HOUGH TRANSFORM WITH DYNAMIC THRESHOLDING FOR ROBUST AND REAL-TIME DETECTION OF COMPLEX CURVES IN IMAGES

Shen-En Shih and Wen-Hsiang Tsai

Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

ABSTRACT

A dynamic thresholding method is proposed for use in the Hough transform to detect complex curves in images robustly. While determining edge pixels contributing to the peak in the Hough space for detecting a curve with noise and other errors, the proposed method can endure the errors by detecting pixels coming from an equal-width shape which is centered at the curve with a small width everywhere along the curve. This equal-width shape detection capability is accomplished by the use of a dynamic threshold for pixel selection, which is derived from the use of the first-order directional derivative of the function describing the curve. Three conventional methods are compared to show the superiority in robustness of the proposed method via experimental results, and a real-time application of the method for quick detection of lines in omni-images is also demonstrated.

Index Terms— Image signal processing, Hough transform, shape detection, real-time application.

1. INTRODUCTION

The Hough transform is widely used in computer vision applications to detect shapes in noisy images. It includes three main steps: image thinning, cell value accumulation, and voting for peak value detection. In the image thinning step, a standard method is to conduct edge detection to extract edge pixels in the input image. In the cell value accumulation step, each edge pixel is transformed into a curve in the parameter space (also called *Hough space*), and the values of the corresponding elements in the Hough space, called *Hough cells*, are all incremented by one. Subsequently, the voting step is conducted to find the peaks (local maximums) in the Hough space, which are taken finally as the parameters of the detected shapes.

The cell value accumulation step is essential in the Hough transform, also known as the *evidence gathering* step. Two different ways are used in this step [1, 2]. The first is to find all the edge pixels corresponding to each Hough cell, which is called the *forward-mapping* method in this study since it directly describes the shape desired to be found. The second way is to find all the Hough cells corresponding to each edge pixel, which is called the *backward-mapping* method in this study since it finds the parameters of the shape in a backward manner from image pixels' coordinates. If the Hough space and the image space are both continuous, these two methods yield the same results. However, since the image space is discretized and represented by pixels, and since the Hough space is discretized and represented by an accumulation array, the two methods become different from each other in detail [2].

Although the forward-mapping method is more straightforward, the backward-mapping one is used more often because it can be accelerated by many ways, e.g., by the use of an inverse function [3,4], a gradient method [5, 6, 7], a one-to-one mapping [8, 9], certain special geometric relations [10, 11], etc. However, since a uniform quantization of the Hough space results in a non-uniform precision of the computed curve in the image space [1], the backward-mapping method is found to yield some undesired effects, especially when used to detect complicated shapes (as shown in Sec. 3.1); and so, it is mostly used to detect simple shapes like lines [3], circles [5, 6, 7, 11], ellipses [4, 8, 10], etc. This problem becomes more serious when wide-angle cameras are used for image taking. For example, omni-images captured by catadioptric omni-cameras are distorted largely, and straight space lines projected on omni-images become quadratic curves. Therefore, to robustly detect such complicated shapes, a forward-mapping method is needed to decide pixels corresponding to each Hough cell.

In the forward-mapping method, if a shape to be detected is described by a function F, then pixels contributing to the accumulation of the (largest) peak cell value in the Hough space theoretically are just those with their coordinates (u, v) satisfying the equation $F(u, v) = 0$. However, since these coordinates (u, v) in practice are with errors coming from quantization, noise, edge detection, imprecise camera calibration, etc. [2][11], the mentioned pixels, with such erroneous coordinates (u, v), instead will *not* all lie precisely on the curve $F(u, v) = 0$. To endure such imprecision, it is desired that pixels contributing to the peak cell value come from an *equal-width* shape both centered at the curve $F(u, v) = 0$ and with a small width W *everywhere* along the curve, instead of coming just from the *thin* curve itself. A simple method to achieve this goal is to define a small *tolerant threshold* value T, so that the aforementioned erroneous coordinates satisfy the inequality

$$\left|F(u,v)\right| \le T \,. \tag{1}$$

Here, the threshold T should not be fixed because when dealing with a complex curve like a quadratic one, a fixed T might even not exist for use to generate an equal-width curve according to our experimental observations (discussed in Sec. 2.2). Instead, T should be defined dynamically for distinct shapes and pixel locations, as done in this study.

In more detail, a *dynamic thresholding* method is proposed to decide the threshold value T dynamically in accordance with the image pixels' positions and the Hough cells' values. Compared with conventional methods, the proposed method has at least the following advantages: (1) being able to detect equal-width curves to absorb aforementioned imprecise coordinates; (2) being capable of detecting complex curves; (3) having no need of parameter tuning; and (4) being good for fast computations.

Subsequently, in Sec. 2 the proposed method is derived, and three conventional methods are described and compared. Experimental results showing the merits of the proposed method and a real-time application of the method for detecting space lines in omni-images are given in Sec. 3, followed by conclusions in Sec. 4.

2. IDEA OF THE PROPOSED METHOD

In the cell value accumulation step where pixels contributing to Hough cell values are determined, it is desired, as mentioned previously, to develop a method for detecting pixels of an equal-width shape both centered at the *thin* curve $F(u, v) = 0$ and with an equal width W everywhere on the curve. The proposed method for this goal is derived in Sec. 2.1, followed by comparisons with conventional methods in Sec. 2.2.

2.1. Derivation of proposed method

Given a pixel P with coordinates (u, v), two cases can be identified. One is that $F(u, v) < 0$, where the coordinates (u', v') of the closest pixel P' on the curve $F = 0$, as depicted in Fig. 1, can be estimated by the use of the direction of the gradient vector at (u, v) to be

$$(u', v') = (u, v) + d \cdot \frac{\nabla F(u,v)}{\left\|\nabla F(u,v)\right\|}, \tag{2}$$

where d is an unknown distance. Then, the function value $F(u', v')$ at (u', v') can be linearly estimated by the use of the first-order directional derivative to be

$$F(u',v') = F(u,v) + \nabla F(u,v) \cdot \left(d \cdot \frac{\nabla F(u,v)}{\left\|\nabla F(u,v)\right\|} \right). \tag{3}$$

Also, because pixel P' with coordinates (u', v') is on the curve, we have $F(u', v') = 0$, so that (3) implies that

$$F(u,v) + \nabla F(u,v) \cdot \left(d \frac{\nabla F(u,v)}{\left\|\nabla F(u,v)\right\|} \right) = 0,$$

or equivalently, that

$$d = -F(u,v) \cdot \left(\frac{\left\|\nabla F(u,v)\right\|}{\nabla F(u,v) \cdot \nabla F(u,v)} \right) = \frac{-F(u,v)}{\left\|\nabla F(u,v)\right\|}. \tag{4}$$

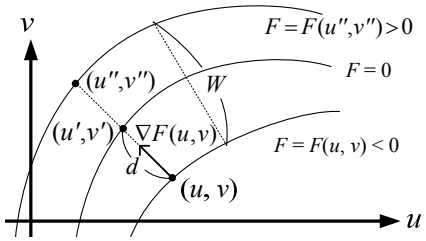


Fig. 1. Illustration of proposed dynamic thresholding method.

Then, as illustrated in Fig. 1, the inequality $d < W/2$ can be used to determine whether a pixel with coordinates (u, v) is within the equal-width curve shape. This inequality, when combined with (4), leads to

$$d = \frac{-F(u,v)}{\left\|\nabla F(u,v)\right\|} < \frac{W}{2},$$

or equivalently, to

$$-F(u,v) < \left\|\nabla F(u,v)\right\| \cdot \frac{W}{2}. \tag{5}$$

In the other case that $F(u, v) > 0$, similarly, the coordinates (u', v') of the closest point P' on the curve $F = 0$ can be linearly estimated by the *negated* gradient vector $-\nabla F(u, v)$, so that coordinates (u', v') can be expressed as

$$(u',v') = (u,v) - d \frac{\nabla F(u,v)}{\left\|\nabla F(u,v)\right\|}.$$

Then, following similar derivations, we can get

$$F(u,v) < \left\|\nabla F(u,v)\right\| \cdot \frac{W}{2}. \tag{6}$$

The two inequalities of (5) and (6) can be combined to get

$$\left|F(u,v)\right| < \left\|\nabla F(u,v)\right\| \cdot \frac{W}{2}, \tag{7}$$

which is of the form of the inequality of (1) used in the *constant thresholding* method. But differently, the threshold T can now be taken to be $(\|\nabla F(u, v)\| \cdot W)/2$ whose value can be dynamically determined for pixels with different coordinates (u, v) as well as for Hough cells with different parameters related to the function F, in order to detect a desired equal-width curve shape in the image space.

Theoretically, the dynamic thresholding method proposed above is based on linear approximation. Accordingly, the estimated function value $F(u', v')$ will become inaccurate when the desired curve width W becomes large. However, since the curve width is used to overcome small errors in the input data, the width W may be taken to be a small number. So, the proposed method is expected to yield good results in most applications.

2.2. Comparison with other conventional methods

As stated in Sec. 1, the forward- and backward-mappings are two ways to conduct the cell value accumulation task. In the following, three conventional methods are described and compared with the proposed method.

2.2.1. Forward-mapping by constant thresholding

In a forward-mapping method with the threshold T defined to be a constant value, the inequality (1) is used to decide whether an edge pixel with coordinates (u, v) contributes to the accumulation of a Hough cell value. This method can generate a desired equal-width curve as shown in Fig. 2(a) when detecting simple shapes; however, it generates bad results when dealing with complicated shapes as shown in Fig. 2(b). Furthermore, even if an equal-width curve can be generated, the relation between the threshold T and the curve width W is still unclear and it needs further analysis or experiments to derive a reasonable value for T.

2.2.2. Backward-mapping by examining each Hough cell

In a backward-mapping method which examines each Hough cell, the cells with their values contributed by an edge pixel with coordinates (u, v) can be derived by: (1) use (u, v) and function F to find a *parametric* equation describing all the curves going through (u, v); (2) regard the equation as the description of a hypersurface in the Hough space, and examine each cell to detect those intersecting this hypersurface, and increment their values. One drawback of this method is that, which pixels contribute to cell value accumulations depends on the cell *size*, and these pixels together are not of a desired equal-width shape, as shown by the example appearing in the 2nd row in Fig. 4.

2.2.3. Backward-mapping by use of an inverse function

In a backward-mapping method using an inverse function, the Hough cells with their values contributed by an edge pixel with coordinates (u, v) are determined by: (1) enumerate the first $n - 1$ parameters of the Hough space, where n is the dimension of the space; and (2) derive the nth parameter by the inverse function of F and the coordinates (u, v). This method is faster than the previous one, but has some drawbacks. First, it cannot generate desired equal-width shapes. Second, the inverse function might be difficult to derive. Furthermore, different parameterizations and different ways of parameter enumerations might yield different results, as shown in Fig. 3.

3. EXPERIMENTAL RESULTS

In this section, the validity, effectiveness, and robustness of the proposed dynamic thresholding method for the Hough transform are shown by comparing the proposed method with four other methods as listed in Table I for detecting four different types of shapes as listed in Table II. In the following, the shape of the pixels contributing to a peak value accumulated in the Hough space is compared in Sec. 3.1, some measures to test the robustness of these methods are shown in Sec. 3.2, and finally, a real-time application of the proposed method for detecting space lines in omni-images is demonstrated.

3.1. Pixels contributing to a Hough cell value

The pixels contributing to the peak in the Hough space for detecting each of the four shapes listed in Table II using each of the five methods listed in Table I are drawn in Fig. 4. Recalling that these pixels are desired to form an equal-width shape, one can see that the proposed method yields the best results for both simple and complex shapes as shown in the leftmost column in the figure, and that the conventional methods cannot generate equal-width curves, especially when detecting complicated shapes.

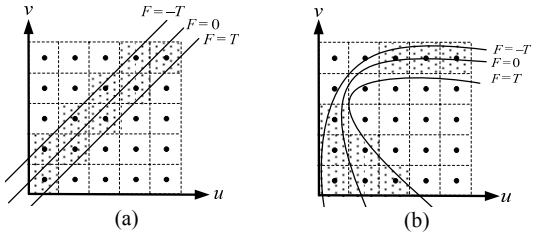


Fig. 2. Pixels (marked as dotted blocks) contributing to a Hough cell value when detecting (a) a line, and (b) a curve.

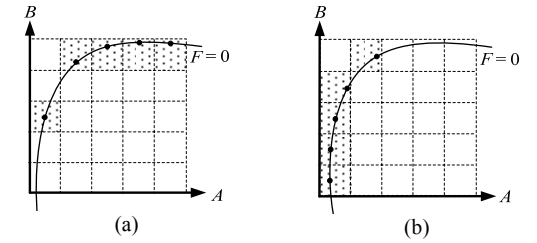


Fig. 3. Hough cells (dotted blocks) with values contributed by an edge pixel when enumerating parameter (a) A, and (b) B.

Table I. Used Hough transform algorithms.

No.	accumulation method	described in …
(H1)	the proposed method	Sec. 2.1
(H2)	constant threshold 1.0	Sec. 2.2.1
(H3)	constant threshold 100	Sec. 2.2.1
(H4)	examining each cell	Sec. 2.2.2
(H5)	inverse function	Sec. 2.2.3

Table II. Shapes used in experiments.

shape	equation	cell size
(S1) line	$v = Au + B$	A: 0.1; B: 10
(S2) circle / ellipse	$\left(\frac{u-u_0}{A}\right)^2 + \left(\frac{v-v_0}{B}\right)^2 = 1$	u_0: 10; v_0: 10; A: 10; B: 10
(S3) sine and cosine	$v = A\sin(Bu) + C\cos(Du)$	A: 10; B: 0.1; C: 10; D: 0.1
(S4) space lines in omni-images	described in (8)	A: 0.1; B: 0.1

3.2. Robustness for noise input data

To test the robustness of each different cell-value accumulation method, the ground-truth curve is first drawn on an image. Then, the pixels on the curve are perturbed within a circle with a diameter of 5 pixels to generate *curve*

pixels with small errors. Also, 1% pixels of the entire image are randomly noisified as *noise pixels*. The true positive rate (TPR) and the false positive rate (FPR) for each of the five methods are calculated accordingly respectively by:

$$TPR = \frac{\#\text{ of curve pixels contributing to the peak cell value}}{\text{total \# of the original curve pixels}};$$

$$FPR = \frac{\#\text{ of noise pixels contributing to the peak cell value}}{\text{total \# of the noise pixels}}.$$

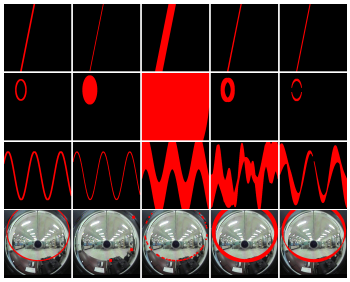


Fig. 4. Pixels (in red) contributing to peak cell value. From left to right: results of algorithms H1 to H5, and from top to bottom: results for shapes S1 to S4. The proposed method yields the best results as shown in the leftmost column.

As stated previously, it is desired that the *curve pixels* all contribute to the peak-value accumulation, so a high TPR is desired. Contrarily, *noise pixels* which make such contributions should be as few as possible, so a low FPR is desired. As can be seen in Fig. 5, the proposed method yields very high TPRs and very low FPRs for all the four types of shapes, showing its robustness; and this is not the case for each of the other four methods.

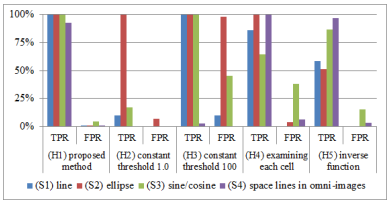


Fig. 5. TPR and FPR of five Hough transform algorithms for detecting four types of shapes. The proposed method yields high TPRs and low FPRs for all the shapes, and the others do not.

3.3. A real-time application using omni-cameras

As the technology advances, the memory is much larger and cheaper in recent days, but many applications did not make good use of them. In this paper, we propose a method to conduct the Hough transform in real-time using a large look-up table kept in memory, and apply the method to carry out a real-time application of detecting space lines in omni-images taken at a very high speed.

The look-up table is constructed as follows. For each pixel's coordinates (u, v) in the input image, the Hough cells whose values are contributed by this pixel are determined by the proposed dynamic thresholding method as stated in Sec. 2.1. Then, these Hough cells are recorded in this table with indices $[u, v]$. After all the coordinates (u, v) are processed and recorded, the table construction is finished. With the table, all the computations in the Hough transform can be removed, so that the Hough transform can be conducted by table lookup operations in a very fast speed.

The omni-camera used in the experiments is a catadioptric one with a hyperboloidal-shaped mirror. The focal length f is 522.45, the eccentricity ε was calibrated [12] to be $-0.022\sqrt{u^2+v^2}+1.9211$, and the resolution is 640×480. The curve of a space line projected on an image taken by the omni-camera can be described by [13]:

$$C_1u^2 + C_2uv + C_3v^2 + C_4u + C_5v + C_6 = 0, \qquad (8)$$

where the coefficients are

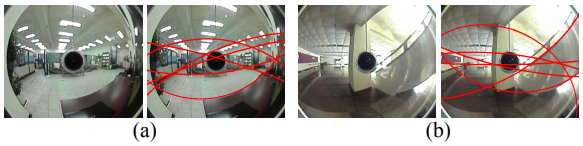
$$C_1 = A^2 - B^2\left(C_7^{\,2} - 1\right); \qquad C_2 = 2A\sqrt{1 - A^2 - B^2};$$

$$C_3 = 1 - A^2 - C_7^{\,2}B^2; \qquad C_4 = 2ABC_7f;$$

$$C_5 = 2BC_7\sqrt{1 - A^2 - B^2}\,f; \quad C_6 = B^2f^2; \text{ and } C_7 = \frac{\varepsilon^2 + 1}{\varepsilon^2 - 1}.$$

The range of the two parameters A and B are both taken to be from −1 to 1 and the Hough space dimension is chosen to be 64×64.

The experiments were conducted on a PC with an Intel i5-2400 CPU with a clock rate of 3.10GHz. The Hough transform process takes about 35~50 milliseconds (≈ 20~28 fps), depending on the number of edge pixels; and the look-up table takes about 800MB in the main memory. Two detection results are shown in Fig. 6, which show the feasibility of the proposed method for real-time uses.



(a) (b)

Fig. 6. Two real-time detection results. Left ones are input images, and the results are superimposed on the right ones.

4. CONCLUSION

A dynamic thresholding method has been proposed to accomplish the cell-value accumulation task in the Hough transform process. Differing from the conventional methods, the proposed method can detect complicated shapes more robustly, and has the merits of generating equal-width curves to endure noise and other errors; detecting complicated shapes; needing no parameter tuning; and conducting fast computations. A real-time application of the method for detecting space lines in omni-images has also been demonstrated. The proposed method can be applied as well to data acquired by 3D sensors (e.g., the Microsoft Kinect) to detect 3D analytic shapes, and its real-time uses can be extended further to on-line calibration, mobile robot navigation, vehicle driving assisting applications, etc.

5. REFERENCES

[1] J. Sklansky, "On the Hough technique for curve detection," *IEEE Transactions on Computers*, vol. C-27, no. 10, pp. 923-926, 1978.

[2] Y. C. Cheng and S. C. Lee, "A New method for quadratic curve detection using K-RANSAC with acceleration technique," *Pattern Recognition*, vol. 28, no. 5, pp. 663-682, 1995.

[3] T. Y. Lee, T. S. Chang, S. H. Lai, K. C. Liu, and H. S. Wu, "Wide-angle distortion correction by Hough transform and gradient estimation," *Proc. IEEE Conf. Visual Communications and Image Processing, 2011*, pp. 1-4, Nov. 6-9, 2011.

[4] C. F. Chien, Y. C. Cheng, and T. T. Lin, "Robust ellipse detection based on hierarchical image pyramid and Hough transform," *Journal of the Optical Society of America A*, vol. 28, no. 4, pp. 581-589, 2011.

[5] M. Wu, J. Yang, and Y. Sun, "A new method for fast circle detection in a complex background image," *Proc. SPIE 8004, MIPPR 2011*: *Pattern Recognition and Computer Vision*, 80040U, Dec. 02, 2011.

[6] S. Lei, P. Hailang, C. Guo, and L. Chengrong, "Chinese chess recognition based on log-polar transform and FFT," *Proc. Int'l Conf. Computational Science and Its Applications 2011*, pp. 50-58, Santander, Spain, June 20-23, 2011.

[7] L. Pan, W. S. Chu, J. M. Saragih, F. De la Torre, and M. Xie, "Fast and robust circular object detection with probabilistic pairwise voting," *IEEE Signal Processing Letters*, vol. 18, no. 11, pp. 639-642, Nov. 2011.

[8] T. Zhou, and N. Papanikolopoulos, "Enhancing the randomized Hough transform with k-means clustering to detect mutually-occluded Ellipses," *Proc. 19th Mediterranean Conf. Control and Automation*, pp. 327-332, June 20-23, 2011.

[9] H. Sun, Y. Mao, N. Tang, and D. Zhu, "A real-time and robust multi-circle detection method based on randomized Hough transform," *Proc. Int'l Conf. Computer Science and Information Processing, 2012,* pp. 175-180, Aug. 24-26, 2012.

[10] Y. Ito, K. Ogawa, and K. Nakano, "Fast Ellipse detection algorithm using Hough transform on the GPU," *Proc. IEEE Networking and Computing 2011*, pp. 313-319, Nov. 30, 2011.

[11] D. Loannou, W. Huda, and A. F. Laine, "Circle recognition through a 2D Hough transform and radius histogramming," *Image and Vision Computing*, vol. 17, no. 1, pp. 15-26, Jan. 1999.

[12] S. E. Shih and W. H. Tsai, "A two-omni-camera stereo vision system with an automatic adaptation capability to any system setup for 3D vision applications," *IEEE Transactions on Circuits and Systems for Video Technology*, accepted and to appear.

[13] C. J. Wu and W. H. Tsai, "An Omni-Vision Based Localization Method for Automatic Helicopter Landing Assistance on Standard Helipads," *Proc. of 2nd Int'l Conf. Computer and Automation Eng. (2nd ICCAE)*, pp. 327-332, vol. 3, Singapore.