DETERMINATION OF END POINTS AND LENGTH OF A STRAIGHT LINE USING THE HOUGH TRANSFORM

M. Atiquzzaman
Dept. of Computer Science & Engineering
La Trobe University, Melbourne 3083,
Australia.
atiq@LATCS1.lat.oz.au

M.W. Akhtar
Dept. of Electrical Engg.
King Fahd University of
Petroleum & Minerals
Dhahran 31261, Saudi Arabia.

ABSTRACT

When the Hough transform is applied to the detection of straight lines in images, it provides only the parameters of the line but not the *length* or the *end points* of the line. We propose a new algorithm for the determination of the length and the end points of a line in an image. It is very efficient in terms of computing time and does not depend on the sharpness of the peak in the accumulator array.

1 INTRODUCTION

Detection of patterns in images is an important operation in machine vision tasks. The Hough transform [1] is a powerful technique for the determination of parameterizable patterns in binary images. The advantages of the transform are its robustness to noise and discontinuities in the patterns. Examples of parameterizable patterns are straight lines, circles, ellipses, etc.

When the Hough transform is used to detect straight lines represented by the $\rho - \theta$ parameterization, it provides only the ρ and θ parameters of the straight lines. It fails to provide any information regarding the *lengths* or the *end points* of the lines. Since, many machine vision tasks require the lengths and end points of lines to accurately determine the location of objects described by the lines, it is *extremely important to determine the length and end points* of a line.

Six different algorithms [2, 3, 4, 5, 6, 7] have been proposed in the literature for finding the length and end points of a line from the Hough accumulator array. The algorithms are either highly *computation bound* or are *not robust* in detecting the end points. The algorithms described in [2, 3] are based on the projections of the line on the x or y axis. They suffer from the drawback of a single line segment being interpreted as multiple line segments. The algorithm described in [4] is based on the concept of surface fitting and is prohibitively expensive in terms of computing power. The method proposed in [5] can detect only the length of a line; it can not detect the end points.

The algorithm described in [6] can detect the length and the end points of a line, but the accuracy of the detection is *dependent on the accurate detection of the line parameters*. The algorithm described in [7] is also based on the accurate determination of the peak and hence suffers from the same drawback as in [6]. The aim of this paper is *to develop a robust algorithm to detect the length and end points of lines*. The algorithm should be independent of the accuracy with which the peak in the accumulator array can be determined.

The proposed algorithm has a time complexity of $O(1)$ which is the same as the complexity of the best available algorithm [6]. Note that the time complexity of the algorithms described in [2], [3], [4], and [5] are $O(n_f n_i + N)$, $O(n_f + N)$, $O(n_f n_i N + N^2)$ and $O(1)$ respectively. n_f, n_i, and N^2 are the number of feature points, the number of iterations, and the size of the accumulator array respectively [5]. The proposed algorithm *differs from the previously published algorithms* in its reduced computational *complexity* and *robustness* in detecting the end points. Because of its low computational complexity and high robustness, the algorithm is very suitable for implementation in *real-time* machine vision tasks. Results show that the proposed algorithm can detect the end points and the line length very accurately.

The rest of the paper is organized as follows. Section 2 introduces the notations used in the proposed algorithm which is described in Section 3. Results illustrating the accuracy of the algorithm in detecting the length and end points are presented in Section 4, followed by concluding remarks in Section 5.

2 NOTATIONS

A straight line, represented by the normal parameterization, is expressed as

$$\rho = x \cos \theta + y \sin \theta \tag{1}$$

where, ρ is the length of the normal to the line and θ is the angle of the normal with the positive x-direction. A cell $a_{\rho,\theta}$ in the accumulator array corresponds in the image plane to a bar-shaped window of infinite length, of width $\Delta\rho$, making an angle θ with the positive x-axis, and at a distance of ρ from the origin [8]. The cells $a_{\cdot,\theta_k}$ in the accumulator array, therefore, correspond to a set of parallel *bars* of width $\Delta\rho$ and each making an angle θ_k with the positive x-axis in the image plane as shown in Figure 2. The sets of cells in the different columns of the accumulator array correspond to different sets of parallel bars in the image plane.

The parameters of a line along with its length and coordinates of the end points are sometimes referred to as the *complete line segment descriptions.* The following notations will be used in describing the proposed algorithm for finding the complete line segment descriptions. To maintain consistency, we use the same notations as in [6].

$\rho_{\text{size}}, \theta_{\text{size}}$ = Size of the accumulator array

$A = \{a_{i,j}, 0 \leq i \leq \rho_{\text{size}} - 1, 0 \leq j \leq \theta_{\text{size}} - 1\}$ = Hough accumulator array.

$\rho_{\min}$, $\rho_{\max}$, $\theta_{\min}$, $\theta_{\max}$ = Minimum and maximum values of ρ and θ axes of the accumulator array.

$\Delta\rho$, $\Delta\theta$ = Resolution of the ρ and θ axes of the accumulator.

$b_{i,k}$ = The bar in the image plane corresponding to the cell $a_{i,k}$ in the accumulator array.

C_k = k-th column in the accumulator array. The column consists of the cells $a_{\cdot,k}$.

C_p = The column in the accumulator array containing the peak.

ρ_a, θ_a = Actual parameters of the line.

ρ_p, θ_p = Line parameters obtained from the coordinates of the peak in the accumulator array.

ρ_c, θ_c = Line parameters calculated from the end points of the line by using the method proposed in this paper.

μ_f^k, μ_l^k = Row indices corresponding to the *first* and *last* non-zero cells respectively in C_k. The number of cells between the first and the last non-zero cells in C_k will be called the *spread of votes* in C_k.

$n_\rho^k = \mu_l^k - \mu_f^k$ = The spread of votes in C_k.

$d_{q,r} = q - r$ = Number of columns between C_q and C_r.

l_c = Length of the line computed from the end points obtained from the proposed algorithm.

ϵ_x, ϵ_y = Error in the x and y coordinates respectively of the end points obtained from the proposed algorithm.

ρ_1^k, ρ_2^k = Length of the normal to the bar (in the image plane) corresponding to the *first* and *last* non-zero cells respectively in C_k.

3 PROPOSED METHOD

The number of non-zero cells in $a_{\cdot,\theta_k}$ (called the spread in $a_{\cdot,\theta_k}$) is equal to the number of parallel bars (in the set corresponding to $a_{\cdot,\theta_k}$) intersected by the line. The proposed algorithm is *based on a microanalysis of the distribution of the votes around the peak in the accumulator array.*

In [6], the end points of a line were obtained by determining the intersecting points between the bar corresponding to the peak (in the accumulator array) and the two bars corresponding to the cells μ_f^k and μ_l^k in any column C_k. The accuracy of the detection therefore, depends on the choice of C_k, and the accuracy with which the peak can be detected. An algorithm for the choice of C_k has been described in [6]. However, an accurate detection of the peak in the accumulator array is a non-trivial task [9]. It has been shown [8] that

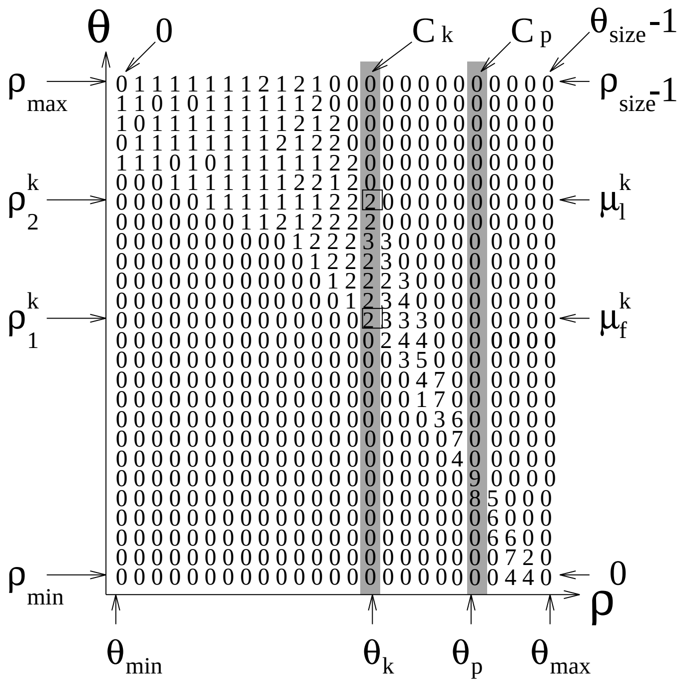

Figure 1: Accumulator array A showing the cells around the peak.

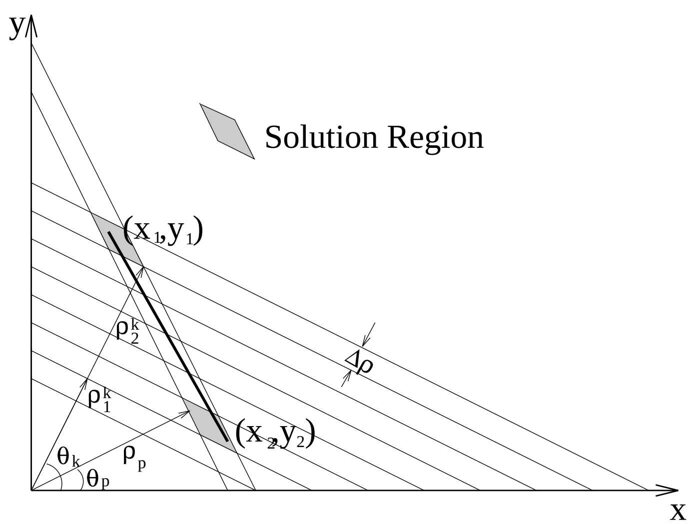

Figure 2: Illustration of parallel bars corresponding to the cells in C_k.

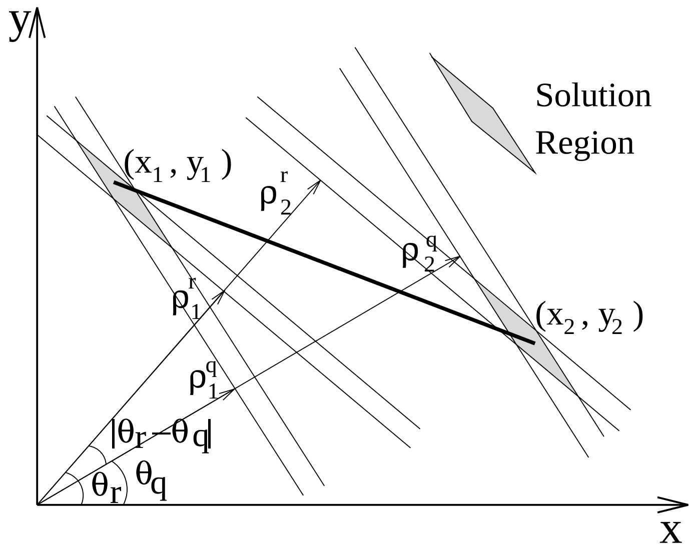

Figure 3: Computation of the end points independent of θ_p for $\theta_p > 45^0$.

the peak in the accumulator array may spread due to discretization of the image space and the accumulator array. Due to the dependence of the algorithm, described in [6], on the accuracy with which the peak can be detected, the end points obtained from the above algorithm are not always reliable. In the algorithm proposed in this paper, the θ value of the peak (θ_p) is only used to determine two columns, C_q and C_r, as described below.

Consider two columns C_q and C_r whose cells correspond to the two sets of parallel bars having their normals inclined at angles θ_q and θ_r respectively with the positive x-axis. The lengths of the normals ρ_1^q, ρ_2^q, ρ_1^r, and ρ_2^r (see Figure 3) to the bars corresponding to the cells μ_f^q, μ_l^q, μ_f^r, and μ_l^r respectively are determined from the accumulator array. The above lengths are the lengths of the normals to the bars corresponding to the first and last non-zero elements in C_q and C_r. These normals can be expressed as

$$\rho_1^q = x_1 \cos\theta_q + y_1 \sin\theta_q \tag{2}$$

$$\rho_1^r = x_1 \cos\theta_r + y_1 \sin\theta_r \tag{3}$$

$$\rho_2^q = x_2 \cos\theta_q + y_2 \sin\theta_q \tag{4}$$

$$\rho_2^r = x_2 \cos\theta_r + y_2 \sin\theta_r \tag{5}$$

ρ_1^q, ρ_2^q, ρ_1^r, and ρ_2^r can be expressed by the following equations (see Figures 1 and 3).

$$\rho_1^q = \rho_{\min} + \mu_f^q \Delta\rho \tag{6}$$

$$\rho_2^q = \rho_{\min} + \mu_l^q \Delta\rho \tag{7}$$

$$\rho_1^r = \rho_{\min} + \mu_f^r \Delta\rho \tag{8}$$

$$\rho_2^r = \rho_{\min} + \mu_l^r \Delta\rho \tag{9}$$

Since $\rho_{\min}$ and $\Delta\rho$ are known, ρ_1^q, ρ_2^q, ρ_1^r, and ρ_2^r in Equations (6)-(9) can be calculated after μ_f^q, μ_l^q, μ_f^r, and μ_l^r are obtained by scanning the columns C_q and C_r for the first and last non zero elements. The calculated values of ρ_1^q, ρ_2^q, ρ_1^r, and ρ_2^r are then substituted in Equations (2)-(5). Since θ_q and θ_r are known from the choice of C_q and C_r, solution of Equations (2) and (3) will give the coordinates (x_1, y_1) of one of the end points. The other end point (x_2, y_2) can be obtained similarly by solving Equations (4) and (5). The coordinates obtained by solving the above two sets of equations are as follows.

$$x_1 = \frac{\rho_1^q \sin\theta_r - \rho_1^r \sin\theta_q}{\sin(\theta_r - \theta_q)} \tag{10}$$

$$y_1 = \frac{\rho_1^r \cos\theta_q - \rho_1^q \cos\theta_r}{\sin(\theta_r - \theta_q)} \tag{11}$$

$$x_2 = \frac{\rho_2^q \sin\theta_r - \rho_2^r \sin\theta_q}{\sin(\theta_r - \theta_q)} \tag{12}$$

$$y_2 = \frac{\rho_2^r \cos\theta_q - \rho_2^q \cos\theta_r}{\sin(\theta_r - \theta_q)} \tag{13}$$

We define a thick line as one which has $\theta_a \neq 45^0$. Thick lines exhibit the phenomenon of a line being split up into a number of smaller horizontal or vertical line segments. The choice of C_q and C_r for the accurate detection of the end points of a thick line are given in detail in [6]. Once the end points of a line are determined from the above algorithm, the line length (l_c) is obtained from the end points by

$$l_c = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{14}$$

and the normal parameters are obtained as

$$\rho_c = \frac{x_2 y_1 - x_1 y_2}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \tag{15}$$

$$\theta_c = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) - 90^0 \tag{16}$$

The *maximum errors* in the coordinates of the end points depend on the area enclosed by the intersection of the two bars (C_q and C_r) as shown by the shaded parallelograms in Figure 3. The maximum errors in the x and y-coordinates can be easily shown to be

$$\epsilon_x = \frac{(\Delta\rho/2)\sin((\theta_q + \theta_r)/2)}{\sin((\theta_r - \theta_q)/2)} \tag{17}$$

$$\epsilon_y = \frac{(\Delta\rho/2)\cos((\theta_q + \theta_r)/2)}{\sin((\theta_r - \theta_q)/2)} \quad (18)$$

From Equations (17) and (18) it is seen that the errors can be reduced by

- increasing $\Delta\theta$, thereby making $|\theta_r - \theta_q|$ large (Increasing $\Delta\theta$ has the additional advantage of requiring less computational time to construct the accumulator array.) and
- choosing two columns C_q and C_r which are far apart, i.e., $|q - r|$ is large.

4 RESULTS

Using the algorithm proposed in Section 3, the errors in the line length are shown in Figure 4 for different values of $\Delta\rho$. A line with $\rho_a = 90$ and $\theta_a = 25^0$ was used with $\Delta\theta = 0.7087$. The errors are lower than those obtained from the previous algorithm [6]. The increased accuracy is due to the proposed algorithm being independent of the accuracy of the line parameters.

Note that the errors decrease with an increase in $d_{q,r}$. This is expected since the area of the solution region (shaded parallelogram in Figure 3) decreases with an increase in $d_{q,r}$ due to a reduction in the angle between the bars corresponding to the cells in C_q and C_r.

Figure 4: Errors in the line length vs. $d_{q,r}$ for a line having $\rho_a = 90$ and $\theta_a = 25^0$.

The variation of the *average errors* in the length and end points vs. $d_{q,r}$ are shown in Figure 5 for different combinations of $\Delta\rho$ and $\Delta\theta$. The average errors were obtained by applying the proposed algorithm to a large number of lines with different values of ρ and

θ and taking the mean of the errors from the different lines. Different combinations of $\Delta\rho$ and $\Delta\theta$ have been used in the above figures to show the effectiveness of the proposed algorithm for different values of $\Delta\rho$ and $\Delta\theta$.

Figure 5: Average errors in the line length and end points vs. $d_{q,r}$ for $\Delta\rho = 0.9$ and $\Delta\theta = 1.0$.

Figure 6 shows the error in the determination of the line length as a function of $\Delta\rho$ for $d_{q,r} = 10$ and three different values of $\Delta\theta$ (0.31, 0.47 and 1.0). It is seen that the errors increase with an increase in $\Delta\rho$ and/or a decrease in $\Delta\theta$. This is because the area of the solution region increases with an increase in $\Delta\rho$ and/or a decrease in $\Delta\theta$, and thus validates Equations (17) and (18). The above observation further confirms the validity of Equations (17) and (18).

5 CONCLUSIONS

The Hough transform, when applied to the detection of straight lines in images, provides only the parameters of the line. It fails to provide the length or the end points of the line. In this paper, we have proposed an improved algorithm to detect the end points and determine the line length. The proposed algorithm is *very robust* against the failure of the standard Hough transform to detect the parameters of the line due to spreading of the peak in the accumulator array. It is not computation intensive and has a time complexity of $O(1)$ which is the lowest among the algorithms reported in the literature.

Results show that the proposed algorithm can *detect the length and the end points very accurately*. The proposed algorithm is *non-iterative* in nature and is based on a micro-analysis of the spreading of the votes

Figure 6: Variation in average error in the line length vs. $\Delta\rho$ for different values of $\Delta\theta$ and $d_{q,r} = 10$.

in the accumulator array. This approach makes the algorithm very robust when compared to previous algorithms, most of which analyze the image space to obtain the line length and the end points.

References

[1] M. Atiquzzaman, "Multiresolution Hough transform – an efficient method of detecting pattern in images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 11, pp. 1090–1095, November 1992.

[2] J. Yamato, I. Ishii, and H. Makino, "Highly accurate segement detection using Hough transformation," *Systems and Computers in Japan*, vol. 21, no. 1, no. 1, pp. 68–77, 1990.

[3] L.F. Costa, B. Ben-Tzvi, and M. Sandler, "Performance improvements to the Hough transform," *IEE Conference Publications*, pp. 98–103, March 1990.

[4] W. Niblack and T. Truong, "Finding line segments by surface fitting to the Hough transform," *IAPR International Workshop on Machine Vision and Applications*, Tokyo, Japan, Nov 28-30, 1990.

[5] M.W. Akhtar and M. Atiquzzaman, "Determination of line length using Hough transform," *Electronics Letters*, vol. 28, no. 1, pp. 94–96, January 2, 1992.

[6] M. Atiquzzaman and M.W. Akhtar, "Complete line segment description using the Hough trans-

form," *Image and Vision Computing*, vol. 12, no. 5, pp. 267–273, June 1994.

[7] J. Richards and D.P. Casasent, "Extracting input-line position from Hough data," *Applied Optics*, vol. 30, no. 20, pp. 2899–2905, July 10, 1991.

[8] T.M. VanVeen and F.C.A. Groen, "Discretization errors in the Hough transform," *Pattern Recognition*, vol. 14, pp. 137–145, 1981.

[9] C.M. Brown, "Inherent bias and noise in the Hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, no. 5, pp. 493–505, September 1983.