DETECTION OF ELLIPTIC AND LINEAR EDGES BY SEARCHING TWO PARAMETER SPACES

Saburo Tsuji and Fumio Matsumoto
Department of Control Engineering
Osaka University
Toyonaka, Osaka 560, Japan

ABSTRACT

The Hough transformation can detect straight lines in an edge-enhanced picture; however, its extension to recover ellipses requires too much computing time. This paper proposes a modified method which utilizes two properties of an ellipse in such a way that it iteratively searches for clusters in two different parameter spaces to find visible ellipses, then evaluates their parameters by the least means squares method.

INTRODUCTION

Analysis of a scene containing many complicated-shaped objects is a current problem of computer vision. Since the usual line finders fail to extract a reliable line drawing from the scene, Falk and Griffith have proposed methods which identify each object in a blocks world by utilizing input data and external constraints so as to suggest and test hypotheses on the scene [Falk 72], [Griffith 73]. In order for a computer vision system to analyze more complex pictures containing real objects such as telephones, cups, or industrial parts, their heuristics have been modified as follows: (1) search for simple familiar patterns such as polygons and ellipses in strong feature points in the pictures, (2) select models of objects whose features contain these patterns, (3) test the validity of the models by examining whether weak feature points around the patterns satisfy the proposed models [Tsuji 75].

Line finding methods utilizing the Hough transformation are useful to find polygons, especially partly occuluded ones, because they utilize global properties of edge points [Hough 62],[Duda 72],[Griffith 73],[O'Gorman 76]. Duda and Hart also discussed possibilities of extending their methods to find curves in the picture [Duda 72], and Shapiro analyzed the performance of the transformation method to detect curves in noisy pictures [Shapiro 75]. Direct application of the parameterization, however, is limited to curves with a small number of parameters, say two or three at most, since necessary memory space and computing time grow exponentially with the number of parameters. For example, detection of circles requires a three-dimensional array of accumulators, and a modification of the algorithm is necessary to recover the circular arcs in a reasonable time [Kimme 75].

This work was supported in part by Grand-In-Aid for Scientific Research from the Ministry of Education, Japanese Government.

This paper describes how we can modify the paramerization technique so as to recover both linear and elliptic edges, which are important cues for analyzing scenes containing artificially constructed objects. We can easily detect and erase long straight lines in an edge-enhanced picture; then the problem is how to efficiently detect ellipses (each of which has five unknown parameters) in a set of feature points on curved or short linear segments. Instead of utilizing a five-dimensional array of accumulators, we sequentially search for clusters in two different parameter spaces; one for finding the approximate positions of centers of ellipses and selecting candidate feature points for an easily detectable ellipse, and the other for testing whether the candidates are exactly on the ellipses or not. There exist interactions between different patterns when one maps edge points into the parameter spaces. As the result, ellipses with many edge elements sometimes mask weak clusters corresponding to other ellipses. Therefore, the computer vision system iteratively searches for an easily visible ellipse in a set of feature points in an edge-enhanced picture from which edge points of the recovered long straight lines and ellipses have been erased.

DETECTION OF ELLIPSES

The direct application of the Hough transformation to the detection of elliptic objects in a digitized picture requires a five-dimensional array of accumulators; the array is indexed by five parameters specifying the location, shape, and orientation of an ellipse. Its usuage, however, is impractical because of excessing computing time. An idea for overcoming the difficulty is as follows: instead of the time consuming process of mapping each edge point in the picture into the five-dimensional parameter space, we sequentially examine clusters in both a two-dimensional and a one-dimensional space, by utilizing two well known properties of an ellipse, and efficiently select a small number of candidate edge points for an easily visible ellipse. Next, the five unknown parameters of the best fitting ellipse to the candidates are evaluated by the least mean squares method.

Erasing Long Straight Lines in Edge-Enhanced Picture

We preprocess a digitized input picture to find edge points by applying a simple gradient operator to every point in the picture and thresholding it, and we thus obtain an edge-enhanced picture *E* in which each edge element is characterized by its location (x,y) and quantized direction 9 which ranges between 0° and 360° (the distinctness of the edge point is not used in the following process). Next, the edge points are arranged in a set *{e }* of edge lists: the edge list e_θ is a collection of edge points of a direction θ. Now the line finding method [0'Gorman 76] detects sets of collinear edge points in *E* and registers them in a list of straight lines, however some of them may belong to ellipses. Straight lines longer than a threshold are considered not to be parts of the elliptic arcs, and their edge points are erased from *E* and *{e }* in order to simplify the following algorithm for detecting ellipses.

Detection of Centers of Ellipses

Let us consider two parallel tangents at P and Q to an ellipse (see Fig.I (a)). *A* simple property, namely that P and Q are at equal distance from the center 0 of the ellipse, is useful for finding locations of ellipses in *E* by a two-dimensional accumulator array *{a* }; the array is indexed two parameters x,y (x,y) specifying the location of an ellipse. For each pair of edge points (x,,y-) and (x_2,y_2) in an edge list *e* (a collection of edge points having an orientation e), an accumulator at ((x,+x_2)/2,(y +y)/2) is incremented by one. After^all pairs of the set {e_Q> are processed in this way, the array is locally averaged by using a 3x3 neighborhood. Thus, the accumulator corresponding to the center of a complete ellipse has a count approximately proportional to the length of its circumference.

Now, we search for the accumulator *a* with the highest count, whose indices specify the location of the center of an easily visible ellipse (or concentric ellipses), and then select all pairs of edge elements which increased *a* in the above-mentioned process, as *m,n* candidates for the ellipse. The process of erasing long straight lines is necessary before applying this center-finding algorithm because two parallel lines generate a mountain ridge, considerably dis-

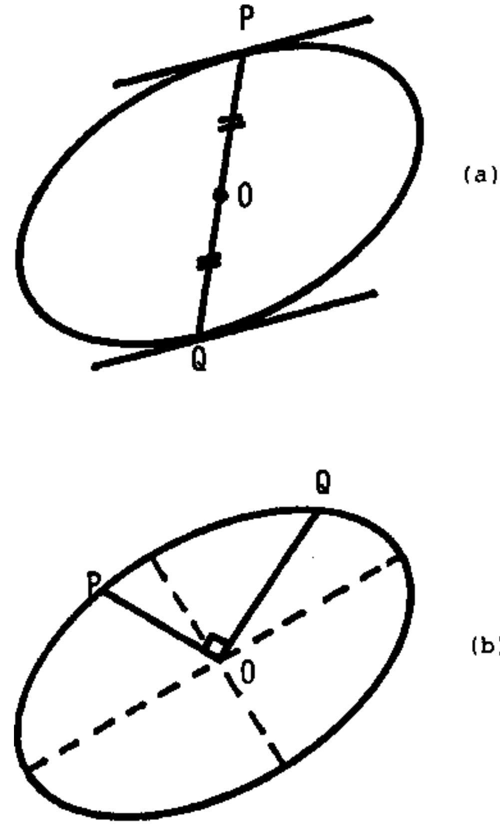


Fig.I Two properties of an ellipse.
(a) OP=OQ, if two tangents are parallel.
(b) $(I/OP^2)+(I/OQ^2)=I/R^2$=constant, if /POQ=90°

turbing the peak-finding process, in the array.

Testing Candidates for the Ellipse and Evaluation of Its Parameters

Let *C* be the set of the selected candidates for the ellipse. The above-mentioned center-finding procedure simply collects edge points such that they are on symmetrical curved or linear segments to the point *(m,n)* in *E;* therefore a member in *C* is not always on an ellipse. Application of the least mean squares method to fitting an ellipse to *C* is likely to give an unsatisfiable result, especially when there exist concentric ellipses, because the fitting process is significantly disturbed by the symmetrical patterns not on one ellipse. Thus we must select from the candidates the edge points which lie exactly on an ellipse by utilizing a property of an ellipse, and then apply the fitting process for evaluating accurate parameters.

Consider two points P and Q on an ellipse such that /POQ=90°, where 0 is the center of the ellipse (see Fig,1(b)). It is easy to prove the property that

$$\frac{1}{OP^2} + \frac{1}{OQ^2} = \frac{1}{R^2}$$
$$= \text{constant} \qquad (1)$$

for an ellipse.

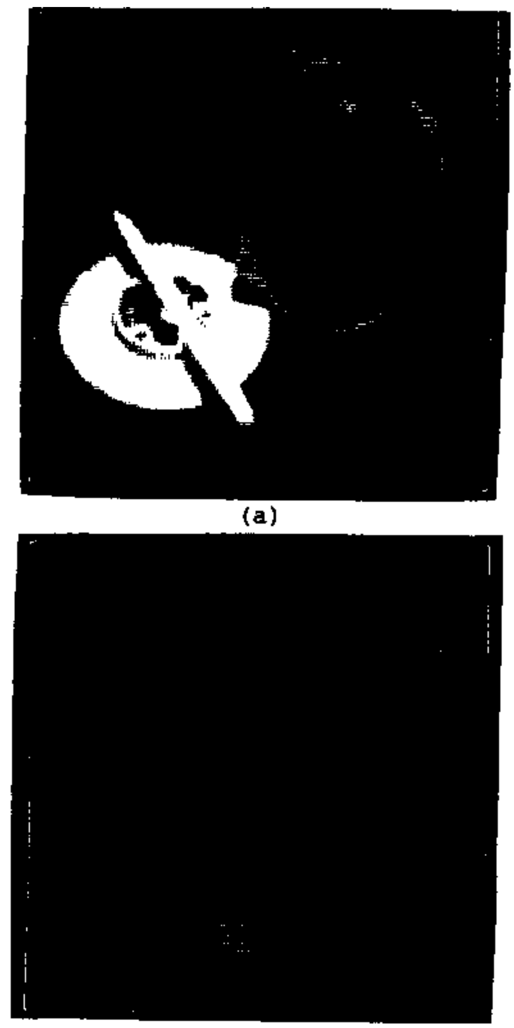
Now, a one-dimensional accumulator array { *a* . } is used to test the candidates. ' We assume that 0 is located at (m,n) obtained by the center-finding algorithm. If any two points P and Q in *C* satisfy /POQ=90°+ 6 (6 is a threshold value), then the" accumulator corresponding to R evaluated by (1) is incremented by one. After processing all members in *C* in this way, we obtain a histogram of R, which gives us valuable information on ellipses in *C* ; multiple prominent peaks in it suggest the existence of concentric ellipses in *C* (however the number of the ellipses is less than that of the peaks in most cases), or we would decide that there is no visible ellipse in *C* , if the histogram contains only low hills.

Mutual interactions between edge points on the concentric ellipses generate significant false peaks in the histogram, sometimes higher than the peaks corresponding to true ellipses. The smallest ellipse in *C* is examined first, because the count in the accumulator corresponding to it is less sensitive to this interference. We select all pairs of edge points, contributing to the leftmost peak in the histogram, and then evaluate five parameters of the best fitting ellipse to these edge points by the least mean squares method. The fitting process is judged to be a failure if the edge points in *E* cover only a small fraction of this ellipse. Otherwise, it is considered a success, and the edge points on the recovered ellipse are excluded from *E, C* and *{e*) in order to eliminate their interaction with other ellipses. The candidate-testing process is iterated by calculating a new histogam of R in updated *C* and examining the leftmost peak in it, until the histogram does not contain any prominent peak.

The above-mentioned procedure for testing candidates is iteratively applied to all prominent clusters in the two-dimensional accumulator array to recover all visible ellipses in the picture. Finally, we test whether each short line in the list of straight lines belongs to the recovered ellipses or not. Since the edge points on these ellipses have been excluded from the updated edge-enhanced picture *E* , we discard the line from the list if the edge points still remaining in *E* are not or sparsely distributed on the line; otherwise it qualifies for a straight line.

EXPERIMENTAL RESULTS

The following example shows some of the features of the ellipse-finding algorithm. Fig.2 (a) shows a 128x128 digitized picture with 64 gray levels, which contains three parts of a gasoline engine; two cylindrical parts and a rod. A simple gradient operator using a 3x3 window [0'Gorman 76] with a noise threshold of 30 is applied to obtain an edge-enhanced picture. Fig.2 (b) shows the result, which contains 2643 edge points.



(a)

(b)

Fig.2 Input picture.
(a) A 128 by 128 digitized picture, containing three industrial parts.
(b) An edge-enhanced picture of (a).

Setting the quantization step of 8 at 6°, we classify the edge points into 30 groups (edge lists). At this point, the line finding algorithm is applied, and all linear segments are detected.

Now let us consider how we can select the threshold for discriminating long straight lines from flat portions of ellipses. The maximum length L of a flat portion of an ellipse described by $(x/a)^2 + (y/b)^2 = 1$ is

$$L_m = 2(aVb)(\pi/180°)(6°/2)$$
$$= (a/b)(a/10) \qquad (2)$$

Considering the size of the picture is 128 by 128, we expect a<50. Therefore, not highly concentric ellipses (a<b) do not include linear segments longer than 20. Thus we register longer segments than 20 in the list of straight lines, and erase their edge points. The edge list and *E* now contain 1993 edge elements.

Next, the ellipse-finding algorithm is applied. After all pairs of the edge points in every edge list are mapped into the two-dimensional (128x128) array of accumulators, the array is locally averaged using a 3x3 neighborhood. The contents of these accumulators are displayed in Fig.3; there exist two prominent clusters higher than a noise threshold of 100.

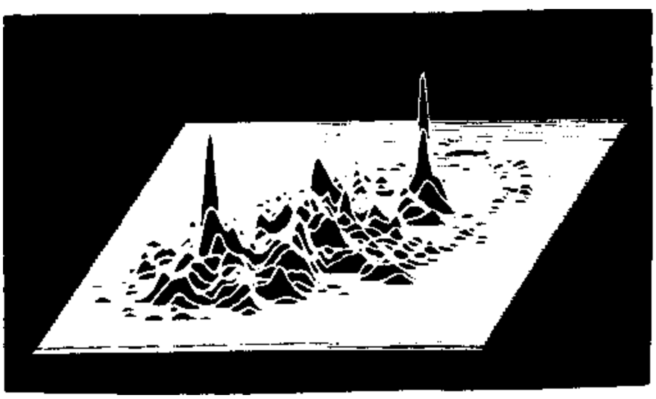
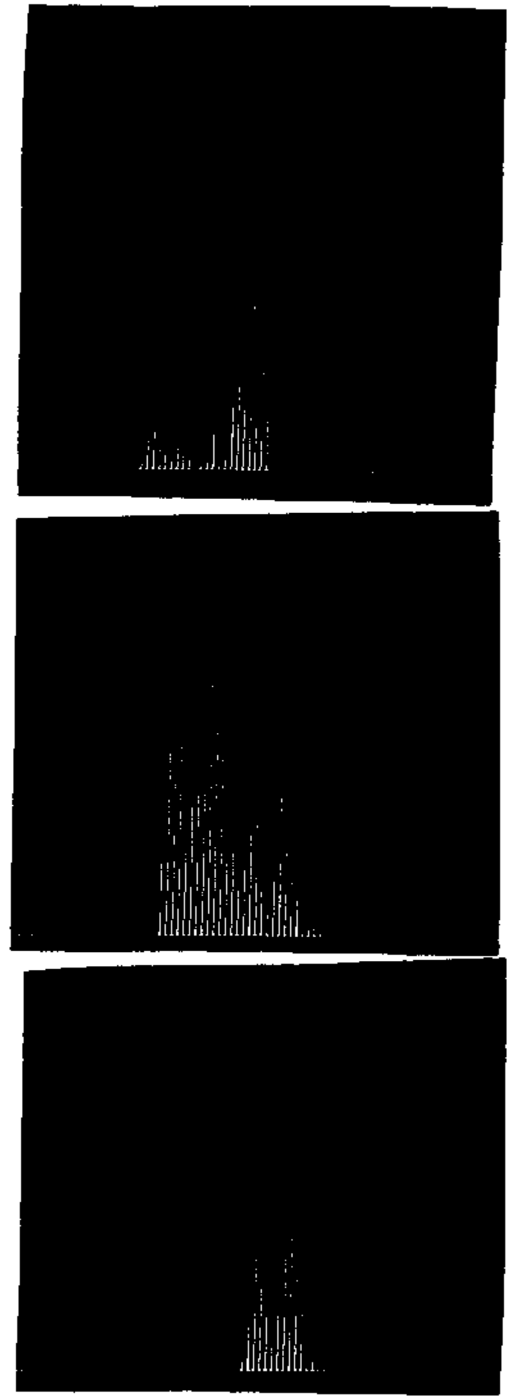


Fig.3 Contents of the two-dimensional array of accumulators after mapping pairs of parallel edge points. There are two prominent peaks.

After all pairs of edge points, which contributed to the highest (left) peak in Fig.3, are selected as candidates for ellipses, we apply to them the candidate-testing algorithm, whose first step is to calculate the^histogram of R of the candidates. The result is illustrated in Fig.4 (a). Since the number of edge points on an ellipse depends on the size of the ellipse, it seems reasonable to use a threshold function which is proportional to R, for suppressing noises in the histogram, and 15R is used in this experiment. After application of this threshold function, a peak finder detects the single prominent peak in Fig.4 (a), and then the least mean squares method can easily fits elliptic arcs to the edge



(a)

(b)

(c)

Fig.4 Contents of the one-dimensional array of accumulators.
(a) Histogram of R for the cluster of the left peak in Fig.3.
(b) Histogram of R for the cluster of the right peak in Fig.3.
(c) Histogram of R after erasing the edge points on the ellipse corresponding to the leftmost peak in (b).

points of the peak. Selection of another threshold function with a smaller coefficient does not influence the finally obtained ellipses, but much computing time is wasted by processing low peaks and rejecting them as noise.

Fig.4 (b) shows another histogram of R, obtained for the second peak in Fig.3, in which we can observe three prominent peaks and a medium peak. Edge points contributing to the left peak are selected as qualified members for the smallest ellipse in C, and then the least mean squares method fits an ellipse to them, which corresponds to the inner ellipse on the cylindrical part at the right side in the input picture. Erasing the edge points on that ellipse from the set *C* of candidates, we again calculate a new histogram of R. The result (Fig.4(c)) implies that the second peak in Fig.4 (b) is a false one, generated by the interaction of the recovered ellipse with the other, because the peak disappears as a result of erasing the edge points contributing to the adjacent peak. Since the left peak in the new histogram happens to be lower than the threshold function of 15R, we examine the right peak and detect the outer ellipse on the right part. If a lower threshold function, say 10R, is used, then the left peak which corresponds to the elliptic arcs between the two ellipses is examined, but the peak is judged to be a false one because the edge points in *E* cover only a small fraction of ellipses evaluated by the least mean squares method. Finally, the linear segments, which are judged as parts of the recovered ellipses, are discarded from the list of lines, and we obtain the result displayed in Fig.5.

The procedure is programmed in FORTRAN, and the total computing time on a minicomputer PDP8/E (12kw of core memory, 24kbytes of buffer memory, and 1.8Mw disk memory) is about 13 min 20 sec; 2 min 10 sec for calculating the gradients and storing them in the disk, 1 min 40 sec for detecting the linear segments, and 9 min 30 sec for recovering the ellipses and evaluating their parameters. This computing time seems rather long; however, we pay more attention to the ratio of the time for recovering the ellipses to that for recovering the straight lines, since the computing time depends on the ability of the computer used. The ratio 6:1 for our example seems to be acceptable when we consider the complexity of the shapes of ellipses.

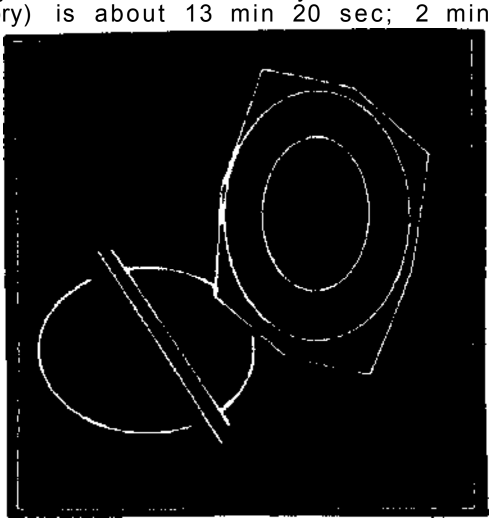


Fig.5 Detected linear and elliptic edges.

DISCUSSION

The proposed method seems to be satisfactory for extracting the global features of the artificially constructed objects. Although it can detect some of partially hidden ellipses, such as the elliptic arcs on the object at the left side in Fig.5, there are many counterexamples. A partially obscured ellipse is not detectable if it contains a less number of edge points symmetrical to the center of the ellipse than the threshold for finding peaks in the two-dimensional array. Setting the threshold at a lower level is not effective, because it can add few ellipses to the result while much computing time is wasted by examining a large number of weak clusters in the array. An example of another class of undetected ellipses is shown in Fig.6. We can recover a cluster corresponding to the center of the ellipse; however, the number of the edge points which have partners mapped together in the one-dimensional accumulator array is very small, so that the method misperceives the elliptic arcs as noise.

Another difficulty arises when we apply the method to a picture containing very large or highly concentric ellipses. Flat portions of the ellipses are erased

Fig.6 An example of elliptic arcs containing few pairs of edge points mapped in the histogram of R.

from the edge-enhanced picture before applying the ellipse-finding algorithm, because they satisfy the conditions for the long straight lines. However, the other parts of the ellipses are generally detected as sets of elliptic arcs, therefore we could improve the method by adding a function for finding straight lines which connect these elliptic arcs, and then fitting again an ellipse to the edge points on them.

Finally, let us briefly study on the computation required for detecting ellipses using the approach of Duda and Hart, who have shown that the computation required for obtaining lines increases with the number of edge points in the picture [Duda 72]. Suppose that the enhanced picture E has n edge points after erasing the long straight lines from it. If we detect ellipses by searching the five-dimensional parameter space for clusters, then the computation required for mapping each edge element into the space is proportional to d^n, where d is the number of quantization of a parameter. Thus, the computation required for detecting ellipses is proportional to nd^4.

The computing time of the proposed method is spent in searching the two parameter spaces and evaluating parameters of the ellipses. If the distribution of the directions of edges is uniform in [0°,180°) , then an edge list has n/d_θ elements, where d_θ is the number of quantization of 6. Therefore, $n^2/2d_\theta^2$ pairs of each edge list are mapped in the two-dimensional array, and the total number of mapping parallel edge elements is $n^2/2d_\theta$. In order to obtain edge elements contributing to the prominent peaks, this mapping process is repeated again, thus time t_1 for the center-finding process is approximately proportional to n^2/d_θ On the other hand, mapping into the one-dimenaional space and evaluating parameters is done on much smaller numbers of candidates than n; thus the ratio of the computing time t_2 of this process to t_1 will become very small for large n.

When we simply compare n^2/d_θ with nd^4_θ superiority of the proposed method over the direct application of parametrization is apparent. However, pictures with many edge points requires too much computing time, so that we must make n as small as possible. One idea for decreasing n is to apply a thinning operator to the edge-enhanced picture before mapping into the parameter spaces.

REFERENCES

Duda R. 0, and Hart P. E., "Use ot the Hough transformation to detect lines and curves in pictures," Commus. Ass. Comput. Vol.15, ppll-15, March, 1972.

Falk G., "Interpretation of imperfect data as a three dimensional scene," Artificial Intelligence, Vol.3, pplOl-114, no.2, 1972.

Griffith A. K., "Edge detection in simple scene using a priori information," IEEE Trans. Comput., Vol.C-22, pp371-381, no.4, 1973.

Hough P. V. C, "Method and means for recognizing complex patterns," U.S. Patent 3 069 654, Dec, 1962.

Kimme C, Ballard D. and Sklansky J., "Finding circles by an array of accumulators," Commus. Ass. Comput., Vol.18, ppl20-122, Feb., 1975.

O'Gorman F. and Clowes M. B., "Finding picture edges through collinearity of feature points," IEEE Trans. Comput., Vol.C-25, pp449-455, Apr., 1976.

Shapiro D. S., "Transformation for the computer detection of curves in noisy pictures," Computer Graphics and Image Processing, Vol.4, pp328-338, 1975.

Tsuji S. and Nakamura A., "Recognition of an object in a stack of industrial parts," Proc. of 4th IJCAI, pp811-818, 1975.