IEICE TRANS. INF. & SYST., VOL.E95–D, NO.6 JUNE 2012

PAPER

Circle Detection Based on Voting for Maximum Compatibility

Yuanqi SU[†a)], ***Nonmember***, **Yuehu LIU**[†], ***Member***, ***and*** **Xiao HUANG**[†], ***Nonmember***

SUMMARY We present a fast voting scheme for localizing circular objects among clutter and occlusion. Typical solutions for the problem are based on Hough transform that evaluates an instance of circle by counting the number of edge points along its boundary. The evaluated value is proportional to radius, making the normalization with respect to the factor necessary for detecting circles with different radii. By representing circle with a number of sampled points, we get rid of the step. To evaluate an instance then involves obtaining the same number of edge points, each close to a sampled point in both spatial position and orientation. The closeness is measured by compatibility function, where a truncating operation is used to suppress noise and deal with occlusion. To evaluate all instances of circle is fulfilled by letting edge point vote in a maximizing way such that any instance possesses a set of maximally compatible edge points. The voting process is further separated into the radius-independent and -dependent parts. The time-consuming independent part can be shared by different radii and outputs the sparse matrices. The radius-dependent part shifts these sparse matrices according to the radius. We present precision-recall curves showing that the proposed approach outperforms the solutions based on Hough transform, at the same time, achieves the comparable time complexity as algorithm of Hough transform using 2D accumulator array.

key words: *Hough transform, circle detection, voting, truncating operation, oriented chamfer distance*

1. Introduction

Circular objects appear quite often in our daily life, such as the coins we use everyday, the circular traffic signs we encounter on the street, the wheels of the cars we drive, and the cells we observe from the microscope. If we continue, there is still a long list. Being one of the basic geometric element that exists in our visual world, circle detection has served as a key part in many visual tasks, such as the iris detection [1], soccer detection in sports video analysis [2] and speed sign detection [3]. All these make it necessary to supply a quick and exact algorithm for circle detection.

During the past 30 yeas, many algorithms have been proposed for it. The majority of them are based on the Hough transform [4] that aims to count the edge points along the boundary of a circle. However, due to the cluttered background, occlusion, noise, imperfect edge detection and sometimes deformation, these solutions are far from being satisfactory.

The paper presents a fast voting scheme that can localize circular objects among clutter and occlusion. It works in a maximizing way, and gets rid of the Hough transform's normalization with respect to circle's radius. The main contributions of the paper include,

- A proposed formulation for circle detection that aims to the cover the circle boundary by a pre-determined number of maximally compatible edge points.
- A fast approximate algorithm that explores the sparsity of the objective function and separates it into radius-independent and -dependent parts such that the time consuming independent part can be shared by optimization for different radii.
- A comprehensive test of the algorithm on three diverse datasets that differ in image types and illuminating conditions.

The following sections explain these points in detail. In Sect. 2, we review some of the proposed algorithms for circle detection, and point out that our algorithm is closely related with oriented chamfer distance [6], [7]. Our formulation for the problem is then presented in Sect. 3 with the optimization process illustrated in Sect. 4. In Sect. 5, we report the experimental evaluations, and discuss the time complexity of the proposed algorithm. The conclusions and some further avenues are summarized in the last section.

2. Related Work

2.1 Circle Detection and the Voting Schemes

Circle detection needs to extract a set of geometric configurations of circle from the given image. A geometric configuration is a combination of circle's center and radius. If a circle in image is represented by,

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

where (x, y) is an edge point, the combination (a, b, r) then gives a geometric configuration. All the possible combinations of (a, b, r) make up the parameter space.

The majority of the proposed algorithms for circle detection relied on Hough/Generalized Hough transform (HT/GHT) [4] and their variants for the goal. In its standard form, the Hough transform parses the edge points, letting each one (x, y) vote for a cone of (a, b, r) define by Eq. (1). The GHT shrinks the voting range of the edge point by considering the edge orientation. It works by making each edge point vote in its tangent direction for the circle centers. Both HT and GHT work in accumulative way. Each entry in their voted map counts the edge points along the boundary of the

Manuscript received August 16, 2011.
Manuscript revised January 19, 2012.

[†]The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, 710049, China.
a) E-mail: yuanqi.su@stu.xjtu.edu.cn

DOI: 10.1587/transinf.E95.D.1636

circle defined by the corresponding geometric configuration.

Different from the accumulative manner, there is another type of voting scheme that usually works in a minimizing way [6], [8]–[11]. Obviously, the scheme is closely related with the HT/GHT, but has subtle difference with the standard ones. It assumes that object is represented by a set of parts. During its voting procedure, the scheme guarantees that a configuration possesses at most one instance for each its part. Some parts may be missed due to the occlusion or edge detection. Though wasn't involved in circle detection, the voting scheme has shown its value in many recent solutions for object detection [6], [8]–[11]. In the paper, we use the scheme for circle detection that is formulated as covering the boundary of a circle by a pre-determined number of edge points. In Sect. 3, we give a formal formulation, where some other strategies are integrated to suppress noises and deal with occlusion.

2.2 Circle Detection

An extensive literature exists, addressing the problem of circle detection. In a paper [12] in 1990, some early methods were summarized and experimentally compared, and one of them was implemented as a standard module in OpenCV (Open Source Computer Vision) [13]. Next, we briefly review the proposed algorithms according to the strategies that they used.

There are various modifications suggested for saving the storing space, improving the robustness of detection, and reducing the time complexity. Besides the use of edge orientation as that in GHT, these major modifications include: the simultaneous consideration of a range of circle radii [12], [14], [15], use of a complex accumulator array with radius modulated phase [15]–[17], incorporation of different sampling strategies [18]–[21], exploration of the symmetry [3], [22], [23], adoption of the hierarchical structure [24], and the implementation of the circle detection as filtering operations [17]. It is worth noting that these modifications are not isolated, in fact, they usually work in cooperation.

Use of Edge Orientation

Kimme *et al.* [25] were the first to suggest the use of the edge orientation. With their modification, each edge point votes for a circle center along the tangent direction. The voting way can also be viewed as the application of GHT [4] for circle detection.

Considering a Range of Circle Radii

To consider a range of circle radii [12], [14], [15] saves the storing space and avoids voting one radius by one. The modification is always implemented by first localizing the circle centers, followed by resolving the radius. In detection, it makes each point vote in a line segment along the tangent direction. The line segment is determined by the range of radius. Algorithm with the modification usually uses two-instead of the three-dimensional accumulator array and thus reduces the memory requirement.

Using Phase Related Radius

The modification [15]–[17] uses phase to code for radius and votes with complex numbers. Since different radii can be discriminated by phase of the complex value, algorithms with the strategy usually consider the whole range of radii and work with 2D complex accumulator array. The localized maxima from the accumulator array contain both the magnitude and the phase, and the latter is used for identifying radius. Zelniker *et al.* [26] proposed to combine the maximum-likelihood estimation (MLE) with the modification. Their approximate version of the MLE used the complex convolution where radius range was modulated to generate the complex template.

Incorporating Different Sampling Strategies

Different sampling strategies [18]–[21] are usually accompanied by the randomized Hough transform (RHT) [27], [28] or geometric hashing (GH) [29], that can be viewed as the variants of HT/GHT. When used for circle detection, RHT randomly samples three edge points, determines a member in parameter space, and then modifies the corresponding entry of accumulator array. Instead of three points, Chen *et al.* [20] sampled four points, and checked their compatibility to determine whether these points came from a circle. If the answer was yes, a candidate was generated, and subsequently proved with an evidence-collection procedure similar to GH [29]. Their method gets rid of the accumulator array and thus saves the storing space. For the same purpose, Chiu *et al.* [21] proposed a sampling-and-searching strategy. It worked by sampling two points, and finding the third one from the circles constrained by the first two. The subsequent generation and checking of the candidates then shared the same process as [20]. Ioannou *et al.* [19] gave their sampling rule based on an assumption that edge points of a connected component usually came from the same object. Their algorithm traversed the pairs of edge points from the same connected component, and casted the vote for the centers. These sampling strategies neglects the edge orientations. Chung *et al.* [30] proposed to use the tangent orientation for checking the sampled points that further reduced the sampling tries.

The sampling strategy can also be combined with other robust statistical estimators, such as the RANdom SAmple Consensus (RANSAC). Hilaire and Tombre [31] proposed a systematical way to vectorize the line drawings. Among their proposed algorithms, one was for arc and circle. Besides RANSAC, Lamiroy *et al.* [32] lent an idea from the image tracking, that gradually deformed the candidates of circular arcs as to precisely fit onto the image strokes, or to reject them if the fitting was not possible.

Exploring the Symmetry

The use of the geometric symmetry was introduced by Ho *et al.* [22]. Their method tried to find the symmetric axes of circles. For the purpose, they vertically and horizontally scanned the image, calculated the midpoints of each pair of edge points in the same scanning line, and fit lines through the midpoints for candidates of symmetric axes. To explore the symmetry avoids the voting stage required by the Hough transform and improves the efficiency. It can be directly generalized to ellipse detection [3], [22], [23].

Adopting the Hierarchical Structure

The hierarchical structure [24] is mainly for improving the efficiency. Its mechanism is to guide the search in parameter space in a coarse-to-fine way. It iteratively splits the parameter space according to some predefined rules. Guil *et al.* [24] presented such a method for circle detection, that spited the parameter space based on the fast Hough transform [33].

Filtering for Hough Transform

The modification views the Hough transform as filtering operation. Atherton *et al.* [17] defined several filters corresponding to different algorithms for circle detection, explored their size-invariant properties, and proposed to use them in combination. As mentioned before, in a recent work by Zelniker *et al.* [26], complex convolution was used for approximating their formulation for circle detection based on maximum likelihood estimation.

Some Other Methods

There are some other methods that are not included in the introduction of modifications. Like the method proposed by Schuster *et al.* [34], they formulated the problem of finding circles as the estimation based on weighted minimum mean square error. Ayala-Ramirez *et al.* [35] gave another choice. They placed the task in the genetic algorithm.

2.3 Oriented Chamfer Distance

Besides, our method is also related with the oriented chamfer matching [6], [7]. The chamfer distance [36] takes in two sets of points, a template T and a test set S, and calculates a distance by,

$$d(S,T) = \frac{1}{|T|}\sum_{\mathbf{x}\in T}\min_{\mathbf{y}\in S}\|\mathbf{y}-\mathbf{x}\|_2 \tag{2}$$

where $|T|$ represents the total number of points in template T, $\|\cdot\|_2$ is the Euclidean distance. Shotton *et al.* [6] modified the chamfer distance, considered the tangent orientation, and gave the oriented chamfer distance as follows.

$$d(S,T) = \frac{1}{|T|}\sum_{\mathbf{x}\in T}\min_{\mathbf{y}\in S}\|\mathbf{y}-\mathbf{x}\|_2 + \lambda|\phi(\mathbf{y})-\phi(\mathbf{x})| \tag{3}$$

where ϕ gives the edge orientation of the assigned point, λ balances the influence of Euclidean distance and orientation difference. To optimize (3), they proposed a two-stage way that adopted distance transform of sampled function [37] for finding the nearest point $\mathbf{y}$ for each $\mathbf{x}$, and then computed the orientation difference with the nearest $\mathbf{y}$. Ming-Yu *et al.* [7] optimized the same objective function (3) in a dense manner, that slides the window over the entire image domain. Instead of approximating the solution, their method draws the extract solution with the aid of dynamic programming. To speed up searching process, they introduced a rejecting strategy to bypass majority of the sliding windows.

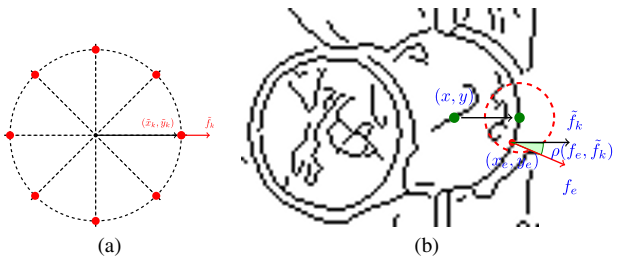
Since we also consider the orientation, our objective function comprises both the spatial and orientational terms, making itself similar to the oriented chamfer distance. However, in order to suppress noises and deal with occlusion, we introduce some other strategies in our objective function. In the optimization stage, we discard the sliding-window way and turn to a more efficient manner that explores characteristics implied in our formulation for circle detection.

3. Voting for Maximum Compatibility

We start by defining some of the notations used by our algorithm. For a circle centered on the origin, let $\{(\tilde{x}_k,\tilde{y}_k)|k=0,\ldots,K-1\}$ denote a set of K evenly sampled points from its boundary, and $\{\tilde{f}_k|k=0,\ldots,K-1\}$ specify the corresponding tangent orientations. The representation for $K=8$ is shown in Fig. 1 (a). For a circle centered on (x,y), its sampled points move to $\{(x+\tilde{x}_k, y+\tilde{y}_k)|k=0,\ldots,K-1\}$, while the orientation components keep the same. On the image side, after applying an edge detector to the input image, we get a set of edge points $\{(x_e,y_e)|e=1,\ldots,E\}$ as shown in Fig. 2 with their orientations denoted by $\{f_e|e=1,\ldots,E\}$.

The problem of evaluating the circle centered on (x,y), needs to obtain a set of K edge points, each close to a sampled point both in spatial position and edge orientation. To measure closeness, we define an error function that is similar to the one used in oriented chamfer distance [6], [7]. Supposing that the edge point (x_e,y_e) matches with the kth sampled point, the matching error between them takes the following form.

$$\begin{aligned}&\mathcal{E}_k((x,y),(x_e,y_e))\\ &\quad=\rho(f_e,\tilde{f}_k)+\frac{\|x_e-\tilde{x}_k-x\|_2^2}{\sigma_{k,x}^2}+\frac{\|y_e-\tilde{y}_k-y\|_2^2}{\sigma_{k,y}^2}\end{aligned} \tag{4}$$



(a) (b)

Fig. 1 Model and the voting range.

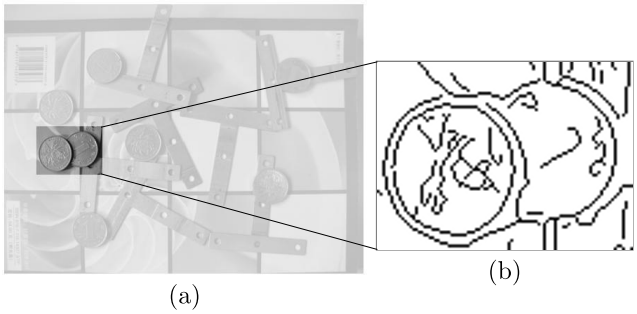


Fig. 2 An image together with its edge map. (a) shows the original image and (b) gives the edge map by Canny detector [5] for the cropped region.

where ρ evaluates the difference of orientation, $[x_e - \tilde{x}_k - x, y_e - \tilde{y}_k - y]$ gives the spatial deviation, $\sigma_{k,x}$ and $\sigma_{k,y}$ are the normalization factor in x and y direction respectively.

As a standard practice, we truncate the matching error to suppresses the noises and further replace the truncated value with the following compatible function.

$$O_k((x,y),(x_e,y_e)) = \max(\tau - \mathcal{E}_k((x,y),(x_e,y_e)), 0), \quad (5)$$

where the maximization is used as the truncating operation, τ bounds the matching error $\mathcal{E}$ from above and is set to 1 throughout the paper. The compatible function guarantees the sparsity and subsequently the computation-saving of the voting process, that will be illustrated in the next section.

From the perspective of voting, Eq. (5) supplies us with another explanation, where each edge point (x_e, y_e) casts a weighted vote for the possible locations of the circle's center. Circle whose center is in the voting range can choose edge point (x_e, y_e) to match with its kth sampled point. Range for the possible locations is subject to the constraint posed by the truncating operation. When the orientation difference $\rho(f_e, \tilde{f}_k)$ is less than τ, the possible locations are within an elliptical region centered on $(x_e - \tilde{x}_k, y_e - \tilde{y}_k)$. Each circle center (x, y) in the region satisfies,

$$\frac{\|x_e - \tilde{x}_k - x\|_2^2}{\sigma_{k,x}^2} + \frac{\|y_e - \tilde{y}_k - y\|_2^2}{\sigma_{k,y}^2} \le \tau - \rho(f_e, \tilde{f}_k).$$

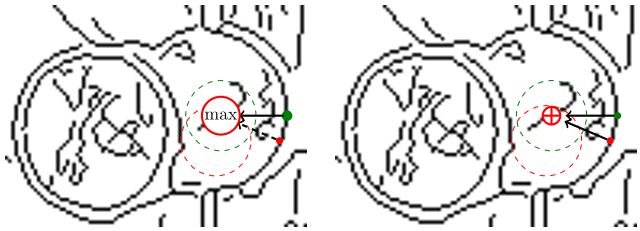
On the other hand, for circle centered on (x, y), there exists more than one edge point that can match with its kth sampled point. In other words, the circle may receive votes from multiple edge points. Among these, only the one with maximum compatible value is accepted, given by,

$$O_k(x,y) = \max_e O_k((x,y),(x_e,y_e)). \quad (6)$$

The compatibility O for the circle is then averaged over k,

$$O(x,y) = \frac{1}{K}\sum_k O_k(x,y). \quad (7)$$

Our voting scheme is illustrated in Fig. 3, together with the comparison with Hough transform. Different from Hough transform, it works in a maximizing way and gets rid of the normalization with respect to the circle's radius.



(a) (b)

Fig. 3 Voting for the maximum compatibility and the Hough voting.

3.1 Difference of Orientation

We define the difference of orientation and show that it is independent on circle's radius. Let r denote the radius of a circle. Since we equally sample the circle's boundary, coordinates for its kth sampled point are $\tilde{x}_k = r\cos\left(\frac{2\pi k}{K}\right)$, $\tilde{y}_k = r\sin\left(\frac{2\pi k}{K}\right)$, and the tangent orientation $\tilde{f}_k$ is,

$$\tilde{f}_k = \mathrm{mod}\left(\frac{2\pi k}{K}, \pi\right). \quad (8)$$

Given an edge point, to measure its orientational difference with respect to $\tilde{f}_k$, we introduce the following metric,

$$\rho(f_e, \tilde{f}_k) = K\frac{|f_e - \tilde{f}_k|_f}{2\pi}, \quad (9)$$

where,

$$|f_e - \tilde{f}_k|_f = \min\begin{pmatrix} \mathrm{mod}(|f_e - \tilde{f}_k|, \pi) \\ \pi - \mathrm{mod}(|f_e - \tilde{f}_k|, \pi) \end{pmatrix}.$$

Due to $\tilde{f}_k$'s independence, the difference of orientation is also independent on circle's radius. Besides, it is easy to verify that $|f_e - f_k|_f$ is between 0 and $\frac{\pi}{2}$. Range for orientational difference is then,

$$0 \le \rho(f_e, \tilde{f}_k) \le \frac{K}{4}. \quad (10)$$

In Fig. 4, the number of sampled points is $K = 200$, and the orientational difference is between 0 and 50.

4. The Optimization Process

The radius-independence of the orientational difference is explored for accelerating the voting procedure. The corresponding mechanism is to separate the calculation of O_k into the radius-dependent and -independent parts, such that to calculate O_k for different radii can share the same radius-independent part. As for the independent part, we set both $\tilde{x}_k$ and $\tilde{y}_k$ in O_k to zero, resulting in,

$$\mathcal{D}_k(x,y) = \max_e \max\left[\tau - \left(\rho(f_e, \tilde{f}_k) + \frac{\|x_e - x\|_2^2}{\sigma_{k,x}^2} + \frac{\|y_e - y\|_2^2}{\sigma_{k,y}^2}\right), 0\right]. \quad (11)$$

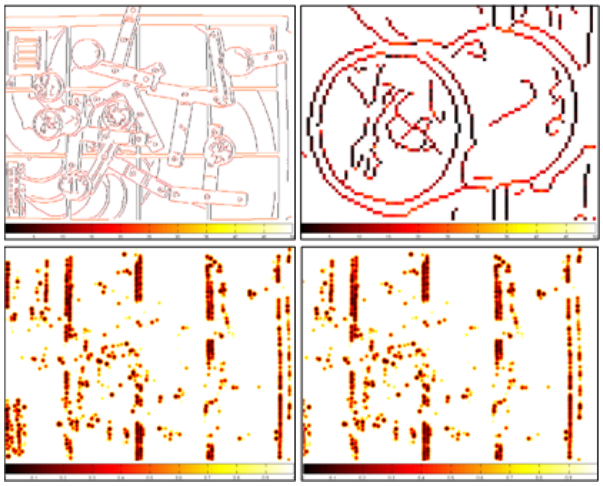


Fig. 4 Demonstration of our voting procedure. From left to right, images in the top row illustrate the orientation difference, and that for cropped region respectively. Images in bottom row then illustrate the matching error $1 - \mathcal{D}_k$ and its shifted version $1 - O_k$.

Input: $\mathcal{D}_k(x, y)$ and searching radii $\{r_1, \ldots, r_{N_r}\}$
Output: Compatible function $\{O^{(1)}(x, y), \ldots, O^{(N_r)}(x, y)\}$
Initialize $O^{(n)}(x, y) \leftarrow 0, \ \forall x, y, n$
for $k = 0$ **to** $K - 1$ **do**
  for $n = 1$ **to** N_r **do**
    $\bar{x}_k \leftarrow \text{round}(r_n \cos(\frac{2\pi k}{K}))$, $\bar{y}_k \leftarrow \text{round}(r_n \sin(\frac{2\pi k}{K}))$
    $O_k^{(n)} \leftarrow$ Shift $\mathcal{D}_K$ by $[-\bar{x}_k, -\bar{y}_k]$
    $O^{(n)}(x, y) \leftarrow O^{(n)}(x, y) + \frac{1}{K} O_k^{(n)}$
  end for
end for

Fig. 5 Proposed algorithm for circle detection.

Because the orientational difference is radius-independent, the same holds for $\mathcal{D}_k$. With $\mathcal{D}_k$, the compatible function $O_k(x, y)$ is approximated via a shifting operation.

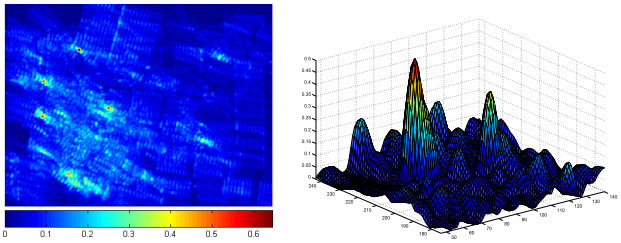
$$O_k(x, y) \approx \mathcal{D}_k(\text{round}(x + \tilde{x}_k), \text{round}(y + \tilde{y}_k)), \tag{12}$$

where elements of $(x + \tilde{x}_k, y + \tilde{y}_k)$ are rounded to the nearest integers. We use the rounding operation in (12) for $(x + \tilde{x}_k, y + \tilde{y}_k)$ to be usually off the grid locations of the image domain. Since (x, y) is an image grid, the rounding operation in (12) can operate on $\tilde{x}_k$ and $\tilde{y}_k$ directly.

$$O_k(x, y) \approx \mathcal{D}_k(x + \text{round}(\tilde{x}_k), y + \text{round}(\tilde{y}_k)), \tag{13}$$

Once $\mathcal{D}_k$ is known, approximating O_k is reduced to the shifting operation. When boundary condition is neglected, O_k is approximated by shifting $\mathcal{D}_k$ with $[-\text{round}(\tilde{x}_k), -\text{round}(\tilde{y}_k)]$. The integer scalars $-\text{round}(\tilde{x}_k)$ and $-\text{round}(\tilde{y}_k)$ specify the shift amount for the corresponding dimension of $\mathcal{D}_K$; and vary with circle's radius, making O_k dependent on r. The proposed algorithm integrating the mechanism is summarized in Fig. 5.

In Fig. 5, $O^{(n)}$ stores the compatible function for radius r_n. Local maxima are then localized and thresholded for candidates. Figure 6 shows a map for such $O^{(n)}$ whose radius is 24.29 pixels. The right side in Fig. 6 is the surface view of compatible function $O^{(n)}$ for the cropped region in Fig. 2.



(a) (b)

Fig. 6 The compatible function. (a) gives the compatible function of the whole image domain. (b) is the surface view of the cropped region.

4.1 Batch Algorithm for $\{\mathcal{D}_k\}$

The left task for the circle detection is to calculate $\{\mathcal{D}_k\}$. First, we show that by using even number of sampled points, the count of the $\mathcal{D}_k$ to be calculated can be reduced by half. The tangent orientation for kth sampled point is $\tilde{f}_k = \text{mod}(\frac{2k\pi}{K}, \pi)$. When K is even, it preserves the following property.

$$\tilde{f}_k = \begin{cases} \frac{2k\pi}{K} & 0 \le k \le \frac{K}{2} - 1 \\ \frac{2k\pi}{K} - \pi = \tilde{f}_{k-\frac{K}{2}} & \frac{K}{2} \le k \le K - 1 \end{cases} \tag{14}$$

Since $\tilde{f}_k = \tilde{f}_{k-\frac{K}{2}}$ leads to $\mathcal{D}_k = \mathcal{D}_{k-\frac{K}{2}}$, we only need to calculate $\mathcal{D}_k$ for $k = 0, \ldots, \frac{K}{2} - 1$.

Next, the influence of the truncating operation on the calculation of $\mathcal{D}_k$ is investigated, that inspires the batch way for calculating $\{\mathcal{D}_k\}$. The truncating value bounds both the orientational difference and the spatial deviation.

$$\rho(f_e, \tilde{f}_k) + \frac{\|x_e - x\|_2^2}{\sigma_{k,x}^2} + \frac{\|y_e - y\|_2^2}{\sigma_{k,y}^2} \le 1.$$

During the calculation of $\mathcal{D}_k$, when the orientational difference is greater than the truncating value, the corresponding edge point is discarded; only those satisfying $\rho(f_e, \tilde{f}_k) \le 1$ are involved. Similarly, if we fix f_e, the range for k is also subject to the constraint. After substituting the definition of orientational difference into the constraint, we have,

$$K \frac{|f_e - \tilde{f}_k|_f}{2\pi} \le 1 \Rightarrow |f_e - \tilde{f}_k|_f \le \frac{2}{K}\pi.$$

For $k \in [0, \frac{K}{2} - 1]$, the definition for $\tilde{f}_k$ allows us to rewrite the constraint as $|f_e - \frac{2k\pi}{K}|_f \le \frac{2}{K}\pi$. In practice, the selection of K guarantees that $\frac{2}{K}\pi$ is less than $\frac{\pi}{2}$. The agreement makes $\tilde{f}_k$ lie in $[f_e - \frac{2\pi}{K}, f_e + \frac{2\pi}{K}]$. The influenced k by edge point (x_e, y_e) then satisfies,

$$-1 + \frac{K f_e}{2\pi} \le k \le 1 + \frac{K f_e}{2\pi}. \tag{15}$$

When a k from the above range is outside $[0, \frac{K}{2} - 1]$, it is modulated with respect to $\frac{K}{2}$.

After determining the sampled point k, we get an elliptical region for the spatial locations.

Input: K
Output: $\{\mathcal{D}_k | k = 0, \ldots, \frac{K}{2} - 1\}$
Initialize $\mathcal{D}_k(x, y) \leftarrow 0, \ \forall x, y, k$
for $e = 1$ **to** E **do**
  $k_e \leftarrow \text{round}(\frac{K f_e}{2\pi})$
  for $k' = -1$ **to** 1 **do**
    $k \leftarrow \text{mod}(k_e + k', \frac{K}{2})$
    $\rho(f_e, \tilde{f}_k) \leftarrow K\frac{|f_e - \tilde{f}_k|_f}{2\pi}$
    if $\rho_k(x_e, y_e) \geq 1$ **then**
      Continue
    end if
    $\beta \leftarrow \sqrt{1 - \rho(f_e, \tilde{f}_k)}$
    for all (x', y') such that $|x'| \leq \beta\sigma_{k,x}$ and $|y'| \leq \beta\sigma_{k,y}$ **do**
      $\mathcal{D}_k(x_e + x', y_e + y') \leftarrow$
      $\max\left(\mathcal{D}_k(x_e + x', y_e + y'), H(x', y') - \rho(f_e, \tilde{f}_k)\right)$
    end for
  end for
end for

Fig. 7 Batch algorithm for $\mathcal{D}_k$.

$$\frac{\|x_e - x\|_2^2}{\sigma_{k,x}^2} + \frac{\|y_e - y\|_2^2}{\sigma_{k,y}^2} \leq 1 - \rho(f_e, \tilde{f}_k).$$

The region is contained in an external rectangle, whose sides are $\left(1 - \rho(f_e, \tilde{f}_k)\right)^{0.5} \sigma_{k,x}$ and $\left(1 - \rho(f_e, \tilde{f}_k)\right)^{0.5} \sigma_{k,y}$ respectively. The maximum rectangular region is achieved when the orientational difference equals zero. To avoid the repeat evaluation of spatial deviations in the lefthand side of the above inequality, we precomputed values defined on the maximum rectangular region. A $(2\lceil\sigma_{k,x}\rceil + 1) \times (2\lceil\sigma_{k,y}\rceil + 1)$ matrix is used, whose entry is given by,

$$H(x, y) = 1 - \frac{x^2}{\sigma_{k,x}^2} - \frac{y^2}{\sigma_{k,y}^2} \tag{16}$$

where, $x = -\lceil\sigma_{k,x}\rceil, \cdots, \lceil\sigma_{k,x}\rceil$ and $y = -\lceil\sigma_{k,y}\rceil, \cdots, \lceil\sigma_{k,y}\rceil$.

In Fig. 7, we give the batch algorithm for $\{\mathcal{D}_k\}$, that takes advantage of the truncating operation. It fills the entries of all $\mathcal{D}_k$ in just one-round sweeping of the edge points.

In fact, each $\mathcal{D}_k$ is sparse, only limited entries are nonzero. In Fig. 4, we give an example where the sampled points amount to 200, and k equals zero. There are several reasons for the sparsity. First, compared to the image grids, the detected edge points are sparse. Second, the truncating operation shrinks the number of edge points in the calculation of each $\mathcal{D}_k$. Third, we chose $\sigma_{k,x}$ and $\sigma_{k,y}$ such that the elliptical region is narrow enough. Finally, it is the use of the compatible function instead of the error function that makes $\mathcal{D}_k$ sparse. If a grid location (x, y) does not appear in any elliptical region of the edge point, the compatible function assigns the zero, while the error function outputs 1. Since the union of the elliptical regions occupies only a small portion of the image domain, the compatible function is preferred for sparsity.

5. Experimental Evaluations

Experiments are conducted on three datasets, containing coins, circular seals, and the circular traffic signs. Comprehensive evaluations are reported, including the average testing time, the performance together with the comparisons. Summary info of these datasets is listed in Table 1. In our test images, the occlusion, multiple instance, and cluttered background are presented, that increase the difficulty of circle detection.

Table 1 Summary info of the datasets.

Name:	Coins	Seals	Traffic signs
Number of images:	17	14	170
Number of instances:	100	94	265
Image type:	artificial lighting, color images	document, binary images	natural, color images

We compare the proposed algorithm against Hough transforms [13], [38], the randomized circle detection (RCD) [20] and algorithm based on maximum likelihood estimation (AMEL) [26]. The implementation of RCD follows the tips in [20] and adds the orientational information [30] when proposing and checking candidates of circle. As for AMEL [26], we add an auxiliary step that counts the number of edge points along each candidate for circle, and discards those with small values. The strategy suppresses the false positives. To make the comparisons fair, all these algorithms uses the same searching range of radius as ours and their radius sample interval is one pixel. Besides, we tune the parameters of these algorithms for their best performance on each dataset.

In detection, the Hough transform by Tao [38] thresholds the gradient magnitudes for edge points, while the Hough transform from Opencv [13], RCD and AMEL all use the Canny edge detector [5]. Our algorithm also chooses Canny, it uses the implementation in Matlab and keeps the parameters default. Besides, we compute edge orientation with Sobel operator. Other parameters specialized for each dataset are reported and analyzed in the following subsections.

5.1 Coin Detection

For coin detection, size of each image is normalized to 512, i.e. $\max(w, h)$ is equal to 512, where w, h are width and height of an image. In order to save the computational time, we discretized the searching range of radius, instead of using one pixel as the radius sample interval. Given a searching range $[r_{lo}, r_{hi}]$, it is discretized into,

$$r_n = \frac{(n-1)(r_{hi} - r_{lo})}{N_r - 1} + r_{lo} \quad n = 1, \ldots, N_r.$$

where $\frac{r_{hi} - r_{lo}}{N_r - 1}$ is the radius sample interval. In coin detection, $r_{lo} = 10$, $r_{hi} = 60$ and $N_r = 10$ are sufficient for accomplishing satisfactory results. Correspondingly, the normalization factors $\sigma_{k,x}$, $\sigma_{k,y}$ are chosen by,

$$\sigma_{k,x} = \sigma_{k,y} = 0.6 \times \frac{r_{hi} - r_{lo}}{N_r - 1}. \tag{17}$$

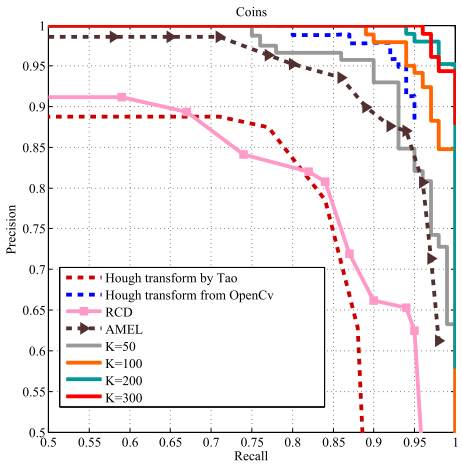
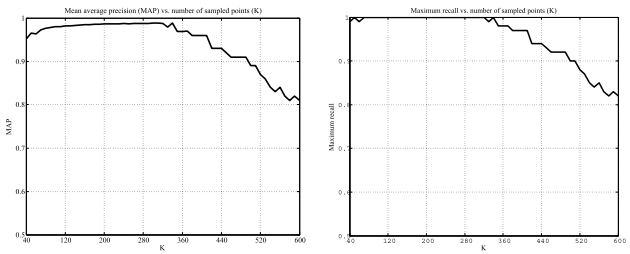


Fig. 8 Precision vs. recall (PR) curves for coin detection.



(a) (b)

Fig. 9 (a) Area under cover (AUC) vs. number of sampled points for coin detection. (b) Maximum recall vs. number of sampled points for coin detection.

The 0.6× radius sample interval guarantees that the searching of circle's radius covers the whole range $[r_{lo}, r_{hi}]$.

We summarize the precision vs. recall (PR) curves in Fig. 8. The proposed algorithm outperforms the others except for $K = 50$; when for $K = 50$, ours is inferior to the Hough transform by Tao, and is better than the other ones. The situation is slightly different when area under cover (AUC) is measured. AUC calculates the covered area by a PR curve, and its value lies in the range of $[0, 1]$. AUC of Hough transform by Tao is 0.946; while for ours with $K = 50$, the AUC is 0.975, that is better than both Hough transform.

We also report the AUC vs. number of sampled points in Fig. 9. For plotting, K is uniformly sampled from the range of $(40, 600)$, with sample interval 10. An increasing tendency can be drawn from Fig. 9 when $K \in [40, 310]$, that with the increase of K, the AUC increases. When $K = 310$, AUC reaches its maximum 0.998, that is very close to 1. After that, the decrease of AUC coincides with that of the maximum recall. To increase K will eliminate more false positives, however at the cost of discarding more true positives. Especially for circles with small radius, there is a high probability that they would be neglected when K is large.

Discretization is responsible for the phenomenon. Due to the discretization, the number of edge points along a circle of radius r is approximately $2\pi r$, that is the perimeter of the circle. For simplicity, we assume that the edge points along an ideal circle of radius r amount to $2\pi r$. The following analysis aims to deduce a loose upper-bound for the compatible function. First, each edge point can vote for at most three ks. The fact can be easily verified from the algorithm in Fig. 7. Thus there exist $6\pi r$ pairs of edge point and k. Each pair adds at most $\frac{1}{K}$ to the compatible function. The upper-bound is then $6\pi r/K$, which decreases with K. With the increase of K, the upper bound decreases. To overcome the defect, experiments show that $\in [100, 300]$ is a reasonable range, and we select K from the range for experiments. The reported results for $K = 100$ and $K = 300$ demonstrate the promising performance.

Table 2 Average time for per image.

K	50	100	200	300	400	500	600
Proposed(s)	0.97	1.02	1.14	1.25	1.32	1.43	1.49
HT by Tao(s)	1.55						

HT by Tao means the Hough transform by Tao.

To analyze the time complexity, we choose the Hough transform by Tao as the benchmark. It uses a 2D accumulator array for speed-up and is implemented in Matlab. For fair comparison, we coded our algorithm in the same environment and run the comparison of average time for per image on the same desktop with 2G memory and Core 2 Quad 2.5G cpu. The results are listed in Table 2.

The comparison shows that our algorithm has the comparable time complexity as Hough transform, and the conclusion can also be verified via the theoretical analysis. The minimizing operation in Fig. 7 is determined by the number of edge points E, the searching range. Given the number of edge points E, the minimizing operation for computing $\{D_k\}$ is less than $3E(1.2\frac{r_{hi}-r_{lo}}{N_r-1}+1)^2$, that is irrelevant to K. Thus the non-zero sites for $\{D_k|k = 1, \ldots, K\}$ is at most $3E(1.2\frac{r_{hi}-r_{lo}}{N_r-1}+1)^2$. The additive operation following the the batch calculation of $\{\mathcal{D}_k\}$ is less than $3EN_r(1.2\frac{r_{hi}-r_{lo}}{N_r-1}+1)^2$. Because of the design of objective function, the complexity is irrelevant to the number of sampled points K.

Besides, we show some localization results in Fig. 10, where we use 200 sampled points, and 0.1568 to threshold the compatible function.

5.2 Circular Seal Detection in Document Images

We also conducted experiment on the scanned documents for detecting circular seals. The experimental setup is the same as coin detection, except that some parameters are tuned to fit the task. Test images are all normalized to 1654, searching range uses $[10, 80]$, and the number of searching radii chooses 15. Other parameters are set using the same way as coin detection. We report the PR curves in Fig. 11 and demonstrate some of the localization results in Fig. 12. For these results, number of sampled points is $K = 300$; and threshold value accepting a circle is 0.074.

The proposed algorithm for both $K = 100$ and $K = 300$ works well and outperformances the others. For the dataset, the clutter is the main challenge. In a test image with clut-

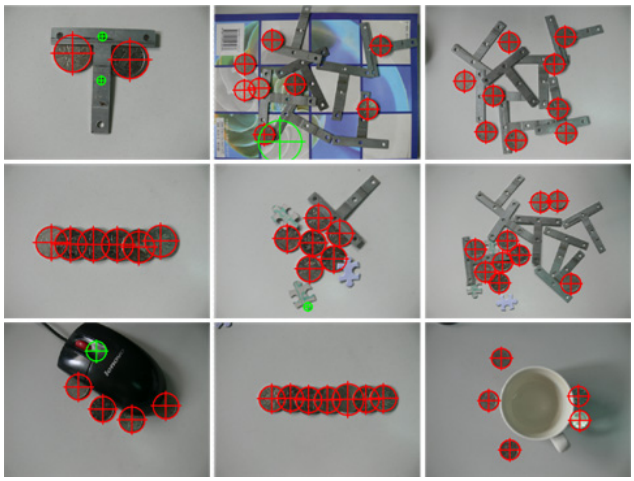


Fig. 10 Some localization results for coin detection. Red lines give the true positives, and green dashed lines are the false positives. The cup in the final example is also a circle, it is not included because its radius is outside the searching range: [10, 60].

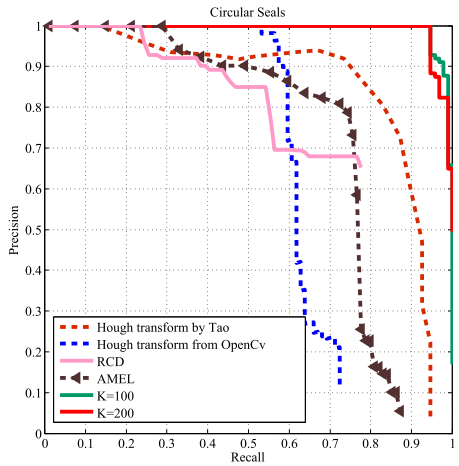


Fig. 11 Precision vs. recall curve for circular seal detection in document images.

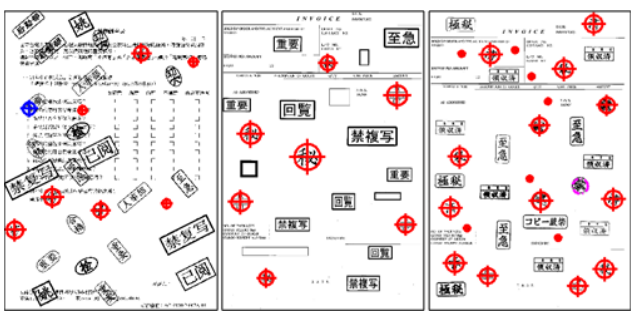


Fig. 12 Some of the localization results for document images. Red lines give the true positives, and the green lines are the false positives, magenta dotted line shows the missed one under current threshold value. For the three images, we missed one instance, and got one false positive.

tered background, the number of edge points coming from background overwhelms that from circular seals. The random sampling step in RCD usually misses the circles since the chance for choosing the edge points from circles decreases in the case. Thus the randomized characteristics of RCD makes it unsuitable for images with cluttered backgrounds. As for AMEL, the complex values casted by the background are aliased with those by the circular objects, that makes its performance degenerate. The results in Fig. 11 also show that compared to both Hough transforms, the proposed method introduces fewer false positives, and thus succeeds in suppressing noises led by cluttered background. On the other hand, the voting way used by both Hough transform accumulates the noises, thus a uniform threshold value may discard those circles with small radius and introduce false positives from clutter.

5.3 Circular Traffic Sign Detection in Natural Images

Challenge for the dataset comes from clutter and deformation. The circular traffic signs in the dataset present considerable deformations that are lead by out-of-plane rotation. The radii of these instances are all between 20 pixels and 120 pixels. In experiment, the setup for our algorithm keeps the same as the previous two, except that we use 21 discretized searching radii; and set both $\sigma_{k,x}$ and $\sigma_{k,y}$ to 0.8× the radius sample interval, i.e. 4 pixels. Besides, we introduce an auxiliary step to resolve the concentric circles. As shown in Fig. 14, the concentric circles are inevitable in traffic sign detection.

Let $A = (x_a, y_a, r_a)$ and $B = (x_b, y_b, r_b)$ denote two candidates. Without loss of generality, we assume that A's radius is greater than B's. With the assumption, if A and B are concentric, then it is necessary that B should be contained in the region enclosed by A. Here, we use a relaxation that requires the bounding box of A contains that of B. When the requirement is satisfied, it is easy to check that the maximum deviation between the two centers is $\sqrt{2}|r_a - r_b|$. Concentric degree for the two circles is defined by,

$$Conc(A, B) = \frac{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}{\sqrt{2}|r_a - r_b|}, \tag{18}$$

where, $\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$ measures the Euclidean distance between two centers, and $\sqrt{2}|r_a - r_b|$ serves as a normalization factor. Threshold for determining the concentric circles is set to 0.5 in the experiment. Besides, we restrict the variation of the two circles' radii, that $r_b \geq 0.6r_a$. Resolving the concentric circles significantly reduces the number of false positives, which can be drawn from Fig. 13. Proposed algorithm with concentric circles' resolving obviously outperforms that without the resolving step. At the same time, it outperformances the other algorithms.

In Fig. 14, we demonstrate some localization results. For these results, number of sampled points is 300; and threshold value accepting a candidate as circle is 0.133.

6. Conclusion

To summarize, we have proposed an algorithm based on

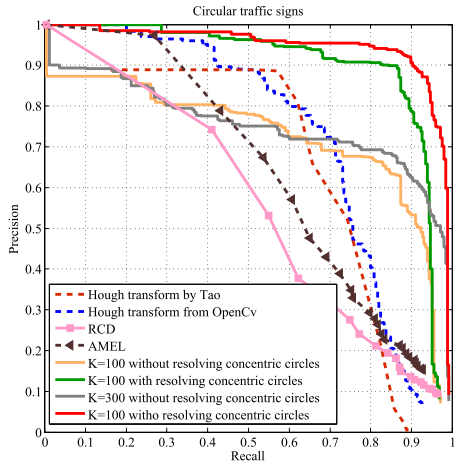


Fig. 13 Precision vs. recall curve for circular traffic sign detection.

Fig. 14 Some of the localization results for circular traffics signs. Red lines give the true positives, and the green lines are the false positives, magenta dotted line shows the missed one under current threshold value. For a candidate to be accepted as correct, its bounding box must agree with the ground truth bounding box based on an overlap score greater than 0.5.

voting for maximum compatibility for circle detection, that outperforms the methods based on Hough transform, and achieves the comparable time complexity as Hough transform using 2D accumulator array. In its fast implementation, the truncating operation plays the key role. It restricts the both the spatial range and orientational range that each edge point can vote for. Besides, it separates the calculation of objective function O_k into the radius-dependent and -independent parts. The radius-independent part is time-consuming, and shared by all searching radii for computation saving.

Experimental evaluations on three diverse datasets demonstrate the encouraging detection results among occlusion and clutter. Our Canny plus voting in maximizing way performs well and steadily for almost all types of images. Compared to Hough transform, it poses fewer constraints on the edge detection and achieves better results. The default parameters for Canny detector in Matlab are fine for all image types involved in the experiments.

There are several avenues for further research. Obviously, our method can be combined with some of the modifications summarized in Sect. 2. For example, the modulation of radii can be combined to further reduce the memory requirement. On the other hand, for fast implementation, our approach is designed that K for all radii are all the same. In the next stage, we are going to design a radius-dependent way for choosing K, and explore the underlying mechanism for fast implementation.

Acknowledgments

This work was supported by NSF of China (Grant No. 90920008 and 91120009). We would like to thank Xiaochun Wang for her valuable suggestions on the writing.

References

[1] Z. Zhu and Q. Ji, "Robust real-time eye detection and tracking under variable lighting conditions and various face orientations," Comput. Vis. Image Understand., vol.98, no.1, pp.124–154, 2005.
[2] X.F. Tong, H.Q. Lu, and Q.S. Liu, "An effective and fast soccer ball detection and tracking method," Proc. 17th International Conference on Pattern Recognit., 2004. ICPR 2004, pp.795–798, vol.4, Aug. 2004.
[3] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," 2004 IEEE Intelligent Vehicles Symposium, pp.566–571, June 2004.
[4] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," Pattern Recognit., vol.13, no.2, pp.111–122, 1981.
[5] J. Canny, "A computational approach to edge detection," IEEE Trans. Pattern Anal. Mach. Intell., vol.PAMI-8, no.6, pp.679–698, Nov. 1986.
[6] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," IEEE Trans. Pattern Anal. Mach. Intell., vol.30, no.7, pp.1270–1281, 2008.
[7] M.Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," 2010 IEEE Conference Computer Vision and Pattern Recognition (CVPR), 2010.
[8] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," ECCV Workshop on Statistical Learning in Computer Vision, pp.17–32, 2004.
[9] B. Leibe, A. Leonardis, and B. Schiele, "An implicit shape model for combined object categorization and segmentation," Toward Category-Level Object Recognition, ed. J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Lect. Notes Comput. Sci., vol.4170, pp.508–524, 2006.
[10] A. Opelt, A. Pinz, and A. Zisserman, "A boundary-fragment-model for object detection," Computer Vision – ECCV 2006, Springer Berlin/Heidelberg, 2006.
[11] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection," Int. J. Comput. Vis., vol.87, no.3, pp.284–303, 2010.
[12] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of Hough transform methods for circle finding," Image Vis. Comput., vol.8, no.1, pp.71–77, 1990.
[13] G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software

Tools, 2000.
[14] R.K. Yip, P.K. Tam, and D.N. Leung, "Modification of Hough transform for circles and ellipses detection using a 2-dimensional array," Pattern Recognit., vol.25, no.9, pp.1007–1022, 1992.
[15] D.J. Kerbyson and T. Atherton, "Circle detection using Hough transform filters," 1995 Fifth International Conference on Image Processing and Its Applications, pp.370–374, July 1995.
[16] T. Atherton and D.J. Kerbyson, "Using phase to represent radius in the coherent circle Hough transform," IEE Colloquium Hough Transforms, pp.5/1–5/4, May 1993.
[17] T.J. Atherton and D.J. Kerbyson, "Size invariant circle detection," Image Vis. Comput., vol.17, no.11, pp.795–803, 1999.
[18] V.F. Leavers, "The dynamic generalized Hough transform: Its relationship to the probabilistic Hough transforms and an application to the concurrent detection of circles and ellipses," CVGIP: Image Understanding, vol.56, no.3, pp.381–398, 1992.
[19] D. Ioannou, W. Huda, and A.F. Laine, "Circle recognition through a 2D Hough transform and radius histogramming," Image Vis. Comput., vol.17, no.1, pp.15–26, 1999.
[20] T.C. Chen and K.L. Chung, "An efficient randomized algorithm for detecting circles," Comput. Vis. Image Understand., vol.83, no.2, pp.172–191, 2001.
[21] S.H. Chiu and J.J. Liaw, "An effective voting method for circle detection," Pattern Recognit. Lett., vol.26, no.2, pp.121–133, 2005.
[22] C.T. Ho and L.H. Chen, "A fast ellipse/circle detector using geometric symmetry," Pattern Recognit., vol.28, no.1, pp.117–124, 1995.
[23] Y. Lei and K.C. Wong, "Ellipse detection based on symmetry," Pattern Recognit. Lett., vol.20, no.1, pp.41–47, 1999.
[24] N. Guil and E.L. Zapata, "Lower order circle and ellipse Hough transform," Pattern Recognit., vol.30, no.10, pp.1729–1744, 1997.
[25] K. Carolyn, D. Ballard, and J. Sklansky, "Finding circles by an array of accumulators," Commun. ACM, vol.18, pp.120–122, Feb. 1975.
[26] E. Zelniker and I. Clarkson, "Maximum-likelihood estimation of circle parameters via convolution," IEEE Trans. Image Process., vol.15, no.4, pp.865–876, April 2006.
[27] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," Pattern Recognit. Lett., vol.11, no.5, pp.331–338, 1990.
[28] W. Lu and J. Tan, "Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT)," Pattern Recognit., vol.41, no.4, pp.1268–1279, 2008.
[29] H.J. Wolfson and I. Rigoutsos, "Geometric hashing: An overview," IEEE Comput. Sci. Eng., vol.4, no.4, pp.10–21, 1997.
[30] K.L. Chung, Y.H. Huang, S.M. Shen, A.S. Krylov, D.V. Yurin, and E.V. Semeikina, "Efficient sampling strategy and refinement strategy for randomized circle detection," Pattern Recognit., vol.45, no.1, pp.252–263, 2012.
[31] X. Hilaire and K. Tombre, "Robust and accurate vectorization of line drawings," IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.6, pp.890–904, June 2006.
[32] B. Lamiroy, L. Fritz, and O. Gaucher, Robust circle detection, pp.526–530, IEEE Computer Society, Los Alamitos, CA, USA, 2007.
[33] H. Li, M.A. Lavin, and R.J.L. Master, "Fast Hough transform: A hierarchical approach," Comput. Vis. Graph. Image Process., vol.36, no.2-3, pp.139–161, 1986.
[34] G. Schuster and A. Katsaggelos, "Robust circle detection using a weighted MSE estimator," 2004 International Conference Image Process., vol.3, pp.2111–2114, Oct. 2004.
[35] V. Ayala-Ramirez, C.H. Garcia-Capulin, A. Perez-Garcia, and R.E. Sanchez-Yanez, "Circle detection on images using genetic algorithms," Pattern Recognit. Lett., vol.27, no.6, pp.652–657, 2006.
[36] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," IEEE Trans. Pattern Anal. Mach. Intell., vol.10, no.6, pp.849–865, Nov. 1988.
[37] P.F. Felzenszwalb and D.P. Huttenlocher, "Distance transforms of sampled functions," Cornell Computing and Information Science TR20041963, 2004.
[38] T. Peng, http://www.mathworks.com/matlabcentral/fileexchange/9168-detect-circles-with-various-radii-in-grayscale-image-via-hough-transform, 2010.

Yuanqi Su received his B.E. degree in mechanical and automatic engineering and M.E. in pattern recognition and intelligent system at the Xi'an Jiaotong University, Xi'an, China in 2003 and 2006 respectively. He is currently involved in doctoral research studies in the institute of artificial intelligence and robotics, Xi'an Jiaotong University, Xi'an, China. His research interests include image processing, computer vision and computer graphics.

Yuehu Liu received his B.E. and M.E. degree from the Xi'an Jiaotong University, China in 1984 and 1989, respectively, and the Ph.D. degree in electrical engineering from Keio University, Japan, in 2000. He is now a professor of Xi'an Jiaotong University, Xi'an China. His research focuses on computer vision, computer graphics and digital content analysis. He is currently member of the IEEE.

Xiao Huang received his B.E. degree from the Xi'an Jiaotong University, China in 2009. He is currently involved in doctoral research studies in the institute of artificial intelligence and robotics, Xi'an Jiaotong University, Xi'an, China. His research interests include computer vision and computer graphics.