Improved Curve Detection Through Decomposition of the Hough Transform

Clark F. Olson
Computer Science Department
Cornell University
Ithaca, NY 14853
clarko@cs.cornell.edu

May 15, 1995

Abstract

This paper describes techniques to perform fast and accurate curve detection using a variant of the Hough transform. It is shown that the Hough transform can be decomposed into many small subproblems, where each subproblem considers only curves that pass through some subset of the points. These curves correspond to a manifold in the parameter space. This property allows the effects of localization error to be modeled more accurately than previous systems. The additional use of randomization techniques leads to efficient algorithms. The time required by this method is $O(n)$, where n is the number of edge points in the image, if we are only required to find curves that are significant with respect to the complexity of the image. Results are given showing the detection of lines and circles in real images.

1 Introduction

The Hough transform is a method to detect parameterized curves in images by mapping image edge points into manifolds in parameter space and then finding peaks in the parameter space [Hough, 1962, Duda and Hart, 1972]. For example, we may parameterize lines by their slope and intercept ($y = mx + b$). Each edge point (x, y) in the image would then be mapped into the line $b = -xm + y$ in parameter space, corresponding to all of the lines that pass through (x, y). Typically, peaks are found by a multi-dimensional histograming procedure, where each manifold votes for each cell of the histogram that it passes through. Cells of the histogram with high scores correspond to curves that pass through or near many edge points.

Recent work [Califano *et al.*, 1989, Xu *et al.*, 1990, Liang, 1991, Bergen and Shvaytser, 1991, Kiryati *et al.*, 1991, Leavers, 1992, Xu and Oja, 1993] has introduced methods for improving Hough transform techniques. One such technique maps point sets rather than single points into the Hough space. Another technique uses randomization to speed up the Hough transform. These techniques have allowed Hough methods to achieve better efficiency than previous methods. While these techniques may also make peaks in parameter space sharper, this paper shows that such methods do not inherently improve the accuracy of Hough methods (i.e. peaks due to the random accumulation of points will occur with the same frequency.)

This paper describes how the Hough transform can be decomposed into many small subproblems without loss of accuracy. Each subproblem considers only those curves that pass through some subset of the points. If we use sets of the correct cardinality, this constrains the curves examines by each subproblem to a 1-dimensional manifold of the parameter space. This allows us to better model the effects of error in the localization of edge points and achieve better accuracy than previous algorithms. The decomposition also allows randomization to be used in a more effective manner than previously. Similar decomposition techniques, but in the context of object recognition, can be found in [Cass, 1991, Olson, 1994].

Section 2 reviews a formal definition of the Hough transform given by Princen *et al.* [1993] and modifies it slightly to better apply to the techniques presented here. This definition is used in our analysis of the Hough transform and its variations given in the following sections. Section 3 discusses the technique of mapping point sets into the Hough space and shows how it relates to the standard Hough transform. Section 4 describes how these Hough transform techniques are decomposed into many small subproblems. The detection of lines and circles are discussed in detail in Sections 5 and 6, respectively. Section 7 analyzes the computational complexity of these algorithms. Section 8 describes experiments that have been performed to test the efficacy of these techniques. Finally, Section 9 discusses related work and Section 10 summarizes the paper. An appendix is given that describes parameterizations for several interesting curves and gives methods to solve for the parameters of each from the minimal amount of information.

2 The Hough transform

We'll use a modified version of the definition of the Hough transform given by Princen *et al.* [1993]. Let $X = (x, y)$ be a point in image space, $\Omega = (\omega_1, ..., \omega_N)$ be a point in the Hough space[1], and $f(X, \Omega) = 0$ be the function that parameterizes the set of curves. The set of edge points in the image, $\mathcal{E} = \{X_1, .., X_n\}$, is represented by the sum of the delta functions at the points' locations:

$$I(X) = \sum_{j=1}^{n} \delta(X - X_j)$$

Princen *et al.* use C_Ω to denote a cell in parameter space centered at Ω. Examining such cells allows the consideration of a discretized Hough space. They define:

$$p(X, \Omega) = \begin{cases} 1 & \text{if } \{\Lambda : f(X, \Lambda) = 0\} \cap C_\Omega \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

So $p(X, \Omega)$ is 1 if any curve in the parameter space cell, C_Ω, passes through the point, X, in image space. The Hough transform can then be written:

$$H(\Omega) = \int p(X, \Omega) I(X) dX \tag{1}$$

or

$$H(\Omega) = \sum_{j=1}^{n} p(X_j, \Omega) \tag{2}$$

$H(\Omega)$ is thus the number of image points that any curve in C_Ω passes through. We will see later that the effects of discretization of the Hough space will not be important in the method presented here, so we will not incorporate discretization of the Hough space into the Hough transform definition. Instead, we consider the error in the localization of the image points. Let's assume that the true location of each edge pixel lies within a bounded region N_X of the determined location X. We can redefine $p(X, \Omega)$ as follows:

$$p(X, \Omega) = \begin{cases} 1 & \text{if } \{Y : f(Y, \Omega) = 0\} \cap N_X \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

Now, $p(X, \Omega)$ is 1 if the curve represented by Ω passes through N_X. With this definition we can still use Equations 1 and 2 to describe the Hough transform.

Localization error is not easy to deal with correctly in Hough transform implementations. For this reason, many implementations have assumed that there is no error in the localization of the edge points. This paper gives an efficient method for

[1] We will use *Hough space* and *parameter space* interchangeably to describe the space of parameterized curves.

dealing with this error in accurate manner. The modified definition of the Hough transform will yield more accurate results for this analysis.

Standard Hough transform implementations typically discretize the Hough space into cells. A counter is maintained for each cell in the discretized Hough space. These counters record the number of edge points that map to a manifold that intersect each of the cells. In the errorless case, each edge point maps to an $N-1$ dimensional manifold in Hough space. (Recall that N is the number of parameters required to describe the curve.) Typical implementations thus need to increment the counters for the cells that cover this manifold. If localization error is to be handled correctly, we would need to determine for each edge pixel, which cells in the Hough space contain a curve that passes within the error boundary of the edge pixel, which is not a simple task.

3 Mapping point sets into Hough space

Let us now consider the technique of mapping point sets into the Hough space. Rather than considering each point separately, this method considers point sets of some cardinality k. For each such set, the curves that pass through every point in the set (or the error boundaries of every point in the set) are determined and the Hough space is incremented accordingly. The benefit of this technique is that each mapping will be to a smaller subset of parameter space. If $f(X, \Omega)$ is an N parameter function, then, in the errorless case, N non-degenerate edge points would map to a single point in parameter space[2]. We would thus need to increment only one bin in the Hough space for each set, rather than the bins covering an $N-1$ dimensional manifold for each point. Such a technique has been used in some form by several researchers [Califano *et al.*, 1989, Xu *et al.*, 1990, Liang, 1991, Bergen and Shvaytser, 1991, Leavers, 1992]. Of course, we don't need to use sets of size N, we could use any size $k > 0$. If $k \leq N$, each non-degenerate set maps to an $N-k$ dimensional manifold in Hough space. The disadvantage to methods that map point sets into the Hough space is that there are $\binom{n}{k}$ sets of image pixels with cardinality k to be considered.

An examination of how the technique of mapping point sets into Hough space is related to the standard Hough transform is informative. Let's label the Hough transform technique that maps sets of k points into parameter space $H^k(\Omega)$. An image curve (a point in Hough space) now gets a vote only if it passes within the error boundary of each point in the set, so we have:

$$H^k(\Omega) = \sum_{\{g_1, \ldots, g_k\} \in \binom{\mathcal{E}}{k}} p(X_{g_1}, \Omega) \cdot \ldots \cdot p(X_{g_k}, \Omega)$$

[2]This is true for the curves considered in this paper (lines, circles and ellipses,) but it is not necessarily true for arbitrary classes of curves. In general, if we assume non-degeneracy, then each set of N edge pixels maps to a finite set of points in the Hough space in the errorless case.

where $\binom{\mathcal{E}}{k}$ is the set of all k-subsets of the edge points, E.

Consider this function at an arbitrary point in parameter space. For some set, $\{g_1, ..., g_k\}$, the product, $p(X_{g_1}, \Omega) \cdot ... \cdot p(X_{g_k}, \Omega)$, will be 1 if and only if each of the $p(X, \Omega)$ terms is 1 and otherwise it will be 0. If there are x points such that $p(X, \Omega)$ is 1 (these are the points such that the curve, Ω, passes through their localization error boundary), then there are $\binom{x}{k}$ sets with cardinality k that will contribute 1 to the sum. $H^k(\Omega)$ will thus be $\binom{x}{k}$. Since the standard Hough transform will yield $H(\Omega) = x$ in this case, we can express $H^k(\Omega)$ simply in terms of $H(\Omega)$:

$$H^k(\Omega) = \binom{H(\Omega)}{k}$$

We can now consider whether this method will have better accuracy than the standard Hough transform. If the standard Hough transform uses threshold $t \geq k$ to find peaks and the method of mapping point sets into Hough space uses threshold $\binom{t}{k}$, they will find exactly the same set of peaks. Thus, the accuracy is the same. No correct peaks will be missed as a result of using this method and no false positives will be eliminated.

In addition, curve detection still requires time that is exponential in N. Let α be the number of bins in each dimension of the Hough space. The standard Hough transform increments $O(\alpha^{N-1})$ bins for each image point, for a total requirement of $O(n\alpha^{N-1})$. We now increment $O(\alpha^{N-k})$ bins for each of the $O(n^k)$ sets, for a total of $O(\alpha^{N-k}n^k)$[3]. So, this technique hasn't really gained us much, yet. The following sections describe how considerable improvement can be achieved through the use of decomposition and randomization.

Let's examine what these transforms look like in a parameter space. Figure 1 shows an example image with two short line segments amid some random noise. Figure 2 shows the standard Hough transform of this image using the ρ-θ parameterization for lines. The peaks corresponding to the line segments in Figure 1 are denoted by letters.

Figure 3 shows the Hough transform when sets of points with cardinality 2 are mapped into Hough space. While the peaks are much higher in this case, the analysis indicates that if there was a high false positive in the standard Hough transform, it would be very high in this plot as well. These plots were made with the assumption that each edge point was located at the center of the pixel that contained it and implicitly assume that there was no localization error.

[3] This assumes that $k \leq N$. If $k > N$, we require $O(n^k) > O(n^N)$ time, but this is inefficient, since we can achieve $O(n^N)$ time by using $k = N$. For this reason, we will not further consider using $k > N$.

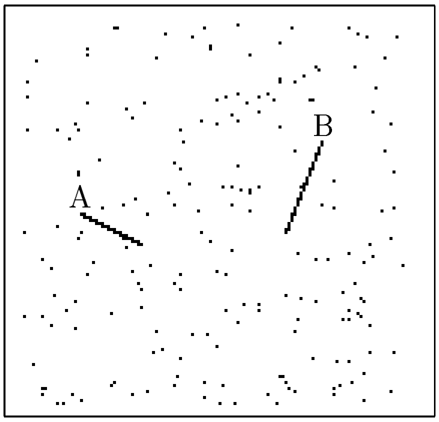

Figure 1: Line segments A and B amid random noise.

4 Decomposition into smaller problems

Let us now consider a new technique, where we map only those point sets into Hough space that share some distinguished set of j edge points, $\mathcal{D} = \{X_{d_1}, ..., X_{d_j}\}$. We will still vary $k - j$ edge points, $\mathcal{G} = \{g_1, ..., g_{k-j}\}$, in these sets. The point sets we are mapping into Hough space are thus $\mathcal{D} \cup \mathcal{G}$. This yields:

$$H^{\mathcal{D},k}(\Omega) = \sum_{\mathcal{G} \in \binom{\mathcal{E} \setminus \mathcal{D}}{k-j}} \prod_{i=1}^{j} p(X_{d_i}, \Omega) \prod_{i=1}^{k-j} p(X_{g_i}, \Omega)$$

Consider this function at an arbitrary point in Hough space. Since we aren't varying the distinguished points, $\{X_{d_1}, ..., X_{d_j}\}$, the curve must pass through the error boundary of each of these to get a non-zero response. If x points lie on a curve and we use a distinguished set of j of these points then $x - j$ of these points remain in $\mathcal{E} \setminus \mathcal{D}$. We thus have:

$$H^{\mathcal{D},k}(\Omega) = \begin{cases} \begin{pmatrix} H(\Omega) - j \\ k - j \end{pmatrix} & \text{if } \prod_{i=1}^{j} p(X_{d_i}, \Omega) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

We should thus use a threshold of $\binom{t-j}{k-j}$ in this case. Any curve that passes through the error boundaries of each of the distinguished points will then be found if it would have been found by the standard Hough transform with threshold t. We can formulate algorithms to recognize arbitrary curves by considering several subproblems, each of which examines a particular distinguished set, as above. A deterministic algorithm using these ideas would consider each possible distinguished set. This would guarantee that we would examine a correct distinguished set for each curve. If we are willing to allow a small probability of failure, we can use randomization to considerably reduce

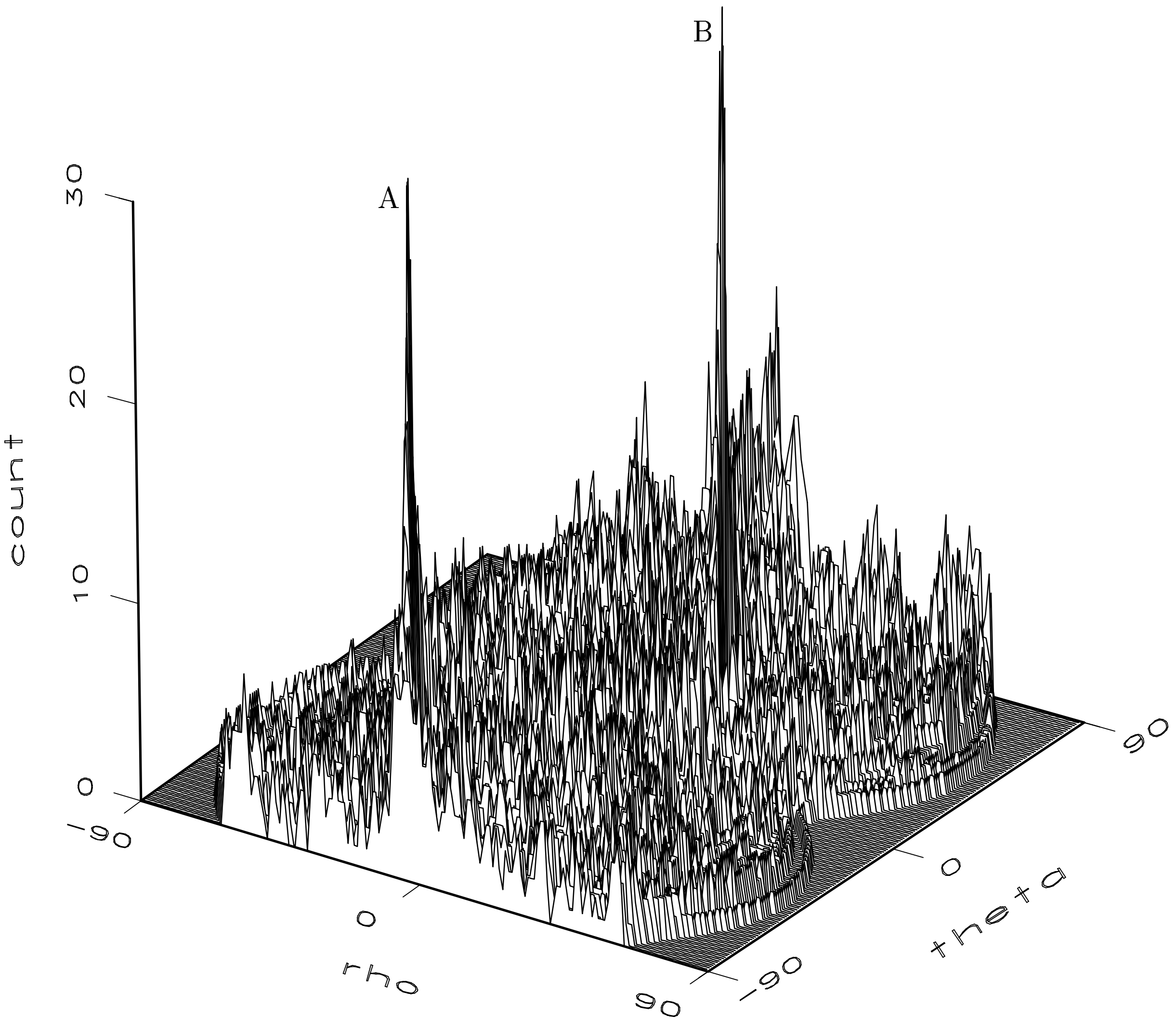

Figure 2: The standard Hough transform of the image in Figure 1. The peaks labeled A and B correspond to the line segments A and B.

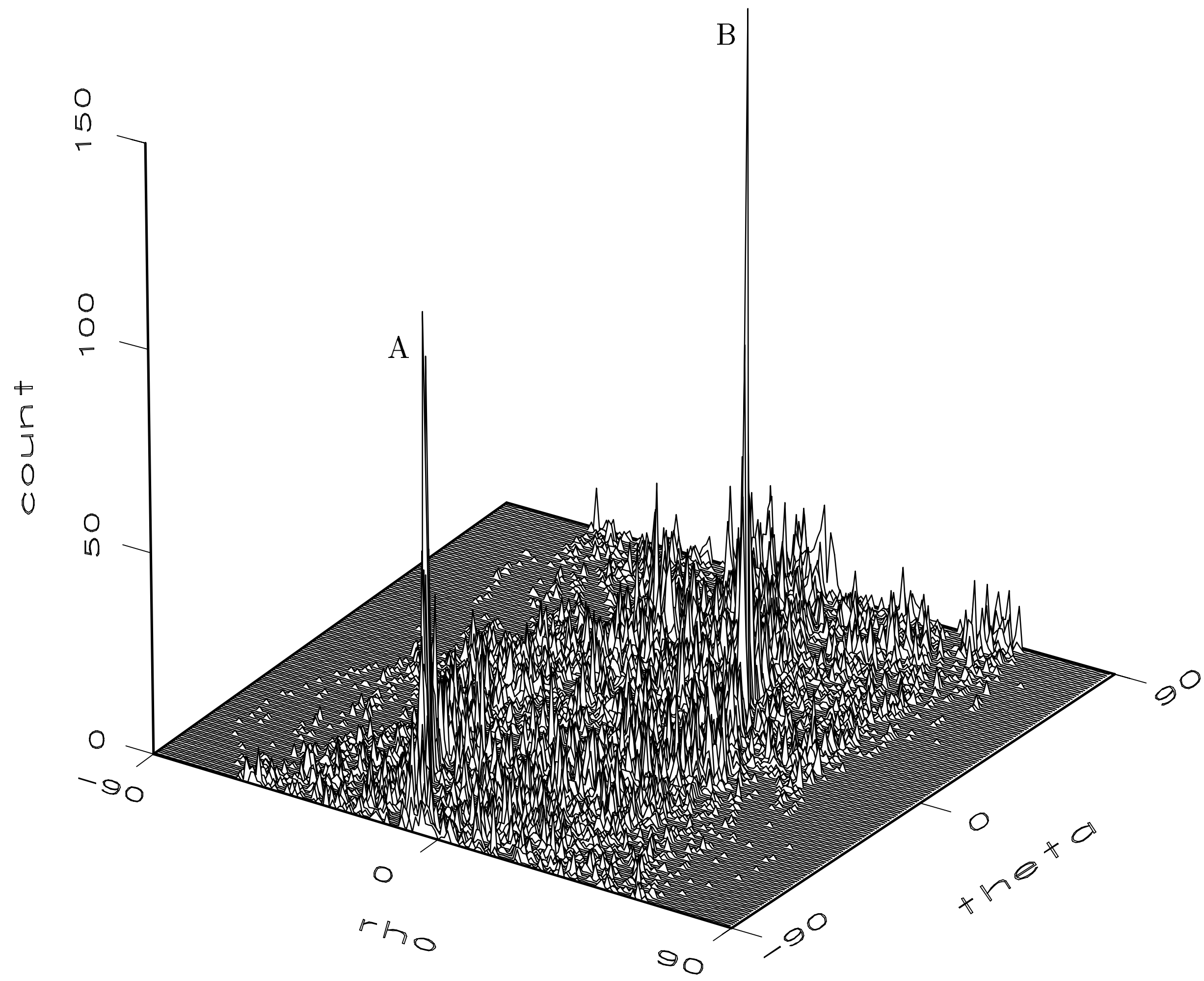

Figure 3: The multipoint Hough transform of the image in Figure 1, using point sets with cardinality 2 to map into Hough space. The peaks labeled A and B correspond to the line segments A and B.

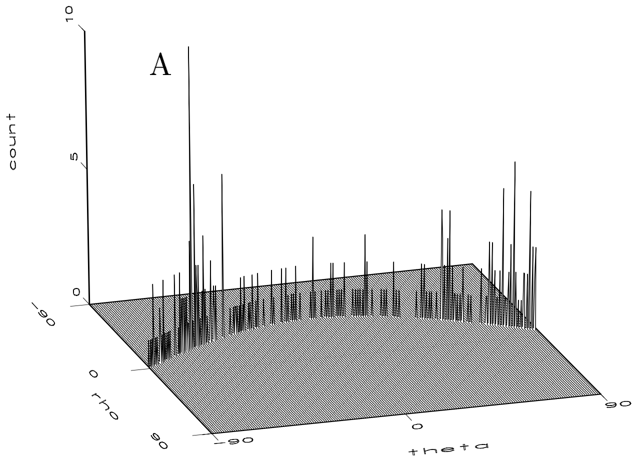

Figure 4: The decomposed Hough transform of the image in Figure 1 using a point from segment A as the distinguished point. The peak labeled A corresponds to the line segment A in Figure 1.

the number of distinguished sets that we need to examine. We will determine have many random distinguished sets we need to examine to maintain high accuracy in Section 7.

To gain the maximum decomposition of the problem, we want j to be as large as possible, but note that if we choose $j \geq k$, we will have $H^{\mathcal{D},k}(\Omega) = 0$ or 1 for all Ω. This means that our response will be 1 if a curve goes through the j points, but it yields no other information, since 1 is the maximum response. We thus want to have $j < k$. Furthermore, we saw earlier that we want to have $k \leq N$ for efficiency reasons. The optimal choice is thus $j = k - 1 = N - 1$.

Figures 4, 5 and 6 show three examples of such a decomposed Hough transform of the image in Figure 1. Figure 4 shows a case where a point on segment A was used as the distinguished point. Figure 5 shows a case where a point a segment B was used a the distinguished point. Figure 6 shows a case where a noise point was used as the distinguished point. Once again, we have implicitly assumed that there was no error in the localization of the edge points. Note that peaks are present where appropriate, but no peak is present when using a noise point as the distinguished point. Notice also that considering sets of edge points that vary in only one point (i.e. when $j = k - 1 = N - 1$) constrains the transform to lie on a 1-dimensional manifold (a curve) in the parameter space. Let's call this curve the *Hough curve*. When localization error is considered, the Hough transform will no longer be constrained to

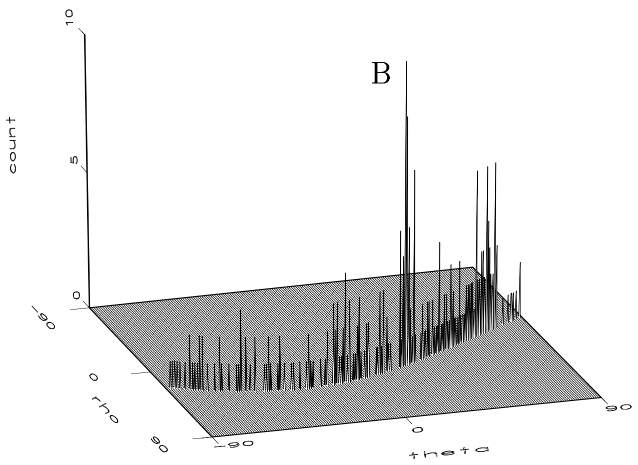

Figure 5: The decomposed Hough transform of the image in Figure 1 using a point from segment B as the distinguished point. The peak labeled B corresponds to the line segment B in Figure 1.

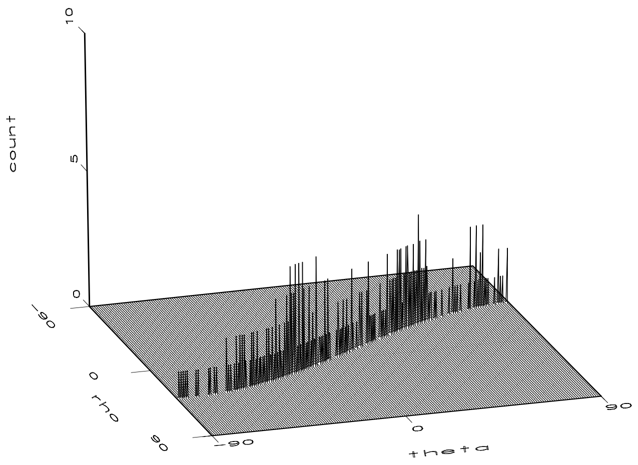

Figure 6: The decomposed Hough transform of the image in Figure 1 using a noise point as the distinguished point.

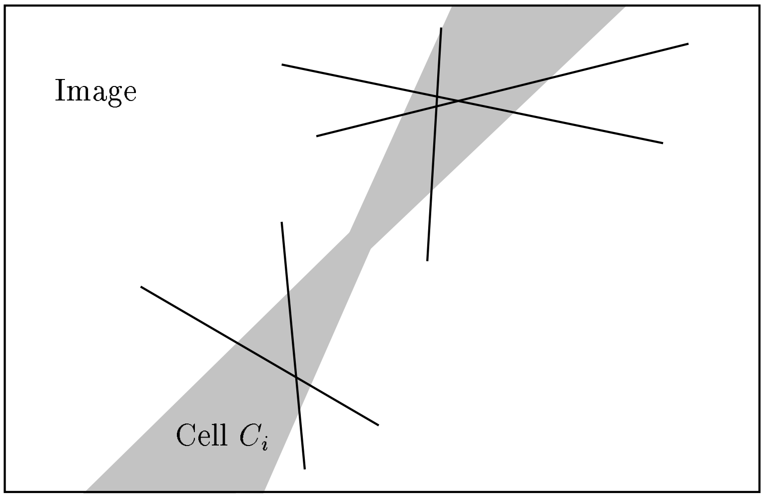

Figure 7: For the case of lines, the range of points in image space that are consistent with a cell in a discretized ρ-θ parameter space is the intersection of two cones. This figure shows an exaggeration of this effect. Due to this effect, implementations that discretize the Hough space may accumulate significant scores in cells even when no line is close to the parameters represented by the cell.

lie on the Hough curve, but the transform points will remain close to this curve.

This decomposition has two very useful properties, both of which derive from the fact that Hough transform is now constrained to lie on (or near) the Hough curve. First, since the Hough curve is 1-dimensional, it is much easier to search than the full Hough space. If we can find a parameterization of the Hough curve in one variable (i.e. $(\omega_1(t), ..., \omega_N(t))$ for an N-dimensional Hough space), we need only search in one dimension (i.e. t) to find the peaks. Second, it is now much easier to model localization error accurately. This will be accomplished by determining tight bounds on the range that a set of points can map to in Hough space.

5 Detecting lines

Let's first consider the case of detecting lines. If we use the ρ-θ parameterization for lines, we can simply parameterize the Hough curve by θ, since ρ is a function[4] of θ. We can thus search the Hough curve by varying θ.

Of course, curve detection is complicated by localization error. Typically, Hough transform implementations have assumed that there is no localization error or that the discretization of parameter space was coarse enough such that a single bin would catch the points from a line. A coarse discretization of parameter space can lead to false positives. Consider the image points that may accumulate in a bin in a discretized

[4] In the strict sense that for each θ, there is exactly one corresponding ρ.

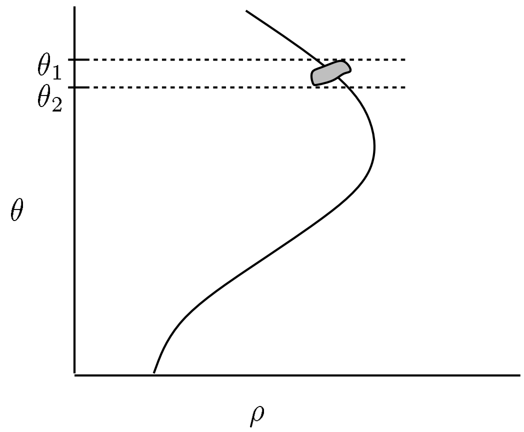

Figure 8: A pair of points (with corresponding error boundaries) maps to an error cloud in the parameter space that intersects the Hough curve. Projecting this error cloud onto the Hough curve yields bounds on the possible location of the line that passes through the correct locations of the points.

ρ-θ Hough space. If the bin was infinitesimal, these points would have to lie on a line in the image space. Since it is not, there is some range in ρ-θ that these points belong to. Rectangular bins in Hough space yield two intersecting cones in the image space that the points mapping to that bin may lie in. (Figure 7 shows an exaggerated example.) In the wide regions of the cones, even a line that is perpendicular to the direction of the cones may contribute several pixels to the bin. If a few lines go through this cone, the accumulation of votes may be significant, even though no line has parameters that are even close to those of the bin.

Now, consider the subset of ρ-θ space that a pair of edge points maps to under given error conditions. Call this subset the *error cloud* of the set of points. Ideally, we would determine how many error clouds intersect each point of the Hough space. For any curve, this would tell us how many of the pairs of points there are such that the error boundaries of both of the points intersect the curve, which is exactly the information we want. This isn't done because it isn't practical to compute this information. For the subproblems we now examine, we can easily compute a good approximation.

We can project the error clouds onto the θ-axis and determine how many of these projections intersect at each $\theta = t$, where t is arbitrary. This provides a bound on the maximum number of error clouds that can intersect at any point on each line, $\theta = t$. Since we can parameterize the Hough curve in θ and the extent of each error cloud does not stray far from the Hough curve, this is a good approximation to the exact value.

If square error boundaries are used to determine the minimal and maximal possible θ that two points could yield, we need only consider the corners of the squares in

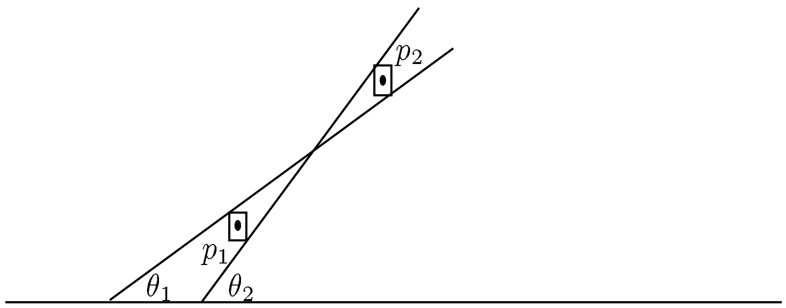

Figure 9: For any two points, we can determine bounds on the range of θ by considering lines that pass through the boundaries of their possible localization error.

determining the minimal and maximal θ values. See Figure 9. This allows us to easily project each Hough curve onto the θ-axis.

Finding the peaks on the Hough curve can now be implemented in two ways. We could simply discretize the parameterized curve and perform voting by incrementing the bins consistent with each θ. Alternately, we could sort the minimal and maximal θ points of each error cloud and use a sweep algorithm. This method would examine the extremal points in order and keep a counter that is incremented each time we hit a minimal point and decremented each time we hit an maximal point. If the counter reaches a large enough value, then a line has been found which passes through (or close to) many points. Since we now explicitly account for localization error and use accurate peak finding techniques, the lines are found with high accuracy.

6 Circles

Let's now turn to the detection of circles. We can parameterize the space of circles by the coordinates of the center of the circle and the radius, so there are three parameters: (x_c, y_c, r). For this case, the optimal decomposition will use $j = N - 1 = 2$ distinguished points.

If we treat the radius as a signed quantity (positive when the center is to the left of the segment connecting the two distinguished points and negative otherwise), we can use it to parameterize the Hough curve in one variable. It would be problematic to use x or y to parameterize the Hough curve, since r cannot always be expressed as a function of x or y, in these cases. (Vertical and horizontal lines, respectively, cause the problems.) Alternately, we could use the signed distance of the center of the circle from the segment connecting the distinguished points. In this case, the Hough curve is a straight line in the Hough space.

We now want to determine error bounds on the signed radius given three points and their localization error boundaries. Recall that the center of the circle passing through three points is the point where the perpendicular bisectors of the segments

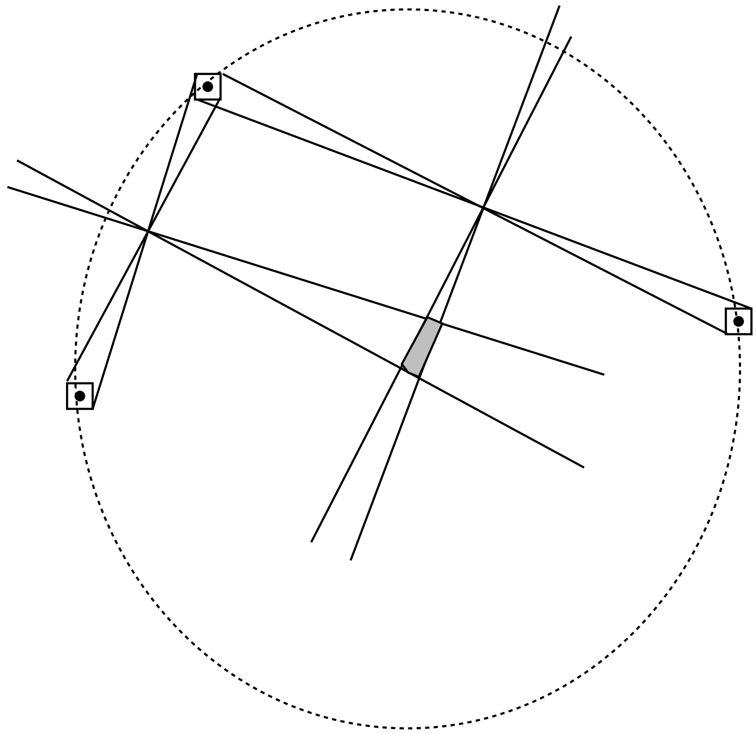

Figure 10: We can determine bounds on the position of the center of the circle by examining the perpendicular bisectors of segments between the error boundaries. Note that this figure illustrates only changes in the orientation of the perpendicular bisector. The perpendicular bisectors may also translate a small amount.

between the points meet. We can thus determine bounds on the location of the centers of the circles passing through the error boundaries of these points by examining the set of points at which two of the perpendicular bisectors of the segments can meet (see Figure 10). The maximum and minimum radius can easily be determined by examining the extremal points of this set. We can now use these bounds to implement either of the peak finding methods described in the previous section.

Note that parameters corresponding to large circles will accumulate more votes from noise points than parameters corresponding to small circles due to their larger perimeter. We may thus want to require that circles have some percentage of their perimeter appearing in the image to be output as a hypothesis. This will improve performance in the case where we want to recognize circles of different sizes, because we will have to set our ϵ such that the small circles will be found, but a low ϵ will allow circles with large radii to surpass the threshold even if a small fraction of the perimeter is present in the image.

7 Complexity

This section determines the computational complexity of the algorithms described in this paper. Let's first determine the number of distinguished sets that we must examine to achieve high accuracy. We'll assume that we only need to find curves that comprise some fraction ϵ of the total points in the image. The probability that a single set of j random points lie on a particular curve can be bounded by:

$$p_0 \geq \frac{\binom{\epsilon n}{j}}{\binom{n}{j}} \approx \frac{(\epsilon n)^j}{n^j} = \epsilon^j$$

since we must have $\binom{\epsilon n}{j}$ distinguished sets that lie on the curve among the $\binom{n}{j}$ possible distinguished sets. If we take t such trials, the probability that all of them will fail for a particular curve is bounded by:

$$p \leq (1 - p_0)^t \approx \left(1 - \epsilon^j\right)^t$$

For each curve, we thus have probability no worse than p that we will not examine a set of distinguished points that is a subset of the curve in t trials. Since conservative clustering techniques are used, we can assume that any trial examining a correct set of distinguished points will lead to the identification of the curve.

We can now choose an arbitrarily small probability of failure and determine the number of trials necessary to guarantee this accuracy:

$$\left(1 - \epsilon^j\right)^t \leq \delta$$

Solving for t yields:

$$t \log(1 - \epsilon^j) \geq \log \delta$$

$$t_{min} \approx \frac{\log \frac{1}{\delta}}{\epsilon^j}$$

The number of trials necessary is thus dependent on the accuracy desired (but only to a logarithmic factor,) the fraction of image points which must comprise the curve, and the size of the distinguished set. Note that while the number of trials is exponential in the size of the distinguished set, the number of bins we will have to increment per trial in the Hough space is inversely exponential in this size. The number of trials is not explicitly dependent on n, although it is implicitly dependent on n, since as the image complexity rises, we would need to lower ϵ to detect the same curves.

For each trial, we now have to find the peaks on the Hough curve. If we use binning, the time is dependent on how finely the Hough curve is discretized. If there are α bins, we need to increment $O(\alpha)$ bins per trial per point, yielding $O(n\alpha)$ time

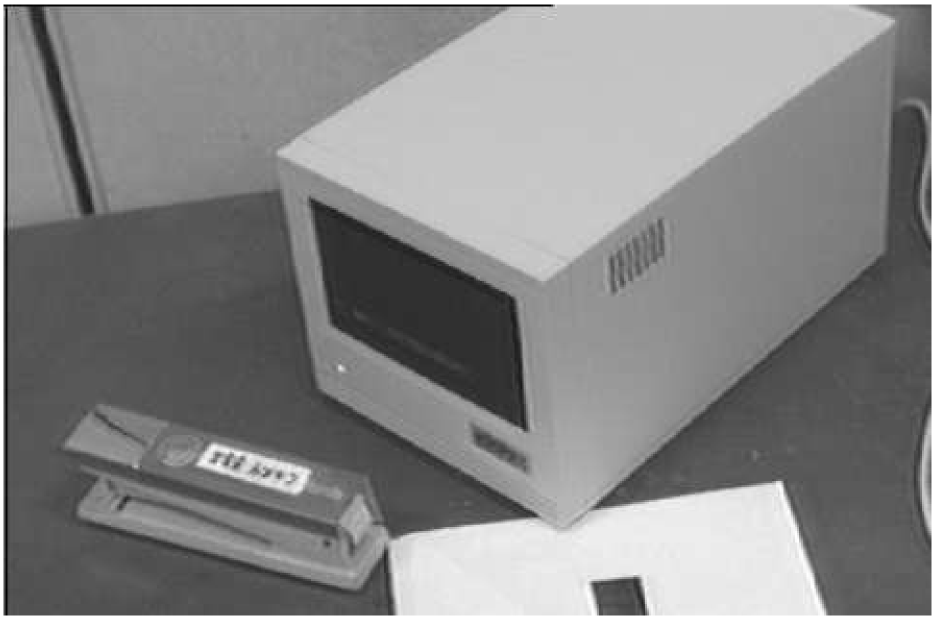

Figure 11: The original image for the first test case.

per trial. The total time requirement is thus $O(\frac{n\alpha\log\delta}{\epsilon^j})$ or simply $O(n)$ when measured by the size of the input.

If we use the sweep algorithm, we must sort the $O(n)$ maximal and minimal points of the error clouds, requiring $O(n\log n)$ time per trial. Processing the sorted points requires $O(n)$ time. We thus require $O(\frac{n\log n\log\delta}{\epsilon^j})$ total time or $O(n\log n)$ when measured by the size of the input.

In both of these methods, we examine each of the $O(n)$ edge points that we could combine with the distinguished set to form a set of N points to map into Hough space. We could use random sampling techniques among these points similar to [Bergen and Shvaytser, 1991] to reduce the number of edge points that we need to examine to $O(1)$. However, the number of samples that must be taken, while bounded by a constant, is enormous (on the order of 10^7 for reasonable values of ϵ, δ, and an additional accuracy parameter.) Since the number of samples is far greater than the population for most applications, this method is not practical.

8 Results

These techniques have been applied to real images to test their efficacy. Figure 11 shows an image that was used to test the line detection techniques. Figure 12 shows the edges found in this image. These edges were determined to sub-pixel accuracy using a version of the Canny edge detector [Canny, 1986].

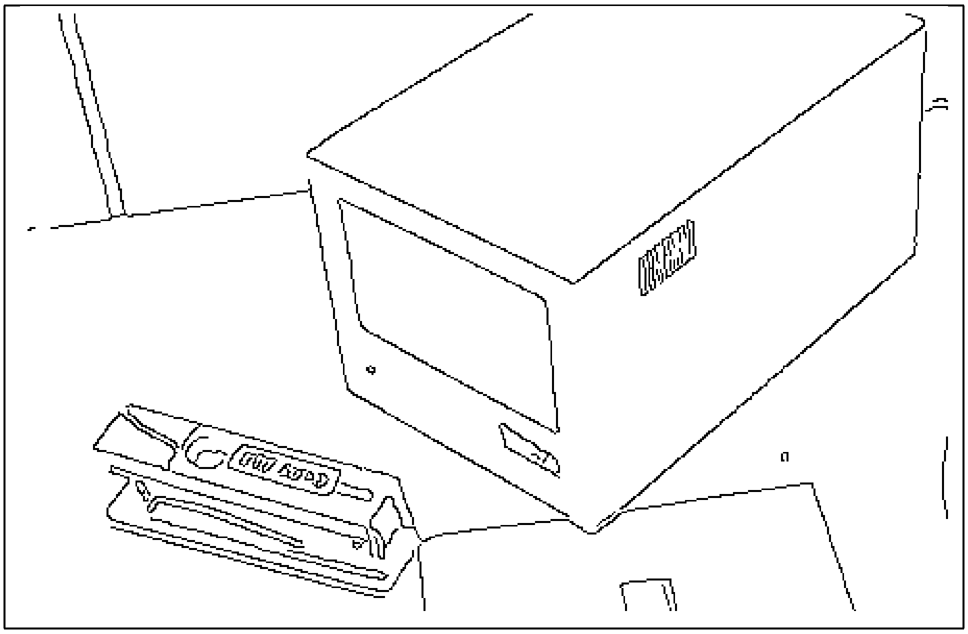

Figure 12: The edges found in the first test case.

For tests on this image, square error boundaries were used such that the true location of the point was specified to be within 0.25 pixels of the measured location in each direction. The parameterized Hough curve was discretized to find the peaks along it. Figure 13 shows the lines found in the image with $\epsilon = .01$ and $\delta = .01$. All of the long lines in the edge image were found. Figure 14 shows the lines found in the image with $\epsilon = .004$. In this case, even short lines were found.

Figure 15 shows an image that was used to test the circle detection techniques. This image is an engineering drawing that has been scanned. For this reason, it was not possible to determine the location of edge pixels to sub-pixel accuracy. In addition, the presence of small and dashed circles and the clutter in the image make this a difficult test case.

Figure 16 shows the circles found with $\epsilon = .04$. While all of the large circles were found, the small and dashed circles did not comprise a large enough fraction of the image to be found. With $\epsilon = .008$, the implementation finds a number of circles, some of which are not perceptually obvious. Figure 17 shows the circles found for this case that are perceptually salient. The implementation had difficulty finding both of the dashed circles with the same center since they were so close together. The circles shown consist of the top half of one of the circles and the bottom half of the other. This may be a result of the circles not being exactly circular in the image. Figure 18 shows circles that were found that are not perceptually salient. Note that in each case the pixels found compose most of the perimeter of a circle and thus if we want

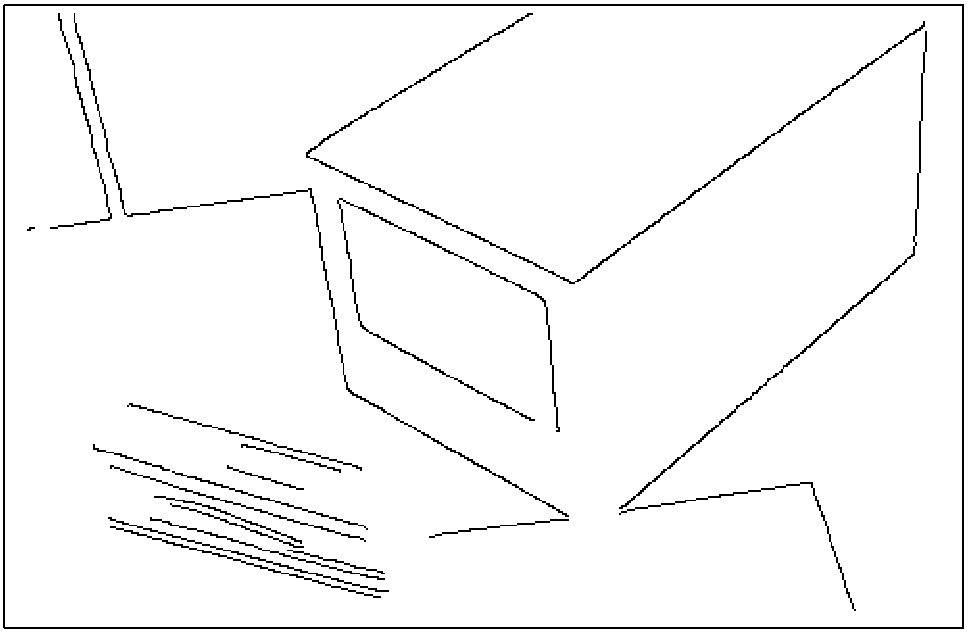

Figure 13: The lines found in the first test case with $\epsilon = .01$.

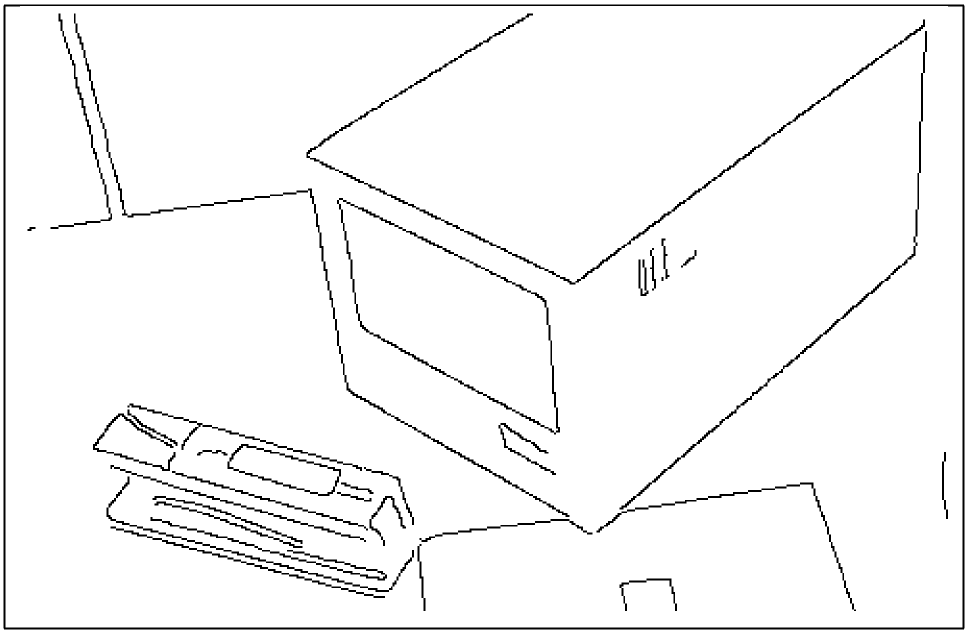

Figure 14: The lines found in the first test case with $\epsilon = .004$.

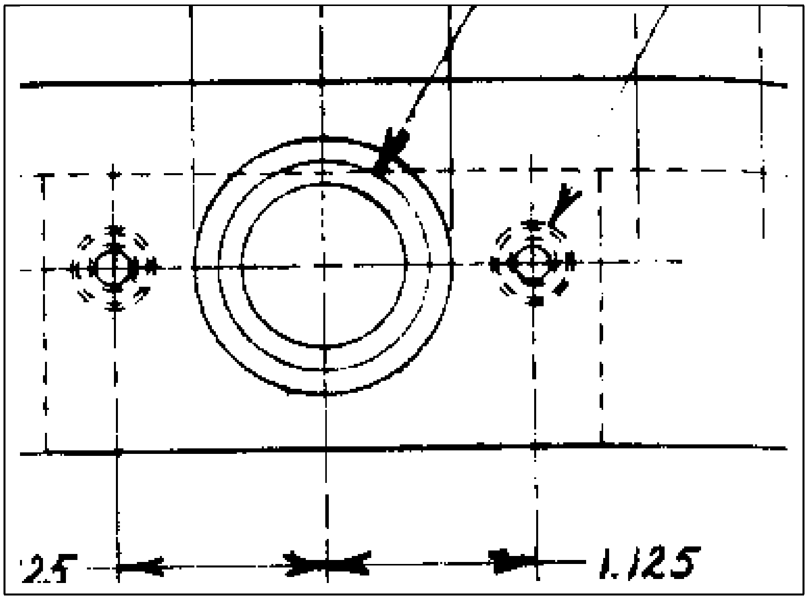

Figure 15: An engineering drawing use to test the circle detection techniques.

to find small and/or broken circles it is hard to rule out these circles without using additional information.

9 Related Work

This section discusses some related work in this area. For a more complete review of Hough transform techniques see [Illingworth and Kittler, 1988, Leavers, 1993].

Califano *et al.* [1989] use generalized neighborhoods which examine multiple image windows to exploit long distance information in Hough transform techniques. Lateral inhibition is used in a connectionist-like framework to improve the accuracy. Additionally, the concept of the radius of coherence of a point is introduced to improve the efficiency of the system.

Xu *et al.* [1990] describe the Randomized Hough Transform. For an N parameter curve, they pick random sets of N points, which they map into points in the parameter space. They accumulate such points until a sufficient peak is found or some threshold number of sets have been examined. Xu and Oja [1993] give a robust stopping criterion for the procedure and they approximate the time required by the algorithm by modeling it as a generalized Bernoulli process. Their method uses linked-lists and/or hash tables as the storage method when finding clusters in the Hough space.

Liang [1991] describes a Hough transform technique that maps sets of image points to single points in parameter space by fitting curves to the points in small neighborhoods of the image. This method allows fast implementation and low storage, but

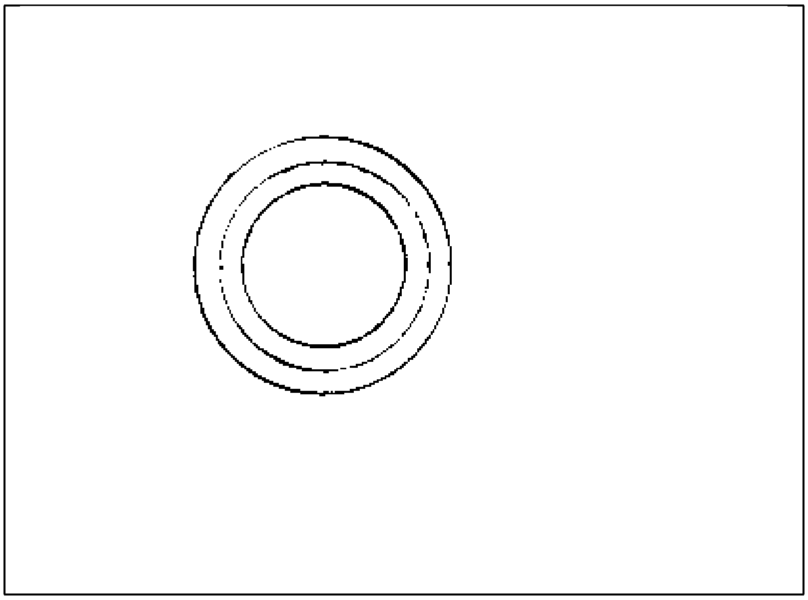

Figure 16: The circles found with $\epsilon = .04$.

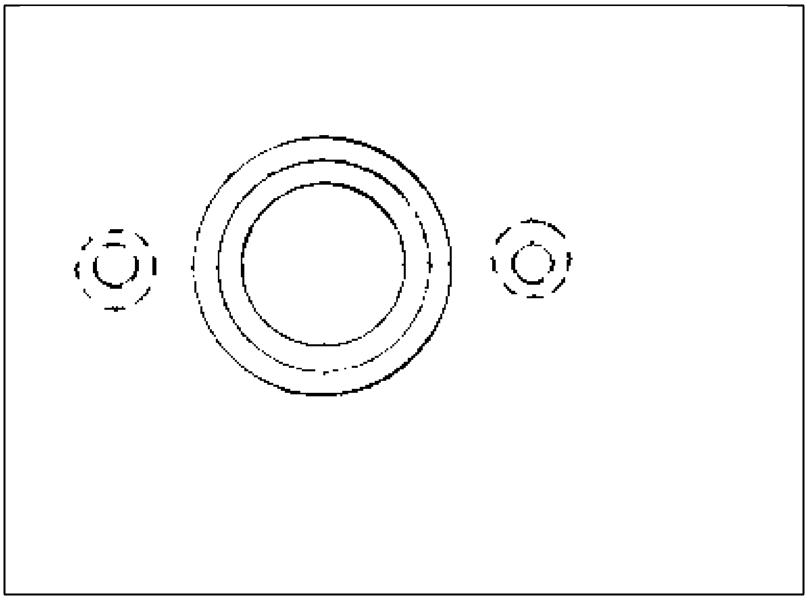

Figure 17: Perceptually salient circles found with $\epsilon = .008$.

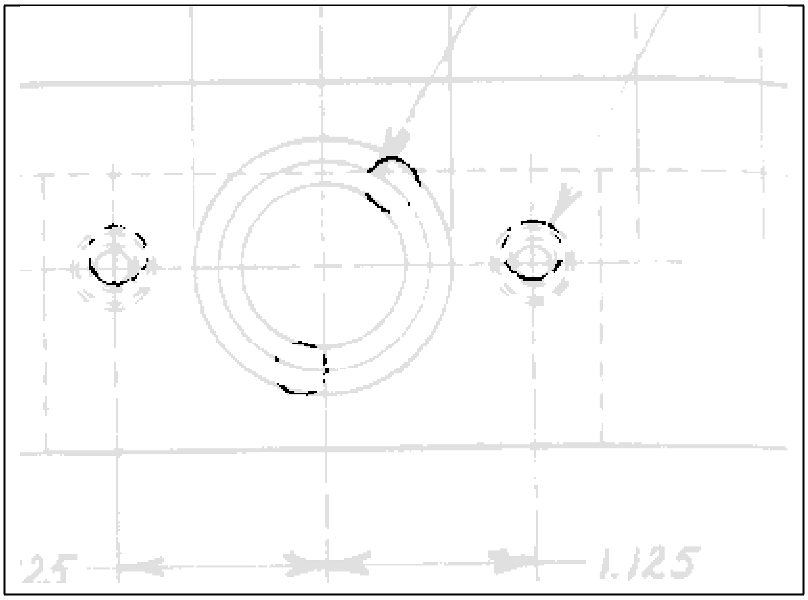

Figure 18: Perceptually insalient circles that surpassed the threshold with $\epsilon = .008$. The original image is drawn in lightly to show how these arose.

will have trouble identifying curves when image noise prevents accurate curve fitting. A list is used to accumulate points in the Hough space.

Bergen and Shvaytser [1991] give a theoretical analysis of the use of randomization techniques to speed up the Hough transform, but no implementation is described. They achieve constant time if only curves that represent some fraction of the total edge points need be found and the approximation to the Hough transform is allowed to be in error by a fractional parameter all of the time and more that this fractional parameter with frequency δ. While the computational complexity achieved by this system is $O(1)$, the constant number of random samples that must be examined is usually more than examining each possibility once and is thus not practical.

Kiryati *et al.* [1991] describe a different method of using randomization. They simply subsample the edge points in the image and then use the standard Hough transform. This is similar to the randomization technique used by other authors, but they do not consider mapping sets of image points into the parameter space. Their analysis implies that the speed can be improved while performance is degraded little.

Leavers [1992] describes the Dynamic Generalized Hough Transform. He uses the technique of mapping N image points into 1 parameter point. Furthermore, a point is used to 'seed' the transformation, similar to our distinguished sets. Leavers uses the heuristic that only edge points that are connected to other edge points are used as this seed. Clustering is performed efficiently by projecting the clusters onto each of the parameter space axes.

10 Summary

This paper has shown that the Hough transform for curve detection can be decomposed into small subproblems. Each of these subproblems consists of examining curves that pass through a distinguished set of edge pixels. Since each of these subproblems examines only a 1-dimensional manifold of the parameter space, efficient and accurate solutions exist. The efficiency of the algorithm is $O(n)$ or $O(n \log n)$ depending on the peak finding method used. Experiments have been performed that show the effectiveness of the method.

A Parameterizations

This appendix describes parameterizations for lines, circles, and ellipses that can be used in Hough transform implementations.

A.1 Lines

One possible parameterization for lines is the standard slope-intercept representation:

$$y = mx + b$$

While this parameterization has the advantage that the points in the image space map to lines in the Hough space, it has the disadvantage that the ranges of m and b are unbounded, since we may have vertical or horizontal lines.

Duda and Hart [1972] proposed to use the following parameterization instead:

$$x \cos\theta + y \sin\theta = \rho \tag{3}$$

where θ is the orientation of the line and ρ is the distance of the line from the origin. While ρ is theoretically unbounded, for any image we can bound ρ by the maximum distance of any image point from the origin. This parameterization maps points in image space to sinusoids in Hough space.

Given two points (x_1, y_1) and (x_2, y_2) we can solve for θ and ρ as follows:

$$\theta = \tan^{-1} \frac{x_1 - x_2}{y_2 - y_1}$$
$$\rho = x_1 \cos\theta + y_1 \sin\theta$$

For a distinguished point, (x_d, y_d), the Hough curve is given by $x_d \cos\theta + y_d \sin\theta = \rho$. In addition, ρ is clearly a function of θ, so we can parameterize the Hough curve by θ.

A.2 Circles

The standard parameterization for circles is (x_c, y_c, r), where (x_c, y_c) is the center of the circle and r is the radius. The parameterization function is:

$$(x - x_c)^2 + (y - y_c)^2 = r^2$$

Given (x_1, y_1), (x_2, y_2), and (x_3, y_3), the circle that passes through them has parameters:

$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \frac{\begin{bmatrix} y_3 - y_1 & y_1 - y_2 \\ x_1 - x_3 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} x_2^2 + y_2^2 - x_1^2 - y_1^2 \\ x_3^2 + y_3^2 - x_1^2 - y_1^2 \end{bmatrix}}{(x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1)}$$

$$r = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}$$

A slightly different parameterization that is useful in the context of Hough curves is (x_c, y_c, d), where d is the distance of the center from the segment connecting the two distinguished points. If we choose d to be positive when the center of the circle is to the right of the segment and negative when it is to the left[5], we can reparameterize the curve in one variable by d. In this case the Hough curve forms a straight line in the Hough space.

A.3 Ellipses

Ellipses are typically parameterized by (x_c, y_c, a, b, θ), where (x_c, y_c) is the center of the ellipse, a and b are the lengths of the major and minor axes, and θ is the orientation of the major axis. Forbes [1987] has determined a stable method to solve for the parameters (a description was found by this author in [Leavers, 1992]). Given $(x_i, y_i), i = 1...5$ we can determine the coefficients in the following equation and solve for (U, V, R, S, T) using Gaussian elimination:

$$x^2 + y^2 - U(x^2 - y^2) - 2Vxy - Rx - Sy - T = 0$$

We then solve for (x_c, y_c, a, b, θ) in the following equations:

$$e = \frac{b}{a}$$

$$U = \cos 2\alpha \frac{1 - e^2}{1 + e^2}$$

$$V = \sin 2\alpha \frac{1 - e^2}{1 + e^2}$$

[5]If the segment is horizontal, then left and right of the segment have no meaning. In this case, we take above be positive and below to be negative.

$$
\begin{aligned}
R &= 2x_c(1-U) - 2y_cV \\
S &= 2y_c(1-U) - 2x_cV \\
T &= \frac{2a^2b^2}{a^2+b^2} - \frac{x_cR}{2} - \frac{x_cS}{2}
\end{aligned}
$$

References

[Bergen and Shvaytser, 1991] J. R. Bergen and H. Shvaytser. A probabilistic algorithm for computing Hough transforms. *Journal of Algorithms*, 12:639–656, 1991.

[Califano *et al.*, 1989] A. Califano, R. M. Bolle, and R. W. Taylor. Generalized neighborhoods: A new approach to complex parameter feature extraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 1989.

[Canny, 1986] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–697, November 1986.

[Cass, 1991] T. A. Cass. Polynomial-time object recognition in the presence of clutter, occlusion, and uncertainty. A. I. Memo 1302, Massachusetts Institute of Technology, 1991.

[Duda and Hart, 1972] R. O. Duda and P. E. Hart. Use of the Hough transform to detect linear and curves in pictures. *Communications of the ACM*, 15:11–15, 1972.

[Forbes, 1987] A. B. Forbes. Fitting an ellipse to data. Report DITC 95/87, National Physical Laboratory, December 1987.

[Hough, 1962] P. V. C. Hough. Method and means for recognizing complex patterns. U. S. Patent 3069654, 1962.

[Illingworth and Kittler, 1988] J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, 44:87–116, 1988.

[Kiryati *et al.*, 1991] N. Kiryati, Y. Eldar, and A. M. Bruckstein. A probabilistic Hough transform. *Pattern Recognition*, 24(4):303–316, 1991.

[Leavers, 1992] V. F. Leavers. The dynamic generalized Hough transform: Its relationship to the probabilistic Hough transforms and an application to the concurrent detection of circles and ellipses. *CVGIP: Image Understanding*, 56(3):381–398, November 1992.

[Leavers, 1993] V. F. Leavers. Which Hough transform? *CVGIP: Image Understanding*, 58(2):250–264, September 1993.

[Liang, 1991] P. Liang. A new and efficient transform for curve detection. *Journal of Robotic Systems*, 8(6):841–847, 1991.

[Olson, 1994] C. F. Olson. Time and space efficient pose clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 251–258, 1994.

[Princen *et al.*, 1993] J. Princen, J. Illingworth, and J. Kittler. A formal definition of the Hough transform: Properties and relationships. *Journal of Mathematical Imaging and Vision*, 1:153–168, 1993.

[Xu and Oja, 1993] L. Xu and E. Oja. Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities. *CVGIP: Image Understanding*, 57(2):131–154, March 1993.

[Xu *et al.*, 1990] L. Xu, E. Oja, and P. Kultanen. A new curve detection method: Randomized Hough transform (RHT). *Pattern Recognition Letters*, 11:331–338, May 1990.