Complete Line Segment Description using the Hough Transform

M. Atiquzzaman

Dept. of Computer Science
La Trobe University
Melbourne 3083, Australia.
Tel: (03) 479 1118
atiq@LATCS1.lat.oz.au

M.W. Akhtar

Research Institute
King Fahd Univ. of Pet. & Minerals
Dhahran 31261, Saudi Arabia.

Abstract

The Hough transform is a robust method for detecting discontinuous patterns in noisy images. When it is applied to the detection of a straight line, represented by the normal parameters, the transform provides only the length of the normal and the angle it makes with the axis. The transform gives no information about the length or the end points of the line. A few authors have suggested algorithms for the determination of the length and the end points of a line. The suggested methods are iterative in nature and are highly compute bound thereby making them unsuitable for real-time applications. In this paper, we propose an efficient non-iterative algorithm to determine the co-ordinates of the end points, the length, and the normal parameters of a straight line using the Hough transform. The proposed algorithm is based on an analysis of the spread of votes in the accumulator array cells, representing orientations which are different from that of the line under consideration. The algorithm uses a coarse resolution accumulator array which reduces the computation time.

Index terms: Hough transform, pattern recognition, line detection, complete line segment description.

1 Introduction

The Hough Transform (HT) is used in computer vision and pattern recognition for detecting geometric shapes (in digitized images) that can be defined by parametric equations. It is a mapping from the image plane to the parameter space, and essentially consists of a voting process followed by a peak detection stage. Every feature point in the image votes for all the possible patterns passing through it, and the votes are accumulated in an accumulator array. The advantage of the transform are its robustness to noise and discontinuities in the image, while the disadvantages are its demand for a large amount of computing time and storage. The time and space requirements depend on the number of parameters required to describe the pattern, the resolution of the accumulator array, and the image resolution. Since a straight line is the simplest of all geometric patterns and can be described with only two parameters, most authors assume a straight line for their study.

Different aspects of the HT have been investigated and reported in the literature. The compute-bound nature of the HT has inspired the development of *efficient algorithms* [1, 2, 3] and *implementations* [4, 5, 6, 7, 8, 9, 10] of the transform in multiprocessor systems. *Performance* of the transform with respect to the accuracy of detection of a pattern has been discussed in [11, 12, 13]. A formal mathematical definition of the transform, its properties and relationships appear in [14]. The effect of aliasing distortion has been studied from a signal processing point of view in [15].

Commonly used parameterizations of a straight line are the slope-intercept and the normal forms. When the HT is used for the detection of a straight line in an image, only the parameters of the line segment are obtained. *It does not provide any information regarding the length, position, or endpoints of the line segment in the image plane.* However, in machine vision application, the length and exact position of the line in addition to its normal parameters are required for locating the straight line. The parameters, the length, and the coordinates of the end points of a straight line constitute the *complete line segment description.* Techniques to find the complete line segment description using the Hough transform have not been thoroughly studied, although a few algorithms are available in the literature [16, 17, 18]. The available methods are based on the projection of the line on either the x or y axis in the image plane, and follow iterative approaches thereby rendering them highly compute-bound.

A new algorithm for obtaining the complete line segment description from the information contained in the Hough accumulator array is proposed in this paper. The proposed algorithm is based on an analysis of the spread of votes (resulting from digitization of the image and the accumulator array) in the accumulator array. The approach does not require iteration and is highly efficient in terms of computing time. Moreover, a small accumulator array is used to reduce the computation time. Since the approach is based on an analysis of the spread of votes in the accumulator array, the rest of this section is devoted to familiarize the reader to the phenomenon of spreading of votes, the reasons for the spreading, and implications of the spreading on the successful detection of a straight line.

A straight line in its normal representation is given by

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

where, ρ and θ are the length of the normal and the angle of the normal with respect to the positive x-axis. The ρ and θ axes of the accumulator array are divided into a number of equal divisions. For each feature point in the image, a set of ρ values are calculated for the set of θ values corresponding to the divisions in the accumulator array. The ρ values are quantized and the corresponding (ρ, θ) cells of the accumulator array are incremented. A set of collinear feature points corresponding to a straight line in the image space is transformed to a set of sinusoidal curves in the parameter space. The curves intersect at a point in the parameter space resulting in a peak in the accumulator array. The line segment is detected by searching for the peak in the $\rho - \theta$ parameter space. The coordinates of the peak give the normal parameters of the line.

A sharp and distinct peak is necessary for an accurate determination of line parameters. The factors which affect the sharpness and the shape of the peak are the discretization of the image, width of the line, and the quantization resolution (denoted by $\Delta\rho$ and $\Delta\theta$ along the ρ and θ axis respectively) of the parameter space. It has been shown previously that the peak in the accumulator array spreads in the ρ-direction as $\Delta\rho$ is decreased [11]. Spreading of the peak adversely affects the successful detection of the peak, and consequently the detection of the parameters of the line. A decrease in $\Delta\theta$ results in a spread of the peak in the θ-direction. To overcome the problem of the spread of peak due to discretization, smoothing of the Hough accumulator array prior to peak searching was suggested by Niblack [18]. The pixels belonging to a line (expressed in the $\rho - \theta$ parametrization) produces a butterfly shape in the parameter space. The angle subtended by the wings of the butterfly and the orientation of the butterfly depends upon the length and position of the line [19]. In the next paragraphs the amount of spread of the peak in quantified.

During the construction of the accumulator array by the voting process, the value of θ is sampled while that of ρ is quantized. 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 [11]. The cells $a_{\cdot,\theta_k}$ in the accumulator array, therefore, correspond to a set of parallel bars of width $\Delta\rho$ and 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 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. It is this spread of votes in the accumulator array which forms the basis of our proposed algorithm for the complete line segment description to be presented in Section 3.

In general, the actual angle of the line (θ_a) will not be equal to any of the sampled values of θ (denoted by θ_k) as shown in Figure 2. The line to be detected will therefore cross several bars of width $\Delta\rho$ in the image plane, resulting in a spread of the peak in the ρ-direction. The maximum spread of the peak in the ρ direction was shown to be [11]

$$n_\rho = \left\lfloor \frac{l \sin(\Delta\theta/2)}{\Delta\rho} \right\rfloor + 2 \tag{2}$$

where, l is the length of the line. The equation was derived in [11] for the purpose of suggesting an optimum value of $\Delta\rho$ which would produce a sharp and distinct peak leading to successful detection of the line parameters. However, we would like to point out that *prior knowledge of l may not be available in real-world applications, in which case an optimum value of $\Delta\rho$ can not be suggested from Equation (2).*

The problems of spread of votes due to quantization of the parameter space have been mentioned above. Another problem arising as a consequence of image quantization is feature points corresponding to a straight line not being collinear in the digitized image (see Figure 1). A digitized line may become split into several smaller horizontal or vertical line segments with several consecutive feature points having the same x or y coordinate. For $\theta_a < 45^0$, several consecutive feature points will have the same x coordinates, while for $\theta_a > 45^0$ the y-coordinates of several feature points will be the same.

Having familiarized the reader with the concept of spread of votes on which our algorithm for complete line segment description is based, we now define the objectives and the layout of this paper. The objectives of this paper are as follows.

- To *develop a new efficient algorithm* which will provide the coordinates of the end points, the normal parameters, and the length of a straight line from the information contained in a Hough accumulator array.

- To reduce the time complexity of the proposed algorithm as compared to those already available in the literature. The algorithm should be non-iterative and should not require reaccumulations of the accumulator array.

- To investigate the efficiency of the proposed algorithm in determining the complete line segment description accurately.

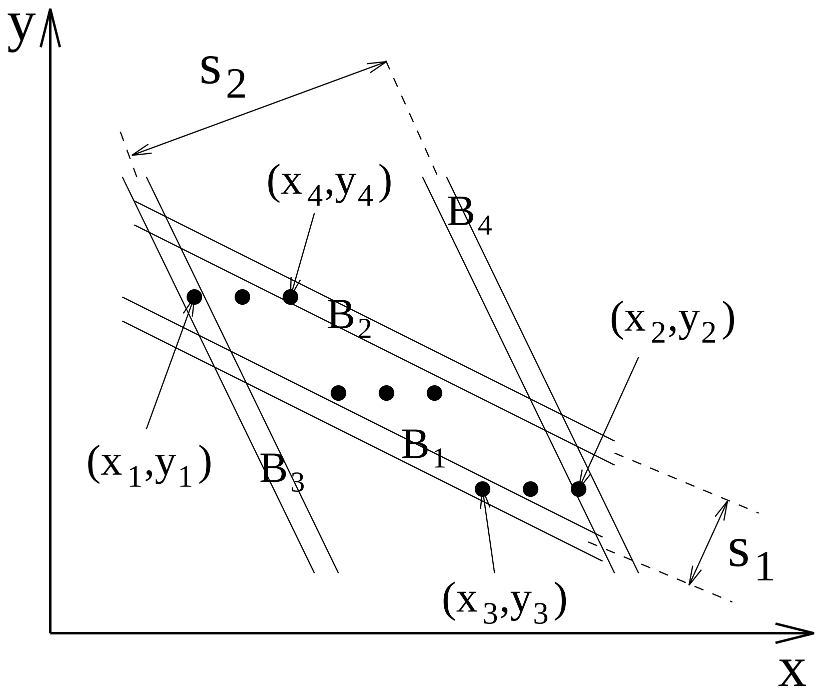

Figure 1: Erroneous detection of end points due to choice of $C_k, k > p$ for $\theta_p > 45^0$.

The rest of the paper is organized as follows. Previous work related to complete line segment description are outlined in Section 2. The proposed algorithm and the results are presented in detail in Section 3. A complexity analysis of the proposed algorithm is given in Section 4 followed by conclusions in Section 5.

2 Previous Work

Four different algorithms that have been suggested in the literature for finding the complete line segment description are briefly described in this section. This will enable a comparison of the time complexities between the proposed algorithm and those available in the literature. Yamato [16] described a method using feedback. According to his method, the values of ρ and θ are obtained from the peak in the accumulator array and a line called the *first likely line* is drawn on the image plane. A set of feature points which are close to the first likely line are determined, and the parameters of a *second likely line* are calculated by performing a least square error fit using the above set of feature points. The procedure is repeated for a maximum of five iterations. The second likely line of an iteration is taken to be the first likely line in the next iteration. Finally, the endpoints of the line are determined by taking the projection of the feature points (lying on the line) on either the x-axis or y-axis. The algorithm is iterative in nature and uses least square curve fitting, thereby making it computation intensive. The complexity of the algorithm, ignoring the votes accumulation time, is $O(n_f n_i + N)$ where 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 [20]. In this method, the Hough transform is used only to get an initial estimate of the line. Subsequent operations are repetitive application of the least square curve fitting algorithm.

Yamato's algorithm has several drawbacks. In his approach, the number of feature points constituting the original line is estimated by the sum of the votes in three adjacent cells along a column of the accumulator array, with the peak cell in the center of the cells. *We believe that this assumption is erroneous*, since the spread of votes can not be taken to be exactly equal to three. The extent of the spread of votes in the ρ direction is given by Equation (2) and depends on l, $\Delta\rho$, and $\Delta\theta$. Secondly, a line segment may be detected as multiple smaller line segments.

In a different approach described by Costa [17], the end points of a line are first computed followed by the determination of the length of the line. The combinatorial Hough transform [21] is used to determine ρ and θ of the line. All possible feature points lying within a distance of $\Delta\rho$ from the detected line are projected onto a one-dimensional histogram along either the x or y axis depending on θ. The end points and the length of the line are obtained from a connectivity analysis of the one-dimensional histogram. The method of projection is similar to that used in [16]. The algorithm suffers from two drawbacks. Firstly, *multiple line segments* may be detected for a single line segment in the image. Secondly, the reliability in the detection of ρ and θ of the line (under spread of votes in the accumulator array) using combinatorial Hough transform may vary with the discretization of the accumulator array. The complexity of the algorithm, ignoring the votes accumulation time, is $O(n_f + N)$.

Another approach for obtaining the length of a line segment from the end points has been proposed by Niblack [18]. The concept of surface fitting, assuming a Gaussian distribution of feature points in the direction perpendicular to the line segment, has been used. An accumulator array (A), is constructed using the feature points in the image and then a second accumulator array, ($\hat{A}$), is constructed for a line by assuming the values of parameters like the end points of the line, length of the line, mean and standard deviation of the Gaussian distribution used to characterize the distribution of feature points of the assumed line, etc. The difference ($A - \hat{A}$) is determined. The assumed parameters are then varied and $\hat{A}$ recomputed as many times as is required to minimize the difference ($A - \hat{A}$) to 0.01. The method suffers from several drawbacks. Because of the iterative nature of the algorithm and with many parameters to be varied, the procedure becomes *highly compute-bound* with too many reaccumulations of the accumulator array. Secondly, the *convergence time* of the algorithm depends on the accuracy of the initial guess of the parameters. With no a priori knowledge of the length or orientation of the line in the image, the algorithm can be very computation intensive. The complexity of the algorithm, ignoring the first accumulation time, can be shown to be $O(n_f n_i N + N^2)$.

A non-iterative method for detecting the length of a line from an analysis of the voting patterns in the accumulator array has been described by Akhtar [20]. The length is obtained by taking any column in the accumulator array and finding the extent of the spread of votes in the column. The limitation of the algorithm is its inability to determine the end points of the line segment. The complexity of the algorithm is $O(1)$.

In the next section, we propose a novel algorithm for finding the end points and the length of a line. The algorithm is *non-iterative* and requires *only one accumulation of the accumulator array.*

3 The Proposed Approach

The following notations will be used in describing the proposed algorithm for finding the complete line segment description. The algorithm is based on a micro analysis of the spread of votes in the accumulator array (see Figures 2 and 3).

$\rho_{\text{size}}, \theta_{\text{size}}$ = 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.

$b_{i,k}$ = 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 to be detected.

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 = row index corresponding to the first non-zero cell in C_k. The *first non-zero* cell in C_k (see Figure 3) is defined to be the first cell containing a non-zero vote when scanning the cells in C_k from $\rho = 0$ to $\rho_{\text{size}} - 1$.

μ_l^k = row index corresponding to the last non-zero cell in C_k. The *last non-zero* cell in C_k is the first cell containing a non-zero vote when scanning the cells in C_k from $\rho = \rho_{\text{size}} - 1$ down to 0. 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.

$d_{k,p} = k - p =$ number of columns between C_k and C_p.

$l_c =$ length of the line computed from the co-ordinates of the end points obtained from the proposed algorithm.

$\epsilon_x =$ error in the x-coordinates of the end points obtained from the proposed algorithm.

$\epsilon_y =$ error in the y-coordinates of the end points obtained from the proposed algorithm.

$\rho_1^k =$ length of the normal to the bar (in the image plane) corresponding to the first non-zero cell in C_k. Note that the first non-zero cell in C_k corresponds to the bar $b_{i,k}$ in the image plane containing the end point (x_2, y_2) in Figure 2.

$\rho_2^k =$ length of the normal to the bar (in the image plane) corresponding to the last non-zero cell in C_k. Note that the last non-zero cell in C_k corresponds to the bar $b_{i,k}$ in the image plane containing the end point (x_1, y_1) in Figure 2.

$\rho_{\min}, \rho_{\max}, \theta_{\min}, \theta_{\max} =$ 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.

We define a thin line as a line having $\theta_a = 45^0$, and a line having $\theta_a \neq 45^0$ as a thick line. Note that a thick line exhibits the phenomenon of the line being split up into a number of smaller horizontal or vertical line segments as explained in Section 1.

When using the HT for the detection of a straight line, the primary objective is to determine ρ_p and θ_p from the coordinates of the peak in the accumulator array. No information regarding the co-ordinates of the end points of the line are obtained. *In the proposed algorithm, the approach is to determine the coordinates of the end points accurately, followed by determination of ρ_p and θ_p from the end points. It has been found that this reverse approach not only provides the complete line segment description, but also results in more accurate values for the $\rho - \theta$ parameters of the line.*

For a *thin line segment*, ρ_1^k and ρ_2^k for any column k can be expressed as follows (see Figures 2 and 3).

$$\rho_1^k = \rho_{\min} + \mu_f^k \Delta\rho \tag{3}$$

$$\rho_2^k = \rho_{\min} + \mu_l^k \Delta\rho \tag{4}$$

Since $\rho_{\min}$ and $\Delta\rho$ are known, ρ_1 and ρ_2 in Equations 3 and 4 can be calculated after μ_f^k and μ_l^k are obtained by scanning C_k.

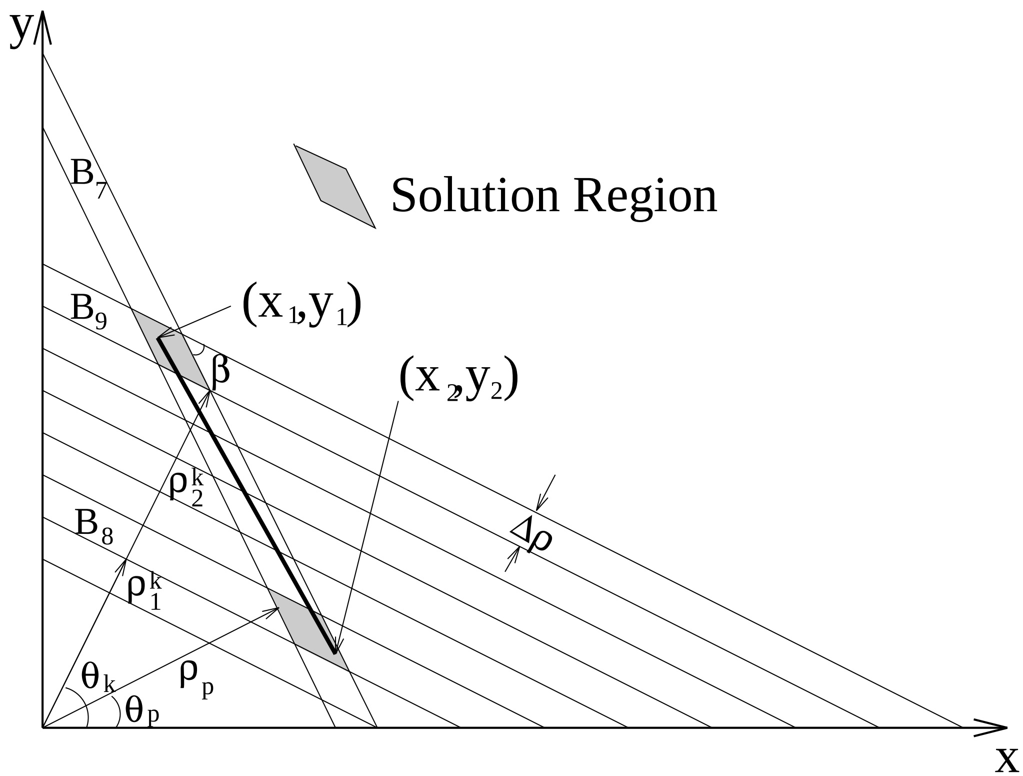

Figure 2: Illustration of intersection of bars showing solution space by shaded parallelograms.

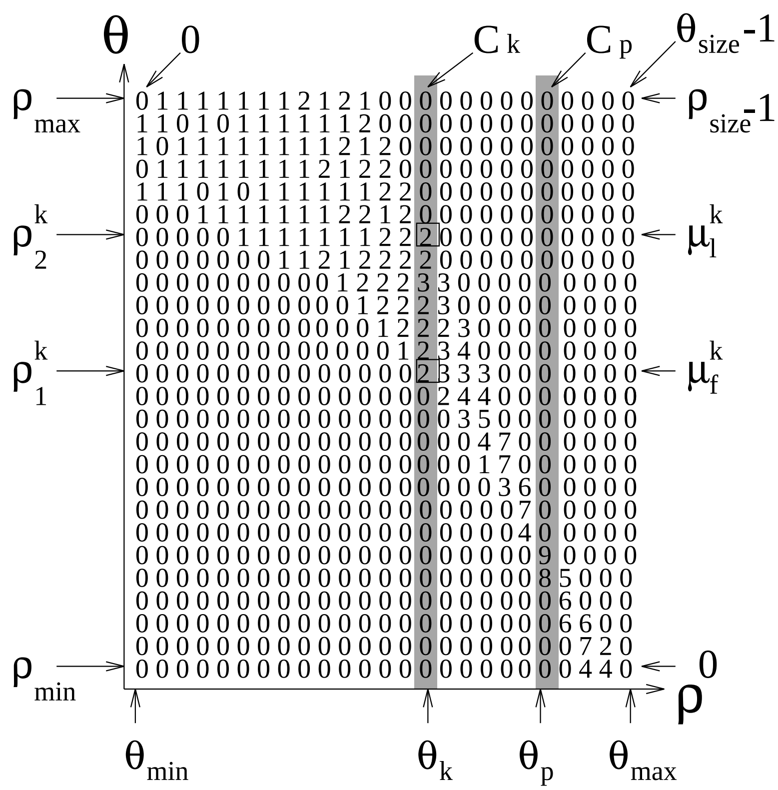

Figure 3: Accumulator array A showing cells around the peak.

By detecting the peak in the accumulator array, the line can be expressed as (see Figure 2)

$$\rho_p = x \cos\theta_p + y \sin\theta_p \tag{5}$$

The lengths ρ_1^k and ρ_2^k corresponding to the cells (μ_f^k, k) and (μ_l^k, k) in any column C_k of the accumulator array can also be expressed as

$$\rho_1^k = x \cos(\theta_p + d_{k,p}\Delta\theta) + y \sin(\theta_p + d_{k,p}\Delta\theta) \tag{6}$$

$$\rho_2^k = x \cos(\theta_p + d_{k,p}\Delta\theta) + y \sin(\theta_p + d_{k,p}\Delta\theta) \tag{7}$$

ρ_1^k and ρ_2^k in the left hand side of Equations (6) and (7) are obtained from Equations (3) and (4) respectively. The terms on the right hand side of Equations (6) and (7) can be obtained as follows. θ_p can be obtained from the coordinates of the peak in the accumulator array, $d_{k,p}$ and C_p from the choice of C_k, and $\Delta\theta$ from the discretization of the accumulator array. It is apparent from Figure 2 that the solution of Equations (5) and (6) will give the coordinates (x_1, y_1) of one of the end points of the line while the solution of Equations (5) and (7) will give the coordinates (x_2, y_2) of the other end point. The end points can, therefore, be shown to be given by

$$x_1 = \frac{\rho_p \sin(\theta_p + d_{k,p}\Delta\theta) - \rho_2^k \sin\theta_p}{\sin(d_{k,p}\Delta\theta)} \tag{8}$$

$$y_1 = \frac{\rho_p - x_1 \cos\theta_p}{\sin\theta_p} \tag{9}$$

$$x_2 = \frac{\rho_p \sin(\theta_p + d_{k,p}\Delta\theta) - \rho_1^p \sin\theta_p}{\sin(d_{k,p}\Delta\theta)} \tag{10}$$

$$y_2 = \frac{\rho_p - x_2 \cos\theta_p}{\sin\theta_p} \tag{11}$$

In Equations (8) - (11), ρ_1^k and ρ_2^k are obtained from Equations (3) and (4) respectively, ρ_p and θ_p from the coordinates of the peak in the accumulator array, and $\Delta\rho$ and $\Delta\theta$ from the resolution of the accumulator array. x_1 and x_2 are first determined from Equations (8) and (10), which are then used to determine y_1 and y_2 from Equations (9) and (11). Once the coordinates of the end points are determined, the length of the line can be computed from

$$l_c = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{12}$$

and the normal parameters are obtained as

$$\rho_c = \frac{x_2 y_1 - x_1 y_2}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}} \tag{13}$$

$$\theta_c = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) - 90^0 \tag{14}$$

In the case of a *thin line*, it is immaterial whether C_k is chosen to the left or right of C_p. Any C_k, such that $k \neq p$, can be chosen. However, because of the breaking of a line into several smaller line segments in the case of a thick line, the choice of C_k in such a case is crucial to the accurate detection of the end points. *For a thick line segment*, the following rules regarding the choice of C_k have to be applied for an accurate detection of the endpoints.

- Rule 1: Select $C_k, k > p,$ if $-45^0 < \theta_a < 45^0, 135^0 < \theta_a < 225^0$.

- Rule 2: Select $C_k, k < p,$ if $45^0 < \theta_a < 135^0, 225^0 < \theta_a < 315^0$.

For applying the above rules, θ_a can be approximated by θ_p which is obtained from the coordinates of the peak in the accumulator array. The reasons behind the above rules can be explained as follows.

Figure 1 shows a line ($\theta_a > 45^0$) which became broken into smaller horizontal line segments during digitization. Points (x_1, y_1) and (x_2, y_2) are the end points of the line, although (x_3, y_3) and (x_4, y_4) will be detected as end points if $C_k, k > p$, is chosen to determine μ_f^k and μ_l^k in the scanning process. This is because if $C_k, k > p$, is chosen, feature points in the bars B_1 and B_2 will be accumulated in μ_f^k and μ_l^k respectively, resulting in a small (and inaccurate) spread $s_1/\Delta\rho$ of the votes in column C_k of the accumulator array. On the other hand, if $C_k, k < p$, is used, the feature points in B_3 and B_4 will be accumulated in μ_f^k and μ_l^k respectively. In this case, the actual end points, (x_1, y_1) and (x_2, y_2), lie in B_3 and B_4. The end points will therefore be detected properly. Hence, the above rules for selecting C_k should be used to avoid any erroneous detection of end points.

Note that the solution of Equations (5) and (6) gives the intersection point of the bars B_7 and B_8 in Figure 2. Similarly, solution of Equations (5) and (7) gives the intersection point between the bars B_7 and B_9. In the above discussion, these intersection points have been taken as the detected end points of the line. However, the actual end points can lie anywhere within the *solution regions* shown by shaded parallelograms. Hence, the accuracy of the algorithm in detecting the end points depends on the size of the solution region – the smaller the region, the more accurate is the detection. Note that the size of a solution region depends on the angle (β) between the bars and the value of $\Delta\rho$.

For evaluating the performance of the proposed algorithm, it has been applied to a 128×128 image containing a line having $\theta_a = 15^0$. The performance of the algorithm has been measured by the accuracy with which the end points can be detected. The performance was measured by the following.

- The errors between the coordinates of the detected end points and the actual end points of the line.

- The error in lengths between the detected line and the actual line.

For example, the error in the x-coordinate of an end point has been obtained by taking the absolute value of the difference in the x-coordinates of the detected and the actual end points.

The errors in the x-coordinates, the y-coordinates, and the length of the line as a function of $d_{k,p}$ (which depends on the C_k chosen for analysis) are shown in Figures 4, 5 and 6 respectively. The graphs are for $\Delta\rho = 0.3$ and two values of $\Delta\theta$ (0.5625 and 0.6767). The two curves corresponding to a value of $\Delta\theta$ in a graph are for the two end points of the line.

The first observation from the figures is that the *errors reduce as $d_{k,p}$ increases* i.e., the errors decrease as C_k is chosen further away from C_p. This can be attributed to the fact that the solution regions (represented by shaded parallelograms in Figure 2) become smaller as the angle β $(= d_{k,p}\Delta\theta)$ increases when C_k moves further away from C_p.

The second observation is that the *errors are larger for smaller values of $\Delta\theta$*. This is because a smaller $\Delta\theta$ (for a fixed $d_{k,p}$) results in a smaller angle β between the two bars. A smaller β means a larger solution space and therefore a reduced accuracy in the computed end points. Use of a large value of $\Delta\theta$ results in more accurately computed endpoints and a reduction in the computation required for constructing the accumulator array.

The third observation from the above figures is that the *errors are higher for negative values of $d_{k,p}$*. Note that negative values of $d_{k,p}$ represent $k < p$. The above observation follows from, and also verifies, the rules presented earlier (regarding selection of C_k) i.e., for the particular line under consideration having $\theta_a = 15^0$, $k > p$ will produce better results.

Note that ρ_p and θ_p, obtained from the peak in the accumulator array, is used in the computation of the end points. Peak detection often becomes difficult because of the spread of the peak in the accumulator array. A shortcoming of the above approach is that, if the peak in the accumulator array is not detected properly, the intersection of the bars and predicted line segment could even lie outside the image plane. This may lead to points being detected as end points which are not even the feature points. The above drawback can be overcome if the end points can be determined independently of ρ_p and θ_p. A method to accomplish this is currently being investigated by the authors.

4 Complexity Analysis

The proposed algorithm is *non-iterative* and does not require reaccumulations of the Hough accumulator array. After the first accumulation of the accumulator array, it takes a constant time to solve the two sets of simultaneous equations to find the coordinates of the end points and subsequently the length and normal parameters of the line. Hence, ignoring the time for the first construction of the accumulator array, the complexity of the algorithm is $O(1)$ which is much lower than of those proposed in the literature (see Section 2). Note that the complexities of all the algorithms mentioned in Section 2 are also based on ignoring the time taken for the first accumulation of the accumulator array. This is because one accumulation of the accumulator array has always to be performed for finding the normal parameters of the line when using the Hough transform, even if the complete line segment description is not desired.

5 Conclusions

The Hough transform provides the parameters of a line in an image. It does not give the length or the end points of a line segment. The end points are required for locating a straight line in the image. A review of previous work on complete line segment description (end points and length in addition to normal line parameters) revealed that the algorithms available in the literature are highly compute bound because of their iterative nature. Moreover, the previous algorithms analyze image space to determine the descriptions. In this paper, an algorithm for determining the complete line segment description has been presented. The proposed algorithm is computationally much more efficient than previous algorithms. The proposed algorithm has a complexity of $O(1)$ in contrast to complexities of $O(n_f n_i + N)$, $O(n_f + N)$, and $O(n_f n_i N + N^2)$ of previous algorithms found in the literature. The low complexity of the proposed algorithm has been achieved through an analysis of the spread of votes in the accumulator array. The proposed algorithm is not iterative and requires only one accumulation of the accumulator array. Results obtained from the proposed algorithm with respect to the accuracy of detection of the end points and the line length have been presented.

The proposed algorithm gives the coordinates of the end points of a line. The length and normal parameters of the line are then obtained from the coordinates of the end points. The normal parameters obtained from the proposed algorithm have higher accuracy than those obtained from the peak in the accumulator array by the conventional Hough transform. The lower accuracy of the conventional Hough transform is because of its inability to detect the peak correctly in the accumulator array when there is a spread of the votes in the peak. On the contrary, the proposed algorithm exploits the spread of the votes in the accumulator array to determine the line parameters, thereby resulting in higher accuracy.

References

[1] J. Illingworth and J. Kittler, “Adaptive Hough transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 690–698, September 1987.

[2] H. Li, M.A. Lavin, and R.J. LeMaster, “Fast Hough transform,” *Computer Vision, Graphics and Image Processing*, vol. 36, pp. 139–61, 1986.

[3] 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.

[4] C. Guerra and S. Hambrusch, “Parallel algorithms for line detection on a mesh,” *Journal of Parallel and Distributed Computing*, vol. 6, pp. 1–19, 1989.

[5] A.L. Fisher and P.T. Highnam, “Computing the Hough transform on a scan line array processor,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 3, pp. 262–265, March 1989.

[6] C.A. Sher and A. Rosenfeld, “A pyramid Hough transform on the connection machine,” Tech. Rep. CAR-421, Centre for Automation Research, University of Maryland, February 1989.

[7] D. Ben-Tzvi, A. Naqvi, and M. Sandler, “Synchronous multiprocessor implementation of the Hough transform,” *Computer Vision, Graphics and Image Processing*, vol. 52, pp. 437–446, 1990.

[8] K. Hanahara, T. Maruyama, and T. Uchiyama, “A real-time processor for the Hough transform,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 1, pp. 121–125, January 1988.

[9] G. Bongiovanni, C. Guerra, and S. Levialdi, “Computing the Hough transform on a pyramid architecture,” *Machine Vision and Applications*, vol. 3, pp. 117–123, 1990.

[10] M. Atiquzzaman, “Implementation of the multiresolution Hough transform on a pyramid machine,” Tech. Rep. 8/92, La Trobe University, Department of Computer Science, Melbourne, Australia, June 1992.

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

[12] T. Risse, "Hough transform for line recognition: Complexity of evidence accumulation and cluster detection," *Computer Vision, Graphics and Image Processing*, vol. 46, pp. 327–345, 1989.

[13] D.J. Hunt, L.W. Nolte, and A.R. Reibman, "Hough transform and signal detection theory performance for images with additive noise," *Computer Vision, Graphics and Image Processing*, vol. 52, pp. 386–401, 1990.

[14] J. Princen, J. Illingworth, and J. Kittler, "A formal definition of the Hough transform: properties and relationships," *Journal of Mathematical Imaging and Vision*, vol. 1, pp. 153–168, 1992.

[15] N. Kiryati and A.M. Bruckstein, "Antialiasing the Hough transform," *Computer Vision, Graphics and Image Processing: Graphical Models and Image Processing*, vol. 53, no. 3, pp. 213–222, May 1991.

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

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

[18] 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.

[19] V.F. Leavers and J.F. Boyce, "The Radon transform and its application to shape parametrization in machine vision," *Image and Vision Computing*, vol. 5, no. 2, pp. 161–166, May 1987.

[20] 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.

[21] L.D.F. Costa and M.B. Sandler, "Improving parameter space for Hough transform," *Electronics Letters*, vol. 25, no. 2, pp. 134–136, January 19, 1989.

Figure 4: Error in the x coordinates as a function of $d_{k,p}$ for different values of $\Delta\theta$.

Figure 5: Error in the y coordinates as a function of $d_{k,p}$ for different values of $\Delta\theta$.

Figure 6: Error in the computed length as a function of $d_{k,p}$ for different values of $\Delta\theta$.