22

Edge Detection in Pictures by Computer Using Planning

M. D. Kelly
Department of Computer Science
Stanford University

Abstract

This paper describes a program for extracting an accurate outline of a man's head from a digital picture. The program accepts as input digital, grey scale pictures containing people standing in front of various backgrounds. The output of the program is an ordered list of the points which form the outline of the head. The edges of background objects and the interior details of the head have been suppressed.

The program is successful because of an improved method for edge detection which uses heuristic planning, a technique drawn from artificial intelligence research in problem solving. In brief, edge detection using planning consists of three steps. A new digital picture is prepared from the original; the new picture is smaller and has less detail. Edges of objects are located in the reduced picture. The edges found in the reduced picture are used as a plan for finding edges in the original picture.

This paper describes the computer solution of a problem in picture analysis and description. It is suggested that a very simple concept, here called planning, can be of general value in similar tasks. In basic terms, planning consists of tentative scene description in a picture of greatly-reduced size and detail.

I. PROBLEM DEFINITION

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

The work described in this paper is a part of a larger project, the visual identification of people by computer. The main purpose of this research is to develop and improve techniques for picture processing by computer. The actual identification of a large group of individuals is only a secondary consideration. The method to be used for identification is the following: measurements such as height, width of head, and distance between eyes are obtained from the picture of the person; these measurements are then used in a simple pattern classification scheme. 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). In addition, Bledsoe has reported considerable success in identifying faces from their measurements in a man-machine system (Bledsoe 1966).

The emphasis in this work has not been on pattern classification, which forms so much of the pattern recognition literature. Rather, the main effort has focused on 'characterization' of the input picture, accurately locating features such as eyes, nose, or ears in digital pictures. 'It is generally recognized that characterization is really the principal problem in pattern recognition.' (Ho and Agrawala 1968 p. 2102).

The picture is represented inside the computer as a matrix of light intensity values. The pictures discussed in this paper contain the head of a man (or woman) against a background of normal room objects. All pictures contain the head in front view, looking at the TV camera.

From the picture of the head, 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.

This paper discusses a method for detecting edges (outlines of objects) in pictures. 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 fits precisely the description of planning given by Minsky (1963). It is believed that planning can be useful in many projects involving computer vision. In addition, it is helpful to relate the search for edges in pictures to the work on reduction of a given search space in the field of artificial intelligence.

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 line in figure 2 represents edges in figure 1.) Typically a gradient operation is applied to local areas of the picture to find the edges. The results of the gradient operation are improved with more global information in various routines which will follow an edge through a picture.

Specific desired edges remain elusive, in spite of many efforts at edge detection. Because of noise and the quantization of the picture, many false edges are found. Desired edges often vanish and reappear in grey scale pictures. Edge followers become confused on temporarily strong false paths. To reduce these problems, planning was adopted.

Edge detection by planning can be explained very simply. It is much like a defocused preliminary scan of a picture, an idea recommended by Kirsch *et al.* (1957). In spite of its simplicity, the idea has apparently never been used. Planning has three steps:

1. From the original picture extract a new small picture; figure 5 is an example, a small picture formed from figure 3.
2. Locate the desired edges in the new small picture; figure 7 shows the edges of the head found in figure 5.
3. Use the edges found in the small picture as a plan for finding edges in the original picture. See figure 2 for the results of this final operation.

The rest of this paper will discuss in more detail the ideas outlined above. Section 2 surveys previous work in finding edges. Section 3 introduces planning as used in artificial intelligence. Section 4 applies planning to edge detection. Section 5 gives the details of the program to find the outline of a head. The final section contains some results and a discussion of this and related methods.

2. COMPUTER EDGE DETECTION

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 4 which shows the edges present in figure 3. It is evident that most of the information of figure 3 is retained.

Edge detection has received considerable attention as a part of computer picture processing. Roberts (1965), in 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 Artificial Intelligence Project (Feldman *et al.* 1969), the SRI Robot project (Forsen 1968), and Rosenfeld, Thomas and Lee (1969).

Edges in pictures are detected by the use of operators which 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. Roberts used the following discrete approximation to the gradient. If A is the matrix representing the picture, the magnitude of the gradient at point $A_{i,j}$ is given by

$$|\mathrm{grad}(A_{i,j})|=\sqrt{[(A_{i+1,j+1}-A_{i,j})^2+(A_{i+1,j}-A_{i,j+1})^2]}$$

Figure 4 is an example of the results of the application of a gradient operator to the picture of figure 3.

There are many other possible approximations to the gradient. Some of these are given by Pingle (1969) [describes an operator developed by Sobel and Feldman], Rosenfeld (1969), Hueckel (1969), and Sakai, Nagao and Fujibayashi (1969). The use of higher derivatives or the Laplacian operator is sometimes suggested for finding edges. These methods seem unsatisfactory because of their tendency to amplify noise.

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 (1969, p. 136). Descriptions of working edge followers are given by Pingle (1966), Pingle, Singer, and Wichman (1968), and by Greenblatt, Holloway, and Sordillo (1966).

In spite of much effort, the accurate selection by computer of 'important' edges in fairly cluttered scenes has not been particularly successful. The methods given for edge detection are fine for presenting information to the human eye (as figure 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, Singer, and Wichman 1968, McCarthy *et al.* 1968, Paul, Falk, and Feldman 1969). Nilsson (1969, p. 513) states the problems with local methods of edge detection from experience at the SRI 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 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.'

You must know what you are looking for in order to make sense out of a complex picture. Global information about the edges in the picture and the structure of the objects they represent is needed. This is the idea behind the development of linguistic picture-processing techniques. A survey of these techniques can be found in Miller and Shaw (1968). Another approach to incorporating global structure into edge detection is the decision tree in use at SRI (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.

3. PLANNING

The previous section may be summarized as follows. Edge detection is necessary for picture analysis and description. For good results in edge detection, a global strategy is needed. When a global strategy is adopted, problems are encountered with a search space that is too large.

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 (1959) in their General Problem Solver.

Planning is discussed by Minsky (1963) in his paper 'Steps toward Artificial Intelligence'. 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. 435)
>
> 'Perhaps the most straightforward concept of planning is that of using a *simplified* model 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 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. 442)

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

4. EDGE DETECTION USING PLANNING

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 picture.

The new problem is much easier than the original problem. Many of the small features in the original picture are no longer present since 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 section 2 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 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.

5. PLANNING APPLIED TO A SPECIFIC PROBLEM

This section describes the program that was used to find the outline of the head.

Input

Picture input is provided by a vidicon television camera attached to a PDP-10 computer. The system permits rapid acquisition of a digital matrix representing light intensities in a scene. The size of the matrix is variable, the maximum size is 256 × 333 points. Each point of the matrix contains a four-bit number ranging from 0 (very dark) to 15 (very light). The picture presented in figure 3 will be used as an example throughout this section. The size of this picture is 226 × 325 points.

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 × 8 square in the original picture. Figure 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 5 is 28 × 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.

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

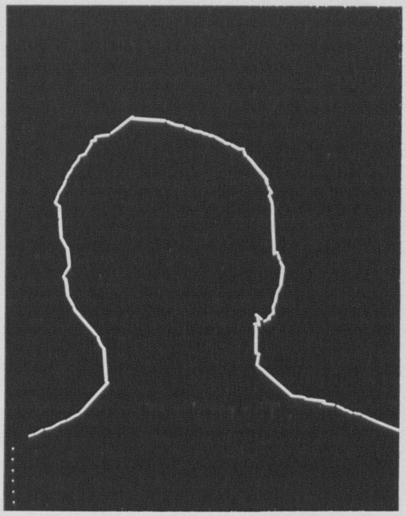

Figure 2. The outline of a head, the result of processing figure 1.

Figure 3. Unprocessed input picture used to obtain figures 4 through 8.

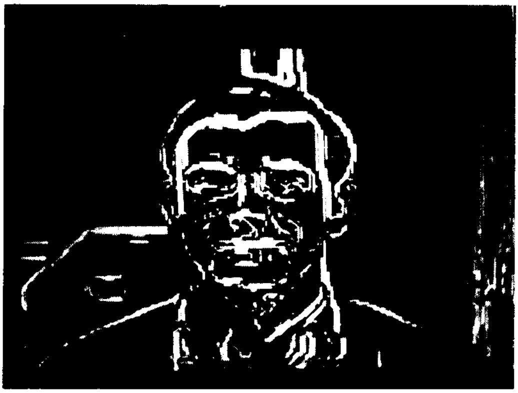

Figure 4. The results of applying a gradient operator to figure 3.

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

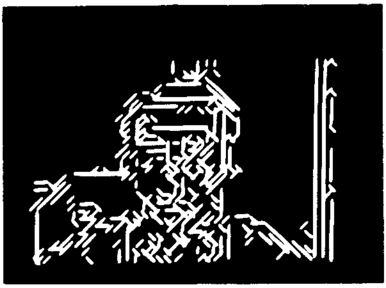

Figure 6 The results of applying an edge-finding operator to figure 5.

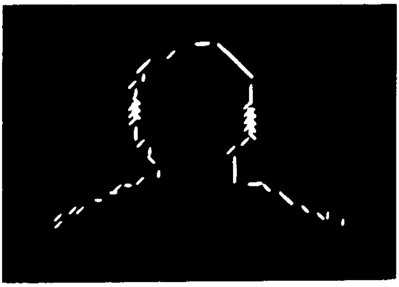

Figure 7. The 'plan' extracted from figure 6 which will be used to follow the head outline.

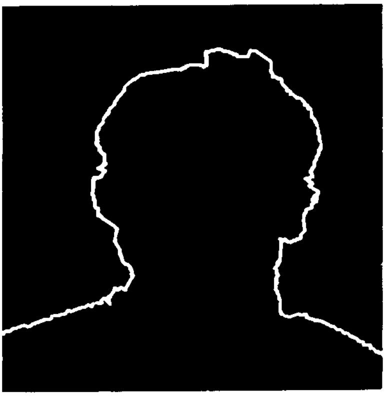

Figure 8. The outline of the head in figure 3, obtained using the plan of figure 7.

Find all edges in the reduced picture

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 *small edge matrix*. Figure 6 is an example of the small edge matrix derived from figure 5. Each element in the matrix contains four bits, representing the four 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 in section 2. 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 even this is preferable to reporting false edges. A more straightforward 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 used effectively in the small picture. Most of the smoothing utilized by Hueckel's operator has already been done during averaging.

The edge operator used is applied to 3 × 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 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 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 SRI decision tree search (Nilsson 1969). If no head was present in the original picture, its absence 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 being part of the top and sides of the head. The spatial relationship between these lines must be reasonable (for example, 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 *plan follower*. 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 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, and so on) 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 3×3 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 5×5 square, for example,

$$\begin{array}{c} A-B-C \\ -\ -\ -\ -\ - \\ D-E-F \\ -\ -\ -\ -\ - \\ G-H-I \end{array}$$

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

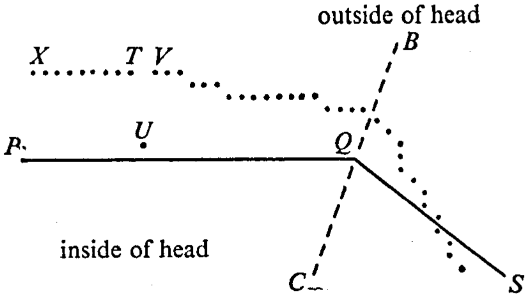

Figure 9. Operation of plan follower

The details of the plan follower are given below. Reference is made to figure 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 edge detection operator along a line perpendicular to *PQ*. Normally the edge detection operator is applied only 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 track correctly 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.

6. RESULTS AND DISCUSSION

The program which has been described does a good job of finding head outlines. It is difficult, however, to quantify what is meant by a good head outline. A qualitative measure has been used; the original picture is compared visually with the outline which is produced. In order to give an idea of the effectiveness of this program, three additional pictures with their resultant outlines are presented in figures 10, 11, and 12.

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 × 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 × 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 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. This increase of speed is particularly important when designing search programs on an interactive computer system.

Some tests were made to determine the speed improvement obtained using planning. The 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 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 seconds. This time, as well as all others mentioned, was measured on the PDP-10 computer and is accurate to ±20 per cent. 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 × 8 squares. This time was about 64 times the speed without planning, namely, 160 seconds.

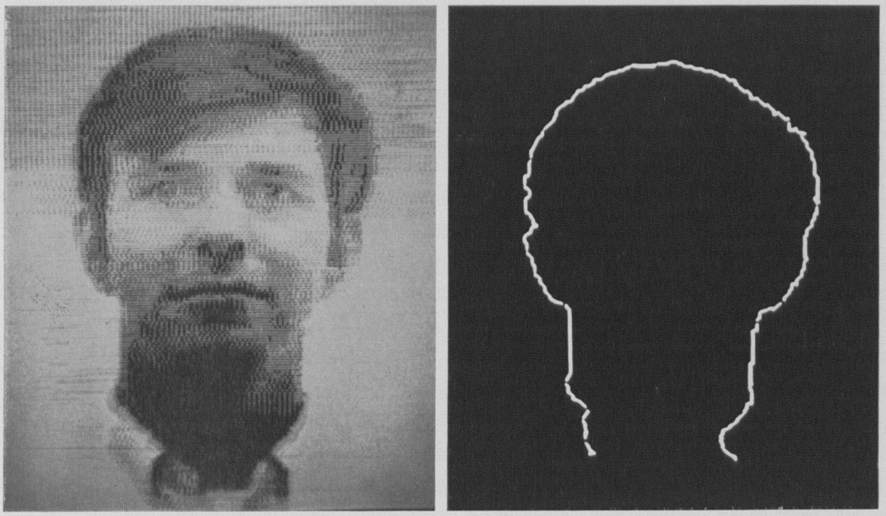

Figure 10

Figure 11

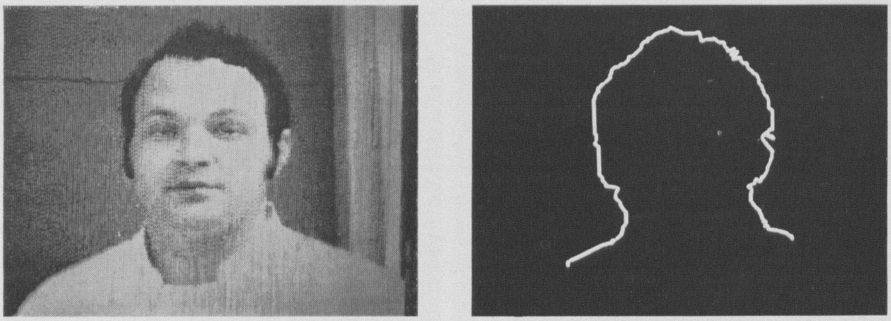

Figure 12

All edges were found in the smoothed picture. This took 34 seconds, again a factor of about 64.

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 its 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.

The plan-following stage has no counterpart time when not using planning.

In summary, the program without planning took 234 seconds, 40 times slower than the time using planning.

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 are known, its detection can be hard. For such edges, planning is essential.

Sakai, Nagao, and Fujibayashi (1969) have recently reported their work on finding faces in photographs. They first produce a picture, the result of a gradient operation, which contains the edges of the input picture. Large templates corresponding to the head outline and facial features are then matched with the edge picture. All reasonable positions and sizes of the template are tried. This method appears to be time consuming when compared to the planning method, and the result is only an approximate location for the head in the picture.

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. The plan found in the $\frac{1}{8}$ size picture could be used to find edges in the $\frac{1}{4}$ size picture. This could then be used as a plan to find edges in the $\frac{1}{2}$ 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 $\frac{1}{2}$ 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.

The greatest deficiency of the program described here is the lack of a high-level mechanism for describing picture structure. In order to find objects in cluttered pictures, a program must know what it is looking for. It would be desirable for that part of the program which finds the plan (the head outline) in the small picture to be general purpose. To such a general purpose program one could specify in some higher level 'language' that a head is desired where a head is:

round on top;
somewhat flat on the sides;
curving inward for the neck below the sides;
and so on.

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 the picture-description languages discussed by Miller and Shaw (1968) which were mentioned earlier. Such a general-purpose program was not used for finding head outlines. Instead, the structure of the head was embedded implicitly in the program. The reason for this is that picture-description languages are not sufficiently well developed to permit the specification of the numerous heuristic tests necessary to find the head outline. Satisfactory formal description of picture structure remains a major obstacle to advances in picture processing.

Acknowledgement

The author would like to thank Professor D.R.Reddy for his ideas, encouragement, and criticism which played a major part in the development of the work reported here. In addition, discussions with J.M.Tenenbaum contributed many fresh viewpoints to this paper.

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

The author is now at the School of Information and Computer Science, Georgia Institute of Technology.

REFERENCES

Bledsoe, W.W. (1966) Man-machine facial recognition. Panoramic Research Inc., Report PRI: 22.

Feldman, J.A., Feldman, G.M., Falk, G., Grape, G., Pearlman, J., Sobel, I. & Tenenbaum, J.M. (1969) The Stanford hand-eye project. *Proc. Int. Joint Conf. on Art. Int.* 1969, pp. 521–6.

Forsen, G.E. (1968) Processing visual data with an automaton eye. *Pictorial Pattern Recognition,* pp 471–501 (eds Cheng, G.C. *et al.*). Washington, DC: Thompson Book Co.

Greenblatt, R., Holloway, J. & Sordillo, D.A. (1966) Sides 21, *Art. Int. Memo No. 104-2 (MAC–M–320).* MIT, Cambridge, Mass.

Guzman, A. (1968) Decomposition of a visual scene into three-dimensional bodies. *Proc. AFIPS 1968 Fall Joint Comp. Conf.,* **33,** 291–304. Washington DC: Thompson Book Co.

Hueckel, M. (1969) An operator which locates edges in digitized pictures. *Art. Int. Memo No. 105.* Stanford University, California.

Ho, Y. & Agrawala, A.K. (1968) On pattern classification algorithms, introduction and survey. *Proc. IEEE,* **56,** 2101–14.

Kirsch, R.A., Cahn, L., Ray, C. & Urban, G.H. (1957) Experiments in processing pictorial information with a digital computer. *Proc. Eastern Joint Comp. Conf.,* pp. 221–9. New York: Spartan Books.

McCarthy, J., Earnest, L.D., Reddy, D.R. & Vicens, P.J. (1968) A computer with hands, eyes, and ears. *Proc. AFIPS 1968 Fall Joint Comput. Conf.*, **33**, 329–38. Washington DC: Thompson Book Co.

Miller, W.F. & Shaw, A. (1968) Linguistic methods in picture processing: a survey. *Proc. AFIPS 1968 Fall Joint Comp. Conf.*, **33**, 279–90. Washington DC: Thompson Book Co.

Minsky, M. (1963) Steps toward artificial intelligence. *Computers and Thought*, pp. 406–50 (eds. Feigenbaum, E.A. & Feldman, J.). New York: McGraw-Hill.

Narasimhan, R. & Forango, J.P. (1964) Some further experiments in the parallel processing of pictures. *IEEE Trans. EC-13*, 748–50.

Newell, A., Shaw, J.C. & Simon, H.A. (1959) Report on a general problem solving program. *Proc. Int. Conf. on Info. Proc.*, pp. 256–64. Paris: UNESCO House.

Nilsson, N.J. (1969) A mobile automaton: an application of artificial intelligence techniques. *Proc. int. Joint Conf. on Art. Int.*, pp. 509–15.

Paul, R., Falk, G. & Feldman, J.A. (1969) The Computer representation of simply described scenes. *Art. Int. Memo No. 101*. Stanford University, California.

Pingle, K.K. (1966) A program to find objects in a picture. *Art. Int. Memo No. 39*. Stanford University, California.

Pingle, K.K. (1969) Visual perception by a computer, *Automatic Interpretation and Classification of Images*, pp. 277–84 (ed. Grasselli, A.), New York: Academic Press.

Pingle, K.K., Singer, J.A. & Wichman, W.M. (1968) Computer control of a mechanical arm through visual input. *Proc. IFIP Conf.*, H pp. 140–6. Edinburgh.

Roberts, L.G. (1965) Machine perception of three-dimensional solids. *Optical and Electro-Optical Information Processing*, pp. 159–97 (eds. Tippett, J.T. *et al.*). Cambridge, Mass.: MIT Press.

Rosenfeld, A. (1969) *Picture Processing by Computer*. New York: Academic Press.

Rosenfeld, A., Thomas, R.B. & Lee, Y.H. (1969) Edge and curve enhancement in digital pictures. *University of Maryland Computer Science Technical Report 69-93*.

Sakai, T., Nagao, M. & Fujibayashi, S. (1969) Line extraction and pattern detection in a photograph. *Pattern Recognition*, **1**, 233–48.

Thorwald, J. (1965) *The Century of the Detective*. New York: Harcourt, Brace & World.