A FAST RANDOMIZED HOUGH TRANSFORM FOR CIRCLE/CIRCULAR ARC RECOGNITION

SHIH-HSUAN CHIU

*Department of Polymer Engineering
National Taiwan University of Science and Technology
43, Keelung Road Section 4, Taipei, Taiwan
schiu@mail.ntust.edu.tw*

JIUN-JIAN LIAW*

*Department of Information and Communication Engineering
Chaoyang University of Technology
168 Jifong E. Rd., Wufong Township, Taichung County, Taiwan
jjliaw@cyut.edu.tw*

KUO-HUNG LIN

*Department of Polymer Engineering
National Taiwan University of Science and Technology
43, Keelung Road Section 4, Taipei, Taiwan
D9304301@mail.ntust.edu.tw*

The main drawbacks of the Hough transform (HT) are the heavy requirement of computation and storage. To improve the drawbacks of the HT, the randomized Hough transform (RHT) was proposed. But the RHT is not suitable for detecting the pattern with the complex image because the probability is too low. In this paper, we propose a fast randomized Hough transform for circle/circular arc detection. We pick one point at random to be the seed point. Then, we propose a checking rule to confirm if the seed point is on the true circle. Compared with the previous techniques, the proposed method requires less computational time and is more suitable for complex images. In the experiments, synthetic and real images are used to show the effect of the proposed method.

Keywords: Hough transform; randomized Hough transform; circle detection; circular arc detection; recognition.

1. Introduction

The key to many tasks in computer vision is to detect the meaningful grouping of points. Hough transform (HT) is a pattern detection technique that was first introduced by Hough.¹⁰ Then the concept of HT is widely used to detect the objects

*Author for correspondence

in the digital image.^{1,5,6,16,18} The major concept of HT is to make each point vote for all possible patterns (candidates) in the parameter space. In other words, the HT converts the image pattern recognition problem into the local peak detection problem by a voting process.¹²

Since the pattern of a circle contains three parameters (the coordinate of center (x_c, y_c) , and the radius (r_c)), detecting the circle by standard Hough transform is a third order computational complexity algorithm.¹² The standard Hough transform is restricted by computation and storage requirements to the dimension of parameter space.⁷ The high dimension leads to high computational time and storage requirements. To reduce the computational time, many Hough-based methods are proposed to solve the problem of high dimension by multistep schemes, such as:

- (1) For the line detection, Chutatape and Guo proposed a modified Hough transform.² Firstly, they selected an edge point as the seed point. Then they computed the angle of another point to the seed point. They extracted the line with the position of the seed point and the angle. However, this method cannot be applied to the circle/circular arc detection.
- (2) In the circle detection, Ioannou *et al.* proposed a method¹¹ with two steps. They computed all mid-perpendiculars of all point pairs in the image space to obtain the center positions of the circles. Then they used a radius histogramming to estimate the radius with the obtained center. This method makes two points vote for a line which is composed of many points. In the complex images, since there are too many point pairs whose two points are not in the same circle, the mid-perpendiculars of all point pairs will also be complex. We can see that it is not easy to use mid-perpendiculars in the complex images for the center position recognition.
- (3) In the circular arc detection, Pei and Horng proposed a method based on executing the HT two times.¹⁷ They first transformed the edge points into the HT space to detect the circle where the arc lies, and applied the HT again to obtain the *near-peak HT* data. They used the eigenvectors of the covariance matrix of the *near-peak HT* data to estimate the parameters of the circular arc. This scheme costs large computation time because it executes the HT two times.

The concept of probability was also suggested to modify the HT. The probabilistic class of the HT reduces their drawbacks by transforming small subsets of points to the parameters space. In this kind of method, the image points vote for the most probable candidate in the parameter space. It makes the effects of noise in the parameter space small and the accumulator structure simple.⁸ In comparison with the standard Hough transform, we can see that the subsets of image feature data and the simple accumulator are used to reduce the computation and storage requirements. Two main probabilistic Hough transforms are the probabilistic Hough transform (PHT)^{8,14} and the randomized Hough transform (RHT).^{9,13,20} The PHT is similar to the standard Hough transform but it uses just a small random subset of the data points. The RHT is a well-known method for the circle detection^{3,19} and it is

widely used in many applications. The RHT picks n points from the data points at random (where n is the number of parameters of the target pattern) and these n points are used to obtain a candidate (a possible pattern). For example, a circle contains three parameters ($n = 3$). When we used the RHT to detect the circle, we picked three points from the data points at random and checked if these three points are on a true circle. In the complex image, since there are many points that are not on the same circle, the probability of picking three points randomly from the same circle is too low. In other words, if three randomly picked points are not on the same true circle, the circle cannot be found. However, it is not easy to extract the pattern with the complex image by the RHT because the probability is too low.¹⁵

In order to detect circles effectively, Chiu and Liaw proposed an effective voting method⁴ (EVM). They sampled the original data point with an interval for reducing the number of data points. Each point pair was selected one by one from the image. When a point pair was selected, they used a checking rule to see if there is a third point whose distance to a point of the selected point pair is equal to the distance of the point pair. These three points (two points in the selected point pair and the third point) are used to vote for a circle candidate. The EVM is a multistep scheme which separates the circle detection into the third point checking and the candidate voting. However, the EVM did not fix the probability problem of the RHT.

To improve the probability problem of the RHT, the windows of the image are used to increase the performance of the RHT. Kälviäinen and Hirvonen proposed a connective randomized Hough transform¹³ (CRHT) to detect lines. They picked a size of window at random in place of the randomized picked points from the image space. And they checked if the connective points in the selected window corresponded to a line. They suggested a local search concept with a selected window, but checking the pattern with connective points was not robust when the edge of the pattern was not connective. However, this method is not extended to recognize the circles. For the circle detection, the adaptive randomized Hough transform⁹ (ARHT) is applied to increase local signal-to-noise ratio for improving the detection rate. The ARHT uses the moving window to decompose the detection task from the original image into small subimages. And the RHT is applied to each subimage to detect the circles. However, if three randomly picked points are not on the same true circle in a window, the true circle cannot be found.

In this paper, we propose a multistep scheme which is called the fast randomized Hough transform (FRHT) to recognize the circles. Since the probability of picking one point which is on a circle is higher than picking three points, we pick only one point from the data point to be the seed point. Then we apply a rule to check whether the seed point is on a true circle. Since the circular arc can be regarded as a part of the circle, the proposed method is also extended to recognize the circular arcs. In the experiments, we compare the FRHT with the previous techniques, such as the RHT, the ARHT and the EVM, and conclude that the proposed method requires less computational time.

2. The Faster Randomized Hough Transform for Circle Detection

If we detect circles by traditional randomized Hough transform (RHT), we have to pick three edge points at random and check if these three points are on a true circle. The probability of picking three points from a circle is too low. If we can pick only one point from data points instead of three points, the probability and the performance will be increased.

When an edge image is given with M edge points, we set all edge points as a set, D , of data points:

$$D = \{d_i = (x_i, y_i) \mid i = 1, 2, \dots, M\}. \quad (1)$$

We randomly pick a point from D to be the seed point, $d_s = (x_s, y_s)$. When a seed point is selected, we check if this seed point is on a true circle. We can see that if the seed point is on a circle, there are two points whose distances to d_s are the same on this circle. For example, see Fig. 1, d_a and d_d are two points and their distances to d_s are the same ($\overline{d_s d_a} = \overline{d_s d_d}$). Similarly, $\overline{d_s d_b} = \overline{d_s d_e}$ and $\overline{d_s d_c} = \overline{d_s d_f}$. Since three points can be used to solve a circle, (d_s, d_a, d_d) , (d_s, d_b, d_e) and (d_s, d_c, d_f) can be used to solve the same circle candidate.

According to the above, we have to find two points whose distances are the same to the seed point. We create a sequence of cells and a pool of candidates. Each cell is denoted as C_j and is used to store a point whose distance is j to the seed point. The pool is a set and is used to store the solved candidate with an accumulator. When a seed point is selected, each cell and the pool are set to be empty. We also create a window that is centered at d_s with $w \times w$ size. We extract each edge point in the window one by one (beside d_s), and compute the distance between the extracted point and d_s . Assume that the distance between the extracted point and d_s is j , if C_j is empty, we put the point into C_j ; otherwise, we use the extracted point, seed point and the point in C_j to solve a circle candidate. For example, see Fig. 2, if the distance from a point, d_a , to d_s is h points and the corresponding cell, C_h , is empty, then we put d_a into C_h . Similarly, d_b is put into C_k . If the distance from a point, d_c , to d_s is also h points, but C_h is not empty, we use d_c , d_s and d_a to obtain a circle candidate.

Fig. 1. There are two points whose distances to the seed point (d_s) are the same on this circle ($\overline{d_s d_a} = \overline{d_s d_d}$, $\overline{d_s d_b} = \overline{d_s d_e}$ and $\overline{d_s d_c} = \overline{d_s d_f}$).

Fig. 2. The seed point checking. If the point has a corresponding cell that is empty, put the point into the cell (such as d_a and d_b), otherwise, use three points (d_s , d_a and d_c) to solve a circle candidate.

If the obtained circle candidate does not exist in the candidate pool, we add this candidate into the pool with an accumulator (the accumulator is set to be 1); otherwise, the accumulator is increased by 1. When every edge point in the window is processed, we select a circle candidate with the highest accumulation value from the candidate pool. Then we extract the edge points on the selected circle from the image. Assume that selected circle is denoted as (x'_c, y'_c, r'_c) , where (x'_c, y'_c) is the center and r'_c is the radius of the circle. The edge points on the selected circle can be described as

$$\tilde{D} = \left\{ d_i \mid \left| \sqrt{(x_i - x'_c)^2 + (y_i - y'_c)^2} - r'_c \right| \leq \varepsilon_r, d_i \in D \right\}, \quad (2)$$

where ε_r is a predefined threshold value (we set $\varepsilon_r = 1$ in this paper). We can recognize the selected circle as a true circle if

$$\frac{N_c}{2r'_c\pi} > R_t, \quad (3)$$

where N_c is number of the points of $\tilde{D}$ and R_t is the minimum ratio between the real points number and the ideal circumference.

Since the FRHT is a kind of the probabilistic methods, the above procedure is iterated until the number of randomly picked seed points is enough. The number of iterations is dependent on the image complexity and effected by the computational time (see Sec. 4 for details).

3. Circular Arc Estimation

Circle arc is also an important feature for the applications of machine vision.¹⁷ Five parameters, the center (x_c, y_c) , the radius (r_c), the angle of the symmetry axis to the

Fig. 3. The parameters of a circular arc: (x_c, y_c) is the center coordinate, r_c is the radius, θ is the angle of the symmetry axis, and the ratio of complete circle is denoted as R .

x -axis (θ) and the ratio of the complete circle (R), are used to describe a circular arc (see Fig. 3). We regard the circular arc as a part of a circle. We can detect three parameters (the center (x_c, y_c) and the radius (r_c)) of the circular arc using the FRHT. After the circle is detected, we assume that the circle arc is continuous and we apply a simple geometric method to estimate another two parameters of the circular arc.

Since the circular arc is symmetric by the symmetry axis, it is easy to see that the centroid of a circular arc is on the symmetry axis of the arc. When a circle is recognized and the points on this circle are put in the set $\tilde{D}$, the angle of the symmetry axis to the x -axis can be described as

$$\theta = \tan^{-1} \left(\frac{\bar{y} - y_c}{\bar{x} - x_c} \right), \quad (4)$$

where $(\bar{x}, \bar{y})$ is the centroid of $\tilde{D}$. The ratio of the complete circle can be found by

$$R = \frac{N_c}{2r_c\pi}, \quad (5)$$

where N_c is the number of points of $\tilde{D}$. Please note that when the circular arc is small, the estimation of θ may fail. In most applications (such as the experiments in Sec. 4), the small arc is the noise and can be filtered out by a radius threshold.

4. Discussion and Experiments

The main disadvantages of Hough transform techniques are large computational time and storage requirements. In this section, we compare the computation and storage requirements of the proposed method (the fast randomized Hough transform,

FRHT) with the randomized Hough transform²⁰ (RHT), the adaptive randomized Hough transform⁹ (ARHT), and the effective voting method⁴ (EVM).

4.1. Analysis of the FRHT

In the proposed method, as like the RHT, we have to decide the iteration times (the number of picking seed points). The iteration times depend on the complexity of the image. We assume that there are M data points and the target circle contains N pixels in the image. The probability of randomized picking of a seed point from the target circle is

$$P_{\text{FRHT}} = \frac{N}{M}. \quad (6)$$

In general, we can pick one seed point that is on the target circle when we pick M/N times at random. The minimum iteration times of the FRHT can be set as

$$I_{\text{FRHT}} = \frac{M}{N}. \quad (7)$$

Since we select a window and every point in the window is used to process with the picked seed point, the computational time is related to the number of points in the window. In the worst case, the size of the selected window is the same as the size of the image, and then the computational time of the FRHT is (with the biggest window size)

$$T_{\text{FRHT}} = M \cdot I_{\text{FRHT}} = \frac{M^2}{N}. \quad (8)$$

After the seed point is selected, we find two points whose distances are the same to the seed point by a sequence of cells. The sequence of cells (C_j) is used to store a point whose distance is j to the seed point. Assume that the size of the image is $W \times H$ and the maximum distance of the image is $\sqrt{W^2 + H^2}$ (which is equal to the diagonal length of the image). Since the digital image is discrete, the diagonal length of the image is equal to the largest value of W and H . We set

$$L = \max\{W, H\}. \quad (9)$$

Since each cell is used to store a point coordinate, the maximum storage requirement of the sequence of cells is L .

We also use a pool to store the candidates. It is not easy to estimate the size of the pool of candidates. In the worst case, each point in the image can be formed into the smallest circle. We set the maximum number of candidates as equal to M . Each candidate contains four elements: three parameters of the circle and an accumulator. The storage requirement of the FRHT (S_{FRHT}) is

$$S_{\text{FRHT}} = L + 4M \quad (10)$$

4.2. Comparing the storage requirement

In the RHT and the ARHT, each picked three points are used to solve a circle candidate and put into the pool of candidates with an accumulator. As with the FRHT, we also assume that each point in the image can be formed into the smallest circle. Then the storage requirement of the RHT (S_{RHT}) and the ARHT (S_{ARHT}) is

$$S_{\text{RHT}} = S_{\text{ARHT}} = 4M. \quad (11)$$

In the EVM, the maximum storage requirement (S_{EVM}) is equal to five times the number of data points.⁴ In other words,

$$S_{\text{EVM}} = 5M. \quad (12)$$

Comparing the storage requirement of the FRHT, the RHT, the ARHT and the EVM [see Eqs. (10)–(12)], it is found that they are similar to each other. We can say that the storage requirement of these four methods is not heavy for the computer system.

4.3. Comparing the computational time

To detect a circle with the RHT, we have to pick three points out of the data points in each iteration. We assume that the same three points will not show up repeatedly. The probability of picking one point from the target circle is N/M . After the first point is picked from the circle, the probability of picking the second point and the third point from the same circle are $(N-1)/(M-1)$ and $(N-2)/(M-2)$, respectively. In practice, $N \gg 2$ and $M \gg 2$, the probability of picking three points from the same circle is

$$P_{\text{RHT}} = \frac{N(N-1)(N-2)}{M(M-1)(M-2)} \approx \frac{N^3}{M^3}. \quad (13)$$

Comparing the probability of the FRHT [Eq. (6)], we can see that $P_{\text{FRHT}} \geq P_{\text{RHT}}$ because $M \geq N$. In other words, the success probability of the FRHT is higher than the RHT.

In the RHT, we pick three points and we solve a circle candidate in each iteration. Therefore, the computational time of the RHT (T_{RHT}) equal the iteration times of the RHT (I_{RHT}). The minimum iteration times (I_{RHT}) of the RHT can be described as

$$T_{\text{RHT}} = I_{\text{RHT}} = \frac{M^3}{N^3}. \quad (14)$$

Comparing the computational time of the FRHT and the RHT, as seen in Eqs. (8) and (14), we find that the ratio of T_{FRHT} to T_{RHT} is

$$\frac{T_{\text{FRHT}}}{T_{\text{RHT}}} = \frac{M^2}{N} \frac{N^3}{M^3} = \frac{N^2}{M}. \quad (15)$$

Fig. 4. The relation between ratio of T_{FRHT} to T_{RHT} and M with a fixed $N = 100$ [see Eq. (15)].

The bigger value of M means the image is more complex. From Eq. (15), we can see that when $N^2 < M$, the FRHT's computational time is less than the RHT's. We set a fixed $N = 100$, and plot the relation between ratio of T_{FRHT} to T_{RHT} and M (see Fig. 4). From Eq. (15) and Fig. 4, we can see that the FRHT spends less computational time when the image is more complex.

In the EVM, we pick the point pairs from the data points and use the chosen pair of points to check the third point. We also use an interval to reduce the original data number. It is similar to the window size selection of the FRHT. We decide the interval is equal to 1, and the computation needed for the EVM (T_{EVM}) is⁴

$$T_{\text{EVM}} = n_p \frac{M(M-1)}{2}, \quad (16)$$

where n_p is the number of the checked third points for each point pair and it may be different for each point pair. Since $M \gg 1$ and we assume the average n_p of all point pairs is equal to 1, Eq. (16) can be written as

$$T_{\text{EVM}} = n_p \frac{M(M-1)}{2} \approx \frac{M^2}{2}. \quad (17)$$

Comparing the computational time of the FRHT and the EVM [see Eqs. (8) and (17)], the ratio of T_{FRHT} to T_{EVM} is

$$\frac{T_{\text{FRHT}}}{T_{\text{EVM}}} = \frac{M^2}{N} \frac{2}{M^2} = \frac{2}{N}. \quad (18)$$

In general, the point number of the target circle should be bigger than two pixels, therefore the ratio of T_{FRHT} to T_{EVM} is less than one. It means that the FRHT requires less computation time than the EVM.

In the ARHT, a moving window is used to decompose the original image into several subimages. Then the ARHT picks three points from each window at random. If the picked three points are on a circle, the circle can be detected. When the window size can cover all points on a circle, the minimum iteration times in each window is

$$I_{\text{ARHT}} = \frac{M'^3}{N^3}, \quad (19)$$

where M' is the number of edge pixels in the window. Since the ARHT uses the moving windows, the computation requirement is the iteration times in each window multiplied by the number of windows. It is not easy to obtain a fixed M' to estimate the computation needed for the ARHT because each window contains different number of pixels.

Since the n_p of the EVM and the computation requirement of the ARHT are dependent on the complexity of the image, we use a noise addition experiment to compare the real computational times of these two methods (the EVM and the ARHT) and the FRHT. We create an image whose size is 256×256 and draw a circle on the created image [see Fig. 5(a)]. We add 1%, 2%, ..., 25% random pepper noise to Fig. 5(a), respectively. An example of an image with added noise is shown in Fig. 5(b). We implement the RHT, the EVM, the ARHT and the FRHT to detect the circle with a counter to record the computational times. We also repeat the

Fig. 5. The examples of noise addition experiment: (a) an image with no noise; (b) the image with 5% pepper noise.

(a)

(b)

Fig. 6. The results of the noise addition experiment: (a) the average computational times; (b) the average error.

experiment 50 times for each noise image. The average computational times are shown in Fig. 6(a). In the noise addition experiment, the accuracy of each method is shown in Fig. 6(b). In Fig. 6(b), the error is defined by

$$d_e = \sqrt{(x'_c - x_c)^2 + (y'_c - y_c)^2 + (r'_c - r_c)^2}, \quad (20)$$

where (x_c, y_c, r_c) is the true circle and (x'_c, y'_c, r'_c) is the detected circle. From the experimental results, the FRHT spends less computational time than that of the RHT, the EVM or the ARHT. We can also see that the FRHT can detect the circle as exact as the previous methods do.

4.4. Comparing the success rates

Since the randomized-based methods are based on picking points from the feature data, the efficiency of the method depends on the probability of picking the points of the true circle. The probability of randomly picked points from the true circle is called the success rate.

We redo the noise addition experiment (see Sec. 4.3) and detect the circle by the RHT, the EVM, the ARHT and the FRHT. We record the times that the true circle is detected. The average times of success detection (success rate) of each method are shown in Fig. 7. According to Fig. 7, when the image is complex (such as 10%–25% noise addition), the success rate of the FRHT is higher than that of other methods. We can see that the FRHT is more suitable for the complex images.

4.5. Real image experiments

We use a printed circuit board (PCB) image and a park picture to compare the performance of the RHT, the EVM, the ARHT and the FRHT. A PCB image that contains several circles is shown in Fig. 8(a). Figure 8(b) is the edge detection result of Fig. 8(a). We apply the RHT, the EVM, the ARHT and the FRHT (with $w = 20$

Fig. 7. The average success rate.

Fig. 8. An experiment for a PCB image: (a) the original image; (b) the edge information; (c) the circle detection result.

and $R_t = 0.7$) to detect circles, respectively. The circle recognition results of all methods are the same and shown in Fig. 8(c). A picture is taken in a park as shown in Fig. 9(a). There is a bicycle under the basketball stand. The edge image of Fig. 9(a) is shown in Fig. 9(b). Since success rates of the above method are different (see Sec. 4.4), the RHT and the ARHT cannot detect the circle sometimes. We only show the success detection result in Fig. 9(c). Since the storage requirements of these methods are not heavy for the computer system (see Sec. 4.2), we compare the computational times with the real images in Figs. 8(a) and 9(a). The computational times is counted when the algorithm recognizes the circle. The comparison of computational times is shown in Table 1. We can see that the FRHT saves more computation requirement than others.

We also use two real images to show the results of circular arc estimation. We apply the proposed method for the circle detection then apply the proposed circular arc estimation (see Sec. 3) to recognize the arc. Figure 10(a) is an image taken from an ATM (automatic teller machine) monitor. A criminal is operating the ATM with a safety helmet. We try to detect the helmet via the circular arc detection by

Fig. 9. An experiment for a park image: (a) the original image; (b) the edge information; (c) the circle detection result.

Table 1. The results of computation (times).

Figure	Fig. 8(a)	Fig. 9(a)
RHT	132,224	2,306,693
EVM	50,920	686,769
ARHT	5611	57,284
FRHT	1184	6543

1000 iteration times, $w = 20$ and $R_t = 0.4$. The result of edge detection and the circular arc recognition is shown in Fig. 10(b). Since the helmet is not a small arc, we keep the arc whose radius is bigger than 20 pixels as the helmet. The result of the safety helmet detection is shown in Fig. 10(c). We took a picture from the roadside as shown in Fig. 11(a). The proposed method is applied to find the circular arc with 1000 iteration times, $w = 30$ and $R_t = 0.4$. The result of edge detection and the circular arc recognition is shown in Fig. 11(b). We keep the arc whose radius is bigger than 30 pixels as the helmet. The result of the safety helmet detection is shown in Fig. 11(c). According to the results, the circle arc detection can be applied to the obtained circle in practical applications.

Fig. 10. A real ATM image for the experiment of circular arc recognition: (a) the original image; (b) the result of the edge detection and the circular arc recognition; (c) the recognized safety helmet.

5. Conclusions

The drawbacks of Hough transform are the high computation time and storage requirement. To improve the Hough based technology, the probability methods (such as the RHT and the ARHT) and the multistep schemes (such as the EVM) have been proposed. In the circle detection with the complex images, since the success rate of the RHT and the ARHT is low and the computational time of the EVM is high, we propose a modified randomized Hough transform that is named the fast randomized Hough transform (FRHT) to increase the success rate and to decrease the computational time. Firstly, we pick only one point from the data points to be the seed point. And a rule is applied to check whether the seed point is on a true circle. The proposed method is also extended to detect the circular arcs. Comparing with the RHT, the ARHT and the EVM, it can be seen that the success rate of the FRHT is higher than that of others. Moreover, the FRHT requires less computational time than others. We can see that the FRHT is more suitable for circle

Fig. 11. The experiment of a roadside picture: (a) the original image; (b) the result of the edge detection and the circular arc recognition; (c) the recognized safety helmet.

detection in complex images. In the experiments, synthetic images and real images are used to show that the proposed method can extract circle/circular arc correctly.

Acknowledgment

This work was supported by National Science Council, Taiwan (NSC 96-2221-E-324-018).

References

1. A. Amin, Recognition of hand-printed Latin characters based generalized Hough transform and decision tree learning techniques, *Int. J. Patt. Recogn. Artif. Intell.* **14** (2000) 369–387.
2. O. Chutatape and L. A. Guo, A modified Hough transform for line detection and its performance, *Patt. Recogn.* **32** (1999) 181–192.
3. T.-C. Chen and K.-L. Chung, An efficient randomized algorithm for detecting circles, *Comput. Vis. Image Underst.* **83** (2001) 172–191.
4. S.-H. Chiu and J.-J. Liaw, An effective voting method for circle detection, *Patt. Recogn. Lett.* **26** (2005) 121–133.
5. E. R. Davies, A modified Hough scheme for general circle location, *Patt. Recogn. Lett.* **7** (1987) 37–43.

6. R. D. Duda and P. E. Hart, Use of the Hough transform to detect lines and curves in pictures, *Commun. Assoc. Comput. Mach.* **15** (1972) 11–15.
7. G. L. Foresti, C. S. Regazzoni and G. Vernazza, Circular arc extraction by direct clustering in a 3D Hough parameter space, *Sign. Process.* **41** (1995) 203–224.
8. J. Y. Goulermas and P. Liatsis, Incorporating gradient estimations in a circle-finding probabilistic Hough transform, *Patt. Anal. Appl.* **2** (1999) 239–259.
9. S.-Y. Guo, X.-F. Zhang and F. Zhang, Adaptive randomized Hough transform for circle detection using moving window, *The 5th Int. Conf. Machine Learning and Cybernetics*, Dalian, China, 2006.
10. P. V. C. Hough, A method and means for recognizing complex patterns, U.S. Patent 3069654 (1962).
11. D. Ioannou, W. Huda and A. F. Laine, Circle recognition through a 2D Hough transform and radius histogramming, *Image Vis. Comput.* **17** (1999) 15–26.
12. J. Illingworth and J. Kittler, A survey of the Hough transform, *Comput. Vis. Graph. Image Process.* **44** (1988) 87–116.
13. H. Kälviäinen and P. Hirvonen, An extension to the randomized Hough transform exploiting connectivity, *Patt. Recogn. Lett.* **18** (1997) 77–85.
14. N. Kiryati, Y. Eldar and A. M. Bruckstein, A probabilistic Hough transform, *Patt. Recogn.* **24** (1991) 303–316.
15. N. Kiryati, H. Kälviäinen and S. Alaoutinen, Randomized or probabilistic Hough transform: Unified performance evaluation, *Patt. Recogn. Lett.* **21** (2000) 1157–1164.
16. N. Lu and Z. Feng, Accumulative intersection space based corner detection algorithm, *Int. J. Patt. Recogn. Artif. Intell.* **22** (2008) 1159–1586.
17. S.-C. Pei and J.-H. Horng, Circular arc detection based on Hough transform, *Patt. Recogn. Lett.* **16** (1995) 615–625.
18. Y. Rianto, S. Kondo and T. Kim, Detection of roads from satellite images using optimal search, *Int. J. Patt. Recogn. Artif. Intell.* **14** (2000) 1008–1023.
19. A. Torii and A. Imiya, The randomized-Hough-transform-based method for great-circle detection on sphere, *Patt. Recogn. Lett.* **28** (2007) 1186–1192.
20. L. Xu, E. Oja and P. Kultanen, A new curve detection method: Randomized Hough transform (RHT), *Patt. Recogn. Lett.* **11** (1990) 331–338.

Shih-Hsuan Chiu received the B.Sc. degree in mechanism engineering from Chung Yuan Christian University in 1983, and the M.Sc. and the Ph.D. degrees in mechanical engineering from the University of Tokyo in 1988 and 1991, respectively. Currently, he is a

professor in National Taiwan University of Science and Technology.

His current interests are mainly in the field of control theories, digital signal processing and machine vision.

Jiun-Jian Liaw received the B.Sc., the M.S. and the Ph.D. degrees in polymer engineering from National Taiwan University of Science and Technology in 1997, 2001 and 2005, respectively. He is a faculty member at Chao-Yang University of Technology.

His research interests include data hiding, pattern recognition, and computer vision.

Kuo-Hung Lin received the B.S. degree from Tamkang University in 2001, and M.S. degree from National Taiwan University of Science and Technology in 2004, both in mechanism engineering. He is currently a student working on his Ph.D.