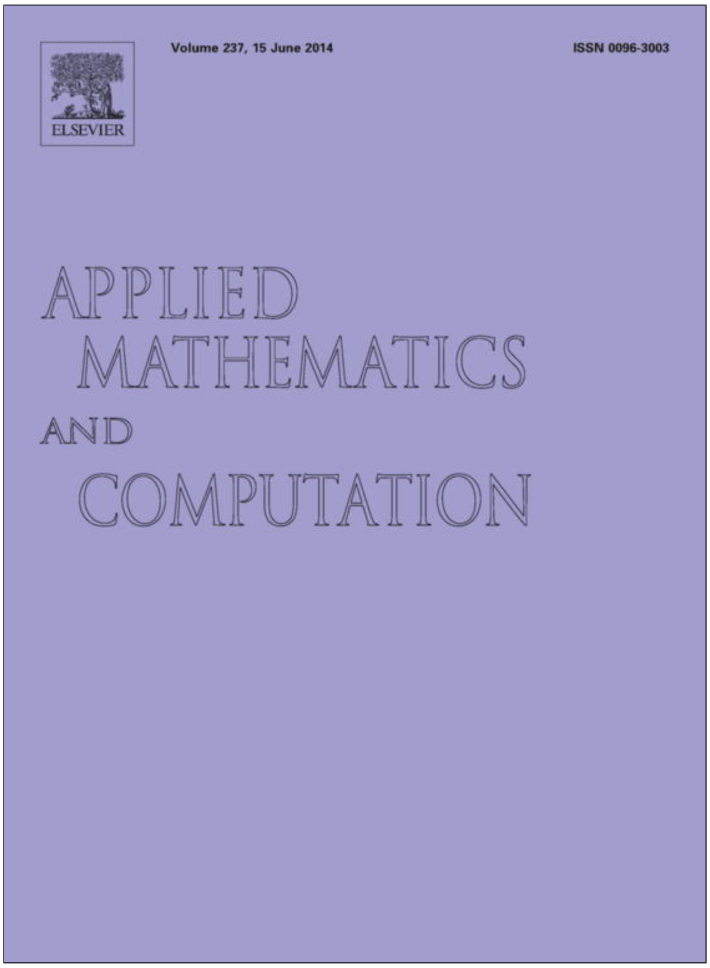
Volume 237, 15 June 2014
ISSN 0096-3003
ELSEVIER
APPLIED
MATHEMATICS
AND
COMPUTATION

Applied Mathematics and Computation 237 (2014) 430–437

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Orientation-based discrete Hough transform for line detection with low computational complexity

Kuo-Liang Chung [a,1], Yong-Huai Huang [b,*,2], Shiang-Ren Tsai [a]

[a] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei 10672, Taiwan, ROC*

[b] *Institute of Computer and Communication Engineering and Department of Electronic Engineering, Jinwen University of Science and Technology, No. 99, AnChung Rd., Xindian Dist., New Taipei City 23154, Taiwan, ROC*

ARTICLE INFO

Keywords:
Accuracy
Computational complexity
Hough transform
Line detection
Parameter space-reduction
Voting space-reduction

ABSTRACT

Using discrete representation of line segments, recently Lee and Park presented an efficient discrete Hough transform (DHT) to improve the robustness of the standard HT (SHT). However, the DHT has much higher computational complexity than the SHT. In this paper, we present an orientation-based DHT (ODHT) which consists of two strategies, the parameter space-selection strategy and the voting space-reduction strategy, to substantially reduce the computational complexity of the DHT. Besides its low computational merit, the proposed ODHT can also improve the detection accuracy of the DHT. Experimental results demonstrated that the proposed ODHT leads to 79.26% average execution-time improvement ratio and better detection accuracy when compared with the state-of-the-art DHT by Lee and Park.

1. Introduction

Detecting lines from a digital image is very important in pattern recognition and computer vision [1–3]. The detected lines are very useful in many applications, such as document analysis [4], content-based image and video retrieval [5], autonomous vehicle navigation [6], and so on. Most of the existing line detection methods are based on the standard Hough transform (SHT) technique [7] which is first invented Duda and Hart. In the SHT, each edge point in the (x, y)-space is mapped into the (ρ, θ)-space where ρ denotes the normal distance and θ denotes the normal angle. Usually, the mapped parameter space is realized by a 2-D accumulator array. Because all edge points are considered in the voting process on the whole accumulator array, the voting process is time-consuming and the memory requirement for realizing the accumulator array is huge. Therefore, many improved HT-based methods [8–16] and randomized HT-based methods [17–23] have been developed.

Besides considering the execution-time and memory costs, the representation of the continuous (ρ, θ)-space often affects the detection accuracy. Recently, using a new discrete representation of line segments, Lee and Park [24] presented an efficient discrete Hough transform (DHT) to improve the detection accuracy of the SHT. Experimental results showed that their proposed DHT is more suitable for detecting isolated lines than the SHT. However, the DHT takes much more execution-time

* Corresponding author.
E-mail addresses: klchung01@gmail.com (K.-L. Chung), yonghuai@ms28.hinet.net (Y.-H. Huang).

[1] Supported by the National Science Council of R.O.C. under the contracts NSC101-2221-E-011-139-MY3 and NSC 102-2221-E-011-055-MY3.
[2] Supported by the National Science Council of R.O.C. under the contract NSC 102-2221-E228-006.

http://dx.doi.org/10.1016/j.amc.2014.03.128

than the SHT due to six parameter spaces used in the voting process. Besides that, we also observe that the accuracy of the DHT may be degraded when many edge points are clustered in a small region, e.g. caption in the image.

In this paper, we propose an orientation-based DHT (ODHT) to substantially reduce the computational complexity of the DHT. In the ODHT, for each edge point, we propose a space-selection strategy to filter out five inappropriate parameter spaces among the six possible spaces in constant time, while retaining the appropriate parameter space instead of considering all six parameter spaces as in the previous DHT. Further, a voting space-reduction strategy is proposed to reduce the voting space in the selected parameter space. The proposed parameter space-selection and voting space-reduction strategies lead to significant computation-saving and accuracy-improvement merits for line detection when compared with the previous DHT. In addition, the computational complexity analyses for the previous DHT and our proposed ODHT are provided to show the low computational cost merit of our proposed ODHT. Experimental results confirm the execution time-saving and high accuracy advantages of our proposed ODHT.

The rest of this paper is organized as follows. In Section 2, the previous DHT by Lee and Park is introduced. In Section 3, our proposed ODHT is presented. Experimental results are demonstrated in Section 4. Some concluding remarks are addressed in Section 5.

2. The current work by Lee and Park: the DHT

Considering a line segment ℓ in the edge map of the input image, first the segment is extended to intersect two boundaries of the edge map and the two intersection points, e_1 and e_2, are used as the two parameters of ℓ. Since e_1 and e_2 may be lying on the top and bottom boundaries, the left and right boundaries, the top and left boundaries, the top and right boundaries, the bottom and left boundaries, and the bottom and right boundaries, the DHT constructs, respectively, six parameter spaces, namely, the (y_l, y_r)-space, (x_t, x_b)-space, (x_t, y_l)-space, (x_t, y_r)-space, (x_b, y_l)-space, and (x_b, y_r)-space, for line detection.

Suppose the edge map, obtained by the Canny edge detection operator [25], is of size $W \times H$. Fig. 1 depicts the mapping for each edge point from the (x, y)-space, $0 \leqslant x < W$ and $0 \leqslant y < H$, to six parameter spaces. Let p_1 and p_2 denote the two points on the line segment and each edge point can be transformed to a line segment or a curve in the parameter space. As shown in Fig. 1(a), substituting the coordinates of p_1 and p_2 into Eq. (1), we can map the two edge points p_1 and p_2 in the (x, y)-space to the two line segments s_1 and s_2, respectively, in the (y_l, y_r)-space for $0 \leqslant y_l, y_r < H$. It is shown that the two mapped line segments s_1 and s_2 are intersected in the (y_l, y_r)-space because p_1 and p_2 are collinear. According to Eqs. (2)–(6), the above mapping process from the (x, y)-space to the other five parameter spaces is shown in Fig. 1(b)–(f), respectively, and it is clear that in the five parameter spaces, s_1 and s_2 have no intersection point. Note that in the (x_t, y_r)-space, (x_b, y_l)-space, and (x_b, y_r)-space, only one of s_1 and s_2 appears in the parameter space.

$$y_r = \frac{y - y_l}{x}(W - 1) + y_l \tag{1}$$

$$x_b = \frac{x - x_t}{y}(H - 1) + x_t \tag{2}$$

$$y_l = -\frac{y}{x - x_t}x_t \tag{3}$$

$$y_r = \frac{y}{x - x_t}(W - 1 - x_t) \tag{4}$$

$$y_l = H - 1 - \frac{y - H + 1}{x - x_b}x_b \tag{5}$$

$$y_r = \frac{y - H + 1}{x - x_b}(W - 1 - x_b) + H - 1 \tag{6}$$

In the DHT, one 2-D accumulator array is used to realize each parameter space and each cell in the array is initialized to 0. In the voting process, each edge point is mapped to a line segment or a curve in each array of the parameter space and simultaneously the value of each related cell is increased by 1. After voting for all edge points, say M edge points, and performing the Gaussian smoothing in each parameter space, each detected line is reported when the number of votes in the cell is larger than the specified threshold.

Due to considering six parameter spaces, the voting process in the DHT is time-consuming. Besides that, it may detect false lines when many edge points are clustered in a small region. For example, as shown in Fig. 2(a), the caption marked with a blue ellipse produces a set of clustered edge points in the edge map (see Fig. 2(b)). The false detected lines by the DHT are shown at the bottom of Fig. 2(c) due to the affect of these clustered edge points.

In the next section, based on the orientation of each edge point, we want to propose an $O(1)$-time, i.e. constant time, space-selection strategy to select the appropriate parameter space from the six parameter spaces considered in the previous DHT. Instead of using the traditional Hough space in the DHT, we propose a reduced Hough space strategy to accelerate the voting process. Based on the proposed two strategies, each detected line is composed of edge points with similar orientations, and thus the shortcoming of detecting false lines caused by the clustered edge points can be avoided. Besides the computation-saving merit, the detection accuracy can be improved by our proposed line detection method.

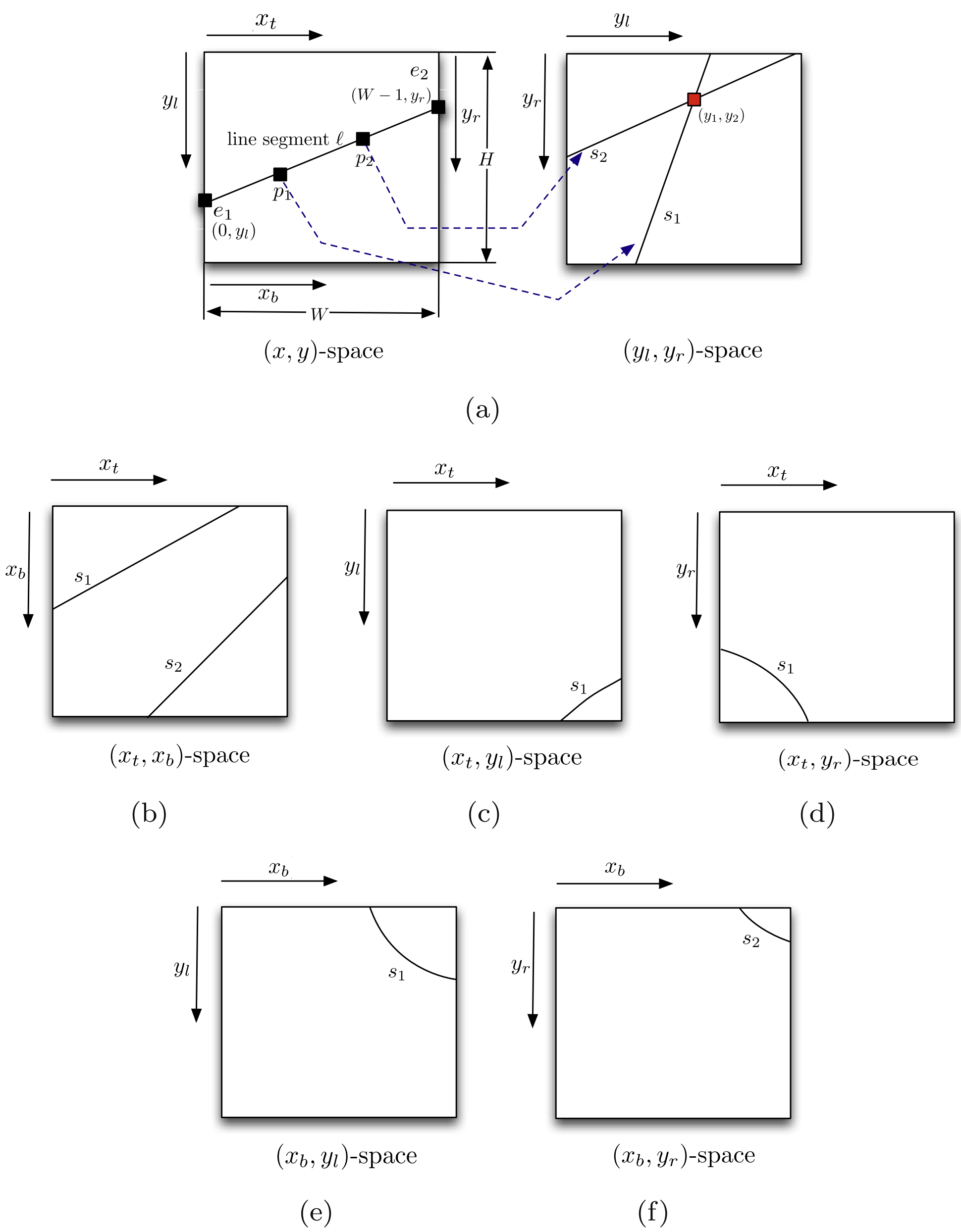

Fig. 1. The mapping from (x, y)-space to six parameter spaces. (a) (y_l, y_r)-space. (b) (x_t, x_b)-space. (c) (x_t, y_l)-space. (d) (x_t, y_r)-space. (e) (x_b, y_l)-space. (f) (x_b, y_r)-space.

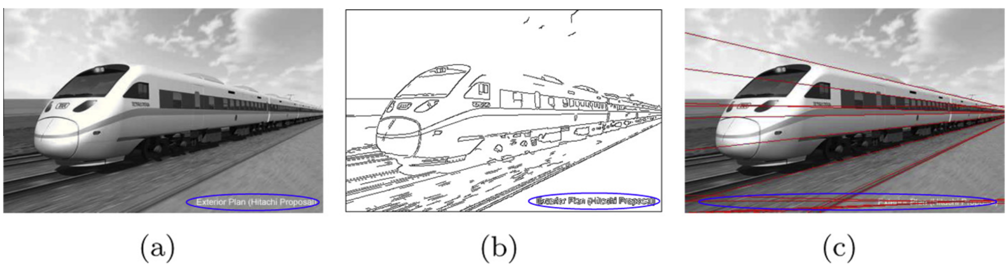

Fig. 2. False lines detected by the DHT. (a) Train image. (b) Edge map of the Train image. (c) False lines detected by the DHT.

3. The proposed orientation-based DHT: the ODHT

Before presenting the proposed orientation-based DHT, the ODHT, we first describe how to calculate the orientation of each edge point. For the edge point $p = (x, y)$ with gray value $f(x, y)$, the orientation of p can be computed by

$$\theta_{x,y} = \tan^{-1} \frac{\nabla_y f(x,y)}{\nabla_x f(x,y)} \tag{7}$$

where $0 \leqslant \theta_{x,y} < \pi$; $\nabla_y f(x,y)$ and $\nabla_x f(x,y)$ denote the gradients in the x-axis and y-axis, respectively, and their values have been provided by the edge detector in advance. Given one edge point p and the calculated orientation $\theta_{x,y}$, we now describe the proposed parameter space-selection strategy to select the appropriate parameter space from the six parameter spaces. The proposed parameter space-selection strategy can be done in constant time.

Return to Fig. 1. We substitute $e_1 = (0, y_l)$ and $e_2 = (W-1, y_r)$ into the normal distance equation $\rho = x\cos\theta_{x,y} + y\sin\theta_{x,y}$ to derive $\rho = y_l \sin\theta_{x,y}$ and $\rho = (W-1)\cos\theta_{x,y} + y_r \sin\theta_{x,y}$, respectively. It yields the same normal distance with three different forms:

$$\rho = x\cos\theta_{x,y} + y\sin\theta_{x,y} = y_l \sin\theta_{x,y} = (W-1)\cos\theta_{x,y} + y_r \sin\theta_{x,y} \tag{8}$$

Dividing each right-hand side of Eq. (8) by $\sin\theta_{x,y}$, it yields

$$y_l = y + x\cot\theta_{x,y} \tag{9}$$

$$y_r = y + (x - W + 1)\cot\theta_{x,y} \tag{10}$$

Putting the edge point $p = (x, y)$ into Eqs. (9) and (10), the rule of the proposed parameter space-selection strategy is listed in Eq. (11). The rule can determine the appropriate parameter space for the edge point p, say the (α, β)-space, in constant time among the six parameter spaces.

$$(\alpha,\beta)\text{-}space = \begin{cases} (y_l, y_r)\text{-space}, & \text{if } 0 \leqslant y_l < H \text{ and } 0 \leqslant y_r < H \\ (x_t, y_l)\text{-space}, & \text{if } 0 \leqslant y_l < H \text{ and } y_r < 0 \\ (x_b, y_l)\text{-space}, & \text{if } 0 \leqslant y_l < H \text{ and } y_r \geqslant H \\ (x_t, y_r)\text{-space}, & \text{if } y_l < 0 \text{ and } 0 \leqslant y_r < H \\ (x_b, y_r)\text{-space}, & \text{if } y_l \geqslant H \text{ and } 0 \leqslant y_r < H \\ (x_t, y_b)\text{-space}, & \text{otherwise} \end{cases} \tag{11}$$

For easy exposition, we assume the selected (α, β)-space to be the (y_l, y_r)-space and present a faster voting process which works on the reduced (y_l, y_r)-space instead of the whole (y_l, y_r)-space. Putting the angle interval $[\theta_{x,y} - \Delta, \theta_{x,y} + \Delta]$ into Eq. (9) yields

$$y_{l,1} = \begin{cases} 0, & \text{if } y_a < 0 \\ y_a, & \text{otherwise} \end{cases} \tag{12}$$

and

$$y_{l,2} = \begin{cases} H-1, & \text{if } y_b >= H \\ y_b, & \text{otherwise} \end{cases} \tag{13}$$

where

$$y_a = y + x \times \min(\cot(\theta_{x,y} - \Delta), \cot(\theta_{x,y} + \Delta)), \tag{14}$$

$$y_b = y + x \times \max(\cot(\theta_{x,y} - \Delta), \cot(\theta_{x,y} + \Delta)) \tag{15}$$

and $\Delta = 20$ empirically. According to Eqs. (12)–(15), the voting process can be confined to the reduced (y_l, y_r)-space for $y_{l,1} \leqslant y_l \leqslant y_{l,2}$ and $0 \leqslant y_r < H$ without degrading the line detection accuracy. By the same argument, the above voting space reduction strategy can be applied to the other selected parameter spaces.

In the next section, the computational complexity analysis is first provided to show that the proposed parameter space selection strategy and the voting space reduction strategy have much less computational complexity when compared with the previous DHT. Finally, some related experiments are carried out to demonstrate the computation and accuracy merits of the proposed method for line detection.

4. Computational complexity analysis and experimental results

In this section, we first compare the required computational complexity between the previous DHT and the proposed ODHT. Then, based on six test images, some experimental results are demonstrated to confirm the computational superiority of the proposed ODHT.

4.1. Computational complexity analysis

Without loss of generality, suppose that the input image is of size $N \times N$ and the obtained edge map by using the Canny edge detector has M edge points. The previous DHT performs the voting process for each edge point on six parameter spaces,

(y_l, y_r)-space, (x_t, x_b)-space, (x_t, y_l)-space, (x_t, y_r)-space, (x_b, y_l)-space, and (x_b, y_r)-space, and each space is realized by the corresponding quantized parameter space with $N_q \times N_q$ cells for $N_q \leqslant N$. Here, we set $N_q = N$ for easy explanation. In the voting process, Eqs. (1)–(6) are called at most N times to transform each edge point to a line segment or a curve in one of the six parameter spaces. Consequently, in the worst case, the voting process in the previous DHT takes $O(6NM)$ time for all M edge points.

Instead of considering all six parameter spaces in the previous DHT, our proposed ODHT only works on the selected parameter space with $N \times N$ cells, say the (y_l, y_r)-space, for the edge point at location (x, y) with orientation $\theta_{x,y}$. We further use Eqs. (12)–(15) to reduce the size of the selected parameter space from $N \times N$ to $N' \times N$ where $N' = y_{l,2} - y_{l,1} + 1$ and $N' \leqslant N$. Thus, the proposed ODHT performs Eq. (1) at most N' times instead of N times in the previous DHT to accumulate the number of votes for the mapped line segment in the (y_l, y_r)-space. Thus, the expected value of N' can be measured by

$$E[N'] = \frac{1}{180 \times N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \sum_{\theta_{x,y}=-90}^{89} (y_{l,2} - y_{l,1} + 1). \tag{16}$$

As shown in Fig. 3, after calculating $E[N']$ for $1 \leqslant N \leqslant 200$, we have $E[N'] \approx 0.46N$. Since the voting processes are similar for all six spaces, it yields that the proposed ODHT takes $O(0.46NM)$ time to perform the voting process for all M edge points. The above complexity analysis reveals the computation-saving advantage of the proposed ODHT and we have the following proposition.

Proposition 1. *The theoretical time complexity improvement ratio of the proposed ODHT over the previous DHT is* 0.92 $(= \frac{6MN - 0.46MN}{6MN})$.

As shown in the next subsection, the experimental results show that the ratio of the time spent in the voting process over the total time spent in the previous DHT is 90% on average. A more accurate estimated time complexity improvement ratio of the proposed ODHT over the previous DHT is estimated by $92\% \times 90\% = 82.8\%$. We have the following result.

Proposition 2. *The more precise estimated time improvement ratio of the proposed ODHT over the previous DHT is* 82.8%.

In next subsection, experimental results will show that the practical average execution-time improvement ratio is very close to the estimated time improved ratio mentioned in Proposition 2.

4.2. Experimental results

As shown in Fig. 4, six test images, the 251×251 Window image, the 256×256 Road image, the 400×224 Bridge image, the 490×603 Tower image, the 482×307 Train image, and the 481×322 Card image are used to compare the time performance between the previous DHT and the proposed ODHT. By using the Canny edge detector, the six edge maps of Fig. 4 are shown in Fig. 5. For fairness, the execution-time requirement includes the extra time requirement for computing the orientations of all edge points (see Eq. (7)). All the related experiments are implemented on an Intel Core i7 3615QM with CPU 2.3 GHz and 16 GB RAM. The operating system used is Mac OS X Mavericks and the C++ programing language with OpenCV library is used to implement the concerned two line detection methods.

After running the two methods on the six test images, the resultant detected lines by the previous DHT and the proposed ODHT are shown in Figs. 6 and 7, respectively. For the first four test images, the two methods have the same accuracy;

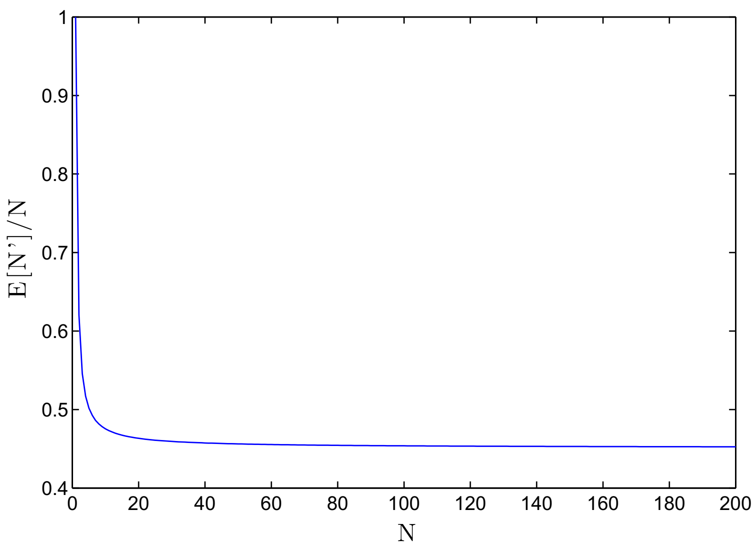

Fig. 3. Ratios of $E[N']$ over N for different values of N.

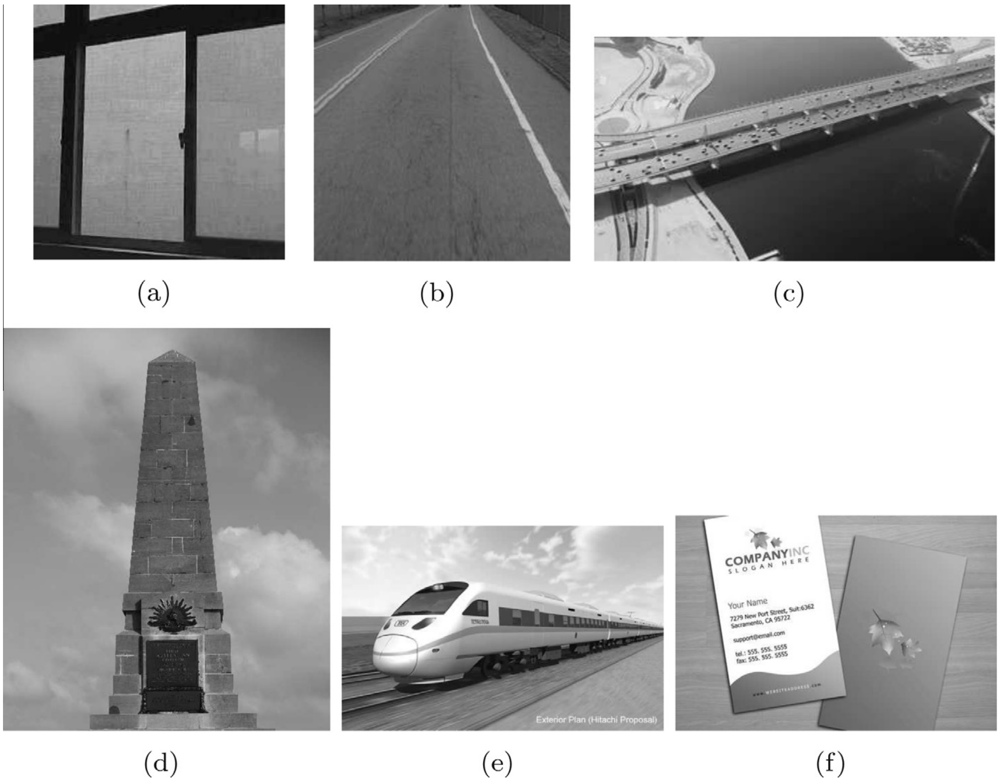

Fig. 4. Six test images. (a) Window image. (b) Road image. (c) Bridge image. (d) Tower image. (e) Train image. (f) Card image.

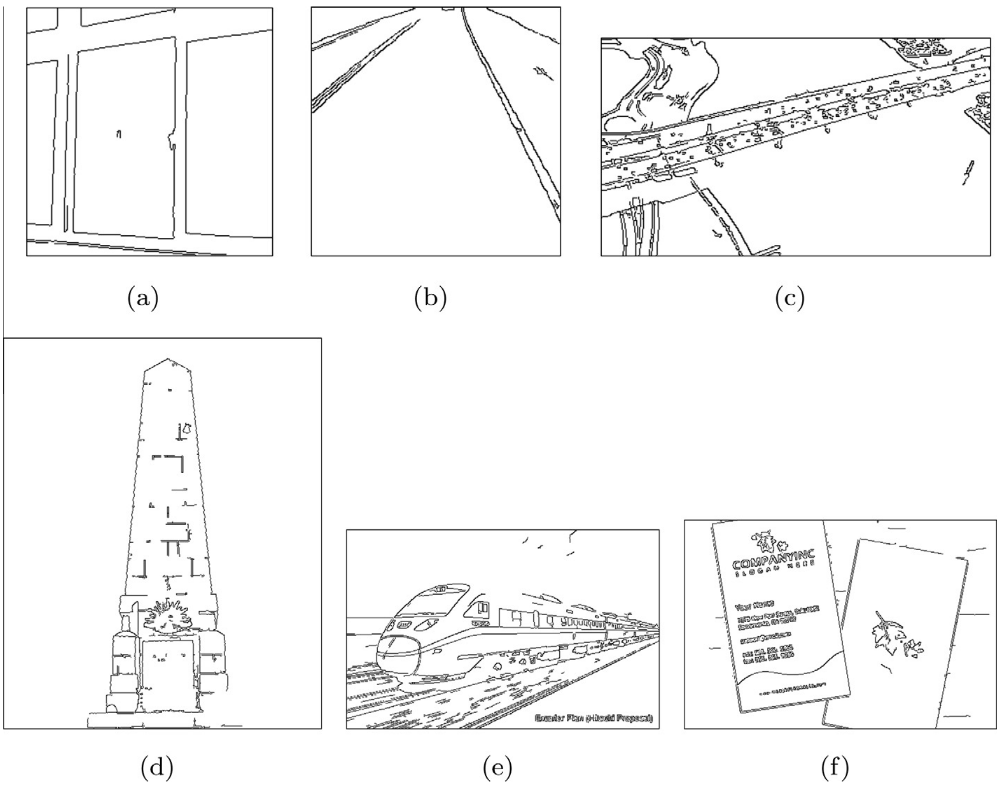

Fig. 5. Edge maps of six test images. (a) Window image. (b) Road image. (c) Bridge image. (d) Tower image. (e) Train image. (f) Card image.

however, for the last two test images, the Train image and the Card image, the previous DHT detects some false lines marked by the blue ellipses due to some clustered edge points. Because of taking the edge orientation information into account, the proposed ODHT has better accuracy for the last two images. The average execution-time improvement ratios of the proposed

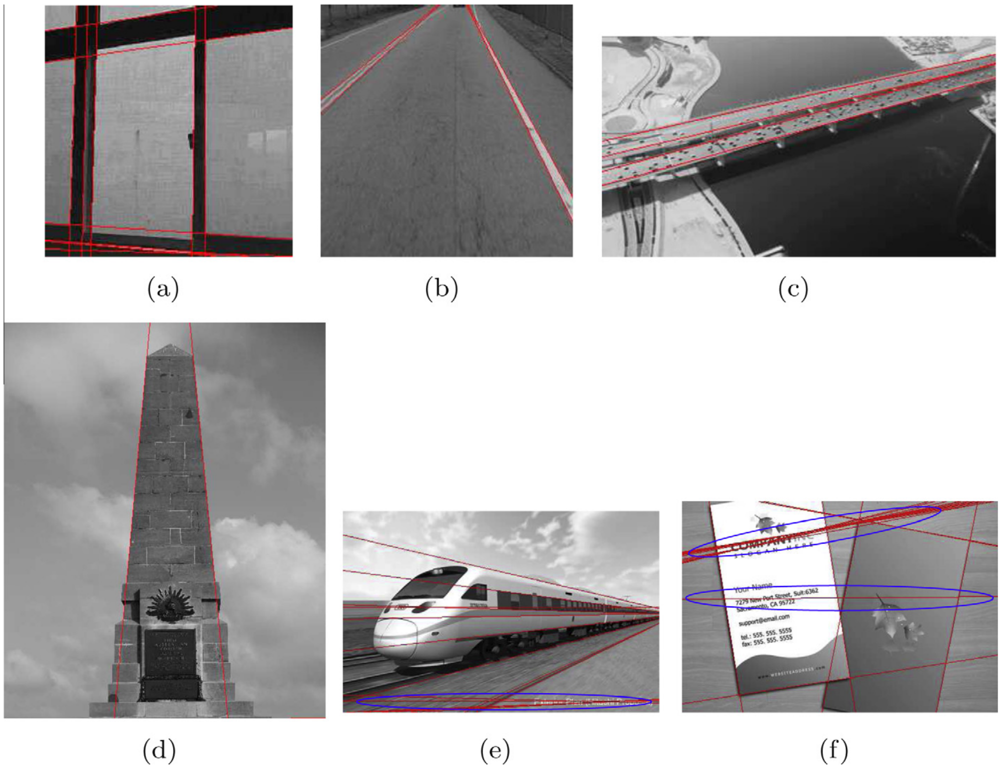

Fig. 6. Detected lines by the previous DHT. (a) Window image. (b) Road image. (c) Bridge image. (d) Tower image. (e) Train image. (f) Card image.

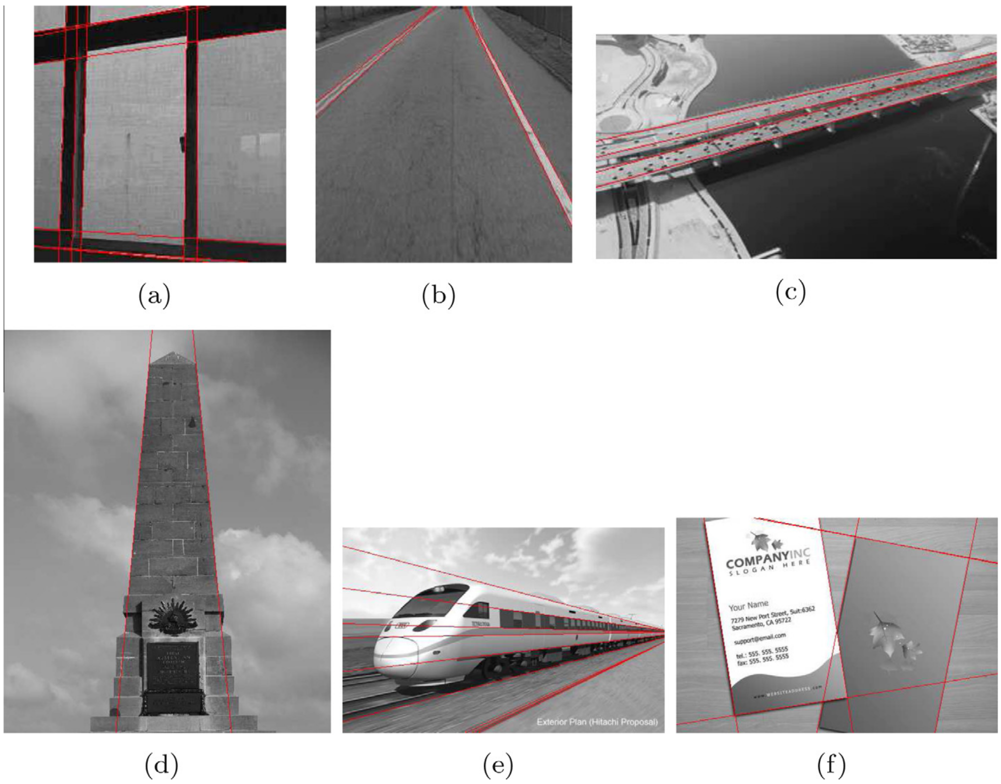

Fig. 7. Detected lines by the proposed ODHT. (a) Window image. (b) Road image. (c) Bridge image. (d) Tower image. (e) Train image. (f) Card image.

ODHT over the previous DHT are listed in Table 1 where the notation 'ms' denotes milliseconds. The table indicates that the average execution-time improvement ratio of the proposed ODHT over the previous DHT is 79.26% and is very close to the estimated time improved ratio 82.8% as mentioned in Proposition 2. The experimental results therefore justify the detection accuracy and execution-time advantages of our proposed ODHT.

Table 1
Execution-time performance comparison between the previous DHT and the proposed ODHT.

Image	Window	Road	Bridge	Tower	Train	Card
DHT	26	20	117	140	240	145
ODHT	7	6	20	27	35	24
Improvement ratio (DHT-ODHT)/DHT	73.08%	70.00%	82.91%	80.71%	85.41%	83.44%
Average	79.26%					

5. Conclusion

For line detection, this paper has presented the proposed orientation-based DHT, the ODHT, which consists of the parameter space-selection strategy and the voting space reduction strategy, to reduce the computational cost and increase the detection accuracy performance of the state-of-the-art DHT. According to the orientation information of each edge point, the proposed parameter space-selection strategy takes only constant time to select the appropriate parameter space from the six parameter spaces. Using the orientation information also results in better detection accuracy performance. Further, the proposed voting space-reduction strategy is adopted to narrow down the selected parameter space to speed up the voting process in a banded parameter space. We also provide the theoretical and estimated computational analyses for the two concerned methods to show the low computational cost merit of the proposed ODHT. The experimental results demonstrated that with better detection accuracy, the proposed ODHT can achieve 79.26% average execution-time improvement ratio, which is very close to the estimated execution-time improvement ratio, when compared with the DHT.

Acknowledgments

The authors appreciate the proofreading help of Ms. Catherine Harrington.

References

[1] D.H. Ballard, C.M. Brown, Computer Vision, Prentice–Hall, Englewood Cliffs, New Jersey, 1982.
[2] E.R. Davies, Machine Vision: Theory, Algorithms, Practicalities, Academic Press, London, 1990.
[3] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison Wesley, New York, 1992.
[4] H.F. Jiang, C.C. Han, K.C. Fan, A fast approach to the detection and correction of skew documents, Pattern Recognit. Lett. 18 (7) (1997) 675–686.
[5] C.W. Su, H.Y.M. Liao, H.R. Tyan, C.W. Lin, D.Y. Chen, K.C. Fan, Motion flow-based video retrieval, IEEE Trans. Multimedia 9 (6) (2007) 1193–1201.
[6] L.W. Tsai, J.W. Hsieh, K.C. Fan, Vehicle detection using normalized color and edge map, IEEE Trans. Image Process. 16 (3) (2007) 850–864.
[7] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1972) 11–15.
[8] J. Illingworth, J. Kittler, The adaptive Hough transform, IEEE Trans. Pattern Anal. Mach. Intell. 9 (5) (1987) 690–698.
[9] D. Ben-Tzvi, M.B. Sandler, A combinatorial Hough transform, Pattern Recognit. Lett. 11 (3) (1990) 167–174.
[10] H.K. Aghajan, T. Kailath, Slide: subspace-based line detection, IEEE Trans. Pattern Anal. Mach. Intell. 16 (11) (1994) 1057–1073.
[11] S. Climer, S.K. Bhatia, Local lines: a linear time line detector, Pattern Recognit. Lett. 24 (14) (2003) 2291–2300.
[12] K.L. Chung, T.C. Chen, W.M. Yan, New memory-and computation-efficient Hough transform for detecting lines, Pattern Recognit. 37 (5) (2004) 953–963.
[13] J. Cha, R.H. Cofer, S.P. Kozaitis, Extended Hough transform for linear feature detection, Pattern Recognit. 39 (6) (2006) 1034–1043.
[14] K.L. Chung, T.C. Chang, Y.H. Huang, Comment on: extended Hough transform for linear feature detection, Pattern Recognit. 42 (7) (2009) 1612–1614.
[15] S. Du, B.J. van Wyk, C. Tu, X. Zhang, An improved Hough transform neighborhood map for straight line segments, IEEE Trans. Image Process. 19 (3) (2010) 573–585.
[16] S. Du, B.J. van Wyk, C. Tu, X. Zhang, Collinear segment detection using HT neighborhoods, IEEE Trans. Image Process. 20 (12) (2011) 3612–3620.
[17] L. Xu, E. Oja, P. Kultanan, A new curve detection method: randomized Hough transforms, Pattern Recognit. Lett. 11 (5) (1990) 331–338.
[18] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognit. 24 (4) (1991) 303–316.
[19] O. Chutatape, L. Guo, A modified Hough transform for line detection and its performance, Pattern Recognit. 32 (2) (1999) 181–192.
[20] T.C. Chen, K.L. Chung, A new randomized algorithm for detecting lines, Real-Time Imaging 7 (6) (2001) 473–482.
[21] K.L. Chung, Y.H. Huang, Speed up the computation of randomized algorithms for detecting lines, circles, and ellipses using novel tuning- and LUT-based voting platform, Appl. Math. Comput. 190 (1) (2007) 132–149.
[22] K.L. Chung, Y.H. Huang, A pruning-and-voting strategy to speed up the detection for lines, circles, and ellipses, J. Inf. Sci. Eng. 24 (2) (2008) 503–520.
[23] K.L. Chung, Z.W. Lin, S.T. Huang, Y.H. Huang, H.Y.M. Liao, New orientation-based elimination approach for accurate line-detection, Pattern Recognit. Lett. 31 (1) (2010) 11–19.
[24] D. Lee, Y. Park, Discrete Hough transform using line segment representation for line detection, Opt. Eng. 50 (8) (2011) 087004-1–087004-4.
[25] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.