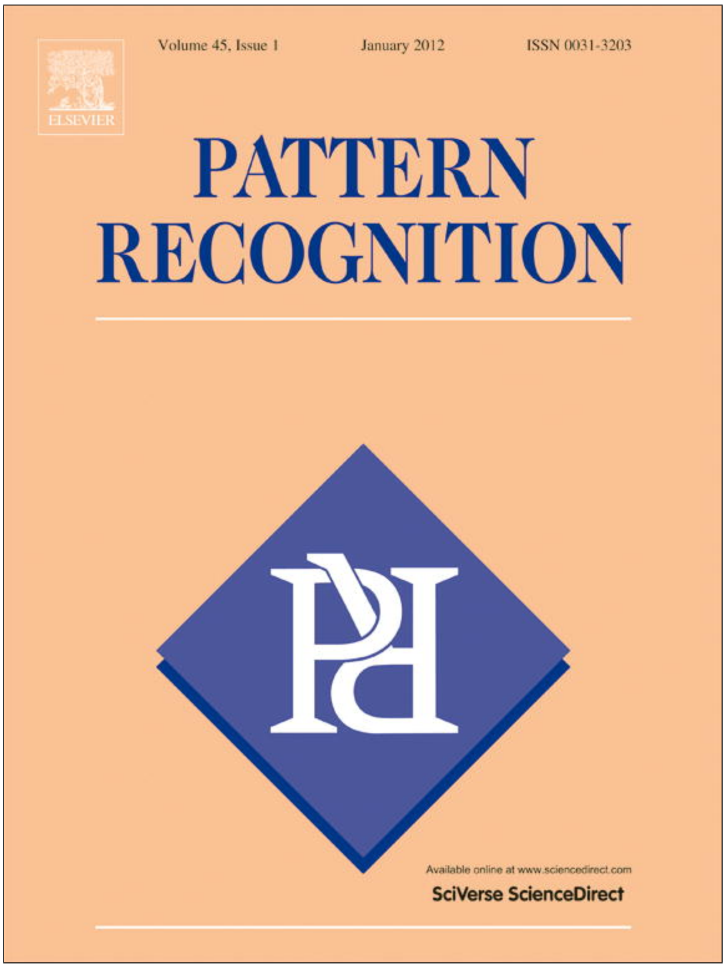
Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

http://www.elsevier.com/copyright

Author's personal copy

Pattern Recognition 45 (2012) 252–263

Contents lists available at ScienceDirect

Pattern Recognition

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

Efficient sampling strategy and refinement strategy for randomized circle detection

Kuo-Liang Chung [a,1], Yong-Huai Huang [b,*,2], Shi-Ming Shen [a], Andrey S. Krylov [c,3], Dmitry V. Yurin [c,3], Ekaterina V. Semeikina [c]

[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, An-Chung Road, Hsin-Tien, Taipei 23154, Taiwan, ROC
[c] Laboratory of Mathematical Methods of Image Processing, Faculty of Computational Mathematics and Cybernetics, Moscow Lomonosov State University, Leninskie Gory, 2nd Educational Building, Office 638, Moscow 119991, Russia

ARTICLE INFO

Article history:
Received 13 September 2010
Received in revised form
30 May 2011
Accepted 5 July 2011
Available online 18 July 2011

Keywords:
Circle detection
Hough transform
Randomized algorithms
Sampling Strategy

ABSTRACT

Circle detection is fundamental in pattern recognition and computer vision. The randomized approach has received much attention for its computational benefit when compared with the Hough transform. In this paper, a multiple-evidence-based sampling strategy is proposed to speed up the randomized approach. Next, an efficient refinement strategy is proposed to improve the accuracy. Based on different kinds of ten test images, experimental results demonstrate the computation-saving and accuracy effects when plugging the proposed strategies into three existing circle detection methods.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Circle detection is important and fundamental in pattern recognition and computer vision [8,10,11,14,19]. In the past two decades, the detection accuracy and computation performance are two main concerned issues and many circle detection methods have been developed. The Hough transform-based (HT-based) approach for recognizing complex patterns is first presented by Hough [13]. Later Duda and Hart [9] use the HT to detect curves. Because of the adaption of voting strategy allowable in the accumulator array, the HT-based approach has the accuracy advantage. To reduce the computing time and memory space requirements, several improved HT-based methods [4,7,12,16–18,20,21,27] have been developed by using either geometrical properties or the decomposition of the parameter space. However, the computing time required in these methods is difficult to be reduced significantly.

To improve the computation performance significantly, several randomized circle detection methods [3,5,6,22,23,24,25,26,28] have been developed. In the randomized HT (RHT) method proposed by Xu et al. [25,26], each time it randomly selects three edge pixels, and then the corresponding mapped points in the parameter space are collected by voting on a 3-D accumulator array or a link-list data structure. Based on the RHT, Lu and Tan [24] presented an iterative RHT (IRHT) to detect circles, lines, and ellipses. Combining the sampling strategy in the RHT and particle swarm optimization technique, Cheng et al. [5] proposed an efficient method for circle detection. Based on a parameter-free approach without using any accumulator arrays, the RCD method proposed by Chen and Chung [3] first randomly samples four edge pixels in which three selected edge pixels are used to construct a possible circle, and the remaining edge pixel is used to confirm whether the possible circle can be promoted to a candidate circle or not. If yes, the RCD performs a voting process to determine whether the candidate circle is a true circle or not. Experimental results show that the RCD is faster than the RHT when the noise level is ranged from light level to modest level. Here the range between two levels means that the number of noisy pixels over the number of true circle pixels is at most 170%. Later, one improved lookup-table based method was presented in [6] to speed up the computation performance. To improving the accuracy of the RCD, Lee et al. [23] proposed an $O(|V|^2)$-time refinement strategy where $|V|$ denotes the number of edge pixels in the edge map. The motivations of our research are twofold: (1) presenting a new multiple-evidence-based efficient

* Corresponding author.
E-mail addresses: k.l.chung@mail.ntust.edu.tw (K.-L. Chung), yonghuai@ms28.hinet.net (Y.-H. Huang).

[1] Supported by the National Council of Science of ROC under contract NSC98-2923-E-011-001-MY3.
[2] Supported by the National Science Council of ROC under the contract NSC99-2218-E-228-001.
[3] Supported by the Russian Foundation for Basic Research under the contract 09-07-92000-HHC_a.

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.07.004

Author's personal copy

sampling strategy to significantly reduce the computation performance and (2) presenting a new linear-time, i.e. $O(|V|)$-time, refinement strategy to improve the accuracy.

In this paper, two novel strategies are presented to improve the computation and accuracy performance of some existing randomized circle detection methods. We first present a multiple -evidence-based sampling strategy which uses three evidences to discard a large amount of invalid possible circles and candidate circles, and this strategy leads to a significant computation-saving effect. For enhancing accuracy, we present a new $O(|V|)$-time refinement strategy, which is quite different from Lee et al.'s method, to refine the parameters of the detected true circle. Based on ten images, experimental results illustrate the computation and accuracy advantages of our proposed two strategies.

The rest of this paper is organized as follows. Section 2 revisits the sampling strategy of the RCD and the refinement strategy by Lee et al.'s. In addition, Section 2 points out the related accuracy and computation overhead problems. In Section 3, the proposed multiple-evidence-based sampling strategy is presented. In Section 4, the proposed refinement strategy is presented. Section 5 demonstrates the computation and accuracy performance improvement. Finally, some concluding remarks are drawn in Section 6.

2. Problems in RCD's sampling strategy and Lee et al.'s refinement strategy

In this section, we first revisit the sampling strategy of the RCD [3] and point out its inherent computation overhead and the bias problem. Then, we revisit Lee et al.'s refinement strategy [23] and highlight the time-consuming problem. The above computation overhead and accuracy problems motivate the research of this paper.

2.1. The computation overhead and bias problems in the RCD

In the RCD [3], the Sobel edge detector [11] is first applied to the input image to construct a set of edge pixels V and each edge pixel in V is denoted by $v_p=(x_p,y_p)$ for $0 \le p < |V|$. Each time, the RCD randomly selects four edge pixels, say v_i, v_j, v_k, and v_l, from V and use three of them, say v_i, v_j, and v_k, to determine a possible circle C_{ijk} where the center (a_{ijk},b_{ijk}) and the radius r_{ijk} are calculated by

$$a_{ijk}=\frac{\begin{vmatrix} x_j^2+y_j^2-(x_i^2+y_i^2) & 2(y_j-y_i) \\ x_k^2+y_k^2-(x_i^2+y_i^2) & 2(y_k-y_i) \end{vmatrix}}{4((x_j-x_i)(y_k-y_i)-(x_k-x_i)(y_j-y_i))} \tag{1}$$

$$b_{ijk}=\frac{\begin{vmatrix} 2(x_j-x_i) & x_j^2+y_j^2-(x_i^2+y_i^2) \\ 2(x_k-x_i) & x_k^2+y_k^2-(x_i^2+y_i^2) \end{vmatrix}}{4((x_j-x_i)(y_k-y_i)-(x_k-x_i)(y_j-y_i))} \tag{2}$$

and

$$r_{ijk}=\sqrt{(x_p-a_{ijk})^2+(y_p-b_{ijk})^2} \tag{3}$$

for $p \in \{i,j,k\}$. Furthermore, we check whether v_l is close to C_{ijk} or not. If no, i.e. the evidence is negative, we kick out C_{ijk} and select next four edge pixels randomly; otherwise, C_{ijk} is promoted to a candidate circle, and then the RCD performs a voting process to count the number of edge pixels lying on C_{ijk} to determine whether C_{ijk} is a true circle or not.

We now investigate the computation overhead problem in the RCD. We find that this problem is highly related to the number of possible circles and candidate circles. Let N_P and N_C denote the number of possible circles and candidate circles appeared in the RCD, respectively. After performing the RCD on ten test images as shown in Fig. 1, namely coin, cake, insulator, gobang, plates, logo, speaker, stability-ball, ball, and swatch, with sizes 256×256, 256×256, 256×192, 256×256, 400×360, 283×344, 485×437, 374×374, 350×350, and 309×356, respectively, Table 1 shows the values of N_P and N_C for each test image and the average value of N_P is 35 976, and it reveals that a considerable computational overhead is needed. Precisely speaking, Eqs. (1)–(3) are called 35 976 times to construct these possible circles. Using the fourth sampled edge pixel as an evidence checker, the RCD can discard 95% of possible circles $(=N_P-N_C/N_P=35\,976-1373/35\,976)$ and promotes the remaining 1373 possible circles to candidate circles. It indicates that the voting process will be called 1373 times for these candidate circles to examine which of them can be promoted to true circles or not. Running the voting process 1373 times on these candidate circles costs a large amount of computation overhead since in real case, only few true circles are existed in one image. In Section 3, our proposed novel multiple-evidence-based sampling strategy will be presented to discard those invalid possible circles and candidate circles to achieve significant computation-saving effect.

Besides the computation overhead problem, the bias problem is existed in the RCD. The main reason is that although each detected true circle collects enough number of votes, its center and radius are only determined by the three edge pixels v_i, v_j, and v_k. Fig. 2 depicts the bias problem. The gray circle as shown in Fig. 2(a) and the dash-lined biased circle as shown in Fig. 2(b) denote the ideal detected circle and the circle detected by the RCD, respectively. Usually, the detected circles by the RCD are somewhat different from the ideal detected circle and it causes a bias problem. Recently, Lee et al. presented a refinement strategy to improve the accuracy of the RCD, but it suffers from the time-consuming problem.

2.2. Time-consuming problem in Lee et al.'s RCD-based refinement strategy

Suppose C_{ijk} is a biased circle detected by the RCD. Lee et al.'s refinement strategy first constructs an annulus A_{ijk} composed of the region lying between two circles concentric with C_{ijk} and their radii are $r_{ijk}-\Delta$ and $r_{ijk}+\Delta$. Let V' denote the edge set within A_{ijk} and it is used to construct a set of new circles. After running the voting process on each constructed new circle, Lee et al.'s strategy selects a refined circle with the maximal number of votes. Fixing two edge pixels, v_j and v_k, and replacing the remaining one, v_i, by each edge pixel in V' except v_i, v_j, and v_k, we can construct $(|V'|-3)$ new circles by Eqs. (1)–(3). For each new circle, a voting process associated with $(|V'|-4)$ edge pixels is performed. It takes $O(|V'|^2)$ $(=|V'|^2-7|V'|+12=(|V'|-3)\times(|V'|-4))$ time to perform voting process $(|V'|-3)$ times. Suppose the circle C_{p_1jk} determined by v_{p_1}, v_j, and v_k has the maximal number of votes, then it will be selected as a better refined circle. Continuing the same way, it takes $O(|V'|^2)$ $(=|V'|^2-7|V'|+20=(|V'|-4)\times(|V'|-5))$ time to determine the next better refined circle $C_{p_1p_2k}$. Further, it takes $O(|V'|^2)$ $(=|V'|^2-11|V'|+30=(|V'|-5)\times(|V'|-6))$ time to obtain the final refined circle $C_{p_1p_2p_3}$. Although Lee et al.'s refinement strategy can improve the accuracy of the RCD, taking $O(|V'|^2)$ time is time-consuming. In Section 4, we will present an $O(|V'|)$-time refinement strategy to improve the accuracy. Experimental results will demonstrate that our proposed refinement strategy has similar accuracy as Lee et al.'s strategy does, but ours has significant execution-time superiority.

3. The proposed multiple-evidence-based sampling strategy

In this section, a novel multiple-evidence-based sampling strategy is presented to significantly alleviate the RCD's computational overhead problem. As shown in Fig. 3, our proposed strategy considers

Author's personal copy

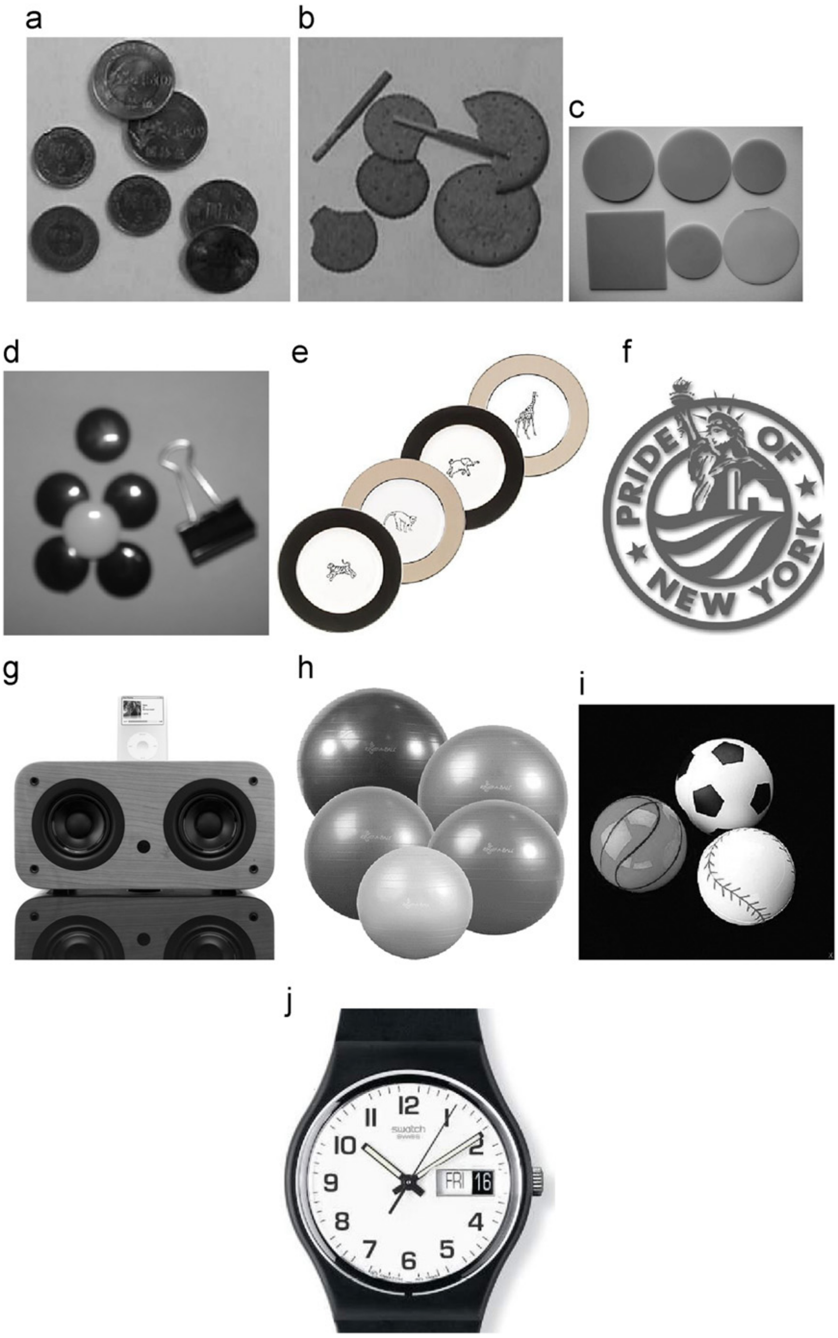


Fig. 1. Ten test images: (a) Coin image. (b) Cake image. (c) Insulator image. (d) Gobang image. (e) Plates image. (f) Logo image. (g) Speaker image. (h) Stability-ball image. (i) Ball image. (j) Swatch image.

Table 1
Number of possible circles and candidate circles appeared in the RCD.

	Coin	Cake	Insulator	Gobang	Plates	Logo	Speaker	Stability-ball	Ball	Swatch	Average
N_P	49051	34704	35928	43868	52048	33377	29362	28259	29254	23910	35976
N_C	2143	1871	1809	2041	1562	930	827	849	1011	691	1373

three evidences where the first evidence can discard invalid possible circles and the remaining two evidences can discard invalid candidate circles. Starting from four random selected edge pixels, their gradient directions are used as the first evidence to determine whether they have high probability to lie on a circle or not. As the first evidence, if it is positive, we thus construct a possible circle. Next, the second and third evidences are used to determine whether the possible circle can be promoted to the true circle or not. Here, we take the distance between the fourth edge pixel and the possible circle as the second evidence. Further, the third evidence is evaluated by checking whether all gradient directions of four edge pixels point to the center of possible circle or not.

If all three evidences are positive, we promote the possible circle to a candidate circle, and then run the voting process on the

Author's personal copy

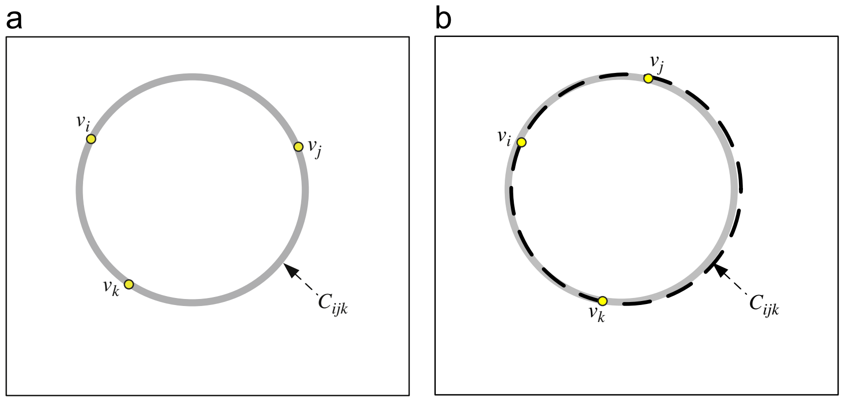


Fig. 2. Bias problem occurred in the RCD. (a) Ideal detected circle denoted by gray line. (b) Biased circle, which is denoted by black dash line, detected by the RCD.

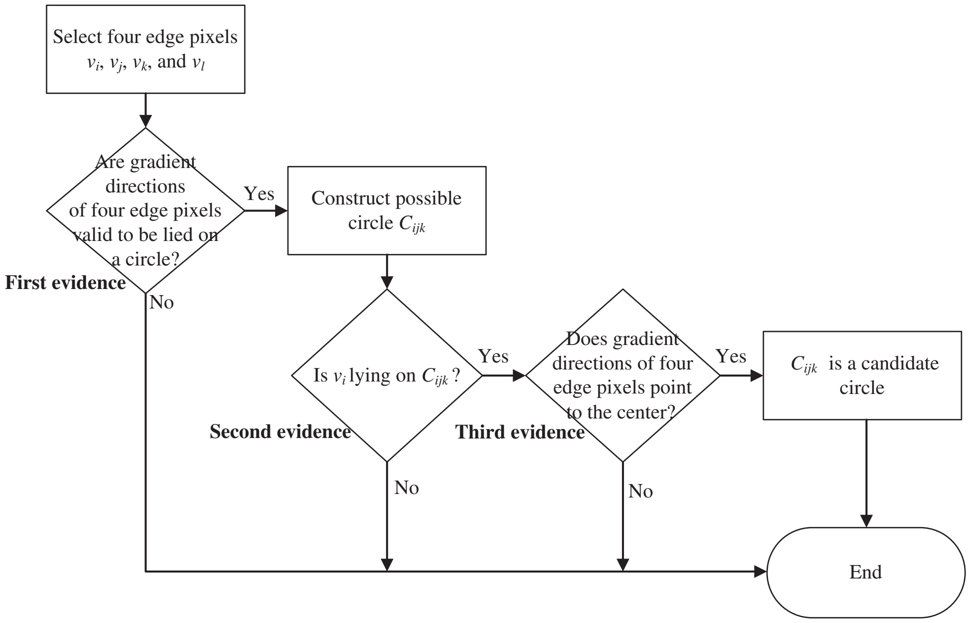


Fig. 3. The flowchart of the proposed multiple-evidence-based sampling strategy.

candidate circle to determine whether it is a true circle or not. Based on the above hierarchical evidence verification process, a large amount of invalid possible circles and candidate circles can be discarded together and it leads to improve the computation performance of the RCD significantly. In the following two subsections, the components in Fig. 3 are described in detail.

3.1. Calculation of gradient directions

Our proposed sampling strategy first calculates the two gradients of each edge pixel $v_p = (x_p, y_p)$ by

$$G_p^x = -\frac{1}{2\pi\sigma^4}\sum_x\sum_y f(x,y)\cdot xe^{-(x^2+y^2)/2\sigma^2} \quad (4)$$

$$G_p^y = -\frac{1}{2\pi\sigma^4}\sum_x\sum_y f(x,y)\cdot ye^{-(x^2+y^2)/2\sigma^2} \quad (5)$$

where G_p^x and G_p^y denote the gradients in x-direction and y-direction, respectively; $f(x,y)$ is the gray-level value at location (x,y) for $x_p - \lceil 3\sigma \rceil \le x \le x_p + \lceil 3\sigma \rceil$ and $y_p - \lceil 3\sigma \rceil \le y \le y_p + \lceil 3\sigma \rceil$; σ is set to 1.25. The gradient direction of v_p is obtained by $\theta_p = \tan^{-1}(G_p^y/G_p^x)$ where $-\pi \le \theta_p \le \pi$.

For two circles in Fig. 4, at the same position (x,y), the absolute difference between θ_p of Fig. 4(a) and θ_p of Fig. 4(b) is π. On the contrary, as shown in Fig. 4, the gradient directions at two different positions may have the same value, e.g. the pixel with $\theta_p = \pi/4$ in Fig. 4(a) and the one with the same gradient in Fig. 4(b) although the two pixels are located at different positions. In fact, the pixel with $\theta_p = \pi/4$ in Fig. 4(a) is located at a convex segment while the pixel with $\theta_p = \pi/4$ in Fig. 4(b) is located at a concave segment. In order to solve this gradient direction inconsistency problem, in what follows, we present a template-based approach to calibrate the gradient direction of one pixel to make the direction point to the center of the circle.

We take an example to explain the proposed gradient direction calibration scheme. As shown in Fig. 5(a), suppose one pixel is with gradient direction θ_p and located at the convex segment of one circle. Fig. 5(b) depicts one pixel with the same θ_p, but located at the concave segment due to different contrast between the circle object and the background. First, we put a 9×9 mask as shown in Fig. 5(c) to the pixel, v_p, denoted by a black square. Next, we compute the gradient directions of the 16 pixels covered by the upper-right gray area of Fig. 5(c), and the median of the corresponding sixteen gradient directions, say θ_u, is taken as a representative of the upper-right gradient direction of v_p. Similarly, we take the median of the 16 gradient directions of the pixel v_p, say θ_ℓ, covered by the lower-left

Author's personal copy

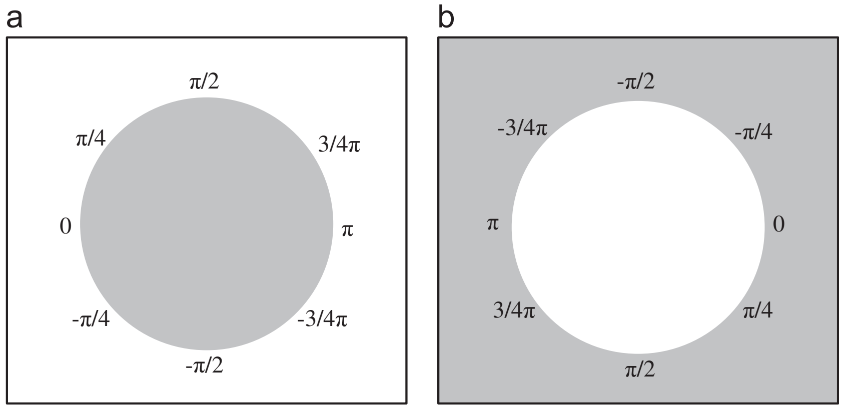


Fig. 4. Gradient direction inconsistence problem. (a) Gradient directions along the circle image with gray foreground and white background. (b) Gradient directions along the circle image with white foreground and gray background.

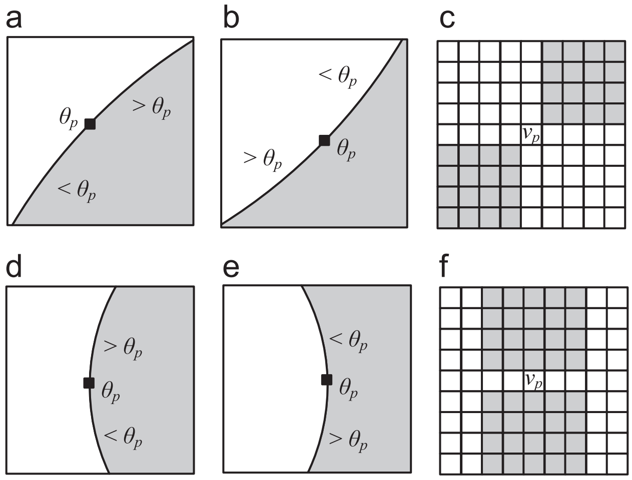


Fig. 5. Gradient direction calibration scheme. (a) and (b) Two cases for $\pi/8 \le \theta_p \le 3\pi/8$. (c) The template used to calibrate the gradient direction of (b). (d) and (e) Two cases for $-\pi/8 \le \theta_p \le \pi/8$. (f) The template used to calibrate the gradient direction of (e).

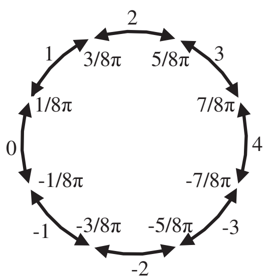


Fig. 6. The quantization of gradient directions.

gray area of Fig. 5(c) as the representative of the lower-left gradient direction of v_p. For Fig. 5(a), θ_u should be larger than θ_ℓ and it yields $\pi/8 \le \theta_p \le 3\pi/8$. For Fig. 5(b), θ_u should be less than θ_ℓ and it yields $-5\pi/8 \ge \theta_p \ge -7\pi/8$. By the same argument, for Fig. 5(d), θ_u should be larger than θ_ℓ and it yields $-\pi/8 \le \theta_p \le \pi/8$; for Fig. 5(e), it yields $-7\pi/8 \le \theta_p \le 7\pi/8$ because $\theta_u < \theta_\ell$. For saving space of the context, we omit the discussion of the remaining cases.

Following the above gradient direction calibration scheme, as shown in Fig. 6, we quantize any gradient direction θ_p into one of eight values, $\{-3,-2,-1,0,1,2,3,4\}$, by performing $\theta_p = \text{round}(\theta_p/\pi/4)$ and setting $\theta_p = 4$ if $\theta_p = -4$. For each quantized θ_p, a 9×9 window is utilized to collect a set of neighboring quantized gradient directions, and then the value of quantized θ_p is calibrated by using the proposed gradient direction calibration scheme.

3.2. Verify the validity of possible circles and candidate circles by multiple-evidence-based sampling strategy

In our proposed multiple-evidence-based sampling strategy mentioned in Fig. 3, we adopt three evidences to perform a hierarchical verification process to discard invalid possible circles and candidate circles. As shown in Fig. 7(a), four examples are taken to explain how the first evidence is used to discard the trial for constructing possible circles. In Fig. 7(a), assume v_i is the first random selected edge pixel and its gradient direction, θ_i, is 1. If v_i and the second selected edge pixel v_j are lying on the same circle and v_j is on the upper side of v_i, i.e. $y_j < y_i$, the quantized gradient direction of θ_j must be 1, 2, or 3; otherwise, as shown in Fig. 7(b), i.e. $y_i \le y_j$, the quantized gradient direction of θ_j must be in $\{0,1,3,4,-1,-2,-3\}$. The above argument is easy to tackle the case when v_j is on the left side of v_i or on the right side of v_i, i.e. $x_j < x_i$ or $x_j \ge x_i$, respectively. For example, in Fig. 7(c), v_j is on the right side of v_i, it is easy to know that θ_j must be in $\{1,2,3,4,-1,-2,-3\}$. On the contrary, in Fig. 7(d), v_j is on the left side of v_i and θ_j must be 0, 1, or -1. We thus claim that for cases in Fig. 7(a) and (b), the set of valid quantized gradient directions of θ_j's should be $\{1,2,3\}$ $(=\{1,2,3\} \cap \{1,2,3,4,-1,-2,-3\})$; for cases in Fig. 7(c) and (d), the set of valid quantized gradient directions of θ_j's should be $\{0,1,-1\}$ $(=\{0,1,3,4,-1,-2,-3\} \cap \{0,1,-1\})$. In other words, given the gradient direction of v_i, θ_i, if the quantized gradient direction of θ_j is not in the valid set, we stop constructing the possible circle.

According to the above description, given $v_i = (x_i, y_i)$ and θ_i, the valid gradient direction set of v_j, i.e. valid θ_j's, is obtained by

$$\Theta(v_i,\theta_i,v_j) = \Theta_V(y_i,\theta_i,y_j) \cap \Theta_H(x_i,\theta_i,x_j) \tag{6}$$

where $\Theta_V(y_i,y_j,\theta_i)$ and $\Theta_H(y_i,y_j,\theta_i)$ are used to determine the valid gradient direction set of $v_j = (x_j, y_j)$ for case_1 and case_2, respectively, and they are defined in Table 2. Based on Eq. (6), we mainly examine the three conditions, $\theta_j \in \Theta(v_i,\theta_i,v_j)$, $\theta_k \in \Theta(v_i,\theta_i,v_k) \cap \Theta(v_j,\theta_j,v_k)$, and $\theta_l \in \Theta(v_i,\theta_i,v_l) \cap \Theta(v_j,\theta_j,v_l) \cap \Theta(v_k,\theta_k,v_l)$, where θ_i, θ_j, θ_k, and θ_l are gradient directions of v_i, v_j, v_k, and v_l, respectively. If any of the three conditions are violated, the construction of possible C_{ijk} can be discarded in advance; otherwise, we perform Eqs. (1)–(3) to calculate the center (a_{ijk}, b_{ijk}) and the radius $r_{i,jk}$ of the possible circle C_{ijk}.

From the constructed possible C_{ijk}, we proceed to use the second and third evidences to determine whether C_{ijk} is a candidate circle or not. The second evidence is to check whether v_l is close to C_{ijk} enough or not. If the second evidence is positive, we further proceed to adopt the third evidence; otherwise, we

Author's personal copy

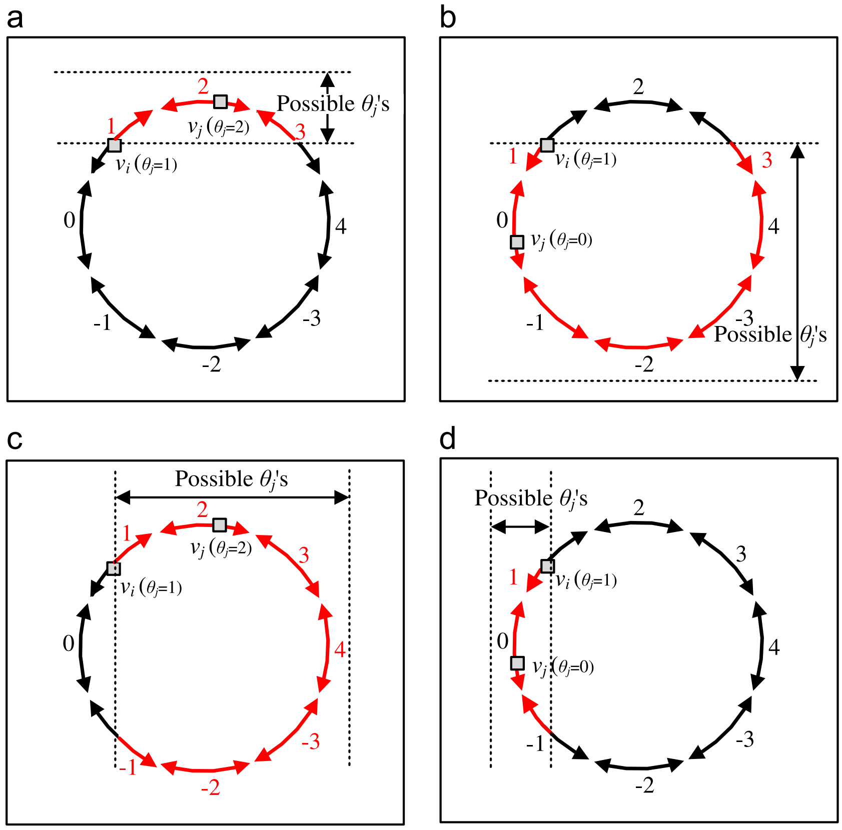


Fig. 7. Valid gradient direction set of θ_j when given v_i and θ_i. (a) and (b) Case_1. (c) and (d) Case_2.

Table 2
Determination of valid gradient direction set.

θ_i	$\Theta_V(y_i,y_j,\theta_i)$		$\Theta_H(x_i,x_j,\theta_i)$	
	$y_j < y_i$	$y_j \geq y_i$	$x_j < x_i$	$x_j \geq x_i$
0	0,1,2,3,4	$0,-1,-2,-3,4$	0	$0,\pm1,\pm2,\pm3,4$
1	1,2,3	$0,\pm1,-2,\pm3,4$	$0,\pm1$	$\pm1,\pm2,\pm3,4$
2	2	$0,\pm1,\pm2,\pm3,4$	$0,\pm1,\pm2$	$\pm2,\pm3,4$
3	1,2,3	$0,\pm1,-2,\pm3,4$	$0,\pm1,\pm2,\pm3$	$\pm3,4$
4	0,1,2,3	$0,-1,-2,-3,4$	$0,\pm1,\pm2,\pm3,4$	4
-1	$0,\pm1,2,\pm3,4$	$-1,-2,-3$	$0,\pm1$	$\pm1,\pm2,\pm3,4$
-2	$0,\pm1,\pm2,\pm3,4$	-2	$0,\pm1,\pm2$	$\pm2,\pm3,4$
-3	$0,\pm1,2,\pm3,4$	$-1,-2,-3$	$0,\pm1,\pm2,\pm3$	$\pm3,4$

stop the circle detection job and randomly select the next four edge pixels. Suppose the first and second evidences are positive for the four edge pixels v_i, v_j, v_k, and v_l, then we examine the third evidence: the gradient directions of the four edge pixels should point to the center of C_{ijk}. The ideal gradient direction, θ_i^*, θ_j^*, θ_k^*, and θ_l^*, can be calculated by

$$\theta_s^* = \tan^{-1}\frac{y_s - b_{ijk}}{x_s - a_{ijk}}, \quad s \in \{i,j,k,l\} \tag{7}$$

Quantizing θ_i^*, θ_j^*, θ_k^*, and θ_l^*, each quantized gradient direction is in $\{-3,-2,-1,0,1,2,3,4\}$. Based on the eight quantized ideal gradient directions, if the following four conditions, $\theta_i = \theta_i^*$, $\theta_j = \theta_j^*$, $\theta_k = \theta_k^*$, and $\theta_l = \theta_l^*$, hold, the third evidence is said to be positive. When three evidences are positive, C_{ijk} is promoted to be a promising candidate circle and we run the voting process to determine whether C_{ijk} is a true circle or not; otherwise, the possible circle C_{ijk} is discarded and we randomly select next four edge pixels from V and repeat the above process until all circles have been found.

In this paragraph, based on ten test images in Fig. 1, we show some experimental data to demonstrate the computation-saving effect of the proposed multiple-evidence-based sampling strategy. Table 3 illustrates the number of possible circles and candidate circles, N_P' and N_C', respectively, by using the proposed sampling strategy. When compared with Table 1, we observe that the first evidence used in the proposed strategy can discard 73% $(=N_P - N_P'/N_P = (35\,976 - 9791)/35\,976)$ of invalid possible circles in the RCD and it indicates that a lot of unnecessary calculations in Eqs. (1)–(3) can be avoided. Besides that, based on the second and third evidences, the number of candidate circles identified by our proposed strategy is only 4% $(=N_C'/N_C = 52/1373)$ of that in the RCD and it results in a significant computation-saving effect since the proposed strategy could eliminate 96% of voting time required in the RCD.

It is natural to combine the proposed multiple-evidence-based sampling strategy with any existing voting schemes, such as the voting scheme in the RCD [3] or the LUT-based voting scheme [6], to constitute faster circle detection methods. Let T_f denote the number of failures that we can tolerate and T_r be the ratio threshold. The two threshold values will be discussed in Section 5. Note that an edge pixel is said to lie on C_{ijk} provided the distance between the fourth edge pixel and C_{ijk} is less than or equal to one pixel. Our proposed multiple-evidence-based RCD is shown below:

1 Input test image I.
2 Perform Sobel edge detector on I to obtain the set of edge pixels, V.
3 Calculate gradient direction of each edge pixel $v_p \in V$, θ_p.
4 $f \leftarrow 0$

Author's personal copy

Table 3
Number of possible circles and candidate circles constructed by the proposed sampling strategy.

	Coin	Cake	Insulator	Gobang	Plates	Logo	Speaker	Stability-ball	Ball	Swatch	Average
N'_P	10810	12507	13500	11370	16317	7307	4454	7900	7495	6247	9791
N'_C	87	85	87	29	58	13	16	62	27	58	52

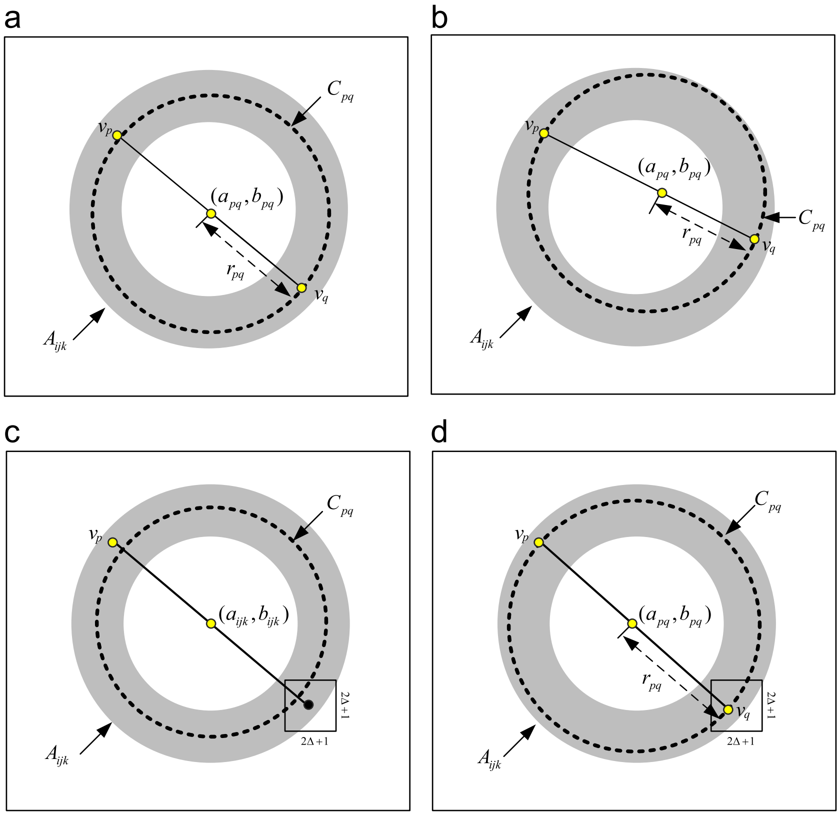


Fig. 8. Depiction of our proposed refinement strategy. (a) and (b) Create new circle C_{pq} by pairing edge pixel v_p and the other edge pixel v_q in A_{ijk}. (c) For edge pixel v_p, circumscribe a square region of size $(2\Delta+1)\times(2\Delta+1)$ on the opposite side of v_p. (d) Create new circle C_{pq} by pairing v_p and the other edge pixel v_q in the $(2\Delta+1)\times(2\Delta+1)$ square region.

while $f \le T_f$ **do**
  Randomly select four edge pixels, v_i, v_j, v_k, and v_l, from V.
  if $\theta_j \in \Theta(v_i,\theta_i,v_j)$, $\theta_k \in \Theta(v_i,\theta_i,v_k) \cap \Theta(v_j,\theta_j,v_k)$, and $\theta_l \in \Theta(v_i,\theta_i,v_l) \cap \Theta(v_j,\theta_j,v_l) \cap \Theta(v_k,\theta_k,v_l)$ **then**
    Calculate (a_{ijk},b_{ijk}) and r_{ijk} for constructing the possible circle C_{ijk} by Eqs. (1)–(3).
    if v_l is lying on C_{ijk} **then**
      Calculate θ_i^*, θ_j^*, θ_k^*, and θ_l^* by Eq. (7) and then quantize them.
      if $\theta_i^* = \theta_i$, $\theta_j^* = \theta_j$, $\theta_k^* = \theta_k$, and $\theta_l^* = \theta_l$ **then**
        Perform voting process to count the number of edge pixels lying on C_{ijk} and save the counted number in N_V.
        if $N_V \ge 2\pi r_{ijk} \times T_r$ **then**
          C_{ijk} is a true circle.
          $f \leftarrow 0$.
        else
          $f \leftarrow f+1$.
      else
        $f \leftarrow f+1$.
    else
      $f \leftarrow f+1$.
  else
    $f \leftarrow f+1$.

4. The proposed new refinement strategy

In this section, a novel refinement strategy is proposed to solve the bias problem mentioned in Section 2. From the detected circle C_{ijk} and the bandwidth Δ, an annulus A_{ijk} defined in Section 2.2 is constructed to cover the set of edge pixels V' as the input of our proposed refinement strategy. In Lee et al.'s refinement strategy, it takes $O(|V'|^2)$-time to create $O(|V'|)$ new circles by Eqs. (1)–(3), and then run a voting process on each created new circle. Quite different from Lee et al.'s strategy, our proposed refinement strategy only needs $O(|V'|)$-time to create $O(|V'|)$ new circles from V' in a more simple way, especially omitting the extra voting process which is required in Lee et al.'s method.

The main concept of the proposed refinement strategy is depicted in Fig. 8. As shown in Fig. 8(a) and (b), we take two edge pixels v_p and v_q from V' and crate a new circle C_{pq} denoted by a dash-lined circle where the midpoint of segment $\overline{v_p v_q}$ is the center of C_{pq}. By this way, we can construct $O(|V'|^2)$ total circles and the circles with the same center and radius are collected as a group. Among the collected groups, the three parameters of the largest group is selected as the initial parameters to be refined later.

In order to decrease the computation overhead, we can reduce the number of the created new circles by pairing two edge pixels which are located on the opposite sides of C_{ijk} each other. As shown in Fig. 8(c), for each edge pixel v_p in V', we circumscribe a

Author's personal copy

square region of size $(2\Delta+1)\times(2\Delta+1)$ centered at the pixel symmetric to v_p about the center of C_{ijk} and let V'_{OS} denote the set of edge pixels in the square region. The setting of the bandwidth Δ will be discussed in Section 5. Since each v_p is paired with each one in V'_{OS} (see Fig. 8(d)), it totally creates $|V'_{OS}|\times|V'|$ new circles. Without loss of generality, the cardinality of V', $|V'_{OS}|$ is bounded by a constant c due to the fact $|V'_{OS}|\ll|V'|$. Consequently, the proposed refinement strategy may have a chance to take $O(|V'|)\ (=|V'_{OS}|\times|V'|=c\times|V'|)$ time to refine each detected circle. In what follows, we shall detail our refinement strategy.

Naturally, we adopt a 3-D accumulator array $A[*,*,*]$ to save these $O(|V'|)$ constructed new circles and count the number of circles of each group although in the next paragraph, a new memory reduction scheme will be presented to reduce it to constant size. Each element of A is set to 0 initially. For each edge pixel $v_p=(x_p,y_p)$ in V', $0\le p<|V'|$, its corresponding position on the opposite side of C_{ijk} is calculated by

$$x_p^{OS}=a_{ijk}-(x_p-a_{ijk}) \tag{8}$$

$$y_p^{OS}=b_{ijk}-(y_p-b_{ijk}) \tag{9}$$

From the position (x_p^{OS},y_p^{OS}), a square region S is circumscribed and its top-left corner and bottom-right corner are $(x_p^{OS}-\Delta,y_p^{OS}-\Delta)$ and $(x_p^{OS}+\Delta,y_p^{OS}+\Delta)$, respectively. After collecting all edge pixels in S to constitute the set V'_{OS}, our refinement strategy pairs v_p and each edge pixel $v_q=(x_q,y_q)$ in V'_{OS}, $0\le q<|V'_{OS}|$, and to create a segment $\overline{v_p v_q}$. The created segment is used to determine a new circle C_{pq} with center (a_{pq},b_{pq}) and radius r_{pq} where

$$a_{pq}=\frac{x_p+x_q}{2} \tag{10}$$

$$b_{pq}=\frac{y_p+y_q}{2} \tag{11}$$

and

$$r_{pq}=\frac{\sqrt{(x_p-x_q)^2+(y_p-y_q)^2}}{2} \tag{12}$$

For each C_{pq}, the assignment statement $A[a_{pq},b_{pq},r_{pq}]=A[a_{pq},b_{pq},r_{pq}]+1$ is performed, and it means that the number of circles belonging to that group is increased by 1. After counting the number of circles in each group, the center (a_{ijk}^R,b_{ijk}^R) and the radius r_{ijk}^R of the refined circle C_{ijk}^R is determine by

$$\{a_{ijk}^R,b_{ijk}^R,r_{ijk}^R\}=\arg\max_{a,b,r}A[a,b,r] \tag{13}$$

To verify whether the refined C_{ijk}^R is better than the candidate circle in the RCD, C_{ijk}, or not, we count the number of edge pixels lying on C_{ijk} and C_{ijk}^R, say N_{V1} and N_{V2}, respectively, and decide that C_{ijk}^R is better than C_{ijk} if the condition $N_{V1}<N_{V2}$ holds; otherwise, C_{ijk} is better than C_{ijk}^R. Consequently, the better one of C_{ijk}^R and C_{ijk} is the final result of the proposed refinement strategy.

In order to reduce the memory required in the 3-D accumulator array, i.e. $O(N^3)$ memory where $N\times N$ denotes the image size, a constant-sized accumulator array $A_r[*,*,*]$ is used in our refinement strategy and the memory size is dependent on the bandwidth $\Delta(\ll N)$ between C_{ijk} and A_{ijk}. For each determined C_{pq}, we first calculate $\Delta a_{pq}=a_{pq}-a_{ijk}$, $\Delta b_{pq}=b_{pq}-b_{ijk}$, and $\Delta r_{pq}=r_{pq}-r_{ijk}$. Since only edge pixels lying on A_{ijk} are considered in the refinement process, the values of Δa_{pq}, Δb_{pq}, and Δr_{pq} are ranged from $-\Delta$ to Δ. Therefore the size of the reduced accumulator array $A_r[*,*,*]$ is only $(2\Delta+1)^3\ (=O(\Delta^3))$. Based on the reduced accumulator array $A_r[*,*,*]$, the original statement $A[a_{pq},b_{pq},r_{pq}]=A[a_{pq},b_{pq},r_{pq}]+1$ is replaced by $A_r[\Delta a_{pq}+\Delta,\Delta b_{pq}+\Delta,\Delta r_{pq}+\Delta]=A_r[\Delta a_{pq}+\Delta,\Delta b_{pq}+\Delta,\Delta r_{pq}+\Delta]+1$ and it leads to a significant memory-saving effect. The whole refinement algorithm is presented below.

```
Input the initial detected circle C_ijk by the RCD.
Initialize each entry of A_r[*,*,*] to 0.
Construct an annulus A_ijk concentric with C_ijk and the two
  radii of A_ijk are r_ijk − Δ and r_ijk + Δ.
Collect all edge pixels within A_ijk to obtain the set V'.
for each pixel v_p ∈ V' do
   Determine (x_p^OS, y_p^OS) by Eq. (8).
   Circumscribe a square region S with the top-left corner
  (x_p^OS − Δ, y_p^OS − Δ) and the bottom-right corner
  (x_p^OS + Δ, y_p^OS + Δ).
```

Table 4
Execution-time performance comparison in the RCD, the LRCD, the GRCD, and the GLRCD in terms of milliseconds.

Image	RCD	LRCD	GRCD	GLRCD
Coin	103	62	28	25
Cake	80	53	26	24
Insulator	92	48	25	24
Gobang	112	66	30	27
Plates	180	114	48	45
Logo	194	68	54	48
Speaker	106	51	30	29
Stability-ball	130	76	34	32
Ball	95	52	28	26
Swatch	132	48	46	40
Average time	122	64	35	32
Execution-time improvement ratio			$71\%\left(=\frac{122-35}{122}\right)$	$50\%\left(=\frac{64-32}{64}\right)$

Table 5
Average differences between the parameters of each circle detected by the HT and that detected by the RCD, the LRCD, and the GRCD.

Image	RCD		LRCD		GRCD		GLRCD	
	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr
Coin	(0.65,0.62)	0.50	(0.69,0.66)	0.69	(0.63,0.57)	0.54	(0.74,0.63)	0.67
Cake	(0.65,0.78)	0.67	(0.67,0.88)	0.71	(0.50,0.70)	0.59	(0.55,0.76)	0.62
Insulator	(0.90,1.06)	0.66	(0.84,1.10)	0.73	(0.83,0.97)	0.63	(0.88,1.06)	0.67
Gobang	(0.96,0.92)	0.86	(1.04,1.18)	0.97	(0.85,0.95)	0.78	(0.89,1.01)	0.77
Plates	(0.77,0.88)	0.77	(0.84,0.90)	0.75	(0.69,0.82)	0.65	(0.73,0.84)	0.68
Logo	(0.45,0.51)	0.18	(0.49,0.52)	0.11	(0.33,0.54)	0.16	(0.49,0.57)	0.10
Speaker	(0.43,0.43)	0.52	(0.43,0.54)	0.48	(0.27,0.29)	0.47	(0.27,0.10)	0.40
Stability-ball	(0.54,0.74)	0.55	(0.67,0.75)	0.54	(0.63,0.73)	0.58	(0.70,0.77)	0.56
Ball	(0.46,0.45)	0.60	(0.51,0.48)	0.47	(0.51,0.49)	0.57	(0.64,0.67)	0.68
Swatch	(0.61,0.98)	0.98	(0.91,1.43)	1.31	(0.69,0.83)	0.95	(0.86,1.53)	1.53
Average	(0.64,0.74)	0.63	(0.71,0.84)	0.68	(0.59,0.69)	0.59	(0.68,0.69)	0.67

Author's personal copy


```
Collect edge pixels within S to obtain the set V'_OS.
for each pixel v_q ∈ V_OS' do
    Determine the center (a_pq, b_pq) and the radius r_pq of the
    new circle C_pq by Eqs. (10)–(12).
    Δa_pq ← a_pq − a_ijk, Δb_pq ← b_pq − b_ijk, Δr_pq ← r_pq − r_ijk.
    A_r[Δa_pq + Δ, Δb_pq + Δ, Δr_pq + Δ] ← A_r[Δa_pq + Δ,
    Δb_pq + Δ, Δr_pq + Δ] + 1
Determine the center (a^R_ijk, b^R_ijk) and the radius r^R_ijk of the
refined circle C^R_ijk by Eq. (13).
Count the number of edge pixels lying on C_ijk and C^R_ijk and
save the counted numbers to N_V1 and N_V2, respectively.
if N_V1 > N_V2 then
    Output C_ijk as the final result.
else
    Output C^R_ijk as the final result.
```

5. Experimental results

In this section, some experimental results are demonstrated to show the execution-time and accuracy advantages of our proposed new multiple-evidence-based sampling strategy and new refinement strategy. All concerned experiments are performed on the Intel CPU E8400 Processor with 3.0 GHz and 2 GB RAM. The operating system adopted is MS-Windows XP and the programming environment is Borland C++ Builder 6.0. To evaluate the accuracy of each concerned method, the traditional HT [9,13] is run on ten test images and the three parameters of each detected circle are taken as the ideal center and radius of the corresponding test image. To meet a high accuracy requirement, the x-coordinate and the y-coordinate of center and the length of radius are quantized to one pixel precision in the traditional HT.

Before evaluating the execution-time and accuracy performance of the concerned circle detection methods, we first discuss two thresholds, T_f and T_r, used in our proposed sampling strategy and the bandwidth Δ used in our proposed refinement strategy. The first threshold T_f is related to the ratio of edge pixels lying on circles over that of total edge pixels. For each test images in Fig. 1, the number of

Table 8
Execution-time performance comparison between the GRCD-R and the GLRCD-R in terms of milliseconds.

Image	GRCD-R	GLRCD-R
Coin	36	36
Cake	31	30
Insulator	35	35
Gobang	48	48
Plates	71	71
Logo	88	86
Speaker	38	36
Stability-ball	50	47
Ball	38	36
Swatch	86	83
Average	52	51

Table 6
Execution-time performance comparison in Lee et al.'s refinement strategy, the IRHT, the RCD-R, and the LRCD-R in terms of milliseconds.

Image	Lee et al.'s	IRHT	RCD-R	LRCD-R
Coin	160	577	115	83
Cake	134	362	91	57
Insulator	208	325	102	80
Gobang	168	1255	120	67
Plates	591	1015	197	133
Logo	650	3279	224	103
Speaker	186	869	114	63
Stability-ball	389	449	112	66
Ball	304	455	109	59
Swatch	998	3100	164	86
Average	379	1169	135	80

Table 9
Average differences between the parameters of each circle detected by the HT and that detected by the GRCD-R and the GLRCD-R.

Image	GRCD-R		GLRCD-R	
	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr
Coin	(0.55,0.34)	0.47	(0.54,0.37)	0.50
Cake	(0.43,0.67)	0.46	(0.47,0.67)	0.51
Insulator	(0.31,0.57)	0.45	(0.39,0.64)	0.46
Gobang	(0.38,0.43)	0.50	(0.40,0.45)	0.51
Plates	(0.50,0.54)	0.52	(0.53,0.51)	0.52
Logo	(0.02,0.40)	0.06	(0.02,0.43)	0.09
Speaker	(0.26,0.01)	0.46	(0.28,0.07)	0.37
Stability-ball	(0.42,0.65)	0.51	(0.43,0.71)	0.48
Ball	(0.28,0.35)	0.50	(0.23,0.45)	0.59
Swatch	(0.41,0.63)	0.71	(0.45,0.86)	0.96
Average	(0.36,0.46)	0.46	(0.37,0.52)	0.50

Table 7
Average differences between the parameters of each circle detected by the HT and that detected by Lee et al.'s refinement strategy, the IRHT, the RCD-R, and the LRCD-R.

Image	Lee et al.'s		IRHT		RCD-R		LRCD-R	
	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr
Coin	(0.43,0.55)	0.42	(0.45,0.42)	0.32	(0.49,0.36)	0.46	(0.61,0.44)	0.58
Cake	(0.47,0.50)	0.41	(0.48,0.58)	0.50	(0.38,0.66)	0.45	(0.42,0.69)	0.50
Insulator	(0.54,0.67)	0.37	(0.35,0.74)	0.28	(0.32,0.60)	0.44	(0.34,0.60)	0.46
Gobang	(0.66,0.66)	0.35	(0.52,0.50)	0.47	(0.45,0.50)	0.55	(0.47,0.41)	0.60
Plates	(0.44,0.56)	0.45	(0.48,0.64)	0.42	(0.55,0.50)	0.57	(0.55,0.55)	0.57
Logo	(0.19,0.48)	0.06	(0.12,0.48)	0.02	(0.02,0.40)	0.20	(0.03,0.40)	0.23
Speaker	(0.17,0.53)	0.31	(0.18,0.29)	0.40	(0.26,0.01)	0.46	(0.28,0.08)	0.38
Stability-ball	(0.61,0.74)	0.43	(0.43,0.65)	0.49	(0.43,0.69)	0.45	(0.44,0.69)	0.49
Ball	(0.35,0.20)	0.28	(0.39,0.43)	0.39	(0.32,0.36)	0.44	(0.35,0.38)	0.45
Swatch	(0.47,0.78)	0.54	(0.25,0.73)	0.78	(0.40,0.65)	0.73	(0.42,0.82)	0.95
Average	(0.43,0.58)	0.36	(0.37,0.55)	0.41	(0.36,0.47)	0.48	(0.39,0.51)	0.52

Author's personal copy

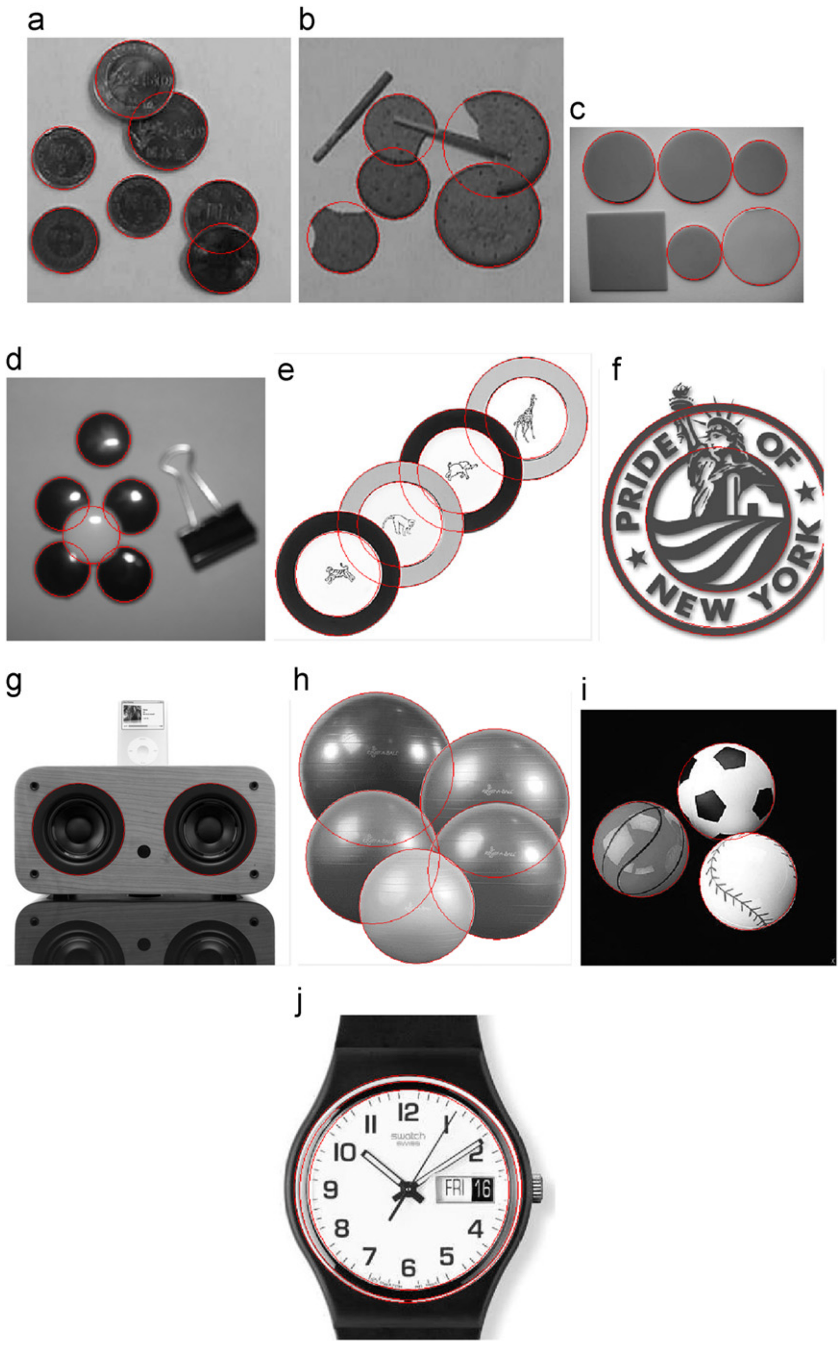


Fig. 9. Detected circles by the GRCD-R. (a) Coin image. (b) Cake image. (c) Insulator image. (d) Gobang image. (e) Plates image. (f) Logo image. (g) Speaker image. (h) Stability-ball image. (i) Ball image. (j) Swatch image.

circles is ranged from 2 to 7 and the ratio of edge pixels lying on circles is ranged from 0.16 to 0.62. Empirically, we set T_f to 16 000 and it is applicable to ten test images. The second threshold T_r is dependent on the completeness degree of the circle and in ten test images, the circle completeness degree is ranged from 60% to 100%. Considering the detected circle is a digital zone, instead of setting $T_r = 0.6$, we set $T_r = 0.8$ in our implementation. The bandwidth Δ used in the proposed refinement strategy is dependent on the radius of each circle which is ranged from 30 to 147 for ten test images. Empirically, we set Δ to 5% of the maximal radius, i.e. $\Delta = \text{round}(147 \times 5\%) = 7$.

To illustrate the execution-time improvement power of the proposed sampling strategy, we first plug the proposed sampling strategy into the RCD [3]. To broaden the comparison, we also apply the proposed sampling strategy to the LUT-based RCD (LRCD) [6]. For convenience, two modified circle detection methods are called the GRCD and the GLRCD. Table 4 indicates that on average, the GRCD and the GLRCD have 71% and 50% execution-time improvement ratios when compared to the RCD and the LRCD, respectively. Because the proposed sampling strategy can discard a large amount of invalid possible and candidate circles involved in the RCD and the LRCD. The average differences between the parameters of each circle detected by the HT and that detected by the RCD, the LRCD, the GRCD, and the GLRCD are given in Table 5. In Table 5, $(\Delta a, \Delta b)$ and Δr denote the average differences between the center and radius of each circle detected by the HT and that in any one of the concerned four methods. Table 5 shows that the accuracy of the proposed sampling strategy is very close to that in the RCD and the LRCD.

Author's personal copy

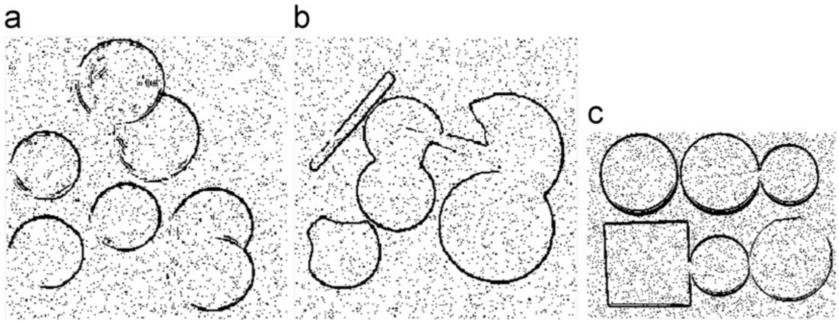


Fig. 10. The noisy edge maps. (a) Coin image. (b) Cake image. (c) Insulator image.

Table 10
Execution-time performance comparison in the GRCD-R and the GLRCD-R for the noisy edge maps in terms of milliseconds.

Image	GRCD-R	GLRCD-R
Coin	88	77
Cake	81	81
Insulator	98	93
Gobang	115	114
Plates	188	187
Logo	202	203
Speaker	68	65
Stability-ball	117	109
Ball	99	95
Swatch	172	160
Average	123	118

Table 11
Average differences between the parameters of each circle detected by the HT and that in each noisy edge map detected by the GRCD-R and the GLRCD-R.

Image	GRCD-R		GLRCD-R	
	$(\Delta a,\Delta b)$	Δr	$(\Delta a,\Delta b)$	Δr
Coin	(0.74,0.65)	0.91	(0.77,0.77)	0.94
Cake	(0.60,0.72)	0.21	(0.66,0.83)	0.72
Insulator	(0.44,0.68)	0.64	(0.58,0.82)	0.52
Gobang	(0.60,0.96)	0.75	(0.65,0.99)	0.76
Plates	(0.76,0.51)	0.75	(0.78,0.55)	0.80
Logo	(0.20,0.47)	0.33	(0.22,0.22)	0.66
Speaker	(0.35,0.08)	0.51	(0.45,0.10)	0.40
Stability-ball	(0.57,0.08)	0.65	(0.64,0.81)	0.63
Ball	(0.42,0.51)	0.70	(0.23,0.55)	0.61
Swatch	(0.50,0.70)	0.80	(0.70,1.05)	1.10
Average	(0.52,0.54)	0.55	(0.57,0.67)	0.71

To evaluate the performance of the proposed refinement strategy, we combine our proposed refinement strategy with the RCD and the LRCD to obtain two modified circle detection methods, called the RCD-R and the LRCD-R, respectively. Table 6 shows the execution-time requirement for Lee et al.'s refinement strategy [23], the IRHT proposed by Lu and Tan [24], the RCD-R, and the LRCD-R. Table 6 shows that both RCD-R and LRCD-R take less execution-time when compared to the IRHT and Lee et al.'s refinement strategy and it confirms the computation advantage of our proposed refinement strategy. Table 7 shows the average differences between the parameters of each circle detected by the HT and that detected by the concerned four methods. In [24], each time, Lu and Tan randomly sample five edge pixels to determine the parameters of an ellipse, and in order to apply the IRHT to circle detection, each time, we randomly sample three edge pixels to determine the parameters of a circle. From Tables 5 and 7, we observe that the accuracies of the RCD-R and the LRCD-R are very close to that of the IRHT and Lee et al.'s refinement strategy; two tables indicate that the accuracies of the RCD and the LRCD have been improved by our proposed refinement strategy.

Combining our proposed multiple-evidence-based sampling strategy and refinement strategy with the RCD and the LRCD, the proposed two modified versions, the GRCD-R and the GLRCD-R, illustrate the execution-time and accuracy advantages in Tables 8 and 9, respectively. From Tables 6–9, the proposed GRCD-R and GLRCD-R have better execution-time performance when compared to the RCD-R and the LRCD-R, and their accuracies are very close to Lee et al.'s refinement strategy. Fig. 9 illustrates the resultant circles detected by using the GRCD-R. These detected circles reveal that the GRCD-R can detect circles efficiently. Note that the circles detected by the other concerned methods are similar to those in Fig. 9, so we only demonstrate the detected results of the GRCD-R for saving space of the context.

Finally, in order to demonstrate the robustness of our proposed sampling strategy and refinement strategy, we add noises to the edge maps of test images and run the GRCD-R and the GLRCD-R on these noisy test edge maps. It is known that the edge map of each test image contains $|V|$ edge pixels. We sprinkle $|V|$ edge pixels whose gradient directions are randomly given on the noise-free edge map so that the number of noisy edge pixels over that of original edge pixels is 100%; Fig. 10(a)–(c) are noisy edge maps of Fig. 1(a)–(c), respectively. After setting T_f to 48 000 and running the GRCD-R and the GLRCD-R on ten noisy edge maps, the execution-time and accuracy performance comparisons are shown in Tables 10 and 11, respectively. From Tables 8–11, although the execution-time and accuracy performance are degraded for noisy edge maps, the resultant performance of the GRCD-R and GLRCD-R is still better than that obtained by running RCD and LRCD on noise-free edge maps (see Tables 4 and 5).

6. Conclusion

We have presented the proposed new multiple-evidence-based sampling strategy and refinement strategy to improve both the execution-time performance and the detection accuracy for some existing randomized circle detection methods. First, from the computation overhead analysis of the RCD's sampling strategy, an efficient multiple-evidence-based sampling strategy is presented to alleviate this computation overhead problem. By using the proposed three evidences, the execution-time performance can be improved significantly since a large amount of possible circles and candidates, which will not be promoted to true circles eventually, can be

Author's personal copy

discarded in advance. To solve the bias problem existed in the RCD, a fast linear-time refinement strategy is presented to enhance the accuracy. Specially, a constant-sized accumulator array is proposed to realize the voting process on a smaller set of edge pixels. Based on ten test images, experimental results demonstrate that under the similar accuracy, the proposed sampling strategy significantly improves the execution-time performance of the RCD and the GLRCD. Experimental results also demonstrate that the bias problem in the RCD can be overcome by using our proposed refinement strategy. When compared to the IRHT and Lee et al.'s refinement strategy, our proposed refinement strategy provides a considerable execution-time improvement under the similar accuracy.

References

[3] T.C. Chen, K.L. Chung, An efficient randomized algorithm for detecting circles, Computer Vision and Image Understanding 83 (2) (2001) 172–191.
[4] S.H. Chiu, J.J. Liaw, An effective voting method for circle detection, Pattern Recognition Letters 26 (1) (2005) 121–133.
[5] H.D. Cheng, Y. Guo, Y. Zhang, A novel Hough transform based on eliminating particle swarm optimization and its applications, Pattern Recognition 42 (9) (2009) 1956–1969.
[6] K.L. Chung, Y.H. Huang, Speed up the computation of randomized algorithms for detecting lines, and ellipses using novel tuning-and LUT-based voting platform, Applied Mathematics and Computation 190 (1) (2007) 132–149.
[7] E.R. Davies, Truncating the Hough transform parameter space can be beneficial, Pattern Recognition Letters 24 (1–3) (2003) 129–135.
[8] E.R. Davies, Machine Vision: Theory Algorithms, Practicalities, third ed., Morgan Kaufmann, San Fransisco, CA, 2004.
[9] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1) (1972) 11–15.
[10] D.A. Forsyth, Computer Vision: A Modern Approach, Prentice-Hall, New Jersey, 2002.
[11] R. Gonzalez, R. Woods, Digital Image Processing, Addison Wesley, New York, 1992.
[12] C.T. Ho, L.H. Chen, A fast ellipse/circle detector using geometric symmetry, Pattern Recognition 28 (1) (1995) 117–124.
[13] P.V.C. Hough, Method and means for recognizing complex patterns, US Patent # 3,069,654, 1962.
[14] P.V.C. Hough, Method and Means for Recognizing Complex Patterns, Addison Wesley, New York, 1992.
[16] J. Illingworth, J. Kittler, The adaptive Hough transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 9 (5) (1987) 690–698.
[17] J. Illingworth, J. Kittler, Survey: survey of the hough transforms, Graphics and Image Processing 44 (1) (1988) 87–116.
[18] D. Ioannou, W. Huda, A.F. Laine, Circle recognition through a 2D Hough transform and radius histogramming, Image and Vision Computing 17 (1) (1999) 15–26.
[19] R. Jain, R. Kasturi, B.G. Schunck, Machine Vision, McGraw-Hill, Columbus, OH, 1995.
[20] H.S. Kim, J.H. Kim, A two-step circle detection from the intersecting chords, Pattern Recognition Letters 22 (6–7) (2001) 787–798.
[21] C. Kimme, D. Ballard, J. Sklansky, Finding circles by an array of accumulator, Communications of the ACM 18 (2) (1975) 120–122.
[22] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognition 24 (4) (1991) 303–316.
[23] J.R.J. Lee, M.L. Smith, L.N. Smith, P.S. Midha, Robust and efficient automated detection of tooling defects in polished stone, Computers in Industry 56 (8) (2005) 787–801.
[24] W. Lu, J. Tan, Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT), Pattern Recognition 41 (4) (2008) 1268–1279.
[25] L. Xu, E. Oja, P. Kultanan, A new curve detection method: randomized Hough transform (RHT), Pattern Recognition Letters 11 (5) (1990) 331–338.
[26] L. Xu, E. Oja, Randomized Hough transform (RHT): basic mechanisms, algorithms, and computational complexities, Computer Vision Graphic Image Process: Image Understanding 57 (2) (1993) 131–154.
[27] R.K.K. Yip, P.K.S. Tam, D.N.K. Leung, Modification of Hough transform for circles and ellipses detection using a 2-dimensional array, Pattern Recognition 25 (9) (1992) 1007–1022.
[28] A. Ylä-Jääski, N. Kiryati, Adaptive termination of voting in probabilistic Hough transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (9) (1994) 911–915.

Kuo-Liang Chung received his B.S., M.S., and Ph.D. degrees in Computer Science and Information Engineering from the National Taiwan University in 1982, 1984, and 1990, respectively. He was a visiting scholar at the University of Washington in the summer of 1999. During 2003–2006, he was the head of the Department of Computer Science and Information Engineering at the National Taiwan University of Science and Technology. He was promoted to University Chair Professor in 2009. During 1996–1998, he was the executive editor of Journal of the Chinese Institute of Engineers. Dr. Chung received the Distinguished Research Award from the National Science Council in 2004 and received the best paper award from the Society of Computer Vision, Graphics, and Image Processing, Taiwan, in 2007. He is now a senior member of IEEE and a fellow of IET. His research interests include shape analysis, pattern recognition, and video coding.

Yong-Huai Huang received the B.S. degree in Information Management from Aletheia University, Tamsui, Taipei, Taiwan, and the M.S. and Ph.D. degrees in Computer Science and Information Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan. He is now an assistant professor in the Institute of Computer and Communication Engineering at Jinwen University of Science and Technology, Hsin-Tieny, Taipei, Taiwan. His research interests include image processing and compression, and algorithms.

Shi-Ming Shen received the B.S. and the M.S. degrees in Computer Science and Information Engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan. He is now an engineer in the TRI Innovation Inc., Taipei, Taiwan. His research interests include image processing and multimedia applications.

Andrey S. Krylov (born 1956) received the Ph.D. degree in Computational Mathematics and Cybernetics from the Lomonosov Moscow State University (MSU) in 1983. Currently he is an associate professor and head of the Laboratory of Mathematical Methods of Image Processing at MSU. His main research interests lie in mathematical methods of multimedia data processing.

Dmitry V. Yurin, Ph.D., is a senior scientist at Institute of Computing for Physics and Technology and at Laboratory of Mathematical Methods of Image Processing; is a chair of Mathematical Physics, Faculty of Computational Mathematics and Cybernetics at MSU. His research interests include computational methods for image processing.

Ekaterina V. Semeikina is 5th-year student at Laboratory of Mathematical Methods of Image Processing, Chair of Mathematical Physics, Faculty of Computational Mathematics and Cybernetics at MSU. Her research interests include computational methods for image processing.