Computer Vision and Image Understanding **83,** 172–191 (2001)
doi:10.1006/cviu.2001.0923, available online at http://www.idealibrary.com on IDEAL®

An Efficient Randomized Algorithm for Detecting Circles

Teh-Chuan Chen and Kuo-Liang Chung[1]

Department of Information Management, Institute of Computer Science and Information Engineering, National Taiwan University of Science and Technology, No. 43, Section 4, Keelung Road, Taipei, Taiwan 10672, Republic of China

Received May 10, 2000; accepted April 19, 2001

Detecting circles from a digital image is very important in shape recognition. In this paper, an efficient randomized algorithm (RCD) for detecting circles is presented, which is not based on the Hough transform (HT). Instead of using an accumulator for saving the information of the related parameters in the HT-based methods, the proposed RCD does not need an accumulator. The main concept used in the proposed RCD is that we first randomly select four edge pixels in the image and define a distance criterion to determine whether there is a possible circle in the image; after finding a possible circle, we apply an evidence-collecting process to further determine whether the possible circle is a true circle or not. Some synthetic images with different levels of noises and some realistic images containing circular objects with some occluded circles and missing edges have been taken to test the performance. Experimental results demonstrate that the proposed RCD is faster than other HT-based methods for the noise level between the light level and the modest level. For a heavy noise level, the randomized HT could be faster than the proposed RCD, but at the expense of massive memory requirements.

Key Words: circle detection; Hough transform; randomized algorithm.

I. INTRODUCTION

Detecting circles from a digital image is very important in shape recognition [5]. Hough transform (HT) [11, 16] is the most well-known method for circle detection. Let (x, y) be an edge pixel on a circle with center coordinates (a, b) and radius r; then the circle can be expressed as

$$(x - a)^2 + (y - b)^2 = r^2. \tag{1}$$

[1] Corresponding author. E-mail: klchung@cs.ntust.edu.tw. Supported by NSC89-2218-E011-017.

1077-3142/01 $35.00

From Eq. (1), every edge pixel in the image can be mapped into a conic surface in the 3-dimensional (3-D) (a, b, r)-parameter space. Using the conventional HT (CHT) [6] for detecting circles, it requires a large amount of computing time to vote on such a 3-D array, i.e., a 3-D accumulator.

Several HT-based methods for detecting circles have been developed. One type of method decomposes the parameter space into many parameter spaces with lower dimension [22]. Another type of method uses the gradient information of each edge pixel to reduce the computing time or the requirement of the accumulator [1, 4, 14, 22]. A third type of method uses the geometry property in the circle to improve the performance [9, 10]. However, these three types of methods still need some amount of computing time and at least a 2-D accumulator array. Some other recent HT-based variants for detecting circles can be found in [12, 17]. These mainly focus on the robustness and accuracy in detecting circles.

Xu *et al*. [20, 21] presented a randomized Hough transform (RHT) which can significantly reduce the storage requirement and the computing time needed when compared to the CHT. In the RHT, three noncollinear edge pixels are used to solve the three parameters (a, b, r) in Eq. (1). That is, three noncollinear edge pixels are mapped into one point in the parameter space. Each time the RHT randomly chooses three edge pixels in the image with equal probability, and their corresponding mapped point in the parameter space is collected by voting on the accumulator implemented by an array or a link-list data structure [21]. Continue the above mapping and voting procedure until some cells in the accumulator have satisfactory scores and each of them represents a possible circle. For each possible circle, another evidence-collecting step follows, which counts the number of edge pixels lying on the possible circle to test whether the possible circle is the desired circle. Furthermore, when a circle is detected, the edge pixels lying on the circle are taken out of the set of current edge pixels which leads to speeding up the detection of the next circle. The circle detection work is performed iteratively until the given stopping criterion is reached. The readers are suggested to refer to the references [11, 13, 16, 23] on the comprehensive overview and comparison between the CHT and RHT.

In this paper, we present a new randomized circle detection algorithm called the RCD. The proposed non-HT-based RCD first randomly selects four edge pixels in the image and defines a distance criterion to determine whether there is a possible circle in the image. After finding a possible circle, we apply an evidence-collecting process to further determine whether the possible circle is the desired circle. Since the proposed RCD is not based on the technique of voting in the parameter space, it does not need extra accumulator storage. In fact, the memory requirements needed in the proposed RCD are only a few variables. The proposed RCD has some other advantages such as real-time speed and being robust to noise. Some synthetic images with different levels of noise and some realistic images that contain circular objects with some occluded circles and missing edges have been taken to justify the memory-saving and computational advantages of the proposed algorithm when compared to previous methods [6, 14, 20].

The remainder of this paper is organized as follows: Section II presents the proposed RCD. Some experimental results to confirm the memory-saving and computational advantages of the proposed RCD are demonstrated in Section III. In Section IV, two remarks about some other advantages of the proposed RCD are addressed. In Section V, some discussions about time complexities of the two randomized approaches, the proposed RCD and the RHT, are given. Finally, some conclusions are addressed in Section VI.

II. THE PROPOSED ALGORITHM: RCD

This section consists of four subsections. The first subsection describes the basic idea of the proposed RCD. The second subsection presents the distance criterion used to determine whether the selected four edge pixels lie on a possible circle or not. The third subsection describes how to check whether the possible circle is a true circle, i.e., the desired circle. The formal algorithm of the proposed RCD is listed in the fourth subsection.

A. Basic Idea

Let V denote the set of all edge pixels in the image. From the RHT [20], it is known that if we randomly pick three edge pixels from V, the three pixels are probably taken from a circle in the image. It is well known that each time three noncollinear pixels can exactly determine one circle. Suppose that many sets of three chosen edge pixels all come from the same circle; then it seems very probable that the circle is real. The RHT uses an accumulator to record all the instances of theses circles iteratively in order to find out a possible circle in the accumulator. In this paper, we modify the above idea [20] and propose a non-HT-based randomized algorithm for detecting circles. Our proposed algorithm randomly picks four edge pixels each time and defines a distance criterion to find a possible circle. This modification can lead to memory-saving and computational efficient effects which will be clarified later.

As shown in Fig. 1, four edge pixels can generally determine four circles. If the four randomly selected edge pixels all come from the same circle, then with high probability the circle seems to be real. When the four randomly selected edge pixels lie on the same circle, we refer to this circle as a possible circle. After a possible circle is found, we apply an evidence-collecting process to further verify whether the possible circle is a true circle or not.

B. Determining Possible Circle

In this section, we describe how to determine a possible circle according to the four selected edge pixels.

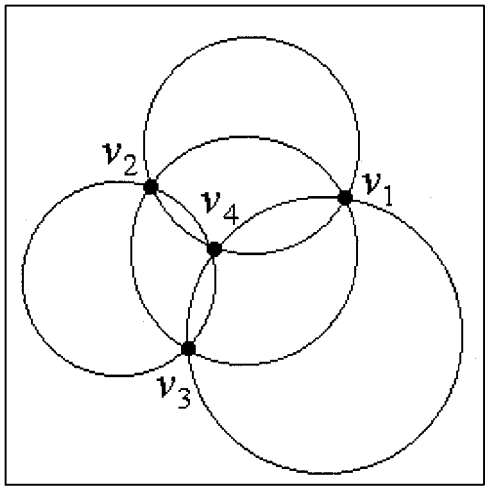

FIG. 1. Four circles determined by four edge pixels.

By returning to Eq. (1), we note that a circle can be written as

$$2xa + 2yb + d = x^2 + y^2, \tag{2}$$

where $d = r^2 - a^2 - b^2$. Let $v_i = (x_i, y_i)$, $i = 1, 2, 3$, be three edge pixels in the image. If v_1, v_2, and v_3 are noncollinear, they can exactly determine one circle (denoted by C_{123}) with center (a_{123}, b_{123}) with radius r_{123}. From Eq. (2) and the fact that the circle passes through the three pixels, we have

$$2x_1a_{123} + 2y_1b_{123} + d_{123} = x_1^2 + y_1^2,$$
$$2x_2a_{123} + 2y_2b_{123} + d_{123} = x_2^2 + y_2^2,$$

and

$$2x_3a_{123} + 2y_3b_{123} + d_{123} = x_3^2 + y_3^2,$$

where $d_{123} = r_{123}^2 - a_{123}^2 - b_{123}^2$. A representation of the above three equations in terms of matrix form yields

$$\begin{pmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ 2x_3 & 2y_3 & 1 \end{pmatrix} \begin{pmatrix} a_{123} \\ b_{123} \\ d_{123} \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ x_3^2 + y_3^2 \end{pmatrix}.$$

Applying Gaussian elimination, we have

$$\begin{pmatrix} 2x_1 & 2y_1 & 1 \\ 2(x_2 - x_1) & 2(y_2 - y_1) & 0 \\ 2(x_3 - x_1) & 2(y_3 - y_1) & 0 \end{pmatrix} \begin{pmatrix} a_{123} \\ b_{123} \\ d_{123} \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 - \left(x_1^2 + y_1^2\right) \\ x_3^2 + y_3^2 - \left(x_1^2 + y_1^2\right) \end{pmatrix}. \tag{3}$$

Since v_1, v_2, and v_3 are noncollinear, we have $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) \neq 0$.

By Cramer's rule, the center (a_{123}, b_{123}) can be obtained by

$$a_{123} = \frac{\begin{vmatrix} x_2^2 + y_2^2 - \left(x_1^2 + y_1^2\right) & 2(y_2 - y_1) \\ x_3^2 + y_3^2 - \left(x_1^2 + y_1^2\right) & 2(y_3 - y_1) \end{vmatrix}}{4((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))} \tag{4}$$

and

$$b_{123} = \frac{\begin{vmatrix} 2(x_2 - x_1) & x_2^2 + y_2^2 - \left(x_1^2 + y_1^2\right) \\ 2(x_3 - x_1) & x_3^2 + y_3^2 - \left(x_1^2 + y_1^2\right) \end{vmatrix}}{4((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))}. \tag{5}$$

After obtaining the center (a_{123}, b_{123}), the radius can be calculated by

$$r_{123} = \sqrt{(x_i - a_{123})^2 + (y_i - b_{123})^2} \tag{6}$$

for any $i = 1, 2, 3$.

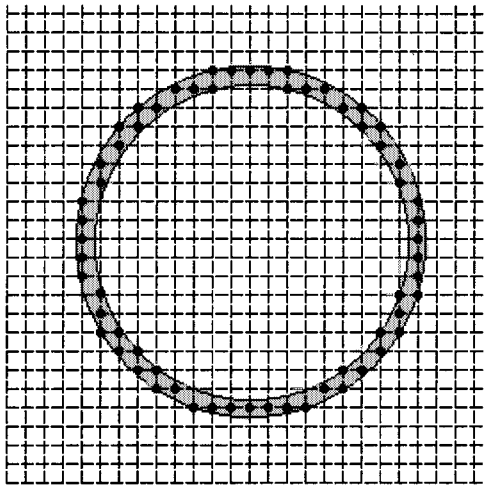

FIG. 2. A digital circle.

The case $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1) = 0$ means that the three pixels are collinear and they cannot form a circle.

Let $v_4 = (x_4, y_4)$ be the fourth edge pixel; then the distance between v_4 and the boundary of the circle C_{123}, denoted by $d_{4\to123}$, can be calculated by

$$d_{4\to123} = \left| \sqrt{(x_4 - a_{123})^2 + (y_4 - b_{123})^2} - r_{123} \right|, \tag{7}$$

where $|z|$ denotes the absolute value of z.

If v_4 lies on the circle C_{123}, the value of $d_{4\to123}$ in Eq. (7) is 0. Since the image is digital, it rarely happens that these edge pixels lie exactly on a circle. Therefore, the goal of circle detection is to detect a set of edge pixels which lie not exactly but roughly on a digital circle (see Fig. 2). For convenience, the set of edge pixels that form a digital circle is also called a circle and these edge pixels are called co-circular. As shown in Fig. 3, v_4 lies on

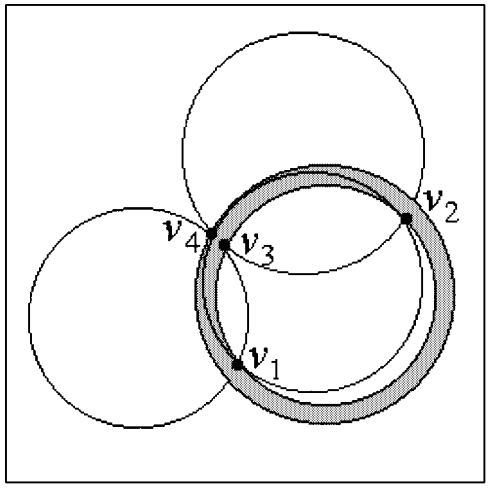

FIG. 3. An example of four pixels in a digital circle.

the boundary of the circle C_{123}; then the value of $d_{4\to123}$ in Eq. (7) is very small. Therefore, Eq. (7) can be used to determine whether v_4 lies on the circle C_{123} or not.

For convenience, we denote the circle which passes through v_i, v_j, v_k by C_{ijk} and its center and radius are denoted by (a_{ijk}, b_{ijk}) and r_{ijk}, respectively. Let the distance between v_l and the boundary of the circle C_{ijk} be denoted by $d_{l\to ijk}$. This center, radius, and distance can be calculated by Eqs. (4)–(7). For example, Eq. (7) can be written as

$$d_{l\to ijk} = \left|\sqrt{(x_l - a_{ijk})^2 + (y_l - b_{ijk})^2} - r_{ijk}\right|. \tag{8}$$

We next want to decide whether there is a circle determined by three of the four edge pixels and whether the fourth edge pixel lies on the circle.

Given four edge pixels, v_i, $i = 1, 2, 3, 4$, there are $\binom{4}{3} = 4$ circles, i.e., C_{123}, C_{124}, C_{134}, and C_{234}, with respect to the four distances, i.e., $d_{4\to123}$, $d_{3\to124}$, $d_{2\to134}$, and $d_{1\to234}$, to be considered. Once we find one distance that is smaller than a given threshold T_d, say $T_d = 0.5$ or 1, we claim that these four edge pixels are co-circular. For example, if $d_{4\to123}$ is the first distance satisfying $d_{4\to123} \le T_d$, we claim that these four edge pixels are co-circular and the circle C_{123} is the possible circle. Here, when C_{ijk} is the possible circle, the three edge pixels v_i, v_j, and v_k are referred to as the agent pixels of the possible circle.

Let us consider an undesirable case. When two of the three agent pixels of the possible circle are too close, the possible circle may not be the true circle. As shown in Fig. 4, v_1, v_2, and v_3 lie on a true circle (the bigger circle), but the circle (the smaller circle) determined by v_1, v_2, and v_3 differs from the true circle. The undesirable case occurs when v_2 and v_3 are too close. To avoid this case, the distance between any two agent pixels must be greater than a given threshold T_a. If so, it means that the three agent pixels have strong evidence to be the representatives of the possible circle.

C. Determining True Circles

After detecting a possible circle with center (a_{ijk}, b_{ijk}) and radius r_{ijk}, whether the possible circle is a true circle can be checked by the following evidence-collecting process.

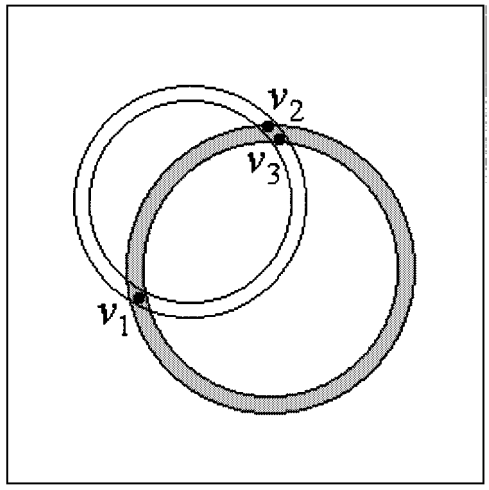

FIG. 4. An undesirable case.

Initially, we set a counter $C = 0$ for this possible circle in order to count how many edge pixels lie on the possible circle. For each edge pixel v_l in V, the distance $d_{l \to ijk}$ can be obtained by Eq. (8). If $d_{l \to ijk}$ is not larger than the given distance threshold T_d, we increment the counter C by one and take v_l out of V; otherwise we proceed to the next edge pixel. We continue the above process until all the edge pixels in V have been examined. In the evidence-collecting process, let n_p denote the number of edge pixels on the possible circle. In fact, the final value of C is equal to n_p. If n_p is larger than the given global threshold T_g, we claim that the possible circle is a true circle. Otherwise, the possible circle is a false circle and we return those n_p edge pixels into the set V. Note that when a true circle is detected, then the edge pixels lying on the circle are taken out of the set of current edge pixels. This leads to speeding up the detection of the next circle.

The above technique using a global threshold T_g has a normalized problem. Circles with different radii have different circumferences. Therefore, employing some large global threshold T_g is unfair to those circles with small radii. To overcome the normalized problem, a ratio threshold scheme is presented. Previously, Kulpa [15] showed that as $r \to +\infty$, then the number of pixels on the boundary of the circle with radius r is $4\sqrt{2}\,r$. Since any circle in a digital image has a finite radius, the number of pixels on the boundary of a circle is estimated to be $2\pi r$. Therefore, when there are n_p edge pixels lying on the possible circle C_{ijk} and the ratio of n_p over the theoretical value $2\pi r_{ijk}$ is larger than the given ratio threshold T_r, we claim that the possible circle is a true circle. Otherwise, the possible circle is a false circle and we return those n_p edge pixels into the set V.

D. The Proposed RCD

From the above description, this section presents the formal RCD consisting of the following six steps.

Step 1. Store all edge pixels $v_i = (x_i, y_i)$ to the set V and initialize the failure counter f to be 0. Let T_f, T_{min}, T_a, T_d, and T_r be the five given thresholds. Here, T_f denotes the number of failures that we can tolerate. If there are less than T_{min} pixels in V, we stop the task of circle detection. The distance between any two agent pixels of the possible circle should be larger than T_a. T_d and T_r are the distance threshold and ratio threshold, respectively. Moreover, let $|V|$ denote the number of edge pixels retained in V.

Step 2. If $f = T_f$ or $|V| < T_{min}$, then stop; otherwise, we randomly pick four pixels v_i, $i = 1, 2, 3, 4$, out of V. When v_i has been chosen, set $V = V - \{v_i\}$.

Step 3. From the four edge pixels, find out the possible circle such that the distance between any two of the three agent pixels is larger than T_a and the distance between the fourth pixel and the boundary of the possible circle is larger than T_d; go to Step 4. Otherwise, put v_i, $i = 1, 2, 3, 4$, back to V; perform $f := f + 1$; go to Step 2.

Step 4. Assume C_{ijk} is the possible circle. Set the counter C to be 0. For each v_l in V, we check whether $d_{l \to ijk}$ is not larger than the given distance threshold T_d. If yes, $C := C + 1$ and take v_k out of V. After examining all the edge pixels in V, assume $C = n_p$, i.e., there are n_p edge pixels satisfying $d_{l \to ijk} \le T_d$.

Step 5. If $n_p \ge 2\pi r_{ijk} T_r$, go to Step 6. Otherwise, regard the possible circle as a false circle, return these n_p edge pixels into V, perform $f := f + 1$, and go to Step 2.

Step 6. The possible circle C_{ijk} has been detected as a true circle. Set f to be 0 and go to Step 2.

III. EXPERIMENTAL RESULTS

All the experiments are performed on a Pentium III 733 MHz computer using C language. The first experiment is tested on the synthetic images that are created by adding noise to an original image at various increasing levels. The original 256 $\times$ 256 synthetic image with 1348 edge pixels is shown in Fig. 5a. It consists of six circles with different radii and some of them overlap. In order to test the robustness of the proposed RCD, we randomly add different levels of noise to the original synthetic image. Here, the levels range from 10%, i.e., adding 135 noises, to 200%, i.e., adding 2696 noises. The resulting two noisy images with levels 100 and 200% are shown in Fig. 5b and Fig. 5c, respectively.

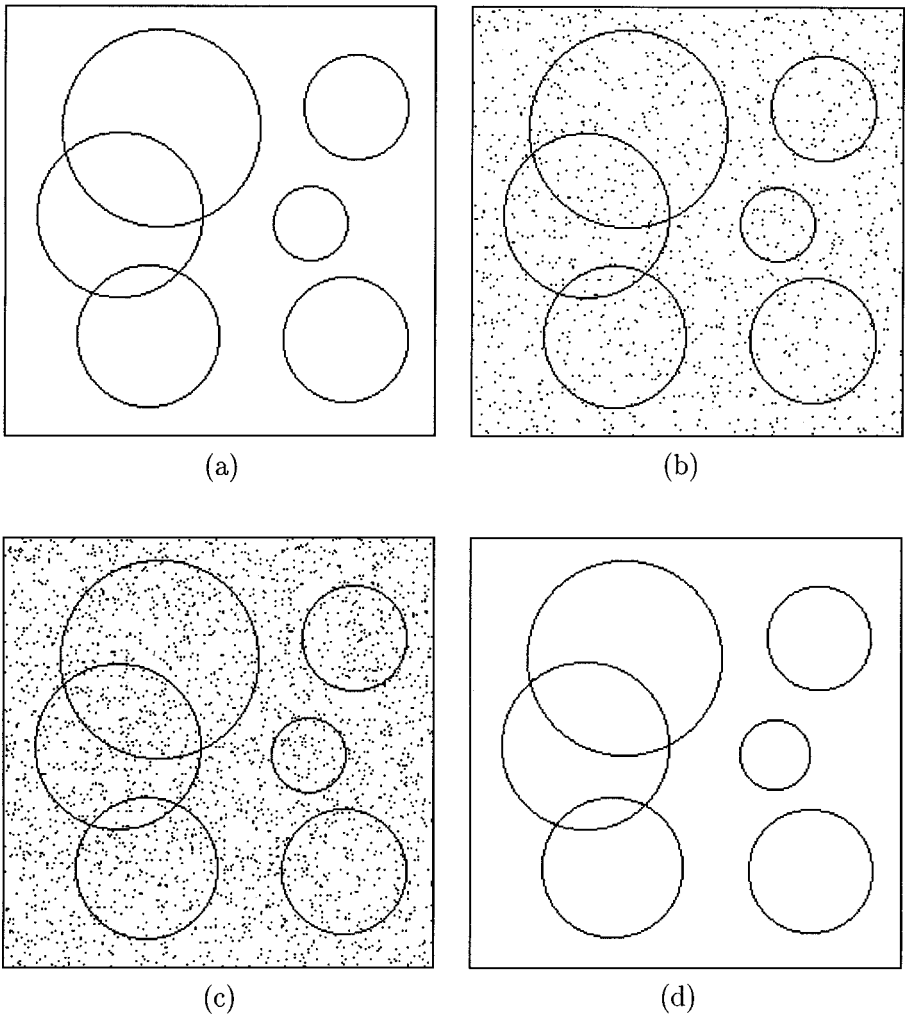

FIG. 5. The first experiment. (a) The original synthetic image. (b) The image with 1348 noises. (c) The image with 2696 noises. (d) The detected circles of the proposed RCD.

For the purpose of comparison, we apply the CHT, the RHT, and our proposed RCD to each synthetic image individually. All three methods can correctly detect the six circles. The detected circles are shown in Fig. 5d. Here, the two randomized algorithms, the RHT and our proposed RCD, are stopped when the six circles are detected.

The execution time required in each method is measured in terms of milliseconds and it is obtained from the average of 1,000 simulations. Figure 6 illustrates the execution time required in the related three methods. It is observed that the execution time required in the CHT is much larger than that in the proposed RCD and the RHT. Figures 6a and 6b are plotted to show their performance.

From Fig. 6b, except the images with high noise level, i.e., $\geq$170% , the proposed RCD is faster than the RHT. In [20], the RHT randomly picks three edge pixels and maps them into one point in the parameter space each time. The mapping is implemented by voting on the accumulator. When finding a cell in the accumulator with a satisfactory score, e.g.,

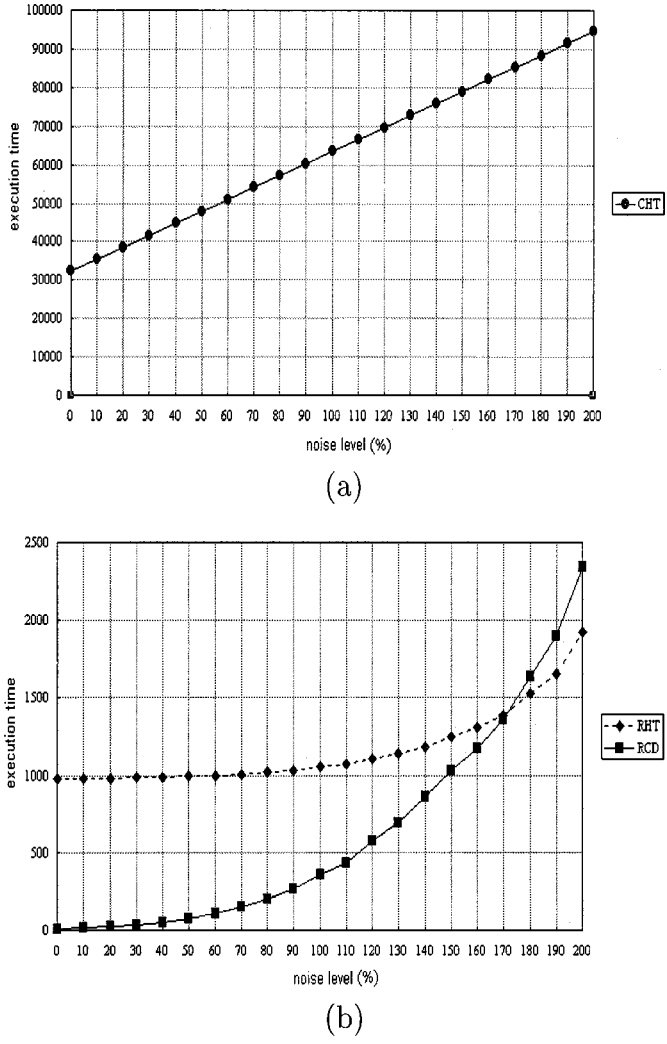

FIG. 6. The execution time comparison for synthetic images. (a) The execution time required in the CHT. (b) The execution time required in the RCD and RHT.

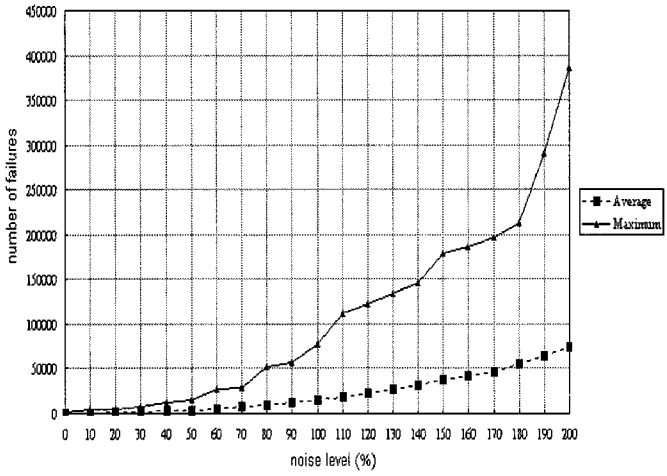

FIG. 7. The average and maximum number of failures by applying the RHT to the synthetic images from 1,000 simulations.

equal to 2, the cell represents a possible circle. The possible circle is further checked by counting the number of edge pixels lying on it to determine whether it is a true circle or not. Therefore, it takes some amount of time to maintain and access the accumulator in the voting procedure. However, the proposed RCD does not need an accumulator, which leads to a considerable time-saving effect. This is why the proposed RCD is faster than the RHT even for the images with a modest noise level.

In order to reduce the memory requirement, the RHT is implemented by using a set P which is a link-list data structure to represent the accumulator. At each time, if the three chosen edge pixels do not result in finding a true circle, it denotes the occurrence of a failure and the number of failures, f, is incremented by one. Otherwise, when a true circle is found, reset $P = null$ and $f = 0$. Due to the satisfactory score being set to 2, most failures lead to a new added cell in P. Therefore, the number of failures can be used to indicate the memory requirement in the RHT. Figure 7 shows the average and maximum number of failures obtained by applying the RHT to the synthetic images from the 1,000 simulations. In Fig. 7, it is observed that the memory requirement in the RHT increases as the noise level increases.

The second experiment is carried out on six 256×256 real images, say the coin_image, the cracker_image, the stationery_image, the culvert_image, the astronomy_image, and the toy_image, to justify the applicability of the proposed RCD. These six images are shown in Figs. 8a–8f, respectively. In Fig. 8a, the coin_image contains seven coins and two coins are occluded. In Fig. 8b, the cracker_image consists of some occluded crackers, three imperfect circular crackers, and two sticks. In Fig. 8c, the stationary_image comprises three circular objects, one pencil, and one clip. In Fig. 8d, the culvert_image includes one semicircular culvert and the other parts can be viewed as noises. In Fig. 8e, the astronomy_image contains three occluded planets with different radii. In Fig. 8f, the toy_image contains some circular plastic toys, a plastic fish, and a circular magnet. The toy_image is more complicated than the other five images. The sets of edge pixels for Fig. 8 are shown in Fig. 9, where some edges are spurious and some edges are missing. Here, the edge detection operator used is the Sobel operator [7]. Because the center of a circle must lie on the line passing through

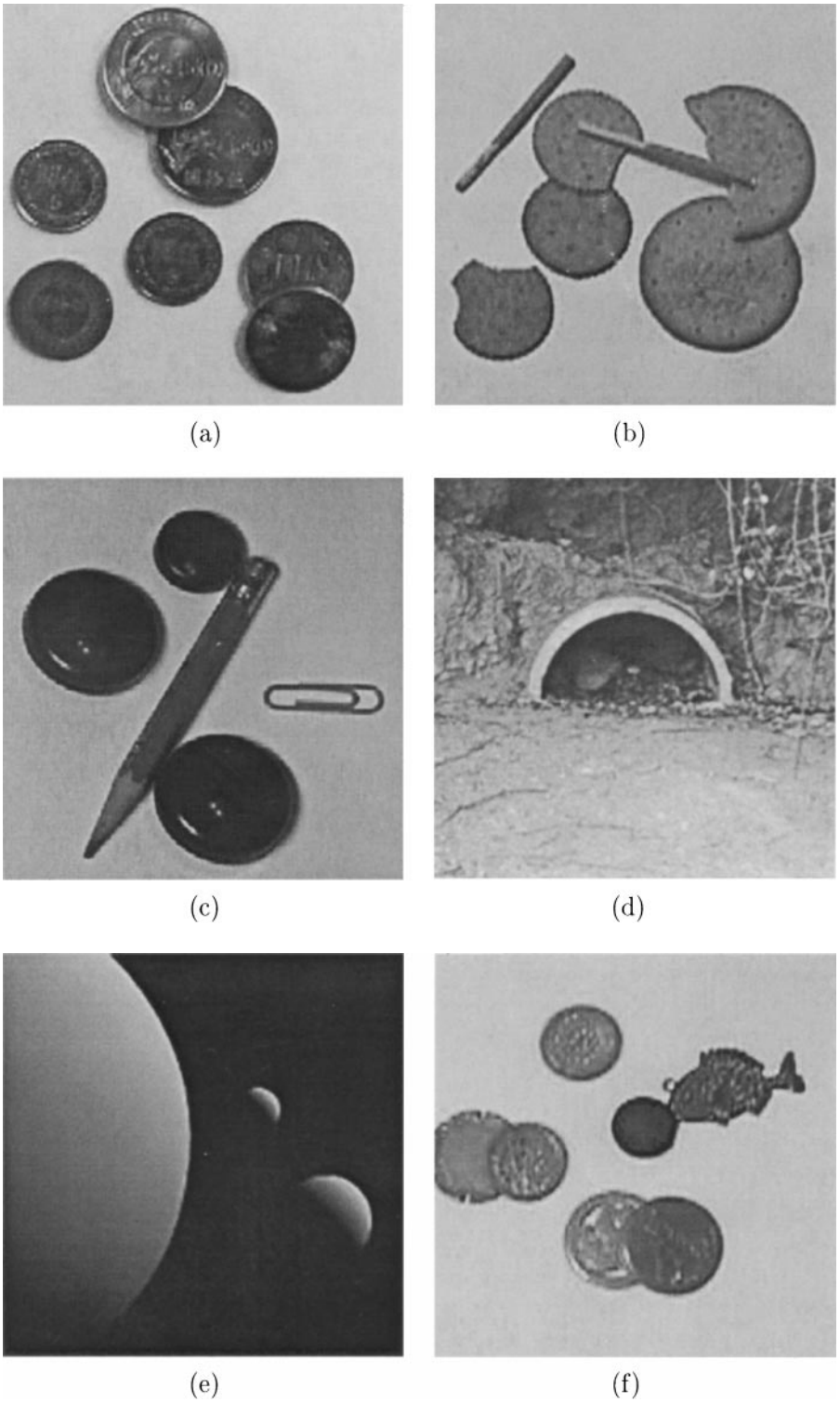

FIG. 8. The second experiment for six real-world images. (a) The coin_image. (b) The cracker_image. (c) The stationery_image. (d) The culvert_image. (e) The astronomy_image. (f) The toy_image.

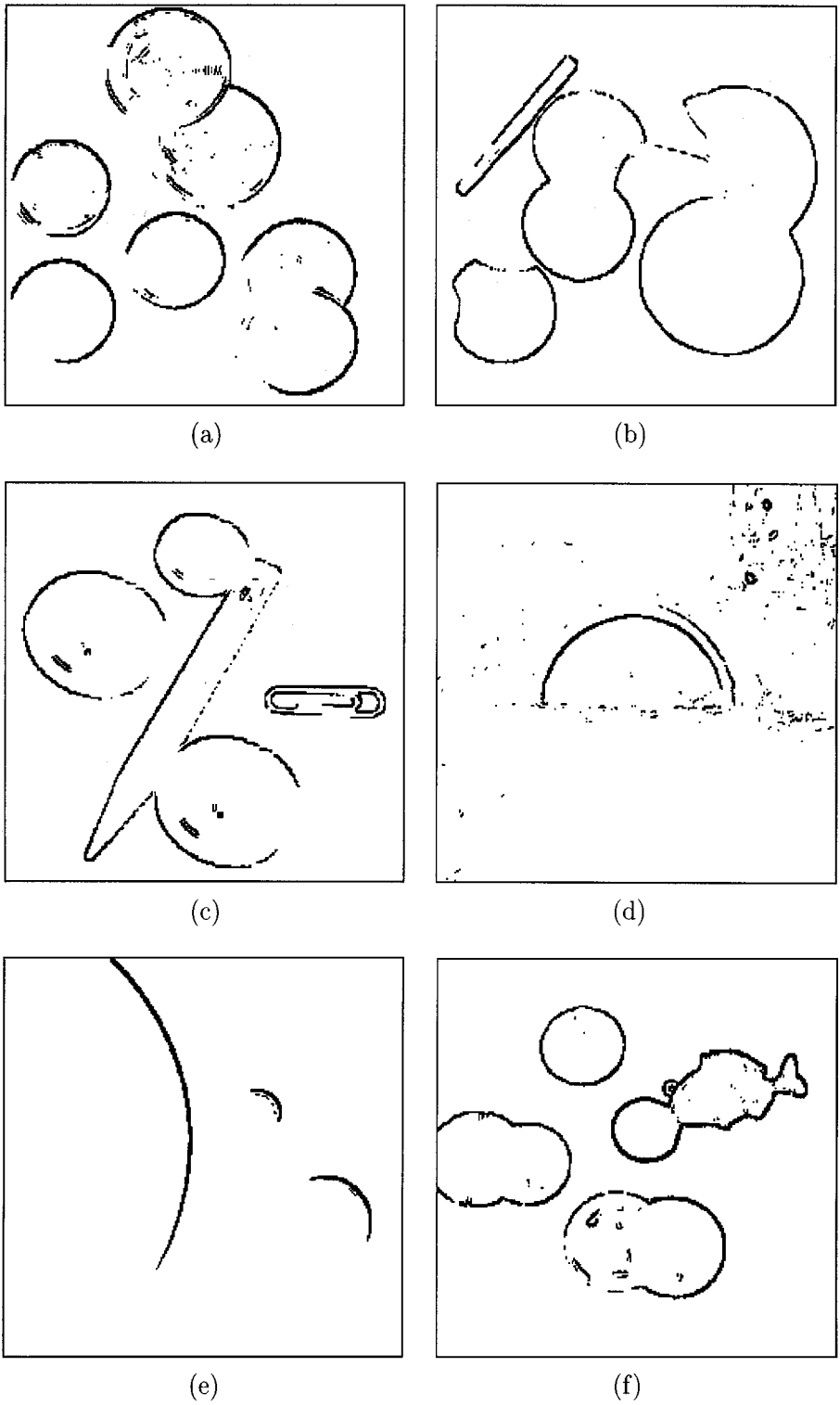

FIG. 9. The sets of edge pixels for Fig. 8. The edge pixels of (a) coin_image, (b) cracker_image, (c) stationery_image, (d) culvert_image, (e) astronomy_image, and (f) toy_image.

TABLE 1
Time Performance Comparison Among the CHT, the CHTG, the RHT, and the Proposed RCD for Six Real Images

	Image					
	Coin	Cracker	Stationery	Culvert	Astronomy	Toy
RCD	140	164	113	65	136	422
CHT	55584	56718	57185	30173	21160	63110
$\frac{CHT-RCD}{CHT}$	0.998	0.997	0.998	0.998	0.994	0.993
CHTG	747	755	754	607	551	794
$\frac{CHTG-RCD}{CHTG}$	0.813	0.783	0.850	0.893	0.753	0.469
RHT	2436	2424	1355	565	1087	5341
$\frac{RHT-RCD}{RHT}$	0.943	0.932	0.917	0.885	0.875	0.921

the edge pixel of the circle along the gradient direction, some methods [14, 22] use this property to help the task of circle detection. Here, we also implement the CHT using the gradient information [14] (denoted by CHTG) for comparison. That is, the CHT, the CHTG, the RHT, and the proposed RCD are all implemented to detect circles of the six real images separately.

While implementing the proposed RCD on the first four images of Fig. 8, the five thresholds T_f, T_{min}, T_a, T_d, and T_r are set to be 30,000, 60, 20, 1, and 60%, respectively. Because the astronomy_image contains planets with smaller radii and/or larger missing edges, the values of threshold T_a and T_r are set to be 10 and 45%, respectively. Moreover, the toy_image seems more complicated than the others, so the value of threshold T_f is set to be 60,000. Figure 10 shows the corresponding detected circles of each image. Note that in Fig. 10f, we detect the head and the body of the fish as the circular objects due to their partial circular contours. The performance comparison among the four methods is shown in Table 1.

In Table 1, the first row denotes the used real images. The execution time required in the proposed RCD is listed in the second row. The execution time of the CHT, the CHTG, and the RHT are displayed in the third, the fifth, and the seventh rows, respectively. Furthermore, each of these three rows is followed by a row which lists the corresponding improvement ratios. The improvement ratio is measured by $\frac{\text{other method} - \text{RCD}}{\text{other method}}$. The experimental results reveal that the proposed RCD is faster than the other three methods for all six real images. For example, Table 1 reveals that the proposed RCD has more than 99, 46, and 0.87% execution time improvement when compared to the CHT, the CHTG, and the RHT, respectively.

IV. TWO REMARKS

In this section, two remarks are presented to demonstrate the other two advantages of the proposed RCD.

In the HT-based methods, e.g., the CHT, the CHTG, and the RHT, due to the fact that the parameter space is quantized and the exact parameters of a circle are often not equal to the quantized parameters, we seldom find the exact parameters of a circle in the image [2, 19]. However, the proposed RCD does not employ the quantization of the parameter

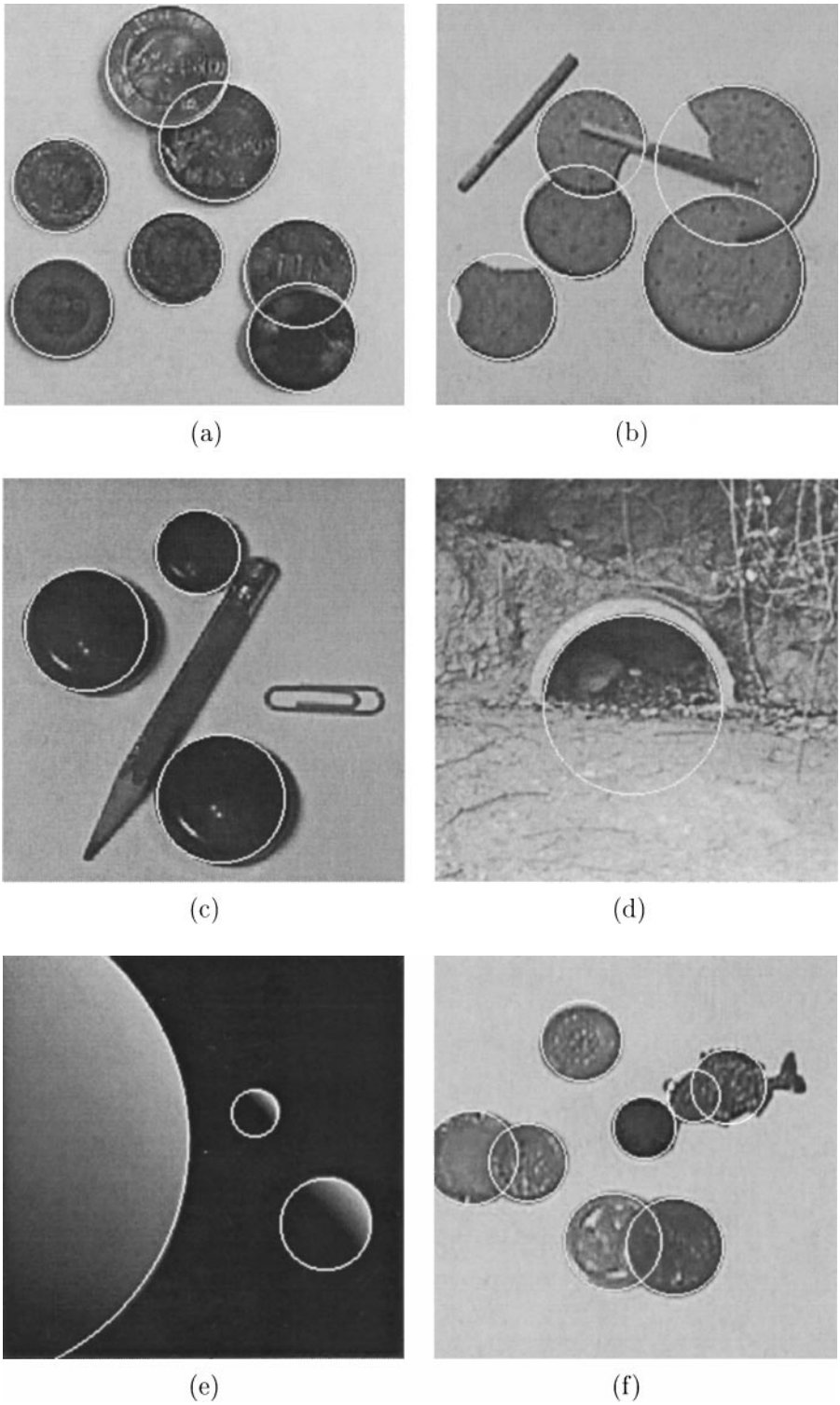

FIG. 10. The detected circles for Fig. 8. Each detected circle is depicted by a white circle. The detected circles of (a) coin_image, (b) cracker_image, (c) stationery_image, (d) culvert_image, (e) astronomy_image, and (f) toy_image.

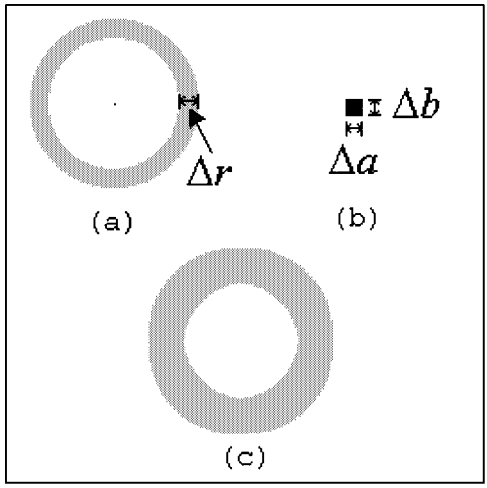

FIG. 11. Quantization bias. (a) The support region of a circle while quantizing the radius r with quantized interval Δr. (b) A cell in the (a, b) parameter space with quantized intervals Δa and Δb, respectively. (c) The support region of a circle while quantizing the (a, b, r) parameter space with quantized intervals Δa, Δb, and Δr, respectively.

space. In the RCD, the detected circles are directly obtained from Eqs. (4)–(6). Therefore, the proposed RCD can detect the circle in a more accurate way. Furthermore, also due to quantization of the parameter space, a cell in the accumulator array corresponds to a region, called a support region [8], which is not an exact annulus in the image space. As shown in Fig. 11, when we quantize (a, b, r) parameter space with quantized intervals Δa, Δb, and Δr, respectively, a cell in the accumulator array will correspond to a region in the image with the shape like Fig. 11c. This quantization effect leads to the fact that the region is not an exact annulus. The proposed RCD uses the distance between a pixel and the possible circle to test whether the pixel belongs to the circle. Therefore, the support region of a circle is an exact annulus in our method.

In the HT-based methods, the higher the resolution in parameter space is, the larger the computation–memory requirement is needed. Therefore, there is a trade-off between the resolution of the parameter space and computation–memory requirement. However, in the proposed RCD, we can dynamically adjust the threshold T_d to fit any resolution of the digital circle without increasing the computation–memory requirement.

V. DISCUSSION

In this section, we use a simple probabilistic model to discuss the computational complexities of the proposed RCD and the RHT. Considering an image containing n edge pixels, assume there exists a circle containing m edge pixels in the image; then the probability of randomly choosing a pixel from the set of edge pixels that belongs to the circle is equal to $p = m/n$. Let the event A be defined as a 3-tuple of randomly chosen pixels that come from the circle and the event B be defined as a 4-tuple of randomly chosen pixels that come from the circle; then the probabilities of A and B are given by

$$P[A] = \frac{m(m-1)(m-2)}{n(n-1)(n-2)}$$

and

$$P[B] = \frac{m(m-1)(m-2)(m-3)}{n(n-1)(n-2)(n-3)}.$$

In practice, both m and n are some what large. For example, in the culvert_image in Fig. 9d, we have $m = 413$, $n = 1255$, and $p \approx 0.33$. Thus, $P[A]$ and $P[B]$ approximate to p^3 and p^4, respectively. In the RHT, according to the satisfactory score of the possible circle being set to 2, the circle is detected when the event A occurs twice and a failure occurs when the event A does not happen. In the proposed RCD, the circle is detected when the event B occurs once and a failure occurs when the event B does not happen.

Let the random variable (r.v.) X_{RHT} be the number of failures until the event A occurs exactly twice and let the r.v. X_{RCD} be the number of failures until the event B occurs exactly once. Then the r.v. X_{RHT} has a *negative binomial distribution* [18] with probability density function (p.d.f.)

$$f_{RHT}(x) = (x+1)(1-p^3)^x(p^3)^2, \qquad x = 0, 1, \ldots,$$

where x denotes the possible number of failures until the event A occurs exactly twice. In addition, the r.v. X_{RCD} has a *geometric distribution* [18] with p.d.f.

$$f_{RCD}(x) = (1-p^4)^x(p^4), \qquad x = 0, 1 \ldots,$$

where x denotes the possible number of failures until the event B occurs exactly once. Figures 12a and 12b show the p.d.f. comparison between the r.v. X_{RHT} and the r.v. X_{RCD} for two different p's. From Fig. 12, it is observed that for $p = 0.5$ (0.25), $f_{RCD}(x)$ is larger than $f_{RHT}(x)$ when $x \leq 4$ and $x \geq 30$ ($x \leq 19$ and $x \geq 223$); $f_{RCD}(x)$ is less than $f_{RHT}(x)$ when $5 \leq x \leq 29$ ($20 \leq x \leq 222$).

The cumulative distribution function $F_{RCD}(x)$ ($F_{RHT}(x)$) can be defined as $F_{RCD}(x) = \sum_{i \leq x} f_{RCD}(i)$ ($F_{RHT}(x) = \sum_{i \leq x} f_{RHT}(i)$). Here, $F_{RCD}(x)$ is the accumulated probability that the number of failures until the event B occurs exactly once is less than or equal to x; $F_{RHT}(x)$ is the accumulated probability that the number of failures until the event A occurs exactly twice is less than or equal to x. Figures 13a and 13b show the comparison between $F_{RCD}(x)$ and $F_{RHT}(x)$ for two different p's. From Fig. 13, it is observed that for $p = 0.5$ (0.25), $F_{RCD}(x)$ is slightly larger than $F_{RHT}(x)$ when $x \leq 14$ ($x \leq 43$); $F_{RCD}(x)$ is less than $F_{RHT}(x)$ when $x \geq 15$($x \geq 44$).

Return to the noise levels discussed in the first experiment. It is known that the heavier the noise level is, the less p is. Figure 13a reveals that for a noise level between the light level and the modest level, say $1/3 < p < 1$, $F_{RCD}(x)$ is only slightly less than $F_{RHT}(x)$ when the number of failures is somewhat large. For this noise level, it seems that the proposed RCD will be slightly slower than the RHT. However, as mentioned before, when a failure occurs in the RHT, it needs some amount of time to maintain and access the accumulator. Therefore, even when the proposed RCD has larger number of failures before detecting the circle, the proposed RCD could be still faster than the RHT. This is why the proposed RCD is faster than the RHT on the noise level between the light level and the modest level. It has been confirmed in the first and second experiments.

On the contrary, for a heavy noise level, the number of failures before detecting the circle for the proposed RCD is some what larger than that for the RHT. Therefore, even considering

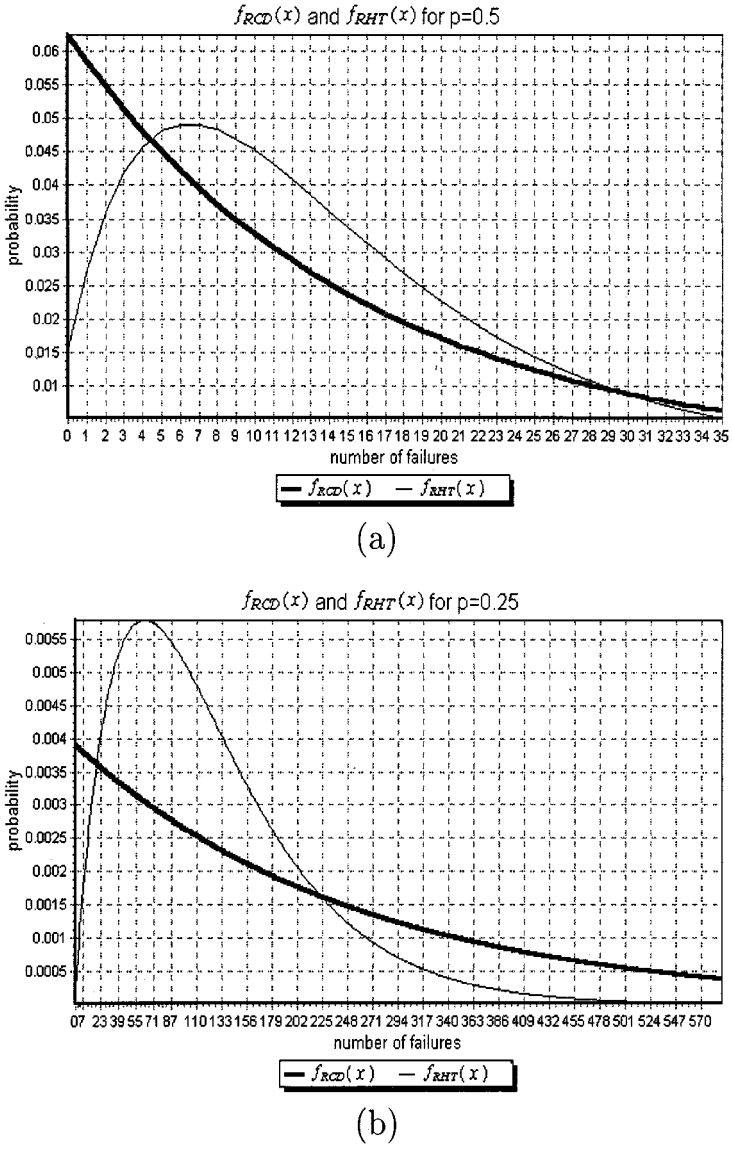

FIG. 12. The comparison between the $f_{RCD}(x)$ and $f_{RHT}(x)$ for two different p's. (a) For $p = 0.5$. (b) For $p = 0.25$.

the overhead required for the accumulator in the RHT method, the proposed RCD still takes more time to detect a circle. Figure 6b partially illustrates this phenomenon. For this noise level, the number of failures before detecting the circle for the RHT is rather large. It implies that the RHT needs a huge amount of memory requirement for the accumulator. However, our proposed RCD does not need any extra memory space for the accumulator.

In summary, the proposed RCD is faster and needs less memory space than the RHT when the noise level is between the light level and the modest level. When the noise level is heavy, the RHT is faster than the proposed RCD, but at the expense of massive memory requirement.

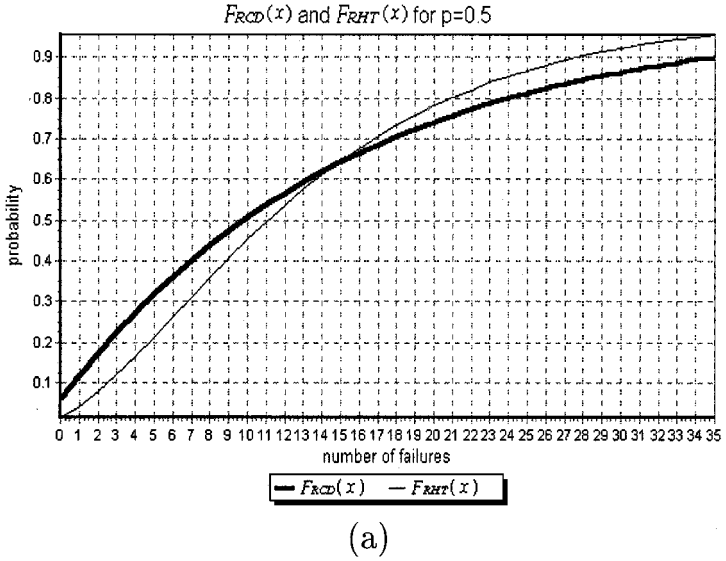

FIG. 13. The comparison between the $F_{RCD}(x)$ and $F_{RHT}(x)$ for two different p's. (a) For $p = 0.5$. (b) For $p = 0.25$.

VI. CONCLUSIONS

In this paper, we have presented an efficient non-HT-based randomized algorithm, the RCD, for detecting circles. The proposed RCD is based on randomly picking four edge pixels in the image. Then using a distance criterion, we find a possible circle. Whether the possible circle is a true circle is further checked by an evidence-collecting process. Unlike the HT-based methods, the proposed RCD does not need to vote in the parameter space. Hence, it indeed does not need any extra storage for representing the accumulator which is needed in the previous HT-based methods. Experimental results demonstrate that the proposed RCD is faster than other HT-based methods, such as the conventional HT [6],

the conventional HT making use of the gradient the information [14], and the randomized Hough transform (RHT) [20], for the noise level between the light level and the modest level. For a heavy noise level, the RHT could be faster than the proposed RCD. However, the RHT needs a huge amount of memory requirement.

How to plug the multiple window parameter transform technique [3] into the proposed RCD to balance the trade-off between accuracy and computational complexity as well as to enhance the robustness [12] and accuracy [17] of the proposed RCD are future research issues.

ACKNOWLEDGMENTS

The authors thank the anonymous referees, Area Editor Dr. Ruud M. Bolle, and L. F. Chieu for their valuable suggestions that led to the improved presentation of this paper.

REFERENCES

1. D. H. Ballard, Generalizing the Hough Transform to detect arbitrary shapes, *Pattern Recog.* **13**, 1981, 111–122.
2. C. M. Brown, Inherent bias and noise in the Hough transform, *IEEE Trans. Pattern Anal. Mach. Intell.* **5**, 1983, 493–505.
3. A. Califano and R. M. Bolle, The multiple window parameter transform, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**, 1983, 1157–1170.
4. E. R. Davies, A modified Hough scheme for general circle location, *Pattern Recog. Lett.* **7**, 1988, 37–43.
5. E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*, Academic Press, London, 1990.
6. R. O. Duda and P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, *Comm. Assoc. Comput. Mach.* **15**, 1972, 11–15.
7. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, New York, 1992.
8. K. Hansen and J. D. Andersen, Understanding the Hough transform: Hough cell support and its utilisation, *Image Vision Comput.* **15**, 1997, 205–218.
9. C. T. Ho and L. H. Chen, A fast ellipse/circle detector using geometric symmetry, *Pattern Recog.* **28**, 1995, 117–124.
10. C. T. Ho and L. H. Chen, A high-speed algorithm for elliptical object detection, *IEEE Trans. Image Process.* **5**, 1996, 547–550.
11. J. Illingworth and J. Kittler, Survey: A survey of the Hough transform, *Comput. Vision, Graphics, Image Process.* **44**, 1988, 87–116.
12. D. Ioannou, W. Huda, and A. F. Laine, Circle recognition through a 2D Hough transform and radius histogramming, *Image Vision Comput.* **17**, 1999, 15–26.
13. H. Kälviäinen, P. Hirvonen, L. Xu, and E. Oja, Probabilistic and nonprobabilistic Hough transforms: Overview and comparison, *Image Vision Comput.* **13**, 1995, 239–252.
14. C. Kimme, D. Ballard, and J. Sklansky, Finding circles by an array of accumulators, *Comm. Assoc. Comput. Mach.* **18**, 1975, 120–122.
15. Z. Kulpa, On the properties of discrete circles, rings, and disks, *Comput. Graphics Image Process.* **10**, 1979, 348–365.
16. V. F. Leavers, Survey: Which Hough transform, *CVGIP: Image Understanding* **58**, 1993, 250–264.
17. C. F. Olson, Constrained Hough transforms for curve detection, *Comput. Vision Image Understanding* **58**, 1999, 329–345.
18. G. G. Roussas, *A First Course in Mathematical Statistics*, Addison Wesley, New York, 1983.
19. T. M. VanVeen and F. C. A. Groen, Discretization errors in the Hough transform, *Pattern Recog.* **14**, 1981, 137–145.

20. L. Xu, E. Oja, and P. Kultanan, A new curve detection method: Randomized Hough transform (RHT), *Pattern Recog. Lett.* **11**, 1990, 331–338.
21. L. Xu and E. Oja, Randomized Hough transform (RHT): Basic mechanisms, algorithms, and computational complexities, *CVGIP: Image Understanding* **57**, 1993, 131–154.
22. R. K. K. Yip, P. K. S. Tam, and D. N. K. Leung, Modification of Hough transform for circles and ellipses detection using a 2-dimensional array, *Pattern Recog.* **25**, 1992, 1007–1022.
23. H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler, Comparative study of Hough transform methods for circle finding, *Image Vision Comput.* **8**, 1990, 71–77.