Mid-Point Hough Transform : A Fast Line Detection Method

Hari C.V.
Jyothi Engineering College
Cheruthuruthy, Thrissur
Kerala, India − 679531
Email: haricv@gmail.com

Jojish Joseph V., Sarath Gopi, Felix V.P.
Naval Physical and Oceanographic Laboratory
Kerala, Kochi, India − 692021
Email: {jojish,gopisarath}@gmail.com
v_paulfelix@yahoo.co.in

Amudha J.
Amrita School of Engineering
Coimbatore, Tamilnadu
India − 641105
Email: ms_amudha@yahoo.com

***Abstract*—This paper proposes a new method for detection of lines in images. The new algorithm is a modification of the Standard Hough Transform by considering a pair of pixels simultaneously and mapping them to the parameter space. The proposed algorithm is compared with line detection algorithms like Standard Hough Transform, Randomized Hough Transform and its variants and has advantages of smaller storage and higher speed.**

I. Introduction

Detection of lines in a image is one of the basic task in image processing. It has got several applications in a number of fields like sonars, mobile applications and biometric security systems. In sonar systems the presence of a target and its features can be obtained from the detected line.

The Hough Transform(HT)[1] is a powerful global method for line detection. It transforms between the image space and parameter space, in which a straight line can be defined. A line in the image space is mapped to a point in the parameter space. Similarly, each pixel of the image space is transformed to a parameterized curve of the parameter space. Each transformed point in the parameter space is considered as a candidate for being a line and accumulated in the corresponding cell of an accumulator. Finally, a cell with a local maximum of scores is selected, and its parameter coordinates are used to represent a line segment in the image space. Randomized Hough Transform (RHT) [2] is an improvised version of Standard Hough Transform (SHT) for line detection. The basic idea behind the Randomized Hough Transform is that, instead of transforming one pixel from image space to parameter space, two or more pixels are randomly selected and mapped to a point in the parameter space.

Q. Ji and Y. Xie proposed [3] a method for line detection and circle detection using Randomized Hough Transform based on error propagation which improved detection robustness and accuracy by analytically propagating the errors with image pixels to the estimated curve parameters. Chan-Ta Ho and Ling-Hwei Chen [4] introduced a high speed method for line detection using the geometric property of a pair of parallel lines. R.S.Stephen [5] proposed a probabilistic Hough transform based method where he proved a strong relationship between the Hough Transform and the Maximum Likelihood method. Ranga Rodrigo, Wenxia Shi and Jagath Samarabandu [6] considered straight line detection as an energy minimization problem and proposed an energy based line detection.

In this paper we are proposing a new line detection algorithm which is a modification of the Standard Hough Transform by considering more number of pixels simultaneously and mapping them to the parameter space. Each pixel mapping to the parameter space can be considered as one transaction. Thus the total number of transactions performed would be a metric of the computational complexity of the algorithm. The proposed algorithm is also compared with line detection algorithms like the Standard Hough Transform, Randomized Hough Transform and its variants. The algorithm gives better performance in terms of speed and memory resources used as compared to the standard Hough transform and better detection capability as compared to Randomized Hough Transform.

II. Hough Transform Algorithms for Line Detection

Any straight line can be represented as

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

Here the parameter ρ represents the perpendicular distance from the origin to the line and θ represents the angle of the vector from the origin to the closest point on the line. It is therefore possible to associate with each line of the image, a couple (ρ,θ) which is unique if $\theta \in [0, \pi]$ and $\rho \in R$, or if $\theta \in [0, 2\pi]$ and $\rho > 0$. The (ρ,θ) plane is referred to as the Hough space.

A. Standard Hough Transform

Every pixel in the image is mapped to the Hough space for all values of θ. This leads to a sine wave in the Hough space for each individual pixel. The detection using Hough Transform is based on a voting procedure. This voting procedure is carried out in a parameter space or accumulator. In Standard Hough Transform method each pixel is mapped into the Hough space and the accumulator value for the corresponding parameters are updated. By finding the bins with the highest values in the accumulator space the most likely lines can be extracted and their geometric definitions read off. The advantage of Hough Transform based detection is that lines need not be continuous thus enabling detection even in presence of noise. The main

978-1-4244-4859-3/09/$25.00 ©2009

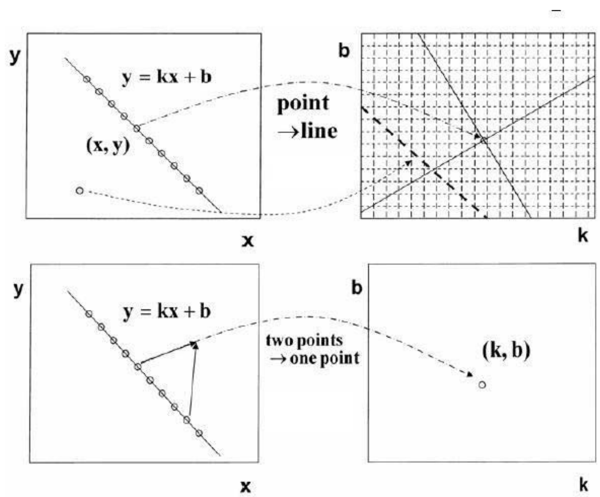

Fig. 1. Randomized Hough Transform[2]

disadvantage is the high computational and memory overhead with the size of the accumulator and the parameter space being the limiting factors.

B. Randomized Hough Transform

The Randomized Hough Transform [2] [7] is an improved version of Standard Hough Transform. This method randomly picks two or more pixels and maps them in to one point in the parameter space. So instead of each and every pixel being mapped into all possible combinations of ρ and θ, a pair of pixels is mapped into a point in the (ρ,θ) space. The accumulator size needed for performing this mapping is much less than the Standard Hough Transform approach. The Randomized Hough Transform is shown in Fig.1

The basic idea behind RHT is summarized as follows. A straight line can be expressed by $y = mx + c$. In this equation the parameters of the line are m and c. If two pixels are in the same line, its m and c values will be same in the parameter space. We know that the parametric equation of line is $\rho = x\cos\theta + y\sin\theta$. So, if two points (x_1, y_1) and (x_2, y_2) are on the same line, it's (ρ, θ) values will be same in the parameter space. The Randomized Hough Transform reduces the number of computations and memory when compared to the Standard Hough Transform and offers infinite parameter space as well as arbitrarily high resolution.

Randomized Hough Transform picks two points simultaneously to detect lines. It can use n points to find an n parameter curve. The method being to solve the n parameters for curve from the n linear equations using the n points selected at random. A modification for the line detection algorithm can also be attempted using more number of random points simultaneously. We try to decrease the errors in parameter estimation for lines obtained by using Randomized Hough Transform by selecting more than two points for estimating ρ and θ. This method is explained below.

1) Randomized Hough Transform using 3 points (RHT-3)
Consider a simple modification of the Randomized Hough Transform using a pair of points. We take three points simultaneously at random from the set of ON pixels from the image. Taking any two sets out of those we transform them to the (ρ,θ) space. We get two discrete points in the (ρ,θ) space. If the three points that we selected were collinear then all possible combinations of them would map to the same point in (ρ,θ) space. This property is made use of in Randomized Hough Transform using 3 points. The random points are taken three at a time. The accumulator values of the (ρ,θ) space is updated only if the first two sets of pairs among the three points selected evaluate to the same (ρ,θ) value. This reduces the number of accumulator cells and consequently the memory needed and improves the error performance.
2) Randomized Hough Transform using 4 points (RHT-4)
The idea given above for RHT-3 can be extended by taking four points at a time. Four points would give six possible pair selections. The (ρ,θ) values for each of them has to be found out. Then a voting procedure is followed. If three or more pairs evaluate to the same (ρ,θ) value that particular accumulator bin is updated. RHT-4 further refines the RHT-3 results but with a caveat of being prone to miss out on shorter lines. Both RHT-4 and RHT-3 use lesser memory than Randomized Hough Transform and give better error performance.

III. Proposed Algorithm - Mid-Point Hough Transform

The new method calculates the mid-point of the two selected random points from binary image. If the mid-point is also an ON pixel we map the random points to the parameter space else we do not map any of those pixels at all. By mapping we mean that for each value of θ we find the corresponding ρ value for the pixel under consideration. We then proceed to pick the next pair of random pixels and apply the same logic on them.

A. Mid-point Hough Transform Algorithm (MP1)

The algorithm of Mid-Point Hough Transform method is described below.

1) Initialize a blank array H.
2) Randomly select two distinct bright or *ON* points $A(x_0, y_0)$ and $B(x_1, y_1)$.
3) Then calculate the mid-point $M(x_m, y_m)$ of A and B

$$M(x_m, y_m) = \left(\frac{x_0 + x_1}{2}, \frac{y_0 + y_1}{2}\right)$$

4) If M is also a bright or *ON* points of the same image, then transform the point M(x_m, y_m) to the parameter space and select other random points. This transformation is represented as

$$\rho = x_m cos\theta + y_m sin\theta$$

and

$$H(\rho, \theta) = H(\rho, \theta) + 1$$

5) If M is not a bright or ON point of the image, then select other random points and repeat the same method.
6) Continue the same method, until no bright or *ON* points are present in the image.

B. Advantages and Disadvantages

The transformation from image space to parameter space is done only if the mid-point is also a bright or *ON* point. So the number of transformations from image space to parameter space is very less when compared to Standard Hough Transform. Due to this, less memory is required for the implementation. This algorithm also requires lesser computational complexity when compared to Standard Hough Transform. The main disadvantage of this algorithm is the missing of small line segments which is a common drawback among all randomized algorithms.

Since only one point is mapped for every pair of points taken, the estimation of parameter (ρ, θ) is prone to errors. This can be alleviated at the cost of added processing by mapping the selected pair also along with the midpoint. It will smooth out the estimation errors in the parameter space as more number of points are now mapped to the parameter space. The modified algorithm is explained below.

C. Modified Mid-point Hough Transform Algorithm (MP2)

1) Initialize a blank array H.
2) Randomly select two bright or *ON* points $A(x_0, y_0)$ and $B(x_1, y_1)$ and ensure both A and B are not equal.
3) Then calculate the mid-point $M(x_m, y_m)$ of A and B

$$M(x_m, y_m) = \left(\frac{x_0 + x_1}{2}, \frac{y_0 + y_1}{2}\right)$$

4) If M is also a bright or *ON* points of the same image, then calculate the parameters ρ and θ for all three points using the Standard Hough Transform equation. If the parameters ρ and θ are equal for all three points, then

$$H(\rho, \theta) = H(\rho, \theta) + 1$$

5) If M is not a bright or *ON* point of the image, then select other random points and repeat the same method.
6) Continue the same method, until no bright or *ON* points are present in the image.

IV. Performance Evaluation

The proposed algorithms were evaluated on Matlab 7.3 and realized on a SHARC based signal processing board. The implementations were tested with simulated SONAR time history data containing multiple targets in the waterfall. The performance of the new methods are compared with the Standard Hough Transform algorithm and Randomized Hough Transform algorithms. The comparison scheme is as follows. A number of images with known (ρ, θ) values are selected as input images. The line is detected using the proposed techniques and the other algorithms. The mean square error between the parameters of the actual and detected line is computed. This is repeated for different signal to noise ratios (SNR). Different SNRs are obtained by synthetically adding white Gaussian noise to the image. Fig. 3 shows the mean square error in various SNR for different algorithms. Fig. 4 shows the number of transactions at various SNR for different algorithms. The number of transactions is the number of computational operations, which is essentially a measure of time consumed for estimation of the parameters. Notations SHT and RHTn represent the Standard Hough Transform and Randomized Hough Transform with n points. Mid-point

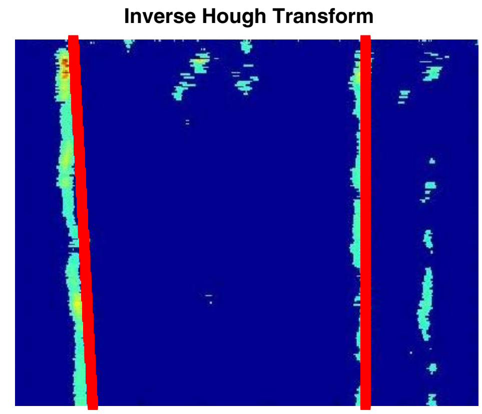

Fig. 2. Line Detected in Real SONAR Image

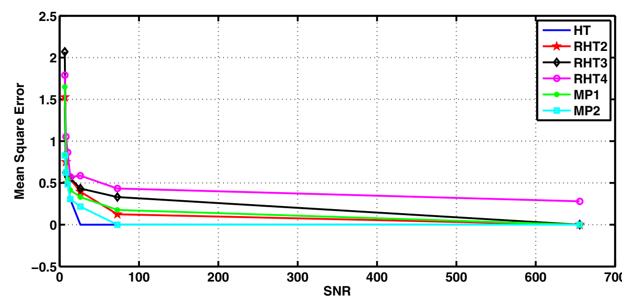

Fig. 3. Comparison of various algorithm in terms of Mean Square Error

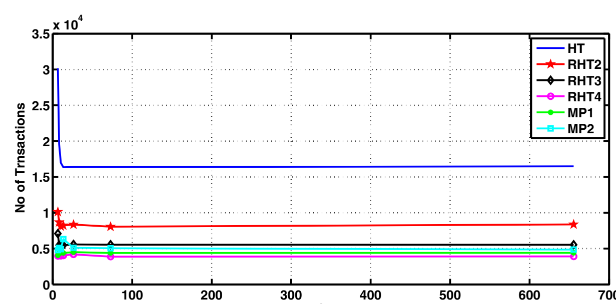

Fig. 4. Comparison of various algorithm in terms of No of Transactions

Hough Transform algorithm and Modified Mid-point algorithms are named as MP1 and MP2 respectively in the figures. Fig. 2 shows the line detected out of a real SONAR image using the proposed algorithm.

It can be seen that the error performance is best for the Standard Hough Transform. But the number of transactions is the maximum for Standard Hough Transform algorithm. The proposed algorithm performs better than the Randomized Hough Transform algorithms. Among the two versions of the proposed algorithms, the Modified Mid-Point Hough Transform algorithm offers better error performance, but at the cost of increased computation.

V. Conclusion

Two new methods of detecting lines using Hough Transform are proposed. The algorithms were simulated on Matlab and a real time implementation was done using ADSP-SHARC based DSP boards. The test vectors used were simulated SONAR time-history waterfall plots containing multiple targets. The Mid-Point Hough Transform method(MP1) is verified to give better results than the Randomized Hough Transform approach and is closer to the Standard Hough Transform results with the minimum number of computational and memory resources. A modification for this algorithm(MP2) is also proposed and gives better error performance while slightly increasing the computational load.

References

[1] R. Duda and P. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Comm. ACM*, vol. 15, pp. 11–15, 1972.

[2] E. O. Lei Xu and P. Kultanen, "A new curve detection method - randomized hough transform (rht)," *Patter Recognition Letters*, vol. 11, pp. 331–338, 1990.

[3] Q. Ji and Y. Xie, "Randomized hough transform with error propagation for line and circle detection," *Patter Analysis Applications*, vol. 6, pp. 55–64, 2003.

[4] C.-T. Ho and L.-H. Chen, "A High Speed Algorithm for Line Detection," *Pattern Recognition Letters*, vol. 17, pp. 467–473, 1996.

[5] R. S. Stephen, "A Probabilistic Approach to the Hough Transform," 1991.

[6] W. S. Ranga Rodrigo and J. Samarabandu, "Energy based line detection," 2006.

[7] L. Xu and E. Oja, "Randomized hough transform," *Encyclopedia of Artificial Intelligence*, pp. 1354–1362, 2008.