愛知工業大学研究報告
第23号Ｂ　昭和63年

Picture Cutting by a Method of Edge Following

Tomio KUROKAWA

輪郭線追跡による画像の切り抜き

黒　河　富　夫

An efficient edge following algorithm is presented. It is a one-pass sequential scanning procedure. Given two separate points near a region boundary of a digital gray scale picture, it extracts a path from one point to the other in such a way that the average of the evaluation value (mainly the edge strength) of the points along the path becomes 'near' maximum. A sequence of points outlining a region of a picture provided, it is possible to extract the entire boundary curve of the region by repeatedly using the algorithm.

Generally, there are two ways, in graph search, to get an optimum path which connects the two point. One is to find a path which gives the minimum cost. The other is to find that of the maximum merit. The former tends to extract a shorter path, because the shorter the path, the smaller the cost gets. The latter inclined to have a longer path. In this regard, the averaging process of the cost or the merit along the path is introduced for normalization which enables to compare a long path with a short one on a common base. Divisions, however, are required to do it. A large number of divisions may sacrifice the processing speed. A method is suggested to avoid them. It is to add a penalty (a negative value) to the sum of the merit along the path whenever the path gets a unit longer while maximizing the sum. A number of experiments are done with good results.

1. Introduction

Superimposition and 'Image Combine' (two pictures transparently overlapped) are a very popular technique of picture composition. Cutout pictures or cutout masks are frequently used in those processing techniques. Edge line formation along a specified region boundary of a picture is a very important part of picture cutting. Many researches from general to specific areas have been made on the edge detection or the edge following of digital pictures[1),2)]. Very few of them, however, are for the picture cutting. This paper introduces an edge following technique applied to the picture cutting (possibly be applied to other area).

A number of methods of picture cutting or mask making are knowm, four of which are described below :

1) Manual cutting. This is a method of pre-digital picture processing. Picture film or mask film is manually cut.

2) Interpolation. This method uses a computer and a coordinate locator devices such as a display monitor, a track-ball or a tablet. The boundary curve is formed by tracing a specified picture region. But a piecewise interpolation is done between the traced points with a straight line or a curve[3)].

3) Thresholding. This makes use of the brightness difference between a specified region and the background. It makes a mask (or a bilevel picture) without forming a boundary curve.

4) Guide line. A sequence of points roughly specifying a picture region are given somehow prior. Connecting those points in sequence gives a sequence of line segments of the region outline. Then the boundary curve search is made scanning the picture, perpendicular to each of those line segments[4),5)]. The boundary curve is formed connecting the pixels with most significant brightness changes. This guide line concept is employed in this research, too, but differently.

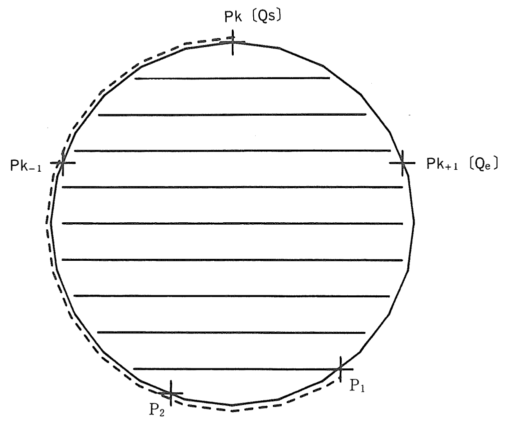


Fig. 1 Outline points of a conceptual picture.

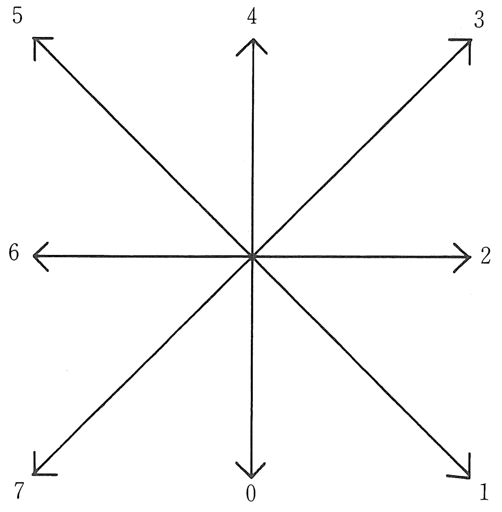


Fig. 2 Direction code assignment.

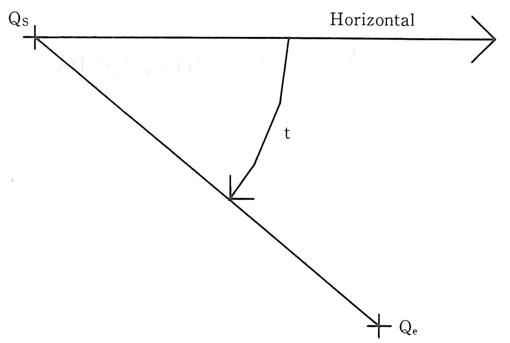
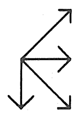
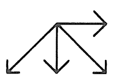
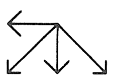
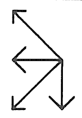
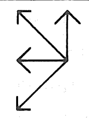
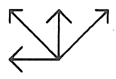
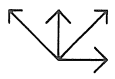
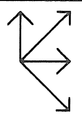


Fig. 3 Angle t.

Table 1 8 direction types.

1) $0^\circ \le t < 45^\circ$	2) $45^\circ \le t < 90^\circ$	3) $90^\circ \le t < 135^\circ$	4) $135^\circ \le t < 180^\circ$
			
5) $180^\circ \le t < 225^\circ$	6) $225^\circ \le t < 270^\circ$	7) $270^\circ \le t < 315^\circ$	8) $315^\circ \le t < 360^\circ$
			

The author has been doing the research for the picture cutting by applying the edge following principles of digital pictures to the problem[6),7)]. The edge following can be considered as an optimization problem of graph search[8)] or DP[9)]. The picture cutting, however, does not necessarily require a precise optimum solution for the problem. A 'good' solution can be accepted. Here, this paper presents an algorithm which gives a 'good' solution. Given two separate points which are considered to be near a boundary curve of a picture region, the algorithm extracts a curve as a 'near' optimal path connecting the two points. Repeated application of the algorithm produces the entire boundary curve of the specified picture region.

2. Theory

Consider to extract a boundary curve of a region in a gray scale digital picture given a sequence of points which makes a rough outline of the region. The boundary curve extraction is done as follows. First, an edge detection operator such as Sobel's is applied near the outline of the region, then a graph search operation is employed to get a boundary curve connecting the each two points in sequence.

Fig. 1 shows the concept of the boundary curve formation. Given a sequence of points (P_1, P_2, . . . , P_k, . . .) near the region (a circle) boundary, a sequence of the boundary curve segments ($P_1 \sim P_2$, . . . , $P_{k-1} \sim P_k$) are extracted. The extracted curve is shown by the dotted curve. The problem, then, is to connect each of the two points in sequence to form a natural boundary curve. A simple method is an interpolation with a straight line or with an arc, which are frequently used in the applications. It does not, however, make a natural curve because the curve is formed without processing the picture data which themselves compose the region bound-

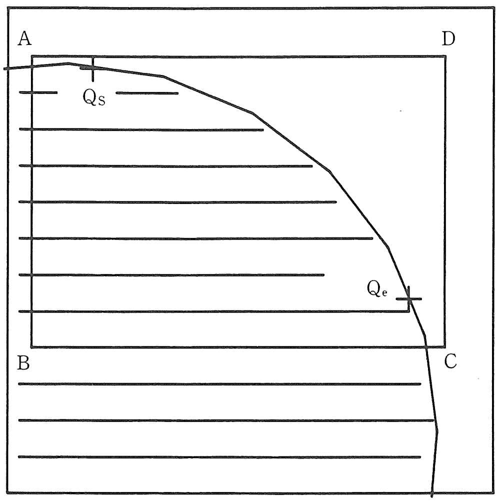


Fig. 4 Area to be processed.

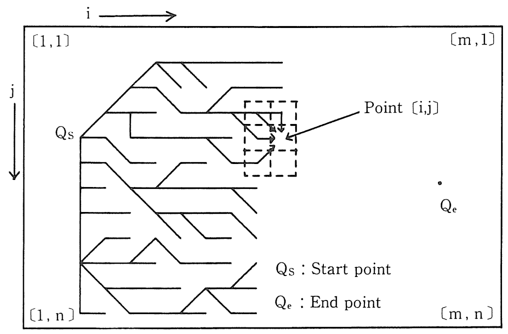


Fig. 5 Concept of the edge following.

ary. A better way is to make use of the edge strength data of the picture to extract a boundary curve. One method to do it is shown as follows.

Consider the two points P_k (Q_s) and P_{k+1} (Q_e) as a start point and an end point, respectively, in Fig. 1. The two points are to be connected from Q_s to Q_e.

Generally, there are eight connecting directions as shown in Fig. 2. All the eight directions, however, are not required to draw a boundary segment of a local picture region. Take Fig. 1 for instance. Connecting Q_s to Q_e only requires the four directions coded 0, 1, 2 and 3 (3 not necessarily required in this case).

Considering the above, determine the angle t, shown in Fig. 3, which depends on the positional relation of the two points Q_s and Q_e, classify the angle into eight groups as in Table 1, and use only the corresponding four directions to form the boundary curve.

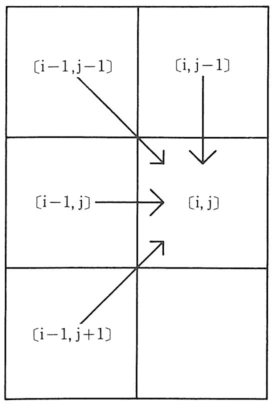


Fig. 6 Predecessor points.

On the other hand, the entire picture area as shown in Fig. 1 is not necessary in order to connect the two points in making use of the picture edge strength. The necessary picture area is a small area which includes the two points. For example, the area surrounding the rectangle ABCD as shown in Fig. 4 provides enough data to do the curve extraction.

Fig. 5 shows the edge following concept of this paper. Starting from Q_s and scanning one vertical line at a time, it simultaneously searches a number of curves repeatedly connecting an optimal point to a midway point (i, j). Each single connection is done by selecting an optimal point from among the four predecessor points and connecting it to (i, j). The predecessor points are defined as in Fig. 6. The criteria of the optimality should be defined beforehand and possibly include the sum or the average of the edge strength along the connecting curve and the curvature of the local curve. A specific algorithm is shown below for the case that angle t is from 0° to 45°. Other cases can be handled by a simple modification of the algorithm.

[Algorithm] A method that uses the edge strength average along the connecting path (curve) as a main optimization criteria. Let $[a_{ij}]$ ($i=1, 2, \ldots, m$; $j=1, 2, \ldots, n$) be the inside area of the above mentioned rectangle. Let (m_s, n_s) and (m_e, n_e) be the coordinates of the starting point Q_s and the end point Q_e, respectively. Let $[b_{ij}]$, $[c_{ij}]$ and $[d_{ij}]$ be the arrays of the same size as $[a_{ij}]$. Each element of those arrays corresponds with that of $[a_{ij}]$. b_{ij} represents the sum of the edge strength along the curve from Q_s to the point (i, j).

[1] Zero the arrays $[b_{ij}]$ and $[c_{ij}]$. Empty $[d_{ij}]$.

[2] Mark the point (m_s, n_s) as connected.

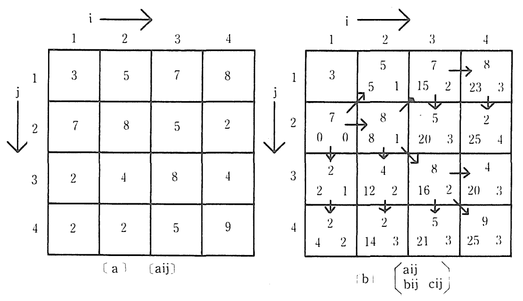


Fig. 7 Sample numerical data.
〔1, 2〕 : start point, 〔4, 4〕 : end point.

[3] Perform step [4] for $i=m_s, m_s+1, \ldots, m_e$.
[4] Perform ⟨1⟩ , ⟨2⟩ and ⟨3⟩ for $j=1, 2, \ldots, n$.
⟨1⟩ Select a point out of the four predecessors (see Fig. 6) of (i, j) so that
$(b_{io,jo}+a_{i,j})/(c_{io,jo}+1)$ (1)
becomes maximum,
where (io, jo) is $(i, j-1)$, $(i-1, j-1)$, $(i-1, j)$, or $(i-1, j+1)$.
⟨2⟩ Do the operations :
$b_{i,j} \leftarrow b_{io,jo}+a_{i,j}$, (2)
$c_{i,j} \leftarrow c_{io,jo}+1$, and (3)
$d_{i,j} \leftarrow d_{io,jo} \;||\; \mathrm{CODE}\,((io, jo), (i, j))$, (4)
where CODE ((io, jo), (i, j)) is the direction code from (io, jo) to (i, j) and ' || ' means concatenation.
⟨3⟩ If (i, j) is (m_e, n_e), then stop.
($d_{me,ne}$ is the sequence of the chain codes of the extracted boundary curve, The curve drawn starting from Q_s by the code sequence is the boundary curve sought.) End of Algorithm.

Fig. 7 (a) is a numerical example of an edge strength array of size (4×4). The algorithm extracts a number of paths shown in Fig. 7 (b). The final result is the path : (1, 2), (2, 2), (3, 3) and (4, 4).

The algorithm tries to extract the path such that the average of the edge strength along the path from the start point to the end point :

$$\Sigma_i\, a(Q_i)\,/\,L \qquad (5)$$

becomes maximum, where $a(Q_i)$ (≥ 0) is the evaluation value at the point Q_i ; L is the path length, which differs from path to path. Since it does not necessarily become maximum, it should be called 'near' maximum. A drawback of the algorithm is the division in (1) or (5). A large number of the divisions sacrifice the processing speed. There are other optimization methods without such divisions.

One method is to maximize the sum of the evaluation value along the path :

$$\Sigma_i\, a(Q_i). \qquad (6)$$

(6), however, becomes larger when the path becomes longer. Accordingly, it tends to extract a longer path possibly erroneous as the boundary curve.

The evaluation value of each individual point can be converted to 'cost'. To minimize the sum of the cost[10] along the path :

$$\Sigma_i\, (M-a(Q_i))\,, \qquad (7)$$

where M is a constant larger than any $a(Q_i)$, is another optimization to extract a boundary curve. In this case, however, the shorter the path, the smaller (7) becomes. Hence, it tends to extract a shorter boundary curve, possibly erroneous, too.

The above discussion suggests that the averaging in (5) is a kind of a normalization process.

Regarding the above,

$$\Sigma_i\, (a(Q_i)-M) \qquad (8)$$

can be considered as an evaluation function which is expected to have the same kind of effect as (5) of averaging. M is a positive constant which is considered to be a penalty for the path gets a unit longer.

On the other hand, there is a method which maximizes (not 'near' maximizes) (5) using DP method[9]. It is to extract a number of paths with different length and then select one which makes (5) maximum. The extracted path should be mathematically optimum. But it requires tremendous computing time and memory.

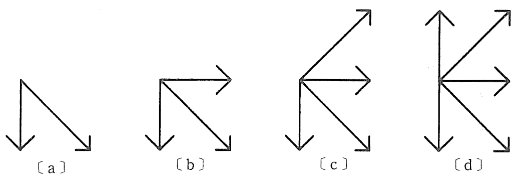


Fig. 8 Other operators.

A desired boundary curve can be obtained by a simpler processing. The mentioned [Algorithm] uses an operator with four connecting directions. It is possible, however, to change the number from 2 to 5 as shown in Fig. 8 depending on the situations. An operator with more directions can cope with relatively more complex boundaries sacrificing the processing speed. The number of divisions in averaging increases accordingly. It is the number of the connecting directions times the number of the pixels in the processed area.

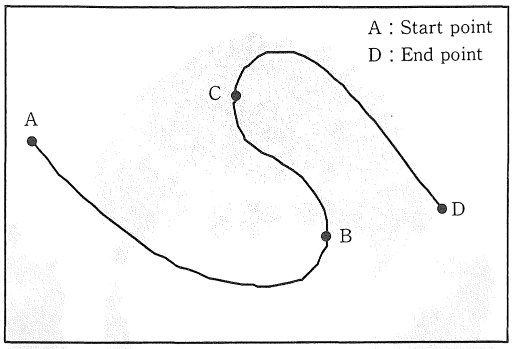


Fig. 9 A sample edge line which the algorithm cannot handle at a time.

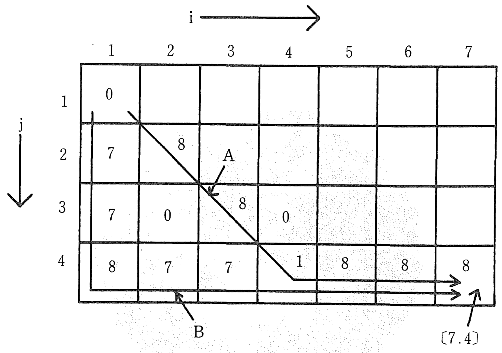


Fig. 10 An example in which optimal path (giving maximum average)is not selected.

When the operators of Fig. 8 (a) rotated as in Table 1 is used, the division of (1) is not necessary. And when the operator of Fig. 8 (b) rotated is used and it corresponds to (2), (4), (6), or (8) of Table 1, the division of (1) is not necessary, too. It is because all the possible paths have the same length. That is, all the paths starting from Q_s, no matter what point they pass through, have the same number of steps to reach the end point Q_e. In those cases, the optimization process corresponds to that of DP, which handles paths of same length.

When the operator of Fig. 8 (d) is used, the evaluation of selecting an optimum point for each single connection has to be done twice. It can, however, handle more complex picture boundaries.

The number of connecting directions is desired to be 8. The reason of limiting the number in the algorithm is to use the one-pass sequential scanning so that the processing becomes simple. Because of it, the curve from A to D in Fig. 9 cannot be extracted at once. That is, the curve segment from B to C, where the variable i of the algorithm decreases, cannot be followed. For such a case, it is necessary to divide the processing as from A to B, B to C, and C to D.

Next, Fig. 10 is an example in which the algorithm does not extract a mathematically optimum path. The processing starts from (1, 1). When it reaches (4, 4), the evaluation value (37/6) of the path B is bigger than that (17/3) of the path A. But, at (7, 4), the value (41/6) of the path A is bigger than that (61/9) of the path B. Since the path A has been selected at (4, 4), the path B would never be selected as the final result.

3. Experiments and Results

In the following experments,

$$a_{ij} = \max(f_1, f_2, f_3, f_4) \qquad (9)$$

is used employing 'Kirsh-motivated' templates (n=1)[11] for the edge detection operator, where

$$f_1 = | g_{i+1,j-1} + g_{i+1,j} + g_{i+1,j+1} - g_{i-1,j-1} - g_{i-1,j} - g_{i-1,j+1} | ,$$

$$f_2 = | g_{i-1,j+1} + g_{i,j+1} + g_{i+1,j+1} - g_{i-1,j-1} - g_{i,j-1} - g_{i+1,j-1} | ,$$

$$f_3 = | g_{i,j+1} + g_{i+1,j+1} + g_{i+1,j} - g_{i-1,j} - g_{i-1,j-1} - g_{i,j-1} | , \text{ and}$$

$$f_4 = | g_{i-1,j} + g_{i-1,j+1} + g_{i,j+1} - g_{i,j-1} - g_{i+1,j-1} - g_{i+1,j} | ,$$

where g represents a pixel's brightness.

Fig. 11 and Fig. 12 are the original pictures, a circle and a child's head, respectively, used in the experiments. The dark circle of Fig. 11 originally drawn on a white paper, photographed by a TV camera and then displayed on a CRT monitor through an AD converter, a digital memory (512×480×8 bits) and a DA converter. The picture displayed on the monitor is finally photographed by a camera and printed. Fig. 12 is obtained in the same process as Fig. 11 from a printed paper picture. Fig. 13 and 14 show the rough outlines of the circle and the child's face, recpectively, by '+' marks. The coordinates of the marks on the array of 512×480 are (341, 179), (310, 109), (262, 81), (196, 83), (131, 144), (123, 200), (169, 275), (250, 291), and (315, 254) for Fig. 13 and (319, 263), (327, 220), (336, 197), (342, 178), (360, 131), (328, 78), (278, 33), (218, 22), (176, 37), (134, 88), (120, 146), (126, 188), (144, 238), (169, 250), (203, 294), and (279, 304) for Fig. 14. In all the experiments below, the edge following is done using those two sequences of the points, both start at the mark

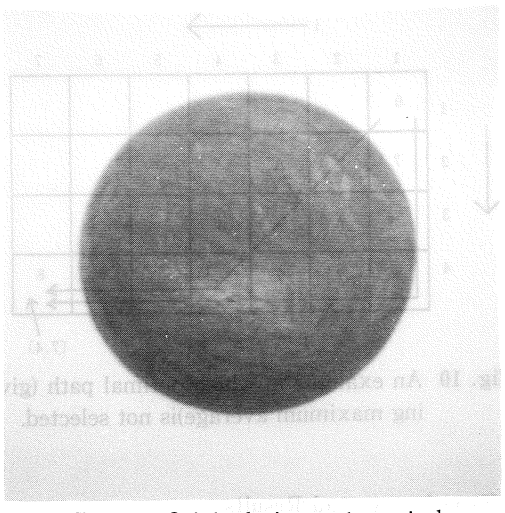


Fig. 11 Original picture 1 : a circle.

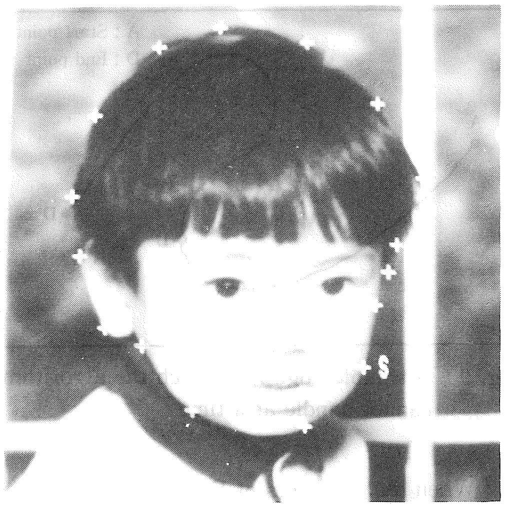


Fig. 14 Outline points of the head.

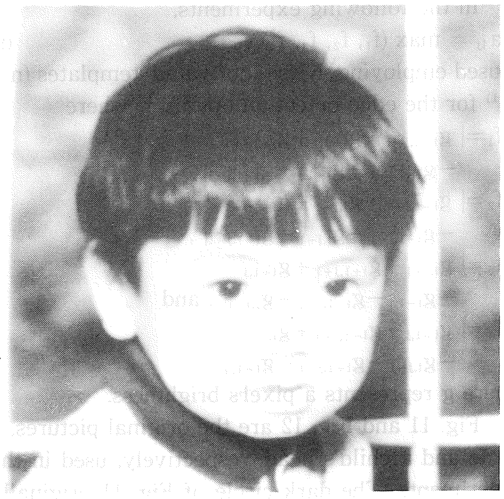


Fig. 12 Original picture 2 : a child's head.

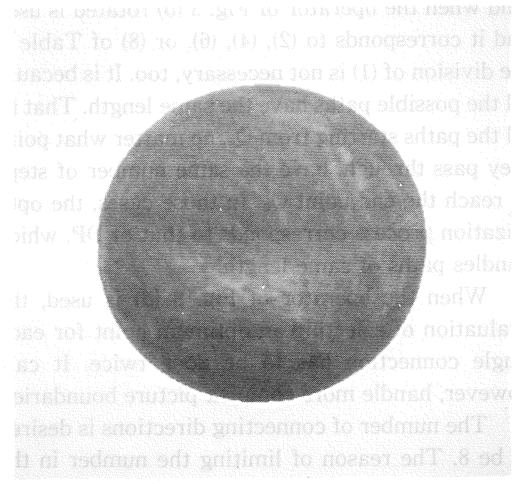


Fig. 15 Edge line by averaging with no smoothing (using the shown algorithm).

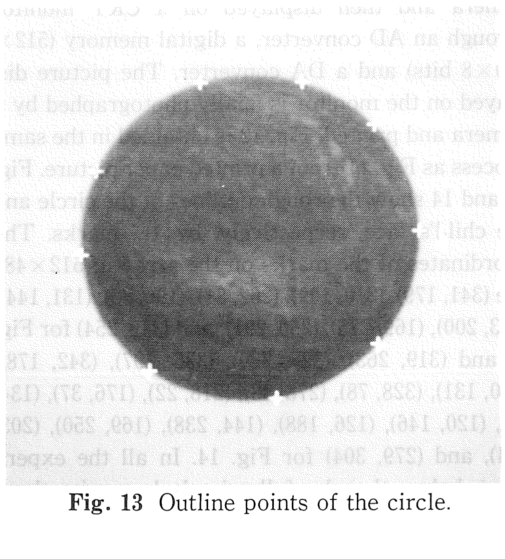


Fig. 13 Outline points of the circle.

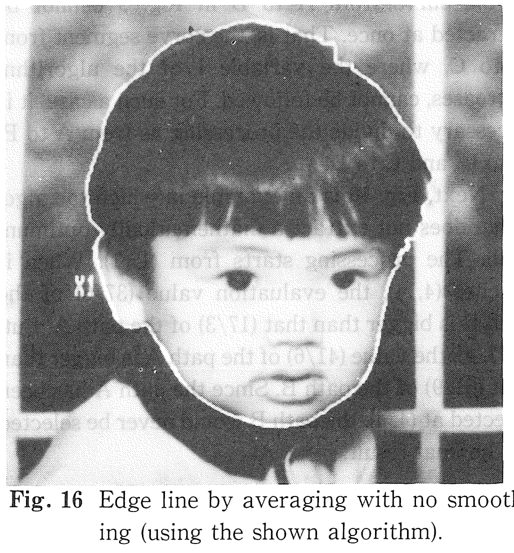


Fig. 16 Edge line by averaging with no smoothing (using the shown algorithm).

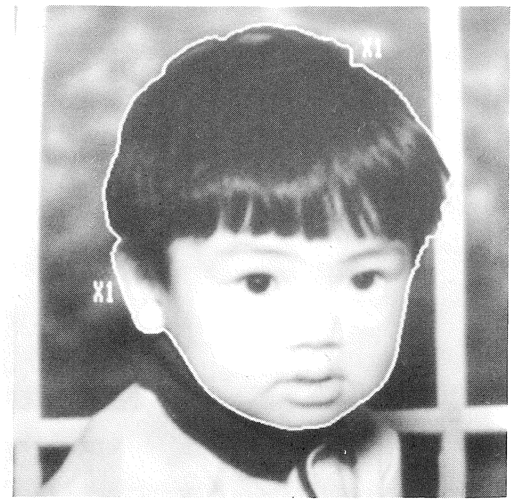


Fig. 17 Edge line by averaging with smoothing (using the shown algorithm and smoothing).

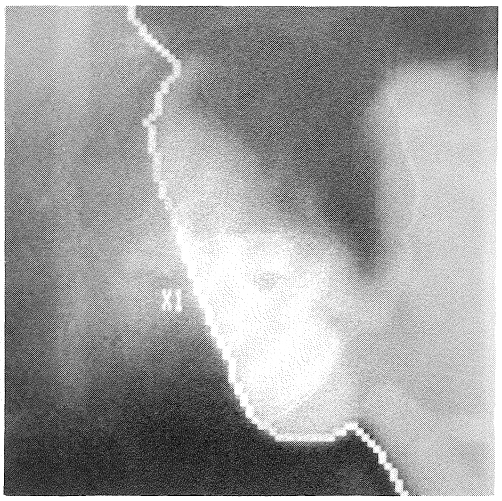


Fig. 19 A part of Fig.17 four times enlarged around the lower X1.

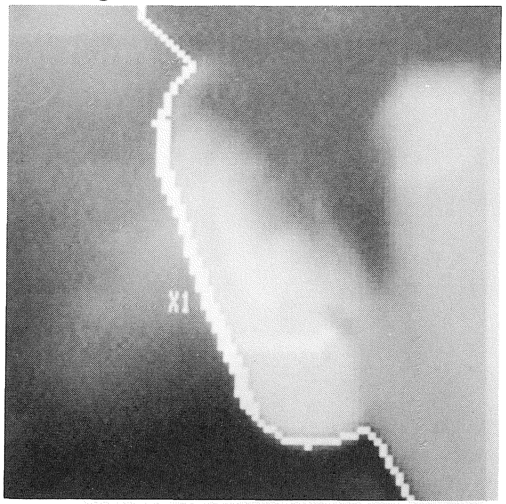


Fig. 18 A part of Fig.16 four times enlarged around the mark X1.

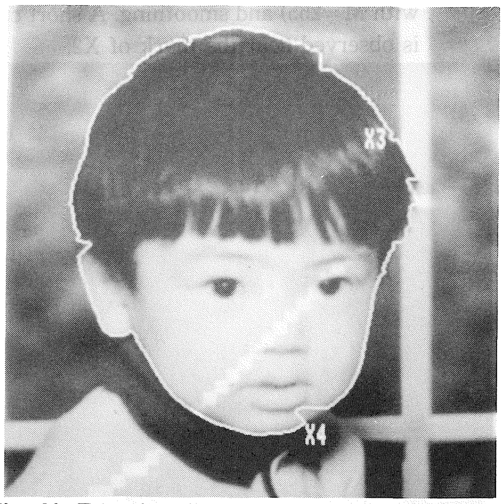


Fig. 20 Edge line by simple summing (maximizing (8) with M=0) and smoothing. The line is erroneusly lengthened near the marks of X3 and X4.

S and go round counterclockwise.

[Example 1] : Fig. 15: It is the result by averaging only. The algorithm is applied exactly as described. The white curve is the extracted boundary curve of the circle. It looks 'not bad'.

[Example 2] : Fig. 16: The same method is used as in Example 1. Some parts of the curve are thick because of the zigzag of the curve.

[Example 3] : Fig. 17 : A modification is made to the previous algorithm so as to remove the zigzag. The modification is the addition of the priority selection process at step [4] . That is, the optimal point (io, jo) is selected in such a way that a big direction change of the connecting curve would not occur at a single point. The point (io, jo) is selected in the priority order: (1) the direction change is within 45°, (2) it is within 90°, (3) it is within 135°. Judging from Fig. 17, it is considered that the curve gets thin and smooth. Fig. 18 is the part of Fig. 16 four times enlarged around X1. Fig. 19 is that of Fig. 17. It is observed that the curve is thin and smooth in Fig. 17 and 19.

[Example 4] : Fig. 20 : (8) with M=0 is 'near' maximized by the algorithm, which is equivalent to

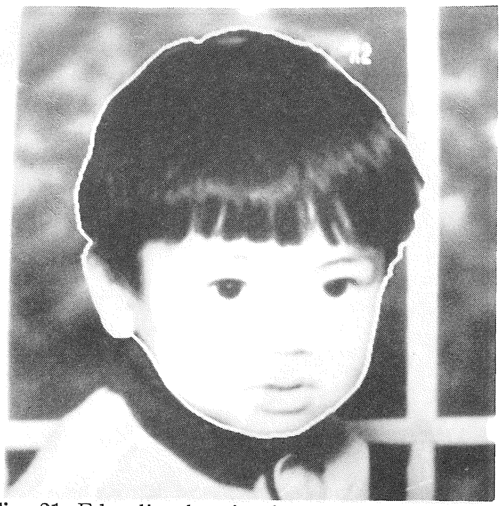


Fig. 21 Edge line by simple summing (maximizing (8) with M=255 or minimizing (7) with M=255) and smoothing. A short cut is observed near the mark of X2.

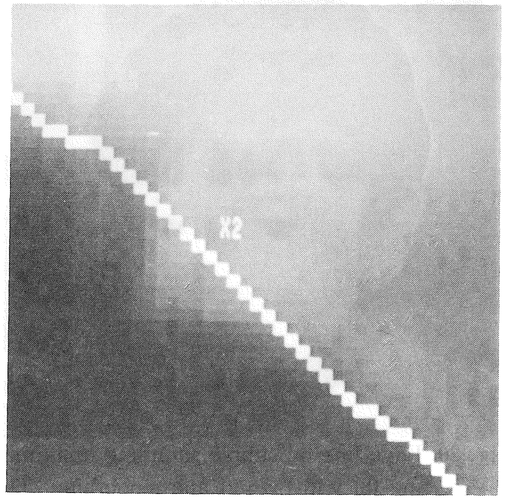


Fig. 22 A part of Fig.21 eight times enlarged around the mark X2.

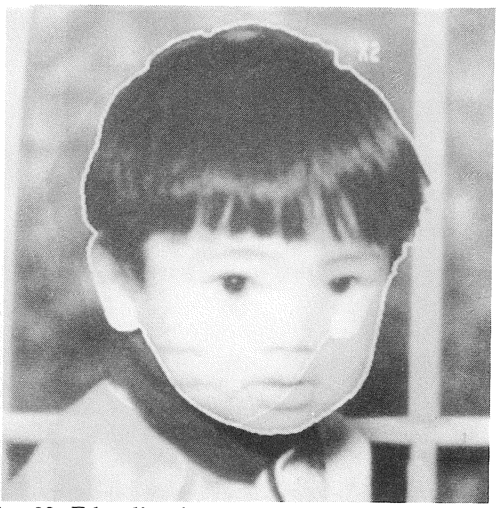


Fig. 23 Edge line by simple summing (maximizing (8) with M=50) and smoothing. This line is close to that of Fig.17.

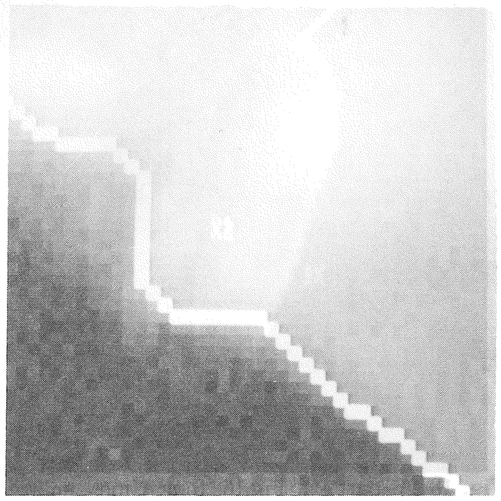


Fig. 24 A part of Fig.23 eight times enlarged around the mark X2.

'near' maximizing (6). It is clear that the curve is erroneously lengthened near X3 and X4.

[Example 5] : Fig. 21 : The same process of Example 4 (maximize (8)) but with M=255 is applied. It is expected that it has the same effect to 'near' minimize (7) with M=255. That is, it should extract a shorter curve. Accordingly, the extracted curve segment near X2 of Fig. 21 is straight (short), though it should be like letter L. Fig. 22 is the part of Fig. 21 eight times enlarged around X2.

[Example 6] : Fig. 23 : The same process of Example 4 (maximize (8)) but with M=50 is applied. The result is considered to be near that of Example 3.

By the way, Fig. 24, 25 and 26 are the parts of Fig. 23, 17 and 20 enlarged near X2 of Fig. 23, respectively. M (=50) is considered to be a 'good' number as a result. No formal optimization is tried to pick up 50 for M. Judging from the results of Fig. 21 and 22 (M=255), Fig. 23 and 24 (M=50), Fig. 20 and 26 (M=0), and Fig. 17 and 25 (averaging), it is expected that there exists an optimum M though it probably depends on pictures.

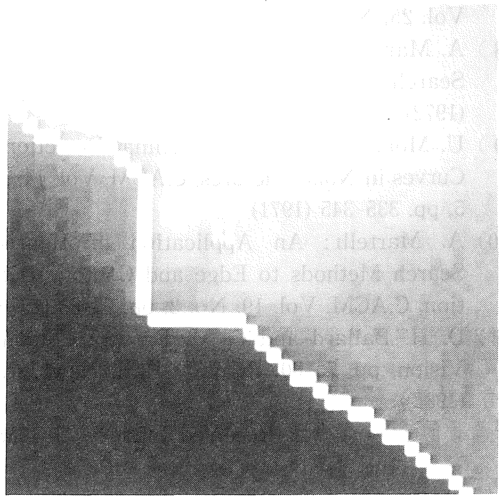


Fig. 25 A part of Fig.17 eight times enlarged around the upper X1.

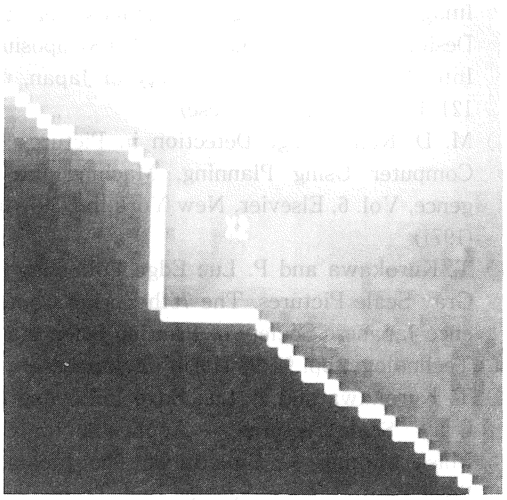


Fig. 26 A part of Fig.20 eight times enlarged (the upper left part of the head).

4. Conclusion

A method which connects two separate points, which are near a region boundary of a digital picture, is proposed. It is an efficient one-pass sequential scanning procedure (algorithm) which extracts a path from one point to the other in such a way that the average of the evaluation value (mainly the edge strength) of the pixels along the path becomes 'near' maximum. The algorithm can be repeatedly used to cut out an entire region of a picture.

Generally, there are two ways, in graph search, to get an optimum boundary path which connects the two points. One is to minimize the cost along the path. The other is to maximize the merit. The former tends to extract a shorter path, the latter a longer one. In this regard, the averaging process of the cost or the merit is introduced for normalization which enables to compare a long path with a short one on a common base. Divisions, however, are required to do it. A large number of divisions may sacrifice the processing speed. A method is suggested to avoid them. It is to add a penalty (a negative value) to the sum of the merit along the path whenever the path gets a unit longer while maximizing the sum.

The features of the algorithm include the following :

1) Since it is a one-pass sequential scanning procedure, the required memory is not too large and the processing speed is relatively high.

2) It extracts exactly one path (a boundary curve segment).

3) The processing speed can be increased by limiting the processing area.

4) A normalization process is introduced so that a long path and a short one can be compared automatically on the common base and a 'better' one would be selected.

5) The processing can be changed in accordance with the boundary complexity of the picture. If the boundary is simple, the speed can be increased.

6) There is a restriction of connecting directions (shortcoming).

References

1) S. Sakane and H. Tamura : An Overview of Image Processing Algorithms : Part 3 — Edge and Line Detection —, Bulletin of the Electrotechnical Laboratory, Vol. 44, No, 7/8, pp. 56-75 (1980) (in Japanese)

2) S. Inokuchi and K. Sato : 'Ejji to Sen no Kenshutu' (possibly translated to 'Edge and Line Detection'), O Plus E, Vol. 80, pp. 88-101 (1986) (in Japanese)

3) Dainippon Screen Mfg. Co., Ltd. : Sigmagraph System 2000 S01 Series Software Manual (1985) (in Japanese)

4) H. Matoba, H. Hirabayashi and Y. Kasahara :

Image Cutting/Collaging Methods for the Design System, Graphics and CAD Symposium Information Processing Society of Japan, pp. 121-128 (1987) (in Japanese)

5) M. D. Kelly: Edge Detection in Pictures by Computer Using Planning, Machine Intelligence, Vol. 6, Elsevier, New York, pp. 397-409 (1971)

6) T, Kurokawa and P. Lu: Edge Following on Gray Scale Pictures, The 76th Spring Conference Japanese Society of Printing Science and Technology, pp. 95-98 (1986) (in Japanese)

7) T. Kurokawa and P. Lu: Edge Following on Gray Scale Pictures — Applicable for Cut Mask Making —, Bulletin of The Japanese Society of Printing Science and Technology, Vol. 25, No. 1, pp. 13-22 (1987) (in Japanese)

8) A. Martelli : Edge Detection Using Heuristic Search Methods, CGIP, Vol. 1, pp. 169-182 (1972)

9) U. Montannari: On the Optimal Detection of Curves in Noisy Pictures, C.ACM, Vol. 14, No. 5, pp. 335-345 (1971)

10) A. Martelli: An Application of Heuristic Search Methods to Edge and Contour Detection, C.ACM, Vol. 19, No. 2, pp. 73-83 (1976)

11) D. H. Ballard and C. M. Brown: Computer Vision, pp. 76-80, Prentice Hall, New Jersey (1982)

(Received January 25. 1988)