EFFECTIVE LINE DETECTION WITH ERROR PROPAGATION

Yonghong Xie *Qiang Ji**

Computer Science Department, University of Nevada at Reno
*Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute

ABSTRACT

In this paper, we introduce a new Hough Transform aimed at improving curve detection accuracy and robustness as well as computational efficiency. Robustness and accuracy improvement is achieved by analytically propagating the errors with image pixels to the estimated curve parameters. The errors with the curve parameters are then used to determine the contribution of pixels to the accumulator array. The computational efficiency is achieved by choosing best-distinguished pixels and by performing progressive detection. The detection approaches were given for line and circle. The concept can be applied to other curves, such as circle and ellipse. The experiments on line detection show improved performance with our technique.

1. INTRODUCTION

Detecting geometric primitives in images is one of the basic tasks of computer vision. The Hough Transform (HT) [4] and its extensions constitute a popular and flexible method for extracting geometric curves. The principal concept of the HT is to define a mapping between an image space and a parameter space. Each edge point in an image is transformed by the mapping to determine cells in the parameter space whose associated parameters are such that the defined primitive passes through the data point. The chosen cells are accumulated and after all the points in an image have been considered, local maxima in the accumulator correspond to the parameters of the specified shape.

However, HT is computationally expensive and memory consuming. For this reason in recent years numerous approaches have been proposed to improve it [1,2,5,9,12,13]. While these approaches improved the computational performance of HT to various degrees, they did not take into account the localization and discretization errors, which are present in the image edge pixels and affect the accuracy and robustness of curve detection. Many techniques have been proposed to deal with these errors. But they have not been fully satisfactory in both robustly locating noisy curves and eliminating false positives from consideration. In this paper, we consider techniques to improve the accuracy and robustness as well as computational efficiency of curve detection using the HT. Specifically, we improve upon the constrained HT scheme developed by Olson [10]. The error propagation techniques introduced by Ji and Haralick [7] are applied to improve accuracy and precision for curve detection, while the techniques proposed by Olson and others are used to improve computational complexity.

2. PREVIOUS WORKS

Researchers have long realized the limitations of the standard HT scheme and have proposed different schemes to improve the detection performance of the HT in relation to localization error in the image and discretization error in both the image and the parameter space.

Stephens [11] formulated a variant of the Hough transform in terms of maximum likelihood estimation. A Probability Density Function (PDF) for the features is used, which has a uniform component modeling the correspondence errors and a component that falls off as a Gaussian with the distance from the curve to model the measurement errors. While this method yields correct propagation of localization error in terms of a Gaussian error distribution, usually it is computationally expensive.

Breuel [3] described a line detection technique related to the HT that searches hierarchical subdivisions of the parameter space using a bounded error model and thus avoids some of the problems of the accumulator method. In this technique, the parameter space is divided into cells that are tested to determine whether they can contain a line that passes within the bounded localization error of a specified number of pixels. If the cell cannot be ruled out, the cell is divided and the procedure is repeated recursively. This continues until the cells become sufficiently small, at which point they are considered to be lines satisfying the output criterion.

Olson [10] modified a formal definition of the HT that allows the localization error to be analyzed appropriately. Under this definition, it was shown that the mapping of pixel sets (rather than individual pixels) into the parameter space did not, by itself, improve the accuracy or efficiency of curve detection. He then considered a new method where the HT is decomposed into several constrained subproblems, each of which examines a subset of the parameter space by considering only those pixel sets that include some distinguished set of pixels. The examination of these subproblems allows, first, to propagate the localization error efficiently and accurately into the parameter space, and second, to use randomization techniques to reduce the complexity of curve detection, while maintaining a low probability of failure. However, the error propagation applied in his work is a) heuristic, b) not systematic, and c) the voting kernel is a top-hat function instead of continuous function.

Ji and Haralick [8] introduced a Bayesian updating scheme that systematically ties the uncertainties computed for each point to its contribution. The contribution of each point to a (θ, ρ) is proportional to its likelihood. The proposed scheme is based on an analytical propagation of input error. Their results showed that the uncertainty of a feature point depends on a) the input perturbation; b) its relative spatial location to the Hough coordinate system; c) edge detector; d) line representation scheme. Their technique, however, suffers from computational complexity. The work presented in this paper combines the work of Olson and Ji. In the sections to follow, we first discuss Ji's error propagation scheme and then show how it can be combined with Olson's method for efficient and accurate curve detection.

0-7803-6725-1/01/$10.00 ©2001 IEEE

3. OVERVIEW OF ERROR PROPAGATION

Error propagation for a computer vision algorithm is concerned with quantitatively characterizing how to characterize the output perturbation as a function of input perturbation and the algorithm. For many computer algorithms, input and output are implicitly related through a non-linear optimization function F. In other words, the output is obtained by minimizing an appropriate criterion function $F(\hat{\Theta}, \hat{X})$, where $\hat{X}$ represents the observed input vector while $\hat{\Theta}$ represents the estimated output vector.

Let $\Sigma_{\Delta\Theta}$ be the covariance matrices of the estimated curve parameters. Based on Haralick's covariance propagation theory [6], we have

$$\Sigma_{\Delta\Theta} = 2\sigma^2 \left[\left(\frac{\partial g}{\partial \Theta} \right)^t \right]^{-1} \tag{1}$$

where σ^2 represents the image error and $g(X, \Theta) = \partial F(X, \Theta)/\partial\Theta$. Detailed derivations may be found in [7].

Given a line expressed as

$$F(x, y, \Theta) = x\cos\theta + y\sin\theta - \rho \tag{2}$$

A least-squares line fitting amounts to finding the line parameter $\Theta = (\theta, \rho)$ that best fits a set of points $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_N, \hat{y}_1, \ldots, \hat{y}_N)$. Error propagation is concerned with estimating the perturbation of $\hat{\Theta}$, a least-square estimate of Θ, given the perturbation with $\hat{X}$. $\hat{\Theta}$ is obtained by minimizing

$$\varepsilon^2 = \sum_{i=1}^{N} (\hat{x}_i \cos\hat{\theta} + \hat{y}_i \sin\hat{\theta} - \hat{\rho})^2 \tag{3}$$

Hence, $\partial g/\partial\Theta$ can be computed from equation (2) as follows

$$\frac{\partial g}{\partial \Theta} = 2 \sum_{n=1}^{N} \begin{pmatrix} k_n^2 & -k_n \\ -k_n & 1 \end{pmatrix}$$

where $k_n = x_n \sin\theta - y_n \cos\theta$.

$$Let \quad \mu_k = \frac{\sum_{n=1}^{N} k_n}{N} \quad and \quad S_k^2 = \sum_{n=1}^{N} (k_n - \mu_k)^2$$

we have

$$\Sigma_{\Delta\Theta} = \sigma^2 \frac{\begin{pmatrix} N & \Sigma k_n \\ \Sigma k_n & \Sigma k_n^2 \end{pmatrix}}{N\Sigma k_n^2 - \Sigma k_n \Sigma k_n} = \sigma^2 \begin{bmatrix} \frac{1}{S_k^2} & \frac{\mu_k}{S_k^2} \\ \frac{\mu_k}{S_k^2} & \frac{1}{N} + \frac{\mu_k^2}{S_k^2} \end{bmatrix} \tag{4}$$

Detailed derivation of equation (4) may be found in [7].

Geometrically, k_n can be interpreted as the signed distance between a point (x_n, y_n) and the point on the line closest to the origin. Hence, S_k^2 represents the spread of points along the line and μ_k is the mean position of the points along the line. From equation (15), we can conclude that the error with estimated line parameters not only depends on input pixel noise σ^2, the number of points that participate in the fitting, but also depends on the spread of points and their centroid. Of particular interest is the centroid of the points. The closer the points centroid to the origin, the smaller the error is. This implies that we can translate the coordinate system to minimize μ_k^2. In other words, coordinate system matters for the error of ρ.

4. OVERVIEW OF PROPOSED SCHEMES

In this section we describe the improvements we made for the line curve detection and the algorithm that is used for line segments detection. In the proposed approach we combined Olson's constrained HT scheme with Ji's error propagation scheme discussed in previous section to achieve robust, accurate and efficient curve detection.

Specifically, the proposed new HT includes the following features.

Rather than considering each pixel separately, we adopt Olson's method which considers a set of pixels with some cardinality k. For each such set, the curves that pass through every pixel in the set (or through the bounded error region of every pixel in the set) are determined and the parameter space scores accumulate accordingly. The primary benefit of using this technique is that each mapping is into a smaller subset of the parameter space. If $f(X, \Omega)$ is an N parameter function, then, in the errorless case, N nondegenerate edge pixels map to a single point in the parameter space. In this case, the accumulator method needs to increment only a single bin in the parameter space for each set, rather than the bins covering an N - 1 dimensional manifold for each edge pixel. Of course, we need not use sets with cardinality N but we could use any size $k > 0$. If $k <= N$, then each nondegenerate pixel set maps into an N - k dimensional manifold in the parameter space.

To further improve the efficiency, each pixel set contains a subset of j distinguished pixels. Distinguished pixels lie on the curves to be detected. This limits the pixels to vary to $k - j$.

To gain the maximum decomposition of the problem, the edge pixel number j for a distinguished set is taken as large as possible, but note that j cannot be greater than k and k must be less than N for efficiency reasons. The optimal cardinality for the distinguished set is thus $j = k - 1 = N - 1$. For example, $k = 2$, $j = 1$ for line and $k = 3$, $j = 2$ for circle. This maximum decomposition of the problem allows each of the subproblems to be processed quickly and the best efficiency to be achieved by the overall algorithm with this decomposition when randomization is used.

In summary, to improve robustness and accuracy, we propose the use of error propagation technique to analytically characterize the uncertainty of the estimated curve parameters and determine the contribution of each pixel set to curve parameters according to the uncertainty of the parameters. To improve computational complexity, we propose to examine a set of pixels at a time and use distinguished pixels to further reduce search space. In the sections to follow, we will discuss our schemes in more details.

4.1 Best-distinguished pixels

We define best-distinguished pixels are the pixels that are most likely located on the curve and produces the most accurate curve parameter estimation. By choosing best-distinguished pixels, only a small part of image pixels will be examined as distinguished pixels. This will reduce the computation time significantly.

In Olson's method, every pixel was chose as a distinguished pixel. For the image with many pixels, this is time consuming. If we can find the best-distinguished pixel on the line before we calculate the parameters for this line, we do not need to calculate line parameters for every pixel and then find the most likely line parameters. We can simply use the most likely distinguished pixel to calculate the parameters for this line. This will reduce the computation time greatly.

For line fitting, the end points of a line are the best-distinguished pixels. We can determine the end points of a line by checking the connectivity around a pixel. If a pixel connects to only one other pixel or it is an arc point where two line segments meet and form a vertex, we think this pixel probably is an end point of a line.

Figure 1 shows an example image with two line segments amid 2% random noise. A and C are two end points of Line 1 and B is the midpoint of line segment AC. Figure 2 shows constrained HT of lines in Fig. 1 with Ji's error propagation. 2a shows a case where a noise pixel was used as the distinguished pixel, 2b shows a case where a pixel on Line 1 was used as the best-distinguished pixel, and 2c shows a case where a pixel on Line 2 was used as the best-distinguished pixel. Note that peaks are present where appropriate, but that no peak is present when a noise pixel was used as the distinguished pixel.

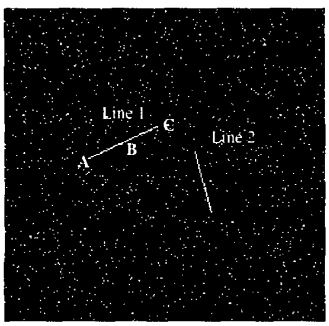

Figure 1. Line 1 and Line 2 with 2% noise.

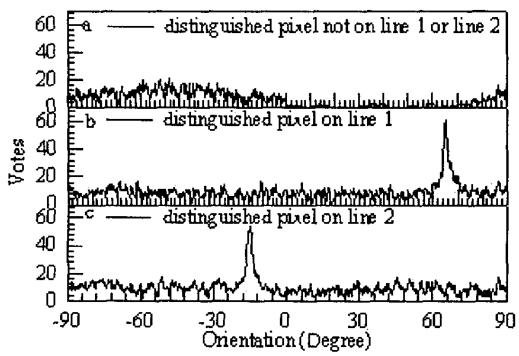

Figure 2. One dimensional HT accumulator of the lines in Figure 1 with Ji's error propagation

4.2 Voting kernel

In this paper, we used a voting kernel that is a smooth function of differences in parameter values for updating accumulator, which is different from the top-hat voting kernel in Olson's work. Assume $\hat{\Theta}$, a curve parameter vector estimated from a feature point, is distributed as $\hat{\Theta} \sim N(\Theta, \Sigma_{\hat{\Theta}})$, where Θ is a quantized parameter vector. Given $\hat{\Theta}$ and the covariance matrix of the estimated curve parameters $\Sigma_{\hat{\Theta}}$, which can be calculated from equation (4) for line, we use following function to calculate voting kernel $P(\hat{\Theta}|\Theta)$:

$$P(\hat{\Theta} \mid \Theta) = \exp^{-\frac{1}{2}(\hat{\Theta}-\Theta)^t \Sigma_{\hat{\Theta}}^{-1}(\hat{\Theta}-\Theta)} \quad (5)$$

It is clear from equation (8) that given each $\hat{\Theta}$ and its covariance matrix $\Sigma_{\hat{\Theta}}$, the bin for a Θ is updated based on $P(\hat{\Theta}|\Theta)$. The far away Θ from $\hat{\Theta}$, the smaller the likelihood is and the less contribution Θ receives from the point as shown in Figure 3.

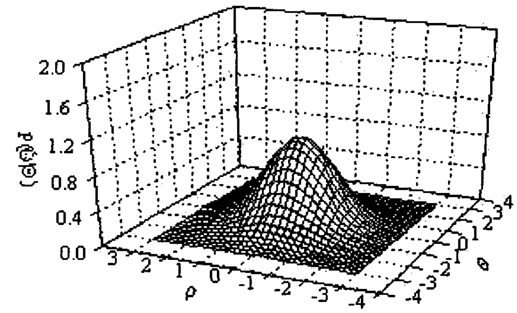

Figure 3. Illustration of the voting kernel function.

4.3 Deterministic algorithms

Based on the discussion above, the algorithms of our detection technique for line can be described as following. The input image in this research is edge image, on which black pixels represent background and white pixels are image pixels.

1) Detecting a best-distinguished pixel from the input edge pixels.
2) Find another edge pixel and pair it with the best-distinguished pixel.
3) Calculate the line parameters $\hat{\Theta}$ determined by the pair of pixels and determine the contribution of the pair of pixels to other line parameters Θ using voting kernel function. (equation 5)
4) Repeat until all edge pixels have been used.
5) Find HT parameters with maximum vote. If the maximum vote is less than the vote threshold then start over.
6) Otherwise, a line primitive has been detected.
7) If selected primitive parameter (such as segment length for line) meets the criteria add it to the output list.
8) Remove the pixels on the primitive from input image.
9) Start over again.

If we are willing to allow a small probability of failure, we can use randomization to reduce the number of pixels that must be examined in step 2.

5. EXPERIMENTAL RESULTS

The proposed scheme has been applied to synthetic and real images for line to test its performance. All experiments were carried out with the following settings. A 900-bin accumulator was used for all images. We have also tested 360, 720, 1800 and 3600 bin accumulators. Considering both computational time and accuracy, 900-bin accumulator is the best choice. Pixels within 3-pixel-wide corridor were assigned to a line. For each line that surpasses the detection threshold in each subproblem, only the parameters at which the most votes occurred were kept.

The detected line segment had no gaps bigger than 3 pixels long. The minimum accepted line length was chosen to reduce the false positives based on the noise level. The higher was the noise level, the larger was the minimum accepted line length.

Figure 4a and 4b show two synthetic images amid noise pixels that were used to test the line detection algorithm. Figure 4c and 4d show line segments that were detected. All of the lines were found in the image.

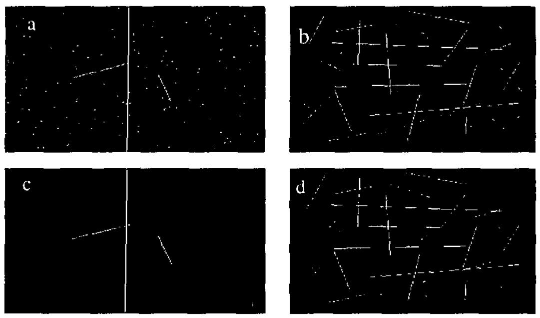

Figure 4. The synthetic images

Figure 5 shows some real world images that were used for line detection, with the detected line segments superimposed on the original images. All of the long lines in the images were found. But some short lines and/or curve lines were not found. The reason for this is that we set the minimum accepted line length to a proper value so that we did not have too many false positives.

CONCLUSIONS

In this paper, we have proposed a new curve detection method aimed at improving accuracy and robustness as well as computational efficiency. Comparing with Olson's approach, this method has following improvements:

- More accurate error propagation scheme. Error propagation is calculated by using Ji's error propagation technique, which is systematic, adaptable to noise characteristics of image and applicable to complex curve.
- Smooth voting kernel. The voting kernel is a smooth function of differences in parameter values instead of a top-hat function.
- Best-distinguished pixels. By choosing best-distinguished pixels, only a small part of image pixels will be examined as distinguished pixels.
- Progressive detection. As soon as the primitive is found we remove all pixels on this primitive from the input image.

Robustness and accuracy improvement is achieved by analytically propagating the errors with image pixels to the estimated curve parameters. The errors with the curve parameters are then used to determine the contribution of pixels to the accumulator array. The computational efficiency is achieved by choosing best-distinguished pixels and by performing progressive detection. We have given algorithms of this method for lines. The concept also can be applied for other curves, such as circle and ellipse detection. The experiments on line detection show improved performance with our technique.

REFERENCES

1. D. Ben-Tzvi, V. F. Leavers, M. B. Sandler, A Dynamic Combinatorial Hough Transform, *Proc. of 5th Int. Conf. on Image Analysis and Processing*, Positano, Italy, September 20--22, 1989, pp. 152--159.
2. D. Ben-Tzvi, M. B. Sandler, A Combinatorial Hough Transform, *Pattern Recog. Lett.*, **11**(3), 1990, 167--174.
3. T. M. Breuel, Finding lines under bounder error, *Pattern Recog.* 29(1), 1996, 167–178.
4. P. V. C. Hough, Method and Means for Recognizing Complex patterns, *U.S. Patent 3069654*, Dec. 18, 1962.
5. G. Gerig, Linking Image-Space and Accumulator-Space: A New Approach for Object Recognition, *Proc. of First Int. Conf. on Computer Vision*, London, England, June 1987, 112--117.
6. R. M. Haralick, Propagating covariance in computer vision. *Proc. of 12th ICPR*, 1994, 493--498.
7. Q. Ji and R. M. Haralick, Error Propagation for Computer Vision Performance Characterization, in *1999 Int. Conf. on Imaging Science, Systems, and Technology*, Las Vegas, June, 1999.
8. Q. Ji and R. M. Haralick, An Optimal Bayesian Hough Transform for Line Detection, in *1999 IEEE Int. Conf. on Image Processing*, Kobe, Japan, October, 1999.
9. N. Kiryati, Y. Eldar, and A. M. Bruckstein, A probabilistic Hough Transform, *Pattern Recog.* **24**(4), 1991, 303--316.
10. C. F. Olson, Constrained Hough Transforms for Curve Detection, *Computer Vision and Image Understanding*, **73**(3), 1999, 329--345
11. R. S. Stephens, Probabilistic approach to the Hough transform, *Image Vision Comput.* **9**(1), 1991, 66–71.
12. L. Xu, E. Oja, and P. Kultanen, A new curve detection method: Randomized Hough transform (RHT), *Pattern Recog. Lett.* **11**, 1990, 331--338.
13. S. Y. K. Yuen, T. S. L. Lam, N. K. D. Leung, Connective Hough Transform, *Image and Vision Computing*, **11**(5), 1993, pp. 295--301.

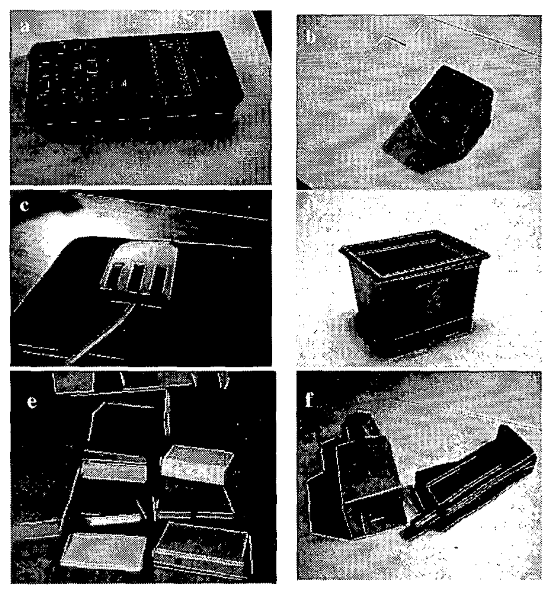

Figure 5. Real images with detected line segments superimposed.