEE.............EDE..EE............E.E.BB.........A
       6899888898898688887888888789EEB55999AA99999AA999999AAD....EBDCDEE.........EE...EDEEEEEFE...........D..DBB.........9
       9A8989987798899887788888879CCC55999A9AA999AAA99A989AACDD...DCDD..............EDCE..E.EEE...........D..ADD.........9
090899899989989689988898888879DDC65999A99999A99AAA999AAAA74D..AADDEE..............EDDE.......E.......EDD.BDDE..........99
       6A9989999999898888888989DDC6599999999AA999AA99A9AAA55CEEDADDEE............EEEDE....................ADDD..........9A
       99989999998999988888878889DDD6399A99999999AAAA9999AAA74DDDDADCDD............DDED.EEE....EE..........ADDD..........99
       6999999999999999988988889DDD759A99AA9A999A9AA9999ABA74DEDEADBAA............DEEEE.EEE....EEE...E......ADDC..........9
       69899998999989999888888889DDD659AAA9AAA9A9AA999999ABA74DDDEBABDD.EE........DE.EDEEEEEE.EEE...E......ACCD..........9
       69989989999989998999888889EED759A999AAA999A9AAA999BAA8BDEDEDACAA.D.......EEE..DDEDDE....E.E.........ACCD..........9
       69999899AA999999998888888889AAA5599A9AA99AAAA9AA9999AAB99EDCEDABAAEEE.....EDDE.EDD.EEE...EE.E.........ABBD............DAA
       99A8999999998999988889898889CCC839AA9AAA99A9AAAAA999AAC...DBDDAABBEE....EEDDC..ECDEEEEE...EEEE.........ABBD...AAAAABABAAAAA
       8A9999899999983988888888888AA6559A9AAAAA99AAA99AA9ABAABBCDDDDBABBEE....EECDD..EDDEEEEEE.E.EEE.......EACCD..8AAAAABABAAABA
       69877787888889988777876779AA5459AA9AAAAAA9AAAA9999AAAAACECDDC9BREE....EDDDD..EDEEEEEE.EE.EEE.........ACCDE.98888997888887
100799898999999999989998978884A5559AAAAAAAAAAAAAAA9AAAAACECCDCABBDE....DDDD..E.EEE..EEE.DDEEE.......ABB.D.88889999999999
       6A9999999999A9698888888887AA5459AAA9999AAAAA99AA997878ADECCDC9AAEE...EDCEE..EEDDE..EEEE.DDEEE.....ABDDED..8999999999999
       9AA999999AAA99999999997699655AA9AAAAAAAA99AAAA999999AD.BBDCAAAE...DCD.....DDBD.EEEF.E...DDDE......9ADD.E..7899999999999
       6AAA99999999999999998DDEBBE77665AA99AAAAAAA9AA99A9999999ACECBDDB99E..EDCE...ECCD.EEEEDE.E...EE.......9ADD.E..7899999999999
       6AAA9AA9AAAA94899999..9A9EBB865AA99AA9A9A9AA99A9999999ADECBDDA99E..DCDE....ECDE.....EEE.....EE.......9AEE.E..8999999999999
       8AAAAAAAAAA9999899977878 6AAA6559AA9AAA9AAAAA99AA9999999E.DBCCB99E..DDE.....DD..EEEEEDE..........9BEEE...8899999999999
       6A9AAA9AAA999999998689766666559AAAAAAAAAAAAAAAA99999AAD.CBCDBAAD.ECE.....DE..E...EDE..E...E.......9AEED...8999999A99A99
       6ABAAAAAAAAA999998896 9D...995359AAAAAAAAA9A9BBA9999A9AAE.DBBCB99E.DDE.....D..EE.EE.EEE.....E.......8DDDC...8999AA999A999
       6AAAAAAAAAAA9999888888D.....44599AA999A9A9AAAAA9999999.EDBCDA88EEDD....EEE..E.....D.E..............9CDDD...7999999A999A9
       6AAAAAAAAAAA9A988897 8A...8844599AAAA99AAAAAAAA9999A9AA..DBPDA77BEEE....EEE....E..EEEE......E.......9DBBDE..789999A99A999
11069BAAAAAAAA99899988988 8AB89964599AAAAA9AAAAAAA9AA999ABB..DBBCA99AEE.....DD.......EEEEEE......E...99CCCD...899999A99A9A9
       69AAAAAAAAA98999889888888AA6459AAAAAAAAA9A9AA99A9A99BB.EDBBBA99.EE....EEE....E....DEEE..........7ACCCE...8899999A99999
       79AAAAAAAA9999899988 98988AA6459AAAAA9AAA9AAAAA9AA999BB..EBABAEE.EE....EE...ED.....EE.......E..7ACDDD...8899999A999A9
       AA99AAA999A99899899 8998989AA6459AAAAAAAA99AA99A999999DDE.DDAAD..EE.....EEE...BD.....EEE.......E.E7BCDDE...A8999999999A9
       63AAA99A99A9989999998989 88AA655AAAAAAAAAAAAAAA99AAAAADDE.ECAB.EEE.......EE...DDE...........E.......6BDDDE...A999999A9AAAA
       68BBAHABAAAA98899999999898 96CC645AAAAAAAAA9AAA9AAAAA99AADD..EDAC...E...............D..........E.......6BDDDE...A89999AA9A999
       68BBABBAAAA989999889989888CC6459AAAAAAABAAAA999AA99AAEEE.EEAD...E...................EE.E....E.......6BDDDE...B69999AA99AAA
       65AAAA99AA99989999999999979BB5459AAAAAAAA9AAAAAA9AAAA9A.CD.DDAD.................EEE...........................AADDDD..EA89999A99AAA9
       69BABAAAAAAA9999999999989 9889CC645AAAAAAAAAA9AAAAAAAA9AA.CD.EDAD......................................DACDDE..DA89999A999999
       68BAAAAAAAAA889A9999988998 9BB6459AAAAAAAAAA9AAA9AAA9AB.CC..DBE..................D.........................B9DDDE..D.799AAA9A99AA
12069CBAAAAAAAA9999988988898AA4449AAAAAAAAAAA9AAA99A999A..E..DC...........EE...CD.EEE..........................8ACDD...E.89999A9AA9A9
       ....99AAAAA9999999999989889BB5459BAAAA9AAAA99AAA99999A..E..DC...........EECD.EEDEEEE......................ABDD...D.699999999A99
       ....899999AA99989989989888CC5459BAAAAAAA9AAAAAA9999AA.CD.EBB.....DDEDEEDDECCDDCDEEDDCDEEEEDDDD..8ADDE...E.999999999999
       .....999AA998999999999998888CCA449AAAAAAAAAAAAAAAA9A9AA.CD.EA8A99999789ABBBBBBDBCCCCBDBCDBA666AD...9BBD..E.7999999A99A9
       .....99999A999999889989889BBB459BAAAAAAA9AAA9AAA9A999A.CD.DA65455555555556669ABBA788977666655585BDA9CCC..E.78999999A99A
       .....AAAAAAA99999888888888AAA459BAAAAAAAAAAAAAA9999A9A.CD.DA644555555655555655556776655566655855 86ABBEE...799999A9A999
       .....9AAAAAA99999999989888AAA449BAAAAA9AAA9A99AA99999EDD.CA845666555555555555585555566666676558ADDEEE..789999999AAA
```

Digitized by Google

```
    ....9AAAAA9899A99998
    DDDDAAAAAAA8899999889889888AAA458CAAAAAAAAAAA99AA99999DDEAABA46665555555555555555694455566676666766 87AAEE...6999999A9AA
    DDDCCE....99899999998988888AAA449BAAAAAAAAAAA9AA9AA999ARBECEC645665565655555555545866565566666667656 8BBD....7999999999AA
130CDDDC.....99999999998988878AAA449AAAAAAAAAA9A99A999999AAD.CA656665655555555555556555566666666776559BBE....889999999999
    8DDDC.....9986899996898888AAA449BAAAA999AAAAAAA9999999E.EB655665666555555555666666566666666776559BB.....88999A99999A
    9DDCD...AA999689996898888AAA44ABAAAAAAAAAAAA999AAA99999E.EB65666566656666666565576666667667777766 59AAE.....8999999999
    AA9996899888888888AAA45ABAAAAAAAA9AA999A9999999E..B6556665565556565556666666666666667775 59BBE.....8999999999
    ACDCDE..BB9996899886588888BBA558CAAAAAAAAAAAAAAA9A9999AAD.EB6565566666556555665556665667667777876559BB......99999999A9
    CRDDDE..BR8997999886698988AAA44ABAAAAAAAAAA9A99A9A999999E.EB656666666665665556556655566777667776558BBDACA9999999999A99
    ARDDDE..AA8997899886588887AAB44ABAAAAAAAAAA9A99A9999999999899656665666665556665655555566667767765 58889ADA989999999999
    8ADCDD..888997999886668788AAB559BAAAAA9A9AAA999AA99999999996566665665656555555665566666677786655999ABDA89999999999A99
    BCDDCD....8899889996569887AAC445BAAAAAAAAAAAAAAA9A9A8AA9AA996666666666566666665556556766567777776559AAABEA899999999AAA
    CBDDDD....899768766655888799B455AAAAAAAAAAA99AA999A999999AAAA65666656555555556666556676666776765569AAAC.E888999999A99
    CADCCD....788768788665877799C545AAAAA99AAAA999AA9999999BBAA7566656666655655656666656666667767655 99AAAD.E899999999999
140CRDCDD....7997687886658 87899B545BAAAA9A99AA9AAA9A9999999BB99756656566656665555665655565666677765 5579AA9C.E899999999AA9
    DACCDD....779768877656888899C555AA9AA9999AA999A99999998BA98755556666565566665665556666667766655668998B..899999999999A
    CADCCD....7796888887678888 99B455AAA9AA9A9AAAAAA999999AAA9897566665555665556555555676666666765558 9996AEE899999A99A9AA
    CBCCCD....788798988867678888C455AAA999999A9AAA99A9999AAA86975556555556555566555655566766777765557 99999CA999999999AAA9
    CBCCBD....BRDRACBAA767767899C445AAA9AA9AAAA9AAA999998999A65A7566555555556656555555666666777665579996AC998A999999AAA9
    DACCC...............68887788B4459AAAAAA9A9AAAA9999999998975A856666656565555556555556566667667776556 8998BB8989AA9A9A9AAA
    CABCC...............68787888B445AAAAA99A999AAAAA9999999988697455666665555565555555555556667777655 89998C89999AA99AA999A
    CABCC..............688877888A335A99AAA9A9999AAA999999999989787566655655555555555555555666556668765589AA8998999999A99999
    DABCB...........688788878788A3459AAAAA99A9A9AAA9A9999998879755565555555555565555555555556666775688 9999899999999999999
    BBBCCE........67788788878788A344AAA9AA9999AA99AA9999999989875555555555555555555555555555666887 89899898999999999999AA9
150B9BCCD......97678887878877 88A3349A9AAAAA999999999999999898885555665565665565554555555555555688889899889999999A999A9A
    BABCCFE....68777988787878777A3459AAAA999A99A999999999999989888655566656555655554555555555565568888888888899999999999A99
    CRACCDEE..978887888787878788A34599AAA9A9AAAA99A99999999999888856666655556555655455555555565558888889988999999999A99A99A
    CRABDDDE..86889877788878777 77A4459AAAAAAA9AAAAAAAA99999999889865556555565555565544555545555555568888989998989999999AAA9
    CRACCCBEEE8678877668779787894459AAAA99A9A9A999A999999998998865565555565555554455555555656556688889888899999999AA9A
    DCCCBBBEEEA87778666887878777 93459AAAAAAA9AAA99A9A98998899989875556555555555545556555555567888889988999999999A99A
    BCBCCCAEEEA78787777788877779335899999AA9A9999AA99A89999899988866555555555555545555555556655556 89788998989999999999A99
    BRABAA9EEEA7677777777777767782 4499A999A9A9A9AA99999999999989986655555555554554555555555555576988899898889999999A99AA9
    555677AEJDA668777777776876778344 89A9AA9AA999AA99999999999998888665555555555545545555555555556 8898889988898999999999A
    666677ADEEA65777777777778677834499A9AAA999AA99999999999999898976655555655555455555555556555568898889999999999999999
160654577 9EDDD66776766777777766844499AAAA999AA9AAA999A99889998988665555555555555455555655555568989888888998999999 9A9AA
    5566879DDDE656677667777777677834 499A9AAAAA999AAAAA999999988989666556555555555555555566555557898989998989999999A99999A
    7666668DDDD566677887877777778445 8AAAA99A9AA9AA99AA999989899887755555555555555455556655565568988898889899999999999A9A
    6666668EDDE66677688677777777773 4499AAAA9AAA99AA9999999AA999988886555565555555545555655566556898988998999999999AA9AAA
    6666668DDDE667677777787777778 34489AAA9999999999A999A99999A9989888655555555556554556656555656668988989999999999999AAAAA
    6686889DDDE6678967777777777778 4449 9AAAAA9A999999A99999999898989886556555555555554555555556655688888999989999999999A9A99A
    8899BCDDDDE668AD6666767767668445 89AAA999AAA99999A99999999989988886556655555555455565555566666688989999998999999999 9A9A
    99AABBDDDDD589B.5667776776668445 899AAAAA9AA999A999999999999998998875556556555554455555565656667999989889999999999A9AAAA
    99AAABBCDDD68A..455677767677844 4399AA9AA9AAA999999999999989889887555555555555545555555565666999899999999999999AA9A9
    777666554458 9D..45566767666684448 99AA9A9A9A999A99999999999998998555555555555545555555655557988989999999A99AA99A99A9A
170443444454469B...75566776676674458 99AAAA99A9A99AA9999999999999998888855655555545545555555555655698999999999999999999A999AA
    555566667666AD....66677776666844598 9AAAA99AAAAAA999A9999999998998885565655555555655555555555556999999999999999AA9A9AAAAA
    65666667667D.....666767676668445 88A9AAAA9A99AAA99A9A99999998899888666555555555557555555555555699989898 8999A999999A99AAAA
    66566676668......666677666668444 99A999A999AAAAAA99999999999999998896556555555555558655555555555666999999998999999999A9AAAA
    65665666779......886776766777444489 9999AAAAAAA9A9A999999999999989986555555555555875555555555666899999899999999AAA9AAAAB
    5555556799A......886677666666844498A9999A9A9A999A999999999999999988886555555555555886444555565776999999999999999999A9A9AAA
    55555668AA.........666766666674446899999999A999A9A99999998998898896655545555555885445555566886999999999999AA9AAA9AAAAAA
    55556789DD.........66776666674444 89AA9AAAAAAAAAA999A9AA99999998886555555555555885555555556899699999999 9A99999A999AAAAAA
    65567788............676666666844458 9AA999A999AAAA9999A99999989998885665555555549845555555669979999999999999999AAAAAAAAA
    555668AE...........66676666685445899AA9A999A99999999AA99999998886665555555548954455565 66AA999999999999999A9999AAAAAA
180555689D............666766666854468 99AAA9999A99AA9A9999999999999996655565555558845555566559999999999999999A999AAAAAA
```

Digitized by Google

```
65668A..............67776666684445899AAAA99AAAA9A9999999999999988945565655555558555555555568899999999999999A9AAAA99AAAAA
56689E..............66756666668445689999999A9AAA9AA9999999999999996555555555544755555555569AA99999999999999A9AAAAAAAAAA
5679B...............66776766674445899999AA99AAA9AA9AA9999999999999866665555555457555665566 9AA999A9999AA99A9AAAAAAAAAAA
658AE...............6767676667544458999999999AAA99A9A9A99999999899986655566655558455556555 8AA999999999A9999A9AAAA9AAAA
679D................667666666744458999999999A9999A9999999999999989999566655555554484555555668999A999A9999A99A99AAAAAAAAAA
78A.................66667666665444589999999A9A9A9999A999A999A999999988665655555555845555565668999999999AA9AA9A9AAAAAAAAABA
8AD.................6676666666544459999999999AA9999A9A9A999999998999865566655554584555655668999999999999AAAAAAAAAAAAAAAA
8C..................6676776666544458 9AA999999A999A9A99AAA999999999987555665555557544556565899999A999999AAAAAAAAAAAAAAA
BD..................67676766675444599AAA99AA9AAA9999999AAA9999999998555655555558455555656988999A9AA99AAAAAAAAAAAAAAAA
190E................86766666665455899999A9999999999999AAAA99999999995555555555564555565568889 99A999AAAA9AAAAAAAAAAAAAA
D...................866666666754458 99AA9999A9AA99A9999999A99999999886565556555457455556555899999999A99AAAAAAAAAAA99AAAA
C...................8777787777544689AA99999999999A9999999999999999985556555554574555555558999999AA999AA9AAAAAAAAAAAAAA
C...................86666766675444589999999999AA99999999999999AA999988655565555555645555556689999999AAA9AAAAAAAAAAAAABAA
B....................88999888854558 9AA999999A9999A99999999A999999998655655555556555555557899999A99999AAAAAAAAA9A9AAAAA
B....................99998988854558999999A899999 9A9A9999999999999999997555655554555555556578999999A9999AA99AAAA99AAAAAA
B....................9999988885455899999999999999999999999999999989885556665555455555555 8788999999999AAAAAAAAAABAAAAA
B....................988888778544598999999999999A99999999999999998988556655655545555555587999999A99AAAAAAAAAAAAAAAAAB
B....................9989888885455889999999999999999999999999999999985555556555455555555687999 99A999AAAAAAAAABBDBDDDEEE
A....................9998999986455999999999999999999999999999999999996556555555455555655 88AAABBBCCDDDE.E...............
200A.................99A999998544589999999999999999999999999A999999999975565565545555555555679EE.........................
B....................AAAA99AA976558999999999AAAAAAAABABBCCCDDDEDDEEEEA45555655555555555779EE...........................
A....................BBBBCBBBBABBABACCBACDDDDDDEEDEDEEEEEEEEE.EEEEEEEEB4555655545555555 6679.....E.EE.....................
A....................DCCCDDDDDCDDDDEDDDDDDDEDDDDDEEEEEEEEEEEEEEEEEEEEEDD455555544555555555679......EEEEE.....................
A....................DDDDDDDDDDDEEEDDDEEDEDEEEEEEEEEEEEEEEEEEEEEEEDDDEB45555554455555555569.......EEEEEEE.E...............
AE...................DDDE.EEEEDDEEEEEDDEEEEDEEEEEEEEEEEEEEEEEEEEEEEEEEDDB44554554445554555559EE....EEEEEEEEE.EE..E.....E.
A....................DEEEEEEEEEEEEDEEDB...EEE..EEEEEEE..EEEDEEEEEEEEEDDA45555455445445666669DD...EEEEEEEEEE....EE.......
AE...................DEEEEEEEEEEEEDEEEDBE.EEDEEEE.E....EE..EEE.EEEEEDED6455555555455555667 68BBBE.EEE.EEEEEEEE......E....E
BE...................DEDDEEEEEEEEEEEEEDAEE.EDEEEEEEE..EEE..EEE.EEE.EEED8455565454555589769CAAE.EEEEEEEEDDEE...EE..E....
AD...................DEDDDEEEEEEEEEEEEE9EE.EDEEEEE.E.EE..EE.E.EEEEEEEEEDB55557775A8998DBCBBB88DEEEEEEEEEEE.DEEEEEEEEDDEED
210AD................DEEEEDDDDEEEEEEEDDAEE.DCEEEEEEE.E.EEEEEEEEEEEEEEDDDDABAAC9655ACCBDDCCDBAAAEDDDDDDDDCCCCDEDDD.EE....
AD...................CDDDDDDDEDDDDDDBDDDDDDDDEEEEDDEEEEEEEEEDDEEEEEEDDDCBCDDAAA755ABBBEEE..AAA9DDDDDDDDDDEEEE..EE........
AD...................DCDDDDDDDDDDDDDBBDCDCDDCDDDDDDDDDDDDCDBDCCBBBBBBCBDCDDDA86559BBBDDE...DD8EDEEEEEEEEEE..............
AD...................BDDCCDCCCDCCCCCAAABBBBAABBABBBBBBCCCCCCCCCCCCCDECCCEEE966559AAADEE.....EEEEEEEEE.................
9D...................9AABABAABBBBBBC9ACCDCCCCCCCCCCCCCCDDCDCCCCDCCCDDDDEDDDA75677588BCD.......DEEE.EEE................
AD...................CBCCCCCCCCCCCCC74CCBBCCBCCCCCCCCCCCCCBCBCCCCBDDDEEE.E9568888669CCDE......E......................
AD...................BBCCBCCCCBCCBCB74CCCBCCBCCCCCCBCCCCBBBCBCBCCCDDDEE..D5689898998CCCCE..........................
AD...................CCCCCCCCCCCCCA9ADCCCCCCDDCCCDCCCCCCDCDDCCCCCCCD....E9799AAABBABCBBC...........................
AC...................CDDCDCDDDCDDDDA9ADDDDCEDDDDDDDDCDDDDCCCCCCCCDDEE....8AABBCCBBBDDBBDD...........................
AB...................CDDDDDDDDDDDDEDAAAEEDDEEEDDDEDDEEDDDDDCCDCCCCC.........CCCCCCCCDCE.......A.......................
2209B................DEEEDDDDDDDDED9DDEEEE.EEEEEEEEEEEEEEEDDCCDDCCA79D..DEEEEEDDDDDDDDCBBBBCCDB........................
AC...................DE.EEEEEEEEEEE8EEEEEE.EE...E.E.EEEEEEDDDDDDDCBBBD.......EE..EEE.........CD........................
AB...................EE.E.E..EEE..D9..E.EEEE.EE....E.............E..........................D........................
AA...................E.E..E..EE.EEDA.........................................................C........................
AB...................EEEE.E..E.E.EAB..........................................................E.......................
AA...................EE.E.........AD..........................................................E.......................
BA...................E..........EA.........................................................................................
AB..............................A..........................................................................................
AB..............................D..........................................................................................
AB.............................BD..........................................................................................
230AB..........................BD..........................................................................................
```

171

Digitized by Google

```
KELB1                               16-JUL-70 08:47
  ITH=15
  167       177       187       197       207       217       227       237       247
  CCCBBCCBBBCBA656666BDDDCDDDDDDDDDDDDDDDDDDDCCDDCCDDCDDCCCDCCCDDCCCCDCCCDCCCDCCDCCCCC
  BBBCCBBCBBCBB655666BDDDDDCDDCDDDCCCDDCDDDCDDCCDDCDDCDCCCCCDCCCCCCCCCCCCCCCCCCBDCCCCD
  CBBCCBBBCCBBA655655BDDCCCDDDDCDDCDDCCCCDDDCDDDDCDCCDCDDCCDCCCCCCCCCCCCCCCCCCCCCCCCC
  CBCBBBBBBBCBA655555CDDDCDDCCCCDCCCDCCDDCDDDDCCDCCDDCCCCCBDCCCCCCCCCDBCCCCCCCCCCCCC
  BBBBBBBBBBBAA655666BDDDDCCDDCCDDDDCDDDBDCCCDCCDCCBDCCCCCCCCCCCCCBCCCCCCBBCCCCCDCBBCC
  BBCBBCBBBBBBA655655BDCDDDCCCCDCCCDCDDCCCDCDCCCDCCCDCCCCCCCCCCCCCBBCCCBCCCCCCCBCCBBCB
  BBBBBBABBBBBA655566ADDCCCCCCCCCDCCCCCCCBDCCDDCCDCCCCCCCCCBCCCCCBCBCCBCBBCCCBCCCBBCB
  BBBCBABBBBAAA655555ACCDCCCCCCCDCCCCCCCCBCCBCCCCCCBCBCBBCBCCCBBBBCCCCBCCCCBCBCCCBCCBB
030BBBBABBBAABAA645555ACCDCCCCDDCCCCCCBBCBCBCCCCCCCBCBCCCCCBCBCBCCBBCCCBCBBCBCCBCCBBBC
  BBBBBBABAABAA655555BCDBCCCCCCCCCCCCBBCCCCCBCBBCBCCBCBCCBCBBCBBBBCBBCCBBBBCBBBBBBCCBA
  BBBABABABBAAA755555ACCCBCCBBCCCCCBCBBCBCBBBBBCBCCCBBCCBBCBBBBBBCBBBBCBBBBBBBBBBBBB
  ABAABBBABBAAA645555ACDCCCCBBBCCBCBCBBBBCCBCCBBCBCBCCBBBCBBCBBBCCBBBBBBBCCCBCBCBCCCBB
  BBBBBABAAAAAA655555ACCCCBCBBCBCCCBCCCCBCCBBCCCCBBBBCBBBBBBBCBBAABBBBBBBBBBBCBBBCBBBB
  BBBABAABAAABA645555ACCBBCCCBBBCBBBCBBCCCBBCBBBBCBBABBBBBBCBBBBBBBBBBBBBBBCBBBBCCBB
  AABABABAAABA9655555ABCCBCCCBCBCBCCCBBCBCBBBBCCBBCBBBCCBBBBBBBBBBBCBBBBBBCBBBBBBB
  BAABABAAAAAA9655555ACBCBBCBBCCCCBBBCCCBCBBBBCCBBBBCBCBBBBBBCBBBBBBBBBBBBBBBBBABBBBB
  BAAAAAABAABA9655555ACBCBBBCCBBBBCBBBCBBBCBBBBCBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBB
  AAABAABABBA99645555ABCBBCBBBCBBBBCBBBCBBCBBBBCBBBBBBBBBBBBBBBBBABBBBBBBBBBBBBBCBBBB
040AAAABAAAAAAAA9645555ACCBCBCBBBBBBCBBBBBCABCBBBBBBBBBBBBBBBBABBBBBBBBBBBBBBBBBABBBBB
  AAAAAAAAAAAAA9645455ABCBBBBCCBBCBBBBBBBBACBABBBBBBBBBBBBBBBBBBBBBBABBBABBABABBBBBBBB
  AAAAAAAAAAAAA9655555ABBCBBBBBBCBBABBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBABBBBCAABB
  AAA9BAAAAAAAAA654555ABCBBBBBBBBBBBBBBBBBBBBBBABBAABBBBBABBBBBABBABBAABBBABBBBA
  AAAAAAAABBAA9645555BBBBCBBBBBBBBBBBBBBCABBBBBBBCBBBBCAABBBBAABBBBBABBBABBBBBAABABBBA
  AAAAAAAAAAAAA9645555ABCBBBBCBBBBBBABCBBBBBBBBBBBBBBBBBBBBABBBBABBBBBBBABBBBBBBABBBBBB
  AAAAAAAAAAAAA9644555ABCBCBBBBBBBBBBBBBBBABBBBBBBBBBABBBAABBAABBAABBBBBBBBBBBBBBBAABB
  AAAAAAAAAAAAAA645455ABBBBBACCBABBBBABBBBBBBBBBBCBBBABBAAABBABBABBBBABBBBAABBBBBBABBBB
  AAAAAAAAAAAAAA645455ACBBBBBBBBBBBBBBBBBBBBBBBBBABBABBBABBABABBBBBABBBBAABBBBBBBBBBBB
  AAAAAA9AAAAA9545555ABBBBBBBBBBBBBAAAABBBABBBBBBBABBBBABABBABBBBBBABBABBABBABBBBBAB
050AAAAAAAAAAAAA9644555ABBBBBBBBBABAA9A99AAAABBABBAABBABBAABBAABAABBBBAABBABBBBBBBBABBBB
  AAAA9AAAAAAAA8645555ABBBBBBAABAA6454454555667778999AAAAAAAAABAAAAABABABBABAAABABBBB
  AAAAA9AAAA9A9544555AABBBAAAABAAA54544554444555454454555454555555555656666665666 56666
  AAAAAAA9AAAA9644544ABABBABAABAA9446445454445445544544444444544445444444444444444444
  AAAAAAAAAAAA99654555ABBBABABAAAA84464444544454444444554454445444444444444444544444 5545
  9AAAAAAAAAA999544555ABBBAABAAAAA74454444454544445544454444444544444454444444444544 45544
  BAAAAAAAAAAAA9644555ABBABAAAAAA9644444454444444444444455454444454444444444454444444444
  ABAAAAA99AA99545455ABABABAAAAA954444444544544554554444544444544444444444444544434444 4
  BAAAAAAAAA999554555AAAAAAAAAAA84444444544444554544444554444444455444444444444454444 44
  AAAAAAA9AAAA9544555ABBAAA9BAAA8444444445454444444445444444445445444554445455554544 54454
060.AAA9AAAAAAAA9545455ABAAAAAAAAA644454444555545444455445444445445444445545555455554 54455
  .BAA9AAAAA9A8545555AAABAAAAAAA96445444445454545545454445545555444445545555555445554 5554
  .A9AAAAAAAA999544455AAAAAAAAAA95444444445444545555544454454454454444544454544455554 545554
  .A99AA9A99988554455AAAAAAAAA984444454444544545554444444444445444554545455555454544 5554
  .A9AAA9AA9A85445555AAAAAAAAAA7444444444454544554454444555445554544554445544444544 45544
  .A9AA9A9AAA98544555 9ABAAA9AAA73444444444455544444544454554454444544454444555545454444 55
  .AA9AAAA99999544555AAAA9A99AA63444444444445445545454545544445455444454454554555455544 55
  .99A99A999999544455AAAAAAAAA9544554444554454555545554544545455544444455455455445555 44
  .A9AAA9A99999554555AAAAAA9AA94544454444554445444445455554455554554554444555455454455 45
  .A99999A88A99444455AAAA9AA99644444554454544445555455554555444454444554544555545555544 54
270.A99999999998544555AAAAA99AA64444444555444444554444545444445554554444454444455545555 4
  .A99A999AAA995444559AAA9A9AA6444444444545444544445444444454545455554545454444554444 55544
  .999999999998444455AAAA9A9996444444444444444444445445544444444454554445554554455554 5554
```

Digitized by Google

```
    .A9999A9999894445559AA9AA99974444454444444445545444445555444444444545545545555455545
    .9999A9999998444455AAA9A99997444444554444444444554554455555444544544444554444544554454
    .999999999998544455 9AA9999997354444554544445554444544444454545554544454444455454555 54
    .99999999999843445 5A9A99998914444544444444444444444544444444444554554454444545545 5544
    8A999999999885454559 9AA99989144444455454544455444445444454455445544455544555445555 54
    999999999999843545 59A999988A54444444334444444444554444445544444544554444445544554455 54454
    9A999999999984444 55AAA99987444444434444444444554444445444454444455445554544444544454444
Ø8Ø9A99999999999844445599A99987444444333444444444444454444544445455445444555445445 55555
    999999999998753445 59A999997444444333344443444444444444454544444554444444455444445455 45
    A99999999998734355 59999998654444333444444444433435444445544444544444444444444544444444
    9989999999998733445 59999987654443344433444444443344444444445444444444444454555444444 554444
    999899999999873344 55998998755444333333444444444444454444444444444455544544444 5445554 5544
    999999899998744445 5A9999875544333333344444444444444445544444454455445444455444444 454444
    9A99999889883444 55A99998655434343344445444444444554453344544444 554444445455545455555 55
    A99989989999744445 59999877443333343344444444444444444445544444444 554454554445455554455 55
    9999998999988434455 9998876452333433334444444444444444444544444444444444455545545454445
    999998999999744445 5999898744333343433444444444455445544444454455545554545555555555 555
Ø9Ø99999999998974444 55A99887633333343344444444444444444444455444545544544455554555545 5554
    AA999999999998444455 99988763334333334444444444444454444444455444555544554444555555 54454
    99999999999884444 55A999874334334433444444444444445455554554555544545455555555555 555
    9999999999997444 455999886344443443444455444455444444454454454454555555555555555555 55
    99999999999898444455 99898633433434443345544444444444454554545545554555455555555455555 554
    999999999997445455A99876434444444444455444444454444444444444444444454444444444444444444
    AA99999999998444555A998754344444444444545444444444444444443444444444444444444444444 434
    A99998999999844445 5A98975444444344344454554444444444354444444444444444444444444444444
    AA99A9999999734445 5A99875434444444444445544444454444443344544444444444444444444444445
    789999999999974444 55A998764444444444444455444444444443444444444444444454444444444 5444444
1ØØ99999A9999997444 4AAA998753444444444444455454444444444444444544444444444444444444444444
    9999A999999974444AAA998853444444444444454544544444434444444444444445444544444444444444
    999999999999644445 5A999753334344443334554444444444444444444444434443344444444444444444 454
    99A99989999973444 99A9987434444444344445544454444444443444444434444454444 33444444444444
    999999A99997444 4P BAA98644344444444344445544444444544444444444444444444444444444444444 454
    999A9A9999997444 5AAA998743444434444444455445544444454444444444444444454444444453444444444
    99999A9A99986344 499AA98753444434544444554454554454544444444544444444444 54444444444444
    99999A9999996344 499A99974344444444444454545545544444544444444444444554444444554454444 54444
    9A9A99A999986445 588A998853444444444 5545555444444454444554444444445544444444544444444443
    9A9999999999644458899998634444443444445555544444444444444444 354444444444455444444444444444
Ø1Ø9A99999A99986344 5AAA999863444444444444455555555544544444444454444444444454444444 34544444
    9999AA999998644 5588A99996344444444444444555544445444444444444444444444444444 5545444 45544
    9999A9A999986444 588A9A9863444443444444455554455444444454454444544444444444444454544 544444
    9A999A9999986444588AA9997444444444544545555544544544444 5444554554444444554444444444 4
    A9999A9A999964445 88A9AA985444444544455555555544555445444444554444455445445444455444454444
    999999AA99996454 488AA99986444444444444555555554545554444444544455445444444455444455544
    A999A9999998635558 8AA99A8744444444454445555555545444445445544555554444545455445444 45544
    9A99AA9A99996445 499A99998855444444444455555554445454454444545445444444544444554545 45544
    99AA9A9A99996445 599AAA89886544444444554556555444555444444444444454444555454444454545 45544
    AA99A999AAA964545 88AA99998755444444444555555555555544544444554554454454545455444444544444
12Ø9A9A99AA99996444 588A9A99987664444444444555555555554454554444545455545544544445444544444
    999999999998645459999A99998964444444444555555445555554444544554455445455554454 44444445
    99999999AA986445 588AAA99998874444454454555555555554455444454545455545455445455544444 4
    9999AA9999996445 5999AA99A9887544444554555555555544555555555545544445554554454544 544444
    AAA99AA999996454599AAA9AA99986444445545566778 89AABCDDEEE...................844544454
    9A99999999998644 5599AAA99999997544456659BAC....................................86677777
    A99999999AA96444599A9AA99AAA98689777778BAA.....................................9689889999
```

Digitized by Google

```
   AAA9999999986445599AAAAAA9AA98999988889AAAD..............................A689A9AA9A
   AAA99999AA995444599AA9AAAAAAA9AAA999999AAAD..............................A79A9AAAAA
   999A999999986444599AAAAAAAAAAAAAAAAA9999AA9E.....E.....,EEEE.EEEEEE.EEEEEEEA79A9ABBAA
130A99AA999999963545AABABABBABBBABABAAAA9B99E.EEEEE.EEEEEEEEEEEEEEEEEEEDDEEEA88ABABBBB
   999A999A999964455AABBBBABBBBBBHBAAAAA9AAAD.EEEEEEEDDEEEDDEEDEDDDDEEDEDDDE998CABCCCD
   99A999999999964545AABBBBBBB3BBBBHBAAAAAAAAA9E.EEEDEEEEEEEEEEEDEEEEDDEDDEDDDDD998BCCCCDD
   9999AAA9999864555AACBBBBBBBBBBBABBAAAAAAAAAD.EEEEEEEEDEEEDEDEEEEEEEEEEEEEDDE8A9CDDDDDE
   9999AA9AAA9854445BBBBHCBBCBBBBBBBBAAAAAA9A..EEEEEEEEDEEEDEDEDDDDEDEEEEEDDD8A8DDDDDDE
   9AAAA9A9AA9864455BBCBHBBBBBBBHAHBBA3BAAAAA....E.EEEEEEEEEEEEEEEEEDEDEEDDDE9A9BCDEEEE
   9AA9A9A9AA9954455BBBBHBHBBBBBBABBAAABAAAA..EE.EEEEEEEEDDCDDDDDDDDCCCBBBDD.A69AADEEEE
   A9A9AAAA9AA854545BBCBBBBBBBBBABAABAAAAAAAAA9AA99889999999AAAAAAAAABAABA999969AADEEEE
   999AAA9A999854445BBBBBBBBBAABBABAAAAAB999A.................E.....EEEEEEEDDA979AADDDEE
   9AAAAAA9AA9954455BBCBBBBBBBBBAAAAA86658AAA.........................E.EED669A9DEEEE
1409999AAA9AA9854445BBBABABBABAAA964565568AAA.............................E697ADEEEEE
   A999A99AAAA854455BBBBAAABAAA555676668AA9E..............................9A9AEEDDEE
   AAAA9AAA999853455BBBBABBABAA46899887768AAAD.............................AA99EEEEEE
   9A9A99AA999854445BBBBBBBAAAA59DDCAA8868AA9E.............................B99BEEDDEE
   9A9A9A9A99986444 5BBCBABABAAA5C...DB9968AAAD..............................9AAEEEEEE
   A999AAAAAAA845455BBBBBBABABA5A....B99699AAD..............................8AAEEEEEE
   A9AA9A9AAA9854445BBBABBBAAAA58E..EA8879AA9B..............................9AAEEEEEE
   9A9A9A9AAA985454ABBCABBAAAAA65BDCBA77689A9B..............................AAAEEEEEE
   999A9A9AAA985445ACCAAABABAAA559A99977689A9A..............................AAAEEEEDE
   99A99AAAAA984344ACCBAABAAAAA54898876658 9A99.............................AAAEDEEEE
150A9AAAA99AA984444ABBHBAAAAAAA556787666699A99E.............................EAADEEEED
   999AAAA99998444 5ACCBBBAAAAAA646777666599A99E..............................A9DDDDDE
   AAAAAA99AA84445ACCCBABABAAA645676666599A99D...............................88DDEEED
   9A9AA9AA99984444 9CCBBABAAAAA546666655599A99D..............................BDEDDDD
   AA9A9AAA99984454ABBCBBAAAAAA545666566 59AA99D...............................DDDDDD
   AAAAAAAAAA984444 9CCCBAAAAAAA645566555599ADCD...............................DDDDDE
   999AA9A9AA984444 9BBBBAAAAAAA645554555549AADE................................EDDDA9
   9AAA9AAAAA9944459CCCBABAAAAA544445455599ADEE................................998887
   A999A9A9AA973444 9CCBBAAAAAAAA66889744499BDEE..............................98788878
   9AAA9AA9AA974444 9CCBBBAAAAAAAAAAA973 3499ADEE............................9988888887
160A9AAAAA9AA9844459CCCBBBAABAAAAAAAAA73359AADEE......................999A8999889988
   A9AAAAAAAA9844459CCCBBABAAAAAAAAAA73359AADED.....................988889799877 8888
   AAAAAAAA999744459CCCBBABABBAAAAAAA96444A9BDEE.................AA99899998999897787
   AAAAAAAAAA974454ACCCCBBHBABAABBAAA9334A9AD.................AA999989899998999897788
   AAAAA9AAAAA74454ABBCBHBBBBBPBBAAAA744499BCD...............999A9998988898888988887788
   AAAAAAA9AA984444ACCCBHBABBAABABAAA93349999D..............999899979889888887887897778
   AAA9AAAAAA974545ADDCBBCBBBBBBBBABA633499A8D.............89988999888988977888 87868887
   AAAAA9A9AA964455ACCCCBCBBCBBBBAAAAA44499A8A.............7999889989888777778778887787
   9AAAAAAA99964445ACCBCBCBCBBBBBBBAAA44599A8A.............8998899988789778878887 7777
   AAAAAAAAAAA74455BCCCCBBBBBBBBABBAACDDCDDDA8D............8988888889888878888788777 8877
170AAAAAAAA99A64455BDDBCCBCBBBBBBA989A..EE..............8788888889888878777787776 7777
   AAAAAAAA99A644458CCBCBBCBBBA56658AADDB................779888878778877878777878778 6677
   AAAAAAAAAAA644558CCCBCBCBA6555669AABB................88878888788899777788877887776687
   AAAAAAAAAAA645458DDCCBCBCA455567A9999E...............77767897788888767777778777676667
   BAAAAAAAAA9645458DDCBCBCBA455558AA9AAA...............77778888778887887778877777677777
   AAAAAAAAAA9644458CCCBBCBBA455459A9999A...............66677887788877787868867776866677
   AAAAAAAAAAA54555CCCCCCBCBA45555AA9ABBC..6...........766677667767777777877878677777 78
   AAAAAAA9AA9645558DDCCCBCBA6555AAAA9AAADDAE..........6666777876688777887777777777766676
   AAAAAAAAAA96555ACCCCCBBBBCBA658A998999AA9.........866777676876778777866887767767 6677
   AAAAAAA999555555DCCCBCBCBCBBCA9999999 7668.........766667677776886687777776766857 6667
180AAAAAAAAAA955557CCCCCBBCCBBBBB9999988679D.........776677786677677776777877667766 56666
```

Digitized by Google

```
   AAAAAAAAAA964569CDDCBCCCBBCCCA999998878B........7777877876777677688676766677766666676
   AAAAAAAAAA955559DCCCCRCCBCBRBA999999998A........87777777777777666767786677667766666666
   AAAAAAAAAA96455ADCCCCCBCBCBBBA99A899998AD......7678678878777667776776667776666666666
   AAAAAAAA9985455ADCCCCCBRCCBBBA999999999BB......67786877776666666767676677867667666 76
   AAAAAAAA9995455BCCCCBCBCBBCCB9999899999AAD.....67677777776666766676766677766766666676
   AAAAAAAA9995545ADCCCBCCBCBBBB999999999AAAB.....66778777776676666676766766667676677 66
   AAAA9AAA99955546DCCCCRBBCRBBB9999999999AAADE...78679788666776766776767766776667666 77
   AAAAAAAAAA95445BDCCCCRCRBCBBB999999999AA99CE...87777788767776766677776767776776766666
   AAAABA9AAA955458DCCCCRCCRCBBB999999999AA999ADEE97877776676677766677767666677767666676
190AAAAAAAA99954458DCCCBBCCBBCCB99A99999AAAA9ADEE9776867776766666667777666776666666666 6
   AAAAAAAAAA944558DCCCCRBCCBBBBA899999AA9A99ADD977777666665776655777667666766766666666
   AAAAAAAAAA944559DCCBCB3RBBBBB99999999AAA99AAAA77888777776767665566676766676666766666
   AAAAAAAAAA945559DCCCCB39BBBBA9998999AAAA9999988787786676767676666766666666766666667
   AAAAAAAAAA954559CCCCCB3CBRBBCBA9899999AA9A998888788866676766665566666768867865666666
   ARBAAAAAAA955558DCCCBCBBBRBBCBB998999AAAA99977787877677676665556667666776666666666 6
   AAAAAAAAAAA55558CCCRCDDE.....8688899999AAA997787678776667656654466666566776676666666 7
   BAABAABABBA7789AEEEE......EEE87789999AAAAA97667777777777666666555566656666666776666666
   EE...............EEEEEEEEEDDD88869999A99A98777677876866666666555556666666666766666666 6
   ...............CCCCDEE.EDDDA89977999AA9A9766667667766666666555444566676666665656666 6
200...........E..EBD..DAD...EEE988997899AAA99776667666776666666655444456566566666766666 6
   ..............86BCCDEAEEEEDD988889699ABAA977667776766666666665545555555656666676676666 6
   ............ED4AB99EDBACDDDDA78898866AAAA877666666654555655545455455555666666666675666 6
   ............ED45799DDDARDDDDA879888778AAA776666666666566665544444435555566666667566666 6
   ..E.EEEEEEEDDC655AACDCC9CDDDA898899886BAA7676665666565565545444444555556665666666666 6
   .EEEDDDDDDDDDDDDDDEEEDA9EEE98999999979AA676666666666555556444444445455555556565666666
   ..EEEEDEEDFDDEEEEEEEEDEDBAEE99999899959A66666666566655555655544443565555665666656666 6
   E..E.E..EEEEEEEE.EEEEDEDDACCB99A999999459675666655565555665655455344555666556566666666
   ....EEEEEEEDDDDDDDDDDDDEDDAAA9A999999955777666566665655556455544444346555666565666665566
   DDDEEEEE.....E.....EEEDEEEDAB9A99A99995566766666666665556554555444356566555665566666 6
210......................EEEEEEDAAAA999AA9557766666666666555655445544445656555565556666666
   .......................EEEE..8AAA99999955776666766656555555555555434566555556657566 65
   ........................EE.EE89AA9A99995577666666656555555555455444435555555556665666 6
   ..........................E...8AAAAA9998578666666666656555555554455443556566666566666 6
   ..........................E...8AAAAA99976877656666665655555555555444334555666555556666 6
   ............................EE8AAAAAAAA76776655566566555555455544443335456655655555555
   ..............................8AAAAAAAA7676756666656646555554554443333555665555665555 5
   ..............................8AAAABABB67666666556655555555454554433445555555565556666 6
   ................................ABABBAA6767666666665645555555545444334455555555555555 5
   ................................ABBBBBB66667666666575555555554554444524455655555555455554
220................................AABBBAA6665766655656555666565545444434556555555555555545
   ................................BBBBBAA6666666666655555566664555433355555554455555554445
   ................................BBACBAA66557566555655666666555443445555555555555544444
   ................................BBCBBBB6666766555665655666655444444555655555555454545 44
   ................................BBBBBAA5556666666665556555555445444435566555555544444445
   ................................BBCBBAA666667655555555555555545444444456566555555 44544444
   ................................BBBBBAA6566666665566554555544444444445556555555454444454
   ................................BBBBBAA7666666666665555555544544444334556666555555 44444444
   ................................BCCCBAA6566666666655554455445444443346665565555544334443
   ...............................BBBBBBAA66666766665554454454444443346556555554443434434
230...............................BBBBBB996666676555555445444444554433466655555543344444 44
```

175

Digitized by Google

```
KELB4                   16-JUL-7Ø Ø8:56
  ITH=15
  Ø91      1Ø1       111       121       131       141       151       161       171       181
Ø2Ø7631125677888888888888999989999999999999999999999999999999999999999999999999999999999999999999999
   663112467778888888888888899999999999999999999999999999999999999999999999999999999999999999999999999
   653112467778888888889888889889999999999999999999999999999999999999999999999999999999999999999999999
   552112467778888888888889888889999999999999999999999999999999999999999999999999999999999999999999999
   552112467778888888888888888889999999999999999999999999999999999999999999999999999999999999999999888
   552112467778888888888888888888899999999998999999999999999999999999999999999999989899999989998899999988888
   442112457778888888888888888888999999999999999999999999999999999999999999999999999999998999899888888
   442111467778888888888888888888889999999999999999999999999999999999999999999999899989999899988888888
   342111467778888888888888888888899889999999999999999999999999999999989999999989999889988888888889888
   342111457778888888888888889888888889899999999999999999999999999899999989899998899888888888888888888
Ø3Ø342111456678888888888888888888888888998889998899999999999999899998899989998999888889888888888888888
   342111456678888888888888888889888888989989999999999999999999989999998999998898888888888888888888888
   342111456678888888888888888898888888888899999999999999999999999999999998898889888888888888888888888
   242111456678888888883888888888888888888899999999999989999999999999998899889888888888888888888888888
   242111456678888888888888888888888888888989989999999999999999999999999999988898888888888888888888888
   242111456678898888838888888888888888888889988999989999899998899988888888888888888888888888888888888
   242111456678889898888888888888888888898889989999888899999998999888899988888888888888888888888888888
   232111356678888888888888888888888888888898888888899888998999888888889888888888888888888888888888888
   242111356678988888888888888888888889888889999888999988999998888988898889888888888888888888888888888
   232111356678888898888888888888888888889988888999999889988888888888888888888888888888888888888888888
Ø4Ø232111356678888888888888888888888898889988888998888888899888888898888888888888888888888888888888888
   242111356678838888888888888888888888888889888889988999988988898889888888888888888888888888888888888
   242111456688888888888888888888888989888988889888888888888898888888888888888888888888888888888888888
   242112467788888888888888888888889888888899888888998888888888888888888888888888888888888888888888888
   2532346899ABA9888888888888888888888888888888888888888888888888888888888888888888888888888888888888888
   25799ABCDDEEEB998888888888888888888888888888888888888888888888888888889888888888888888888888888888888
   3699AABCDDE.EA98888888888888888888888889888888888888888888888888888888888888888888888888888888888
   2589AABDDDE.EA9888888888888888888888888888886666666678888898888889988888888888888888888888888888888
   2689ABCDEEE.EB998888888888888888888888888864444444446788889998888888888888888888888888888888888888
   2699AABBCCDEEB99888888888888888888888888888776323233333357888889998888888888888888888888888888888888
Ø5Ø268889 9AAABDEB9988888888888888888888888876542222333332356888898889988898888888888888888888888888888
   2688899AAABDEB9988888888888888888888888888864422222333322224688888999999998888888888888888888888888888
   268899AABBBDEBA98888888888888888888888888865332111122223222258888889989999986988888888888888888888888
   26999ABCDDEEECA9888888888888898888988888888863222111223334542246888889998999989888888888888888888888888
   269ABCDDEEE..DA98888888889888889988888888895222222235566786633588888888888888888888888888888888888888
   259ABCDDEE...DA98888888888888899888888888865222344456778886644588888888888898888888888888888888888888
   259ABCCDEEE..DA98888888888888899999999999888642224544356776676545888999999999999988888888888888888888
   269ABCCDEEE..DA9888888888888899AAAAAAA99997422245433467754554358899999999999999999999998999898888888
   269ABCCDEEE..DA988888888888889AAAAAAAAAAA8533356655589964454458 89AAAAAAAAAAAAAAAAAAA9999999999999
   259ABBCDEEE..DA98888888888889AAABAAAAAAAA965445887789AA955655899AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Ø6Ø259ABCCDDDE..DA98888888888889ABDDEEEEEEDB9655689988 9AA989968ABBBCCCCCCCCCBBBBBBBBBAAAAAAAAAAAAAAAA
   2589ABCDDDE..DA99888888888888ABD...........D755689987666799997A.............................EEEDDDDD
   25689BCDDDD..DA98888888888888AB............E96668867654458999A.................................
   24689BCDDDD..DA98898888888888AB.............B99677776555678990...................................
   25689BCDDDE..DA99888888888888AB..............AA87766555556789AC..................................
   2589ABCDDDE..DA98888888888888AB...............EE97665555555689D..................................
   259ABBCDDDE..DA98888888888888AB...............CB766655555569C....................................
   259ABBCDEEE..DA99888888888888AB................DA667666666 8A.....................................
   259ABCDDEE...DA99998888888888AB.................E67676677790.....................................
   258ABBCDEEE..DA98888888888888AB.................E6666556669E.....................................
Ø7Ø2589999AAABCCA998888888888889AB..................E6655555569E.....................................
```
176

Digitized by Google

```
   2554334577889988888888899888AB................CD7665555569D..........................................
   2442112566788889988888888889AC................CD7665556668A..........................................
   2442112566788898888888998889AC..................9666566667A..........................................
   2442112466788888888888888889AC..................B866565558D..........................................
   2554423566788988888888998999AC..................EA8666666C...........................................
   2455554566788838888888998889AC..................DA9766668D...........................................
   2566655566788988888889999889AC..................BAA977779BE..........................................
   2579875666788988888889988889AC..................CBBAABBBABE..........................................
   2586865566788888888889899889BD..................EDDDDECCAC...........................................
08025888645667888989888888888889H...................EEEEECCBD...........................................
   25889645667888888888893688889D...................E..EDDDD............................................
   25889545667788888888888888889E......................EDDDE............................................
   25889545666888888888888888899E......................E................................................
   2588854566678888888888888888A........................................................................
   2578853566778888888888888899C..........................................................EBAAAAABAC....
   25799535667788888888888888AADE.........................................................EBA99AAAABE...
   257884356669888888887838898BEEEEE......................................................ECAAA9AAAC....
   257884356669888888888889E89DDEEEEEE....................................................EBAAAA9AABE...
   2578953566788888888888888ADD.EEEE......................................................EBBAAAA9AB....
0902589953566788888888968889CEE.EEDD.....................................E................ECBBBAAAAD....
   358984356668888888888888889C...EDDDDDDDE....................EE........EE.............................
   36898435666888888899888888AD..EEDDCBBCCD....................E........EDE.............................
   36898435666798888888888889ADEEEDDDBABDDE.............................DDE.............................
   368984356667888879978989CDDDDDDDBABDDE...............................DDEEEE..........................
   357764256677888878888889CDDDDCDDBAADDDE..............................EDEEEE..........................
   245532245568888888777778ACDDDCCCCA9ACCCDEEEEE...E...................EDDEDDD..........................
   24521114666788886887999CDDDDDCCBA9ABBBDEEEDE...EE.................ECBBCDDDD..........................
   244211146667888689777789CDDDDDCCBA9ABBBCDDDD....EE................DA9ACCCCD..........................
   24531114666788879777778ADDDDDCCCBA9AAABDDDDE..EEEE..........EEEEEECA9ABBCCD..........................
1002453111455778868877779CDDDDDCCCBAAAAABDDDE..EEEE...........EDDDDDDDAAABBCCD..........................
   2453111466678887777778ACDDCCCCCCBAAAAABDDDE..EDD..........EDDCCCCDDAAABBCCD...EDDDCCCCCCCDDDDDDDDDDCC
   2454111455678877777779CDDDCCCCCCBAAABBCDDD..EEEE..........EDCCCCCDDA9ABBCCDE..EBA999899888889888898888
   1454211466778877777779DEEEDCCCCCBAABCCDDDE..EEE...........ECBBBCCCDCA9AABCCDE...A988877777777777788988
   3564211466788897777779ACDEDDDDCCBA9BDDDDEE.EE.............EECAAAABBDDAAAABCCDEE..C76665555555555555555
   886532246677887777777789ABCDDDDCBA9ACCDDE..EE.............ECAAAAABCA99ABCCDE...E966666666666666666666
   97655435666789877777766789AACCCBA9ACCDE..EE................ECCBAAAB999ABBBDE..EA866666666666666666666
   66554325556788987777766666668ABBA9ABBDE..EE.................EDCBBBA899ABDDDECA87666666666666666666666
   3333222455678888877776666555689998ABBDE.EE..................EDDDDBA99ACCDDD96666666666666666666666666
   22321114556788988777766655666777779CDDD.EEE...................EEEEDCAAABBAA865666666666666666666666666
110123211145567889787777666556667778ACDDD.EE......................EEE.DA99777655556666666666666666666666
   333211145567889886677666566677779CDDDE.............................B877677655556666666666666666666666
   34421014556778898777766556667898BD................................E9677777755556666666666666666666666
   23321113556778888777666555667889CD....................EEE.........D7677777655555666666666666666666666
   12321113556778877777666555667779CE......................EE........D8677766655555666666666666666666666
   1232111355677978777766655566678ADE..............E.......EDDDE.....E9676766655555666666666666666666666
   1242111455677887777766655555668ADE.......................EDDEE....97766666555556666666666666666666666
   1242111455677897777766655555668BDE........................EEEE....A8766666555556666666666666666666666
   1243211455677777777766655555569BDE.........................EEDE...B9666556555556666666666666666666666
   125421145567777777776666555555569CDEE........................E....D9666666555556666666666666666666666
12012542114556777777777766655555569CEEEEE............EEEEEE..........E9666666555556666666666666666666666
   12542114556777777777766655555556ADEEE..........E.EDEEEEEE.........E966566655555555666666666666666666
   1254211355677777777766665555558BDEE.....EDBBCCDEDCCCCCCCCDD.......E966566655555555666666666666666666
   1254211455677777777766665555558BD......BA9999AACBA99999999887899C.D866566655555555666666666666666666
   1354211355677777777766665545558BD...EEB654566888976666667654322335BA655566655555555666666666666666666
```

Digitized by Google

Digitized by Google

```
   12542113556777777666665555559ABA7554211112333333333333322221222235544555565555556666666666666666666
   1354211255677777777666555555456542111111111111111112222221122222224556665555555556666666666666666666
   1354211355677777777766665555554222111111111111111111222211111222233245566655555566666666666666666666
   1254211255677777776666666555555422111111111111111111122221111222223333556665555565666666666666666666666
   125421125567777767766666655445322221111111111111111112221111222223323456665555556666666666666666666
13Ø12542112556777777777666665555532222222121111111111111111122222333234566655555566666666666666666666
   1254211255677777777766666555553222221122111111111111111111222222222346666555555666666666666666666666
   1254211255677777766666655555532222211111112111111111111122222222232249AAAB75556666666666666666666666
   12552112556777777666666555555322222222111111111111111111222222222322488888866565665666666666666666666
   125421125567777766666665555554222221122211111111111111111122222332235655865555665566666666666666666
   12552112556777777666666555556422222111222221111111111111122222222245566655555566556666666666666666
   125421125567777766666665555554222222221111111111111111111112222222223456665555566665666666666666666
   125421125567776767766665555556532222222111111111111111111111122222222245666655555555556566666666666666
   1254211255677767766666655566765222222212111111111111112222222222222245668875555556655656666666666666
   12542112556777676666666555666865322222111111111111111222222122222223566788655555555556666655666666666
14Ø1254211255677776666666556888964222221111111111111111111112222224567899955555555555656555656665666
   1254211255677767677666666788876422211111111111111111111112211122222456899996555556655666566666666666
   1254211255667776676666665678876532221111111111111111111111221112222245668999655555555565656556656665666
   1255211255677676766666656777665322222111111111111111111111111112222234558999555556665556565565666666
   12442112556767766666666667665542222211111111111111111111111111222223345688755555556565656556666666
   12542112556776676666666656554542222221111111111111111111111111222223344555545555566556565655665666666
   1255211255667676766666655555454222222111111111111111111111111122222334455545555566555656666665655566
   12542112556767676666666555554432222111111111111111111111111111222222344444445555555555655566656666566
   1244211255676676666666655555432211111111111111111111111111112222233333334455555555556666566665666666
   12542112556666676776666655444432212221111111111111111111111112222232333344555556655666666666566666
15Ø125521125567767676666666554444322111111111111111111111111111112222222323334455555665565666666666665666
   124421125566766667766666555543221111111111111111111111111112222222233334555556665665666666666656655
   124421125566766766666666555543322111111111111111111111111111122223223333334455555665565666666656665666
   124421125566777676676666555544322111111111111111111111111111122222223333344555556655666656556665655566
   124421124466667676666666655554432211111111111111111111111111112222222333334455555555556565666565656566
   12442112445666767666666665555443221111111111111111111111111122222323333344555555555656566656666666
   124421124467666666666666555544322111111111111111111111111111122332333334444555555555565666666666666655
   1144211244566666667766666555554432111111111111111111111111111122223233333444455565665565666666656665655
   11442111445666766666666655555443211111111111111110111111111111222333333333445555555555555555555556565 55
   1144211144566666666666666555554432111111111111111111111111111112223333333444455555555555555665565655 5
16Ø1144211144566666666666666555554432211111111111111111111111111112223333333444455555555556555665565656555
   11442111445666666666666666555544422111111111111110111111111111223333333444455555555555655566555555555
   1144211144566666666666666655554443211111111111111011111111111122233333334444555555555555555555555555
   11442111445666666666666666555554443221111111111111001111111111112223333444445555555555555555555555555
   11343111445666666666666666555544432211111111111111001111111111112223333444445555555555555555555555555
   11343111445666666666666665555444322111111111111110001111111111122233344444455555555555555555555555
   1135311144566666666666666665555444322211111111111110001111111111122233344455555555555555555555556555555
   1134311133566666666666666665555544322211111111111110011111111111122234444455555555555555555555555555
   11343111445666666666666666655555544222211111111110100111111111111222334444445555555555555555555555555
   1135311133566666666666666665555554432211111111111100011111111111223334444555555555555555555555555555
17Ø112531112256666666666666666555555443222111111111100001111111111122233444455555555555555555555555555
   112532112256666666666666665565554432221111111111100011111111112223334444555555555555555555555555555
   11244211225666666666666666665555444222111111111111011111111111122233444455555555555555555555555555
   112432112256666666666666666655555443321111111110111111111111111222334444555555555555555555555555555
   11243211225666666666666666556555444332111111111111111111111111122333444455555555555555555555555555
   11244211225666666666666666655555443322111111111111111111111112223344445555555555555555555555555555
   112442112256666666666666666655555443321111111110111111111111111122334444555555555555555555555555555
   1124321122566666666666666666555554433221111111011111111111111111223344455555555555555555555555555555
   112542112256666666666666666555555554442111111110111111111111111122344445555555555555555555555555555
```

178

Digitized by Google

```
   112442112256666666666666666666655555444321111111Ø11111111111111223444455555555555555555555555555555
18Ø112542112256666666666666666666655555444321111111Ø1111111111111122344445555555555555555555555555555
   1124421122566666666666666666666555554444311111111ØØ1111111111111122334455555555555555555555555555555
   112442112246666666666666666666555555443211111111Ø11111111111111122344455555555555555555555555555555
   11244211224666666666666666666666555555443211111111Ø1111111111111122344455555555555555555555555555555
   1124421122466666666666666666666555555443211111110Ø1111111111111122344445555555555555555555555555555
   11144211114566666666666666666665555554432111111ØØ1111111111111112234445555555555555555555555555555
   1124421112466666666666666666666555555443111111ØØØ11111111111111122344455555555555555555555555555555
   11144211114566666666666666666665555554432111110ØØØ1111111111111112234445555555555555555555555555555
   1114421111456666666666666666666555555443111111ØØ111111111111111112234445555555555555555555555555555
   1113421111456666666666666666666565555544321111111111111111111111122344455555555555555555555555555555
19Ø1113421111456666666666666666666565555544321111111111111111111111122344455555555555555555555555555555
   11134211124566666666666666666665555555444111111111111111111111111223444555555555555555555555555555555
   111342111245666666666666666666655555544321111111111111111111111122344455555555555555555555555555555
   11134211124566666666666666666665555555532111111111111111111111111223445555555555555555555555555555555
   1123421111456666666666666666666555555443211111111Ø11111111111111122344455555555555555555555555555555
   112442111245666666666666666666655555544321111111ØØØØ1111111111111122344455555555555555555555555555555
   23345311113566666666666666666665555555532111111111Ø1111111111111122344555555555555555555555555555555
   333453211145666666666666666666655555544321111111111111111111111111333445555555555555555555555555555555
   44445311113566666666666666666666555555542111111111111111111111111233344555555555555555555555555555555
   444453111235666666666666666666665555554432111111111111111111111112333445555555555555555555555555555555
22Ø4444542112356666666666666666666565555544321111111111111111111111123334455555555555555555555555555555555
   5555542112456666666666666666666656555555411111111111111111111111123334455555555555555555555555555555555
   55555421113566666666666666666665555555541111111111111111111111111233344555555555555555555555555555555
   5555542111356666666666666666666555555544421111111111111111111111123334455555555555555555555555555555555
   55555421113556666666666666666665555555555421111110ØØ111111111111112333445555555555555555555555555555555
   555554211135556666666666666666555555555555422111111111111111111111123334555555555555555555555555555555
   555654211125566666666666666666555555555555422111111111111111111111123334455555555555555555555555555555
   66666421113556666666666666665555555555555422111111111111111111111123333455555555555555555555556656666666
   7676642112355566666556665655555555555555321111221110Ø1111122112333455555566666667777777777888888888888
   978665211135555565566666666666666666655422222222ØØØ0111112322345556667777777778888877888888888888888
21Ø778876543355666666666666766666666666665422222221101112233223456666767677777777778888768898998888899
   6666666666666666666666666666655555555432222222110111223322235666666667667777777777777777777777777777
   666666666666666666666666656555555555555322222222110111112332234556666666666677777777777766667666666
   65666666666565565666555555555555555555432222222211111111222323445666666666676767677666766666676766666
   55556565666555555555555555555555555554222232222112211112233333346666666666766766667666766667666766666
   555565656665565655555555655555555555554222232222223332222333333334566666666676766666666666666666666677
   565656666665666566655565555555555555554333232222344443332222333335666666666676667666666767667777776777
   5656666666666666666665666666666656555543333222224555554432223333335666667777677777777777777777777777777
   66666666666666666666666666666666666554333222334566665665333333333356667777777777777777777777777777777
   66666666666666666666666667656666667665332235556666666666654332322366777777777777777777889777776777777
22Ø666676666667676777767777777777777776555566776776767667666555555577777777777777777777897888777777777
   76767777777777777777777777777777777777777777777777777777777777777888777777777777787889997777777777
   777777777787788988878989888487989489989886887699786778887887788889888788877868899777788977788777777
   988888888888888888888888889988889899888888899888886888878887888888887888887888888887887777777778677
   8898888888878888888888888888888888888888888888888889888888888888888888688888888866897967777867777777
   888888888968888888888888888888838888838888888888888888888898888888888888888868888888868888988888877886987777
   888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888789987777
   888888888888888888888888888888888888888888888888888888888888888888888888888888888888888888886878899
   688898888898888888888888888888888888888888888888888888888888888888888888888888888888888998886889888
   889888999998888888888888888888888888888888888888888888888888888888888888888888888888888888896889888
23Ø6888899888988889988888888888888888999888988888888998888888888889989998988888888888888888888888888888888
```

Digitized by Google

APPENDIX B

Two pictures of heads: Input data

Digitized by Google

Digitized by Google

```
KELH1                           17-JUL-70 15:29
 ITH=15                                                                                                                D
 Ø5Ø       Ø6Ø       Ø7Ø       Ø8Ø       Ø9Ø       1ØØ       11Ø       12Ø       13Ø       14Ø       15Ø       16Ø   D
    ....E..........E....E.....................................E.EE..E......................E...EE..E..EE.........EE.....E.
    ....E..............EE............E.E...............E...E..E.E.E.E...E.......EE...E......E.E.E.....E...E..EE.EEEEEE.EE.EE.E
    ..E.....EE.....E.EEE...E..E.......E...E...E.....EE.E.E...EE.EE...E.E.E..........E.EEE..EEEEEE.E.EEEEEEE...EEE.EEEE.E.E.
    .....E..EE.E.........E..E.....EE..E..EEE...EEEEEE..EEE...EEEEE....EE..E.E...E..E......E.EEE..EE.E.EEEE.E.EEE.EEEEEEEEEE
    .............E....E....E.EE..E..EE.EEEE...E....E..EEE...EEEE.E....EEE.E.E...EEE.E..E..EE.E.EE...E.EEEEEE..EE.EEEE.EEEEE.E
    .....E.E..E.E..E.EEEEEEEE.EEEEEEE.E.....E.E.EE..EE.EEEE.EEEE.E..E.EEEEE.EEE.E.E.EEEE..EEEEEEEE.EEEEEEEEEEEEEFEEEEEEEEE
    ..E.EE.EEE..EE.EE..E.EEEEE..E..EEEEEEEEE.EEEEEE.EE.EEEEE.E.E.EEEEEE.EE.EEEEEEE.EEEE.E.E..EEEEEEE.EEEEEEEEEEEEEEEEEEEE
    ...EE.E.EEE..E.EE...EE.E.E..EEEEE.EEEEEE.EEE..EEE.EEEEEEE..EEEEEEEE.EEEEEE.EEE.EEEEE.EEE.EEEEEEEEEEEEEEEEEEEEEEEEEFEFEE.
Ø3Ø.EEE....EEE.EE.EE...EEE.EE..EEEEEE.EEEEEEEEE..EEEEEE.EEEEEEEE.EEEEEE.EE.EEEEEEEEEEEFEEEEEEEE.E.EEEEEEEEEEE.E.EE.EEEEEE
    ..EE..E....E..EEEEEEEE.EEEEE.EEEEEEEEEEEEE.EEEEEEDEEEEEE.E.EEEEEEE.EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE.EEE.EEEEEEEEEEEEE.EE
    EEEFEE.EEEEEEEEEEFFEEEEEEEFEFEEEFEEEEEEEEEFEE.EEFEEEEEEFEFEEEFEFEEEEEEEEEEEEEEEEEEEEE.EEEEFEEEFEEEEEEEEEEEEEEEEEEEE.FEEEEEEFEFE
    .FEEE.EEEEEE.E.EEFE.EEEEEEEEEEEFEFEEEEEEEEEEEFEFEEEEEEEEEEEFEED.EEEEEEEEEEFEEEEEEEEFEEEEEFEEEEEEEEEEEDEEDEEE.DEEEEDEEEEDEEEFE
    EEEEEE.EEE.FEEEEEFEEEE.EEEEEEDEEEEEEEEEEEEEEEEEEDEEEEEEEEEEEEFEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDEEEEEEEDDEEEEEE
    EEEEEEEEEE.FEEEE.EEEEEEEEEEEEEEEEEEEEEEEEDEEEEEEEEEEEEDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDEEEEEEEEEEEEEDEEDEEEEEEEEEED
    EEEEEEEEEEEEEEEE.EEEEEEEEEEEEEEEEEEEEEDEEEEDDEDDDEEEEEEEEDFDEEEEEEEEEEEEEEEEEEEEEEFEEEEEEEEDDEEEEEEEEEEEEEEEEEEEEEEEDEDDEEE
    E.EEEFEEEEEEEEEEEEEEEEEEEEEEEEEEEEDEEEEED.EEEEEEEEEEEEEEEEEEEDEEEEEEEEEDEEDEEEDEEEDEDDEEEEDEEEEEEEEEEEEEEDDDEDEEEEEEEEEEDEEEE
    EEE.EFEEEEEFEEEEEEEEEEEEEEDEDDEDEDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEFEFEEEEEEEEEEDDEEEEDDDDDDDDDFDDDEEEEEEEEDEEEDDDEEEEDEDDEEEEEEED
    EEEEEEEEEEE.EEEEEDDEEEDEEEDDEDDEEEEEEEEEEEEEDEEEEEEEEDDEEEEEEEEDEEEDEEEDDDDDCBBBBBBBBBCCDDDDDDEDDEEEEEEEDDEEDEEDDDEEEEEDE
Ø4ØEEEEEEEEEEFEDDEEEEEEEEEDEEFEEEDDEEDDEEEEEEEEEEEDDEEEEEEEEEDDEEEEDEEEDDEEEEEEDDCBAAAAAAAAAAAAAAAAABBCCDEDEDEDEEEEEEEDEEDDEEEEDDEE
    EEDEEEEEEEEFEEEEEEEEEEEDDEEEEEEEEEEEEEEDDDEEDEEEEEEEEEEEEEEDDEEEEEDDDECBABAAAAAAAAAAAAA99A8A99AAACDDEEEEEEDEDEEEEDDDDDEDDED
    EEEEEFEEDDEEEDEDDEDDEEDEDDEDEEEDDEEEDEEDDEEEDEEDEEEDEDEDDCCBBAAAA9AAAAA99A9AAA9A9989999ABCDEEEEEEDDEEDEDDDDEDEEE
    EEEEEEDEEEEEEEEDDEEDEDDDEDDEDEEEDDEEEEDDEEEDDEEDEEDEDDDDDDCBAAA9AAA9AAAA99999988999898999987899ABDDDDDEEDDEDDDEDDDDD
    EEEDEEEEDDEEEEEDEEDDDDDEEDEEEDEDEDEDDDDDEDEEDDEEEEDDDDDDDCBAAA99AAAA9A99AAAAAAA999998899999998888999BDDDDEDDDDDEDDEED
    EEDEEEEEDDEEDDDEEEDDDEDEDDDDDDDDDEEDEEEDEDEDEEEEDDEDDDDDDA99AAAAA9999999AAAA99998998898888988879888989999ACDDEEEDDEEDEEDD
    EEDEEEEEEEDEDDEDDDEDEDEDDDDEDDDDEEEEEDEEEDEDDEEEDEDCBBA9999999999999999999999999988988988988998989888998ACDEDDDDDEDDED
    DDEEEEDEEEEFDDDEEDDEDDDDDDDDDDDDDDEDDEDEEDDDDDDDDDCBA9999AA9AA99999999999999989888888898998898887888888899BDDDDEEEDDED
    EEEEDDDDEEEEDDEEDDDDDEEDEEEEDDDDDEDDDEDDEDEEEEDBA9999999AA9999999988998999999999989899989888888888877889ACDEDDDDDDD
    EEEEEEEDDDEDDEDEDDDEDDDDDDDDEDDDDEDDDDDDDDDDDDDEDA9998999999999988998999899988898988898998887887778887787789AACDEDEDDE
Ø5ØDEEEDEDEEEEEDDEDEDDDDDDDDDDDDDDDEDDEDDDDEDDEDDDA89999889999999999889997888998A9A99998888888889888778878778899ACDDDED
    EEEDEEDEDDDDDDEEDEEEDDDDDDDDDDEDDDDDDDDDDDDDEEB99999999988999877879887899878999999998888888888788888877787 8AAAABCDDD
    DDEDEDDEDDDDEDDDDDDDDDEDDDDDDDDDEDDDDDDDDDAA988889899899898988989888988887888888899898888878888998888877869899AACDD
    DDDEDDDEDDEEDDEDDDDDDDDDDDDDEDDDDDDDDDDDDDA99887898899998988988898888888899877877888878888788878887788778677668 9AAACD
    EDEDDDDDEEEDEDDEDDDEDDDDDDDDDDDDEDDDDDDDDC988888888899998889888879888888888888786678888777887878888777676777 7789AAAB
    DEEEDDDDDDDDDEDDDDDDDEDDDDDDDDDDDDDDEDDDDB9887788888888877878778888888788888878887778898888877766767787687677878889888
    DEEDDEDDDDDFDDDDDDDDDDDDDDDDEDDEDDDDDDB978777878888788787777788988888887887888889987988889887678778777777887878788
    DDDDEDDDEEDEEDDDDDDDDDDDDDDDDDDDDDDDDDDA778899989887777767888888788887888888668888888869787788978887777767777777676777
    EEDEDDDDDDEEDDDEDDCCDDDDDDDDDDDDEDDDDDDDA77888888878776777778777776776776787777677887778878887898877888777777766766667
    EDDDEDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDA777878778887776667888777666766766777787678887778789887888888888777776677766 77678
Ø6ØDDDDDDEDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDB788888888878777778787776766767778888887888877777878788886778788877877676676667 66877
    DEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCC68888787778867778867767777788888887888877878877767678888887778888777777788676 7777
    DEDDEDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDAA7887868778876778887676787777888878887787877667677766778888777888877877877877777768
    DDEDEDDDDDDDDDDEECCDDDDDDDDDDDDDDDCAA8887777887867877777767777888877778896766677778788777776888788778787786777 6778776
    DDDDDDDDEEDDDDDDCDDDDDDDDDDDDDDDDDDA8887777887777678677666778777787866666666767766778788899977887877687778769877 77878
    DDDDDDDEDDDDDDDDDDDDDDDCDDDCCDDDDDB8787776777668777767777767677667777677676666667666778787888987777888788887776677 76677
    DDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD97887677787887677666777677666777676766777767766667788787788877777778678767777667 7667
    DDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDB877667778787777866776677666776676766787876687786677666666677877777767676666666 76777
    DDDDDDDDDDDDDDDDDDDDDCEDDDDDDDDA766696678776666776766776766666776667766766866766776676667666666667777677677777766 77766
    DDDDEEEDDDDDDDDDDEDDDDDEDEDDDDDD7666676788766687777777776676676667767777667766666656666767777676667887677777887767 6667
Ø7ØDDDDDDDEDDDDDDDDDDDDDEDDDDDDDDA766666686667776767676777676767666677767677766676666676566677666666677787667767777667 6777
    DDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDB76666667666676656676666666676666767677676666766667766767766667676677887786777767 77767
    DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCA76666667767666766677677676676677676766666767677777667676666788866666788767777767 7676
```

181

Digitized by Google

```
    DEDDDDDDDDDDDDDCDDDDDDDDDDCCDC86566666666666666666666676676766666666666666666667677776666667768877676666777777666888776
    DDDDDDDDDDDDDCDDDDDCDCDDDDDDDB766666666667666667666676666666666666666666666678666666666776878887887767777667787867 6
    DDDCDDDDDDDDDDDDDDDDDDDDDDDDDA66666555676666777677666766676666666666666676666676666666667677768888888876666766677896 76
    DDDDDEDDDDDDDDDDDDDDDDDDDDDDD96576666558787777778766666666656666666665666666666666666677667777888877777667769788996 76
    DDDDDDDDDDDDDDDDDDDDDDDDDDDC6676666669AAA99AAAA766666676666666666566666556666666666677667666776777788987667788878877 76
    DDDEDDDDDDEFDDDDDDDDCDDDDDDD67676666 8BDCCCCDDDC87676666666666666666656666665667777666767776767768889987668777988 7777
    DDDDDDDDDDDDDDDDDDDDDCDDDCDDB67777 6799CEEDDDDDE,BAA676666666666666666656666666676666667676666677889999877677787867777
080 DDDDDDDDDDDDDDDDDDDDCDDDCDCCB6776678ADDEEDDEEE,EECDBA7767666665566666665566666666666666676666777889889A987887888777787
    DDDDDDDDDDDDDDDDDDCCDCDCCDDCC8777 6679ACEEEEEE,....EEDDAA87667666667666666676666666656766666666888889 99AAAA988999998787
    DDDDEDDDDDDDDDDCDDCCDDCDDB867666679ACEEEEEEEE,...EEDCC966666677677776676665666566666666668888 9999AAAAAAA9999A9888
    DDDDCDDDDDDDDDDDDDDDCDCDCCAA76766789ACEEEE,EEEE,..EEEEDA766666888998777766766666776666667777888 99AAAABAABABBAABBAA988
    DDDDDDDDDDDDDCDDCDDDDDCDDD88676777789ACEEEE,EEEE,...EEDDE987899AABBA99788887867676666676787798899AAAAARABBRBBBACBBA99
    DDDDCDDDDDDDDDDDDDDDCDDDD886776688AACEEEEEEE,.....EE,..EECCCCDBBCCCB9999999878776666777888889 9AAAABBBABBBCCCCBCCCCAA9
    DEDDDCDDDDDDDDDCDDDDCDDDCA7777777899ACDDDEEE,.E,.E,..EE,EEEDEDDDDDDCAAAAAA9999868877889898999AABABBBBCBCCCCCDDCDDDDCBA
    DDDDCDDDDDDDDDDDDDDDCCDDCDA777667788AACDDE,DDEEEE,....EE,.EEEEEDDD,EDCCBBBBBA9999779 8AA99AAAABBBBCCCCCCCDDDEEEEEEDDDB
    DDDDDDDDDDDDDDDDDDDDDDDCD9776667779AACDDDEEEDD,...E,....EEEEEEEEEEEEDDCCBCBBAAAAAAA9ABBBBBCCCBCCDDDDDDDDDDDEEEEEEEDDDB
    DDDDDDDDDDDDDCDCDDDDDDCDB7666667776 89AADDDEEEEEE,E,..E,....E,EEEEEEEEEDDDDDCBCBBBBBBCCCCDDDDDDCCDEEEEE,EEEDEEEE,EEEEB
090 DDDCDDDDDDDDDDDDDDDDDCCDDB866666767789AACDDDEEEEE,...........EEEEEEEEEEEEDDDCDCCCDDCCDDEDEDDDDDDEEEE,E,E,EEEEEEEEEEC
    DDDDDDDDDDDDDDDDDDCDDDCDDCA7666676679 9ABDCDDDEEEE,........EE,EEEEEEEE,EEDDEDDDDDDDDDDEEEDEEDDCDEDDE,......EE,...E,E,DD
    DDDDDDDDDDDDDDDDDDDDDDDDDDB76667667789AACCCDEE,.E,..........EEEEEDDEEEEEEDDDDEDDDDEEEEDEEEDDDDDDDDDDEDEE,............EC
    DDDDDEDDDDDDDDDDDDDDDCCDCA76666676788 99CCDDEEEEE,.........DEEEEDDEEEDDEEEDDEDDCDDDDEFDDDDDDDDDDDDDEEEEEEE,...........ED
    DDDDDDDDDDDDDDDDDDDDDCDCDCA866666667 8899CCDDEEE,..........EEEEEDDDDDDDDDDDDDDDCCDDEEDDDDDDDCCDDDDDDEEEEEEE,..........ED
    DDDEDDDDDDDDDDDDDDDDDDDCDCCB955676767789 9BCCDEE,...........EEDDEDDDDDDDDDCDDDEDDCDDDDDDDDCDDDCCCDCCDDDDDDDEEE,........EED
    DDDDDDDDDDDDDDDDDDDDDDDDCCA6666667868 99BCDDEE,........EEEEEDEDDDDDDCCCCDCDDDCCDCDDDDDCDDDCCCCCCCCDDCDDDDDEEEE,.......ED
    DDEDDEDDDDDDDDDDDDDDCDDDCA6656666778 99BCDDE,....E,EDEDDDCDBCCCCCCCBBCCDCDDDDCCDDDDDCCDCCBCCBBCBCCCCCCCDCDDE,.......ED
    DDDEDDDDDDEDDDDDDDDDDDDDA665666777899CDDEEE,.E,ECBBCAABBBBBBBBBBBBBBBCCDDDCCDDCCCCCCDCBBBBAABAAAAAAABBABBCDDE,.....EE
    DDDDDDDDDDDDDDDDDDDDDDDDCA77666677789 9CDDEEE,.EDCAAAA9AAAA9989AA8AAABBCCDDCDCCCDCCCCBCBBAAA99899999999AAAAABAD,.....EE
100 EDEEDDEEEEEEEEDEDDDEDDDA9966767776899DDEEEE,..EDAA99999898887877889 9ABBCCCCDCCCCCCCBCBAA999797788899 9AA99BBCE,...EE
    EEEEEEEEEEEF,FEFEEEDE,EEDAA6676667899DDEDEEEEDBA99998877767667779 89ABACBCBDDCDDCCCBBBBAA99877777777898999ADCDE,E,E
    ..EE,.EE,...E,E,EEEEEEEEEDBB6666677899DEEDDDBBBAAA98877867677787899AABBBBBCECDDDDCBBBBAA999887768777898998999BBDD,.E
    ......E,.....E,EEEE,EE,EDBB76676778 9EDEDDDA99A999988777898998899999AAAAACDDDDDDDDBBBBAAA9989989999889 8AAAABBCDD..
    ..E,...E,.....E,...EEE,E,EDD96966687ADDEDCCAAAAAA989899999988888999 9A9AAAABCDEEEDDDCAAAAA999888788999999AAAAAABCDDEEE
    .E,...E,...E,..E,EEEE,EEEEEE966666788EEEEDDAAAAAA98999998777777799A999AAACDE,EEDDDCBAA99999776776888 9AAAAAAABBCD..E
    E,..E,.....E,.EEEEEE,EEFEFEEA566676 8BDEEDCBAAAAAA9999887676777777899AAAAACDEEEEEEEBAAA8988787776776789 9ABAACCBCDE..
    ....E,..EE,E,.EE,EEEE,EEEEEEA6767678ADEEDCAABBAAAA98766767778888788899 9AAADDEE,EEEECA9988888888878777678 89ABBCBCDDE..
    ..E,..E,EE,..E,E,EEE,EE,EEEED666677 8ADEEDBABBBBBA986788789987999799899 9AAADDEEEEEEECA9889888999798997899789AACDDDEE..
    .........EE,E,E,EEE,E,E,.EEB6667779ADEECBABBBAB977699968778 79AA98999A9AACCDEEEEEEEBA9889998 9AA7866679BD9689ADDDEDDE.
110 E,....E,EE,EEEE,EEEEEEEEEEEED6676778ADEDCBBCBBAA77689997767889BB9878AA9AABCDDEEEEEEEDA98A88889BA887668AD,977999CDDDDE.
    ......E,EE,..EEE,EEEEEE,E,EEE7576679ADEEDDBBBBB8877 8ABB889888ABBA8899BAAACCDEEEEEE,DBA9989999BA88888 9BDCA88999BDDEE..
    ..E,..E,..E,EEEEEEEEEEEE,EEEEA6676899EEEDCCCCCA889999AAA89889AAA99AAAABBBBDEEE,....DCAA9999999AA88889AAA9ABAA9BDDEE..
    ...E,..E,....EEEEEEE,EE,.EEEDB76777AAE,EDDCCCCA9A9BAAAAB999AAAAAABBCBBCBBCDE,......EDBAAABBBABABAAAAAAAABDDAAABCDEE..
    .E,E,EE,...E,.E,.EEEEEE,EEEEEDA9777AAEEEDDCCBBBBBBBBBBBBBABABAAAABBCBACCDDEEE,.....EECBAAAABAAAABBAABBBCDDDCBBBDDD...
    ....E,.....EEEE,.EEE,EDDDDDDBA767BBE,EEDDCCCDEDDDDCCCCBABAAAAAABBBACCDDDDE,E,....EEECCABBBAA9AABAABCCDDEDDDDCCDDE..
    .E,EEEE,EEEEEEEEEEEEEEEECBAABDC8878AAEEEEEDDDDEEED,EDDDDBBBABABBCBBBCDDDDDEE,.......EEDDCBBBBBABAABBACDDDEDDDDDDDEE..
    DEDDDDDDDDDDDDDDDDDDDDDDBA999ADDB878DDE,EEEEEEE,EEEEEEDDDCCCCCCCCBCDDEDEEDEE,.....EEEDCDDBBCCCBBCCBCBDDEEEEEEE......
    DDDDDDDDDDDDDDDDDDDDDDDDA99999CDC868DDEEEE,EE,..EEEEEEDDDDDDDDDCDDCCDDEEE,EEEE,........EEEEDDDDBCCCCBBCCCCDDDEEEEE.......
    DEEEEEEEEEEEEEEEEEEEEEEED99988ABDA979DDE,E,E,...EE,EEEEEEEEDDDDCCDDDDEEEE,EEEDE,.........EE,EEDDDCCBCCCCCDDDE,EE,.........
120 ......................A98A99ACCD869DDD,E,........E,EEEEDDCDDEEEEE,EEE,.DDEE,.......E,..DEDDDCCBCCCCDDDEE...........
    ......................99AADDCCA79DDE,E,..........EEDDEDDDDDD,.E,..EE,EEEE,.............EEEEDDDCCBCCDDEE..............
    .....................899AADECBA79DDDEEE,.......E,.EDDDDEEEEEEEE,..EEEEEED,............EEEEEDDDCCDDDDEEEE............
    ......................9ABBEEBAA79DDD,DE,......E,EEEEDDDEEDEEEEEEEEEEEEEEDD,.............EEEEEDEDDDDDDEEE............
    ......................9ACCEEDCA79DDEEEEE,........EDDDDEEEDEE,..EEEEEEEE,...............EEEEEEEEEDDEEE,E,.........
    ......................AADDEEDCA89DDDEDEE,........EEEEEEDEDEE,.EEEEEDEEEE,.............EEEEEEEEEEEEEEE,EE,EEE,E,.....
    ......................ABDDEDBA889CCDEEEEEE,.E,..EEEEEEEEEEEEEEEEEEEEDEEEEE,..............EDDDEEEEE,EEEEE,...EE,.......
```

Digitized by Google

```
   ........................BBDDECA9779BBDDDEDEE..EE.EEEEEEEEEEEEEEEEDDCDDEEE...EEEEEEE.....DDCEED.EE...E..E.........E..
   ........................BCDDECA9889BBDDDDEEEEEE.....EEEEEEDEEEEDEDDDD.EEEEEEEEDDDDD......EDCDDDEEE.....E...EE.......
   ........................BCEEECA9889AADDDDDEDDDEE...E.EE..EE.EEEEDDCDDEEEDDDDDDDCCCDE.EE...CBDDEDEEEE.....E.........E.
130........................BEEEBA9989AADDDEEEDDDD.....E.....EEEDDDDCDDDEDDDCDDDDCDCCDEEDEEE.CCCEDDD.EEE..E.........E...
   ........................AEEEBA99999ADDDDEEEEEE...E..EE...EEEDDDCDBCDDCCBBCCDCDCBBCDDBCDEEDBCDDDDEEEE..............E.E
   ........................AEEEBA99999ACDDDEDEDDEEEE..E.....EEDDDDDCCBCDCBAA7ACDCDBBBCB769CDEDCBCCDDEEEE...............
   .........................DDEAA9AA98BDDDDDEDDDDEEE..EEEE.EEDDCCCBCCBCBA7557ACDDBBBC6679BDDDCBCCDDDDDEEE........E.....
   .........................DD.AA9AA99ADDDDDEDDD..E.EEEEEEEDDDCDCCCCBBBA976689BCCBBBC8889ACDDBBBDDDCDDEEEE.............
   .........................DDEBAAAA88BDDDDDDDEEE.EE.E...EEEDDCCCCBCDBAA98889AABBABBAAAAABBBBBCBBCCCDDDEEEE............
   ...........................EBAAAA8889DDDDDEDDEEEEEEE...EEDDCDBBBCDDAA99998AAAA99AA9BBAAACDDCCBCBCCCCDDDEE............
   ...........................EDABDD9778DEEEEDEEEEEEEEEEEEEECDCCCCCDDCAAA9AAA99999889BBAABDDDDDCCBBCDDCDEEE............
   ............................ECE.DA778DDEEEDDDEEEEE.EEEEDDCCCCCCCDDCCABAABA98988999ABABCDDDDEDCBBBCCCDEDE.............
   ............................DEEEDA778DEEDDDDDEDEEEEEEEECCCCCCDDDDDCBBBBBBAA9898999ABABDCDDD.DDBBBCCCDDEE........E...
140.........................ACDEDB888CDDEDDDDDEDEEEEDDCDCBCCDDDDEDDCCCCAAAA988889ABCBDDDDDEEDDCBCCBCDEDE.....E.....
   .............................ACDDB997BDDEDDDDDDEDEEDEDCCCABCDDDEDDCBCBBBBBA9889889ABBBCDDDDDDEDCCBBCDCDDEE..........
   .............................CDDDA887BCDDDDDDDD.EEDDDDCCCBCCDEEEDDCBBCDABA989999999ABCCCDDDCEEEDCCCBBCDDD.....E....E
   ..............................EDDAAA7ADDDDDEEDEEEEEDDDDDCCDCDDDDDCDCBBCCBBB9999AAAAABBCCDDDDDDEEDDCCCCDDDD.....EEEEE.
   ..............................EDBBAA9ACCDDDDDDDEEDDDCCDCDDCCDCCCCCBCBBBBBAA998999ACBBCCDCCDDDEDDDDDCCCDDEE...EEEEEE.
   ...............................ECCAABBCCDDDDDDDDDDDDDDCCDDCCCDDDCCBBBBBBBCBA9AAAABDDDEDDDDDDDDCDDDDCCDDDD.EE.EEEEDD.
   ...............................EECCCCACCDDDDDDDDDDECCCDDDCBDBDDDCCBCBCDDCDCCCBCCCDCBDDDDDDDDEDCDBDDDDCDDEEEEEEDEDDE
   ................................EDDDBACBDDDDDDDDDDDCCCCDDBBCCCCBBBBCCCDABBABBCCDDBAABABBCCDDDDDBBCCDDDDDDDEEEEEEDDEE
   .................................CCCAACBBDDDDCDDDCBCDDDCCABBBAAABCCCBABAA99A9AAAAAAAAAAAAADCDEDCBCCDEDDEDDEEEDEEDDEE
   ..................................AAA9CBCCCCCDDCCCDDDDDCPABBBAABBBBAA999999899999989989AABCDCBABBBCDDEEEEDDDDDDDED
150...................................CCECCCCBCCCDDCCDDDDDBBAAAABBBBA99878877667778877666667799ABBAAAAADDDDEDDDEEDDDDDD
   ........................................8CBBBCDCCDDDDDCBA99AABBB98866688888768778889887777889988AAABDDEDDDEDDDDDEE
   .........................................BCCCCDCBCCDCCCBA999AAA9777899AAAAAA9A9AAAAAAA9A99989765899ABCDDEDDDDDDDEDDE
   .........................................BCCCCBCBCDDCCBAA8867886789ABBCCBBABABBBBAAABCCCBBCBCA968999BCDEEDDDDDDDDDDD
   .........................................BCBBCCCBBDCCCB998668889AACBBCBBBAAAABBAAABBBCBDDDDDCBA78889BDDEDDEDDDDCDDD.
   ..........................................BCCCCBBCDCCCB98979AAABBBBABBAAAAAAAA9AAAAAABBCBBCDDCBA9999BBDDDDDDDDDDDD.
   ..........................................BBBBBCCCCCBBB9999ABBBCBABAA9999988998999999AABBCCCDDDCAA999ABCDDDDDCDDDDCD.
   ..........................................BBBBBBBCBCBBA9AAABCBBBABAA999998888888899A9ABBBBDDCDCBAAAABBCDDDDDCDCDCDD.
   ..........................................CBBBBCCCBBAAA9ABBCCCCBBAAAA9998888788888899ABBCCCDCDCBBBBBABBCDDDCCCCDDCDD.
   ..........................................BBBBBBCBBBAA99ABBBBBBBAAAAAA99987888888889AABBBCCCCDCCCBAAABBCDDCCCCCDDDCD.
160.......................................BAABBBBBBAAAABBBBCABBBABAA999998788898899AAABACCCCCBCBBBBABBCDDCBBCDDDDCD.
   ..........................................8AABBBBBBAAAAABBBBABBBBAAA998998888899999AAABAACCCBBBBAAABCCDCCCBCCDCDCE.
   ..........................................8AABBBABBBAAABBAABBAAAAAA999999889888889AAAAABAABCBBBABAAACBBCCBCCCDDCCD..
   ...........................................AAAABABBBAAAABABBABBAAAAA8999888989888999A9AABBBCBBBBBAABCBBBCBCCCDDCDC..
   ...........................................AAAABAABAAABBBAABAAAAABAAA9999998999999999A99ABBACBBBBCAABBCBCCCCCCDDCDD..
   ...........................................AAABAABAAAABBBAABAAABBABA9AAAAA999A9AA999AAAACCBCCBBBCBBABDCBBCBBCCDCCD..
   ...........................................AAAAAAAAAAAAABBBBBAAAABBBBAAAAAAA999999A9ABBBBBACCBCBAAACCBCBCBBBCCDCCD..
   ...........................................99AAABBAA99ABCABABBBBBBCBBBBAA999899999AABACBBBACBBCCBAABCCCCBCBADCDDDD..
   ...........................................AAAAABAAAAABBABBACCCBBABBBBBBAAA99999999AAAABABBBBBBCBBBBBCCBBBBCCBCDCDD..
   ...........................................AA9AAAAABBBAAABABCBBCBAABAABAAAA9999AA9AAABAAAABBBBCBCBBCBBCDBCCCCDDBCD..
170........................................AAA99AAAAAAAABAABCAAAAA9AAAAAAAA899999AAABBAAAABBBBBBCCCBACCCBBBBCDDCCD..
   ...........................................AAAA99AAAAAAAABBCBBAAAAABAAA9A9989999999AAAAAAAABBBCDBBABBCCBBCCCCDCCD..
   ...........................................BBA999AAAAAA9A9ABAAAAAAAAAAAA9A99998999AA9AAAAAAAAACBCBBBBABBBCCCCCCCDE..
   ...........................................AAA99AAAAAAA9A99ABAAAAA9A9A9999989899999AAA9A99AAABCCCAACBBBBDBCCCDCDDE..
   ...........................................AAAA9999A999AA99AA999A9AA999999999989999A9999999AABBCBBAABBCACBBCCCCDCDE..
   ...........................................AAAA99999AA9AA999A999A999A9999899888999999999999AAAACBBAAAABBBBCBCCCCCDDD..
   ...........................................AAA9A999999A9A999988999999999998888889999999AA999A9ABAAABBBBCBBBCCCBDCDDD..
   ...........................................88A9AAA99AAAA9A9998898989999999998899998999A9A99AAAAAAAAAABBBBBCBBCCCDDD..
   ...........................................AAA99A99A99AAA9999889999999999999888899999A9999999AAAAAAABCBACBBCCCBDDDD..
   ...........................................BBAAA9998AAA9A999999889998899888998888999989899999AA9AABBBBBABBBBBBCDDDD..
180........................................88AAAA99999A89989888899988899898889888898898999999AABAABBABCBBBBCCCDDD..
```

Digitized by Google

```
   ........................BBDDECA9779BBDDDEDEE..EE.EEEEEEEEEEEEEEEEEEDDCDDEEE...EEEEEEE.....DDCEED.EE...E..E.........E..
   ........................BCDDECA9889BBDDDEEEEEE......EEEEEEDEEEEDEDDDD.EEEEEEEEDDDDD......EDCDDDEEE.....E...EE.........
   ........................BCEEECA9889AADDDDDEDDDEE...E.EE..EE.EEEEDDCDDEEEDDDDDDCCCDE.EE...CRDDEDEEEE....E............E.
130.........................BEEEBA9989AADDDEEEDDDD.....E.....EEEDDDDCDDDEDDDCDDDDCDCCDEEDEEE.CCCEDDD.EEE..E..........E...
   .........................AEEEBA99999ADDDDEEEEEE..E..EE...EEEDDDCDBCDDCCBBCCDCDCBBCDDBCDEEDBCDDDDEEEE...............E.E
   .........................AEEEBA99999ACDDDEDEDDEEEE..E....EEDDDDCCBCDCBAA7ACDCDBBBCB769CDEDCBCCDDEEEE..................
   ..........................DDEAA9AA98BDDDDDEDDDEEE..EEEE.EEDDCCCBCCBCBA7557ACDDBBBC6679BDDDCBCCDDDDDEEE..........E.....
   ..........................DD.AA9AA99ADDDDDEDDD..E.EEEEEEEDDDCDCCCCBBBA976689BCCBBBC8889ACDDBBBDDDCDDEEEE..............
   ..........................DDEBAAAA88BDDDDDDDEEE.EE.E...EEEDDCCCCBCDBAA98889AABBABBAAAAABBBBCBBCCCDDDEEEE..............
   ............................EBAAAA8889DDDDDEDDEEEEEEE..EEDDCDBBBCDDAA99998AAAA99AA9BBAAACDDCCBCBCCCCDDDEE.............
   ............................EDABDD9778DEEEEDEEEEEEEEEEEEEECDCCCCCDDCAAA9AAA99999889BBAABDDDDDCCBBCDDCDEEE.............
   .............................ECE.DA778DDEEEDDDEEEEE.EEEEDDCCCCCCCDDCCABAABA98988999ABABCDDDDEDCBBBCCCDEDE.............
   .............................DEEEDA778DEEDDDDDEDEEEEEEEECCCCCCDDDDDCBBBBBBAA9898999ABABDCDDD.DDBBBCCCDDEE.........E...
140..............................ACDEDB888CDDEDDDDDEDEEEEDDCDCBCCDDDDEDDCCCCAAAA988889ABCBDDDDDEEDDCBCCBCDEDE.......E.....
   ..............................ACDDB997BDDEDDDDDDEDEEDEDCCCABCDDDEDDCBCBBBBBA9889889ABBBCDDDDDDEDCCBBCDCDDEE...........
   ..............................CDDDA887BCDDDDDDD.EEDDDDCCCBCCDEEEDDCBBCDABA989999999ABCCCDDDCEEEDCCCBBCDDD......E.....E
   ...............................EDDAAA7ADDDDDEEDEEEEDDDDDCCDCDDDDDCDCBBCCBBB9999AAAAABBCCDDDDDDEEDDCCCCDDDD......EEEEE.
   ...............................EDBBAA9ACCDDDDDDDEEDDDCCDCDDCCDCCCCCBCBBBBBAA998999ACBBCCDCCDDDEDDDDDCCCDDEE....EEEEEE.
   ................................ECCAABBCCDDDDDDDDDDDDDDDCCDDCCCDDDCCBBBBBBBCBA9AAAABDDDEDDDDDDDDCDDDDCCDDDD.EE.EEEEDD.
   ................................EECCCCACCDDDDDDDDDDECCCDDDCBDBDDDCCBCBCDDCDCCCBCCCDCBDDDDDDDDEDCDBDDDDDCDDEEEEEEEDEDDE
   .................................EDDDBACBDDDDDDDDDDDDCCCCDDBBCCCCBBBBCCCDABBABBCCDDBAABABBCCDDDDDBBCCDDDDDDDEEEEEEEDEE
   ..................................CCCAACBBDDDDCDDDCBCDDDCCABBBAAABCCCBABAA99A9AAAAAAAAAAAAAADCDEDCBCCDEDDEDDEEEDEEDDEE
   ...................................AAA9CBCCCCCDDCCCDDDDDCPABBBAABBBBAA9999998989999999899989AABCDCBABBBCDDEEEEDDDDDDED
150....................................CCECCCCBCCCDDCCDDDDDBBAAAABBBA99878877667778877666667799ABBAAAAADDDDEDDDEEDDDDDDDD
   .........................................8CBBBCDCCDDDDDCBA99AABBB988666688888768778889887777889988AAABDDEDDDDEDDDDDDEE
   ..........................................BCCCCDCBCCDCCCBA999AAA9777899AAAAAA9A9AAAAAAA9A99989765899ABCDDEDDDDDDDDEDDE
   ..........................................BCCCCBCBCDDCCBAA8867886789ABBCCBBABABBBAAABCCCBBCBCA968999BCDEEDDDDDDDDDDDDD
   ..........................................BCBBCCCBBDCCCB998668889AACBBCBBBAAAABBAAABBBCBDDDDDCBA78889BDDEDDEDDDDDCDDD.
   ...........................................BCCCCBBCDCCCB98979AAABBBBABBAAAAAAAA9AAAAAABBCBBCDDCBA9999BBDDDDDDDDDDDDDD.
   ...........................................BBBBBCCCCCBBB9999ABBBCBABAA99999889989999999AABBCCCDDDCAA999ABCDDDDDCDDDCD.
   ...........................................BBBBBBBCBCBBA9AAABCBBBABAA9999988888888889A9ABBBBDDCDCBAAAABBCDDDDDCDCDCDD.
   ...........................................CBBBCCCBBAAA9ABBCCCCBBAAAA99988887888888899ABBCCCDCDCBBBBBABBCDDDCCCCDDCDD.
   ...........................................BBBBBBCBBBAA99ABBBBBBBAAAAAA99987888888889AABBBCCCCDCCCBAAABBCDDCCCCCDDDCD.
160...........................................BAABBBBBBBAAAABBBBCABBBABAA9999987888898899AAABACCCCCBCBBBBABBCDDCBBCDDDCD.
   ...........................................8AABBBBBBAAAAABBBBABBBBAAA998998888888899999AAABAACCCBBBBAAABCCDCCCBCCDCDCE.
   ...........................................8AABBBABBBAAABBAABBAAAAAAA999999889888889AAAAABAABCBBBABAAACBBCCBCCCDDCCD..
   ............................................AAAABABBBAAAABABBABBAAAAA89998889898888999A9AABBBCBBBBBAABCBBBCBCCCDDCDC..
   ............................................AAAABAABAAABBBAABAAAAABAAA99999989999999999A99ABBACBBBBCAABBCBCCCCCCDDCDD..
   ............................................AAABAABAAAABBBAABAAABBABA9AAAAA999A9AA999AAAAACCBCCBBBCBBABDCBBCBBCCDCCD..
   ............................................AAAAAAAAAAAAAABBBBBAAAABBBBAAAAAAA999999A9ABBBBBACCBCBAAACCBCBCBBBCCDCCD..
   ............................................99AAABBAA99ABCABABBBBBBCBBBBAA99989999999AABACBBBACBBCCBAABCCCCBCBADCDDD..
   ............................................AAAAABAAAAAABBABBACCCBBABBBBBAAA99999999AAAABABBBBBBCBBBBBCCBBBBCCBCDCDD..
   ............................................AA9AAAAABBBAAABABCBBCBAABAABAAAA9999AA9AAAABAAAABBBBCBCBBCBBCDBCCCCDDBCD..
170............................................AAA99AAAAAAAABAABCAAAAA9AAAAAAAA8999999AAABBAAAABBBBBBCCCBACCCBBBBCDDCCD..
   ............................................AAAA99AAAAAAAAAABBCBBAAAAABAAA9A9989999999AAAAAAAABBBCDBBABBCCBBCCCCDCCD..
   ............................................BBA999AAAAAA9A9ABAAAAAAAAAAAA9A99998999AA9AAAAAAAAAACBCBBBBABBBCCCCCCCDE..
   ............................................AAA99AAAAAAAA9A99ABAAAAA9A9A99999899999999AAA9A99AAABCCCAACBBBBDBCCCDCDDE..
   ............................................AAAA9999A999AA99AA999A9AA999999999999999A9999999AABBCBBAABBCACBBCCCCDCDE..
   ............................................AAAA99999AA9AA999A999A999A999989988899999999999AAAACBBAAAABBBBCBCCCCCDDD..
   ............................................AAA9A999999A9A99998999999999998888899999999AA999A9ABAAABBBBCBBBCCCBDCDDD..
   ............................................88A9AAA99AAAA9A99988989899999999988999989999A9A99AAAAAAAAAABBBBCBBCCCDDD..
   ............................................AAA99A99A99AAA9999889999999999999888899999A9999999AAAAAAABCBACBBCCCBDDDD..
   ............................................BBAAA9998AAA9A9999998899988998888998888899989989AA9AABBBBBABBBBBBCDDDD..
180............................................88AAAA99999A8998988889998889989889888889889989999999AABAABBABCBBBBCCCDDD..
```

Digitized by Google

Digitized by Google

```
   ............................................BBAAA98999999999998888989889888888898998888888888899AABBBBABACBCBBBCDCDDD..
   ............................................88AAAA999999999989888898987889988898889988888889889AA9ABAAABBBBBBBBBCCCCDEE.
   ..............................................AAAA999AA9998999888888789888887889778899999999AAAABBBBBBBAABBBBCCCCCDE.
   ..............................................AAA999999999999898888978888888888899989999998899BABABAAABBAAABBBBBCCDDDE
   ............................................BBAAAA9A99999999999998987978888888888899999999AAAABAAAAAABBACBBBBCCCCCDDB
   ............................................AAAAAAA99999999988888888788888889889889999999AAAAABBAAAAAABBBAABBBBBDDDD9
   ...........................................ABBAAAAAA9999999988997998888888788788988899A9AAAAAAAAAAAAAAABBABBBBCBCCCDC9
   ...........................................AAABAAAA9A99999999998899989878888788988887999 9BAAB9ABAAAAAABAAABBBBBBCCCCB9
   ...........................................AAABAAAA9999999989988988998788887798877889999AAAABABBAAAAAAAAABBCBBBBCBCBB7
190.........................................8ABBAAAAA9A99998A99999998999987888788987789999AAAABAAABBABBA99AABBBBBBBCBA6
   ..........................................9AAAAAAA99999999999998898899988998788988888999AAAAAABAAAAAAAA9AABBAABBBBBBA6
   ..........................................8ABBAAABAA9999999A9988898998888787888888899AAAAAAAAAAAAAAAA9ABAABAABBBBA85
   ...........................................9AAAAAAA99999999999999999898888888898888999AAA9AAAAAAAAAAAAAAAAAAAAAAABAA965
   ...........................................999AAAAA99998899998999898898888888789989AAA9999AAA9AAAAAAA9A9AA9AAAAA8874
   ...........................................A889AAA999AA999989899898988988899878788899999AAAAA9AAAAAAAA999A99AAAA98775
   ............................................998AAAA9999989999888898888888887777789999A9AAAAA9AAA99A99999999998888765
   ............................................AA99AA99988999999988899887888888878888889A999AAAA9AAAA9999999998999 8887765
   ..............................................9899A999999A999888999988899888898888899AA99AAA9A9A9AA999998988877778756
   ..............................................E789A99999899998999989989888888888999999A99AA99999999988888888688788 7676B
200...............................................6689999999989899989989898888888889889999999AAA98999988887877678777768B.
   ...............................................A678899998998999999989899898888888999999A99999999888887778767777766 78A..
   ................................................6677888988989888888989988889888899 9A95A9999988878877777768667678A...
   ................................................C766788888888888888888788998988898899 9B989999878777776777676666669E...
   .................................................D76677788888877888878888788888989999999999877777666766776666668D....
   ..................................................9656678887787778888888877888897788989888887767667766666666668A.....
   ..................................................C8555678787777777878777777787888888888878887676676666667666668DDD...
   ..................................................ED75566777777777778787778777878888788887997667766667676666668A767AE
   ...................................................DA6656676676677776877777877878878887777766667667766666665568865459
   ................................................CA9AAAA65666666776677777766877787887877876777666666777666666666 8965554
210.....................................E.......6534589995666666667777778878878878887877767776766677666766666555C966444
   .......................................EE......2234568BA65666655666778788888787877877776777766677677666566555D865443
   .......................................EDE...913344577986566666666666687887787787787876776676666676666665556 69E865434
   .......................................EED..EE5334455558DA8656666666667777878778788777867677766777666666566555 59.665442
   .......................................EDDE.DD6533445668BDEBA866566666677778787877777767676666666766665656555A..764323
   ........................................EDD.DD6543445779CBBCDDAA666666666667977777777766667666666667666666555 55AD.754222
   .........................................DDDEE7654-45668DCCCBCBBA55556566666777688767666666666667666656555556AD9653225
   .......................................EEEDDEE9676344667CBBBDDCCA555565556666677997766676666667667555656655 89EA632245
   ......................................EEEEDCDDB887434556ABABBCCCB655555555566766766666766766666766655555555599.A522457
   .......................................EEEDDCCCA986434458A99ABBBB85555565566666666666767677667666666566555 5BB.6223589
220.....................................DDEDCCDAA99643346889AABBAB755555555556666666666777667667666666555555A...62358AA
   .......................................EDDDDCCCBBAA954435 6889AAAABA6455555555565666676766666767666655555558E..84238ABD
   ........................................ECCDDDCCBA9644455578999AAA75555555555555666776677777777666665559ADD..72359CDE
   ........................................EDBCDDDBCCBA855455556899AAAA755455565566666666777777777666555558B.DD.E6348CDDD
   .........................................DBACCDCCCBBA744445557999BABA644445556566666777667777666665558BEEDD.D535BECCD
   ........................................E.DBAADDBCDCCAA6455556669ABDB64455555566556667776777676656555D.EEDD.965AEEBDE
   ..........................................ECAABDDCCCCDDA4555556669BCC975555555555556666677776676666558...EDD.E69E.CBD.
   .........................................E.DBB9DDDBDDCCB65555566678 98CAAA7545566556667777668776656 88D..EEDDDD6D.DBDD.
   ...........................................ECCACDDDDDCCD75555566687889ABBB7645555566676686677666 6AEE..EEEDE69E.EDBEE.
   ..........................................EDDBBDCCDCDDE966666777888898799BA7655666666767777767 9C.....E.EDB5AE.EBDE..
230.........................................EECACCDDCCCDB8677777888899888889ABA75666667677777679D.......EED64BD..DD..E
```

Digitized by Google

```
KELH1                        17-JUL-70 15:30
   ITH=15
   167       177       187       197       207       217       227       237       247
   .E...E.E..E.EEE.EEEE.EE.EEEE...EE.EEEE..EE.E....E..EEEE.EE.EEEEEE.EE.EEEEEEEE.EEEEE.
   EEE.EE..EE.EEEEEEEEE.EE..EEE.E.E.E.EEEEEEE.EEE.E.EE.EEEEE..EE.EEE..EE.EEEEE.EEE.EE.E
   .EEE.EEEEEEEEEEEEEEEFEEEEEEEEE.EEEEE..EEE.EEEEE.EEEEE.EE.E.EEEEEEEEEEEEEEEEEE.EEEEEEE
   EEEE.EEEEEEE.E.E.EEEEEEEEEEEEEEEEEE..EEE.EEE..EEEEEEEEEEEEEEE.EEEE.EEEEEEE.EEEEEEEEE
   EEEEE..EEEEE.FE.EEE.E.EFEEEE..EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE.DDEEEEEEEEEEE
   EFEFEEEEEEEFEEEEEEEEEEEEEEE.EEEEEEEE..EEEEEEEEEE.EEEEEEEEEE.EEEEEEEEEEEEEEEEEEEEE.EEEE
   EEE.EEEEEE.FEEEEEEEEEEEE.EEDDEE.EEEEDDEEEEEEEEEEEEEEEEDEEEEEEEEDDEEEEEEEEEEEEEEEEEEEEEE
   .EEEEEEEEEEEFE.EEEEE.EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDEEEEEEEEEEEEEEEEEEEEDEEEEEEEEE
Ø3ØEFEEEEEEEEEEFEEEEEEDD.EFEFEEEEEEEEEEEEDDEEEEEEEEEEEEEEEEEEEEFEEEEDEEEEEEEEEEEEEEDDEEEEEEEEEEE
   EEEEEEEEEEEEEEEEEEEDEEEEEEEFEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDDEDEDEEDEDDEEEEDDEEEEEEEEEEEEDDE
   EEEEFEEEEEEFEEEDEFEEEEFEEEEDDEEEEEEEEEDDEEEDEEEEEEEEEEEEEEDDDEEEDDEEDDEDEEEEDDDEEEEEDEEDDEF
   EEEEEFEEEEEDEEEEEEEEEDEEEEEEEDDEEEEEDDDEDEEDEDEEEDEEDDEDDEDEEEDDDDDDDEDDEEEDEEEEDDDDDEE
   EEEEEEEEEEEEDDDDEEEEDEEDFEEEEEEEDEEDEEEEEEDDEEDDEEEEEEDEEEDEEDEEDEEEEEDEDEEDDDEEEEDDEEEEE
   DEEEEEEEEEEEEDEEEEEEEEEEEEDEEEEEEDDDEDDDDDEEEDEEDDDEDDEEEEDDEDDEEEEEDDEEDDEEEEEDDDDEEDDE
   EEDEEEEEEDDEFEEDEEDDDDDEEEDEEEDDEDDEDDDDDEDEEDDEEEDEDDEEEEDDDDEDDDEDDDDDDDEEEDDEDDDDE
   EDEEEDDEEEEDEEEEDEEEDEEEDEEEEEEDEDEEDDEEEEEEDDDDEEDDDDDEEEDEDDEDEDDEDEDDDEEEDDDDDDDDDDEE
   DEEDEEDDEEDEDEEEEEEEDEDFEDDDDDEEDDDDEDDDEDEDDEEEDDEEEEDDDDDDEDDDDDEDEEDDDDEDDDEDDEE
   EEEEDEEEEEEDEDDEEEEDEDDDEDEDDDDEDDDEDDDDDDEEDDEEEDDDDDEDDDDDDDDDDDDDDDDDEEDDDEDDDE
Ø4ØEDEEDEEDDDDDEDEDEEEDDDEDEEDDDEEDEDDEEDDEDDEEDDEDEDEEDDDEDEDDDDDDDEEDEDDDDDDDDDEDEEDDD
   DDDDEEDDEEDDDEDEDDDEEEDEDEDDEDDDDDDDDDDDDDDEDEEEEEEDDFEDDEDEEDDDDDDEDDEDEDDDDDDEDEDDEE
   EEDEDEDDEEEDDDEEDDDEEEDDEEEEEDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDEDDDDDDDDDDEDDDDDDDDDDEDDDD
   DDDEDDDDEEDDEDDDEDDEDDDDDEDDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEDDDDDDDDDDDDDDDDDDEEDD
   DDDDDDDDEEEDEDDDDDDDDDDDDEEDDDDEDEDDDDDDDDDEDDDDEDDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDEEDDEDDDDDDDDDDDDDDDDDDDEDDDDDDDDEEDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDD
   DDEEEEEEEEEDDDEDDEDDDEDEEDDDDEDDDEDDEEEEDDDDEDDDDDEDDDDDDDEDDDEDDEEDDDDDEEDDDDEEDDDDE
   DDDEEDEDDDDDDDDDEDDDDEDDDEDDEDDEDDDDDDDDDEDDEEDEEDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDEDDEDDDEDDDDEEDDDDDEDEDDDDDDDDDDDDDDEDDDDDDDDDDDDDEDDDDE
   EDDDDDDDDDDDDDDDDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Ø5ØDDDDDEDEDDDDDDDDDDDDEDDDDDDEDDDDDDDDDDDDDDDDDDDDEEEDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDEDDDDDDDDD
   DDDDEFDDDCDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCEDDEEDDCDDDDDDEDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDEDDDDDDDDDDEDDEDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDEDDDDDD
   8DDDDDDDDDDDDDDDDDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDEDDDDDDDDDDDDDDDDDDDDDDDCCDCDDDDDDDDD
   8ADDDECDDDDDEDDEDDDDDDDCDDDDDDDEDDDDDDDDDDDDDDEDDDEDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDD
   89BCDEDDDDDDDEDDDDDEDDDDDDEDDDDDDDDDCCEDDDDDDEEEDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   789BDDEDDDDDDEDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   788ACDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDEDDDEDDDDDDDDDDDDDCDDDDDCDDDDDDDEDDDDDDDDDDDDDDD
   8889BDDDDDDDDDDDDEDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDCCDDDDDDDDDDDDDDDDDDDDDE
Ø6Ø77889CDDEEEDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDEDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDD
   77778ABDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDDD
   878888ADDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDCDDEDDDEDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDCDDDDDDD
   6677878ACCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   87777889AADDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCDD
   776777789ABDDDDDDDDDDDDDDDDDDDDDDDDDECCDDDDDDDDDDDDDDDDDDDDDDDDCDDDCCDCDDDDDCCDCDDDDDDDDD
   778776789AADDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDCCDD
   7677666789CDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDCECDDDDDDDDDDDDDDDDDDDDDDD
   66767766889BDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEEDCDDDDDDDDDDDDDDCDCDDDDDDDDDCCDD
   7666677777 8ACDDDDDDEDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDCDDDDCDCDDDDDDDDDCDDDDD
Ø7Ø77677777788798CDDDDDDDDDDDDDDCDDDDDDDEDDDEDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   7677777888799BCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDCDDDDDDDDDC
   6779777788899ABDCDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDCDDCDDDDDDDDDDC
```

185

Digitized by Google

```
   6666787777799AABDDDDDDDDDDDDDDCECDDDDDDDDDDDDDDDDDDDDDCDDDDCDCDDDDDDDDDDDDDDDDDDDDDDCCDD
   667877787789999ADDDDDDDDDDDDDDDDDDDDDDDCDCDDDDDDDDDDDDDDDDDDDCDDDDDDCDDCDDDDDDDDDDDDDDDDC
   6678688777889999CDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCDDDDDDDDDCDDDDDDDDDCDDDDDDDD
   6677787777789999BDDCDDDCDDDDDDDDDDCDDDCDDDEDDDDDDCDDDDDDDDDDDDDDDDDCDCDDDDDDDCDDDCDDDD
   6767877788799999ADDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDCCCDCCDDCCDDDDDDDDDDDDDDDDDDCCDC
   7767887777869999ADDDDDDDDDDDDCDDDDDDDDDDDDDDDDDCDCDDDDDDDDCDCDDDDCDDDDCDDDDDDDDDDCCDD
   7777777877768989ABBDDDDDDDDDDDDDDDDDDCCDDDDDDDDDDDCDDDCDDDDDDCDDDDDCCDCDCDCCDDDDCCDCCDD
0807776777877789999ACCDDDDCDDDDDDDDDDDCDDDCDDDDDDDDDDDDCDDCDDDCDDDDCDDDDDDDDDDDDDDDDDDDDDCC
   7777768777789898ABBDDDDDDCDDDDDDCDDDCDDCDDDDDDDDDDDDDDDCDDCDDDDDDDDDCDCDDDDCDDCDCDDCC
   8667767866678888BBDDDDDDCDDDCCDCDDDCCDDDDCCDCCDDCDDDCDDCDDCDCCDDDDDDDCCCCDDDDDCCCCD
   8766666876688889AADDDDDDDDDCDDDDCDCCDDCDDDCDDCDCDDCDCCDDCDDDCDDCCDDDCDCCDCDDCDDCCDC
   98665567766888898AADCDDDDCDDCCDDCCDDDDDDDCDCCDDCDDCDDDDCCDDDCCDDCDCDCDDDDDDDDCDDDDC
   99666667666788798AACDCDDDCDDCDCDDCDDDDDDDDCDCCDDDCCDCDDDDCDDCCCDDCDCDCDDDDCDCCCDDDDC
   AA666666666888999AABDDDCDDCDDCCDDDCDDCDCDCDDCCCCDDCCDCCCDCCDDDCDDCDCCDCCCCDCDCDCDDCC
   BA66676677688889A9AABDDCCDDDDCDDDDDDDDDDDDDDDCDDDDDDDDCCCDDCCDDCCCDCDCDDDDCCDDCDDCC
   BA66676666788889A9AABDDCDDCDDDDDDDDCCDDCDDCDDDCCDCCCDDCDDDDDCCDDCDDCDDCDDDDCCCDDCDCCDD
   BA6666766668989A9BBCDDDDDCDDDDDDCDCDDDDDDDDCCCDCDDCDDDDCDCCCCDCDDDCDDCCDDDDCDDDDCCCD
090CA6666776668899A9BBCCDCDDDDDCDCDCDDDDCDDDDCDDDDCCDDDCDDCCCDDDDCCCDCCCDDDDCDCDDCCCDD
   DA76677766788889AABBCCDDDCCDDDCDDCDDCCDDDCCDDDDCDDCDDCDDDCCCDDCCDDCDDCDCDDCDDCDCDCCDC
   CA7667676678888ABBBBCDDDCDDDDDDDCDCDCCDDCDDDDDDDDCDCDCDDDCDCDCDDDDDCDDCDCDDDDCDCDCDDCC
   DA767667777988ABBCCDCDDDCDDDDDDDDCDDDDDDDDDDDDDDDCDCDCCDDCDCDCDDCDDCDDDDDDDDCCDCDDDDC
   DB766677887888ACCCCDDDDDCDDDDDDCDDDDDDDDDDCDDDDDDDDDDCDDCCDCDDDCCCDDDCDDDDCDDCDCDCCDD
   DA766677778787ACCCCCDDDDDDDDDDDDDDDDDCCDDDDCCDDDCDDDDDCDCDCDDCCCCDDDCDDDCDDCCDDCCCC
   DB677778887888BDDDDCDDDDDDDDDDEDDCDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDD
   DB767778889788DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDEEDDEDEDDED
   EB786768887779DDEDDDDEDEDEDDEEDDEEEEEEEEDEEEEEEDEDEDDEEDEDEEEEEEEEEEEEEEEEEEEEEEEEEEEE
   DC777888887789DEEEEEEEEEEEEEEEEEEE.EEEEE.EEEEEEEEE.EEEEEEEEEEEEEEEEEEEEEEE..E.EEEEE..EE
100DC776878777888E......E.EE.....EEE..EE..EEE.EEE.EEE.EEEE.DED.E.EEEE.EEEEEEEEEE.EEEEEE
   ED77778877779DE.E....E.E...EEE.E.EE..EEEE.EE..E.EE.EEEEEEEE.DEEEEE.EEEEEE.EEEEEEEEEE
   ED7778887777AEE.EEE.E...E..EE.....EEEEE.EE..EE..EEEEEEE.E.EE.EEEE.EE..EEE..E.EEEEEED
   .E67778777768A.E.EEE.E...EE.EE.EE.E...EE.E.EE...EE.EEE..EE..EEE..EE.E.EEEEE.EE.EEEEEE
   EE7668877779CE.EE..EEEE.EEEEEEEE.EEEEEEEEEE.EEEEEEEEEEEEE.EEEEEEEE..EE.EE.EEEEEF..EE
   EE7667877779DEE..EE..E.E.EEE..E.EE.EEE.E.E.EEEEEE.EEE..EE.EE.E.EEEEEEEEEEE.E..EEEEE.
   .E8668777779DE.EEEE..EE.EEEE..E.EEE..EE.E..EEE.EEEE..EEEEEE.EEEEEEE.E.E..EEEEEEE..EE
   ..9668867779D.EEEEEEE..EEE.EE.EE.EE..EEEEE..EEE.EEE.EEEE.EEEE.EEE..E.EEEE.E..EEEEEEE
   ..A76788778AD...EEE..EE.EE...E...E...EEEE..E..E.E...EEEE.EEEEEEEE.EE.E.EE.EEE..EEEEE
   ..B768867779C...EEEEEEE..E.EE.E.EE...EEE.E..EEEEE.EE.EE.EEEEEE...E..EEEEE.EEEE.EEEEE
110..D767677787A...EEEEE.EE.EEE.EE.EEEEEE.EEEE...EEEE.EEEEEEEEEE.EEEEEEE.EEEEEEE.EEEEEF
   ..E8777777879E.EE..E.EE..E.....E.E.EE......EE.EE...EEEEEE.E.EEEE.EE....EEEEEEE..EE
   ..E96777788ACE.E.EEEEE.E...EEE..EE.EEE...E.EEE.E...EEEE.EEEEEEEEEEEE.EEEEEEEEE.EEEEF
   ...A777777EE....E..EEE..EE.EE.E.E.EEEE.EE...EEEEEEE.EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
   ...B87789AB......EE......EEEEE.EEEEEE.E.EEEEEEEEEEE.EEEEEEEDEDDDEEDEDEEEEEEEEEEEEEEEE
   ...D7779AAD.EDD..EEEEDEEEEEEEEEEEEEEE..EEEEEEEEDDDEEDDEEEDDDDEEEEDDDEEDEEEEEEEEDEEDEEEE
   ...E9769BBDDCAABE..DDDDEDEDDEDEEDDEEEEEEEEEEEEEEEDEEEEEEEEE.E.EEEEE..................
   ...EA779CCEDBAAADEE..................................................................
   ...EA67ACCEDBA9AB....................................................................
   ....A77ACCCDBA9AB....................................................................
120....D77ABBADDA9ABDD..................................................................
   ....A78AAAADDAAABDD..................................................................
   ....A879AAADEBAAACC..................................................................
   ....D87ACCDEDBAAACC..................................................................
   ....D78ACCDEEBAAADD..................................................................
   ....A889CCCEED9ABDD..................................................................
   ...EB889BBBDEEABBDD..................................................................
```

Digitized by Google

Digitized by Google

```
   ...EA889AABD.EABC..............................................................................
   ...E9889AABDE.BAC..............................................................................
   ...DA889AAACE.BBD..............................................................................
130...DA8999AACE.CBE..............................................................................
   E..C9969AAAC..CC...............................................................................
   .E.A98999AAD..DD...............................................................................
   ...B9999AAAD..D................................................................................
   ...A899BAABD.EE................................................................................
   ..E989AAAABE.E.................................................................................
   ..D989AABBBE.E.................................................................................
   ..DA8AECEEE.E..................................................................................
   .EBA8DE...EDE..................................................................................
   .EB88A...CBD...................................................................................
140..A89CE.DDB....................................................................................
   .E98ACE.DDD....................................................................................
   EEA9ACE........................................................................................
   .EAABDDE.......................................................................................
   .EDCBD.........................................................................................
   .ECEE..........................................................................................
   EECD...........................................................................................
   EDACD..........................................................................................
   EEABDE.........................................................................................
   DECD...........................................................................................
150DE.............................................................................................
   E..............................................................................................
   E..............................................................................................
   D..............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
160...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
170...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
   ...............................................................................................
180...............................................................................................
```

Digitized by Google

```
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   E...................................................................................................
   8D..................................................................................................
   99..................................................................................................
   98..................................................................................................
   95E.................................................................................................
   75D.................................................................................................
19Ø63A.................................................................................................
   639E................................................................................................
   539.................................................................................................
   549.................................................................................................
   44A.................................................................................................
   54D.................................................................................................
   55D.................................................................................................
   55..................................................................................................
   6B..................................................................................................
   B...................................................................................................
2ØØ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   ....................................................................................................
   E...................................................................................................
   9E..................................................................................................
   49..................................................................................................
21Ø45..................................................................................................
   33E.................................................................................................
   43D.................................................................................................
   23A.................................................................................................
   34A.................................................................................................
   25B.................................................................................................
   55C.................................................................................................
   56D...E.............................................................................................
   79...EE.............................................................................................
   9A...DEEEE..........................................................................................
22ØAD..ECDD............................................................................................
   DE.EABD.............................................................................................
   EE.A9C..............................................................................................
   DEEAC...............................................................................................
   DECD................................................................................................
   E.DE................................................................................................
   ..DEE...............................................................................................
   ..DE................................................................................................
   .EE.................................................................................................
   .E..................................................................................................
23ØE...................................................................................................
```

Digitized by Google

```
KELH4                        17-JUL-7Ø 15:31
   ITH=15                                                                                                              E
   Ø50       Ø60       Ø70       Ø80       Ø90       100       110       120       130       140       150       160   E
Ø2ØDDEDDDDDEEEDDDDDDDEEEEEEDEEEEEEEEDDDDDDDDDDDDDDDDDDDDDDEDDDEEDDDDDEDDDEDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDCDDDDDDDDDDEEEEEEEDDEEEEEEDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDEEDDDDEDDDEEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDCDDDDDDDCDCCCDDCDDDDDDDDCDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDCCDDDDDCDDCCCCCCDDDCCCCC
Ø3ØDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDCDDDDDDDCCDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDCCDCDCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDCDDCDDDDDDCDDDCCDCCCDDCDDDCCCDDCCCCCCCCCCDDCDCCCCCC
   DDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDDCDDDDDCCCDDCCCDDCCDCDCCDCDDCCCCCCDCCCCCCC
   DDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDCDDDDDDDDDDCDDDDDDDDDDDDDDDCCDDDDDDDCCCCDDDDCDDDDDDDCCCDDDDDDCCCCCCCCCCCCCDD
   CDDDCDDDDDDDDDDDDDDDDDDDDDDDDDCCDDCCCDDCCCDDDDDDDCDCCCCDCCCCDDCCCDDCDCCDDDCCCCDDCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   DDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDCCCCDDDDDDDCDCDDCCDDDCCCCDDCDDDCCCDDCCCCCCCCCDDDCCDDCCCCCCCCCDCCCCCCCCCCCC
   DDDCCDDCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDCCDDCCCDCCCDCCCCCCCCDDDDDCCCCCDDCCCCCCCCCCCCDCCCDDCDDCCCCCCCCCCCCCCCCCCCCCCCC
   CDDCDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDCCCDCCCCCCCCCCCDCCCCCDCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDCCCCCCCCCCCCCC
   CDDDDDDCCCDDDDDCDDDDDDDDDDDDDCDDCCCCCCCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   CDDDCDDDDDDCDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCDDCCCDCCDCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDDCCCCCCCCCCCCCCCC
Ø4ØCDDDDDDCCCDDDDDDDDDDDDDDDDDDDDDDDDDCCDCCCCCCCCCCCCCCCCCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   DDCCCCCCDDDDCCDDDDCDDDCDDDDCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCCCCCCCCCBCCCCCCBBCCCCCCCCCCCB
   CDCCCCCCCCDDCDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBAABBBBBBBCCCCB
   CDDDDDCDDDCDDCDDDDDDDDDDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBAAAAAAAA99999999AABBBC
   CDCCCDCCCCCCDDCCCCCDDDDDDCDDDCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCBBBBAA999999999888888888686998899AABB
   CDDCDDDCDDCCDDDDDDDDDDCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBA8898888888888888889866666767 8899
   CDCDDCCDDDDDDCDDDDDDCDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBAA99986777777777989889889896988666666667
   DDDDDDDDDCCCCCCDCCDDDCDDCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBA988699776666666666777797796696977776665555
   CDDCCDDDDDCCCDDDDDDDDDDDCDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCCCBBA86766666666666666666666777777666666665555
   CCCCDDDCCCDCDCDDCCDDCCCDDCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBAA866666566555555566666666777777767666656555555
Ø5ØDDDCDDDCCCDDDDDCDDDDDDDCDDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBA9776655555544445556666666677777766767766665555 55
   DDDCCCDDCCDCDDDDDDDCCCCDDCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBA966654444444444444555555566676666566666666555555
   CCCCCCCCCCCCCCCCDDDCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBCBBA96555444445555555556555555555566666555555555555555
   CDDDCCCCCCCCCCDDCCDDDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBA866555455555555555666666666666666665555555555455555
   CDDCCDDCCCDDCCDDCCCCDCCCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBB966554555555565665556655666667766666555555555555555 5
   CCCCCCCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBBBBBA655544444455555555555565655666777665555565665555555 5
   CCCCCCCDDCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCBBBBBAA7544444335444444444445555555555667676665444556655555
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBBBBCBBBA977544444455544444445544455555555555667667655444555665555
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCBCBBBBBBA965544444455554454444444454555544455566677666554445566555
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCCBBCBBBCBA96544444445544443344444555555555544555555676976755544555 56
Ø6ØCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBA9655444445544444434444444445555555555555556676767765544455
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCBBBCCBA96554444444434444444433333455555555555566666565666766666 5555
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCCBBBBB96554444444444444433333333455555555666666666665555566666 65554
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCBCCBCCBBBBAAA6554444444444333323334445555566666666666666555555555555665555
   CCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCBBBCCBCBBBBBBA5544444444443332233333444444555556666767666665555555554555555 5
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCBCCCCBBBBAA954444445554333323333344433345556677666666555555565544444555555
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCCCBBBBBA99644444444433333333443333335556665555554455555556555555543334 44
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCBBBBBBBA6654444554433333333333333344444555555554555555555444444555444 3333
   CDCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCCCCCCBBBCCBCBBBA96654443333333333443333334444444444444555544443333333332332333 4444
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCBCBBBBBBBBBBA655543333333333333333333433333334434443323232222333444444344344 34
Ø7ØCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBCBBBBCBBBBBBBBBBBA955543322232233232223333333333322233223222333333333334444455554 44
```

189

Digitized by Google

Digitized by Google

```
KELH4                        17-JUL-70 15:31
  ITH=15                                                                                                              E
  050       060       070       080       090       100       110       120       130       140       150       160  E
020DDEDDDDDEEEDDDDDDDDEEEEEEDEEEEEEEEDDDDDDDDDDDDDDDDDDDDDDEDDDEEDDDDDEDDDEDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDEEEEEEEDDEEEEEDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDEEDDDDEDDEEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDCDDDDDDCDCCCDDCDDDDDDDDDDCDDDD
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDCCDDDDDCDDCCCCCCDDDCCCCC
030DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDCCDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDCCDCDCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDCDDDDDCDDDCCDCCCDDCDDDCCCDDCCCCCCCCCCDDCDCCCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDCCDDCCCDDCCDCDCCDCDDCCCCCDCCCCCC
   DDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDCDDDDDDDDDCDDDDDDDDDDDDDDCCDDDDDDDCCCCDDDDCDDDDDDCCCDDDDDCCCCCCCCCCCCCCDD
   CDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDCCDDCCCDDCCCCDDDDDDDDDCCDCCCCDDCCCDDCDCCDDDCCCCDDCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   DDDDDDCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDCCCCCDDDDDDDDCDDCCDDDCCCCDDCDDDCCCDDCCCCCCCCCDDDCCDDCCCCCCCCDCCCCCCCCCCCC
   DDDCCDDCCCDDDDDDDDDDDDDDDDDDDDDDDDCDCCDDCCCDCCCDCCCCCCCCDDDDDCCCCCDDCCCCCCCCCCCDCCCDDCDDCCCCCCCCCCCCCCCCCCCCCCCCCC
   CDDCDDDCDDDDDDDDDDDDDDDDDDDDDDCDDCCCDCCCCCCCCCCCDCDCDCCCCCCDCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   CDDDDDDCCCDDDDDCDDDDDDDDDDDDDDCDDCCCCCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCDCCCCCCCCCCC
   CDDDCDDDDDDCDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCCCCCDDCCDCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCDCCCCCCCCCCC
040CDDDDDDCCCDDDDDDDDDDDDDDDDDDDDDDDDCCDCCCCCCCCCCCCCCCCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   DDCCCCCCDDDDCCDDDDDCDDDCDDDDDCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCCCCCCCCCBCCCCCCBBCCCCCCCCCCCCCB
   CDCCCCCCCCDDCDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBAABBBBBBCCCCB
   CDDDDDCDDDCDDCDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBAAAAAAAAA999999999AABBBC
   CDCCCDDCCCCCDDCCCCCDDDDDDCDDDCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCBBBBAA9999999998888888886869988 99AABB
   CDDCDDDCDDCCDDDDDDDDDCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBA88898888888888888888898666667678899
   CDCDDCCDDDDDDCDDDDDDCDDDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBAA99986777777779898898898698866666667
   DDDDDDDDCCCCCDCCDDDCDDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBA988699776666666667777977966969777766 65555
   CDDCCDDDDCCCDDDDDDDDDDCDDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCCCBBA8676666666666666666666777777666666666665555
   CCCCDDDCCCDCDCDDCCDDCCCDDCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBAA866666566555555666666667777777676666565555 5
050DDDCDDDCCCDDDDDCDDDDDDDCDDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBA977665555555444455566666666777776676776666555555
   DDDCCDDCCDCDDDDDDDCCCCDDCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBA966654444444444444455555566676666566666666555555
   CCCCCCCCCCCCCCCCDDDCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBCBBA965554444455555555565555555566666655555555555555
   CDDDCCCCCCCCCCDDCCDDDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBA866555455555555556666666666666666555555555455555
   CDDCCDDCCCDDCCDDCCCCCDCCCCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBB96655455555555656655566556666677666665555555555555 5
   CCCCCCDDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBBBBBA655544444445555555555556565566677766555556566555555
   CCCCCCCCDDCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCBBBBBAA75444444335444444444445555555555667676665444556655555 5
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCBBBBBCBBBA977544444455544444445544455555555556676676554445556655555
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCBCBBBBBBBA965544444455544544444444545555444555666776665544455 66555
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCCBBCBBBCBA96544444444554444334444455555555554455555567697675554455556
060CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBA965544444554444443444444444555555555555555566767677765544455
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCBBBCCBA96554444444443444444433333345555555555556666665656676666655555
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCBBBBB96554444444444444444333333345555555555666666666666555556666665554
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCBBCCBCBBBBAAA65544444444444443333233344455555566666666666666655555555555665555
   CCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBCCBCBBBBBBA554444444444443332233333444444455555666676766665555555555545555 5
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCBCCCCBBBBBAA9544444555433332333334443334555667766666665555555655444444555555
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCCCBBBBBA99644444444443333333334433333355566655555544555555565555554333444
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCBBBBBBBBA66544444554433333333333333344444555555554555555554444445554443333
   CDCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCCCCCBBBBCCBCBBBA966544433333333334433333344444444444455554444333333333233233334444
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCBCBBBBBBBBBBA6555433333333333333333333433333344344433232322223334444443443434
070CCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBCBBBBCBBBBBBBBBBBA95554332223223323222333333333333222332232222333333333344444555444
```

Digitized by Google

```
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCBBBCCBBBBBBBBB965443322232332222233333333333332223333223334444444444554455555555554
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCBBBBBBBBBBBBBA654333223323223323333332323233223333333455555555555555555555555555
   CDCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBB99554332233323223323232332222323333333345555555555555555555555655566
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCBBCCCCCCBBBBBBB96542222233333333323222222233333333333455565555565666655556666666666677
   CCCCCCCCCCCCCCCCCCCCCCCCCCBBCBCCCBBBBCCBBBBBBBBA75432222333322222222222332333333333345555666566665566666666666666667888
   CCCCCCCCCCCCCCCCCCCCCCCCCCBBCCBBBBCBBCCBBPBBBBB9553323333332322223233333333333333334555565555666666666666676667798889
   CCCCCCCCCCCCCCCCCCCCCCBBBBCCCCCBBBBBBBBBBBBBBBB75433333333232223333333333444433333333345555555555566666666666666777898899 9
   CCCCCCCCCCCCCCCCCCCCCCBCCCCCCCBCBBBBBBBBBBBBBA543333333232223333333333344433333334444444444445566666666676777899889999AA
   CCCCCCCCCCCCCCCCCCCCCCCCBCCCBBBBCBBCCCBBBBBBAAA5434433332323333333444333333333444444443445555666666666777889988999AAAAB
080CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBBBAA95334433323333334444443333333334343333333345566666667777778888899999AAAABBB
   CCCCCCCCCCCCCCCCCCCCCCBCCCBBCCCBBBBBBBBBBBBAA74333333333333334444443344443333333333345555666666779866899999AAAABBBCC
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBA996444333333333444444444333333333333333334455666667888889 8999AAAAAABBCCDDD
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBA775444443333333344444444433333333333333333455666778889999999AAAAABBBCCCDEEE
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBCBBBAA6654444433333334444444444332333333333333455567988899999AAAAAABBBBBBCDDDDEEE
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBBA66544444333333444443333332333333333344455678899999AAAAAAABBBBBCCCCDDEEE...
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCBBBBBBA6654444433333344444333333222323322333335555799999AAAAAAPBBBBBCCCCCDDDCEEE....
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBA665454443333344444333333222323333333555667 99AABBBBCBBBBBCBBCDDDDDDDDEEE.....
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCRBBBBBBBBB955544433333334433333332222333333455 67889AABBCCDDCCCCCCCCCCDDDDDDDDEEE.....
   CDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCBBBBBA9555443323222443333333333323333556799AAARBBCCDDDDDCCCCCCCCCDDDDDDEEEEEEEEE.
090DDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBA95554333333333445555555433333444 56799AABBBCCCCCDDDDDCCCCCCCCCDDDDDDEEEEEEEDDE
   DDDCCDDDCCCCCCCCCCCCCDCCCCCCCCCCCCCCBBBBBBBA755533333333334566666665555556679 9AAABBBCCBCCCCCCCCCCCCCCCCDDDDDDDDDDDDDDDD
   DDDDDDDCCCDDDCDDDDDDDCCCCCCCCCCCCCCBBBBBBBA755433333333334566666666666788999AAABBBBBBCBBBBBBBBBBCCCCCCBCDDDDCDDDCCCCCCC
   DDDDDDDDDDDCDDCDDDDDDDDDDCCCDCCCCCCCCCBBBBBA766532323333334566776678889 9ABBBBBBBBBBBBPBBBBBBBBBBBBCBBBCCCCCCCCCCBBBBBB
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCCBBBA95553233333333456678889 99AABBBBBAAAAAAAAAAAAAAAAAAAAABBBBBBBBCCCCHBCCPBBBBBB
   EEDDDFDDDDDDDDDDDDDDDDDDDDDDDDDDDCCDDCCCCCCCCB966533333333334567899999ABBBAAA98888889999999AAAAAAAAABBBBBBBBBBBBBAAAAABB
   E,EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDDDDDDDDEEDB9555333333333445799999AAAAA9887676666666667888999AAABBBBBBBAAAAAAAA99999A
   ..................EE.E.E.EE..EEEEEEEEEEEEEEEDA66533333433335679999AAAAA98666665555554445566678899AAAAAAABAAAAAAAA8897777
   ..........................E.....EEEEEEEEEEEEDDCA6653333333333568899 99A99887666655444333334455666899 99AAAAAAAAAAAAA7766555
   .............................EE.EEEEEEEEEEDDDDCA66533333333335688999999876666665544433333333455556889 99AAAABAA99999 8665544 4
100.........................E.E...EEEEEEEEEEDDDDCA665333333333345689 9AAA7766666654555555555555555566678899ABBBBAA998E896655444
   ..............E....EE..E......EEEEEEEEEEDDDDDDB7753333333333457899AA96666666655566665555555566666777899BBBCBA989996655555
   ............E..EE.E...E.EE.E..EEEEEEEEEEEDDDDDA77533334433345799AAAA7666666666666655555455566666667799ABBDCCA9986676665555
   .......................EE.EEEEEEEEEEEEEEEDDDDDA775333343333345899AAAA86766666666554444444444456666667999ACCDEDBA987766555555
   ..............................E,EEEEEEEEDDDDDA665333344332459 9AAAA98896776664433333345555566667899BCCD.EDB99666555 5444
   ...............................EEEEEEEEEDDDDCA6654333443335689AAAA998898655543345555565556666678899ACCD..EB976665555544
   ...............................E,EEEEEEDDDDCA666'444443335899AAAA998896555444455555576656677678899ACCD...DA76666665556
   .............................EEEEEEEEEEDDDDA776444444433358 9AAAAA998776544445555544555886666788 7999ABCCE...EB86666655666
   .............................EEEEEEEEEDDDDDDB8854444444333589ABAAAA9876544456655555558987667788899ACDDE....D96696556776
   ............................EEEEEEEEEEEDDDDDC9964444444333 59AABBAAA98975666676655555698669777 98999ACDDE....EA9986667886
110............................EEEEEEEEEEEDDDDDCAA7444444333 59AABBAAA9899788888897677788 89999AAA999AACDD.....EB9986677886
   .............................EEEEEEEEEDDDDDDCAA8544444433399ABBBAAA999AAAAAA99998999988 9999999AABCDD.....EDAA99998888
   .......EEE.E.E.,E..E..E.EE..EEEEEEEDEDDDDDDCCCA8665543338 9ABBBBBAAABCCBBBRAAA998888988888889 9AABBCDD......DBBA8998888
   EEEEEEEEEEEEEDEEEEDDEEDDDDDDDDDDDDDDDDDDDDCCCDDDBA96644334 89ABBBBBBBCCCCCCCBAA99988999999 8999AAAABBDD......DCCA8886869
   ...........................EE.EEEEEDDDDDCCCDDCCBA976443349ABBBBBBBCCCCCCCBBAAAAAAAAA9999AAAABBBBBCEE......EDDB9888889
   ........................................DBB98866543349AABBBBBCCCCCCCCRBBBAAAAAAAAAAAABBBBBBBBD........EDDCA998999
   ........................................B99666765444449ABBBBCCCCDCDDCCCCCBBAAAAAAAAABBBCCCBBBBBCD........EDDDBA99999
   ........................................96666676544459ABBBBCCCCCDDDCCCCCBAAABBBBBBBCCCCCCBBBBBDE........EDDDCBAA999
   ........................................96666776544459ABBBBCCCCCCDDDDDCCBAABBBBBBBCCCCCCBBBBBDE........EEEDBBBAAAA
   .......................................E977679865444599BBBBCCCCDDDDDDDDDCBBBBBBBBCCBCCCBBBBBBCDE.........DDDCCCBBBA
120.......................................D96679996544459ABBBBCCCDDDDDEDDDCCCCCCCCCCCCCCBBBBBBCDDE.........EEECCCDDDC
   .......................................D97789AA6544459AABBBBBCDDDEEDDDDDDCCCCCCCCCCBBAAABBCDEEEE.......EEDCCCDDEE
   .......................................D98899AA6654459AABBBBBBDDDDDDDDDDDDDDDDCCCDDCCCAAAAABBBCDFEFE.E...EEEEBBBDDDE
   .......................................DA9989AA9654459AAABBBBBBCCCDDDDDDDDDDDDDDDDDCCBAAAAACBBBCDDDDDDDDDDDDDDBABBCDE
   .......................................DA999AAA7654458AAAABBBBCBCDDDDDDDDDDDDCCCDDCCBAAAACCAAAAAAAAAAAABBCCCDBAAABCD
```

Digitized by Google

```
    .........................................EA999AA965444589AAAAAABBBCDDDDDDDDDDDCCCCCBAAAAABAA99999999999ABBCCAAAABC
    ..........................................AAAAA9664433489AAAAAAABBCDDDDDDDDDDDCCCCCBA999AAA98969888989989AAAABAAAABB
    ..........................................CAAAA9765444489AAAAAAABBCCCCDDDCDDCCCBBBBAAA99999976667777767799889AA999AB
    ..........................................DAAAA9665444489AAAAAAAABBBBCCCCCCCCBBAAAA999888654235666666676555599998999
130.......................................... EAAAA866544447999AAAAAABABBBBBBBBBBBAAAA99888666632112455666653222478899999
    ..........................................EBBAA866544446999999AAAAAAABBBBBBAAAAA999988896655321123555555322224688889
    ...........................................CCAA8665544469999AAAAAAAAAAAAAAAAA9999888999666532334455544443333356888888
    ...........................................DDBA765554444599999999AAAAAAAAAA99988889999986554444443333333444445899879 8
    ...........................................EEBA8655655458999999999999999999999888889AA9776555554443333344444459AA9899
    .............................................DA86656665599999999999999999999877997899AAA99766655444333334555556ABBA986
    .............................................EA96667666899999888888999999888676776 89AAAA9989666554433333555668ABCCA97
    ..............................................A99998667999998888888888888899766766799AAA9998666665544334456677 9ABCDCA8
    ..............................................A999986679999889988888988887776677778 99AAA999876666544444556788 9ABBCBA9
    ..............................................A968796668999988888988889888766677778999999888666665444455567999AABBBAA
    ..............................................E979866668999888868989889967666779989999988886766655555556 79999AABBAAA
140...............................................DA998776999888886869888776676767779999988888977776555566669899AAAAAAA
    ................................................DA9877699988966678869877766777677888888789989896666667799999AAAAAA9
    ................................................EBA8777899886997777777766767677786896778889888896778899888889999AA99
    .................................................DA988689989899777767677677776666666788886665555666796655556 6799998
    .................................................EBA99889989977676767677777666555666666665544433333344443334444567998
    ..................................................EAAA9799886776666667777976655556565544433222222222222222122234556 77
    ...................................................CAA8689986776766667777976655444555433222111111111111111000012335 66
    ....................................................DDA6899897767666667787665443333322211111334444433333332222111122234
    .........................................................EA99776666666667665432221111123445666666666666666666653211111
    ..........................................................DA866766666666666543222133445666776666666666666666676553221
150.......................................................... EB96676666666676655444445566666555555555545555566676665 42
    ..........................................................A9777666666666655555666565555555544332333333333455566676 65
    ..........................................................CA88766666666655555666665555444333222222223333333355667766
    ..........................................................DA99666666666655555667766555544433222222223333333355667797
    ..........................................................DA99666666666655555666666555544433222222222223244445667786
    ..........................................................DA99666665555555556666666555544432222233232333444455666767
    ..........................................................DA99866555555545555666656655554433222233333232335555666666
    ..........................................................DA998665555555445555555555554443332222233233334445666666
    ..........................................................C988976555555544445555555555544332323222223233334445666666
    ..........................................................C9999766555555555555555555555444333333333332333334556666 6
160.......................................................... B96677665544444555555554555555555554444444444555555666666
    ..........................................................C96697665544444455555555555566655555555545555656666666666
    ..........................................................B96676665544434455555555566666665555555554556667776667776
    ..........................................................B86666666655444455555555556666666555555555455666666667676
    ..........................................................B96676666554454554555566666665655655555444555556667776676
    ..........................................................B86665666655444444455555556555555555544444555556656666676
    ..........................................................B76665566555554444455555555555555555544444555555555555666
    ..........................................................B66655556655555444445555555555544455554433334445555555 5666
    ..........................................................B66655555555555444444444444444433343333222333344444445555556
    ..........................................................C6555555555555544444333344444433322222222222333333344455555
170.......................................................... B76655544555544444433333333333322222222222323333333335555
    ..........................................................A7555444444555444444333222222222221222222222222233222 33345
    ..........................................................B75554444444555444443222212222222111111112111111222222 23345
    ..........................................................A7555444444455544444322211111111111111111111111111222 2345
    ..........................................................B655544433334444333332221111111111111111111111111111112355
    ..........................................................A65544343333334443333222111111111101111111111111111122355
    ..........................................................A655444333233333333222211111100010000000100110111111123455
    ..........................................................A655443332222333322222211110000000000000000000000001123355
    ..........................................................8655443322222222232222221111100000000000000000011111233455
```

Digitized by Google

```
   .............................................................E96554333222222322222221111111111000000000000011112233455
180.............................................................A755543332222222222222111111111100000000000001111112233345
   ............................................................E96544433322222222112211111111111100000000000011111223345 5
   ............................................................D866554433222222222221111111111111000000010011111113233455
   ............................................................A86555543332222222222111111111111100000000000111111333455 4
   ............................................................D8655544332221111122111111111111110000000000011122232334 55
   ............................................................E8655543332222211222111111111111110000000000011122233345 53
   ............................................................E96555443332222222222222211111111111110000000111222333444 3
   .............................................................DA6555433322222211222222111111111110000000011112223333333
   ..............................................................A65554433222222112221111111111111100000000111112223333 33
   ..............................................................DA775443322222111122111111111111111000000011111122323333
190..............................................................EDAA754433222221111222111111111111111000001111112322333 3
   ................................................................EEA54433222222122222211111110111000001111111112222333 3
   ..................................................................D754333332221222221111111111111000001111111122222333
   ..................................................................EB65433322222222211111111111111100011111111111222323
   ...................................................................EA754333322222222211111111111111111111112111222232 2
   ....................................................................EA6543333222211111111111111111111111111222222222 22
   .....................................................................EB64333322211111111111111111111111111111111111222
   .............................................................E.E......E964433222111112222221111111111111111111111111121
   .............................................................EEEEE.....E96643222211111122221111111111111111111111111111
   .............................................................EEEEEE....EEAA53222111111111111111111111111111111111111111
200.............................................................EEEEEE.....EEA42221111111111111111111111111111111111111111
   ............................................................EEEEDDE........E632211111111111111111111111111111111111111
   .............................................................EEEEEE.........9642111111111111111111111111111111111111111
   .............................................................EEDDDDE........DA6421111111111111000001111111111111100000
   ............................................................EEDDDDEE........EDB8421111111111110000000111111111110001 11
   ..........................................................EEEDDDDDDE..........EB632211111111111111111111111111100111 11
   ..........................................................EDCCCCCCDEE.........ED96221111111111111111111111111111111111
   ..........................................................EDDCCCCCCDE..........ECA55211111111111111111111111111111111 1
   ..........................................................EEDCCBCCCDEE..........EB99411111111111111111111111111111111 1
   ...........................................................EDDCBBBBDEE..........DA66411111222211111111111111111111111 1
210............................................................EDDBBBBCDE.........D854443111111222211111111111111111111 111
   .............................................................EDCBBBCDE.E......D7433356532111112222211111111111111111 11
   ..............................................................DDCCBBCDEE.....D643333579653111122222222111111111111111 2
   ..............................................................EECCBBCDDEE...E954334456999642211122222222211111111111 1A
   ...............................................................EDDCBBCDEE...C64334555689AA9553111222222221111111111 15B
   ................................................................EEDBBBDDEEEEA543455569899AAAA7312222222221111111111 19D
   ..................................................................ECBBCCDDDE74335555796899ABBA432222222222222111111 29C
   ...................................................................CBBBCDDDD64335566776999AAAA642222222222221111111 AAC
   ...................................................................DCBBBBCCC64345555666899AAAA642222222222222222212BA B
   ...................................................................ECBBBBCCB5434555556689AAAAB9432223323222222222159C B
220.................................................................DCBBBBB5434555556799999AA84423233333333222222229 99
   ....................................................................EDBBBAAB544555555667889999753233333333333322224998
   .....................................................................ECBBAAA65556666667798999965433333444333333323A998
   ......................................................................DCBBBB7567998877678998898665555555555444569ABCB8
   ......................................................................ECBBBB85689AAAA99989888989ABBBBBAABAABCCE..EDCA7
   .......................................................................DCBBBA6679ABBBAA9888899899D..............EDBA86
   ........................................................................DCCBA9678ABBCBAA988888999AD............EDBA966
   ........................................................................EDDCBA9889AACDBBBAAAAAAAABD............DBA9676
   .........................................................................EEDCBAA9999BDDDDDDDABBBBCE.........EEECBA6667
   ...........................................................................ECCBBBAAAADDEEEDDDCBBABBDEE......EEEBA87777
230............................................................................DCBDDCBBACDDDEDCCBBBBBCDEE.....EEEDA977679
```

Digitized by Google

Digitized by Google

```
KELH4                                   17-JUL-70 15:32
   ITH=15
   167       177       187       197       207       217       227       237       247       257       267
020DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCDDCCCDDCDDDCCCCCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCCCDDCDDDCCDDDCCCCCCCCCCCCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDCDCCDDDDDDCCCDDCCCDCCCDDCCCCCCCCCCCCCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDCDDCCCCCDCCCCCCCCCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   DDDDDDDDDDDDDDDDDDDDDDDDDDDCDDDDDDDDDDDDDDDDCCDDCCCCCDCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
   DDDDDDDDDDDDDDDCCCDDDDDDDDDDDDCDDDDDDCCCCCDDDCCCCCCCCCCCCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBB
   DDDDDDDDDDDDDDDDDDDCDDCCCDDDCCCDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCBB
   DDDDDDDDDDDDDDDDDDCCDDCCDDCDDDDCDDDCCCDCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCBBBBBB
   DDDDDDDDDDDDCDDDDDDDDDDDCDCDDCCCCDDDDCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCBBBBBBCB
   CDCCCDDCCCCCCCDCDDDCCCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCBBBBBBBBBBB
030CDCCCCCDDDDCDCCCDCCDDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCBBCCCCCBBBBBBB
   CDCCCCDCCCDCCCCCCCCCDDCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCBBBBBBBBBBBBBBB
   CCCCCDCCCCDCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBCBBBCBBBCBBBBBBBBBBB
   DCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCBBBBBBCCBBBBBBCCBBBBBBBBBB
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCCCCBBBCCCCCBCBBCCBCBBBBBBBBBCBBBBBBBBBBBBBBBB
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCCCCBBBCBBBCBBBCCBBBBBBBBCBBBBBBBBBBBBBBBB
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCBCCCCCCBCCBBBCBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCBCBBBBCBBBBBBBBBCCBBBBBBBBBBBBBBBCBBBBBBBBBBBB
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCCCCCCCBCCBCCBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCCCCCCBBBBCCBCCBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
040CCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCCBCBCCCCCCBCCCCCBBCBBBBBBCCBBBBBBBBBBBBBBBBBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBB
   BCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBCCCBBCBBBCCCCCBBBBBBBBBBBBBBBBBCBBBBBBBBBCBBBBBBBBBBBBBBB
   BCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBCCCCCCCBCCCCCCCBCCBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   CCCCCCCCCCCCCCCCCBBCCCCCCCCCCCCCCCCCCBBCCCCCBBBBBBBBBBBBBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   BBCCCCCCCCCCCCCCCCCCBCCCCCBCCCCCCCCBBCCBCCBBBCBBBBBBBCBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   9ABBBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBCCCCCBBBBBBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   7799AABCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCBBBCCBBBCBBBBBCCBBBBCBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   5567999ABBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCBCBBBBBCCBCBCBBCCCBBBCCBBBBBBBBBBBBBBBCBBBBBBBCBBBBBBBBBBBBBBBBBBBBBB
   55556799AABCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCBCCBCCCBBBCCBBBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   55444567999BCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBBBBBCCBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
05254433456779ABBCCCCCCCCCCCCCCCCCCCCCCCCCBCCCCCCCCBBBBBBBCBBBBBBCBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   554333356669AABBBCCCCCCCCCCCCCCCCCCBBBCCCCCCBBCCBBBBBCCCCBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBB
   55444334555 89ABBBBBBCCCCCCCCCCCCCCCCCBBBBBBCBBBBBBBBCBBBBBBBBBBBBBBBBBBBCCBBBBBBCBBBBBBBBBBBBBBBBBBBBB
   5555444444456889ABBBCCCCCCCCCCCCBCCCCCBCBBBBBBBCBBBBBBBBBCBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   55555444444456899ABBBCCCCCCCCCCCCCCCCCCBBBCBBBCBBBBCCBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   5445544444444456799AABCBBCCCCCCCCBBBCCCCCBBBCCBBBBBCBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   544444444444456899ABBBBCCCCCCCCCCCCCCCBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   5544444444444455688 9BBCBCCCCCCCCCCCCCBCCCCCCCCCCCCBBBBBBBBBCBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   55554444444433344557AABBBCCCCCCCCCCCCCBBBBBCCBCCBBCCBBBCBBBBBBBBCBBBBBBBBBBBBBBBBBABBBBBBBBBBBBBBBBBBBBBB
   6655555444443444455589ABBBCCCCCCCCCCCCCBBCBBBBBBBBBBBBBCCBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
06055555555555444444444699ABBCCBCCCCCCBBBCCCCCBCCBBCBCBBCCCBCBBBBCCCCBBBBBBBBCCBCBBBBBBBBBBBBBBBBBBBBBBBBBB
   5544444455544544444445799ABBBBBCCCBCCCCCCCCBBCCBCBBBCBBB3BBBBBBBBBBBCBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   44333334444444444444445799ABBBCBBBCCBBBCBCBBCBBBBBCCBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   54332333333344444444444568992ABBCCCCCCCCCBBBCBBCCCCCCCBBBCBBBBBBBBCBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   5543333322333333333344456799 8ABBBBCCCCBBBCCCBBBCBBBBBCCBCCBBBBBBBCBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   55444433333232333334444567798ABBBCBBBCCBCCCBBBCCCBBBCCBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBB
   44444454444433333334444445666699ABBBBBCBBBBBBBCCCBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBB
   344444444554444333334444444556668 9AABBBBBBBBCBBBCBBBBCBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   4444555544444444333334433335566669 9ABBBBCCCCCCCCBCCBBBBCBCBBCCBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   44455555555433333333233232344566668 9AAABBBCBBCCBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
07044444554444333232222232323335666668AAAABBBBBBBBBBBBBBBBBBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

Digitized by Google

```
   44455555555444333332222334445666799AAABBBBCBBBBBBBBBBBBBBBCBBBBBBBBCBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   5555556566555444333233334444466667999ABBBBBBBBBCCBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   66666767666555544443333333444456666899ABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   777888888866665544433333333334566666889ABBBBBBBCBBBBBBBCCBBBBCBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   88899999998876665554323333333556656668AABBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   999AAAAAAAA9876655553332233345555556669ABBBBBBBBBBBBBBBBBBBCBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   9AABBAAAAAAA98666555322223344455555689ABBBBBBBBBBBBCBBBCBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   AABBBBBBAABAA867666543222233334444555589ABBBBBBBBBBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   BBBBBBBBBBBBA967666543222233333444444579AABBBBBBCBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
080BBCCBBBCBBBBA9666666544323233333333344568AABBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBABBBBB
   CDDCCCCBBBBBBA766555443333333333333335589ABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAABBBB
   DDDCCCCCBBBCBBA7655433333443443333344558 9ABBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   EEDDCCCBBBBCCCBA866544444444445444445579ABBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   E,EEDDCCCCCCCCCCA886555554444455455555568ABBBBBBBBBBBBBBBBBBBBBBBBCBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   ...,EEDDDDDDCCCCCBAA7655555555545555555668ABBBBBBBBBBBBBBBBBBBBCBBBCBBBBCBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBB
   ....,EEDDDDDCCDCCCAA965554444444555555556668AABBBBBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBB
   ....,EEDDDDDDDDDCBBA7655444445555555566689ABBBBBBBBBBBBBBBBBBBBBBBCBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBABBB
   ....,EEEDDDDDDDDCBBA8665443344455555565679AABBBBBBBBBBBBBBBBBBBBBBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBABBBBBBB
   ......EEEEEDDDDDCCCA8865433333345555555679AABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBCCBBBBCBBBBABBBBBBBBBBBBBAABBBB
090EEEEEEEEEEEDDDDDCCCB986544333334455555556689ABBBBBBBBBBBBBBBBCBBBBBBBCBBBBBCCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   DDDDEEEEEEEEEEDDDDDCCB986544333334455555556689ABBBBBBBBBBBBBBBBBBBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBABA
   CCDDDEEEEEEEEDDDDDDDC9865543333344555555555689ABBBBBBBBBBBBBBCBBCCBBBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAABBB
   BCCCDDDEEFEFEEEDDDDCA8655433333455555555555689ABBBCCCBBBBBCCBBCBBBCCCCCBCBBBCBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
   BBCCCDDDEEEEEEEEEDDDCA866443333345555555454569ABBBBCBBCCBBBCCCCCCCCCCCCBBBCCBCCCBBBBBBBBBBBBBBBBBBBBBBBBBB
   BBBBBCDDDDEEEEEEEEDDDA96654333334555555554569ABBBBCCCCCBBBCCCCCCCCCCCCCCCCCCCCBCCCCCCBBBBCBBBBBBBBBBBBBBBBB
   AAAAABBBBBCDEEEEEDDDA96644333335654455554569ABBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBBBCBBBBBBBBBBBBBBBBBBBB
   778889999AABCDDEEDDDB976433334555445554569ABCCCCCCCDCCCCCCCCDCCCCCCCCCCCCCCCCCCCCCCBBBBBBBBBBBBBBBBBBBB
   5556677888899ABCDDDDDBA665333355555445544568ACCCCDDDDDDDDDDDDDDDDDDDDDDDDDDCCCCCCCCCCCCCCBBCCCBBBBBBBBBBB
   43445566666899ACDDDDCA75433335554455544569BCDDDEEEEEEEEEEEEEEEEEEEEEEEEDDDDDDDDDDDDDDDDCCCDCCCCCCCCCCCC
1004333344566667789ABCCDBA75533344444445544569BDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDD
   5444445555566789ABBCCA955333444555555544569BDDDDEEEEEEEEEEEEEDEEEDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEED
   555555555555667799AABBA8553333455555544589BDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEED
   5455666666766788999ABA96533333555554444458ACDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDEDEDDDDDDD
   4445556677777888999ABA9553333455565444568ACDDDDDDDDDEDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDEED
   4433344566678889999ACB965322345566544 4568ACDDDEDDDDEDDEDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDEED
   65444334555788999AAACCA653223456664444568ACDDDDDDDDDDDDDDEDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEED
   65554435554679999AAACCA653223456655444569ACDDDDDEEDEEEEEEDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDD
   654444566655689 99AAACDB74422355655444458ABCDDDEEEDDEEEDEEDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDE
   66444569885456889AABDDC95422355655444459ABCDDDDDDDDDDDDEDDEDDDDDEDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDDEDDD
11066555689666556799AAADDCA6533356654433579ACCDDDDDDDDDDDEEDDDEEEEDEEEEDDDDDDEEEEEEEEEEEEEEEEEDEEDDDDDDDDDD
   866556776666668999ADDDA643325655444449ABCDDDDDDEDDDDEEEEDDDDDEEEEDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDEDD
   8887967888996679999ACDDB644435554444459BCDDDDDDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
   9866998888998898999ACDDC95443554434556ABCDDDDDDEDDDEDDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDDDDDDDD
   96778889999AAA999AAADEED95443454334558BBCDDDDDDDDDDDDEEDDEDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDDDDDD
   988999999AABBBBBBBBBDDEDA6554553335669BCCDDDDDDDEDDDEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEDEDDDDDDED
   999999AAAABBCDDDDCDDEDA655455435688ACDDDDDDDDEEEEEEEEEEDDDDDDDDDDDDDDDEEEEEEEEDDDDEEEEDEEDEEDDEEEEEEEE
   99999AAABBCCCDDDDDDDDEDDB7554455689AAD...........................................,EE,EEEEEEDDDDDDDDDDDDDDDD
   A9999AABBBCDDDDDEEEEEEEEB755456899ACCE.....................................................................
   AAAAAABCCCDDDDDDDEEEEEEEDC855458676799D....................................................................
120CCBAABCDDDDEEDDEDEEEEEDDC855457676688B.....................................................................
   EEDDDDDDDDDEEDDEDDDEEDDDD955556766677A.....................................................................
   EEEEEDDDDDDDDDEDDDDEDDDDA6645676667779.....................................................................
   EE,EEEEEEEDDDDDDDDDDDDDDD9664567766779.....................................................................
   DD,E,E,EEEEDCCCCCCCCDDDDDD9665578877779.....................................................................
```

Digitized by Google

Digitized by Google

```
   CDEEEEEEEEEDCBBBCCCCCDDC9665599977778................................................................
   BCDDEEEEEDDCCCBBBBBCDDDC8555799986998................................................................
   BBCCDDEDDDCDCCBBBBBCCDEB8555689989889................................................................
   9ABBCDDCCCCCCCBBBBBBCDDB6555588998889................................................................
   99BBBCCCCCCCCCBBBBBBCCDA6555579999999................................................................
13099AAABBCCCCBCBBBBBBBCDDA6555579AA999A................................................................
   89899AABBBBBBBBBBBBBCCCA6554568AAAAAA................................................................
   899899AAAABBBBBBBBBCCDD96554568ABAAAB................................................................
   9678899AAAABBBBBBBBCCDD95554568ABBBBD................................................................
   66778889999AABBBBBBCDEC85554568ABBCC.................................................................
   77777888999AAAABBBBCDDB65555689BBBDD.................................................................
   866777788899AAABBBBBCCEB6566568ABBB..................................................................
   9876657888899AAABAAACCDA7666669ABBD..................................................................
   A8766666778899AAAAAACDDBA999889BBC...................................................................
   A97666667788999AAAAACDDCBBBBBAABCD...................................................................
140A966666677889999AAAACDDAAAAAB99AD....................................................................
   9977666666788999AAAACDDA9999A79B.....................................................................
   9887667777789999999ACDD9899999BD.....................................................................
   8899767677689999999ACDC98999ACD......................................................................
   8978767677689999999ACDDA7999AD.......................................................................
   7667799966989999999ACCCA9AAAC........................................................................
   6556689888889999999ACDBBAACCC........................................................................
   4455679899889999999ACCAABBBBB........................................................................
   1245567788988898999ACA999ABBE........................................................................
   1134566677989988999ACA899BDD.........................................................................
1502235566777988988999ACDCD.............................................................................
   5444566777989888999A.................................................................................
   6545556777789889999A.................................................................................
   75555667777898899AAB.................................................................................
   66555566777899999AAC.................................................................................
   76665556667788999AAD.................................................................................
   6665556666677899ABB..................................................................................
   6665555566667999ADD..................................................................................
   6655555566667999B....................................................................................
   665555556666799AC....................................................................................
160665555556667789AC....................................................................................
   665555555567899AB....................................................................................
   6666665555679999A....................................................................................
   6666666655689999A....................................................................................
   6656666666699AA9A....................................................................................
   6655666666899999D....................................................................................
   6656666677899999C....................................................................................
   6666667777999989D....................................................................................
   6666667788899989E....................................................................................
   5676667888889989E....................................................................................
1705676667888999989E....................................................................................
   5667678988989989E....................................................................................
   566678986688889AD....................................................................................
   556767787696989AD....................................................................................
   566767676987889AD....................................................................................
   556667676677889A.....................................................................................
   556676666667889A.....................................................................................
   555666566666889A.....................................................................................
   556655566556889A.....................................................................................
```

195

Digitized by Google

```
   555555555556789A.......................................................................................
18Ø555555555556789A.......................................................................................
   555555555556689A.......................................................................................
   554554555555679A.......................................................................................
   443554445555679A.......................................................................................
   5345555555555689E......................................................................................
   3344455444555689C......................................................................................
   3344444555556789B......................................................................................
   3343345555556789A......................................................................................
   3333234555556688A......................................................................................
   3332334455556678A......................................................................................
19Ø3332333444456668A......................................................................................
   3332323455555658.......................................................................................
   3332223555445449.......................................................................................
   32222333334443AB.......................................................................................
   22222323334333B........................................................................................
   22222223223325C........................................................................................
   22222222222249.........................................................................................
   111122222222BD.........................................................................................
   111112111115A..........................................................................................
   11111111111A...........................................................................................
2ØØ11111111124D...........................................................................................
   1111111122B............................................................................................
   11111111AAD............................................................................................
   Ø1111115AA.............................................................................................
   1111Ø129...............................................................................................
   1111128E...............................................................................................
   111Ø19C................................................................................................
   11Ø15B.................................................................................................
   1115B..................................................................................................
   1129E..................................................................................................
21Ø11768A.................................................................................................
   1A86459E...............................................................................................
   2A86533A...............................................................................................
   AC654324BB.............................................................................................
   BD755323BBD............................................................................................
   DD865422338............................................................................................
   CB67653222B............................................................................................
   CB766432225............................................................................................
   BB766542224............................................................................................
   B9665542224............................................................................................
22Ø97555432222............................................................................................
   86555442222D...........................................................................................
   86555543334D...........................................................................................
   86566554334D...........................................................................................
   76666554444D...........................................................................................
   66666655445E...........................................................................................
   66776665556............................................................................................
   66889867668............................................................................................
   78999988669............................................................................................
   799AAAA9889............................................................................................
23Ø99BBBBBAAAB............................................................................................
```

Digitized by Google

APPENDIX C

Programs for locating the outline of the head

Digitized by Google

Digitized by Google

```
      SUBROUTINE MAIN
C     MAIN CONTROL FOR HEAD PROCESSING.
      IMPLICIT INTEGER (A-Z)
      COMMON /HEDMES/ XL,XR,YT,MEASX(4),MEASY(4)
      COMMON /CON/ ADUM(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
     1       TVPNTR,TVSIZ,NPERWD,ILEN,BDUM(8),ITEXT(10)
      INTEGER BITS,FLINE,RSIDE,TVPNTR,TVSIZ
      COMMON /PTRS/ KPTR,NPTR,IPTR(20)
      COMMON /FLAGS/ NUTN1,PICNA1,INVERT,NUTN2,KEEPLP,NUTN3(3),
     1                USETV,KG1G2,NUTN4(40)
      COMMON /OUTFGS/ IFDTTY,IFDLPT,IFDDPY,IYORN

      DATA OUTLINEONLY/.FALSE./

      CALL GIVEUP(-1)
      CALL KILLLL

      HEDPOG=IADPST(400)

C     READ IN PICTURE OF HEAD.
      IF (USETV.NE.0) GO TO 100
      IF (PICNA1.NE.0) PICNA1=3
      JP=INDAT(0)
      GO TO 200
100   CALL TVSET(LENPB)
      CALL GIVEUP(-1)
      JP=INTV(0)
      CALL DECDMX(IPTR(JP))
C     JP IS ORIGINAL HEAD.

C     INITIALIZE G1 AND G2.
200   CALL FILCON(JP,ADUM)
      KG1G2=-1
      CALL G1(LSIDE)
      CALL G2(FLINE)
      KG1G2=0

      JP2=IRED3(JP,8)
C     JP2 IS JP REDUCED IN SIZE BY A FACTOR OF 8.
      KEEPLP=-1
      IF (KRAPPE.NE.0) CALL PRTPIC(0)

      JP3=JSURE3(JP2)
C     JP3 IS ALL OF THE EDGES FROM JP2.
      IF (IFDDPY.EQ.0) GO TO 330
      PLNPOG=IADPST(1000)
      CALL DSLINS(JP3,250,450,PLNPOG)
330   INVERT=-1
      IF (KRAPPE.NE.0) CALL PRTPIC(JP3)

      JP4=IHEDED(JP3)
C     JP4 IS THE EDGES FROM JP3 WHICH FORM THE BORDER OF THE
```

Digitized by Google

```
C        HEAD.
         IF (IFDDPY.EQ.Ø) GO TO 43Ø
         CALL SETPOG(PLNPOG)
         CALL DSLINS(JP4,25Ø,25Ø,PLNPOG)
 43Ø     INVERT=-1
         IF (KRAPPE.NE.Ø) CALL PRTPIC(JP4)

         JP5=JBIGHE(JP,HEDPOG)
C        JP5 CONTAINS THE EDGES IN THE ORIGINAL PICTURE WHICH
C        FORM THE BORDER OF THE HEAD.
         IF (IFDDPY.NE.Ø) CALL DPYOUT(HEDPOG)
C**      CALL PRTPIC(JP)
C**      CALL PRTPIC(JP5)
         INVERT=Ø

C        IDENTIFY BOUNDARIES OF THE HEAD.
         CALL ISIL(XL,XR,YT,DUMMY)

C        RELEASE BUFFERS.
         CALL IGIVEU(JP5)
         CALL IGIVEU(JP4)
         IF (IFDDPY.EQ.Ø) GO TO 73Ø
         CALL HYDPOG(PLNPOG)
         CALL IGIVEX(PLNPOG)
 73Ø     CALL IGIVEU(JP3)
         CALL IGIVEU(JP2)

         IF (OUTLINEONLY) RETURN

C        FIND FACIAL FEATURES.
         KPTR=JP
         CALL FDFEAT(HEDPOG)

         END
```

Digitized by Google

```
              FUNCTION JSURE3(ISPTR)
              IMPLICIT INTEGER (A-Z)
              COMMON /CON/ ADUM(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
              1         TVPNTR,TVSIZ,NPERWD,ILEN,BDUM(8),ITEXT(10)
              INTEGER BITS,FLINE,RSIDE,TVPNTR,TVSIZ

              IF (SWITCH) IDPTR=INITOP(ISPTR,3,'J SURE 3')
              IF (.NOT.SWITCH) IDPTR=INITOP(ISPTR,2,'2X2 MAR5')

              T=1

              DO 500 J=FLINE,LLINE
              DO 500 I=LSIDE,RSIDE
              IF (.NOT.SWITCH) GO TO 100
              IVAL=JEDGE3(I,J,-1)
              GO TO 460

C             1 |     2 -     4 /     8 \

C             A B
C             C D

100           A=LOAD(I,J)
              B=LOAD(I+1,J)
              C=LOAD(I,J+1)
              D=LOAD(I+1,J+1)
              IF (A-B) 140,200,120
C             A,GT,B
120           IF (MIN0(A,C)-MAX0(B,D),GT,T) GO TO 401
              GO TO 200
140           IF (MIN0(B,D)-MAX0(A,C),GT,T) GO TO 401
200           IF (A-C) 240,300,220
C             A,GT,C
220           IF (MIN0(A,B)-MAX0(C,D),GT,T) GO TO 402
              GO TO 300
240           IF (MIN0(C,D)-MAX0(A,B),GT,T) GO TO 402
300           CONTINUE
C         A:D
              IF (IABS(A-D),LE,T) GO TO 350
              IF (A-D) 330,350,310
C         A,GT,D
310           IF (A-MAX0(B,C,D),GT,T) GO TO 404
              IF (MIN0(A,B,C)-D,GT,T) GO TO 404
              GO TO 350
C         A,LT,D
330           IF (D-MAX0(A,B,C),GT,T) GO TO 404
              IF (MIN0(B,C,D)-A,GT,T) GO TO 404
C         B:C
350           IF (IABS(B-C),LE,T) GO TO 450
              IF (B-C) 380,450,360
C         B,GT,C
360           IF (B-MAX0(A,C,D),GT,T) GO TO 408
```

Digitized by Google

```
              IF (MINØ(A,B,D)-C,GT,T) GO TO 4Ø8
              GO TO 45Ø
      C    B,LT,C
      380     IF (C-MAXØ(A,B,D),GT,T) GO TO 4Ø8
              IF (MINØ(A,C,D)-B,GT,T) GO TO 4Ø8
              GOTO 45Ø
      401     IVAL=1
              GO TO 46Ø
      4Ø2     IVAL=2
              GO TO 46Ø
      4Ø4     IVAL=4
              GO TO 46Ø
      4Ø8     IVAL=8
              GO TO 46Ø

      45Ø     IVAL=Ø
      46Ø     CALL STORE(I,J,IVAL)
      5ØØ     CONTINUE

              JSURE3=IDPTR
              END
```

Digitized by Google

Digitized by Google

```
              FUNCTION INITOP(SPTR,N,NAME)
      C       INITIALIZE FOR N●N OPERATOR.
              IMPLICIT INTEGER (A-Z)
              DIMENSION NAME(2)
              COMMON /CON/ A(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
              1       TVPNTR,TVSIZ,NPERWD,ILEN,B(8),ITEXT(1Ø)
              DIMENSION Z(32)

              CALL FILCON(SPTR,A(1))
              CALL INLOAD(SPTR)
              DO 1ØØ I=1,32
      1ØØ     Z(I)=A(I)
      C       ALLOCATE 1 WORD PICTURE BUFFER.
              DPTR=IALLOC(1)
              CALL FILCON(DPTR,A(1))
              BITS=Z(4)
              IWID=Z(5)-(N-1)
              NPERWD=Z(13)
              LINLEN=(IWID+NPERWD-1)/NPERWD
              FLINF=Z(7)+(N-1)/2
              LLINE=Z(8)-N/2
              LSIDE=Z(9)+(N-1)/2
              RSIDE=Z(1Ø)-N/2
              ILEN=Z(14)-(N-1)
              ITEXT(1)=Z(23)
              ITEXT(2)=Z(24)
              ITEXT(3)=NAME(1)
              ITEXT(4)=NAME(2)
              CALL PUTCON(DPTR,A(1))
      C       NOW ALLOCATE NEEDED LENGTH.
              CALL ALOCAD(DPTR,LINLEN*ILEN)
              CALL INSTOR(DPTR)
              CALL FILCON(DPTR,A(1))
              INITOP=DPTR
              END
```

Digitized by Google

Digitized by Google

```
                FUNCTION IHEDED(ISPTR)
                IMPLICIT INTEGER(A-Z)
                COMMON /CON/ A(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
                1        TVPNTR,TVSIZ,NPERWD,ILEN,B(8),ITEXT(10)
                COMMON /LSENTS/ VBL,VLSB,VLST,VTL,VTR,VRST,VRSB,VBR

        C       LIST STRUCTURE ENTRIES:
        C
        C                 VTL     VTR
        C
        C       VLST                          VRST
        C
        C       VLSB                          VRSB
        C
        C            VBL              VBR

        C       ALLOCATE NEW BUFFER.
                IDPTR=INITOP(ISPTR,1,'HEAD EDGES')

        C       INITIALIZE FAIL DPY BUFFER.
                CALL IFAILB(IDPTR)

        C       ZERO NEW BUFFER.
        50      CALL ZROBUF(IDPTR)
                CALL IFAIL('IHEDED')
                CALL DELINE(-1,ISPTR,IDPTR)
                CALL FILCON(ISPTR,A(1))
                CALL INLOAD(ISPTR)

                LLIMIT=LLINE-ILEN/5

        C       FROM LEFT, LOOK FOR VERTICAL LINE.
                DO 1100 I=LSIDE,RSIDE
                KOUNT=0
                DO 1100 J=FLINE,LLIMIT
                IF ((LOAD(I,J).AND.1).EQ.0) GO TO 1050
                KOUNT=KOUNT+1
                GO TO 1100
        1050    IF (KOUNT.GE.1) GO TO 1200
                KOUNT=0
        1100    CONTINUE
        C-----------------------------------------------------------------
        CONSTRAINT:     THERE MUST BE A VERTICAL LINE REPRESENTING THE LEFT
        C               SIDE OF THE HEAD.
        C               IF NOT, FAILURE.
        C-----------------------------------------------------------------
                CALL FAIL('1100')

        C       FOUND:  LEFT SIDE OF HEAD.
        1200    LSX=I
                LSYT=J-KOUNT
                LSYB=J-1
```

Digitized by Google

```
              VLST=LESTAB(LSX,LSYT)
              V=VLST
              DO 1220 J=LSYT,LSYB
              IF (J.NE.LSYT) V=LLINS(V,LSX,J)
      1220    CALL STORE(LSX,J,LOAD(LSX,J))
              VLSB=V

      C       FROM RIGHT, LOOK FOR VERTICAL LINE.
              DO 1300 I=RSIDE,LSX,-1
              KOUNT=0
              DO 1300 J=FLINE,LLIMIT
              IF ((LOAD(I,J).AND.1).EQ.0) GO TO 1250
              KOUNT=KOUNT+1
              GO TO 1300
      1250    IF (KOUNT.GE.1) GO TO 1400
              KOUNT=0
      1300    CONTINUE
      C-----------------------------------------------------------------
      CONSTRAINT:      THERE MUST BE A VERTICAL LINE REPRESENTING THE RIGHT
      C                SIDE OF THE HEAD.
      C                IF NOT, FAILURE.
      C-----------------------------------------------------------------
              CALL FAIL('1300')

      C       FOUND:  RIGHT SIDE OF HEAD.
      1400    RSX=I
              RSYT=J-KOUNT
              RSYB=J-1
              VRST=LESTAB(RSX,RSYT)
              V=VRST
              DO 1420 J=RSYT,RSYB
              IF (J.NE.RSYT) V=LRINS(V,RSX,J)
      1420    CALL STORE(RSX,J,LOAD(RSX,J))
              VRSB=V

      C-----------------------------------------------------------------
      CONSTRAINT:      THE LEFT AND RIGHT SIDES OF THE HEAD MUST BE SEPARATED.
      C                IF NOT, FAILURE.
      C-----------------------------------------------------------------
      C       ARE THEY LEFT AND RIGHT?
              IF (RSX-LSX.LT.4) CALL FAIL('1500')

      C       FROM TOP, BETWEEN LSX AND RSX, LOOK FOR
      C       HORIZONTAL LINE.
              DO 1600 J=FLINE,LSYT
              KOUNT=0
              DO 1600 I=LSX,RSX
              IF ((LOAD(I,J).AND.2).EQ.0) GO TO 1550
              KOUNT=KOUNT+1
              GO TO 1600
      1550    IF (KOUNT.GE.1) GO TO 1700
              KOUNT=0
```

Digitized by Google

Digitized by Google

```
1600    CONTINUE
C-----------------------------------------------------------------------
CONSTRAINT:      THERE MUST BE A HORIZONTAL LINE REPRESENTING THE TOP
C                OF THE HEAD.
C                IF NOT, FAILURE.
C-----------------------------------------------------------------------
        CALL FAIL('1600')

C       FOUND:  TOP OF HEAD
1700    TLX=I-KOUNT
        TRX=I-1
        TY=J
        VTL=LESTAB(TLX,TY)
        V=VTL
        DO 1720 I=TLX,TRX
        IF (I.NE.TLX) V=LRINS(V,I,TY)
1720    CALL STORE(I,TY,LOAD(I,TY))
        VTR=V

C-----------------------------------------------------------------------
CONSTRAINT:      THE TOP OF THE HEAD MUST BE ABOVE THE SIDES OF THE
C                HEAD.
C                IF NOT, THE SIDE(S) IS(ARE) IN ERROR.  DELETE SIDE(S)
C                          AND TRY AGAIN.
C-----------------------------------------------------------------------
C       KABOVE CHANGED FROM 3 TO 2 ON 8 JUN 70 WHILE DEBUGGING RPHH1
        KABOVE=2
        IF (LSYT-TY.GT.KABOVE) GO TO 17320
        CALL NFATAL
        CALL FAIL('1732')
        CALL YFATAL
        CALL DELINE(0,VLSB,VLST)
        IF (RSYT-TY.GT.KABOVE) GO TO 17330
        GO TO 17325
17320   IF (RSYT-TY.GT.KABOVE) GO TO 17340
17325   CALL NFATAL
        CALL FAIL('1732')
        CALL YFATAL
        CALL DELINE(0,VRST,VRSB)
17330   CALL KILLLL
        GO TO 50

C-----------------------------------------------------------------------
CONSTRAINT:      THE TOP OF THE HEAD MUST BE BETWEEN THE SIDES OF THE
C                HEAD.
C                IF NOT, THE TOP IS IN ERROR.  DELETE THE TOP AND
C                          TRY AGAIN.
C-----------------------------------------------------------------------
17340   IF (LSX.LT.TLX-2.AND.RSX.GT.TRX+2) GO TO 1738
        CALL NFATAL
        CALL FAIL('1734')
        CALL YFATAL
```

Digitized by Google

```
              CALL DELINE(Ø,VTL,VTR)
              CALL KILLLL
              GO TO 5Ø

      C       ASSERT:  TENTATIVE SIDES AND TOP OF HEAD HAVE BEEN LOCATED.
      C                LEFT    LSX,LSYT,LSYB   VLST,VLSB
      C                RIGHT   RSX,RSYT,RSYB   VRST,VRSB
      C                TOP     TLX,TRX,  TY    VTL,VTR
      C       SEARCH FOR LONG, FAIRLY STRAIGHT, SIDES OF THE HEAD.
      C       LONG AND FAIRLY STRAIGHT ARE DEFINED IN TERMS OF
      C       HEAD WIDTH AND Y COORDINATE OF TOP.
      1738    HEDWID=RSX-LSX
      C       FAIRLY STRAIGHT:  MUST BE IN BAND FSBAND WIDE.
              FSBAND=2+(HEDWID/1Ø)
              ININC=FSBAND/2
              OUTINC=FSBAND-ININC-1
      C       LONG:  SEARCH BAND FROM BNDTOP TO BNDBOT.
              BNDBOT=TY+HEDWID-FSBAND
              BNDTOP=TY+FSBAND
      C-----------------------------------------------------------------
      CONSTRAINT:      BAND MUST BE ENTIRELY WITHIN PICTURE.
      C                IF NOT, SIDE CLOSEST TO EDGE IS IN ERROR.  DELETE
      C                    IT AND TRY AGAIN.
      C-----------------------------------------------------------------
              IF (BNDBOT.LT.LLINE-2) GO TO 174Ø
              CALL NFATAL
      1739    CALL FAIL('1739')
              CALL YFATAL
              LDIST=LSX-LSIDE
              RDIST=RSIDE-RSX
              IF (LDIST-RDIST) 183Ø,187Ø,187Ø

      174Ø    RIGHT=1
              LEFT=-1
      C       LEFT TOP.
              IF (BNDTOP.GE.LSYT) GO TO 1741
              CALL SIDSRC(VLST,RIGHT,LSX-OUTINC,LSX+ININC,LSYT-1,BNDTOP)
              LSYT=LYCOR(VLST)
      C       RIGHT TOP.
      1741    IF (BNDTOP.GE.RSYT) GO TO 1742
              CALL SIDSRC(VRST,LEFT,RSX+OUTINC,RSX-ININC,RSYT-1,BNDTOP)
              RSYT=LYCOR(VRST)
      C       LEFT BOTTOM.
      1742    IF (BNDBOT.LE.LSYB) GO TO 1743
              CALL SIDSRC(VLSB,LEFT,LSX-OUTINC,LSX+ININC,LSYB+1,BNDBOT)
              LSYB=LYCOR(VLSB)
      C       RIGHT BOTTOM.
      1743    IF (BNDBOT.LE.RSYB) GO TO 1758
              CALL SIDSRC(VRSB,RIGHT,RSX+OUTINC,RSX-ININC,RSYB+1,BNDBOT)
              RSYB=LYCOR(VRSB)

      C       FOLLOW EDGES FROM SIDE TO TOP OF HEAD.
```

Digitized by Google

Digitized by Google

```
1758    V=VLST
        CALL UHEF(1,LXCOR(VLST),LSYT,TLX,TY,V,FAILSW)
        IF (FAILSW.EQ.Ø) GO TO 176Ø
C-----------------------------------------------------------------
C       UHEF FAILED.  LEFT SIDE OF HEAD MUST BE IN ERROR.
C       REMOVE "LEFT SIDE LINE" FROM SOURCE PICTURE AND TRY AGAIN.
C-----------------------------------------------------------------
        GO TO 183Ø

176Ø    CALL LLINK(V,VTL)
        V=VRST
        CALL UHEF(-1,LXCOR(VRST),RSYT,TRX,TY,V,FAILSW)
        IF (FAILSW.EQ.Ø) GO TO 178Ø
C-----------------------------------------------------------------
C       UHEF FAILED.  RIGHT SIDE OF HEAD MUST BE IN ERROR.
C       REMOVE "RIGHT SIDE LINE" FROM SOURCE PICTURE AND TRY AGAIN.
C-----------------------------------------------------------------
        GO TO 187Ø

178Ø    CALL LLINK(VTR,V)

C       FOLLOW EDGES FROM SIDE TO BOTTOM OF PICTURE.
        V=VLSB
        CALL LHEF(1,LXCOR(VLSB),LSYB,LLINE,V,FAILSW)
        VBL=V
        IF (FAILSW.EQ.Ø) GO TO 185Ø
C-----------------------------------------------------------------
C       LHEF FAILED.  LEFT SIDE OF HEAD MUST BE IN ERROR.
C       REMOVE "LEFT SIDE LINE" FROM SOURCE PICTURE AND TRY AGAIN.
C-----------------------------------------------------------------
183Ø    CALL DELINE(Ø,VLSB,VLST)
        CALL KILLLL
        GO TO 5Ø
185Ø    V=VRSB
        CALL LHEF(-1,LXCOR(VRSB),RSYB,LLINE,V,FAILSW)
        VBR=V
        IF (FAILSW.EQ.Ø) GO TO 19ØØ
C-----------------------------------------------------------------
C       LHEF FAILED.  DO AS ABOVE.
C-----------------------------------------------------------------
187Ø    CALL DELINE(Ø,VRST,VRSB)
        CALL KILLLL
        GO TO 5Ø

C       REMOVE CONCAVITIES IN SIDES.
19ØØ    CALL REMOCC(VLSB,VLST,1)
        CALL REMOCC(VRST,VRSB,-1)

C       RELEASE FAIL DPY BUFFER.
        CALL RFAILB

        IHEDED=IDPTR
```

Digitized by Google

Digitized by Google

```
25100       END
```

Digitized by Google

Digitized by Google

```
            SUBROUTINE SIDSRC(VPTR,LORR,XSTART,XEND,YSTART,YEND)
            IMPLICIT INTEGER (A-Z)
      C     SIDE SEARCH.

            XSTEP=1
            IF (XEND.LT.XSTART) XSTEP=-1
            YSTEP=1
            IF (YEND.LT.YSTART) YSTEP=-1
            DO 200 J=YSTART,YEND,YSTEP
            DO 100 I=XSTART,XEND,XSTEP
            L=LOAD(I,J)
            IF ((L.AND.13).NE.0) GO TO 150
      100   CONTINUE
            GO TO 200
      150   VPTR=LPINS(LORR,VPTR,I,J)
            CALL STORE(I,J,L)
      200   CONTINUE
            RETURN
            END



            SUBROUTINE REMOCC(V1,V2,WHERE)
            IMPLICIT INTEGER (A-Z)
            V=V1
      10    X1=LXCOR(V)
      20    V=LRSIB(V)
            IF (V.EQ.V2) RETURN
            X2=LXCOR(V)
            IF ((X2-X1)*WHERE) 30,30,40
      30    X1=X2
            GO TO 20
      C     X2 IS INWARD OF X1.
      40    K=1
      50    V=LRSIB(V)
            IF (V.EQ.V2) RETURN
            X2=LXCOR(V)
            IF ((X2-X1)*WHERE) 100,100,60
      C     STILL INWARD.
      60    IF (K.GT.4) GO TO 10
            K=K+1
            GO TO 50
      C     FIX UP K OF THEM.
      100   W=V
      110   W=LLSIB(W)
            TX=LXCOR(W)
            TY=LYCOR(W)
            CALL LSETXY(W,TX-WHERE,TY)
            CALL STORE(TX,TY,0)
            CALL STORE(TX-WHERE,TY,15)
            K=K-1
            IF (K.GT.0) GO TO 110
```

Digitized by Google

Digitized by Google

```
Ø53ØØ        GO TO 1Ø
Ø54ØØ        END
```

Digitized by Google

Digitized by Google

```
          SUBROUTINE DELINE(SW,P1,P2)
          IMPLICIT INTEGER(A-Z)

          IF (SW) 100,200,200

C         SW=-1,    P1 AND P2 ARE PICTURE POINTERS.  THIS IS AN
C         INITIALZATION CALL.
100       PTR1=P1
          PTR2=P2
          RETURN

C         SW=0    P1 AND P2 ARE LIST POINTERS, P2 ON RIGHT.
C         DELETE POINTS IN PICTURE WHICH ARE ON LIST, INCLUDING
C         END POINTS.
200       CALL INSTOR(PTR1)
          J=P1
250       X=LXCOR(J) .
          Y=LYCOR(J)
          CALL STORE(X,Y,0)
          IF (J.EQ.P2) GOTO 300
          J=LRSIB(J)
          GO TO 250
300       CALL INSTOR(PTR2)
          RETURN

          END
```

Digitized by Google

```
            SUBROUTINE UHEF(WHERE,SX,SY,TX,TY,V,FAILSW)
            IMPLICIT INTEGER (A-Z)
C           UPPER HEAD EDGE FOLLOWER.
C           FOLLOWS A CURVING EDGE FROM THE TOP OF THE SIDE OF
C           THE HEAD (SX,SY) TO THE TOP OF THE HEAD (TX,TY).
C           WHERE = 1  :  UPPER LEFT CURVE OF HEAD.
C           WHERE =-1  :  UPPER RIGHT CURVE OF HEAD.
C           [BRACKETED COMMENTS APPLY TO WHERE=-1.]
            COMMON /CON/ A(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
            1        TVPNTR,TVSIZ,NPERWD,ILEN,B(8),ITEXT(10)

            CALL NFATAL
            FX=SX
            FY=SY
            FD=LOAD(FX,FY)
      100   IF (WHERE.LT.0) GO TO 120
            OKDIRS=1+2+4
            OKSIDE=1+4
            OKTOPS=2+4
            GO TO 3000
      120   OKDIRS=1+2+8
            OKSIDE=1+8
            OKTOPS=2+8

C           FX = LAST X FOUND
C           FY = LAST Y FOUND
C           FD = DIRECTION AT FX,FY

C           IF FY=TY, TOP HAS BEEN REACHED.
     3000   IF (FY.LE.TY) GO TO 4000

C           DOES FD INCLUDE VERTICAL EDGE?
            IF ((FD.AND.1).EQ.0) GO TO 3500

C           YES.
            DO 3100 KOUNT=1,5
            IF (FY-KOUNT.LT.TY) GO TO 3200
            IF (KOUNT.GT.1) GO TO 3070
            ILIM=FX+WHERE
            GO TO 3075
     3070   ILIM=FX+2*WHERE
     3075   DO 3100 I=FX-WHERE,ILIM,WHERE
            IF (I.LT.LSIDE.OR.I.GT.RSIDE) GO TO 3100
            P=LOAD(I,FY-KOUNT)
C           | OR / ?  [ | OR \ ? ]
            IF ((P.AND.OKSIDE).NE.0) GO TO 3800
     3100   CONTINUE
            CALL FAIL('3100')
C           CANNOT FIND EDGE ABOVE FX,FY.  TRY FOR EDGE AS IF FD WAS A SLANT.
            GO TO 3500
     3200   CALL FAIL('3020')
            GO TO 5300
```

Digitized by Google

```
C         NO.
C         THEREFORE FD HAS / OR - EDGE.  [ \ OR - EDGE ]
 3500     I3500=0
          DO 3600 KOUNT=1,5
          IF (FY-KOUNT-TY) 3700,3520,3520
C         SEARCH FROM VERTICALLY ABOVE FX,FY INVARD TO
C           LINE BETWEEN FX,FY AND TX,TY.
C         RATIOS:         NRSTEPS  /  COUNT  =  TX-FX  /  FY-TY
 3520     NNUM=(TX-FX)*WHERE*KOUNT
          NDEN=FY-TY
          NRSTEP=NNUM/NDEN
C         ROUND UP.
          IF (MOD(NNUM,NDEN).NE.0) NRSTEP=NRSTEP+1
          T3530=FX+NRSTEP*WHERE
 3530     DO 3600 I=FX,T3530,WHERE
          IF (I.LT.LSIDE.OR.I.GT.RSIDE) GO TO 3600
          P=LOAD(I,FY-KOUNT)
C         | OR / OR -?  (I.e. ANYTHING BUT \ ONLY.)
C            [ \ ]                       [ / ]
          IF ((P.AND.OKDIRS).NE.0) GO TO 3800
 3600     CONTINUE
          CALL FAIL('3600')
          GO TO 5300
 3700     CALL FAIL('3520')
          GO TO 5300

C         FOUND
 3800     FX=I
          FY=FY-KOUNT
          FD=P
          IF (FX.EQ.TX.AND.FY.EQ.TY) GO TO 5000
          CALL STORE(FX,FY,P)
          V=LPINS(WHERE,V,FX,FY)
          GO TO 3000

C         TOP HAS BEEN REACHED.  INCLUDE ANY - OR / [ - OR \ ]
C         OVER TO PREVIOUS TOP LINE.
 4000     DO 4100 I=FX+WHERE,FX+3*WHERE,WHERE
          IF ((TX-I)*WHERE.LE.0) GO TO 5000
          P=LOAD(I,TY)
          IF ((P.AND.OKTOPS).EQ.0) GO TO 4100
          CALL STORE(I,TY,P)
          V=LPINS(WHERE,V,FX,FY)
          FX=I
          GO TO 4000
 4100     CONTINUE
C         A GAP OF MORE THAN TWO POINTS HAS BEEN FOUND
C         IN THE TOP OF THE HEAD.  FAILURE.
          CALL FAIL('4120')
          GO TO 5300
```

Digitized by Google

Digitized by Google

```
5ØØØ     FAILSW=Ø
         GO TO 56ØØ
53ØØ     FAILSW=1
56ØØ     CALL YFATAL
         RETURN
         END
```

Digitized by Google

Digitized by Google

```
        SUBROUTINE LHEF(WHERE,SX,SY,BYPAR,V,FAILSW)
        IMPLICIT INTEGER (A-Z)
C       Lower Head Edge Follower.
C       Follows a curving edge from the bottom of the
C       side of the head (SX,SY) to the bottom of the
C       picture (BY).
C       WHERE =  1  :   lower left curve of head.
C       WHERE = -1  :   lower right curve of head.
C                 x           x
C                 x           x           Type of
C                  x         x            curves
C                   x       x             looked
C                   x       x             for.
C                   x       x
C                  x         x
C                 x           x
C                x   .         x
C       All comments apply to WHERE=1.
        COMMON /LHEFCM/ BY,FX,FY,FD,FAILCM,VCM,L OR R

        CALL NFATAL
        FAILCM=0
        VCM=V
        L OR R = -WHERE
        BY=BYPAR
        FX=SX
        FY=SY
        FD=LOAD(FX,FY)
        IF (WHERE) 120,7000,110
110     DOWNIN=1+8
        SLNOUT=4  +2
        DWNOUT=1+4  +2
        GO TO 5900
120     DOWNIN=1+4
        SLNOUT=8  +2
        DWNOUT=1+8  +2

C ##########################################
C       Stage 1, going in.
5900    SAVEX=FX
C       If FY=BY, bottom has been reached.  In Stage 1,
C       this is an error.
6000    IF (FY.GE.BY) GO TO 6150
C       Does FD include vertical edge?
        IF ((FD.AND.1).EQ.0) GO TO 6110
C       Yes.
        CALL SEARCH(FX-WHERE,FX+2*WHERE,WHERE,DOWNIN,'6105',0)
        GO TO 6115
C       No, FD is \ .
6110    TEMPFX=FX
        CALL SEARCH(TEMPFX,FX+2*WHERE,WHERE,DOWNIN,'6110',0)
C       OK, new edge.
```

Digitized by Google

Digitized by Google

```
C        If new is close to vertical starting line continue
C        with Stage 1.
6115     IF (FAILCM,EQ,1) GO TO 6990
         IF (FY,GE,BY) GO TO 6150
         IF ((FX-SAVEX)*WHERE.LT.1) GO TO 6000
C        NEXT STMT NICE FOR KOCT2.  IS IT GENERALLY USEFUL TO
C                  OMIT STAGE 2?
C???     GO TO 6300
C        8 JUNE 70  -  NO FOR LDEH2
         GO TO 6200

6150     CALL FAIL('6150')
         GO TO 6990

C ##########################################
C        Stage 2, going down.
6200     IF (FY.GE.BY) GO TO 7000
         IF ((FD.AND.SLNOUT).EQ.0) GO TO 6210
         STAGE=2
         GO TO 6303
6210     CALL SEARCH(FX-WHERE,FX+WHERE,WHERE,1+4+8,'6205',-1)
C        OK, new edge.
C        If not back under vertical edge, continue with Stage 2.
         IF (FAILCM.EQ.1) GO TO 6990
         IF ((FX-SAVEX)*WHERE.GT.0) GO TO 6200

C ##########################################
C        Stage 3, going out.
6300     IF (FY.GE.BY) GO TO 7000
         STAGE=3
C        Does FD include a slant edge?
         IF ((FD.AND.SLNOUT).EQ.0) GO TO 6310
C        Yes.
C                  IF POINT BELOW IS NOT A 2,
C                  FIRST TRY TO FOLLOW OUTWARD.
6303               TFD=LOAD(FX,FY+1)
                   IF ((TFD.AND.2).NE.0) GO TO 6307
                   TFD=LOAD(FX-WHERE,FY)
                   IF ((TFD.AND.DWNOUT).EQ.0) GO TO 6307
                   FX=FX-WHERE
                   FD=TFD
                   CALL STORE(FX,FY,FD)
                   VCM=LPINS(L OR R,VCM,FX,FY)
                   IF (STAGE.EQ.2) GO TO 6200
                   GO TO 6300
6307     TEMPFX=FX
         CALL SEARCH(FX-2*WHERE,TEMPFX,WHERE,DWNOUT,'6305',-1)
         GO TO 6315
C        No.  FD is | .
6310     CALL SEARCH(FX-2*WHERE,FX+WHERE,WHERE,DWNOUT,'6310',-1)
6315     IF (FAILCM.EQ.1) GO TO 6990
         IF (STAGE.EQ.2) GO TO 6200
```

Digitized by Google

Digitized by Google

```
        GO TO 6300

 6990   FAILSW=1
        GO TO 7010
 7000   FAILSW=0
 7010   CALL YFATAL
        V=VCM
        RETURN
        END
```

Digitized by Google

```
17-JUL-70     0833            BOR34      1,MDK                PAGE 8-1

                SUBROUTINE SEARCH(FROMP,TO,STEP,DIRS,FLABEL,WIDEN)
                IMPLICIT INTEGER (A-Z)
                COMMON /LHEFCM/ BY,FX,FY,FD,FAILCM,VCM,L OR R
                FROM=FROMP
                DO 120 KOUNT=1,5
                FY=FY+1
                IF (FY.GT.BY) RETURN
                DO 100 FX=FROM,TO,STEP
                FD=LOAD(FX,FY)
                IF ((FD.AND.DIRS).NE.0) GO TO 200
        100     CONTINUE
                IF (WIDEN.NE.0) FROM=FROM-STEP
        120     CONTINUE
                CALL FAIL(FLABEL)
                FAILCM=1
                RETURN

        200     CALL STORE(FX,FY,FD)
                VCM=LPINS(L OR R,VCM,FX,FY)
                END
```

Digitized by Google

Digitized by Google

```
              SUBROUTINE DSLINS(PTR,X,Y,POG)
              IMPLICIT INTEGER (A-Z)
              COMMON /CON/ A(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
              1       TVPNTR,TVSIZ,NPERWD,ILEN,B(8),ITEXT(1Ø)

              CALL FILCON(PTR,A(1))
              CALL INLOAD(PTR)

              DO 5ØØØ J=FLINE,LLINE
              DO 5ØØØ I=LSIDE,RSIDE
              P=LOAD(I,J)
              IF (P.EQ.Ø) GO TO 5ØØØ

C             |   1
              IF (MOD(P,2).NE.1) GO TO 32ØØ
              IF (J.EQ.FLINE) GO TO 311Ø
              IF (MOD(LOAD(I,J-1),2).EQ.1) GO TO 32ØØ
     311Ø     IF (J.EQ.LLINE) GO TO 3125
              DO 312Ø JJ=J+1,LLINE
              IF (MOD(LOAD(I,JJ),2).NE.1) GO TO 313Ø
     312Ø     CONTINUE
     3125     JJ=LLINE+1
     313Ø     JJ=JJ-1
              CALL ALINE(X+6*(I-LSIDE),Y+3-6*(J-FLINE),
              1       X+6*(I-LSIDE),Y-3-6*(JJ-FLINE) )

C             -   2
     32ØØ     IF (MOD(P/2,2).NE.1) GO TO 33ØØ
              IF (I.EQ.LSIDE) GO TO 321Ø
              IF (MOD(LOAD(I-1,J)/2,2).EQ.1) GO TO 33ØØ
     321Ø     IF (I.EQ.RSIDE) GO TO 3225
              DO 322Ø II=I+1,RSIDE
              IF (MOD(LOAD(II,J)/2,2).NE.1) GO TO 323Ø
     322Ø     CONTINUE
     3225     II=RSIDE+1
     323Ø     II=II-1
              CALL ALINE(X-3+6*(I-LSIDE),Y-6*(J-FLINE),
              1       X+3+6*(II-LSIDE),Y-6*(J-FLINE) )

C             /   4
     33ØØ     IF (MOD(P/4,2).NE.1) GO TO 34ØØ
              IF (I.EQ.RSIDE.OR.J.EQ.FLINE) GO TO 331Ø
              IF (MOD(LOAD(I+1,J-1)/4,2).EQ.1) GO TO 34ØØ
     331Ø     KK=MINØ(I-LSIDE,LLINE-J)
              IF (KK.EQ.Ø) GO TO 3325
              DO 332Ø K=1,KK
              IF (MOD(LOAD(I-K,J+K)/4,2).NE.1) GO TO 333Ø
     332Ø     CONTINUE
     3325     K=KK+1
     333Ø     K=K-1
              CALL ALINE(X+3+6*(I-LSIDE),Y+3-6*(J-FLINE),
              1       X-3+6*(I-K-LSIDE),Y-3-6*(J+K-FLINE))
```

Digitized by Google

Digitized by Google

```

C       \   8
3400    IF (MOD(P/8,2).NE.1) GO TO 5ØØØ
        IF (I.EQ.LSIDE.OR.J.EQ.FLINE) GO TO 341Ø
        IF (MOD(LOAD(I-1,J-1)/8,2).EQ.1) GO TO 5ØØØ
341Ø    KK=MINØ(RSIDE-I,LLINE-J)
        IF (KK.EQ.Ø) GO TO 3425
        DO 342Ø K=1,KK
        IF (MOD(LOAD(I+K,J+K)/8,2).NE.1) GO TO 343Ø
342Ø    CONTINUE
3425    K=KK+1
343Ø    K=K-1
        CALL ALINE(X-3+6*(I-LSIDE),Y+3-6*(J-FLINE),
        1       X+3+6*(I+K-LSIDE),Y-3-6*(J+K-FLINE))

5ØØØ    CONTINUE
55ØØ    CALL DPYOUT(POG)

        END
```

Digitized by Google

Digitized by Google

```
            FUNCTION JRIGHE(BIGPT,HEDPOG)
            INTEGER BIGPT,HEDPOG
            COMMON /CON/ A(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
            1        TVPNTR,TVSIZ,NPERWD,ILEN,B(8),ITEXT(10)
            INTEGER A,BITS,FLINE,RSIDE,TVPNTR,TVSIZ,B
            COMMON /LSENTS/ VBL,VLSB,VLST,VTL,VTR,VRST,VRSB,VBR
            INTEGER VBL,VLSB,VLST,VTL,VTR,VRST,VRSB,VBR
            COMMON /FOLCOM/ IP1,JP1,IP2,JP2,IP3,JP3,IQ,JQ,MPOG,
            1        WSTART,WFIN
            INTEGER WSTART,WFIN
            EXTERNAL JEDGE5

C           MAPPING FROM REDUCED POINTS TO BIG POINTS.
C           **** BEWARE **** THESE ASF'S ARE IMPLICITLY IN IVFIX AT
C                                 STATEMENTS 40 AND 140.
            IMAP(I)=8*I+LSIDE
            JMAP(J)=8*J+FLINE
C           OLD MAPPING.
C           IMAP(I)=8*I+LSIDE-4
C           JMAP(J)=8*I+FLINE-4

C           ALLOCATE AND ZERO NEW BUFFER.
            NEWPTR=INITOP(BIGPT,1,'BIG EDGES')
            CALL ZROBUF(NEWPTR)

C           INITIALIZE DPY STUFF
            MPOG=HEDPOG
            CALL FOLLOW(1,NUTN)

C           START PROCESSING
            LIN=VBL
            IL1=LXCOR(LIN)
            JL1=LYCOR(LIN)
            IP1=IMAP(IL1)
            JP1=JMAP(JL1)
            IQ=IP1
            JQ=JP1
            LIN=LRSIB(LIN)
            IL2=LXCOR(LIN)
            JL2=LYCOR(LIN)
            RTEMP=IL2-IL1
            IF (RTEMP) 120,110,120
      110   SLOP12=999.
            GO TO 130
      120   SLOP12=(JL2-JL1)/RTEMP
      130   LIN=LRSIB(LIN)
            IF (LIN.EQ.0) GO TO 180
            IL3=LXCOR(LIN)
            JL3=LYCOR(LIN)
            RTEMP=IL3-IL2
            IF (RTEMP) 150,140,150
      140   SLOP23=999.
```

Digitized by Google

```
              GO TO 160
      150     SLOP23=(JL3-JL2)/RTEMP
      160     IF (SLOP12-SLOP23) 200,170,200
      C       SLOPES ARE EQUAL
      170     IL2=IL3
              JL2=JL3
              GO TO 130

      C       END OF LIST. SET UP DUMMY P3.
      180     IP2=IMAP(IL2)
              JP2=JMAP(JL2)
              IP3=2*IP2-IP1
              JP3=2*JP2-JP1
              GO TO 250

      C       SLOPES NOT EQUAL. P1,P2, AND P3 FOUND.
      200     IP2=IMAP(IL2)
              JP2=JMAP(JL2)
              IP3=IMAP(IL3)
              JP3=JMAP(JL3)

      C       NOW PROCESS SECTION FROM P1 TO P2.
      250     CALL FOLLOW(0,JEDGE5)

      C       SECTION PROCESSED.  ON TO NEXT.  DUM-DI-DUMP-DUMP.
              IF (LIN.EQ.0) GO TO 400
              SLOP12=SLOP23
              IP1=IP2
              JP1=JP2
              GO TO 170

      C       ALL FINISHED.
      400     CALL FOLLOW(-1,NUTN)
              JBIGHE=NEWPTR

      C       PUT NEW LIST ENTRIES IN.
              VBL=WSTART
              VLSB=IVFIX(VBL,VLSB,1)
              VLST=IVFIX(VLSB,VLST,1)
              VTL=IVFIX(VLST,VTL,-2)
              VTR=IVFIX(VTL,VTR,-2)
              VRST=IVFIX(VTR,VRST,-1)
              VRSB=IVFIX(VRST,VRSB,-1)
              VBR=WFIN

              RETURN
              END
```

Digitized by Google

```
17-JUL-70     0840           HEAD26     1,MDK                    PAGE 2-1

              FUNCTION IVFIX(V1,V2,ISIGN)
              IMPLICIT INTEGER (A-Z)
        C     |ISIGN| = 1 : TEST Y
        C     |ISIGN| = 2 : TEST X
              COMMON /CON/ A(3),BITS,IWID,LINLEN,FLINE,LLINE,LSIDE,RSIDE,
              1        TVPNTR,TVSIZ,NPERWD,ILEN,B(8),ITEXT(10)
              INTEGER A,BITS,FLINE,RSIDE,TVPNTR,TVSIZ,B
              IF (IABS(ISIGN).EQ.1) GO TO 140
        C     **** BEWARE **** NEXT STMT IS ASF IMAP FROM JBIGHE.
        40    I=LSIDE+8*LXCOR(V2)
              J=V1
        100   J=LRSIB(J)
              ITEST=LXCOR(J)
              IF (ISIGN*(I-ITEST)) 100,200,200
        C     **** BEWARE **** NEXT STMT IS ASF JMAP FROM JBIGHE.
        140   I=FLINE+8*LYCOR(V2)
              J=V1
        160   J=LRSIB(J)
              ITEST=LYCOR(J)
              IF (ISIGN*(I-ITEST)) 160,200,200
        200   IVFIX=J
              END
```

Digitized by Google

```
              FUNCTION MSD(SLOPE)
      C       MAIN SLANT DIRECTION.
              IF (SLOPE) 1140,1110,1120
      1110    MDIRS=2
      C       ORIGINALLY 14
              GO TO 1200
      1120    IF (SLOPE-1.) 1135,1130,1133
      1130    MDIRS=4
      C       ORIGINALLY 7
              GO TO 1200
      1133    MDIRS=5
              GO TO 1200
      1135    MDIRS=6
              GO TO 1200
      1140    IF (SLOPE+1.) 1155,1150,1153
      1150    MDIRS=8
      C       ORIGINALLY 11
              GO TO 1200
      1153    MDIRS=10
              GO TO 1200
      1155    MDIRS=9
              GO TO 1200
      1200    MSD=MDIRS
              END
```

Digitized by Google

```
        SUBROUTINE FOLLOW(ISW,JFUNC)
        COMMON /FOLCOM/ IP1,JP1,IP2,JP2,IP3,JP3,IQ,JQ,MPOG,
        1       WSTART,WFIN
        INTEGER WSTART,WFIN
C       DPY STUFF -----------------------
        INTEGER G1,G2
        REAL GCONST
        DATA GCONST/6.0/
        INTEGER TPOG,UPOG,SPOG,SFLAG
        DATA LSBUF/40/
        LOGICAL NODPY,ANYM,ANYN,ANYS
        DATA NODPY/.TRUE./
C         PIECE OF GLASS            CONTENTS
C             TPOG          PLAN LINES:  P1-P2, P2-P3, BISECTOR
C             UPOG          POINTS IN PERPENDICULAR SEARCH.
C             SPOG          D FOR DELETED POINTS.
C             MPOG          FINAL OUTLINE.
C             NPOG          TEMPORARY EXTENSION OF OUTLINE.
C       --------------------------------
        LOGICAL HORIZ,WIDE,NXTWID,ANYLIS
        DIMENSION II(-4/11),JJ(-4/11),ISTOP(-4/11),JSTOP(-4/11)
        DATA NB1/-4/,NB2/11/
        COMMON /OUTFGS/ IFDTTY,IFDLPT,IFDDPY

        IF (ISW) 400,1000,100

C       INITIALIZE
100     IF (IFDDPY.EQ.0) NODPY=.TRUE.
        ANYM=.FALSE.
        SFLAG=0
        ANYS=.FALSE.
        WIDE=.TRUE.
        ANYLIS=.FALSE.
        IF (NODPY) RETURN
        TPOG=IADPST(25)
        UPOG=IADPST(50)
        NPOG=IADPST(200)
        RETURN

C       ALL FINISHED.
400     WFIN=JLISPT
        IF(IFDDPY.EQ.0) RETURN
        IF (NODPY) RETURN
        CALL OUTD('+DIS OK?&')
        NUTN=YORN(-1)
        CALL HYDPOG(TPOG)
        CALL HYDPOG(UPOG)
        CALL HYDPOG(NPOG)
        IF (.NOT.ANYS) GO TO 430
        CALL HYDPOG(SPOG)
        CALL IGIVEX(SPOG)
430     CALL IGIVEX(NPOG)
```

Digitized by Google

Digitized by Google

```
            CALL IGIVEX(UPOG)
            CALL IGIVEX(TPOG)
            RETURN

      C*************************************************************
      C*********** PROCESS SECTION FROM P1 TO P2 *******************
      C     INPUT TO THIS PART OF CODE IS:
      C             COORDINATES OF P1,P2,P3, AND Q WHICH ARE
      C                     IP1,JP1
      C                     IP2,JP2
      C                     IP3,JP3
      C                     IQ,JQ

      C     DRAW LINES CONNECTING THE 3 POINTS
      1000  IF (NODPY) GO TO 1010
            CALL CLRPOG(TPOG)
            CALL ALINE(G1(IP1),G2(JP1),G1(IP2),G2(JP2))
            CALL AVECT(G1(IP3),G2(JP3))
            CALL DPYOUT(TPOG)

      C     DETERMINE LINE L WHICH BISECTS ANGLE BETWEEN P1P2 AND P2P3.
      1010  X3=IP3-IP2
            Y3=JP3-JP2
            DIST3=SQRT(X3**2+Y3**2)
            X1=IP1-IP2
            Y1=JP1-JP2
            DIST1=SQRT(X1**2+Y1**2)
      C     R IS POINT ON LINE L.  CONSIDERING P2R, P2P3, AND P2P1 AS
      C     VECTORS VR, V3, AND V1;  R IS DEFINED BY:
      C             VR=(V3/ABS(V3)) + (V1/ABS(V1))
      C     [RELATIVE COORDINATES FOR POINT R ARE CALCULATED.  TRUE COORDINATES
      C      WOULD BE IP2+XR AND JP2+YR.]
            XR=(X3/DIST3)+(X1/DIST1)
            YR=(Y3/DIST3)+(Y1/DIST1)

            IF (NODPY) GO TO 1100
      C     DISPLAY LINE IN DIRECTION OF R OF LENGTH 8.
            DISTR=SQRT(XR**2+YR**2)
            IR8=IP2+IFIX(8.*XR/DISTR+0.5)
            JR8=JP2+IFIX(8.*YR/DISTR+0.5)
            CALL ALINE(G1(IP2),G2(JP2),G1(IR8),G2(JR8))
            CALL DPYOUT(TPOG)

      C     DETERMINE MAIN SEARCH DIRECTIONS FOR P1P2.
      C     1=|     2=-     4=/     8=\
      1100  IF (X1.EQ.0) GO TO 1175
      C     SLOPE IS IN NORMAL RIGHT HANDED COORDINATE SYSTEM (HENCE MINUS).
            SLOPE=-Y1/X1
            MDIRS=MSD(SLOPE)
            GO TO 1180
      1175  MDIRS=1
      C     ORIGINALLY 13
```

Digitized by Google

```

C             DETERMINE SECONDARY SEARCH DIRECTIONS FROM P2P3.
      1180    IF (X3.EQ.0) GO TO 1190
              EPOLS=-Y3/X3
              MDIRS=MDIRS .OR. MSD(EPOLS)
              GO TO 1200
      1190    MDIRS=MDIRS .OR. 1

C             If the search angle changes greatly (more than 45 degrees)
C             at P2, set flag to force a wide search at P2.
C                       cos α = (-V1.V3) / (|V1|*|V3|)
      1200    COSALF=(-X1*X3-Y1*Y3)/(DIST1*DIST3)
              NXTWID=.FALSE.
              IF (COSALF.LT.0.707) NXTWID=.TRUE.

C             IS P1P2 MOSTLY HORIZONTAL?
              IF (ABS(X1).LT.ABS(Y1)) GO TO 1400
C             YES
              HORIZ=.TRUE.
C             IN GOING FROM P1 TO P2, THE X INCREMENT WILL BE +1 OR -1.
              IINC=ISIGN(1,IP2-IP1)
C             THE Y INCREMENT.
              YINC=-Y1/ABS(X1)
C             THE INCREMENTS FOR SCANNING ACROSS TO LOOK FOR AN EDGE.
              XSCNIN=-YINC
              JSCNIN=IINC
C             FILL IN OFFSET TABLE FOR SCANNING
              DO 1310 K=NB1,NB2
              II(K)=K*XSCNIN+0.5
      1310    JJ(K)=K*JSCNIN
C             FILL IN STOPPING TABLE
              QSQRD=XSCNIN**2+JSCNIN**2
              QDOTV=XSCNIN*XR+JSCNIN*YR
              XHZERO=XR*QSQRD/QDOTV
              DO 1330 K=NB1,NB2
      1330    ISTOP(K)=IP2+K*XHZERO+0.5
C             DETERMINE STOPPING SIGN.
              IF (IP1-ISTOP(0)) 1350,1350,1360
      1350    ISTPSG=1
              GO TO 1370
      1360    ISTPSG=-1
C             SET Q AND P.
C             FIND POINT Q RELATIVE TO NEW SEARCH COORDINATES.
C             START SEARCH THERE.
      1370    IP=IP1
              YP1=JP1
              IF (SLOPE.NE.0) GO TO 1380
              IQDIST=(IQ-IP1)*IINC
              GO TO 1383
C             Let point QQ be the intersection of line P1P2 and
C             a line perpendicular to P1P2 which passes through Q.
C             Calculate the X coordinate of QQ.
```

Digitized by Google

Digitized by Google

```
1380     XQQ=(YP1-JQ+SLOPE*IP1+(1./SLOPE)*IQ)
         1        / (SLOPE+(1./SLOPE))
         IQQ=XQQ+0.5
         IQDIST=(IQQ-IP1)*IINC
1383     IF (IQDIST) 1385,1390,1390
1385     N=0
         KQ=0
         WIDE=.TRUE.
         GO TO 13960
1390     N=IQDIST
         IP=IP1+N*IINC
         JP=YP1+N*YINC+0.5
         KQ=(JQ-JP)*JSCNIN
13960    IF (IQ.NE.I22SAV.OR.JQ.NE.J22SAV) GO TO 13963
         IF (IQDIST.LT.0) GO TO 13962
         LASTKQ=KQ
         GO TO 1500
13962    IF (SLOPE.EQ.0) GO TO 13964
         ISDIST=IQDIST
         GO TO 13969
13963    IF (SLOPE.NE.0) GO TO 13966
13964    LASTKQ=(J22SAV-JP1)*JSCNIN
         GO TO 1500
13966    XSS=(YP1-J22SAV+SLOPE*IP1+(1./SLOPE)*I22SAV)
         1        / (SLOPE+(1./SLOPE))
         ISS=XSS+0.5
         ISDIST=(ISS-IP1)*IINC
13969    JPS=YP1+ISDIST*YINC+0.5
         LASTKQ=(J22SAV-JPS)*JSCNIN
         GO TO 1500

C        MOSTLY VERTICAL.
1400     HORIZ=.FALSE.
         XINC=-X1/ABS(Y1)
         JINC=ISIGN(1,JP2-JP1)
         ISCNIN=-JINC
         YSCNIN=XINC
         DO 1410 K=NB1,NB2
         II(K)=K*ISCNIN
1410     JJ(K)=K*YSCNIN+0.5
         QSQRD=ISCNIN**2+YSCNIN**2
         QDOTV=ISCNIN*XR+YSCNIN*YR
         YHZERO=YR*QSQRD/QDOTV
         DO 1430 K=NB1,NB2
1430     JSTOP(K)=JP2+K*YHZERO+0.5
         IF (JP1-JSTOP(0)) 1450,1450,1460
1450     JSTPSG=1
         GO TO 1470
1460     JSTPSG=-1
1470     XP1=IP1
         JP=JP1
         IF (X1.NE.0) GO TO 1480
```

Digitized by Google

```
                JQDIST=(JQ-JP1)*JINC
                GO TO 1483
        1480    YQQ=(SLOPE*JQ+(1./SLOPE)*JP1-IQ+XP1)
                1       / (SLOPE+(1./SLOPE))
                JQQ=YQQ+0.5
                JQDIST=(JQQ-JP1)*JINC
        1483    IF (JQDIST) 1485,1490,1490
        1485    N=0
                KQ=0
                WIDE=.TRUE.
                GO TO 14960
        1490    N=JQDIST
                IP=XP1+N*XINC+0.5
                JP=JP+N*JINC
                KQ=(IQ-IP)*ISCNIN
        14960   IF (IQ.NE.I22SAV.OR.JQ.NE.J22SAV) GO TO 14963
                IF (JQDIST.LT.0) GO TO 14962
                LASTKQ=KQ
                CO TO 1500
        14962   IF (X1.EQ.0) GO TO 14964
                JSDIST=JQDIST
                GO TO 14969
        14963   IF (X1.NE.0) GO TO 14966
        14964   LASTKQ=(I22SAV-IP1)*ISCNIN
                GO TO 1500
        14966   YSS=(SLOPE*J22SAV+(1./SLOPE)*JP1-I22SAV+XP1)
                1       / (SLOPE+(1./SLOPE))
                JSS=YSS+0.5
                JSDIST=(JSS-JP1)*JINC
        14969   IPS=XP1+JSDIST*XINC+0.5
                LASTKQ=(I22SAV-IPS)*ISCNIN
                GO TO 1500

        C       INITIALIZE SEARCH.
        1500    IF (KQ.GE.NB1.AND.KQ.LE.NB2) GO TO 1510
                KQ=NB1
                IF (KQ.GT.NB2) KQ=NB2
                WIDE=TRUE
        1510    IF (.NOT.WIDE) GO TO 1525
                KS1=NB1
                KS2=NB2
                IBACK2=100
                IBACK1=100
        1525    I=IQ
                J=JQ

        C************************************
        C***********  SEARCH  ****************

        C       UNTIL TRACE OF Q CROSSES L
        1600    IF (HORIZ) GO TO 1610
                IF (JSTPSG*(J-JSTOP(KQ))) 1675,1675,3000
```

Digitized by Google

```
1610    IF (ISTPSG*(I-ISTOP(KQ))) 1650,1650,3000

C       SEARCH
1650    N=N+1
        IP=IP+IINC
        JP=YP1+N*YINC+0,5
        GO TO 1680
1675    N=N+1
        IP=XP1+N*XINC+0,5
        JP=JP+JINC

1680    IF (NODPY) GO TO 1690
        CALL CLRPOG(UPOG)

C       SEARCH ALONG S FROM S1 TO S2 FOR EDGE,
1690    DO 1700 K=KS1,KS2
        I=IP+II(K).
        J=JP+JJ(K)
        IF (NODPY) GO TO 1695
        CALL APOINT(G1(I),G2(J))
        CALL DPYOUT(UPOG)
1695    ISEDGE=JFUNC(I,J,MDIRS)
        IF (ISEDGE) 1900,1700,1900
1700    CONTINUE
C       NO EDGE FOUND
        IBACK1=100
        IBACK2=100
        IF (WIDE) GO TO 1840
C       SEARCH ALONG S FROM B1 TO S1-1 AND FROM S2+1 TO B2 FOR EDGE,
        DO 1820 K=NB1,NB2
        IF (K,GE,KS1.AND.K,LE,KS2) GO TO 1800
        I=IP+II(K)
        J=JP+JJ(K)
        IF (NODPY) GO TO 1795
        CALL APOINT(G1(I),G2(J))
        CALL DPYOUT(UPOG)
1795    ISEDGE=JFUNC(I,J,MDIRS)
        IF (ISEDGE) 1810,1800,1810
1800    CONTINUE
1810    WIDE=,TRUE,
        KS1=NB1
        KS2=NB2
        IF (ISEDGE) 1820,1840,1820
C       EDGE FOUND DURING WIDE SEARCH,
1820    KQ=K
        GO TO 2200
C       NO EDGE FOUND,
1840    KQ=4*KQ/5
        I=IP+II(KQ)
        J=JP+JJ(KQ)
        GO TO 1600
C       EDGE FOUND DURING NARROW SEARCH,
```

Digitized by Google

```
1900     KQ=K
         IF (KQ.LT.NB1+2) GO TO 2000
         IBACK=LOAD(IP+II(KQ-2),JP+JJ(KQ-2))
         IF (IBACK2.EQ.100) GO TO 1925
         IF (IABS(IBACK-IBACK1).GT.1 .OR.
         1       IABS(IBACK-IBACK2).GT.1) GO TO 2000
C        OK FOR NARROW SEARCH.
         GO TO 2000
         WIDE=.FALSE.
         IBACK2=IBACK1
         IBACK1=IBACK
         KS1=MAX0(KQ-4,NB1)
         KS2=MIN0(KQ+4,NB2)
         GO TO 2200
1925     IBACK2=IBACK1
         IBACK1=IBACK
         GO TO 2200
C        SET FOR WIDE SEARCH.
2000     WIDE=.TRUE.
         IBACK2=100
         IBACK1=100
         KS1=NB1
         KS2=NB2
         GO TO 2200

C        RUN SIMPLE FILTER ON POINTS FOUND.
2200     IF (SFLAG) 2250,2210,2220
C        LASTKQ IS UNDEFINED.  SET IT UP.
2210     ASSIGN 1600 TO LABSP
         SFLAG=1
         LASTKQ=KQ
C        NEXT TWO VALUES USED ONLY IF DIRECTION IS CHANGED BETWEEN POINTS.
         I22SAV=I
         J22SAV=J
         GO TO 2400
C        NORMAL SITUATION.  TEST KQ.
2220     IF (IABS(KQ-LASTKQ).GT.3) GO TO 2230
C        NO BIG CHANGE IN KQ.
C        ASSERT:  SFLAG=1,  LABSP=1600.
         LASTKQ=KQ
C        NEXT TWO VALUES USED ONLY IF DIRECTION IS CHANGED BETWEEN POINTS.
         I22SAV=I
         J22SAV=J
         GO TO 2400
C        A BIG CHANGE IN KQ.  HOLD I,J FOR A TEST ON THE NEXT POINT.
2230     I22SAV=I
         J22SAV=J
2235     KQ22SV=KQ-LASTKQ
C        ASSERT:  LABSP=1600.
         SFLAG=-1
         LASTKQ=KQ
         GO TO 1600
```

Digitized by Google

```
C       Previous I,J is being held.  Test new I,J to see what to do.
225Ø    IF (IABS(KQ-LASTKQ).GT.3) GO TO 226Ø
C       No big change in new KQ.
C       Store previous I,J.  Then new I,J.
        ASSIGN 2255 TO LABSP
        GO TO 235Ø
C       Now for new I,J.
2255    I=I22SAV
        J=J22SAV
        ASSIGN 16ØØ TO LABSP
        SFLAG=1
        LASTKQ=KQ
        GO TO 24ØØ
C       A big change in new KQ.  Compare signs of this change and
C       the previous change (which was also big).
226Ø    IF (KQ22SV*(KQ-LASTKQ)) 228Ø,2265,2265
C       Signs are the same.  Therefore previous I,J is okay
C       and should be stored.  New I,J should be held as at
C       223Ø above.
2265    ASSIGN 227Ø TO LABSP
        GO TO 235Ø
227Ø    ASSIGN 16ØØ TO LABSP
C       ASSERT: I22SAV,J22SAV = NEW COORDINATES.
        I=I22SAV
        J=J22SAV
        GO TO 2235
C       SIGNS OF TWO CONSECUTIVE BIG CHANGES ARE DIFFERENT.
C       DELETE PREVIOUS POINT.  DISPLAY A "D" AT PREVIOUS POINT
C       IF DESIRED.
228Ø    IF (NODPY) GO TO 229Ø
C       DISPLAY "D" AT I22SAV,J22SAV.
        IF (ANYS) GO TO 2285
        ANYS=.TRUE.
        SPOG=IADPST(LSBUF)
        GO TO 2288
2285    CALL SETPOG(SPOG)
2288    CALL DPYTXT(G1(I22SAV),G2(J22SAV),'D',1)
        CALL DPYOUT(SPOG)
C       NOW STORE NEW I,J.
C       ASSERT:  LABSP=16ØØ.
229Ø    SFLAG=1
        LASTKQ=KQ
C       NEXT TWO VALUES USED ONLY IF DIRECTION IS CHANGED BETWEEN POINTS.
        I22SAV=I
        J22SAV=J
        GO TO 24ØØ
C       STORE PREVIOUS I,J.
C       ASSERT:  LABSP contains proper exit.
235Ø    ITEMP=I
        I=I22SAV
        I22SAV=ITEMP
        ITEMP=J
```

Digitized by Google

Digitized by Google

```
      J=J22SAV
      J22SAV=ITEMP
      GO TO 2400

C     STORE POINT (AND DISPLAY).
2400  CALL STORE(I,J,1)
      IF (ANYLIS) GOTO 2410
      ANYLIS=.TRUE.
      JLISPT=LESTAB(I,J)
      WSTART=JLISPT
      GO TO 2420
2410  JLISPT=LRINS(JLISPT,I,J)
2420  IF (IFDDPY.EQ.0) GO TO LABSP
C   COMPUTE DPY COORDS. OF POINT.
      NINEW=G1(I)
      NJNEW=G2(J)
C   INITIALIZE MPOG.
      IF (ANYM) GO TO 2450
      ANYM=.TRUE.
      NI=NINEW
      NJ=NJNEW
      CALL SETPOG(MPOG)
      CALL AIVECT(NI,NJ)
C   INITIALIZE NPOG.
      IF (NODPY) GO TO 2445
      CALL SETPOG(NPOG)
      CALL AIVECT(NI,NJ)
2445  ANYN=.FALSE.
      GO TO LABSP
C   COMPUTE NEW SLOPE.
2450  DXNEW=NINEW-NI
      IF (DXNEW.EQ.0) GO TO 2453
      SLONEW=(NJNEW-NJ)/DXNEW
      GO TO 2455
2453  SLONEW=9999.
C   IS SLOPE INITIALIZED?
2455  IF (ANYN) GO TO 2460
      ANYN=.TRUE.
      GO TO 2480
C   TEST NEW SLOPE.
2460  IF (SLONEW.EQ.SLOPEN) GO TO 2465
      TDIST=(TSLOPE*NINEW-NJNEW+TB)/TDENOM
C     GCONST = 1.5 * DISPLAY SCALE FACTOR.
      IF (ABS(TDIST).GE.GCONST) GO TO 2470
C   SET UP OLD NPOG.
2465  IF (NODPY) GO TO 2485
      CALL SETPOG(NPOG)
      GO TO 2485
C   ERASE OLD NPOG.
2470  IF (NODPY) GO TO 2473
      CALL HYDPOG(NPOG)
C   DISPLAY NEW MPOG.
```

Digitized by Google

```
2473     CALL SETPOG(MPOG)
         CALL AVECT(NI,NJ)
         IF (NODPY) GO TO 2480
         CALL DPYOUT(MPOG)
C    SET UP NEW NPOG.
         CALL CLRPOG(NPOG)
         CALL AIVECT(NI,NJ)
C    SET UP SLOPE PARAMETERS.
2480     SLOPEN=SLONEW
         TSLOPE=SLONEW
         TB=NJ-TSLOPE*NI
         TDENOM=SQRT(TSLOPE**2+1)
2485     NI=NINEW
         NJ=NJNEW
         IF (NODPY) GO TO 2490
C    DISPLAY NEW NPOG.
         CALL AVECT(NI,NJ)
         CALL DPYOUT(NPOG)
2490     GO TO LABSP

C        FINISHED WITH THIS P1 TO P2 SECTION.
3000     IQ=I
         JQ=J
         WIDE=NXTWID.OR.WIDE
         RETURN

         END
```

Digitized by Google

```
        TITLE JEDGOP;   (I,J,MDIRS)
; MDIRS:  4 low bits choose which directions to use.
;         Sign means:  +  quit after one sucess.
;                      -  try all.
;JEDGE3 FOR      ***
;                ***
;                ***
;JEDGE5 FOR      *-*-*
;                -----
;                *-*-*
;                -----
;                *-*-*
VALUE=0
K1=1
MDIRS=2
M1=3
M2=M1+1
M3=M2+1
M4=M3+1
M5=M4+1
M6=M5+1
M7=M6+1
M8=M7+1
LASTAC=M8

DEFINE  DOIT (BIT,N1,N2,N3,N4,N5,N6,NEXT,XLT)
        <DEFINE DOONE (N1,N2,N3,N4,N5,N6)
        <MOVE K1,N1      ;K1=N1                    ;K1=MIN(N1,N2,N3)
        CAMLE K1,N2      ;IF (K1,GT,N2) K1=N2      ;"
        MOVE K1,N2       ;"                        ;"
        CAMLE K1,N3      ;IF (K1,GT,N3) K1=N3      ;"
        MOVE K1,N3       ;"                        ;"
        SKIPN TFLAG      ;*********NEW
        SOS K1                                     ;K1=K1-1
        CAMG K1,N4       ;IF (K1,LT,N4) GOTO NEXT        ;IF (K1,GT,MAX(N4,N5,N6))
        JRST NEXT        ;"                        ;   THEN SET BIT
        CAMG K1,N5       ;IF (K1,LT,N5) GOTO NEXT        ;   ELSE GO TO NEXT
        JRST NEXT        ;"                        ;"
        CAMG K1,N6       ;IF (K1,LT,N6) GO TO NEXT       ;"
        JRST NEXT        ;"                        ;"
        ORI VALUE,BIT    ;SET BIT                  ;"
        JUMPL MDIRS,NEXT
        JRST EXIT>
        MOVE K1,N1
        SUB K1,N4
        JUMPL K1,XLT
        JUMPLE K1,NEXT   ;*********NEW
        SKIPN TFLAG      ;*********NEW
        SOJLE K1,NEXT
        DOONE (N1,N2,N3,N4,N5,N6)
XLT:    SKIPN TFLAG      ;*********NEW
        AOJGE K1,NEXT
```

Digitized by Google

```
                DOONE (N4,N5,N6,N1,N2,N3)>

        ;       M1 M2 M3
        ;       M4    M5
        ;       M6 M7 M8

                INTERN JEDGE3
        JEDGE3: 0
                SETZM TFLAG#    ;*********NEW
                MOVEI 0,TEMP
                BLT 0,TEMP+LASTAC
                JSA 17,SPLOAD   ;RETURNS WITH A1=POINTER TO I-1,J
                EXTERN SPLOAD   ;                   A2=LINLEN
                LDB 0,[POINT 6,1,5]
                CAIE 0,44       ;DOES POINTER IN AC1=44BB00,,A ?
                JRST PNTOK      ;NO.
                TLZ 1,770000.   ;YES. SET TO 00BB00,,A-1
                SOS 1
        PNTOK:  MOVE 0,1
                SUB 0,2 ;PREVIOUS LINE     A0=A1-LINLEN
                LDB M1,0
                ILDB M2,0
                ILDB M3,0
                MOVE 0,1        ;CURRENT LINE
                LDB M4,0
                IBP 0
                ILDB M5,0
                ADD 1,2 ;NEXT LINE  A1=A1+LINLEN
                LDB M6,1
                ILDB M7,1
                ILDB M8,1
                SKIPN NEWWAY
                JRST X99        ;OLD METHOD. NEWWAY.EQ.0
                JRST NEW99      ;NEW METHOD. NEWWAY.NE.0
        NEWWAY: 0

                INTERN JEDGE5
        JEDGE5: 0
        ;NEWER******    SETOM TFLAG     ;*********NEW
                MOVEI 0,TEMP
                BLT 0,TEMP+LASTAC
                JSA 17,SPLOAD   ;RETURNS WITH A1=POINTER TO I-1,J
                EXTERN SPLOAD   ;                   A2=LINLEN
                JUMPL 1,X69
                ADD 1,[040000000000]    ;BYTE POSITION IS 4 TO 34(OCTAL)
                JRST PNTOK5     ;ADD 4 TO IT.
        X69:    TLZ 1,700000    ;BYTE POSITION IS 40 OR 44(OCTAL).  SET TO
                SOS 1           ;00 OR 04.  ADDR TO ADDR-1.
        PNTOK5: MOVE 0,1
                SUB 0,2 ;PREVIOUS LINE     A0=A1-LINLEN
                SUB 0,2
                LDB M1,0
```

Digitized by Google

Digitized by Google

```
        IBP 0
        ILDB M2,0
        IBP 0
        ILDB M3,0
        MOVE 0,1        ;CURRENT LINE
        LDB M4,0
        IBP 0
        IBP 0
        IBP 0
        ILDB M5,0
        ADD 1,2 ;NEXT LINE  @1=A1+LINLEN
        ADD 1,2
        LDB M6,1
        IBP 1
        ILDB M7,1
        IBP 1
        ILDB M8,1
X99:    MOVE MDIRS,@2(16)
        SETZM VALUE

        TRNE MDIRS,1
        JRST DO1
TEST2:  TRNE MDIRS,2
        JRST DO2
TEST4:  TRNE MDIRS,4
        JRST DO4
TEST8:  TRNE MDIRS,↑D8
        JRST DO8
EXIT:   MOVSI LASTAC,TEMP+1
        HRRI LASTAC,1
        BLT LASTAC,LASTAC
        JRA 16,3(16)

DO1:    DOIT (1,M1,M4,M6,M3,M5,M8,TEST2)
DO2:    DOIT (2,M1,M2,M3,M6,M7,M8,TEST4)
DO4:    DOIT (4,M1,M2,M4,M5,M7,M8,TEST8)
DO8:    DOIT (↑D8,M2,M3,M5,M4,M6,M7,EXIT)

TEMP:   BLOCK 20

TDS=2   ; THRESHOLD FOR DOSIM
DEFINE  DOSIM (BIT,N1,N2,NEXT)
        <MOVE K1,N1
        SUB K1,N2
        MOVMS K1
        CAIG K1,TDS
        JRST NEXT
        ORI VALUE,BIT
        JUMPL MDIRS,NEXT
        JRST EXIT>

NEW99:  MOVE MDIRS,@2(16)
```

Digitized by Google

```
        SETZM VALUE

        TRNE MDIRS,1
        JRST D01S
TEST2S: TRNE MDIRS,2
        JRST D02S
TEST4S: TRNE MDIRS,4
        JRST D04S
TEST8S: TRNE MDIRS,↑D8
        JRST D08S

D01S:   DOSIM (1,M4,M5,TEST2S)
D02S:   DOSIM (2,M2,M7,TEST4S)
D04S:   DOSIM (4,M1,M8,TEST8S)
D08S:   DOSIM (↑D8,M3,M6,EXIT)


        END
```

Digitized by Google

Digitized by Google

Digitized by Google

Digitized by Google

Digitized by Google

3781
1970
K
Stack

Stanford University Libraries
Stanford, California

Return this book on or before date due.

JULY 18, 1976
JUN 15 1978
MAR 2 8 1983
AUG 1 [illegible] 1983
I.L.L.
MAY 1 1 1993 - ILL

Digitized by Google

Digitized by Google