# A PARALLEL MECHANISM FOR DETECTING CURVES IN PICTURES

Philip M. Merlin
David J. Farber

TECHNICAL NOTE #39

DECEMBER 1973

Department of Information and Computer Science
University of California, Irvine

## ABSTRACT

HOUGH has proposed a procedure for detecting lines in pictures. DUDA-HART extended the method for a more general curve fitting. This paper shows how this method can be used to detect any given curve. The procedure presented here can be easily implemented and is more efficient in a parallel machine.

## 1. INTRODUCTION

A current problem in computer picture processing is the detection of the best fit of a given curve in a digitalized image. In the simplest case, the picture is formed by black points in a white background. In the general case the picture is represented by a grid of grey levels. The problem is to detect the presence of points in the picture that match or almost match to a given curve.

HOUGH [1] and DUDA-HART [2] have developed a method for detecting straight lines. KIMME-BALLARD-SKLANSKY [3] extended the same mechanism to finding circles. These methods transform each point in the picture plane to a curve in a parameter space. The parameter space is defined by the parametric representation used to describe the curves in the picture plane. In the case of line detection, HOUGH uses the slope-intercept parameters and DUDA-HART the normal parameters. In the latter case, a line is given by the angle $\theta$ of its normal and its algebraic distance $\rho$ from the origin. Thus the equation of the line is:

$$x \cos \theta + y \sin \theta = \rho$$

If we restrict to the interval $(0,\pi)$, then the normal parameters of a line are unique. With this restriction, every line in the X-Y plane corresponds to a unique point in the $\theta$-$\rho$ plane.

Suppose that we have some set $\{(x_1, y_1)\ (x_2, y_2)\ \text{----------}(x_n, y_n)\}$ of n figure-points and we want to find a set of straight lines that fit them. We transform the points $(x_i, y_i)$ into the sinusoidal curves in the plane defined by:

$$x_i \cos \theta + y_i \sin \theta = \rho$$

It can be shown that the curves corresponding to colinear figure points have a common point of intersection. This point in the $\theta$-$\rho$ plane, $(\theta_0, \rho_0)$, defines the line passing through the colinear points. Thus, the problem of detecting colinear points can be converted to the problem of finding concurrent curves. HOUGH and DUDA-HART describe a method of finding the point of concurrence $(\theta_0, \rho_0)$. DUDA-HART extends the transform method to other curves.

In general, with this method, for each point (element) in the picture plane, it is necessary to trace a different curve in the parameter plane. Therefore, we have to execute different computations for each point in the original picture. This leads to two disadvantages:

(1) The total number of operations during the implementation of this algorithm in a sequential computer is substantial.

(2) There is not an obvious way to implement that method in a parallel picture oriented machine.

The possibility of implementing the above algorithm for picture processing in a parallel organization is of importance because the number of points in a real picture leads to an extremely long computation time, thus increasing cost and preventing real time analysis of the image.

This paper describes an extention of the HOUGH algorithm allowing the detection of any given curve. The method presented applies the same transformation to each element of the original picture requiring less computation time and permitting the implementation of the algorithm in a parallel organized machine.

## 2. THE METHOD

Suppose that we have a picture described by the set of points $\{(X_1, Y_1)\ (X_2, Y_2)\ \text{------}\ (X_n, Y_n)\}$ and we want to find the best fit of some given curve to the points; for example the curve E shown in figure 1. Mark a point A on the curve. Now the problem is to find a best point A' in the X-Y plane so that if we trace the curve E in the plane and the points A and A' coincide, then most of the points $(X_i, Y_i)$ will match the curve E. Each possible trace corresponds to a translation of the curve in the plane of the picture.

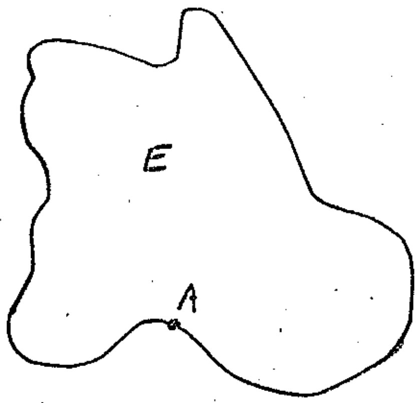

Figure 1

Each $(X_i, Y_i)$ may belong to an infinite number of different traces of E. Some sample traces are shown in figure 2. For each $E_j$ traced through

## 2. THE METHOD

Suppose that we have a picture described by the set of points $\{(x_1, y_1)\ (x_2, y_2)\ \text{------}\ (x_{n'}, y_m)\}$ and we want to find the best fit of some given curve to the points; for example the curve E shown in figure 1. Mark a point A on the curve. Now the problem is to find a best point A' in the X-Y plane so that if we trace the curve E in the plane and the points A and A' coincide, then most of the points $(x_i, y_i)$ will match the curve E. Each possible trace corresponds to a translation of the curve in the plane of the picture.

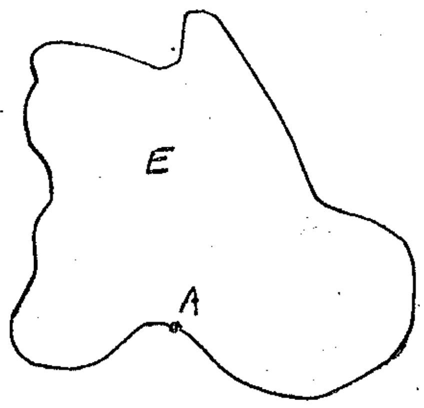

Figure 1

Each $(x_i, y_i)$ may belong to an infinite number of different traces of E. Some sample traces are shown in figure 2. For each $E_j$ traced through the point $(X_i, Y_i)$ there is a candidate $A_j$, so that if this candidate is elected for A', then $(X_i, Y_i)$ will be on the trace of E. In the next step, we will find the locus of all possible candidates $A_j$ so that its traces, $E_j$, fit the point $(X_i, Y_i)$.

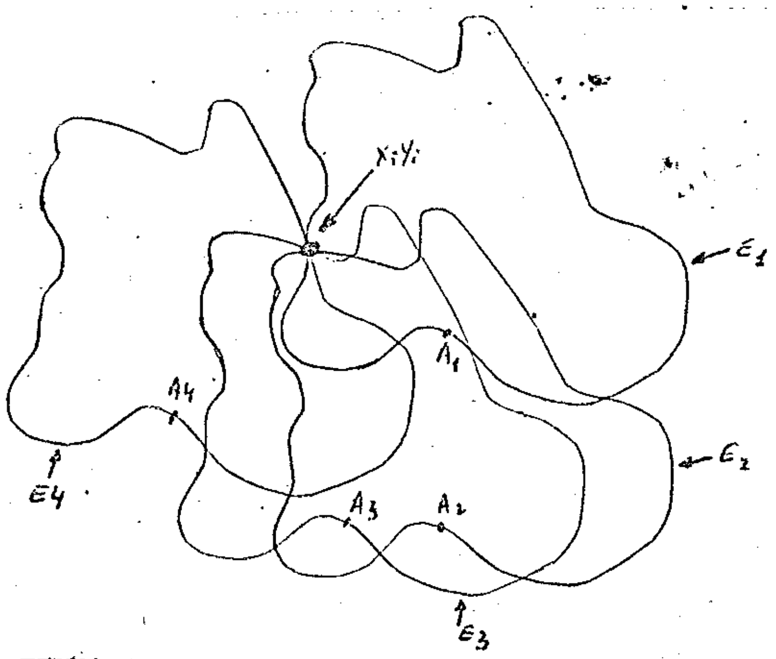

Figure 2: Different traces of E that fit the point $(X_i, Y_i)$

Suppose we chose $(x_i, y_i)$ so that it is the point A', and we draw the respective E; we then draw another possible $E_j$ as shown in figure 3. Now, we trace the line defined by the points $(x_i, y_i)$ and $A_j$; the intersection of this line with E will be called P. It is easy to show that the segments

$\overline{P, (x_i, y_i)}$ and $\overline{(x_i, y_i), A_j}$ are equal. Therefore, the locus of all the possible points $A_j$ that determine curves $E_j$, passing through $(x_i, y_i)$, is symmetric relative to the point $(x_i, y_i)$ of the curve E. This locus, called $L_i$, is shown by a dotted curve in figure 3. Note that the symmetric of a curve relative to a point is a 180° rotation of the curve around the given point.

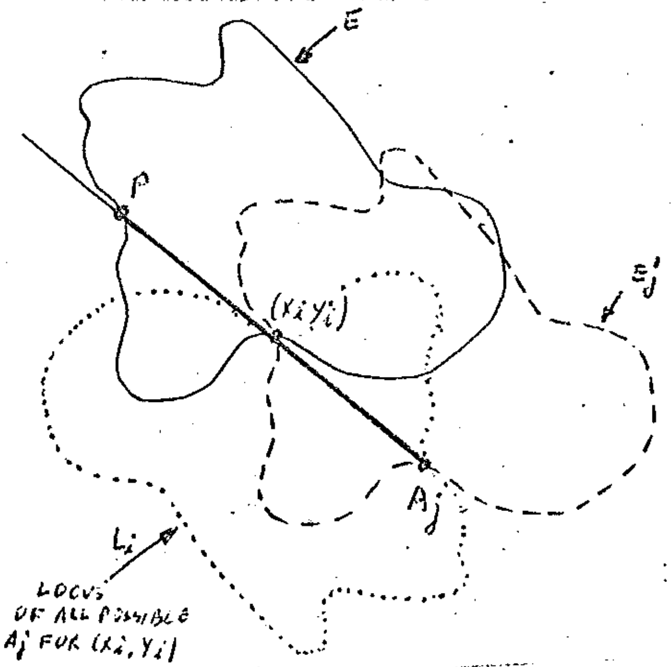

Figure 3: The locus of all possible Aj for $(x_i, y_i)$

For each point $(x_i, y_i)$ in the picture, the respective locus $L_i$ is traced. The point where the maximum number of locuses intersect is the best candidate for A'. Then the maximum possible number of points $(x_i, y_i)$ will fit the curve E' (figure 4).

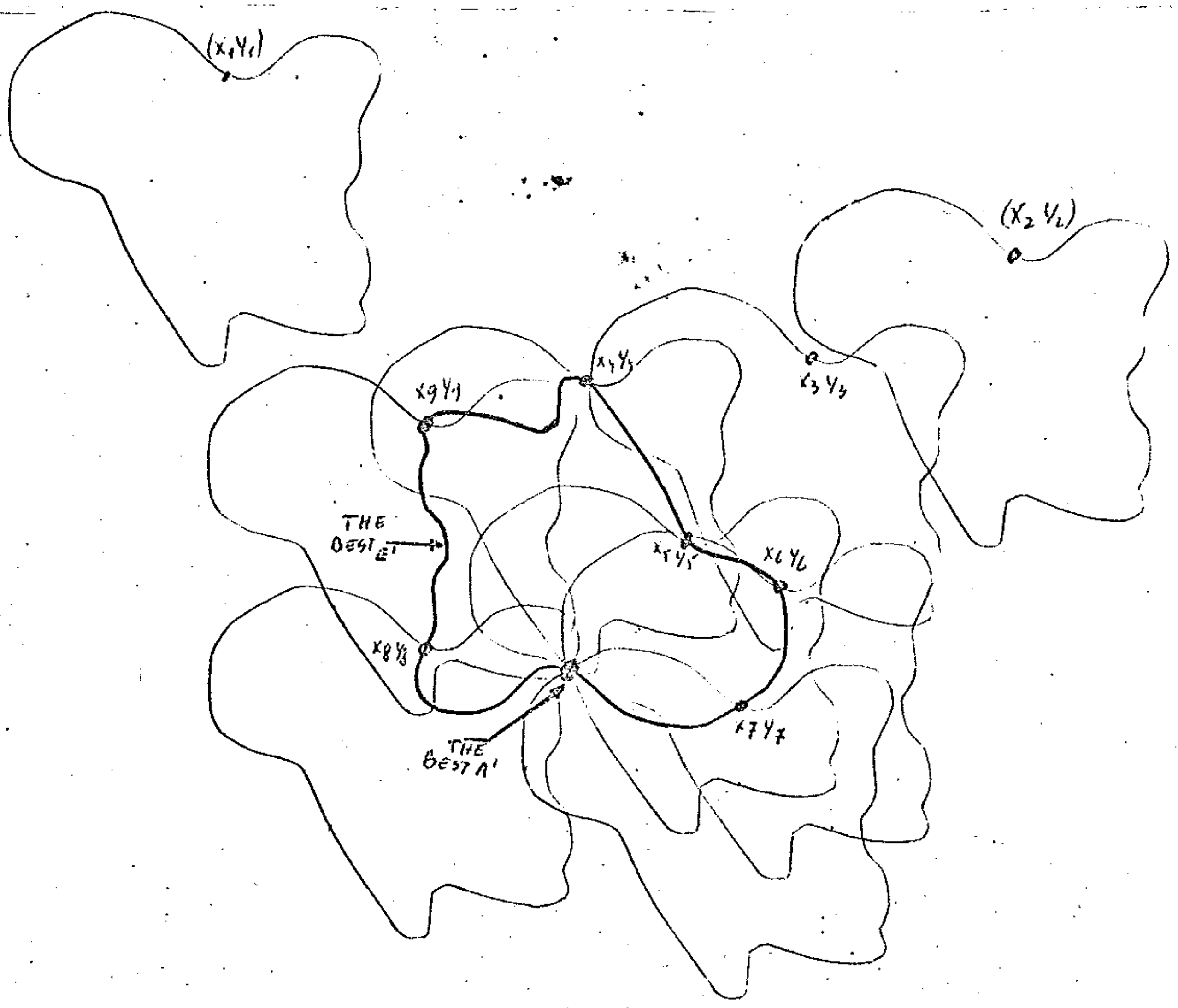

Figure 4: An example of the best candidate for A .

## RECAPITULATION

We can summarize the algorithm in the following steps:

1. Mark a reference point called A in the given curve.
2. Rotate the given curve 180°.
3. Trace the rotated curve with A on each of the points of the picture.
4. Find the point where the maximum number of curves traced in step 3 intersect.
5. The best fit to the given curve is the trace with A on the point found in step 4.

Most of the computation time during the execution of the algorithm is spent in step 3. But, because we have to trace the same curve for all the points in the picture, it is possible to execute this step in parallel. For pedagogical reason, in the discussion of the method we choose the reference point A on the given curve E. In general, it is possible to mark this point anywhere in the plane of E. If the given curve E is a circle, for example, it would be better to choose its center for the reference point; then, the 180° rotation and the curve are the same.

## 3. THE PARALLEL IMPLEMENTATION

The implementation of the method is similar to that suggested by HOGH and DUDA-HART but exploits the parallel nature of the algorithm.

Suppose the plane of the picture is quantized according to the acceptable error. Now, the picture is represented by the matrix $P_{ij}$. In the simplest case, $P_{ij}$ takes on the values "0" or "1". We define another matrix $M_{ij}$ that represent the plane on where the locuses will be traced. The dimension of $M_{ij}$ is equal to that of $P_{ij}$. $M_{ij}$ is treated as a two-dimensional array of accumulators.

We define the set of pairs R (i,j) as the collection of coordinates of the points in the quantized plane that belong to a 180° rotation of the given curve E with the point A on (i,j). Then R(i,j) is a translation of R(o,o). That is, if:

$$R(o,o) = [\ (a_o,b_o)\ (a_1,b_1)\ ----\ (a_n,b_n)] \qquad (1)$$

then:

$$R(i,j) = [\ (i-a_o,\ j-b_o)\ (i-a_1,\ j-b_1)\ ----(i-a_n,\ j-b_n)\ ] \qquad (2)$$

Now step 3 in the algorithm can be restated as follows. First, clear the matrix M. For each (i,j) in the quantized picture add the value of $P_{ij}$ to the elements of M defined by the set of points R(i,j). In a formal notation:

$$\forall i\ \forall j\ [M_{lm \in R(i,j)} \longleftarrow M_{lm \in R(i,j)} + P_{ij}] \qquad (3)$$

after substitution of R(i,j) from (2):

$$\forall i\ \forall j\ \forall k\ [M_{i-a_k,j-b_k} \longleftarrow M_{i-a_k,\ j-b_k} + P_{ij}] \qquad (4)$$

This can be done in parallel for all the points in P by changing the order of the operations, i.e.:

$$\forall k\ \forall i\ \forall j\ [M_{i-a_k,j-b_k} \longleftarrow M_{i-a_k,\ j-b_k} + P_{ij}] \qquad (5)$$

In other words, to each point in $M_{ij}$ we add the value of $P_{i-a_k,j-b_k}$; and return to this procedure for all the pairs $(a_k,b_k)$ that define the 180° rotation of the given curve. With this procedure we have to compute each value for the increment index ($a_k$ and $b_k$) only once for all the points in $P_{ij}$, which implies that processing time is decreased.

But major gain is achieved by the implementation of the algorithm in a parallel machine [4,5]. In such a machine, the algorithm can be executed by translations of the $P_{ij}$ matrix and additions of the translated array to the $M_{ij}$ matrix; the total number of these parallel operations is the same as the number of points in the discrete plane X-Y that define the curve to be detected.

Now, after the accumulation of all the traces, the best candidate for A' will correspond to the cell with the maximum value in the matrix M, because through this cell passes the maximum number of traces, as explained in Section 2.

The method described is also suitable for the case in which the picture $P_{ij}$, has different grey levels.

## REFERENCES

1. Hough, P.V.C. Method and means for recognizing complex patterns. U.S. Patent 3,069,654, December 18, 1962.

2. Duda, R. O. and Hart, P. E., Use of Hough Transformation to Detect Lines and Curves in Pictures., Communication of the ACM, January, 1972.

3. Kimme, C., Ballard, D., Sklansky, J., Finding Circles by an Arrav of Accumulators; Interim Report IP-73-5, School of Engineering, UC Irvine, California

4. J. Sklansky, A Parallel Natural Image Computer; Personal Communications, School of Engineering, UC Irvine, California, September 1966.

5. B. Kruse, A Parallel Picture Processing Machine; IEEE, Transactions on Computers, volume C-22, number 12, December 1973.

ACKNOWLEDGMENT

The authors gratefully acknowledge the helpful coments of the members of the Pattern Recognition Group in the School of Engineering at UCI during the preparation of this paper. Particular thanks goes to Prof. Jack Sklansky, the Chairman of this group.