

A KNOWLEDGE DIRECTED LINE FINDER FOR ANALYSIS OF COMPLEX SCENES

Masahiko Yachida, Motozo Ikeda and Saburo Tsuji
Department of Control Engineering, Osaka University
Toyonaka, Osaka 560, Japan

As a part of complex scene analysis system, this paper presents a line finder which can find a globally good line with respect to the knowledge available of the line to be sought. A variety of knowledge available of the line to be sought can be communicated to the line finding system simply in the form of descriptions so that the higher level of the program can easily use this line finding system. The optimization method is utilized to detect a globally good line with respect to the given knowledge of the line. In order to reduce the required storage space and computation time in the optimization process, we use the following techniques: 1) represent a line by a sequence of linear segments and find the good sequence of segments by the optimization method and 2) use a locus model of search to prune unpromising lines during the optimization process. Some experimental results applied to various scenes with different amount of noise and knowledge are given in the paper.

1. INTRODUCTION

Boundary detection is an important and critical stage in scene analysis to segment an input picture into meaningful regions. This process usually consists of the following two steps: (1) detection of short edge elements, and (2) connecting these edge elements into boundaries. Many methods have been proposed to detect short edge elements which apply local operations to every point of the input picture. An operator tuned up for characters of edges to be detected is usually used to enhance the edges to be sought in noisy pictures. In order to detect edge elements in noisy pictures, relatively large operators are often used to examine statistic properties of local regions. When edge elements have been found, then they are connected to form boundaries or lines. This line finding process must utilize properties about lines to be detected and must find globally good lines with regards to the properties in order to distinguish meaningful edges in noisy pictures. These properties of the lines to be sought must be separated from the line finding program and given in the form of data to the program, since minor changes of the characteristics of the sought boundary may require a major reprogramming if the properties are embedded in the program[1].

An optimization method is known to be an excellent method to find a globally good line with respect to given properties of the sought line in noisy pictures. Montanari[2] has embedded properties of lines to be detected in a merit function and utilized a dynamic programming method to determine an optimal sequence of points with respect to the given merit function from a number of possible combinations of points. Although the method is good for finding lines in noisy pictures, its drawbacks were the required computation time and storage space, and Martelli[3]

has proposed an more efficient method which finds sub-optimal lines instead of optimal lines using a heuristic search method.

Both of the methods represent a line as a sequence of consecutive points, and need much computation time and storage space if they are applied for high-resolution pictures which are utilized in many applications such as medical images, aerial photographs, etc. Furthermore, shapes that the lines are expected to have, such as 'smoothness', are very important criterion to discriminate lines from noise. However, they need too much computation time and storage space if these shape information is embedded to the merit function because number of combinations of points to be searched expands to a large amount.

In this paper, we propose an efficient search method of lines in noisy pictures which represent a line as a sequence of linear segments instead of a sequence of points and find near-optimal sequence of line-segments with respect to a given figure of merit. We believe that there is little problem to approximate a line by a sequence of linear segments since (1) Small variations on a line is considered as statistically meaningless, (2) Lines we would like to find are smooth ones in many applications, and (3) Since edge elements are determined by statistical distribution of gray levels, it seems natural to determine line segments by statistical distribution of edge elements.

The advantage of the proposed method comparing with the existing methods are as follows:

(a) The required storage space and computation time is much smaller and therefore much more efficient since number of lines among which the optimum is sought is much less than the point-wise search.

(b) It can find lines in noisier pictures since shape information of the line we want to detect can be easily embedded in the figure of merit.

In order to further decrease the required storage space and computation time, we utilize the locus model of search which has been proposed by Rubin et al[4] in image interpretation task. We will utilize this search method for finding lines where only a beam of near-miss alternatives around the best line are extended to determine a near-optimal line.

For complex scenes containing many kinds of objects, analysis is usually performed in the following steps[5]: 1) detect distinct lines as the initial information that can be easily found in the scene, 2) select models of objects which contain these extracted lines and propose the lines to be examined utilizing the models of objects and their contextual relations, and 3) examine the proposed lines using the knowledge of the sought lines given by the above higher level of program (or scene analyzer) for final interpretation of the scene.

In such a scene analysis system, a line finder which can find a globally good line with respect to the knowledge given by the scene analyzer is necessary. The knowledge available on the line to be sought is not only properties of the line which can be embedded in the merit function, but also a variety of knowledge such as expected location, when to terminate the line following, etc. All of these variety of knowledge available of the line to be sought must be easily expressed by and communicated to the line finder. We will describe a line finding system which can find a good line with respect to the knowledge given in the form of descriptions by the scene analyzer.

2. FINDING SMOOTH LINES IN NOISY PICTURES

2.1 MULTISTAGE OPTIMIZATION PROCEDURE

Let us first briefly review the multistage optimization procedure. In general, an optimization problem can be formulated as follows: given a merit function $f(x_1, x_2, \dots, x_N)$, where $0 \leq x_i \leq n_i$, $i=1, \dots, N$, find (a) the maximal value M of the merit function f , and (b) a value $(X_1, X_2, \dots, X_N)$ for which the maximum is achieved. If the merit function f can be expressed by a sum of terms, each of which depends on only a few variables, then a multistage optimization procedure applies. For example, if f has a simple form as

$$f(x_1, \dots, x_N) = f_1(x_1, x_2) + f_2(x_2, x_3) + \dots + f_{N-1}(x_{N-1}, x_N) \quad (1)$$

then the following recursion formula can be used:

$$h_1(x_1) = 0$$

$$h_{k+1}(x_{k+1}) = \max_{0 \leq x_k \leq n_k} (f_k(x_k, x_{k+1}) + h_k(x_k)) \quad (2)$$

where $k=1, \dots, N-1$ and $0 \leq x_{k+1} \leq n_{k+1}$. At each stage, the values $h_{k+1}(x_{k+1})$, $0 \leq x_{k+1} \leq n_{k+1}$, and the

values $m_{k+1}(x_{k+1})$ of x_k for which the maximum is achieved must be saved in a table with n_{k+1} entries. At the end of this process, maximum value M of f can be obtained as

$$M = \max_{x_N} h_N(x_N) \quad (3)$$

The values $(X_1, \dots, X_N)$ for which the maximum is achieved are obtained by tracing back the tables with the recursion formula:

$$X_N = m_N = \arg \max_{x_N} h_N(x_N)$$

$$X_k = m_{k+1}(X_{k+1}) \quad k=N-1, \dots, 1 \quad (4)$$

2.2 REPRESENTATION OF A LINE BY POINTS

The existing methods to find a line in noisy pictures, represent a line as any sequence of points P_i , $k=1, \dots, N-1$, are neighbors. The properties of the sought line are embedded in a merit function f and an optimal line is searched with respect to a given merit function using the optimization method. Let us first consider a simple problem of finding a low-curvature line having high gray values. Then, a good merit function will be the sum of gray levels along the line, minus the sum of the curvature at every point. If we denote the coordinate vectors of a point P_i by $z_i = (x_i, y_i)$ and slope of the line by $\theta(z_i, z_{i+1})$, then a curvature of a point P_i defined by three points can be given as

$$\theta(z_{i-1}, z_i) - \theta(z_i, z_{i+1}).$$

Then, a merit function is given as

$$f(z_1, \dots, z_N) = \sum_{i=1}^N g(z_i) - k \sum_{i=2}^{N-1} |\theta(z_{i-1}, z_i) - \theta(z_i, z_{i+1})| \quad (5)$$

where $g(z_i)$ are gray level of a point P_i . This is a merit function of the form

$$f(z_1, \dots, z_N) = f_1(z_1, z_2, z_3) + \dots + f_{N-2}(z_{N-2}, z_{N-1}, z_N) \quad (6)$$

and can be solved by the multistage optimization procedure.

However, this measure of curvature defined at such a small length causes serious erroneous results. Fig. 1 shows a simple example of such cases where the problem is to find a smooth line connecting A and F. In the figure, values of points marked as $\circ$ are one and the others are zero. The optimization method with the merit function defined in (5) finds a path A-B-C-D-F as the optimal solution, because sum of curvatures of this path is smaller than another path. However, we perceive another path A-B-E-D-F as more meaningful or good-shaped and the path B-C-D as the line caused by noise. This error has been caused by the very local measure of curvature defined only by three consecutive points. To avoid this problem, more global properties of the line defined over a wider range should be embedded into the merit function.

If the property of the line defined by a wider range, say k points, are embedded into the merit function, it becomes the form of

$$f(z_1) = f_1(z_1, \dots, z_k) + f_2(z_2, \dots, z_{k+1}) + \dots + f_i(z_i, \dots, z_{i+k-1}) + \dots \quad (7)$$

where all terms are functions of k variables. This is called a merit function of type k , and it is known that the required tables and computation time are exponential with k to solve the problem with multistage optimization procedure.

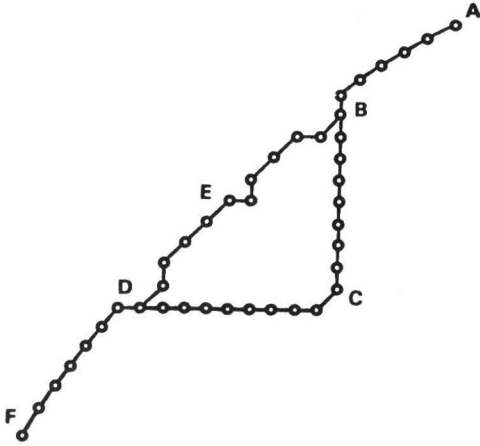


Fig. 1. An example in which curvature defined at a small length causes erroneous result.

2.3 REPRESENTATION BY LINEAR SEGMENTS

We have studied a different approach to avoid this difficulty, in which a line is represented by a sequence of connected linear segments $L_1, L_2, \dots, L_N$ of length ℓ instead of consecutive points, as shown in Fig. 2. Let us represent a linear segment L by an ordered pair of their endpoints (z_1, z_2) and call z_1 the head and z_2 the tail of the segment. Then two linear segments L and L' are called connected if the tail of L coincides with the head of L' . If we call a connected linear segment of a segment L a successor of L , then each segment of length ℓ has $2\pi\ell$ possible successors as shown in Fig. 3. However, since the line we want to detect are smooth ones, successors having curvatures less than a certain threshold α are considered as 'legal' successors. That is, if $\theta(z_1, z_2)$ denotes the direction of the segment $L(z_1, z_2)$, then segments $L'(z'_1, z'_2)$ having

$$|\theta'(z'_1, z'_2) - \theta(z_1, z_2)| \leq \alpha \quad (8)$$

are legal successors as shown in Fig. 3.

With this representation, a merit function to find a low-curvature curve having high gray levels in which the curvature is defined by the length of 2ℓ points can be given as

$$f(z_1, \dots, z_N) = \sum_{i=2}^N g(z_{i-1}, z_i) - k \sum_{i=2}^{N-1} |\theta(z_{i-1}, z_i) - \theta(z_i, z_{i+1})| \quad (9)$$

with the constraint defined in (8), where $g(z_{i-1}, z_i)$ is a mean of gray values of the points lying on the line $L_{i-1}(z_{i-1}, z_i)$. It will be noted that the problem of type 2 ℓ if the curve is represented by a sequence of points is reduced to the problem of type 3 with this representation. Number of stages necessary to find a curve of N points is also reduced from N stages with pointwise search to N/ℓ stages with this linewise search. Next, we will describe an algorithm to find optimal sequence of linear segments with respect to a given merit function.

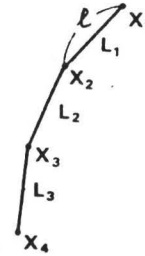
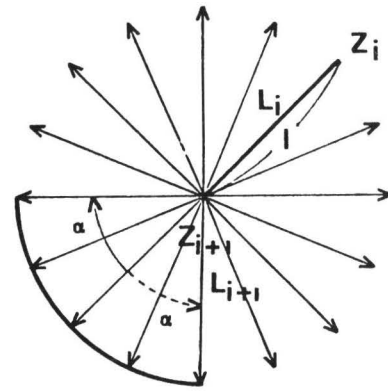


Fig. 2. Representation of a line by a sequence of linear segments $L_1, L_2, \dots, L_N$.



Legal successors

Fig. 3. Legal successors of line L .

2.4 FINDING THE OPTIMAL SEQUENCE

For the sake of simplicity, let us first consider the problem of extracting from a picture a curve which starts in the first row and ends in the last row. First of all, for each admissible z_1 , successors z_2 are determined with a constraint $\pi \leq \theta(z_1, z_2) \leq 2\pi$. Then, at the k -th stage, successors z_{k+1} are expanded

for each admissible z_k with the constraint

$$\theta(z_{k-1}, z_k) - \theta(z_k, z_{k+1}) \leq \alpha$$

where $k=2, \dots, N-1$. For every value of z_k and z_{k+1} , we compute

$$h_k(z_k, z_{k+1}) = H_{k-1}(z_{k+1}, z_k) + g(z_k, z_{k+1})$$

$$+ k |\theta(z_{k-1}, z_k) - \theta(z_k, z_{k+1})| \quad (10)$$

where $h_1(z_1, z_2) = g(z_1, z_2)$ and H_{k-1} is the value of the output of the previous stages for all admissible z_{k-2} . Then we obtain the maximal value of h_k , H_k , with respect to z_{k-1} and the value of z_{k-1} , Z_{k-1} , for which the maximum is achieved. These values, H_k and Z_{k-1} , are stored in the table for every value of z_k and z_{k+1} at each stage.

This process terminates at the N -th stage when some of the segments (z_{N-1}, z_N) intersect with the last column. At this stage, the tail of each segment, z_N , which intersects with the last column is determined as the intersecting point of the segment (z_{N-1}, z_N) and the last column. For every value of z_{N-1} and z_N , we compute h_{N-1} defined in (10) and obtain the maximal value of $h_{N-1}(z_{N-1}, z_N)$, $H_{N-1}(z_{N-1}, z_N)$, and the value of z_{N-2} , Z_{N-2} , for which the maximum is achieved. Then, for all admissible z_{N-1} , we compute the maximum of $H_{N-1}(z_{N-1}, z_N)$ with respect to z_{N-1} and the value of z_{N-1} , Z_{N-1} , for which the maximum is achieved. Finally, we obtain the maximal value of $H_N(z_{N-1}, z_N)$, $M(H_N)$, and a value of z_N , Z_N , for which the maximum is achieved. Then, tracing back the tables with the recursion formula defined in (4), we obtain $Z_N, Z_{N-1}, \dots, Z_1$ for which the maximum has been achieved.

2.5 LOCUS MODEL OF SEARCH

In order to decrease storage space and computation time, we introduce the locus model of search to this line finding system. In the multistage optimization process, all the paths generated during the optimization process are saved and evaluated until the last stage. However, it is usually not necessary to find the optimal path but it is sufficient to find good or near optimal solutions in practical applications. Therefore, we prune unpromising paths from the search tree at each stage. That is, after computing H_k at each stage, those paths having H_k less than a certain threshold are not saved in the table; and therefore, their successors are not generated at the subsequent stages. Let us denote the maximum value of H_k at the k -th stage as $M(H_k)$, then the threshold is set as $\tau * M(H_k)$, $0 < \tau < 1$. This search method suppresses exponential growth of successors and only a beam of near-miss alternatives around the best path are retained as shown in Fig. 4.

3. FINDING A LINE WITH GIVEN KNOWLEDGE

In this section, we will describe a line finding system which finds a globally good line with respect to the knowledge of the line given by the scene analyzer. A

variety of knowledge is available on the line to be sought. These various knowledge of the sought line must be easily expressed by and communicated to the line finder. The line finder consists of three components; a menu, a line finding program and an interpreter as shown in Fig. 5. The menu is a list of knowledge which can be utilized by the optimization process. Then the properties of the sought line can be expressed simply by descriptions which specify which items of the menu should be used in finding the line. The descriptions given by the scene analyzer are interpreted and embedded into the line finding program by the interpreter.

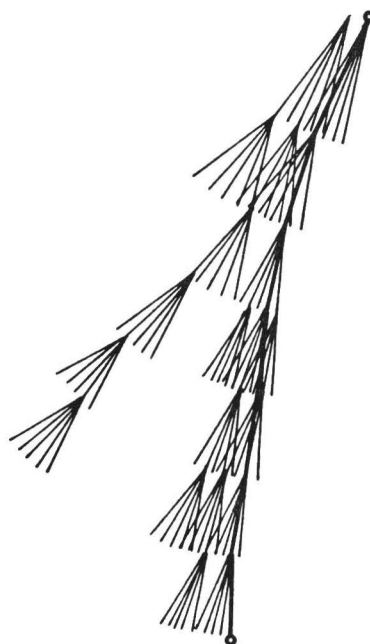


Fig. 4. A locus model of search suppresses exponential growth of successors.

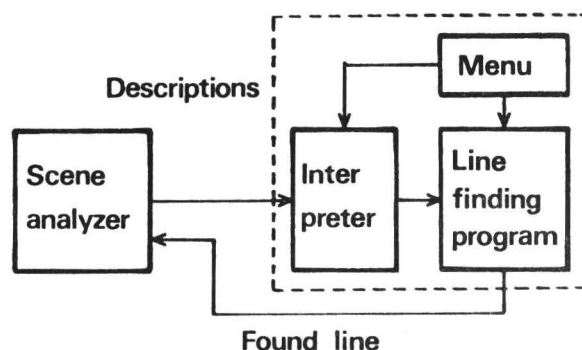


Fig. 5. Line finding system.

3.1 MENU AND INTERPRETER

The menu consists of properties of the line, some constraints on the line, constraints on the starting point, and terminating conditions, each of which is described below.

3.1.1 PROPERTIES OF THE LINE

The menu for the properties of the line is listed in table I. The properties of the sought line are specified by the format (item, weight, parameter). For example, if we would like to find the low-curvature line of homogeneous gray level showing strong edgeness, then descriptions are given to the line finder as:
(HOMGRY k_1),(EDGE k_2),(LOWCURV k_3). Then, these properties on the line are embedded into the merit function as follows:

$$f = k_1 f_1 + k_2 f_2 + k_3 f_3$$

3.1.2 SOME CONSTRAINTS ON THE LINE

Constraints on the line which are difficult to embed in the merit function are given to the line finder as follows.

Search region: the expected location of the sought line is known using the relationship with the already found lines. This knowledge is given to the line finder as the search boundary of the line. The search boundary is given to the line finder by a sequence of representative points as (SEARCH (x_1, y_1), (x_2, y_2), ...). If the search boundary is given, a line is sought in the region bounded by the search boundary as shown in Fig. 6. This information greatly reduces the computation time and storage space, and increases reliability of the extracted line.

Length of linear segment ℓ : we have four values for the length ℓ : 5, 10, 20, 30, and one of them can be specified considering the maximal curvature of the line to be sought. If it is not specified, length 20 is utilized.

Constraint on the successors α : the parameter α determines number of successors generated and is given considering the maximal curvature of the line to be sought. If it is not specified, 1 radian is used for α .

Threshold t : the threshold t determines the width of the beam, that is a number of lines saved in the table at each stage. If t is not given, 0.5 is used as t .

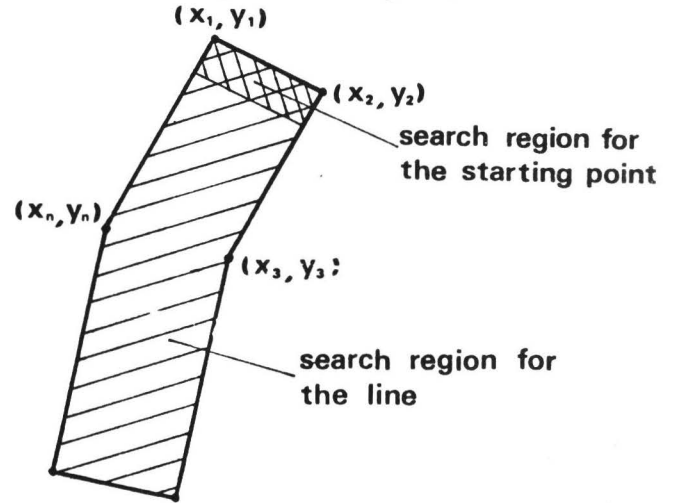


Fig. 6. Search region for the line and for the starting point.

Table I. Properties of the line

EDGE	Find a line having strong edgeness	$\sum e(z_k, z_{k+1})$
GVNGRY	Find a line having gray level G	$\sum g(z_k, z_{k+1}) - G$
HOMGRY	Find a line having homogeneous gray levels	$\sum g(z_{k-1}, z_k) - g(z_k, z_{k+1})$
GVNDIR	Find a line having edge direction D	$\sum d(z_k, z_{k-1}) - D$
HOMDIR	Find a line having homogeneous edge direction	$\sum d(z_{k-1}, z_k) - (z_k, z_{k+1})$
GVNSLP	Find a line having slope θ	$\sum \theta(z_k, z_{k+1}) - \theta$
LOWCURV	Find a low-curvature line	$\sum \theta(z_{k-1}, z_k) - \theta(z_k, z_{k+1})$
CONCURV	Find a constant-curvature line such as a circle	$\sum (\theta(z_{k-2}, z_{k-1}) - \theta(z_{k-1}, z_k)) - (\theta(z_{k-1}, z_k) - \theta(z_k, z_{k+1}))$
GVNCURV	Find a line having the curvature C	$\sum \theta(z_{k-1}, z_k) - \theta(z_k, z_{k+1}) - C$

$E(z_k, z_{k+1})$; mean value of edgeness of the segment (z_k, z_{k+1})

$G(z_k, z_{k+1})$; mean value of gray level of the segment (z_k, z_{k+1})

$D(z_k, z_{k+1})$; mean value of edge direction of the segment (z_k, z_{k+1})

$\theta(z_k, z_{k+1})$; slope of the segment (z_k, z_{k+1})

3.1.3 CONSTRAINTS ON THE STARTING POINTS

Selection of the starting points of the line is very important to reduce growth of successors in subsequent stages. We have two constraints on selecting the starting point: 1) search boundary for the starting points and 2) properties of the starting point. In many cases, more restricted search boundary is available for the starting point than for the sought line. This search boundary is given to the line finder by a sequence of representative points as

$$(SSERCH(x_1, y_1), (x_2, y_2), \dots).$$

If a search boundary is given, the starting point is sought in the region bounded by the search boundary. If it is not given, the search boundary for the line is utilized.

Properties of the starting point listed on the menu are gray level, edgeness and edge direction. The properties on the starting point is given in the similar manner to the line. For example, if we would like to find the points having strong edgeness and edge direction D with weights k_1 and k_2 respectively, then the following descriptions are given:

$$(SEGE\ k_1), (SDIREC\ k_2\ D).$$

Then the following merit function is generated by the interpreter:

$$f = k_1 * \varepsilon(z) + k_2 * |d(z) - D|$$

Then, the promising points are selected with respect to the merit function for the starting point in the search region.

3.1.4 TERMINATING CONDITIONS

There are four terminating conditions in line following as follows.

- 1) Line segments of the current stage intersect with line segments of the first stage. This terminating condition is used when the sought line is closed.
- 2) Line segments of the current stage intersect with the other lines which are already found in the picture.
- 3) Line segments of the current stage intersect with the line L specified as the terminating one.
- 4) There is no promising line segment at the current stage. That is, score of the best line segment at the current stage M_k is much smaller than the score of the best line segment of the last stage M_{k-1} , such as $M_k \leq \delta M_{k-1}$ where $\delta < 1$.

If we would like to use the terminating conditions, for example 1 and 4, then the descriptions are given to the line finder as follows:

$$(TERM\ 1), (TERM\ 4\ \delta),$$

where δ is a threshold.

3.2 LINE FINDING PROGRAM

When the descriptions on the sought curve are given, these are interpreted and embedded into the line

finding program by the interpreter as described before. Then the line finding program seeks a globally good line with respect to the given knowledge as follows.

1) Compute the value of merit function $f(z_1)$ for every point z_1 in the search region for the starting point, and select promising starting points. Let us denote the maximum of $f(z_1)$ as $M(f(z_1))$, then those points having value of $f(z_1)$ larger than $T * M(f(z_1))$ are selected as promising starting points.

2) At the first stage, determine successors z_2 for each selected points z_1 and select promising ones among z_2 . For each selected point z_1 , its successors z_2 are determined with the constraint that z_2 must be in the search region for the line, and the merit function $f(z_1, z_2)$ are computed. Then those segments having $f(z_1, z_2)$ larger than $T * M(f(z_1, z_2))$ are selected as promising ones. If the search region for the starting point is given around the middle of the sought line, then two beams are obtained as shown in Fig. 7. In this case, one of them is followed first.

3) At the k -th stage, determine the successors z_{k+1} for each z_k and select promising ones among z_{k+1} . For each z_k , its successors are determined with constraints that z_{k+1} must be in the search region and $\theta(z_{k-1}, z_k) - \theta(z_k, z_{k+1}) \leq \alpha$. If the given merit function is type i ($2 \leq i \leq 4$), then we compute

$$h_k(z_k, z_{k+1}) = H_{k-1}(z_{k-1}, z_k) + f_k(z_{k-i+2}, \dots, z_{k+1}) \quad (11)$$

for every value of $z_{k-i+2}, \dots, z_{k+1}$. We obtain the maximal value of h_k , H_k , with respect to z_{k-i+2} and the value of z_{k-i+2}, z_{k-i+2} , for which the maximum is achieved, for every value of $z_{k-i+3}, \dots, z_{k+1}$. Let us denote the maximum of H_k as $M(H_k)$. Then, those points z_{k+1} having H_k larger than $T * M(H_k)$ are selected as promising points. The value of H_k and z_{k-i+2} of the promising points z_k are saved in the table.

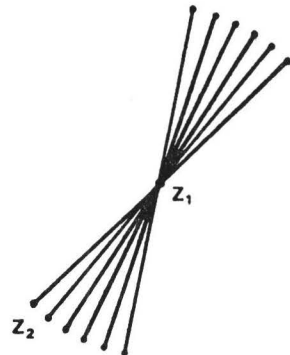


Fig. 7. Two beams are obtained if the starting point is found around the middle of the line.

4) Check the terminating conditions. If one of the terminating conditions specified by the descriptions is satisfied, then the line following is terminated. At this last stage, the tail of each segment z_N is determined as the intersecting point of the segment (z_{N-1}, z_N) and the line specified by the terminating conditions.

5) Try to follow the other end of the line using the above-mentioned procedure, with (z_2, z_1) as the starting line segments. When both sides terminate, then obtain the maximal value of the merit function, and a sequence of line segments for which the maximum is achieved by tracing back the tables.

4. EXPERIMENTAL RESULTS

The proposed line finding system was implemented in Fortran IV on a HP-2108 computer and was utilized for analysis of heart walls, blood vessels and complex industrial parts. We show a result on the test image shown in Fig. 8(a) by adding different amount of noise to the image. The image consists of 256 by 256 points, each point having 16 levels of gray. Our purpose is to detect smooth line 1 in the presence of noisy lines 2 and 3. In the image, the gray levels of the lines 1, 2 and 3 are 10, 11 and 12 respectively and the gray level of the background is 5. In Fig. 8 (b) and (c), independent amount of noise is added to every point. The statistical distribution of the noise at every point is normal, with mean value zero and standard deviation 4 and 6, respectively. Note that the line is recognizable in Fig. 8(b) but quite unrecognizable in Fig. 8(c).

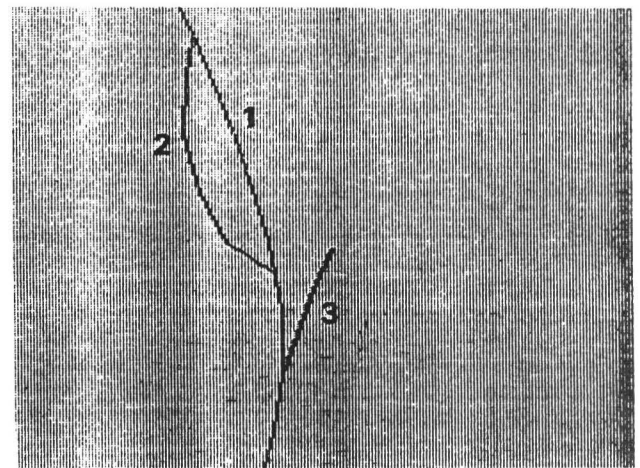
The knowledge available of the curve is smooth, has large value of gray, starts in the first row and ends in the last row. Therefore, the descriptions given to the line finding system are as follows:

(EDGE 1), (LOWCURV 5), (SSERCH (1,1),(256,1))
(TERM 3 (1,256),(256,256))

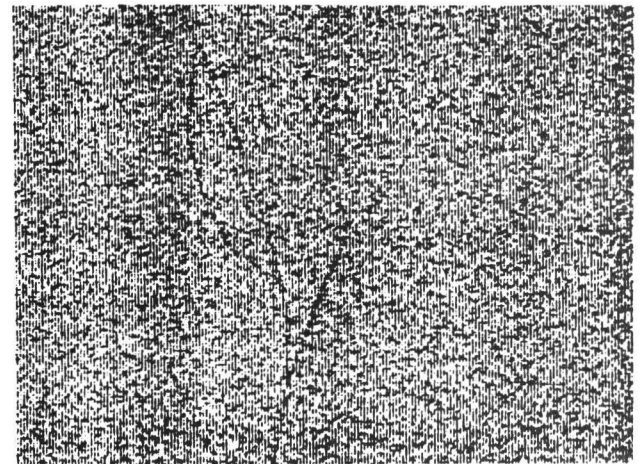
Fig. 9(a) and (b) show the results for the images of Fig. 8(b) and (c). It will be noted that the line is detected correctly even in the presence of heavy noise. If the second description is eliminated, that is the low-curvature is not utilized, then the pass 2 is selected as shown in Fig. 9(c) which is the result for the image of Fig. 8(b).

5. CONCLUSION

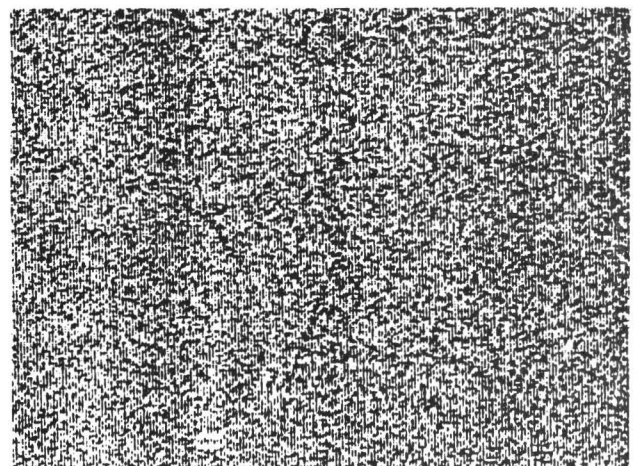
We have presented a line finding system that can detect a globally good line with respect to the given knowledge of the line. The proposed method is very efficient since 1) size of tables required at each stage and number of stages is greatly reduced by representing a line by a sequence of linear segments, 2) a locus model of search is utilized to suppress the exponential growth of search trees, and 3) several constraints available of the sought line are utilized to limit search space or growth of successors, such as search boundary, constraints on the starting point, etc.



(a)



(b)



(c)

Fig. 8. Input pictures with different amount of noise.

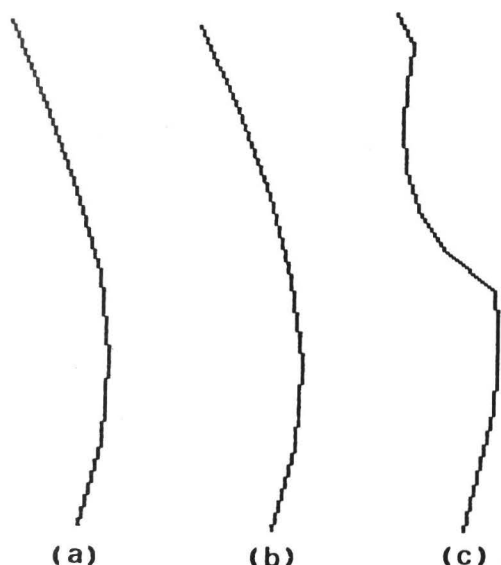


Fig. 9. Results obtained (a) for Fig. 8(b); (b) for Fig. 8(c); and (c) for Fig. 8 (b) without the knowledge of shape.

The length of linear segment ℓ seriously affects computational efficiency. Therefore, if the sought curve is smooth and can be approximated by long linear segments, then large value should be given to ℓ . On the contrary, if small variations on the line is important or the sought line has high curvature, then small value must be given to ℓ . However, it is more efficient that we first obtain an approximate line with large ℓ and then obtain more details with small ℓ in the neighborhoods of the approximated line than obtaining the line with small ℓ from the beginning.

We have used a locus model of search instead of the best first search of A* algorithm for obtaining sub-optimal solution. The reason of this selection is that its control structure is simpler than the A* algorithm and it is easier to understand the behavior of search process since the locus model does not need backtracking. Another reason which is more important is that it permits parallel computation of all the linear segments at every stage.

In this system, it is very easy to communicate the knowledge of the sought line to the line finder since a variety of knowledge can be expressed simply in the form of descriptions. The menu provided in the system is sufficient for our current tasks. However, it is easy to change the menu if it is necessary.

We have applied the proposed method for a variety of scene analysis tasks and the results have shown the effectiveness of the proposed method to find a globally good line with respect to the given knowledge. However, for complex scenes where many kinds of objects exist and a priori knowledge

on the scene is not available, we must first extract distinct lines as initial information on the scene to select possible models of objects in the scene. For this purpose of extracting initial information, there exist several methods such as planning[6], detection of distinct edges[7] and extended Hough transformation[8]. The proposed line finder is then used to find subtle lines using knowledge of the objects and their contextual relations.

REFERENCE

- [1] A. Martelli : *An Application of Heuristic Search Methods to Edge and Contour Detection*, Commu. ACM 19, pp. 73-83, 1976.
- [2] U. Montanari : *On the Optimum Detection of Curves in Noisy Pictures*, Commu. ACM 14, pp. 335-345, 1971.
- [3] A. Martelli : *Edge Detection Using Heuristic Search Methods*, Comput. Grap. Image Proc. 1, pp. 169-182, 1972.
- [4] S. M. Rubin and R. Reddy : *The Locus Model of Search and Its Use in Image Interpretation*, Proc. 5th IJCAI, pp. 590-595, 1977.
- [5] M. Yachida and S. Tsuji : *A Versatile Machine Vision System for Complex Industrial Parts*, IEEE Trans. C-26, pp. 882-894, 1977.
- [6] M. D. Kelly : *Edge Detection in Pictures by Computer Using Planning*, in Machine Intelligence, 6, pp. 379-409, 1971.
- [7] A. K. Griffith : *Edge Detection in Simple Scenes Using a Priori Information*, IEEE Trans. C-22, pp. 371-381, 1973.
- [8] S. Tsuji and F. Matsumoto : *Detection of Ellipses by a Modified Hough Transformation*, IEEE Trans. C-27, pp. 777-781, 1978.