Detection of Ellipses by a Modified Hough Transformation

SABURO TSUJI AND FUMIO MATSUMOTO

Abstract—**The Hough transformation can detect straight lines in an edge-enhanced picture, however its extension to recover ellipses requires too long a computing time. This correspondence 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 almost complete ellipses, then evaluates their parameters by the least mean squares method.**

Index Terms—**Curve detection, line fitting, Hough transformation, pattern recognition, picture processing, scene analysis.**

I. INTRODUCTION

Analysis of a scene containing many complicated-shaped objects is a current problem of computer vision. Since usual line finders fail to extract a reliable line drawing from the scene, Falk has proposed a method which identifies each object in a blocks world by utilizing input data and external constraints so as to suggest and test hypotheses on the scene [1]. In order for a computer vision system to analyze more complex pictures containing real objects such as telephones, cups, or industrial parts, Falk's 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 simple patterns, and 3) test the validity of the models by examining whether weak feature points around the patterns satisfy the proposed models [2].

Line finding methods proposed by Hough [3], Duda and Hart [4], and O'Gorman and Clowes [5], are useful to find the polygons, especially partly occluded ones, because they utilize global properties of edge points. Duda and Hart also discussed the possibilities of extending their method to find curves in the picture. 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 by Kimme *et al.* [6] is necessary to recover the circular arcs in a reasonable time.

This correspondence describes how we can modify the parameterization technique so as to recover both linear and elliptic edges, the important cues for analyzing the scene containing the artificially constructed objects. Applying O'Gorman and Clowes' method to an edge-enhanced picture, we select long lines from a list of the obtained linear segments and erase the edge points on the long lines from the picture (short linear segments may belong to flat portions of some ellipses). Thus, the problem is how to efficiently search a set of feature points on curved or short linear segments for ellipses which have five unknown parameters. 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 ellipse or not. There exist interactions between different patterns when one maps edge points into the parameter spaces. As the result, strong ellipses with many edge points sometime mask weak clusters corresponding to other ellipses. Therefore, the computer vision system iteratively searches for an almost complete ellipse in a set of feature points in the edge-enhanced picture from which the edge points on the recovered long straight lines and ellipses have been erased. Although this method assumes that the ellipses in the picture are complete, it can also find ones of which small portions are obscured.

Manuscript received April 18, 1977; revised October 17, 1977.
The authors are with the Department of Electrical Engineering, Faculty of Engineering Science, Toyonaka, Osaka, Japan.

II. 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 location, shape, and orientation of an ellipse. Its usage, however, is impractical because of too long a 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 examine sequentially clusters in both a two-dimensional and a one-dimensional parameter space, by utilizing two well known properties of an ellipse, and select efficiently a small number of candidate edge points for an almost complete 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

Preprocessing a digitized input picture to find edge points by applying a simple gradient operator to every point in the picture and thresholding it, we obtain an edge-enhanced picture $\boldsymbol{E}$ in which each edge element is characterized by its location (x, y) and quantized direction θ which ranges between 0° and 180°. Next, the edge points are arranged in a set $\{\boldsymbol{e}_\theta\}$ of edge lists: the edge list $\boldsymbol{e}_\theta$ is a collection of edge points of a direction θ. Now, O'Gorman and Clowes' method [5] detects sets of collinear edge points in $\boldsymbol{E}$ and registers them in a list of straight lines, however, some of them may belong to ellipses. Longer straight lines than a threshold are considered not to be parts of the elliptic arcs, and the edge points on them are erased from $\boldsymbol{E}$ and $\{\boldsymbol{e}_\theta\}$ 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. 1(a)]. A simple property, that P and Q be at equal distance from the center O of the ellipse, is useful to find the locations of ellipses in $\boldsymbol{E}$ by a two-dimensional accumulator array $\{\boldsymbol{a}_{x,y}\}$; the array is indexed by two parameters (x, y) specifying the location of an ellipse. For each pair of two edge points (x_1, y_1) and (x_2, y_2) in an edge list $\boldsymbol{e}_\theta$ (a collection of edge points having an orientation θ), an accumulator at $((x_1 + x_2)/2, (y_1 + y_2)/2)$ is incremented by unity. After all pairs of the set $\{\boldsymbol{e}_\theta\}$ of the edge lists are processed in this way, the array is locally averaged by using a 3 × 3 neighborhood. Thus, the accumulator corresponding to the center of a

0018-9340/78/0800-0777$00.75

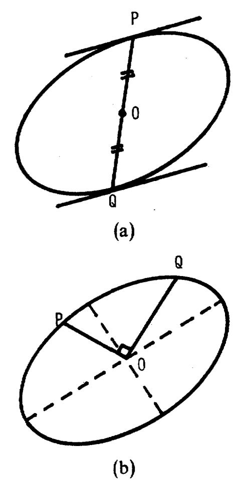

Fig. 1. (a) $OP = OQ$, if two tangents are parallel. (b) $1/OP^2 + 1/OQ^2 = 1/R^2$ = constant, for an ellipse if $\angle POQ = 90°$.

complete ellipse has a count approximately proportional to the length of its circumference.

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

Testing Candidates for Ellipse and Evaluation of Parameters

Let C be the set of 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 unsatisfactory 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 edge points which are exactly lying on an ellipse from the candidates by utilizing a property of an ellipse; then apply the fitting process for evaluating accurate parameters.

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

$$\frac{1}{OP^2} + \frac{1}{OQ^2} = \frac{1}{R^2} = \text{constant} \tag{1}$$

for an ellipse.

Now, a one-dimensional accumulator array $\{a_i\}$ is used to test the candidates. We assume that O is located at (m, n) obtained by the center-finding algorithm. If any two points P and Q in C satisfy $\angle POQ = 90° \pm \delta$ (δ is a threshold value), then the accumulator corresponding to R evaluated by (1) is incremented by unity. 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 to us 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 consider that there is no visible ellipse in C, if the histogram contains only low hills.

Mutual interaction between edge points on the concentric ellipses generates significant false peaks in the histogram, sometimes higher than 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; then evaluate five parameters of the best fitting ellipse to these edge points by the least mean squares method. If edge points in E cover only a small fraction of this ellipse, the fitting process is judged as a failure; otherwise, it is considered as a success, and the edge points on the recovered ellipse are excluded from E, C, and $\{e_\theta\}$ in order to eliminate their interaction with other ellipses. The candidate-testing process is iterated by calculating a new histogram 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 almost complete 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 edge points remaining in E are not or sparsely distributed on the line; otherwise it is qualified as a straight line.

III. Experimental Result

The following example shows some of the features of the ellipse-finding algorithm. Fig. 2(a) shows a 128×128 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 3×3 window [5] 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. Setting the quantization step of θ at 6°, we classify the edge points into 30 groups (edge lists). At this point, the line-finding algorithm is applied, and linear segments longer than 10 are registered in the list of lines. Now let us consider how one can select the threshold for discriminating long lines from flat portions of ellipses. The length L_m of the longest flat portion of an ellipse described by $(x/a)^2 + (y/b)^2 = 1$ is

$$L_m = 2\left(\frac{a^2}{b}\right)\left(\frac{\pi}{180°}\right)\left(\frac{6°}{2}\right) \simeq \frac{1}{10}\left(\frac{a}{b}\right)a. \tag{2}$$

Considering the size of the picture is 128 by 128, we expect $a < 50$.

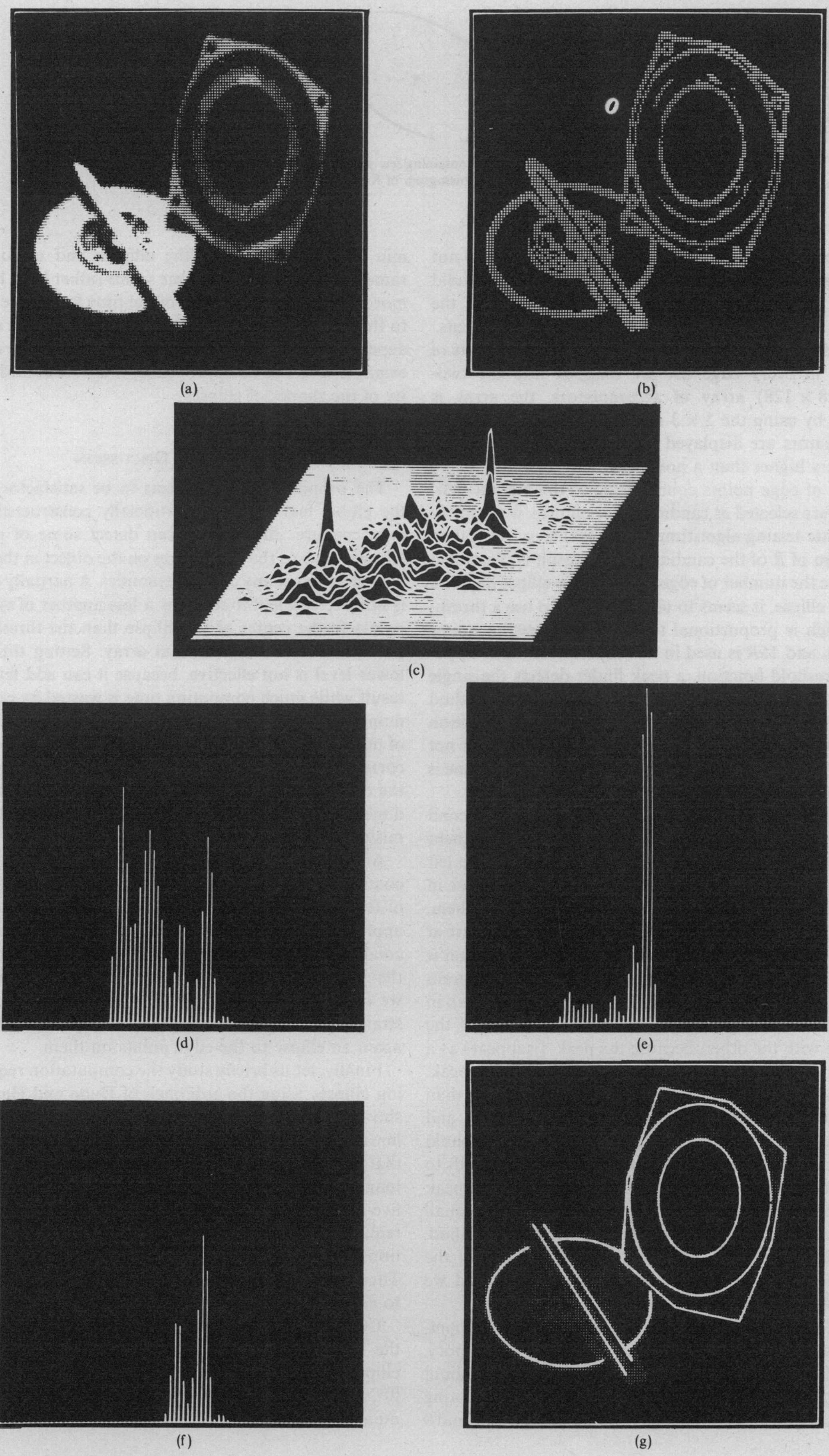

Fig. 2. (a) A 128 by 128 digitized picture, containing three industrial parts. (b) Edge-enhanced picture of (a). (c) Contents of two-dimensional array of accumulator after mapping pairs of parallel edge points. There are two prominent peaks. (d) Histogram of R for the cluster of the left peak in (c). (e) Histogram of R for the cluster of the right peak in c). (f) Histogram of R after erasing edge points on the ellipse corresponding to the leftmost peak in (e). (g) Detected linear and elliptic edges.

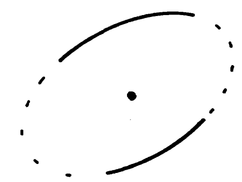

Fig. 3. Example of elliptic arcs containing few pairs of edge points mapped in the histogram of R.

Therefore, ellipses which are not highly eccentric ($a < 4b$) do not include linear segments longer than 20. Thus we set the threshold at 20 and erase the edge points lying on lines longer than the threshold. E and the edge lists now contain 1993 edge elements.

Now, the ellipse-finding algorithm is applied. After all pairs of the edge points in every edge list are mapped into the two-dimensional (128×128) array of accumulators, the array is locally averaged by using the 3×3 neighborhood. The contents of these accumulators are displayed in Fig. 2(c); there exist two prominent clusters higher than a noise threshold of 100.

After all pairs of edge points contributing to the highest (left) peak in Fig. 2(c) 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. 2(d). Since the number of edge points on an ellipse depends on the size of the ellipse, it seems to us 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. 2(d), then the least mean squares method can easily fit elliptic arcs to the edge 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. 2(e) shows another histogram of R, obtained for the second highest peak in Fig. 2(c), 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, 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 it from the set C of candidates, we calculate again a new histogram of R. The result [Fig. 2(f)] implies that the second highest peak in Fig. 2(e) 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 as a false one because 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. 2(g).

The procedure is programmed in Fortran, and the total computing time on a minicomputer PDP8/E (12 kW of core memory, 24 kbytes of buffer memory, and 1.8 MW disk memory) is about 13 min 20 s: 2 min 10 s for calculating the gradients and storing them in the disk, 1 min 40 s for detecting the linear segments and 9 min 30 s for recovering the ellipses and evaluating their parameters. This computing time seems rather long, however we pay more attention on 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 appropriate, when we consider the complexity of the shapes of ellipses.

IV. Discussion

The proposed method seems to be satisfactory for extracting the global features of the artificially constructed objects in the input picture. Although it can detect some of partially hidden ellipses, such as the elliptic arcs on the object at the left side in Fig. 2(g), there are many counterexamples. A partially obscured ellipse is not detectable if it contains a less number of symmetrical edge points 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. 3. 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, thus the method misinterpretes 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 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 the computation required for detecting ellipses using the approach of Duda and Hart [4], who have shown that computation required for obtaining lines increases linearly with the number of edge points in the picture. 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 computation required for mapping each edge point into the space is proportional to d^4, where d is the number of quantization of a parameter. Then computation required for detecting ellipses is proportional to nd^4.

Computing time by 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 θ. 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 points 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-dimensional space and evaluating parameters are 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, superiority of the proposed method over the direct application of parameterization is apparent. However, pictures with many edge points require too much computing time, then 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

[1] G. Falk, "Interpretation of imperfect data as a three dimensional scene," *Artificial Intelligence*, vol. 3, pp. 101–144, 1972.

[2] S. Tsuji and A. Nakamura, "Recognition of an object in a stack of industrial parts," in *Proc. 4th Int. Joint Conf. Artificial Intelligence*, Tbilisi, USSR, 1975.

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

[4] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Commun. Assoc. Comput. Mach.*, vol. 15, pp. 11–15, Mar. 1972.

[5] F. O'Gorman and M. B. Clowes, "Finding picture edges through collinearity of feature points," *IEEE Trans. Comput.*, vol. C-25, pp. 449–455, Apr. 1976.

[6] C. Kimme, D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," *Commun. Assoc. Comput. Mach.*, vol. 18, pp. 120–122, Feb. 1975.

On Universal Single Transition Time Asynchronous State Assignments

TAKASHI NANYA AND YOSHIHIRO TOHMA

***Abstract*—One of two constructive procedures for universal single transition time (STT) state assignments presented in the paper [1] by Friedman *et al.* is not correct. A correct constructive procedure can be obtained by simply modifying the procedure of the literature with an additional restriction.**

***Index Terms*—Asynchronous sequential circuits, single transition time (STT) state assignments, universal state assignments.**

The constructive procedures for universal single transition time (STT) state assignments for asynchronous sequential machines presented in the paper [1] by Friedman *et al.* seems well accepted in common. However, this correspondence shows that the second procedure, which is described in Theorem 4 in Friedman's paper, is not correct in some cases. A correct constructive procedure based on the same idea can be obtained by a slight modification with an additional restriction.

Manuscript received January 7, 1977; revised July 28, 1977.

T. Nanya is with Central Research Laboratories, Nippon Electric Co., Ltd., Kawasaki-shi, Japan.

Y. Tohma is with the Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan.

A counterexample against the procedure of Theorem 4 will be presented.

Consider two matrices M'_1 and M'_2 as shown below.

$$M'_1 = \begin{bmatrix} 000 \\ 011 \\ 101 \\ 110 \end{bmatrix} \qquad M'_2 = \begin{bmatrix} 000000 \\ 011011 \\ 101101 \\ 110110 \\ 000111 \end{bmatrix}.$$

Obviously, M'_1 is a (2, 2)-separating system with four rows and three columns and M'_2 with five rows and six columns, too. That is, $m'_1 = 4$, $n'_1 = 3$, $m'_2 = 5$, and $n'_2 = 6$. Note that $m'_2 > m'_1$ and m'_2 is odd. Using M'_1 and M'_2 as M_1 and M_2 in Theorem 4, respectively, we obtain a matrix M' with $m'_1 \cdot m'_2$ $(= 20)$ rows and $n'_1 + 3n'_2$ $(= 21)$ columns. M' is as follows:

$$M' = \begin{array}{|c|c|c|c|l}
\cline{1-4}
000 & 000000 & 000000 & 000000 & \leftarrow l \\
000 & 011011 & 011011 & 011011 & \\
000 & 101101 & 101101 & 101101 & \\
000 & 110110 & 110110 & 110110 & \\
000 & 000111 & 000111 & 000111 & \\
\cline{1-4}
011 & 000000 & 000111 & 110110 & \leftarrow i \\
011 & 011011 & 000000 & 000111 & \\
011 & 101101 & 011011 & 000000 & \\
011 & 110110 & 101101 & 011011 & \\
011 & 000111 & 110110 & 101101 & \leftarrow k \\
\cline{1-4}
101 & 000000 & 110110 & 011011 & \\
101 & 011011 & 000111 & 101101 & \\
101 & 101101 & 000000 & 110110 & \\
101 & 110110 & 011011 & 000111 & \\
101 & 000111 & 101101 & 000000 & \\
\cline{1-4}
110 & 000000 & 101101 & 000111 & \\
110 & 011011 & 110110 & 000000 & \leftarrow j \\
110 & 101101 & 000111 & 011011 & \\
110 & 110110 & 000000 & 101101 & \\
110 & 000111 & 011011 & 110110 & \\
\cline{1-4}
\end{array}$$

Denote the 6th, the 17th, the 10th, and the 1st rows of M' as i, j, k and l, respectively. Then, any column of M' cannot dichotomize $(i, j/k, l)$. Namely, M' is not the universal STT state assignment.

The reason why this happens is due to the imperfectness of the proof of Theorem 4. In Case 6 of the proof, it is true that "for some $q = 1$, 2 or 3, we must have both $j_q \neq k$ and $l_q \neq i$," but we may have $j_q = l_q$ for such q that $j_q \neq k$ and $l_q \neq i$. Then, any of columns $n_1 + (q-1)n_2 + 1$ through $n_1 + qn_2$ may not dichotomize $(i, j_q/k, l_q)$. Therefore, any of columns of M may not dichotomize $(i, j/k, l)$.

A correct constructive procedure can be obtained by adding M_a shown below for column $n_1 + 3n_2 + 1$ through $n_1 + 4n_2$ of M, where number m_2 of rows in a block must not be divided either by 2 or by 3.