MVA '98 IAPR Workshop on Machine Vision Applications, Nov. 17-19, 1998, Makuhari, Chiba, Japan

3—7 On a High-Speed Hough Transform AlgorithmMRHT

Kunihito Kato *
Faculty of Engineering,
Gifu University

Toshio Endo
Fujitsu Laboratories Ltd.

Kazuhito Murakami
Faculty of I.S.T.,
Aichi Pref. University

Takashi Toriu
Fujitsu Laboratories Ltd.

Hiroyasu Koshimizu
School of Computer and Cognitive Sciences,
Chukyo University

Abstract

Line detection using Hough transform is one of the robust image processing methods for noisy images. But Hough transform has a problem whose computation cost is very large in general. In order to ease this problem, many high-speed algorithms had been proposed. In this paper, we propose a new high-speed algorithm called MRHT (Multiple Randomized Hough Transform) which combines randomized edge point selection process of RHT and block division process of CHT. And we clarify experimentally and theoretically that the MRHT is faster than RHT and CHT.

1 Introduction

Hough transform is very useful algorithm for pattern recognition, especially for line detection. Therefore, now a days, this algorithm is applied for many application fields of image processing. But memory cost and computation cost problem were always inevitable. Thus, many high-speed or high performance algorithms had been proposed. In this paper, we propose a new high-speed Hough transform algorithm MRHT.

Xu and Oja[1] proposed RHT (Randomized Hough Transform) which reduces the computation cost by selecting the pair of the edge points at random. On the other hand, D. Ben-Tzvi and M. B. Sandler[2] proposed CHT (Combinatorial Hough Transform) which reduces the computation by dividing the image to some blocks and by limiting the combination of the pairs of the edge points in these blocks.

Our MRHT algorithm is basically constructed by combining randomized edge point selection process of RHT and block division process of CHT. Therefore, this MRHT has two randomized selection processes. First process is randomized selection of a block from the divided blocks at random. In second process, MRHT selects the pair of the edge points at random from the selected block in the previous selection. Thus, MRHT has multiply randomized selection processes in two serial steps.

*Address: 1-1 Yanagido, Gifu, 501-1193, JAPAN. E-mail: kkato@info.gifu-u.ac.jp

2 Algorithm of MRHT

In RHT algorithm, a vote which is calculated from two edge points (x_i, y_i) and (x_j, y_j) randomly selected to the cell (θ_k, ρ_k) is executed on the parameter space. RHT is expected to reduce the computation cost by this operation. On the other hand, in CHT algorithm, the pairs of the edge points are voted to the parameter space just as same way as RHT. Although the number of combinations of all edge points in the image became very large, CHT algorithm improves this problem by dividing the image into the disjoint some blocks in order to suppress the number of combinations of edge points.

MRHT is basically an extension of CHT with the randomized edge point selection process of RHT, or an extension RHT with the image division process of CHT. But there are some important differences between these two algorithms: CHT is designed to detect the peaks on the parameter space after the voting of all the edge points, and RHT is designed to detect the peaks repeatedly in the line detection procedure. Therefore, it is not easy to combine these two algorithms because of the peak detection procedures in these two algorithms are quite different.

Figure 1 shows the flow of MRHT algorithm. MRHT has two randomized selection processes to solve this problem. First process is to select at random one block from the divided blocks. In second process, MRHT selects the pair of the edge points at random from the selected block in the previous selection. Thus, the multiply randomized selection processes provided the basic characteristics of MRHT from the probabilistical viewpoint.

In MRHT algorithm, the voting is executed to the cell (θ_k, ρ_k) on the parameter space which is calculated from two edge points (x_i, y_i) and (x_j, y_j) randomly selected from the block. When one cell in the two dimensional array which represents the parameter space becomes greater than the threshold, the

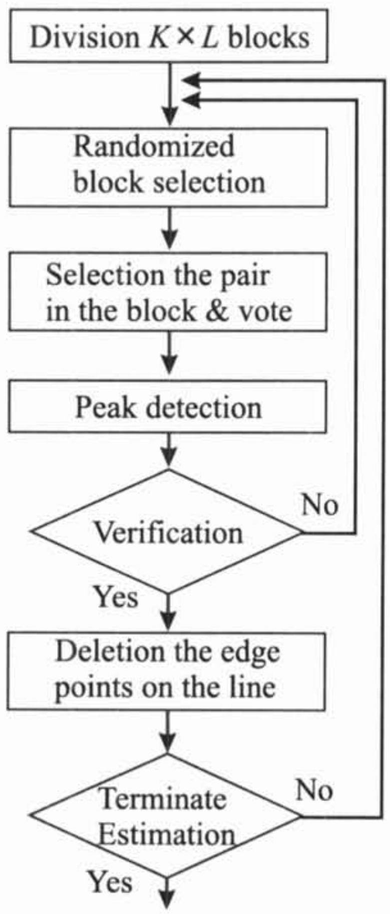


Figure 1: Flow of MRHT

process of line detection is terminated and the cell (θ_k, ρ_k) is detected as a peak.

MRHT has the block division process at first. Thus, when the peak is detected by the thresholding procedure from the parameter space, we have to normalize the threshold because of the total number of the pairs of the edge points will be gradually reduced in this process. Since image is divided into $X \times Y$ blocks, the length of the line segments in this image is in proportion to $\sqrt{X \times Y}$. Therefore, it would be adequate to let the threshold in the 1×1 division be t, and let the threshold in condition of the $X \times Y$ division be $t' = t/(\sqrt{X \times Y})$.

After this peak detection, all edge points on the line corresponding to this peak are deleted as shown in Fig. 2. Then, parameter space is initialized, and the voting procedure is resumed. This procedure is just the same as RHT algorithm. This procedure is repeated until the ending condition is satisfied.

3 Theoretical consideration of MRHT with RHT and CHT

3.1 Comparison between MRHT and CHT

Let the total number of the edge points in the image be N, and all of the combination be ${}_NC_2$. In CHT algorithm, let the number of blocks be $K(K = X \times Y)$, and the number of the combinations of the edge points be ${}_{N/K}C_2 \times K$. Therefore, the computation performance is improved, because total number of the pairs of the edge points will become small.

The line detection process of CHT is basically equivalent to the pair voting type Hough transform without the block division process. And MRHT is basically an extension of CHT with the randomized edge point selection process of RHT. Thus, considering that the computation cost of these algorithms is estimated by the voting counts, relation between MRHT and CHT in computation cost is equal to the relation between the pair voting type Hough transform and RHT. Therefore, it is clear that RHT is faster than the pair voting type Hough transform, and that the computation cost of MRHT is smaller than CHT.

3.2 Comparison between MRHT and RHT

We compare the computation costs between MRHT and RHT basing on a measure of the number of votings until a moment when the first line is detected. In RHT algorithm, let the total number of the edge points in the image be N, average number of the edge points on the one line be m, and the threshold to detect the peak in the parameter space be t. Then, the probability that a voting generated by the specific pair of the edge points hits a cell in the parameter space corresponding to the specific line becomes ${}_mC_2/{}_NC_2$. Therefore, the number of the edge points until when the votes of one cell grown up greater than threshold t can be estimated as

$$n_{RHT} = t\,{}_NC_2/{}_mC_2 \quad (1)$$

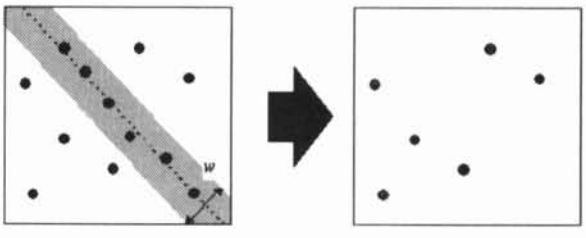


Figure 2: Edge point deletion process after the line detection

On the other hands, since the number of blocks is K, the number of the combination of the edge points becomes ${}_{N/K}C_2 \times K$. Since line segments in the image are divided into L segments by the block division process in MRHT algorithm procedure, the total number of the pairs of the edge points on the specific line becomes ${}_{m/L}C_2 \times L$. And the probability that a voting generated by the specific pair of the edge points hits a cell becomes $({}_{m/L}C_2 \times L)/({}_{N/K}C_2 \times K)$. The number of the edge points until when the votes of one cell grown up greater than threshold t' which normalized by K division MRHT can be estimated as

$$n_{MRHT} = (t_{N/K}C_2 \times \sqrt{K})/({}_{m/L}C_2 \times L) \quad (2)$$

Thus, since $n_{RHT} > n_{MRHT}$ is satisfied, it is known that the voting counts of MRHT is smaller than that of RHT.

4 Experiments

We present a line detection experiment for comparing MRHT with RHT and CHT. The input image has about 50 line segments, and the image size is 640×480 shown in Fig. 3. And we experimented under the division conditions of 1×1 (=RHT), 2×2, 3×3, ···, 16×16 with this input image. And we compared MRHT with RHT based on these conditions by means of the voting counts, which is the average value of 10 times experiments.

Figure 4 shows a results of RHT, and Fig.5 (a),···,(d) show the results of MRHT with conditions of 2×2, 4×4, 8×8 and 16×16 division, respectively. We show the voting counts to detect the lines in Table 1. Figure 6 shows the graph of this Table 1, where the vertical axis means voting counts, and horizontal axis means the division conditions. And Figure 7 shows the graph of the processing time, which is the average value of 50 times experiments.

The result of RHT, that is MRHT(1×1) gives the best result in the performance for line detection. The performance of line detection of MRHT became gradually worse until 2×2, 4×4,···. But CHT algorithm has the same problem of the degradation of the performance of line detection.

From the viewpoint of computation time, 8×8 division MRHT gives 56 times faster results than the results of RHT in experiment of voting counts, and 8 times faster than CHT because of voting count in 8×8 division CHT is 834,785. But in experiment of real computation time, MRHT is not always faster than CHT. Because, there are some over head in MRHT algorithm sequence, for example such as edge points deletion process or parameter space initialization process. The most fast result in MRHT is 1.37 seconds in 5×5 division. In CHT algorithm, we have to divide to 15×15 blocks to get same the speed as MRHT.

5 Conclusion

In this paper, we proposed a new algorithm MRHT that combines randomized edge point selection process of RHT and block division process of CHT. And we showed experimentally that the MRHT is quite faster than RHT and CHT.

In CHT algorithm, we have to suffer from the degradation sacrifice of the quality of the detected lines, because we divide the image to many blocks to suppress the computation cost. But it was known that MRHT becomes very fast by a few block divisions, and therefore that MRHT do not suffer from any fatal degradation of the quality of the detected lines.

Our future subjects are to investingate the effect caused by the block division using some probabilistic models and to develop the methods providing the best division condition corresponding to the properties of input image.

Acknowledgments

This paper was partially supported by 1997-8 Grant-in-Aid for General Scientific Research, 1997-8 IMS Research Promotion, 1996-7 the Promotion of Advanced Software Enrichment (IPA), and 1998 High-Tech. Research center Promotion. We would like to express thanks to these supports.

References

[1] L.Xu and E.Oja:"Randomized Hough transform (RHT): Basic mechanisms,algorithms,and computational complexities", IU,57,2, pp.131-154 (1993)

[2] D.Ben-Tzvi, M.B.Sandler:"A Combinatorial Hough Transform", Pattern Recognition Letters, 11, pp.16-174 (1993)

[3] K. Kato, T. Endo, K. Murakami, T. Toriu, H. Koshimizu: "Theoretical & Experimental Considerations on the Detectability of Lines for Randomized Voting Hough Transform", Technical report of IEICE Vol.96, No.384, (PRMU96-87), pp53-60 (1996.11) (in Japanese)

[4] K. Kato, K. Murakami, H. Koshimizu : "Segmentation Algorithm of Lines of Hough Transform Using Discriminant Analysis", Proc. of ACCV'95, Vol.3, pp. 191-195 (Singapore) (Dec.1995)

[5] J.Hayashi, K. Kato, T. Endo, K. Murakami, T. Toriu, H. Koshimizu : "On Typical Implementation of Hough Transform for Improving Its Performances", Proc. of ACCV'98, Vol.2, pp. 1-8 (Hong Kong) (Jan.1998)

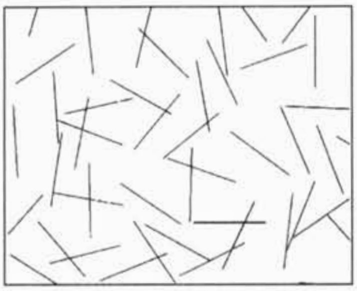


Fig.3. Input image

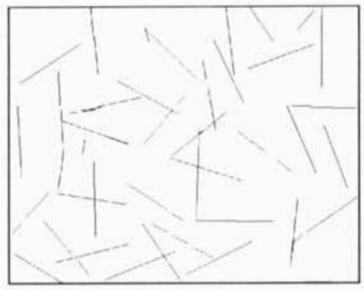
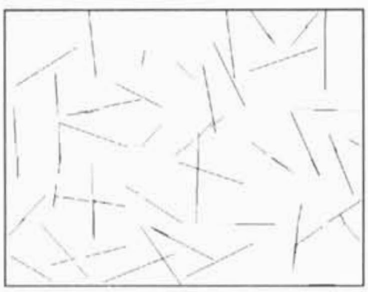
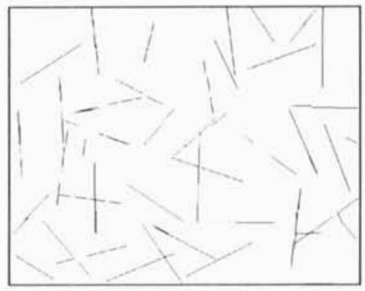


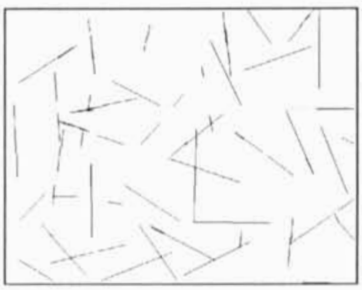
Fig.4. Result of RHT



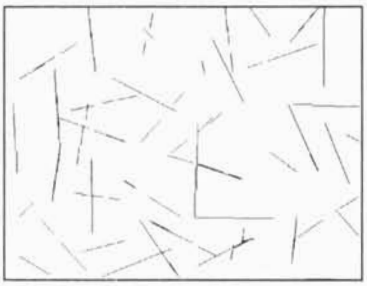
(a) 2×2



(b) 4×4



(c) 8×8



(d) 16×16

Fig.5. Results of MRHT

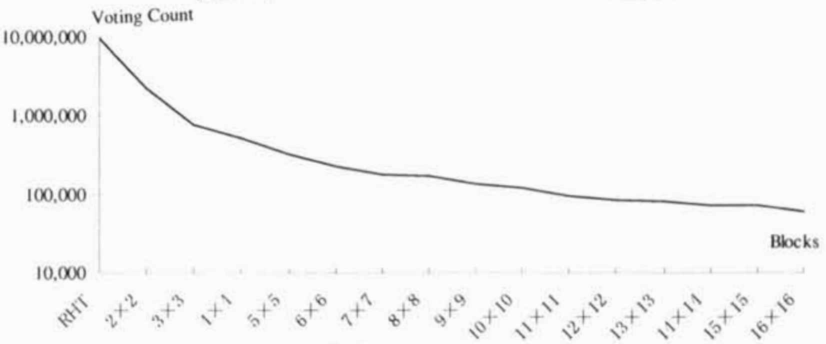


Fig.6. Graph of the voting counts

Table 1. Voting counts for several divisions

Blocks	1×1 (=RHT)	2×2	3×3	4×4	5×5	6×6	7×7	8×8
Voting Counts	9,585,812	2,224,660	767,868	526,415	318,548	230,231	178,667	170,063
Blocks	9×9	10×10	11×11	12×12	13×13	14×14	15×15	16×16
Voting Counts	134,643	121,528	96,370	84,948	81,059	72,476	72,964	60,068

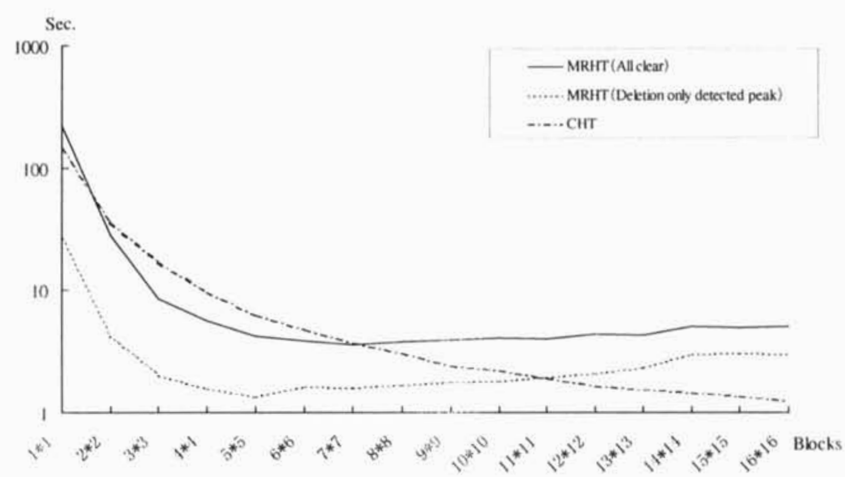


Fig. 7 Detection times by several divisions