Comparative Study of Selected HT based Geometric Feature Extraction Techniques used for Determining Length of a Line Segment in an Image

[1]Kishor V. Ugale, [2]M.A. Pund, [3]Ram Deshmukh, [4]Satish Bahale

[1]P.G. Student, [2]Professor Dept. of CSE, [3]P.G. Student, [4]P.G. Student
PRMIT & R Badnera (M.S.), India

Abstract : Line segments are important when analyzing geometric shapes in images for machine vision applications. In particular this problem also involves a need to extract parameters of line segments in images, such as width and length. Typical feature extraction methods for line detection apply least-square fitting. These methods are in general sensitive to outliers; they require that feature points are clustered. The Hough transform (HT) defines an alternative class of methods. The basic HT does not provide length or width of a detected line segment; it only provides the two normal parameters d and Θ of a line. In principle, the HT is capable to detect the length of a line segment. Objective of this paper is to review the various methods based on Hough Transform which can be used to determine the parameters of the lines in an image, especially the length and compare the performance of some selected effective methods from the view point of time and space optimization. Various relevant works have been reviewed and in common what were the major problems faced by HT based methods in determining accurate geometric features and how the research has progressed to eliminate them is described. In the remaining part, three selected methods are compared.

***IndexTerms* - Hough transform, length of segment, width, curve fitting, HT Butterfly**

1. Introduction

Line segments are important when analyzing geometric shapes in images for machine vision applications. In particular this problem also involves a need to extract parameters of line segments in images, such as width and length. Traditional methods for line detection use least-square fitting technique. These methods are in general sensitive to outliers; they require that feature points are clustered.

The Hough transform (HT) [10,11,12] defines an alternative class of methods. Basically HT does not provide directly length or width of a detected line; it only provides the two normal parameters ρ and Θ of a line. In principle, however the Hough Transform can help to detect the length of a line segment. In HT, each feature point (x,y) of a segment is mapped to a sine curve via the following equation:

$$\rho = x \cos \Theta + y \sin \Theta \qquad (1)$$

By discretizing the HT space by resolutions of $\Delta\rho$ and $\Delta\Theta$, the feature point (x,y) votes for every cell located on the curve. After the voting process, the position of the cell receiving the most votes, that is, the peak, is considered as the (ρ,Θ) values of the segment. Obviously, the HT converts the pattern recognition problem to a peak generating and seeking problem. This makes the HT robust to the noise and complex background. How to generate a strong and distinct peak and how to seek the ‘‘correct’’ peaks is one of the most focused research on HT where the computational complexity, storage requirement, accuracy, and resolution are widely considered, especially in the situations of high accuracy and high resolution HT is required. Various methods were proposed to improve the voting process [28,29], the discretization of the HT space [30], feasible resolutions settings [31], post-voting process etc. These research works put most emphases on the distinct peaks generating and seeking, and hence can be called ‘‘peak based’’ methods. A common problem of these methods is the infeasible computational complexity and memory storage requirement [32,33].

Another important branch of HT researches is the microanalysis on the ‘‘HT butterfly’’. In fact, during the voting process, not only the peak is generated, but also, a large area composed of all mapped sine curves is built. By considering the part of this area around the peak, a butterfly shape is obtained. For example, the segment displayed in Fig. 1.2 (a) is mapped to the HT space and considering only the area around the peak, the butterfly-shaped voting area shown in Fig. 1.2 (b) can be obtained where the height of the peak is denoted as h, the thickness of the wings at δΘ far from the peak is denoted as $\delta_{\delta\Theta}$, and the width of the wings is $w_{\delta\Theta}$.

After having the direction of a set of approximately collinear pixels detected, we can project the estimated collinear image features on the x- or y-axis in image space; the length of the line-segment is then determined as the Euclidean distance between the estimated two endpoints.

There are also HT methods which use the buttery distribution in the Hough space, as identified in [13]. These buttery-techniques have origins in methods proposed earlier. Akhtar [14] calculates the length of a detected line segment based on the spreading of voting cells in a column around the peak. Ioannou [15] estimates the line-segment length by analyzing the total vote values of cells in the peak column. The endpoints are detected by resolving simultaneously equations obtained by the first and the last non-zero-value voting cells in any two columns around the peak; the length is then again calculated as the Euclidean distance between the estimated two endpoints.

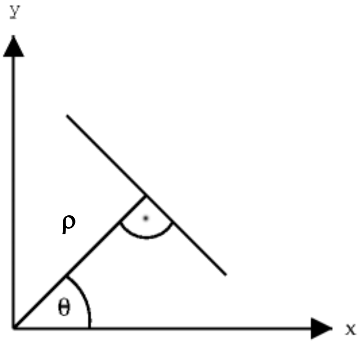
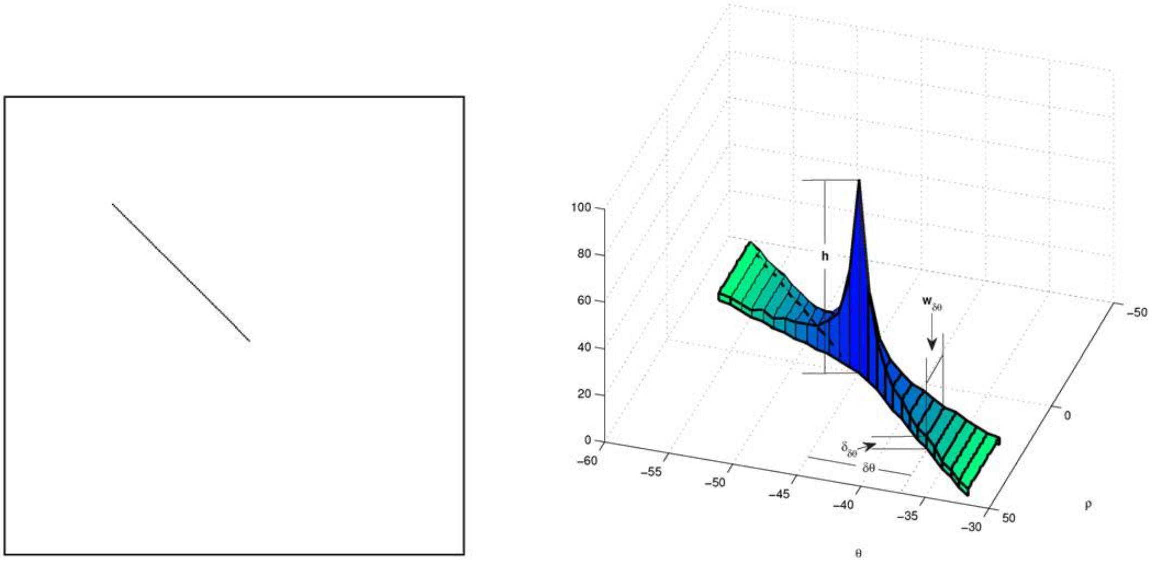


Figure 1.1 **:** Normal parameters of a line

Objective of this paper is to review the various methods based on Hough Transform which can be used to determine the parameters of the lines in an image, especially the length and compare the performance of some selected effective methods from the view point of time and space optimization. Various relevant works have been reviewed and in common what were the major problems faced by HT based methods in determining accurate geometric features and how the research has progressed to eliminate them is described. In the remaining part, three selected methods are described and compared.



(a) A straight line segment

(b) The HT butterfly of the segment

Figure 1.2 : A straight line segment and its HT butterfly.

2. Review of Related work

The main objective of HT is to divide the parameter space (ρ,Θ) into cells, where the size of each cell is (Δρ,ΔΘ). ρ is the algebraic distance from the origin to a straight line and Θ is the angle between the normal and *x* axis. In this way, a feature point with coordinates (*x*, *y*) in the image space is mapped to a sinusoidal curve in the HT space. The curves corresponding to collinear points will intersect at a common cell, resulting in a peak on the accumulator matrix. Usually, the highest peak represents the most prominent straight line. However, the Standard HT(SHT) suffers from a variety of problems stemming from its discretisation and voting process.[4, 5, 7, 8, 16, 17, 41, 42, 43, 44, 45, 46] Different methods were proposed to alleviate these problems. The HT voting framework was modified to increase the number of accumulators for higher precision and resolution. In fact, the discrete voting process is the main source of problems such as peak splitting (a peak is splitting to several peaks lying close to one another). The voting process also might spread the peak to several cells around the "true" position. These problems dim the peak and hence, affect detection accuracy and reliability.

Du et al. [16,17] consider the complete parameter description of a line segment, denoted by direction, length, width, and position. In their method length is obtained by determining the vertical width of a butterfly wing. The width of a line segment was computed by comparison of the actual voting value and theoretical voting values in a specific column. Reliable length and width can be obtained using a Mean Square Error (MSE) estimation by considering multiple columns. This method is affected by image noise. The *detection* accuracy relies on a very fine quantization of the Hough space.

Many researchers have conducted experiments and proposed various methods to eliminate the problems in using Hough Transform for geometric feature extraction and parameter determination and contributed in improving the accuracy by various aspects. These works can be classified under the following two categories.

2.1 Improvement in the Resolution and Accuracy

Due to the rounding operations in the voting process, the discretisation of image space and parameter spaces have several problems regarding the accuracy and resolution of HT. These include peak splitting, flattening and resolution limitations. Many researchers have analysed the drawbacks of HT and suggested several methods to improve the accuracy and resolution. The quantization errors in both the image space and parameter space obviously affect the accuracy of straight line detection. Shapiro et al.[5] introduced graphical methods estimating HT performance for straight line detection in the presence of noise. They also demonstrated the effects of the quantization errors on the accuracy of estimating the underlying set of collinear points. Veen et al.[6] analysed the influence of quantization to the parameter and image spaces and the width of the line segments. It was reported that the accuracy of the HT was a function of the size of the cell ($\Delta\rho$,$\Delta\Theta$) and the width of the line segments. A method was proposed using a gradient weighting function in the transform to reduce peak scattering. Atiquzzaman et al.[7] stated that a spread of the peak will occur due to the quantisation of the parameter space when decreasing resolutions $\Delta\rho$ and $\Delta\Theta$. A non-iterative algorithm was proposed to detect straight line segments based on the analysis of HT data. This algorithm improves the efficiency of computing and obtains higher accuracy. Based on this algorithm, they also proposed another robust HT method[8] for the determination of the length and the endpoints of a line.

Niblack et al.[9] reported a method to improve HT accuracy which smoothes the HT space prior to finding a peak location and interpolates this peak to find a final sun-bucket peak. Morimoto et al.[10] reported a high resolution HT based on a variable filter, in which the filter is designed and applied to the HT data before detecting the peak. Magli et al.[11] proposed an algorithm based on interpolation and multi-scale matched filtering, in order to achieve high accuracy line detection from the HT. The voting process is one of the hot topics that obtained intensive attention. Ji et al. [12] proposed a statistically efficient HT based on an analytical propagation of input errors. This method used a Bayesian probabilistic scheme to compute the contribution of each feature point to the accumulator. Thus, it improved accuracy and robustness. Shapiro et al. [13] summarised and proved the problem of discrete HT accuracy and adequacy to be the reason for the vote spreading problem. A non-voting Hough-Green transform was proposed to improve the accuracy of HT. Guo et al. [14] modified the HT voting scheme to suppress the impact of noise edges on the accumulation of votes in HT. They used surround suppression to assign the weights of votes for HT. This method improved the quality of detection results. Leandro et al. [15] presented an improved voting scheme for the HT that allowed a soft-ware implementation to achieve real-time performance, even with relatively large images. It improved the performance of the voting scheme and the robustness of HT.

2.2 Decreasing the Computation Load and Storage Requirements

As the applications of straight line recognition obtained intensive attention, the computational cost and storage requirements became a concern. According Eq.(1), the HT first transforms the feature points in the image space into sine curves in the parameter space. The parameter space is divided into an array of “accumulators”, and the accumulator receiving the largest number of votes along these curves is considered the peak. The position of the peak, that is, ρ and Θ, is used to interpret the dominant straight line in the image space. In this way, the HT requires largest or ageand computational requirements, which limit the applications of HT. Therefore, methods to improve the detection speed and decrease the storage requirements in the HT process are being investigated. Avoiding unnecessary accumulators is the main method towards this goal. Li et al.[18] developed Fast HT(FHT) through an hierarchical approach. This approach divided the parameter space into hypercubes, from low to high resolutions. It performs the subdivision and subsequent “vote counting” only on hypercubes with votes exceeding a selected threshold. This greatly reduces both computation and storage. This method can also extend to more complex object detection. Illingworth et al.[19] introduced the Adaptive HT(AHT) for line and circle detection. This method used a small accumulator array and the idea of a flexible iterative “coarse to fine” accumulation. Itutilised a search strategy to identify significant peaks in the Hough parameter spaces. The method increased efficiency and saved storage. Atiquzzaman et al.[20] addressed a multi-resolution implementation of the HT that reduced the computing time. Ben-Tzvi et al.[21] presented an algorithm for computing the HT using information available in the distribution of the image points, rather than depending solely on information extracted from the transform space. Hence, the processing of HT was calculated more efficiently.

3. Comparative study of Selected HT based methods

Among the reviewed research, from the viewpoint of determining the length of lines in an image accurately, three different HT based effective and efficient techniques are selected for comparative study of their performance. First method, applying Microanalysis of the distribution of votes around the peak proposed by Atiquzzaman and M.W. Akhtar[20], which is non-iterative and faster; Second method based on HT butterfly by Shengzi du et al[ ] having immunity from collinear disturbances and third technique by ZeZhong Xu et al. based on statistical analysis of the Hough space.

3.1 Microanalysis of the distribution of votes around the peak

The algorithm proposed by Atiquzzaman and M.W. Akhtar[14] is *based on a microanalysis of the distribution of the votes around the peak in the accumulator array.*

In *[20]* 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^k_f *and* μ^k_l (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 are called the *spread of votes* in C_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[26]. It has been shown that 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 *[20]*, on the accuracy with which the peak can be detected, the end points obtained from the above algorithm are not always reliable. In this algorithm proposed by Atiquzzaman and M.W. Akhtar , the Θ value of the peak *(Θp)* is only used to determine two columns, C_q and C_r as described below.

Two columns C_q, and C_r, were considered, whose cells correspond to the two sets of parallel bars with their normals inclined at angles Θ_q, and Θ_r, respectively with the positive x-axis. The lengths of the normals; $\rho^q{}_1$, $\rho^q{}_2$, $\rho^r{}_1$ and $\rho^r{}_2$ (see Figure 3.1.2) to the bars corresponding to the cells $\mu^q{}_f$, $\mu^q{}_l$, $\mu^r{}_f$ and $\mu^r{}_l$ respectively were determined from the accumulator array.

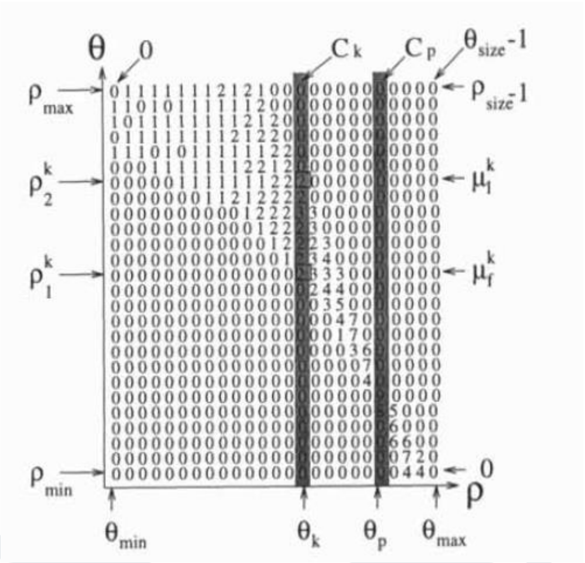


Figure 3.1.1**:** Accumulator array ***A*** showing the cells around the peak.

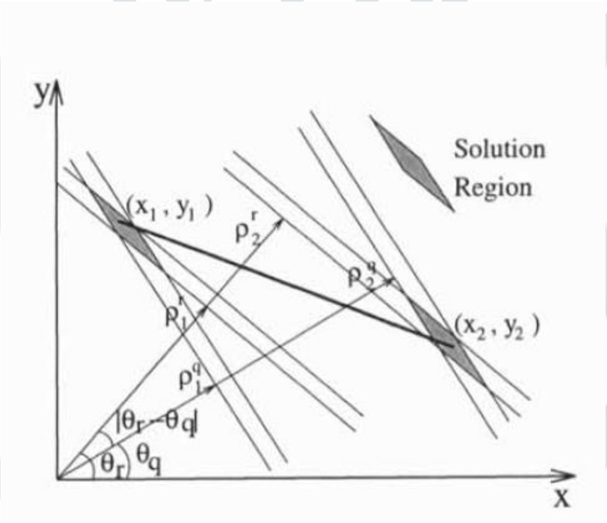


Figure 3.1.2 **:** Computation of the end points independent of Θ, for Θ > **45'.**

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 are expressed as

$$\rho^q{}_1 = x_l \cos \Theta_q + y_1 \sin \Theta_q \qquad (2)$$

$$\rho^r{}_1 = x_l \cos \Theta_r + y_1 \sin \Theta_r \qquad (3)$$

$$\rho^q{}_2 = x_2 \cos \Theta_q + y_2 \sin \Theta_q \qquad (4)$$

$$\rho^r{}_2 = x_2 \cos \Theta_r + y_2 \sin \Theta_r \qquad (5)$$

$\rho^q{}_1$, $\rho^q{}_2$, $\rho^r{}_1$ and $\rho^r{}_2$ were calculated by the following equations

$$\rho^q{}_1 = \rho_{min} + \mu^q{}_f \Delta\rho \qquad (6)$$

$$\rho^q{}_2 = \rho_{min} + \mu^q{}_l \Delta\rho \qquad (7)$$

$$\rho^r{}_1 = \rho_{min} + \mu^r{}_f \Delta\rho \qquad (8)$$

$$\rho^r{}_2 = \rho_{min} + \mu^r{}_l \Delta\rho \qquad (9)$$

Left side parameters in Equations (6)-(9) can be found after determining the first and last nonzero cells in the columns C_q and C_r as ρ_{min} is known. These values can be 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) give the coordinates (***x*₁**, y_l) of one of the end points. The other end point (x_2, y_2) can be obtained similarly by solving Equations (4) and (5).

Once the end points of a line are determined by the above algorithm, the line length (l_c) is obtained from the end points by

$$l_c = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \qquad (10)$$

This algorithm has thus a time complexity of O(1) which is the same as the complexity of the algorithm described in [20]. Note that the time complexity of the algorithms described in [21,22,23 and 24] 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 [24]. This algorithm by Atiquzzaman and M.W. Akhtar differs from the earlier algorithms in its reduced computational complexity and robustness in detecting the end points. Because of its low computational complexity and high robustness, the algorithm can be very suitable for implementation in *real-time* machine vision tasks.

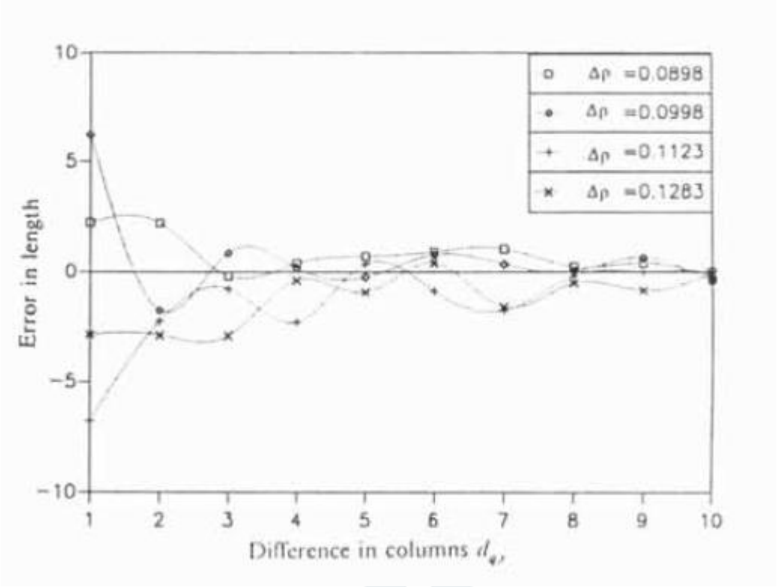


Figure 3.1.3: Errors in the line length vs. d_{qr} (No. of columns between C_q, and C_r) for a line having $\rho_a = 90$ *and* $\Theta_a = 25^0$ [27]

It was mentioned that the errors increase with an increase in Δρ and/or a decrease in ΔΘ**.** This is because the area of the solution region increases with an increase in Δρ and/or a decrease in ΔΘ**.** Results showed that the proposed algorithm can detect the length and the end points very accurately. The algorithm is thus non-iterative in nature and is based on a micro-analysis of the spreading of the votes in the accumulator array.

3.2 Measuring Straight line segments using HT Butterfly

The ‘‘HT butterfly based’’ methods use not only the peak but also the data in the area around the peak. These methods have potentials to uncover segment parameters with high accuracy from low resolution data exploiting the one-to-one mapping between a segment and its butterfly. The features of the butterfly are popularly discussed and employed in these methods, such as the self-similarity [34] and the symmetry [35] are used to improve the resolution and accuracy of HT. The features of butterflies are used to identify and enhance the peak. Collinear segments detection is another important extension of the ‘‘HT butterfly’’ to the commonly used ‘‘peak based’’ HT methods. Because the butterflies of collinear segments intersect on a common peak and are independent/separated at the area beyond the peak, the ‘‘peak based’’ methods obviously lose the distinguishing ability of these segments. However, the ‘‘HT butterfly based’’ methods have this distinguishing ability naturally [36]

Shengzi du et al[ ] proposed a HT butterfly based method to determine the end points of line and to determine the length from it. They generated the butterfly as shown in Fig. 3.2.1(a). For a given segment S_0 lying on a straight line (Θ_0, ρ_0), the HT cells lying on the column with distance δΘ to the HT peak correspond to the belts bounded by parallel straight lines …,l_{n-1}, l_n,l_{n+1},l_{n+2},….. The angle between S_0 and these straight lines is δΘ. All the belts that intersect with S_0 contain some feature points of S_0 and hence the corresponding cells receive some votes. With δΘ increasing, the number of belts intersecting with S_0 increases and the feature points contained in each belt decreases. This means that when the column moves further from the peak, more cells will receive votes. Therefore, a butterfly shaped voting area is generated with δΘ increasing from 0 to a given value. In this manner, given a segment and the scope in HT space, a unique butterfly can be generated. It was implied that if the butterfly is known, one can expect to cover the segment from its HT butterfly.

Representation of segment measurements by the features of its HT butterfly

Segment length vs the width of its HT butterfly Wings :

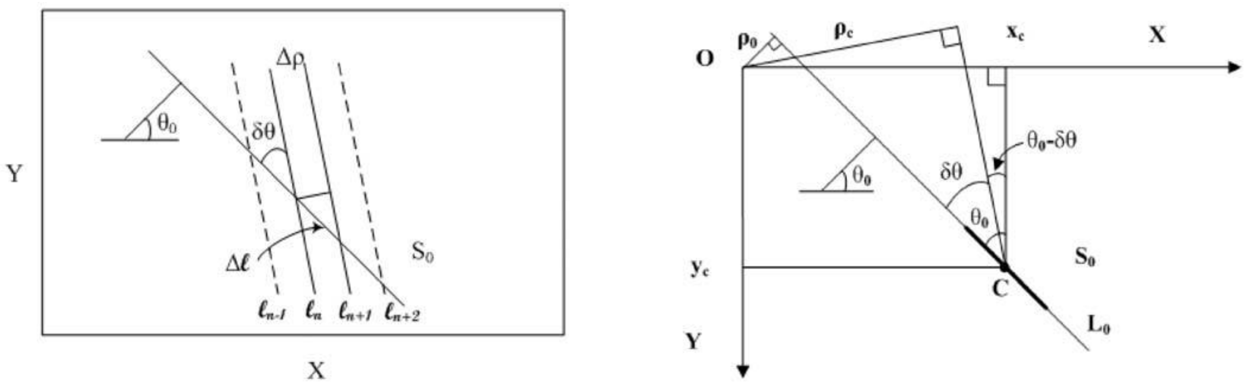
The generation of the butterfly demonstrated in Fig. 3.2.1(a) clearly indicates that for a given δΘ a longer segment leads to more of the cells in the column receiving votes. This implies that the butterfly wings of a longer segment is wider than the ones of a shorter segment. Therefore, it is possible to detect the segment length by means of the width of its butterfly wings. In Fig. 3.2.1(a), it is obvious that the length of the intersection of the segment and a belt (if they intersect) can be obtained as follows:

$$\Delta l = \frac{\Delta\rho}{\sin \delta\Theta} \qquad (11)$$

Therefore, the number of cells in the column, that is, the width of the butterfly wings, corresponding to δΘ can be used to obtain the length of the segment as follows:

$$l = w_{\delta\Theta}\, \Delta l \qquad (12)$$

where $w_{\delta\Theta}$ is the number of cells receiving votes in the column of δΘ as shown in Fig. 1.2 (b), Δ*l* is the length of the intersection of the segment and the belt shown in eq. (12), and *l* is the detected length of the segment. It should be noted that the belts that intersect with the segment at both ends might contain fewer feature points than the ones in the middle, implying that the length of these intersections are smaller than the Δ*l* shown in eq. (12). Hence the detected length may be bigger than the ‘true’ value by up to 2 (Δρ/sin δΘ). This error is depressed when Δρ is small enough and δΘ is large enough.[37]



(a) The generation of butterfly (b) Segment position

Figure 3.2.1. Principle of geometric derivation

Segment width vs the thickness of its HT butterfly wings

The demonstration in Fig. 3.2.1(a) indicates that for a given δΘ the length of the intersection between the segment and the belt is fixed and can be obtained, likewise the number of feature points contained in the intersection, that is, the number of votes received by the corresponding cell, is related to the width of the segment. It is obvious that each intersection, except the ones at both ends, contains similar number of feature points if the segment is uniform. Considering that the number of votes received by the cell is represented by the thickness of the butterfly wings, the thickness of wings is nearly constant for a given column. Therefore, it is possible to derive the segment width via the thickness of its butterfly wings. Considering the nature of the digital images used in the computer, all straight lines are represented by piece-wise connected horizontal or vertical short segments. The number of pixels composing a straight line is equal to the projection of the straight line on X (when $|\Theta_0| > 45^0$) or Y (when $|\Theta_0| \leq 45^0$) axes. For a single-pixel width line, the number of pixels contained in the belt corresponding to δΘ as shown in Fig. 3.2.1(a), that is, the thickness of the butterfly wings corresponding to δΘ, is:

$$\delta^0_{\delta\Theta} = \begin{cases} \Delta l \,.\, \cos\Theta_0, & |\Theta_0| > 45^0 \\ \Delta l \,.\, \sin\Theta_0, & |\Theta_0| \leq 45^0 \end{cases} \qquad (13)$$

Substituting eq. (11) and eq. (12) in (13), one obtains

$$w = \frac{\delta\Theta}{\delta_{\delta\Theta}} \Delta\rho \qquad (14)$$

Δρ and δΘ are predefined, and $\delta_{\delta\Theta}$ can be obtained from the butterfly. Given the butterfly and the value of Δρ, the width of the segment w (counted by pixel) can be obtained.

Thus, in this method the authors formulated the segment measurement by means of simple geometric analysis. The measurement of full parameters of a segment is discussed and represented by the features of its butterfly, including the length, width, position, continuity and non-uniformity of the segment. The relationship between butterfly features and segment parameters were demonstrated, and the illustrations showed that it is reasonable to detect and measure segments via their butterflies. It was found that, the involvement of HT data around peaks renders the relationship independent of the sharpness of the peaks. Based on these relationships, this effective and robust segment measuring method was proposed and applied to synthetic and real images in order to demonstrate the performance and the performance was verified.

Further improvement was obtained by employing the butterfly isolating method, where the edge of the butterfly becomes very clear and the effects of most collinear disturbances/clutters are removed. It means that, this method has immunity from collinear disturbances and hence outperforms others. The authors concluded that based on this improved butterfly, not only this, but also all the methods based on butterflies can get a much better detection performance. The graph showing Length detection accuracy for various values of δΘ was prepared to evaluate the performance.

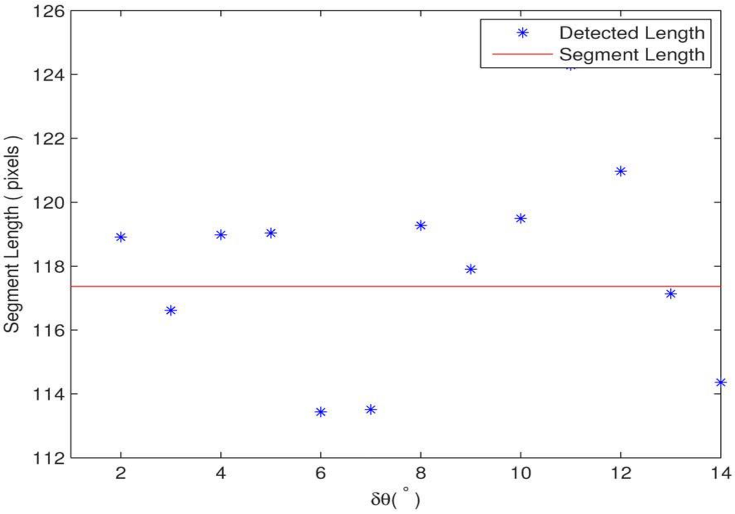


Figure 3.2.2 : Relationship between δΘ and deviation in detected length.

3.3 Length and Width of a Line Segment by Statistical analysis of voting variance.

An HT method for obtaining the length and width of a detected line segment using statistical analysis of voting variance, was proposed by ZeZhong Xu et al. [9]. The voting variance was analyzed in image space, and a 2nd order functional relationship was deduced. Using this method, in Hough space, the statistical variances of columns around a peak was computed and used to derive a quadratic polynomial function. Geometric parameters of a line segment were determined by solving the equations generated by comparing the corresponding coefficients of the two fitted functions.

(a) Voting Analysis in Image Space

All pixels on the line-segment in an image vote for all possible cells (αi; dj) in Hough space. For a pixel, given a voting angle αi ϵ [0; π), the corresponding dj-value is computed. The cell (αi; dj) is voted for by increasing the voting value at this cell by 1. Hij was considered as the voting value of cell (αi; dj) in Hough space.

In this method, for a voting angle αi, the number of voting cells and voting values of each cell are analyzed first, then a functional relationship between voting variance and voting angle was deduced. The actual normal parameters of a line segment are denoted by (α_0; d_0). L and T denoted the length and the width of the line segment. For abbreviation, S and C were used as short form of the values of sine and cosine of | αi - α0 |, respectively:

$$S = \sin | \alpha i - \alpha 0 | \quad \text{and } C = \cos | \alpha i - \alpha 0 | \qquad (3.3.1)$$

Regarding a voting angle αi, the number of voting cells found proportional to the number of parallel bars intersected by the considered line-segment. The voting value H_{ij} , corresponding to the voting angle αi and the distance dj , found proportional to the length of the bar intersected by the line-segment. For detecting line segments with different length and width, two cases were considered for estimating the number of voting cells and voting values.

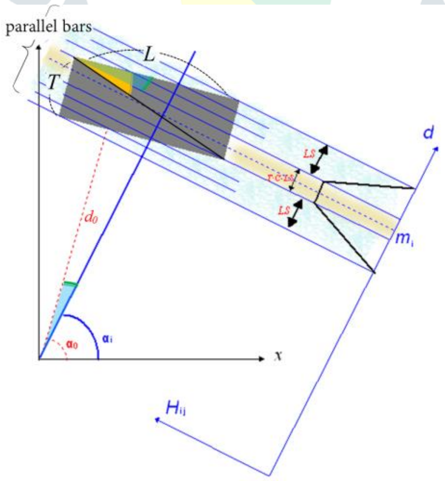


Figure 3.3.1 : The number of voting cells and voting values for | αi - α_0 |< arctan(T/L) .

For a voting angle αi, if | αi - α_0 | < arctan(T/L) then there are T.C +L.S parallel bars crossing the considered line-segment in total. This is illustrated in Fig. 3.3.1. The voting values are found identical at the middle of the parallel bars, hence the number of voting cells equals (T.C - L.S) . On both outer sides of parallel bars, there are L.S voting cells for each side; and the voting values decrease to 0 gradually. For a voting angle αi, if | αi - α_0 | > arctan(T/L) then there are L.S+T .C parallel bars crossing the considered line-segment in total. This is illustrated in Fig.3.3.2. At the middle of the parallel bars,the number of voting cells equals (L. S - T . C) and the voting values are identical. On both outer sides of the parallel bars, there are T.C voting cells on each side; and the voting values decrease to 0 gradually.

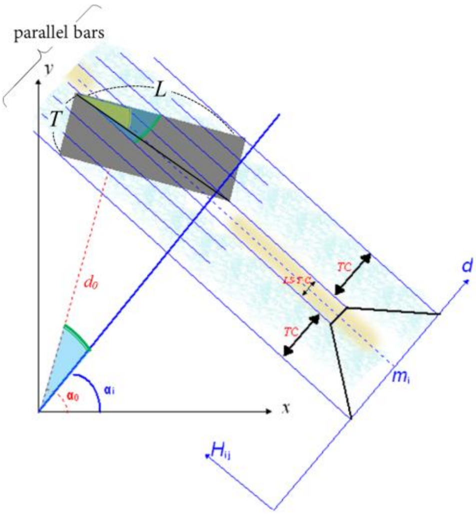


Figure 3.3.2 **:** The number of voting cells and voting values for | αi - α₀ | > arctan(T=L) .

(b) Voting Variances

In both cases for a voting angle αi, the voting cells along the axis d can be considered to be a random variable. The voting values of corresponding cells define a probabilistic density function. The voting variance σ_i^2, which corresponds to voting angle αi, is calculated based on the corresponding probabilistic density function.

For both discussed cases, the voting variance σ_i^2 was calculated as follows:

$$\sigma_i^2 = \frac{L^2 \sin^2 |\alpha i - \alpha_0| + T^2 \cos^2 |\alpha i - \alpha_0|}{12}$$

$$= \frac{(L^2 - T^2) \sin^2 |\alpha i - \alpha_0| + T^2}{12} \qquad (3.3.2)$$

ZeZhong Xu et al. considered the voting cells around the peak in Hough space. It means that | αi - α₀ | is small, and that one can approximate sin | αi - α₀ | by | αi - α₀ |. Thus,

$$\sigma_i^2 = \frac{(L^2 - T^2)\,\alpha i^2 - 2\alpha_0 (L^2 - T^2)\,\alpha i + (L^2 - T^2)\,\alpha_0^2 + T^2}{12} \qquad (3.3.3)$$

This shows that the functional relationship between voting variance σ^2 and voting angle α can be approximated by a 2nd order curve (called f for later reference) as expressed in above equation.

(c) Statistical Distribution in Hough Space

For a line segment in an image, all collinear pixels vote for all possible cells in the Hough space. Due to various uncertainties, the voting in a column is considered as being a random variable. The voting value at each cell defines a probabilistic distribution. The authors computed the statistical variances in columns near the peak and use them to fit a quadratic polynomial curve, called g for later reference. After voting, a peak is detected and represented by α_p; d_p). This was considered as a coarse estimate for the actual normal parameters. In the αi-column which was close to the peak α_p, the middle cells were found with approximately identical voting values. The voting values in the middle cells of the column were found larger than the voting values at outer cells.

Statistical Variances

For each column αi in a peak region, the statistical mean m_i and statistical variance σ_i^2 were computed as follows:

$$m_i = \sum_{j\epsilon W} [H_{ij} \,.\, d_j] \;/\; \sum_{j\epsilon W} H_{ij} \qquad (3.3.4)$$

$$\sigma_i^2 = \sum_{j\epsilon W} [H_{ij} \,.\, (d_j - m_i)^2] \;/\; \sum_{j\epsilon W} H_{ij} \qquad (3.3.5)$$

where W denotes the peak region in the Hough space.

(d) Quadratic Polynomial Curve Fitting

Based on a voting analysis as discussed above, the functional relationship between statistical variance σ_i^2 and angle α was approximated by a quadratic polynomial curve. A quadratic polynomial curve g was fitted to pairs (σ_i^2, αi), all calculated in the peak region. Formally, this was denoted by

$$g : \sigma_i^2 = g(\alpha)$$

$$= e_2\alpha^2 + e_1\alpha + e_0 \qquad (3.3.6)$$

(e) Length and Width of Line-segment

Length and width of a detected line segment was computed based on the coefficients of the fitted function. Using Equations (3.3.3) and (3.3.6), the following equational system was obtained:

$$(L^2 - T^2)/12 \quad = e_2 \qquad (3.3.7)$$

$$- \; 2\alpha_0\,(L^2 - T^2)/12 \quad = e_1 \qquad (3.3.8)$$

$$((L^2 - T^2)\,\alpha_0^{\,2} + T^2)/12 \quad = e_0 \qquad (3.3.9)$$

By solving simultaneously these equations, the length L and width T of the line segment were derived as follows:

$$L = \sqrt{12}\;\sqrt{[e_2 + e_0 - (e_1^{\,2}/4e_2)]} \qquad (3.3.10)$$

$$T = \sqrt{12}\;\sqrt{[e_0 - (e_1^{\,2}/4e_2)]} \qquad (3.3.11)$$

Thus by analyzing the voting variance, a functional relationship between the voting variance and the voting angle was derived. The corresponding statistical variances were computed and used to fit a quadratic polynomial curve g. Three equations are obtained by comparing the coefficients of functions f and g. The length and width of a line segment was calculated by solving simultaneously these three equations.. The accuracy and feasibility of this solution proposed by ZeZhong Xu et al. for line-segment length and width detection was verified by means of experiments.

As this method focused on the accuracy of length and width calculation for a single detected line segment, for the accuracy of length and width determination, three cases were been considered in terms of different quantization steps and noise scales. In the common case there were no noisy pixels, and the quantization of the Hough space equals (Δα, Δd) = (1^0 , 1p). For the coarse-quantization case, parameter quantization was set equal to (Δα, Δd) = (4^0, 4p). For the heavy-noise case, 1,000 noisy pixels were randomly generated in each of the 200 x 200 images used. 500 synthetic images were generated randomly for each of the three cases and this method was tested. For each of the three cases, 500 resulting detection errors for length and width are documented by Figs. 3.3.3, 3.3.4, and 3.3.5.

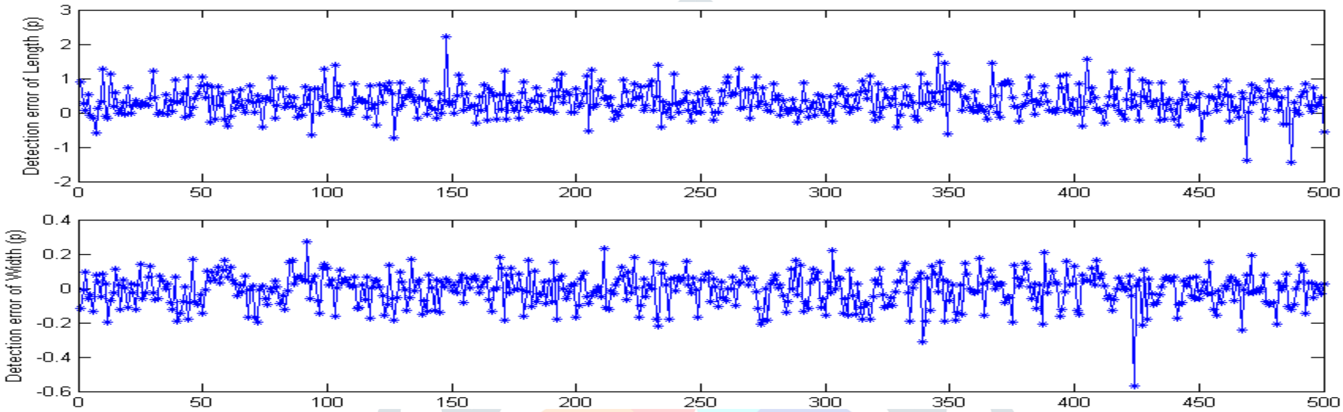


Figure 3.3.3. : Detection errors in the common case. Top: Length. Bottom: Width

In the common case, the calculated values for length and width were found accurate. The mean errors of length and width were equal to 0:4853 and 0:0781, respectively. When the Hough space was quantized at (Δα, Δd) = (4^0, 4p), the mean errors of length and width were equal to 0:5796 and 1:1478 (pixels), respectively. The length detection was accurate, while the width detection found sensitive to the quantization interval Δd.

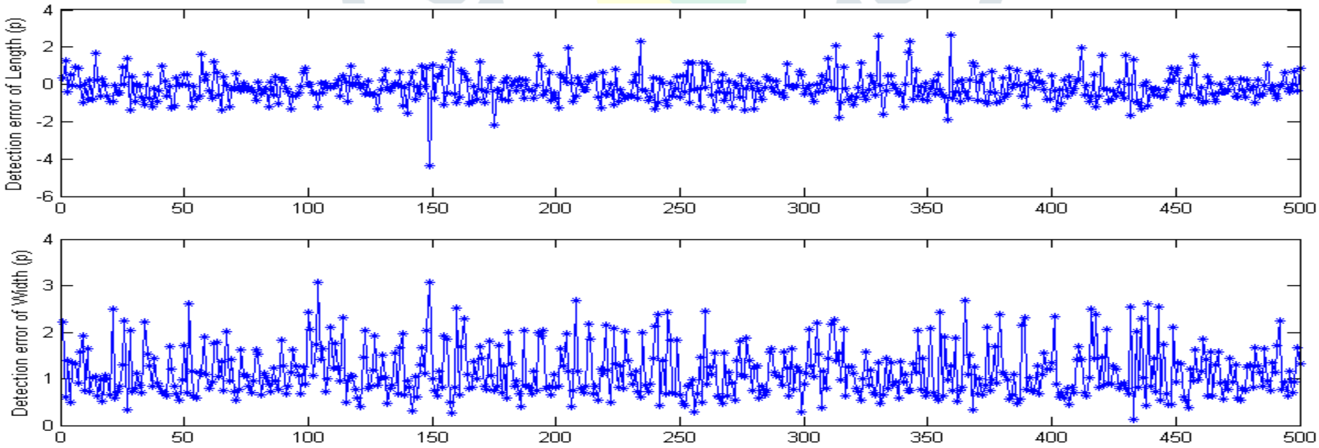


Figure 3.3.4 : Detection errors in the coarse-quantization case. Top: Length. Bottom: Width

By adding 1,000 noisy pixels, the mean errors of length and width were equal to 3:0772 and 0:2613, respectively. The calculated width values were still accurate but length calculation was now effected by the given image noise.

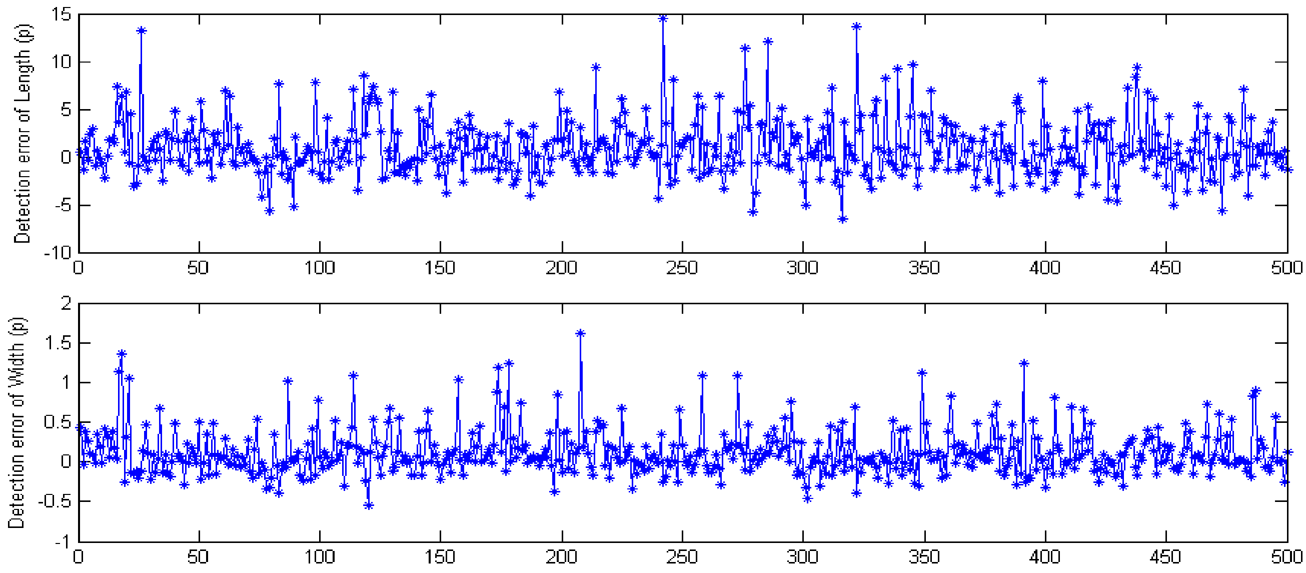


Figure 3.3.5 : Detection errors in the heavy-noise case. Top: Length. Bottom: Width

4. Comparative Performance

The selected three methods as described above were compared by means of accuracy in length detection, computational complexity, storage requirement and accuracy dependence. The comparative performance is summarized in the table No. 1, given below.

Table 1 : Performance comparison of the selected HT based method used for length determination

Method	**Computational complexity**	**Storage requirement**	**Error in length detection**	**Accuracy Depends on**	**Peak detection dependence**
Microanalysis of area around the peak – By Atiquzzaman and M.W. Akhtar	O(1)	Comparatively more. Dependent on Image size and resolution	< 1 pixel	Selection of Cq and Cr	Indirectly dependent on peak detection
HT Butterfly - By Shengzi du et al	$O(\Theta_{Th}\ (P\text{-}P^{Th}))$	Only Selective portion (peak region) need to be stored	0.5 – 2 pixel	Proper selection of δΘ	No
Statistical Analysis of voting variance around a peak – By ZeZhong Xu et al.	Θ(i.j) i – range of quantized angle j – range of quantized length in peak region selected	Only Selective portion (peak region) need to be stored	0.48-0.58 pixel	Image noise	No

As seen from the table, Statistical analysis method is provides more accuracy but is sensitive to image noise. The Microanalysis method is fast and robust so can be used for real time applications. But it needs comparatively more storage space for accumulator processing. The butterfly based method can be effective for length determination constrained to proper selection of quantization angle.

5. Conclusion

In HT, both the image space and the parameter space are discretised which introduces errors in the HT data. The image space discretisation errors are mainly manifested in the position biases of feature points , while the parameter space discretisation errors mainly affect the biases between the peak position and the “true” parameters of the segment. These errors are the main error sources of HT. In this paper various Hough Transform based methods which can be used for geometric feature extraction are reviewed and how the accuracy and complexity was improved has been discussed. From the view point of effective and accurate length determination three different methods are selected and described. Finally the comparative performance evaluation of these methods is presented. In this study, the authors conclude that HT butterfly based method can be robust, accurate and effective if angle quantization is set properly.

ACKNOWLEDGMENT

This work has been supported in part by the Dpark solutions, Buldana. The authors would like to thank Dr. D. R. Dhotre for providing hands on practice of MATLAB's image transformation functions and necessary motivation. Authors acknowledge **HAL,** a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents,

REFERENCES

[1] Corkish, Richard, "Can Solar Cells Ever Recapture the Energy Invested in their Manufacture?". Solar Progress 18 (2): 16-17, 1997.

[2] Ugale K. V. et al. "Centralized Automatic Tracking System for Concentrating Photovoltaic in Villages", RTIT-2007 Conference Proceeding, 2007.

[3] ftp://concord.org/pub/haze/vhs1-download/

[4] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes", Pattern Recognition, Vol.13, No.2, p.111-122, 1981

[5] Biswajit Sit, Md. Iqbal Quraishi, "A Review Paper on Hough Transform and its Applications in Image Processing", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, Special Issue 13, October 2016

[6] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15, January, 1972

[7] Harvey Rhody, "Hough Circle Transform", Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, October 11, 2005

[8] A. A. Kassim, T. Tan, K. H. Tan, "A comparative study of efficient generalised Hough transform techniques" Image and Vision Computing, Volume 17, Issue 10, Pages 737-748, August 1999

[9] Zezhong Xu et al., "Determination of Length and Width of a Line-segment by using a Hough Transform", IEEE Trans. Image Processing, 17, 2014

[10] Aggarwal, N., Karl, W.: Line detection in images through regularized Hough transform. IEEE Trans. Image Processing, 15, 582-591, 2006

[11] Duda, R.O., Hart, P.E.: Use of the Hough transformation to detect lines and curves in pictures. Comm. ACM, 15, 11-15, 1972,

[12] Hough, P.V.C.: Methods and means for recognizing complex patterns. U.S. Patent 3.069.654, 1962

[13] Furukawa, Y., Shinagawa, Y.: Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space. Computer Vision Image Understanding, 92, 1-25, 2003

[14] Akhtar, M.W., Atiquzzaman, M.: Determination of line length using Hough transform. Electronics Letters, 28, 94-96, 1992

[15] Ioannou, D.: Using the Hough transform for determining the length of a digital straight line segment. Electronics Letters, 31 782-784, 1995

[16] Du, S., Tu, C., van Wyk, B.J., Chen, Z.: Collinear segment detection using HT neighbourhoods. IEEE Trans. Image Processing, 20, 3912-3920, 2011

[17] Du, S., Tu, C., van Wyk, B.J., Ochola, E.O., Chen, Z.: Measuring straight line segments using HT butteries. PLoS ONE, 7(3): e33790, 2012

[18] G. K. Damaryam and H. A. Mani, A Preprocessing Scheme for Line Detection with the Hough Transform for Mobile Robot Self-Navigation, In Press, International Organisation for Scientific Research – Journal of Computer Engineering, 18(1), 2016

[19] Gideon Kanji Damaryam, "A Method to Determine End-Points ofStraight Lines Detected Using the Hough Transform", Int. Journal of Engineering Research and Applications www.ijera.com ISSN: 2248-9622, Vol. 6, Issue 1, (Part - 1), pp.67-75, January 2016.

[20] M. Atiquzzaman and M.W. Akhtar, "Complete line segment description using the Hough transform," Image and Vision Computing, vol. 12, no. 5, pp. 267-273, June 1994.

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

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

[23] 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.

[24] 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.

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

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

[27] M. Atiquzzaman M.W. Akhtar, "Determination of End points and Length of a straight line using the Hough Transform", MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-1 5, 1994.

[28] Aggarwal N, Karl W, “Line detection in images through regularized hough transform”. IEEE Transactions on Image Processing 15(3): 582–591, 2006.

[29] Shapiro V , “Accuracy of the straight line hough transform: The non-voting approach”, Computer Vision and Image Understanding 103(1): 1–21, 2006.

[30] Duan H, Liu X, Liu H, “A nonuniform quantization of hough space for the detection of straight line segments”, Proceedings of Pervasive Computing and Applications, 2007 ICPCA 2007. pp 149–153. doi:10.1109/ICPCA.2007.

[31] Walsh D, “Accurate and efficient curve detection in images: the importance sampling Hough transform”, Pattern Recognition 35(7): 1421–1431, 2002.

[32] Yang L, He Z, “Detection of line segments using a fast dynamic hough transform”, Proceedings of 1993 IEEE International Symposium on Circuits and Systems (ISCAS ’93) 1: 543–546, 1993.

[33] Murakami K, Naruse T, “High speed line detection by hough tranform in local area”, Proceedings of 15th International Conference on Pattern Recognition 3: 467–470, 2000

[34] C. Tu BvWKD S Du, Hamam Y., “High resolution hough transform based on butterfly self-similarity”, Electronics Letters 47(26): 1360–1361, 2011.

[35] S Du MS C Tu, “High accuracy hough transform based on butterfly symmetry”, Electronics Letters 48, 2012.

[36] Du S, van Wyk BJ, Tu C, Zhang X, “An improved hough transform neighborhood map for straight line segments”, IEEE trans Image Process 19(3): 573–585, 2010.

[37] Shengzhi Du et al., “Measuring Straight Line Segments Using HT Butterflies”, PLoS ONE, www.plosone.org, 1 March 2012, Volume 7, Issue 3

[38] Choplin, R., “Picture archiving and communication systems: an overview”, Radiographic January 1992 12:127-129, 1992.

[39] J. Gao, Y. Wang, J. Bao, “Valid region recognition in digital images of medical X-ray imaging” Image Information Processing Lab, Hefei University of Technology, Hefei, 2000.

[40] R. K. Satzoda, S. Sathyanarayana, T. Srikanthan, and S. Sathyanarayana, "Hierarchical Additive Hough Transform for Lane Detection", IEEE Embedded Systems Letters, Vol. 2, No. 2, 2010, pp. 23-26.

[41] A. K. Bajwa, and R. Kaur, " Fast Lane Detection Using Improved Hough Transform", Journal of Computing Technologies, Vol. 2, No. 5, 2013, pp. 10-13.

[42] T. Sun, S. Tang, J. Wang, and W. Zhang, "A Robust Lane Detection Method for Autonomous Car-like Robot", in Fourth International Conference on Intelligent Control and Information Processing (ICICIP), 2013, pp. 373 - 378.

[43] A. F. Cela, and L. M. Bergasa, "Lanes Detection Based on Unsupervised and Adaptive Classifier", in Fifth International Conference on Computational Intelligence, Communication Systems and Networks, 2013, pp. 228- 233.

[44] S. Pavuluri, K. Fujimura, and S. Wood, "Robust Lane Localization Using Multiple Cues on the Road”, in IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 , pp. 141 - 146.

[45] S. Lee, H. Son, and K. Min, "Implementation of Lane Detection System Using Optimized Hough Transform Circuit", in IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2010, pp. 406 - 409.

[46] X. Zhe, and L. Zhifeng, "A Robust Lane Detection Method in the Different Scenarios", in IEEE International Conference on Mechatronics and Automation, 2012, pp. 1358 - 1363.