Use of the Hough Transformation To Detect Lines and Curves in Pictures

Richard O. Duda and Peter E. Hart
Stanford Research Institute,
Menlo Park, California

Hough has proposed an interesting and computationally efficient procedure for detecting lines in pictures. This paper points out that the use of angle-radius rather than slope-intercept parameters simplifies the computation further. It also shows how the method can be used for more general curve fitting, and gives alternative interpretations that explain the source of its efficiency.

Key Words and Phrases: picture processing, pattern recognition, line detection, curve detection, colinear points, point-line transformation, Hough transformation

CR Categories: 3.63

This work was supported by the Advanced Research Projects Agency and the National Aeronautics and Space Administration under Contract NAS12-2221.

1. Introduction

A recurring problem in computer picture processing is the detection of straight lines in digitized images. In the simplest case, the picture contains a number of discrete, black figure points lying on a white background. The problem is to detect the presence of groups of colinear or almost colinear figure points. It is clear that the problem can be solved to any desired degree of accuracy by testing the lines formed by all pairs of points. However, the computation required for n points is approximately proportional to n^2, and may be prohibitive for large n.

Rosenfeld [1] has described an ingenious method due to Hough [2] for replacing the original problem of finding colinear points by a mathematically equivalent problem of finding concurrent lines. This method involves transforming each of the figure points into a straight line in a *parameter space*. The parameter space is defined by the parametric representation used to describe lines in the picture plane. Hough chose to use the familiar slope-intercept parameters, and thus his parameter space was the two-dimensional slope-intercept plane. Unfortunately, both the slope and the intercept are unbounded, which complicates the application of the technique. In this note we suggest an alternative parameterization that eliminates this problem. We also give two alternative interpretations of Hough's method, one of which reveals plainly the source of its efficiency. Finally, we show how the method can be extended to find more general classes of curves in pictures.

2. Fundamentals

The set of all straight lines in the picture plane constitutes a two-parameter family. If we fix a parameterization for the family, then an arbitrary straight line can be represented by a single point in the parameter space. For reasons that become obvious, we prefer the so-called *normal parameterization*. As illustrated in Figure 1, this parameterization specifies a straight line by the angle θ of its normal and its algebraic distance ρ from the origin. The equation of a line corresponding to this geometry is

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

If we restrict θ to the interval $[0, \pi]$, then the normal parameters for a line are unique. With this restriction, every line in the x-y plane corresponds to a unique point in the θ-ρ plane.

Suppose now, that we have some set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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 θ-ρ plane defined by

$$\rho = x_i \cos \theta + y_i \sin \theta. \qquad (1)$$

It is easy to show that the curves corresponding to colinear figure points have a common point of intersection. This point in the θ-ρ plane, say (θ_0, ρ_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.

A dual property of the point-to-curve transformation can also be established. Suppose we have a set $\{(\theta_1, \rho_1), \ldots, (\theta_k, \rho_k)\}$ of points in the θ-ρ plane, all lying on the curve

$$\rho = x_0 \cos \theta + y_0 \sin \theta.$$

Then it is easy to show that all these points correspond to lines in the x-y plane passing through the point (x_0, y_0). We can summarize these interesting properties of the point-to-curve transformation as follows:

Property 1. A point in the picture plane corresponds to a sinusoidal curve in the parameter plane.

Property 2. A point in the parameter plane corresponds to a straight line in the picture plane.

Property 3. Points lying on the same straight line in the picture plane correspond to curves through a common point in the parameter plane.

Property 4. Points lying on the same curve in the parameter plane correspond to lines through the same point in the picture plane.

In Section 3 we apply these results to the problem of detecting colinear points in the picture plane and show how significant computational economies can be realized in certain situations.

Fig. 1. The normal parameters for a line.

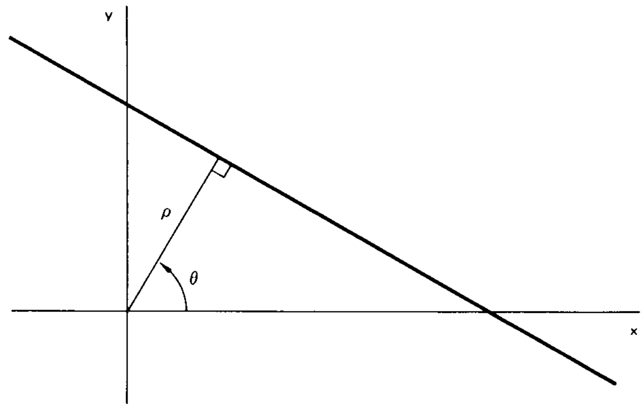
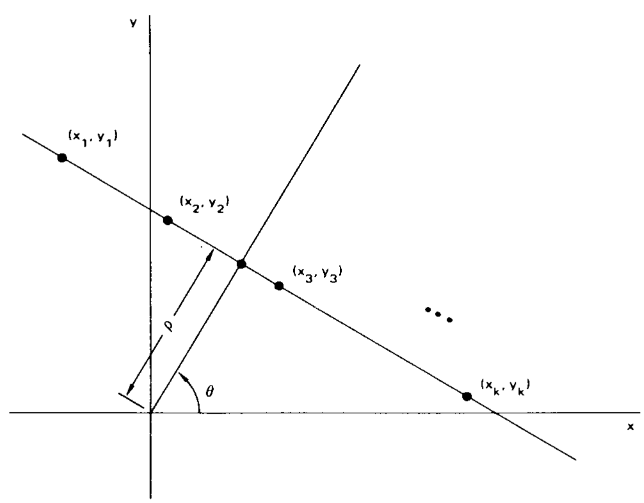


Fig. 2. Projection of colinear points onto a line.



3. Applications and Alternative Interpretations

Suppose we map all of the points in the picture plane into their corresponding curves in the parameter plane. In general, these n curves will intersect in $n(n-1)/2$ points corresponding to the lines between all pairs of figure points. Exactly colinear subsets of figure points can be found, at least in principle, by finding coincident points of intersection in the parameter plane. Unfortunately, this approach is essentially exhaustive, and the computation required grows quadratically with the number of picture points.

When it is not necessary to determine the lines exactly, the computational burden can be reduced considerably. Following Hough's basic proposal, we specify the acceptable error in θ and ρ and quantize the θ-ρ plane into a quadruled grid. This quantization can be confined to the region $0 \leq \theta < \pi$, $-R \leq \rho \leq R$, where R is the size of the retina, since points outside this rectangle correspond to lines in the picture plane that do not cross the retina. The quantized region is treated as a two-dimensional array of accumulators. For each point (x_i, y_i) in the picture plane, the corresponding curve given by (1) is entered in the array by incrementing the count in each cell along the curve. Thus, a given cell in the two-dimensional accumulator eventually records the total number of curves passing through it. After all figure points have been treated, the array is inspected to find cells having high counts. If the count in a given cell (θ_i, ρ_j) is k, then precisely k figure points lie (to within quantization error) along the line whose normal parameters are (θ_i, ρ_j).

An alternative interpretation of the point-curve transformation can be obtained by recognizing that the ρ computed by (1) locates the projection of the point (x_i, y_i) onto a line through the origin with slope angle θ. Thus, if a number of figure points lie close to some line l, their projections onto the line normal to l are nearly coincident (see Figure 2). A given column in the θ-ρ accumulator array is just a histogram for these projections, so a high count in a given cell clearly corresponds to a nearly colinear subset of figure points. A variation of this approach was used by Griffith [3] to find long lines in a picture.

Let us investigate how the computation required by the accumulator implementation varies with the number of figure points. To be more specific about the quantization, suppose that we restrict our attention to d_1 values of θ uniformly spaced in the interval $[0, \pi)$. Suppose further that the ρ axis in the interval $[-R, R]$ is quantized into d_2 cells. For each figure point (x_i, y_i), we use (1) to compute the d_1 different values of ρ corresponding to the d_1 possible values of the independent variable θ. Since there are n figure points, we need to carry out this computation nd_1 times. When these computations are complete, the d_1d_2 cells of the two-dimensional accumulator are inspected to find high counts. Thus the computation required grows linearly with the number of figure points. Clearly, when n is large compared to d_1, this approach is preferable to an exhaustive procedure that requires considering the lines between all $n(n-1)/2$ pairs of figure points.

Another alternative interpretation exposes the source of this efficiency. Consider again Property 4 in Section 2: Points lying on the same curve in the θ-ρ plane correspond to lines through the same point in the picture plane. When the curve corresponding to figure point (x_i, y_i) is "added" to the accumulator, we are really computing and recording the parameters of the d_1 lines in the picture plane passing through (x_i, y_i), and because θ is quantized, these are "all the lines in the plane" passing through (x_i, y_i). Should a given parameter pair ever recur as a result of computing the d_1 lines through some other figure point, the recurrence will be reflected in an increased count in the appropriate accumulator cell. Roughly speaking, then, for each figure point the quantized transform method considers only the set of all d_1 lines through that point, whereas more exhaustive methods consider all $(n - 1)$ lines between the given point and all other figure points.

4. Example

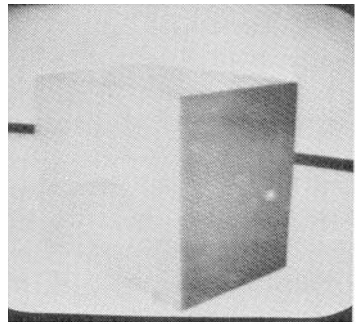
The following example illustrates some of the features of the transform approach. Figure 3(a) shows a television monitor view of a box, and Figure 3(b) shows a digitized version of that view. A simple differencing operation locates significant intensity changes and produces the binary picture shown in Figure 3(c). This 120×120 picture contains many nearly colinear figure points that can be fit well by a few straight lines.

Sampling θ at $d_1 = 9$ twenty-degree increments in θ and, quantizing ρ into $d_2 = 86$ two-element cells, we obtain the two-dimensional accumulator array shown in Table I. If the array entry at (θ_0, ρ_0) is k_0, then k_0 figure points lie on parallel lines for which $\theta = \theta_0$, and ρ lies between ρ_0 and $\rho_0 + 2$. When many points are nearly colinear, the entry for the line that fits them best is large. The largest entry in Table I occurs at $(0°, -5)$ and corresponds to the middle vertical edge of the box. The nine circled entries correspond to locally maximum values that exceed the arbitrary threshold of 35. The corresponding nine groups of nearly colinear figure points are shown in Figure 3(d). In this example, it happens that every group corresponds to some physically meaningful line in the picture. However, two significant lines on the top of the box were not found—one, because it contained very few points, and the other, because it fell between the lines at $\theta = 80°$ and $\theta = 100°$. The 20° angular quantization interval was chosen to keep the accumulator array small. Clearly, we were fortunate to have found as many lines as we did, and a smaller quantization interval would have to be used in practice.

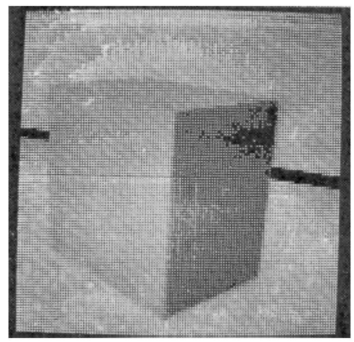
A few remarks concerning some limitations of the transform approach are in order. First, the results are sensitive to the quantization of both θ and ρ. Finer quantization gives better resolution, but increases the computation time and exposes the problem of clustering entries corresponding to nearly colinear points. Second, the technique finds colinear points without regard to contiguity. Thus the position of a best-fit line can be distorted by the presence of unrelated figure points in another part of the picture. A related problem is that of meaningless groups of colinear points being detected. In our example, a false line would be detected if the threshold were reduced from 35 to 24, the value needed to detect the top left-hand edge of the box.

The transform approach does successfully find groups of colinear or nearly colinear figure points. If the minimum size of a significant group is known, all such groups can be detected. If additional properties such as contiguity are known, they can be used to reject meaningless results. In general, the transform approach should be viewed primarily as a computationally efficient way of accomplishing a conceptually simple step in scene analysis.

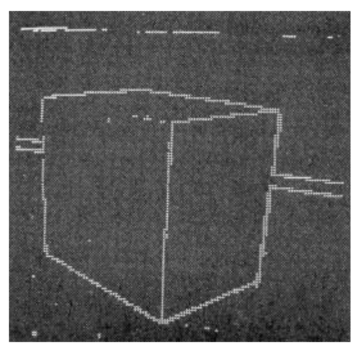
Fig. 3 An illustrative example.



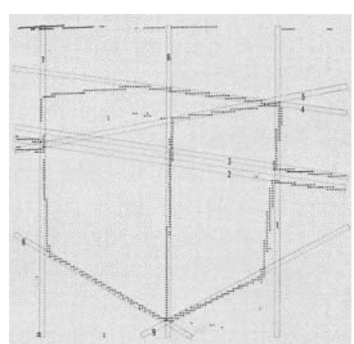
(a) Monitor



(b) Digitized



(c) Gradient



(d) Lines

5. Extensions and Conclusions

The transform method can be generalized and specialized in several ways. We note immediately that any parametrization of the family of straight lines can be used. As we have mentioned, Hough used the slope-intercept parameterization. However, this parameterization has the disadvantage of being sensitive to the choice of coordinate axes in the picture plane. If several figure points lie on a nearly vertical line, for example, both the slope and the intercept may be arbitrarily large. Thus the entire two-dimensional parameter plane must be considered. As Rosenfeld [1] has pointed out, one could do the entire problem twice, interchanging the x- and y-axes, but this would introduce additional complications. The normal parametrization avoids these disadvantages, fundamentally for the same reasons that make it useful in integral geometry: It allows us to place an invariant measure on the set of all straight lines.

An important special use of the transform method is to detect the occurrence of figure points lying on a straight line and possessing some specified property. For example, suppose we want to find whether a significant number of figure points lie on a line through the point (x_0, y_0) in the picture plane. As we have seen from Property 4, the normal coordinates of any such line must lie on (or, in practice at least near) the curve $\rho = x_0 \cos \theta + y_0 \sin \theta$. Hence, the transform process can be carried out in the usual way, but attention can be restricted to the region of the θ-ρ plane near this curve. If we find a cell with count k near this curve, then we are assured that k figure points lie on a line passing (nearly) through the point (x_0, y_0). Similarly, suppose we are interested only in lines having a given direction, say θ_0. Again, we carry out the process in the usual way, but restrict our attention to a subset of the θ-ρ plane in the vicinity of $\theta = \theta_0$.

It is clear that the general transform approach can be

Table I. Accumulator Array for Figure 3(c)

ρ \ θ	0°	20°	40°	60°	80°	100°	120°	140°	160°
85								2	
83							1	2	
81							4	4	
79			2				6	5	
77				2			8	4	2
75							6	6	2
73							4	3	3
71		2		1		1	4	2	3
69			1	4		12	4	3	5
67			3		2	14	2	3	4
65			1			11	1	2	4
63					5	2		2	4
61						1		3	9
59	4	1			11	9	1	8	12
57	4	3		3	10	12	3	10	15
55	9	5		4	5	4	5	11	12
53	6	6		4	10		11	9	14
51	4	9		4	20	2	11	10	8
49	5	6		2	10	3	11	13	8
47	8	4	4	4		2	13	10	10
45	4	7	14	3			11	6	8
43	4	18	21	5		1	12	10	8
41	9	17	21	15		25	18	7	8
39	8	20	21	13		22	11	11	7
37	12	17	22	17		9	10	9	10
35	38	14	17	17	38	8	7	9	6
33	37	16	22	21	42	10	5	9	9
31	35	11	21	23	23	8	11	9	10
29	13	18	18	23	20	14	13	9	9
27	7	16	12	30	20	20	7	9	6
25	7	18	12	32	19	27	8	7	8
23	8	12	11	20	17	52	11	6	7
21	7	17	12	23	8	11	15	11	10
19	9	14	12	16	7	7	14	6	7
17	9	12	12	16	6	9	16	12	7
15	8	13	13	11	7	10	16	14	10
13	10	9	15	11	7	10	16	13	6
11	12	11	13	14	40	10	16	13	13
9	10	10	16	14	8	9	14	21	22
7	10	8	22	12	41	6	7	12	21
5	11	12	15	11	23	6	11	14	14
3	13	15	15	8	18	7	11	16	15
1	10	14	17	11	7	8	9	10	12

ρ \ θ	0°	20°	40°	60°	80°	100°	120°	140°	160°
−85									
−83									
−81									
−79			3	1					
−77			1	3					
−75									
−73									
−71		2							
−69		2							
−67									
−65			1	2					
−63		1			2				
−61	5								
−59	7		2	1		1			
−57	6					2	1		2
−55	0	10	6			1	1		4
−53	16	13	12		18	4	1		6
−51	40	15	11		15	16			6
−49	32	18	11	57	15	23	1	1	5
−47	10	16	11	22	14	16	21	9	5
−45	7	17	11	11	16	18	41	21	6
−43	8	12	14	10	13	17	12	17	6
−41	6	7	14	11	14	14	7	19	12
−39	7	10	9	8	12	8	11	20	23
−37	7	7	14	8	17	9	12	18	24
−35	8	9	17	8	10	7	10	23	23
−33	6	12	15	8	12	9	11	22	26
−31	5	9	19	9	8	11	16	18	15
−29	9	10	12	9	8	9	18	18	15
−27	7	12	10	8	6	9	18	19	19
−25	5	10	8	8	7	7	22	9	14
−23	6	11	9	9	6	11	19	12	9
−21	7	15	9	7	10	10	16	10	11
−19	6	13	8	16	9	11	17	9	10
−17	7	17	9	15	7	11	16	14	13
−15	6	15	10	17	8	13	10	14	9
−13	10	15	9	15	9	17	11	13	12
−11	10	13	10	7	8	17	9	11	15
−9	7	14	8	7	8	23	8	12	15
−7	9	15	12	7	8	21	7	13	12
−5	77	13	15	9	7	14	10	12	15
−3	26	14	14	6	8	12	9	11	18
−1	10	13	18	9	8	8	11	12	15

extended to curves other than straight lines. For example, suppose we want a method to detect circular configurations of figure points. We can choose a parametric representation for the family of all circles (within a retina) and transform each figure point in the obvious way. If, as a parametric representation, we describe a circle in the picture plane by

$$(x - a)^2 + (y - b)^2 = c^2,$$

then an arbitrary figure point (x_i, y_i) will be transformed into a surface in the a-b-c parameter space defined by

$$(x_i - a)^2 + (y_i - b)^2 = c^2.$$

In this example, then, each figure point will be transformed into a right circular cone in a three-dimensional parameter space. If the cones corresponding to many figure points intersect at a single point, say the point (a_0, b_0, c_0), then all the figure points lie on the circle defined by those three parameters. As in the preceding case of straight lines, no saving is effected if the entire process is performed analytically. However, the process can be implemented efficiently by using a three-dimensional array of accumulators representing the three-dimensional parameter space.

In principle, then, the transform method extends to arbitrary curves. We need only pick a convenient parameterization for the family of curves of interest and then proceed in the obvious way. A parameterization having bounded parameters is obviously preferable, although this is not essential. It is much more important to have a small number of parameters since the accumulator implementation requires quantization of the entire parameter space and the computation grown exponentially with the number of parameters.

Received November 1970, revised April 1971

References

1. Rosenfeld, A. *Picture Processing by Computer*. Academic Press, New York, 1969.
2. Hough, P.V.C. Method and means for recognizing complex patterns. U.S. Patent 3,069,654, Dec. 18, 1962.
3. Griffith, A.K. Computer recognition of prismatic solids. Ph.D. Th., Dep. of Math., MIT, June 1970.