Pattern Recognition 81 (2018) 112–130

Contents lists available at ScienceDirect

Pattern Recognition

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

Accurate detection of ellipses with false detection control at video rates using a gradient analysis

Huixu Dong[a,*], Dilip K. Prasad[b], I-Ming Chen[a]

[a] Robotics Research Centre, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
[b] School of Computer Science Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 22 October 2017
Revised 1 December 2017
Accepted 23 March 2018
Available online 29 March 2018

Keywords:
Ellipse detection
Geometric approach
Gradient analysis
Centre estimation
Arc classification

ABSTRACT

Accurate ellipse detection in image streams at real-time execution is an open challenge. We present a novel fast and robust ellipse detection method. The method adopts arcs selection, smart grouping, and repeated utilization of gradient information to significantly reduce the computations otherwise needed without compromising the detection effectiveness. Geometric properties calculable with few computations, such as arc smoothness, relative placement of curves, and region of confidence for ellipse centres, are utilized for this purpose. An exhaustive sensitivity analysis of the method's control parameters has been performed. It reveals range of values that support consistent performance over diverse challenging datasets with complex background, multiple differently sized ellipses, and occluded, overlapping ellipses. The method's performance is compared with six state-of-the-art detectors over four diverse datasets. Among all the tested methods, the proposed method demonstrates the best balance between detection effectiveness (the best or the second best F-measure scores) and computation time (>40 Hz) across all the datasets.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

An ellipse (or a circle) is a common man-made nonlinear geometric shape. The ellipse detection from image streams is a common need in a variety of applications such as industrial inspection [1], medical diagnosis [2,3], recognition of traffic signs [4], security [5], face recognition [6,7], and object tracking for a robotic platform [8–10]. Thus, it is important to realize a robust ellipse detection method for real images and image streams. However, ellipse detection is a challenging task. The presence of noise substantially overwhelms edge pixels of real ellipses and breaks an ellipse's boundary into multiple disconnected arc segments. This issue, in addition to complex background, causes degradation in the performances of the existing ellipse detection methods in terms of either the detection accuracy or the execution time, often both. Moreover, efforts to improve detection accuracy often result in longer execution time, while algorithmic efforts in reducing execution time often compromise the detection accuracy. In our paper, we focus on improving the ellipse detection performance, in terms of both the detection accuracy and the execution time, even while dealing with occluded ellipses, overlapping ellipses, and incomplete ellipses in images taken in practical scenarios.

1.1. Related work

Five-dimensional Hough Transform (HT) is a classical method for detecting ellipses. Regardless of the robustness, the standard Hough transform [11] involves a large amount of computational cost. To improve the computational efficiency of HT, various HT-based approaches [12–16], such as Randomized Hough Transform (RHT) [15] and Probabilistic Hough Transform (PHT) [16], have been developed. However, these variants of HT cannot reach video rate computation speeds due to the process of voting among numerous candidates. Hybrid methods that combine Hough Transform and other geometric approaches have also been proposed to overcome the shortcomings of Hough Transform. For instance, Cakir et al. [17] combine feature-based models with HT to improve the detection accuracy. Chen et al. [18] integrate the advantages of HT and edge segment detection, described next, which allows its applications to industrial scenarios.

Another class of ellipse detection approaches is edge segment detection techniques. Methods in this class exploit the connectivity between edge pixels to detect ellipses. The main steps of these methods are to extract arcs and then group them by exploiting geometric or algebraic properties of ellipses. Specifically, edge arcs that may consist of an ellipse are found by variety of techniques such as the statistical regression method [19], curve segmentation by fitting a set of short line segments on edges [20–26], connec-

[*] Corresponding author.
E-mail address: dong0076@e.ntu.edu.sg (H. Dong).

https://doi.org/10.1016/j.patcog.2018.03.023
0031-3203/© 2018 Elsevier Ltd. All rights reserved.

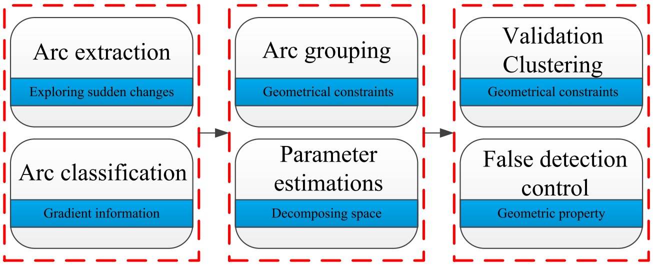

Fig. 1. The flowchart of the proposed detector.

tivity and curvature conditions [22,27]. Subsequently, these arcs are grouped according to the convexity-concavity [21,28,29], arc curvature [21], and geometric constraints [1,30–33]. Lastly, ellipses are fitted on the grouped arc candidates using RANSAC [19,32] or least squares fitting technique [28]. Prasad et al. [21] use arc convexity to determine search regions where suitable arc candidates for grouping may be located. Fornaciari et al. [28] use geometric constraints as a selection strategy of arcs belonging to the same ellipse and estimate parameters by decomposing the parameter space. Mai et al. [22] adopt line segments to approximate potential elliptic-arcs according to connectivity and curvature conditions. Bai et al. [29] use the property of elliptical concavity to group arcs for identifying candidate ellipses. Liu et al. [33] propose a hierarchical approach which is motivated by the fact that any segment of an ellipse can identify itself in ellipse reconstruction by geometric constraints. The edge segment detection methods are regarded as the most effective in detecting multiple ellipses in digital images and require less computational cost than HT-based methods. However, because of dependence on the preciseness of arcs detected, the detection accuracy drops when these methods are applied to images containing complicated occluded contours.

Many algebraic functions or geometric characteristics have been utilized for selection of arc segments to improve the performances of ellipse detectors. For instance, symmetry of arcs in image is applied for decomposing the parameter space and reducing the clustering parameters [2,34–36]. Distance measures are used to perform registration of a set of arcs [36–38]. Mulleti et al. [39] rely on the sampling structure of finite-rate-of-innovation signals to estimate the parameters of ellipses from partial data. Pătrăucean et al. [40] propose a line segment and elliptical arc detector without any parameter tuning based on a contrario theory. Chang Liu et al. [41] fit the elliptical projections of the cross sections of a surface of revolution with the given intrinsic matrix of a camera. Jia et al. [42] use a projective invariant operator to significantly prune the undesired candidates and pick out elliptical ones. Points and tangents together are regarded as the basic units of information to find an ellipse in Ref. [43]. The aforementioned methods can detect ellipses with less bias, but the performances of these methods are inadequate to detect occluded and overlapping ellipses when input images contain a large amount of noise.

1.2. Proposed method

In this paper, we propose a novel fast and robust ellipse detector based on geometric properties of elliptic arcs and gradient analysis, which relieves the shortcomings of the ellipse detectors presented previously. The work provides improvement of execution speed and accuracy of ellipse detection, providing a good balance between these two important and often conflicting aspects of performance of ellipse detection algorithms. The schematic flowchart of the proposed method is presented in Fig. 1. We briefly describe them below to create context for the discussion of novelty.

Arc extraction and classification: After obtaining the edge image from a given image, we pre-process the edge image to remove straight lines and noise, which speeds up the subsequent procedures. We first classify pixels in edge image according to the orientations of the gradients at the edge pixels and then compute curves (or arcs) as sets of 8-connected edge pixels. Building upon the idea of edge segment detection, we use line segments to fit a curve and then calculate the intersection angles between consecutive line segments to confirm sudden changes where the curve is split into potentially elliptic arcs. Since this procedure may generate short arcs or discrete pixels that do not have enough curvature to constitute an ellipse, we identify such curves and remove them in the pre-processing stage itself.

Ellipse detection: The gradient information at edge pixels is used for classifying the arc candidates into four quadrants. Then, we utilize the fact that arcs belonging to an ellipse have the same ellipse centre and their positions have well defined placement in the space to define two geometric constraints, which permit identification of sets of arcs potentially belonging to the same ellipses. We confirm the centre of such a set of arcs using a mean-shift clustering algorithm and obtain other parameters in a decomposed parameter space. The existing methods that determine ellipse centres by voting or accumulating in a two/dimensional parameter space suffer in the presence of noise as the noise reduces the accumulation of votes at the actual centre of the ellipses. Unlike these methods, the proposed method of confirming the ellipse centre using mean/shift clustering can realize high accuracy based on a statistical principle.

Ellipse verification: After detecting ellipses from all the sets of arcs, validation and clustering procedures can extract ellipse candidates with high confidences. Finally, a strategy for false detection control based on geometric property of ellipse is used to alleviate false detections.

1.3. Novelties of the proposed method

The strengths of this work are derived from several novelties across the algorithmic steps, each having a specific impact on the performance of our method. The guiding principle behind the novel features of our proposed method is to achieve good performance of the current task with minimum computations and alleviate the computational requirements of the subsequent tasks. This principle helps us to reduce execution time as compared to contemporary edge following methods significantly. The novelties are enumerated below:

1) This paper presents an ellipse detection method that combines the advantages of arc extraction and arc grouping. Arc detection involves important arc splitting steps, where our novel propositions given in points 2 and 3 below help in precise arc detection at low computation expense. Arc grouping using geometric constraints, discussed more in points 4 and 5 below, guarantees the effectiveness of ellipse detection and optimizes the computation cost.
2) In the step of smooth arc extraction, we propose a novel approach of identifying the precise splitting points (sudden changes) in order to achieve better segmentations from curves to smooth arcs that may belong to ellipses. A coarse search for sudden changes is first performed with a big range, and then such points are determined with a finer scope.
3) We use just one inequality with angle vectors to confirm splitting points (sudden changes), as compared to more than one inequalities used by other contemporary methods. This provides advantage of computation speed over other methods.
4) The direction of vector formed by bounding box enclosing an arc is used for determining a convexity-concavity of arc pre-

cisely to classify an arc further. This speeds up the efficiency of grouping of arcs.

5) We adopt the ratio of half of the circumference of the bounding box enclosing an arc and the sum of the semi-axes lengths to measure the integrity of ellipse to improve the detection accuracy. This is significantly faster than other more complicated approaches of measuring the integrity of the ellipse.
6) We propose a new approach of false determination control to determine detection results based on the intrinsic geometric attribute of ellipse expressed by a mathematical model, which avoids false detections effectively.

1.4. Outline of the paper

The rest of this paper is organized as follows. Section 2 presents a new method of arc extraction by confirming sudden changes where a curve is to be split. Two geometric constraints applied for identifying sets of arcs potentially belonging to the same ellipse and the method of determining ellipse parameters are presented in Section 3. Section 4 presents the procedures of ellipse verification and clustering. A strategy with false detection control is provided in Section 5. Section 6 presents experimental results including (a) parameter tuning and sensitivity analysis experiments, (b) ellipse detection experiments for synthetic images and real images with complex scenarios, and (c) validation of the superior performance of the proposed method in comparison with six state-of-the-art ellipses detectors. Finally, the work is concluded in Section 7.

2. Arc extraction and classification

2.1. Pre-processing stage

The pre-processing stage helps to speed up the subsequent processes and improve the computational efficiency [28,32,42]. First, Canny edge detector [44] with auto-thresholding is applied to obtaining the edge image for an input image. Edge pixels are denoted as $p_i = (x_i, y_i, \eta_i)$, where $i = 1, 2, \ldots, N$, (x_i, y_i) are the coordinates of the ith edge pixel, η_i is the gradient angle at the edge pixel computed using Sobel derivatives [28], and N is the number of edge pixels. Subsequently, to obtain non-branched edge curves, the curves are classified into two classes based on edge directions. Due to the discrete characteristics of the digital image, the accurate gradient direction of the curve at a pixel [45] can not to be calculated. However, we are interested in the general orientation of the gradient and not in the exact gradient vector. Therefore, we use only the sign of the gradient as representative of the general orientation of the gradient. Each edge point p_i is assigned with either a positive or negative direction by the following gradient's sign function $\mathrm{X}(p_i)$,

$$\mathrm{X}(p_i) = \begin{cases} +, \mathrm{sign}(dx_i) \cdot \mathrm{sign}(dy_i) > 0 \\ -, \mathrm{sign}(dx_i) \cdot \mathrm{sign}(dy_i) < 0 \end{cases} \quad (1)$$

The function $\mathrm{X}(p_i)$ becomes zero when the Sobel derivatives dx or dy are zero, i.e. the gradients are along the vertical or horizontal directions. Thus, after the classification of curves, an edge pixel p_i whose gradient sign function is zero are removed from the edge image.

The 8-connectivity of two consecutive edge points in the same direction is used to extract connected edge curves. Since short curves may be a result of noise and are inconsequential in ellipse detection due to the lack of curvature, we identify and remove short edge curves as described next. We use an oriented bounding box with the minimum area, which is denoted as OBB_{min}, to enclose a curve [46]. If the ratio of the long side to short side of OBB_{min} is more than a threshold $thre_r$, we can discard this curve since it may be a line segment or its radius is too big, implying that the centre of the ellipse is not within the image. Moreover, if the area of OBB_{min} is less than a threshold $thre_a$, this curve is also readily removed because it may be a result of noise or a line segment that is not salient enough to contain information pertaining an ellipse. The remaining curves are deemed to be curve candidates that may make up ellipses and are used in the subsequent steps for ellipse detection, as shown in Fig. 2. All the thresholds are investigated in Section 6.

2.2. Arc extraction

It is well known that the edge curvature of an ellipse changes continuously and smoothly. To detect ellipses, we need to obtain the edge curves with continuous and smooth curvature. A smooth curve does not have a sudden change in curvature, characterized by the amount of change or the direction of change [21]. In this paper, the amount of change of curvature and the direction change of curvature are generally referred to as the turning corner and inflexion point, respectively. All the points on any curve have the same gradient orientations after extracting the edge image by Canny edge detector [44] with auto-thresholding. There exist two additional cases of sudden changes in edge curves, as shown in Fig. 3.

We use a series of line segments to fit an edge curve to obtain an approximate representation of the curvature of the edge curve [18,21,31,33,47,48]. These line segments are used for extracting smooth edge curves. Specifically, depending on the pixel order from left to right on a curve, we explore points, such as the turning corner and inflexion point, which introduce irregularities in the curvature, and break the edge curve at those points. The methods that explore sudden changes [18,21,31,33,47,48] cannot do so with precision since they investigate the changes only at the dominant points of the curve obtained by fitting the line segments rather than every point on the curve. We present an approach for exploring points from a large range to a small scope for dealing with this issue.

First, we perform a coarse search for turning corners and inflexion points. For this, we adopt the method presented by [21] to fit a series of line segments $\{l_1, l_2, l_3, \ldots, l_n\}$ on an edge curve. θ_i denotes the angle between a pair of consecutive line segments $\{l_{i-1}, l_i\}, i = 2, 3, \ldots, n$, and the direction of θ_i is from l_{i-1} to l_i. In terms of the curvature of an edge curve, the value of angle θ_i is an indicator of the change. If θ_i is small, it indicates that the amount of change in curvature is small so that a part of a curve fit by line segments is considered as collinear ones. If the angle θ_i is too big, the point p_i joining two consecutive line segments is regarded as a turning corner (Fig. 4(A)). Specifically, if the angle θ_i satisfies the following inequalities, the point p_i is determined as a turning corner,

$$|\theta_i| > thre_\theta.$$

The sign of the angles formed by a series of consecutive line segments represents the direction of the curvature in a curve. In other words, if the sign of an angle θ_i is different from that of the previous angle θ_{i-1}, it implies that there is a change in the direction of a curve. If θ_{i-1} and θ_i have the same signs, their relationship conforms to the expression $||\theta_i| - |\theta_{i-1}|| = |\theta_i - \theta_{i-1}|$. We use the following inequality to identify an inflexion point p_i in order to make a coarse decision on where the edge curve may be split, as shown in Fig. 4(A),

$$||\theta_i| - |\theta_{i-1}|| < |\theta_i - \theta_{i-1}|.$$

Thus, the turning corners and inflexion points determined so far may be such points at which an edge curve needs to be split to obtain smaller smooth curves.

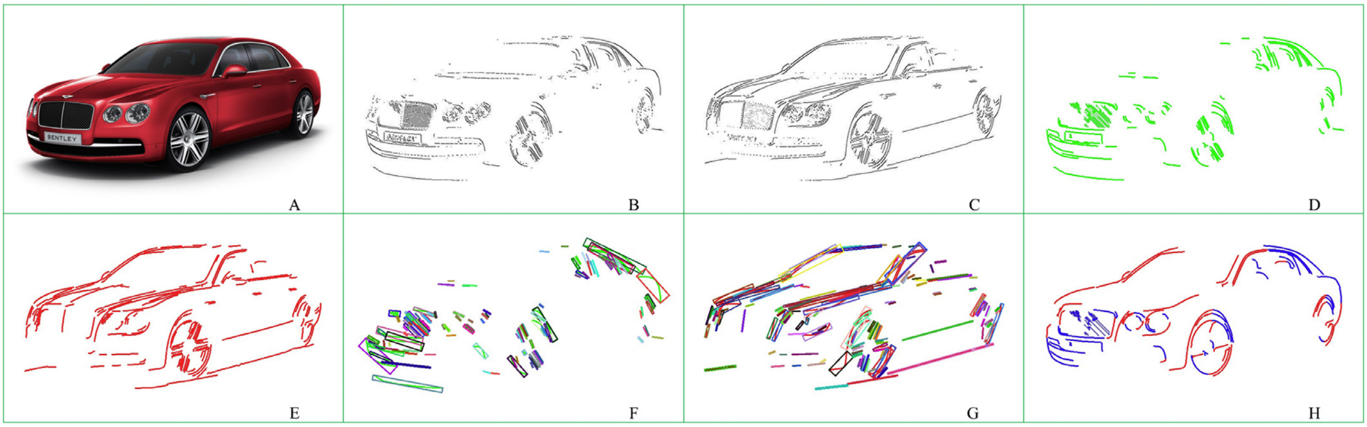

Fig. 2. The initial image(A); the edge images along two edge directions by gradient signs (B,C);the edge images after the process of 8-connectivity(D,E);the arcs enclosed by oriented bounding boxes(F,G);the image by a pre-processing procedure(H).

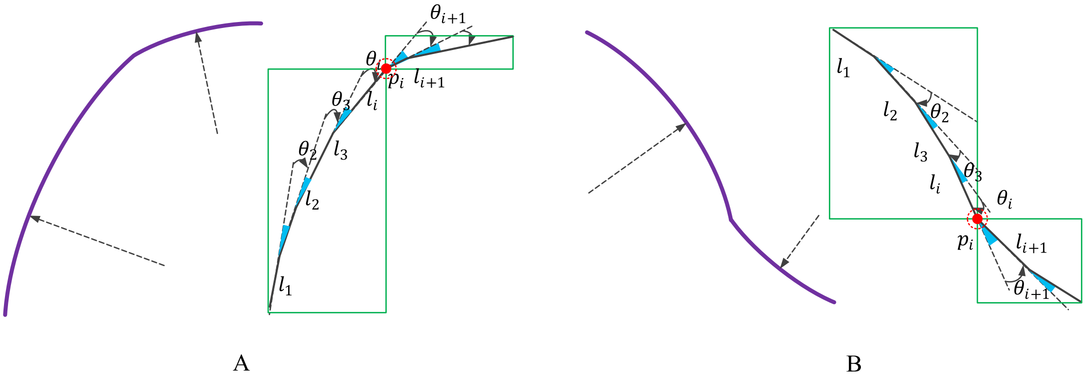

Fig. 3. The curves with the large change of curvature (A) and the change of direction of curvature (B). The green frames represent bounding boxes enclosing the split arcs. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

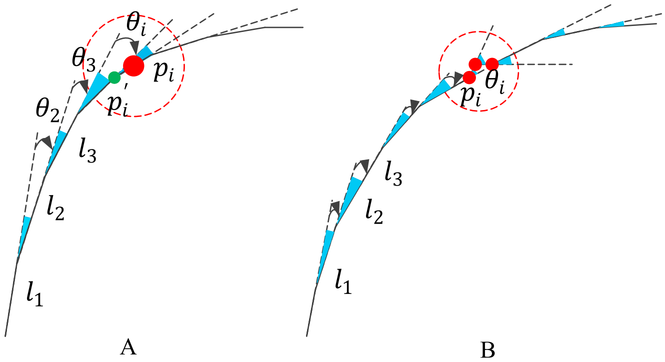

Fig. 4. The exploration of an accurate sudden change (A) and the over-splitting case due to over-many sudden changes (B). The green dot represent a rough sudden change (A); the red one donates the precision sudden change (A); the three red dots that indicate three sudden changes lead to an over splitting (B). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Second, by exploring a certain number of pixels on either side of the coarsely determined points of sudden changes, we can obtain points of sudden change precisely. The gradient η_i of pixels is applied to calculating the tangent angle of each pixel [32,40]. The two inequalities above are then applied for precise determination of the points that are turning corners or inflexion points, as shown in Fig. 4(B). Since the values of η_i have been computed in the pre-processing step already, the extra computations for precise determination of points of sudden changes is meagre. Moreover, image noise around the edge pixels may introduce several turning corners or inflexion points on an otherwise smooth curve [18], thus resulting into several small arcs after the splitting that are not salient for ellipse detection. In order to reduce the subsequence processing time, we reuse the pre-processing approach to removing arcs enclosed by small oriented bounding boxes. Table 1 provides the specific algorithm for extracting arcs that may consist of an ellipse.

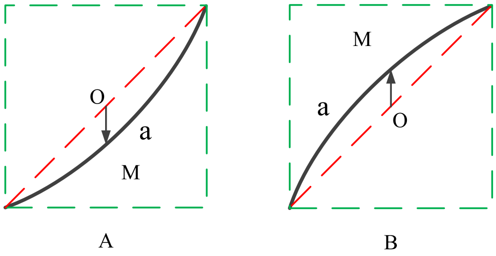

Fig. 5. The convex arc (A) and the concave arc (B).

2.3. Arc classification based on the convexity-concavity

The convexity-concavity property of each arc is used for classifying arcs, which are now smooth curves of sufficient length, having no pixels with gradients along the horizontal or vertical directions and no sudden changes in the curvature. The arcs are enclosed by bounding boxes such that the edges of the bounding boxes are along the horizontal or vertical directions, as shown in Fig. 5. This is different from the previously used bounding boxes

Table 1
The algorithm of smooth arc extraction.

```
Smooth arc extraction by detecting the turning corners and inflexion points
Input: $thre_\theta$, curves,$thre_a$
Output: smooth curves consisting of an ellipse
For $i = 0$ to the size of curves do
    Fit curve $i$ by line segments;
    Calculate the angles $\theta$ formed by a series of consecutive line segments
    For $j = 1$ to the size of line segments do
      If $(abs(\theta_j) > thre_\theta \ or \ abs(abs(\theta_j) - abs(\theta_{j-1})) < abs(\theta_j - \theta_{j-1}))$
        For $m = 0$ to the size of the $i^{th}$ curve do
          If (line_segments[$j$].x==curve_$i$[m] .x && line_segments[$j$].y==curve_$i$[m].y)
            For $h$ =curve[$i$].(x-5) to 10 (from curve[$i$].(x-5) to curve[$i$].(x+5)) do[1]
              Calculate the tangent angle $\delta_h$ of the pixel
              If $(abs(\delta_h) > thre_\theta \ or \ abs(abs(\delta_j) - abs(\delta_{j-1})) < abs(\delta_j - \delta_{j-1}))$
                  Reserve these points in the container $F$;
            Reserve points with the maximum $\delta_h$ chosen from the container $F$ in container $E$;
  For $k = 0$ to the size of $E$ do
    For $n = E[k]$ to $E[k+1]$ do
      Obtain the area $\Psi_n$ of each bounding box that encloses curve[$i$][$k$]
      If $(\Psi_n > thre_a)$
        Reserve curve [$i$][$k$] in the container $C$.
```

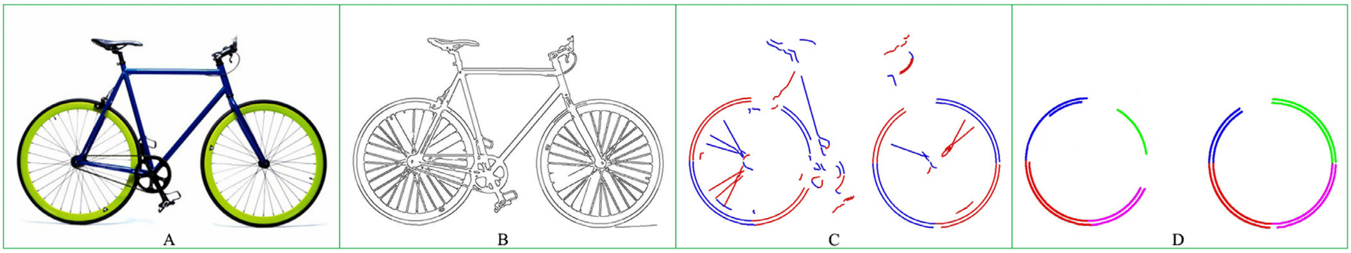

Fig. 6. The original image (A); the edge image(B); arc classification by the gradient directions (C); arc classification in the four quadrants via the convexity-concavity(D).

of minimum area enclosing a curve snugly and may be oriented in the image space randomly, as seen in Fig. 2(F,G). For such bounding box with edges aligned with the horizontal and vertical directions fitted on an arc e, O represents the centre of the bounding box and the middle point of the line segment (diagonal) jointing two endpoints of an arc. M is the midpoint of the arc along the horizontal direction, i.e. it is the point of intersection of the arc with a line joining the midpoints of the horizontal edges of the bounding box. If the vector direction δ of $\overrightarrow{OM}$ is positive, the arc is labelled as convex, otherwise as concave. We define a function $\Phi(e)$ to describe the convexity-concavity of an arc as follows,

$$\Phi(e) = \begin{cases} +, \ \delta > 0; \\ -, \delta < 0. \end{cases} \tag{2}$$

Based on the functions $\mathrm{X}(p_i)$ (see Eq. (1)) and $\Phi(e)$ (see Eq. (2)), an arc extracted can be classified into four quadrants (Fig. 6),

$$\Psi(a) = \begin{cases} \text{I}, \ \langle \mathrm{X}(p_i), \Phi(e) \rangle = \langle +, + \rangle \\ \text{II}, \ \langle \mathrm{X}(p_i), \Phi(e) \rangle = \langle -, + \rangle \\ \text{III}, \ \langle \mathrm{X}(p_i), \Phi(e) \rangle = \langle -, - \rangle \\ \text{IV}, \ \langle \mathrm{X}(p_i), \Phi(e) \rangle = \langle +, - \rangle \end{cases} \tag{3}$$

where e represents an arc with a sequence of connected pixels $\{(p_1, p_2, \ldots, p_i, \ldots, p_n)\}$, and I, II, III, IV denote the first, second, third, fourth quadrants, respectively. We note that the entire arcs are used and not just the line-segmented approximations of arcs identified in Section 2.2. These quadrants denote the quadrants in which the edge curves would lie if they belonged to a hypothetical ellipse and the centre of such ellipse is used as the origin. See example in Fig. 6 for an illustration.

3. Ellipse detection

Based on the arc classification described above, we choose two arcs $\tau_{ab} = (a_a, a_b)$ from the arcs distributed in four quadrants as a quadrant constrained arc set, which indicates that the arcs likely belong to the same ellipse. Subscripts a and b represent two distinct quadrants. Accordingly, six combinations of a quadrant constrained arc set can be listed as follows, $(a_{\mathrm{I}}, a_{\mathrm{II}})$, $(a_{\mathrm{I}}, a_{\mathrm{III}})$, $(a_{\mathrm{I}}, a_{\mathrm{IV}})$, $(a_{\mathrm{II}}, a_{\mathrm{III}})$, $(a_{\mathrm{II}}, a_{\mathrm{IV}})$, and $(a_{\mathrm{III}}, a_{\mathrm{IV}})$, as shown in Fig. 7. We further retain only those sets that satisfy the constraint of relative position and calculate the centres of the ellipses that likely constitute these arc sets as their edges. Then, we create super-sets of these arc sets based on the proximity of the centres estimated by the arc sets. This is performed on the basis that if two centres estimated by two arcs lie in an area with an acceptable range, it is highly likely that such arc sets belong to the same ellipse.

3.1. The selection of arcs consisting of an ellipse

The function Ω is defined to describe the constraint on relative position between two arcs to discard an arc set whose arcs cannot make up the same ellipse [28] as follows,

$$\Omega(a_a, a_b) = \begin{cases} {a_I}^l(x) - {a_{II}}^r(x) & if \ (a_a, a_b) \in (\text{I, II}) \\ min\Big({a_{\mathrm{I}}}^r(y) - {a_{\mathrm{III}}}^l(y), {a_I}^r(x) & if \ (a_a, a_b) \in (\text{I, III}) \\ \ - {a_{III}}^l(x)\Big) & \\ {a_I}^r(y) - {a_{IV}}^l(y) & if \ (a_a, a_b) \in (\text{I, IV}) \\ {a_{II}}^l(y) - {a_{III}}^l(y) & if \ (a_a, a_b) \in (\text{II, III}) \\ min\Big({a_{II}}^l(y) - {a_{IV}}^r(y), {a_{IV}}^l(x) & if \ (a_a, a_b) \in (\text{II, IV}) \\ \ - {a_{II}}^r(x)) & \\ {a_{IV}}^l(x) - {a_{III}}^r(x) & if \ (a_a, a_b) \in (\text{III, IV}) \end{cases}$$

where $a_i^l(x)$, $a_i^r(x)$, $a_i^l(y)$, and $a_i^r(y)$ represent the arc coordinates of the left most and the right most endpoints along the x and y directions in the i^{th} quadrant, respectively. In Fig. 7, along the horizontal axis x, ${a_{\mathrm{I}}}^l(x)$ must be bigger than ${a_{\mathrm{II}}}^r(x)$ and ${a_{\mathrm{III}}}^r(x)$ that must be smaller than ${a_{\mathrm{IV}}}^l(x)$. Similarly, along the vertical axis y, ${a_{\mathrm{I}}}^r(y)$ must be bigger than ${a_{\mathrm{III}}}^l(y)$ and ${a_{\mathrm{IV}}}^r(y)$ which are smaller than ${a_{\mathrm{II}}}^l(y)$. If $\Omega(a_a, \ a_b)$ is less than the tolerance $thre_p$, the set is discarded as

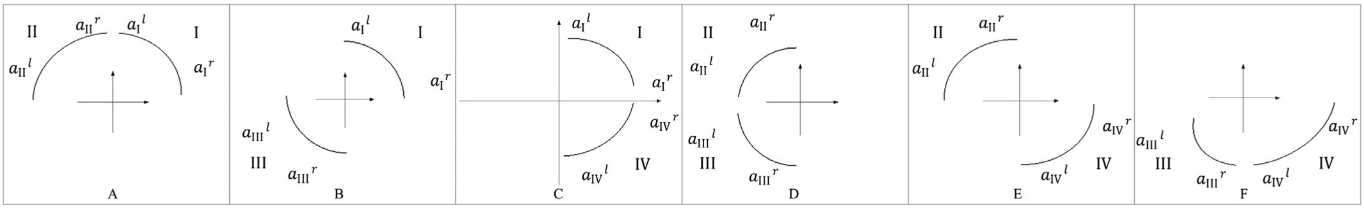

Fig. 7. The four grouping phases. I, II, III and IV represent the first, second, third and fourth quadrants. $a_i{}^l$ and $a_i{}^r$ indicates the left and right endpoints of an arc and i is the number of the quadrant where an arc rests.

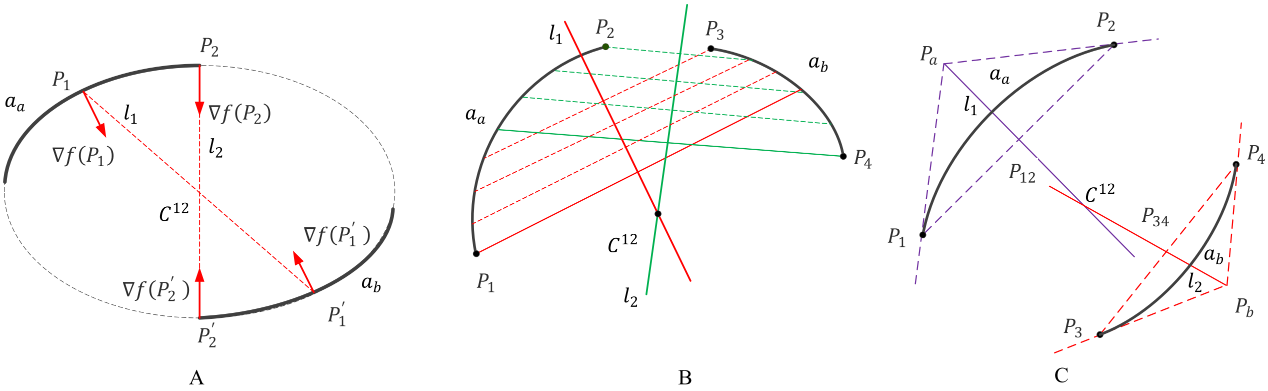

Fig. 8. Illustration of the computation of the elliptic centre. $\nabla f(P_1)$,$\nabla f(P_2)$, $\nabla f(P_1^{'})$, $\nabla f(P_2^{'})$ are the grandient vectors of the arc points P_1,P_2,$P_1^{'}$, and $P_2^{'}$,respectively; l_i denote the line formed by the points P_i and $P_i^{'}$ that are symmetrical regarding the centre of an ellipse,$i = 1, 2$(A). l_1 and l_2 represent the lines through the middle points of parallel chords formed by the arcs a_a and a_b, respectively(B). P_a and P_b are the intersection points of the tangent lines through the end points (P_1,P_2,P_3,P_4) of the arcs a_a and a_b,respectively; l_1 and l_2 represent the lines passing the middle points (P_{12}, P_{34}) of the line segments (P_1P_2, P_3P_4) and the intersection points (P_a, P_b),correspondingly(C). C^{12} denotes the intersection point of the lines l_1 and l_2.

improbable for fitting ellipse. The value of $thre_p$ is set as 1 since the pixels on the coordinate axes have been already discarded.

3.2. Confirmation of ellipse centre

The geometric characteristics used for retrieving the centres of the ellipses are presented here. For an ellipse, it is obvious that the gradient vectors of the edge points that are symmetrical with respect to the centre are parallel or antiparallel [49–51], as shown in Fig. 8(A). We further use the following geometric property of ellipses. The intersection of two lines passing the midpoints of two different sets of parallel chords at a pair of arcs is the ellipse centre [5,28,42], as shown in Fig. 8(B). The approach to calculate the ellipse centre by parallel chords is robust to outliers with a small cost in computational efficiency. However, this approach is ineffective if the lengths of arcs are very small. Specifically, if the number of points on an arc is small, the distance between two nearest parallel chords formed from that arc alone is also tiny. In this case, the noise has influence on the centre position. The geometric properties of points and tangents in ellipses, shown in Fig. 8(C), may be instead used to find ellipse centres [12,21,43,49,52]. In this approach, the ellipse centre is the intersection of lines connecting the middle points of chords and the intersection formed by tangent lines through the ends of arcs. Although this method can detect the centre of a small ellipse, the slopes of tangent lines could contain many errors due to the image noise and the digitization of the image [25].

Based on the considerations above, we propose a new approach for calculating the ellipse centre here. Unlike the geometric methods in [21,28], the proposed method need not consider the size of the ellipse or precise tangent information for robustness to noise and computational efficiency. As shown in Fig. 9, we denote the midpoint of the arcs a_a and a_b as P_{12} and P_{34}, respectively. Let us consider a set of four distinct pixels, P_1, P_2, P_3, and P_4 on elliptic arcs, and P_1^a, P_2^a, P_1^b and P_2^b represent the intersections of the tangent lines through these four points, respectively. Further, we construct a line l_i that passes through P_i^a and the midpoint of the line segment P_iP_{12} and a line l_j connecting P_j^b and the midpoint of the line segment P_jP_{34}; the lines l_i and l_j must pass through the ellipse centre [53], where $i = 1, 2$ and $j = 3, 4$, as seen in Fig. 9.

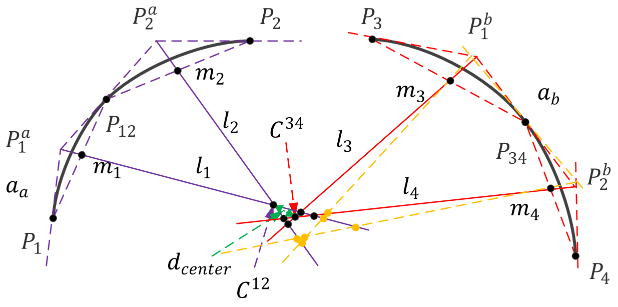

Fig. 9. Illustration of the presented calculation method of the elliptic centre. C^{12} denotes the intersection of the lines l_1 and l_2; C^{34} represents the intersection of the lines l_3 and l_4. d_{center} represents the distance between $C^{12}$and C^{34}. The shifted intersections (yellow dots) are formed by the shifted lines (yellow lines) jointing the midpoints of the chords and the intersections of the tangent lines when the slope of the line through the midpoint P_{34} of the arc a_b is not correct due to the image noise. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

If the coordinate and gradient of the endpoints and midpoint of an arc are expressed as $\{x_i, y_i, \theta_i | i = 1, 2\}$ and $\{x_{12}, y_{12}, \theta_{12}\}$ respectively, the coordinates of the midpoints (m_1, m_2) of the line segments P_1P_{12} and $P_{12}P_2$ are derived using Eq. (4). The coordinates of the intersections (P_1^a, P_2^a) of three tangent lines through the points P_1,P_2,P_{12} are expressed in Eq. (5). We calculate the slopes (q_1, q_3) of the line segments P_1P_{12} and $P_{12}P_2$ using Eq. (6) and the slopes (q_2, q_4) of the lines (l_1, l_2) passing through point pairs (m_1, P_1^a)

and (m_2, P_2^a) are obtained through Eq. (7), respectively.

$$x_{m_1} = \frac{x_1 + x_{12}}{2},\ y_{m_1} = \frac{y_1 + y_{12}}{2},$$
$$x_{m_2} = \frac{x_2 + x_{12}}{2},\ y_{m_2} = \frac{y_2 + y_{12}}{2}; \tag{4}$$

$$x_{P_1^a} = \frac{y_{12} - \theta_{12}x_{12} - y_1 + \theta_1 x_1}{\theta_1 - \theta_{12}},\ y_{P_1^a} = \frac{\theta_{12}y_1 - \theta_1 y_{12} + \theta_1\theta_{12}(x_{12} - x_1)}{\theta_1 - \theta_{12}},$$
$$x_{P_2^a} = \frac{y_{12} - \theta_{12}x_{12} - y_2 + \theta_1 x_2}{\theta_2 - \theta_{12}},\ y_{P_2^a} = \frac{\theta_{12}y_2 - \theta_2 y_{12} + \theta_2\theta_{12}(x_{12} - x_2)}{\theta_2 - \theta_{12}}; \tag{5}$$

$$q_1 = \frac{y_{12} - y_1}{x_{12} - x_1},$$
$$q_3 = \frac{y_{12} - y_2}{x_{12} - x_2}; \tag{6}$$

$$q_2 = \frac{y_{P_1^a} - y_{m_1}}{x_{P_1^a} - x_{m_1}} = \frac{(\theta_1 + \theta_{12})(y_1 - y_{12}) + 2\theta_1\theta_{12}(x_{12} - x_1)}{2(y_1 - y_{12}) - (\theta_1 + \theta_{12})(x_{12} - x_1)},$$
$$q_4 = \frac{y_{P_2^a} - y_{m_2}}{x_{P_2^a} - x_{m_2}} = \frac{(\theta_2 + \theta_{12})(y_2 - y_{12}) + 2\theta_2\theta_{12}(x_{12} - x_2)}{2(y_2 - y_{12}) - (\theta_2 + \theta_{12})(x_{12} - x_2)}. \tag{7}$$

Thus, given an arc, the coordinates of the centre C^{12} can be derived depending on the proposed method as follows,

$$x_{C^{12}} = \frac{y_{m_2} - q_4 x_{m_2} - y_{m_1} + q_2 x_{m_1}}{q_2 - q_4},$$
$$y_{C^{12}} = \frac{q_2 y_{m_2} - q_4 y_{m_1} + q_2 q_4 (x_{m_2} - x_{m_1})}{q_2 - q_4}. \tag{8}$$

Through the above analysis, we can obtain the centres C^{12}, C^{34} of a pair of a_a, a_b. If the distance of the centres C^{12}, C^{34} is smaller than a given threshold $thre_{center}$ which accounts for errors due to digitization and image noise, the third constraint of the selection strategy is satisfied.

In the subsequent stage, the ellipse parameters are estimated only for a pair of the arcs a_a, a_b that satisfy the presented three geometric constraints, namely quadrant constraint, relative position constraint, and centre proximity constraint. For such pairs, we refine the estimates of the centres as follows. If a line set consists of such lines as l_1,l_2, l_3, l_4, the ellipse centre can be identified by their line pairwise intersections belonging to the set. However, because of unavoidable distortions, the intersections are usually distributed in a small range around the exact centre. For getting a better result, we use a set of 7 points, consisting of the two centres C^{12}, C^{34}, their mean, and the other 4 intersections formed by a set lines of l_1,l_2, l_3, l_4, as represented by black points in Fig. 8, to estimate the ellipse centre. If the gradients of lines through the endpoints and midpoints of arcs contain errors in a noisy image, the geometric centres are shifted into neighbourhood of real points, as shown in Fig. 9. We need to get the rigorous geometric centre and thus, seven intersections should be coincident with each other. The smaller the errors included in the gradients of points on an arc set are, the closer seven intersections become. Then the rigorous ellipse centre can be confirmed by exploring high density spaces of the intersections with a clustering algorithm. Conventional methods detect ellipse centres by obtaining the median of the coordinates of a set of intersection points in [28,42]. Contrary to this, we generate the rigorous centre C by using the mean-shift clustering algorithm [54] upon the intersections formed by a line set. Specifically, the iterative mean-shift clustering is applied for detecting the converged centre that lies in an acceptable area decided by ROI (Region of Interest) of the mean-shift algorithm, which improves its robustness to noise.

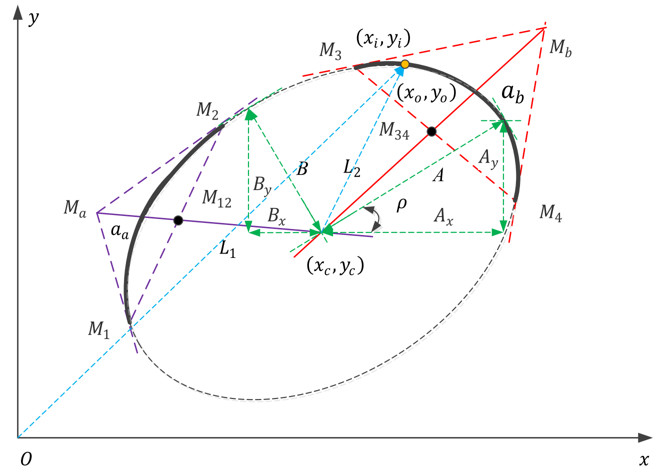

Fig. 10. Geometric schematic of parameter estimation. M_a and M_b represent the intersections of the tangent lines through the points M_1, M_2,M_3, M_4 of two arcs a_a, a_b,respectively;L_1 and L_2 denote the lines passing the midpoints M_{12}, M_{34} of the line segments M_1M_2, M_3M_4 and the intersections M_a, M_b; A and B are the long and short semi-axes lengths; ρ indicates the orientation of an ellipse; (x_i, y_i) and (x_o, y_o) are the coordinates of the points on an arc in the world coordinate system and the ellipse coordinate system, respectively.

3.3. Parameter estimation

We use the information $\{x_i, y_i, \theta_i\}$ of pixel on an arc to estimate the remaining parameters. Following the Eqs. (4–8), we obtain an expression involving the two slopes defined by the line segment jointing two points on the same arc and the line passing the intersection and the midpoint of two points of an arc, as illustrated in Fig. 10, as follows,

$$x_{M_{12}} = \frac{x_1 + x_2}{2},\ y_{M_{12}} = \frac{y_1 + y_2}{2}, \tag{9}$$

$$x_{M_a} = \frac{y_2 - \theta_2 x_2 - y_1 + \theta_1 x_1}{\theta_1 - \theta_2},\ y_{M_a} = \frac{\theta_2 y_1 - \theta_1 y_2 + \theta_1\theta_2(x_2 - x_1)}{\theta_1 - \theta_2}, \tag{10}$$

$$s_1 = \frac{y_2 - y_1}{x_2 - x_1}, \tag{11}$$

$$s_2 = \frac{y_{M_a} - y_{M_{12}}}{x_{M_a} - x_{M_{12}}} = \frac{(\theta_1 + \theta_2)(y_1 - y_2) + 2\theta_1\theta_2(x_2 - x_1)}{2(y_1 - y_2) - (\theta_1 + \theta_2)(x_2 - x_1)}, \tag{12}$$

To obtain the remaining parameters, we decompose the parameter space of an ellipse into two orthogonally projected ellipses, as described in [12,28,55,56]. Based on the derivation of [55,56], the following two parameters N, $K$are computed from the Hough Transform as follows,

$$N = \frac{B}{A} \tag{13}$$

$$K = \tan\rho, \tag{14}$$

where the orientation ρ represents the anti-clockwise rotation angle from an ellipse long axis, and$N$ is the ratio of the ellipse short semi-axes length B and long semi-axes length A. The equation that expresses N in terms of K is

$$N^2 = -\frac{(s_1 - K)(s_2 - K)}{(1 + s_1 K)(1 + s_2 K)}, \tag{15}$$

where s_1 and s_2 are the slopes of the lines L_1, L_2, respectively. When calculating each centre of a pair of arcs consisting of an ellipse, we use the point pairs of arcs to calculate s_1 and s_2 by Eq. (9)–(12). If only a pair of endpoints is applied to calculating N and K, it cannot obtain the high accuracy. However, the Eq. (15) is valid for any two points. We use two pairs of points that contributes to detecting the centre to improve the accuracy of the values N and K. Thus, the following equations are given as

$$-\frac{(s_1-K)(s_2-K)}{(1+s_1K)(1+s_2K)} = -\frac{(s_3-K)(s_4-K)}{(1+s_3K)(1+s_4K)}. \tag{16}$$

where s_1, s_2 are derived by the first pair of points on one arc and s_3, s_4 are obtained from the other pair of points on the other arc. After a series of simple algebraic manipulation, we obtain the following equation,

$$\alpha K^2 + \beta K - \alpha = 0 \tag{17}$$

with $\alpha = s_1 s_2 - s_3 s_4$ and $\beta = s_2 s_4 (s_3 - s_1) + s_1 s_3 (s_4 - s_2) + (s_1 + s_2 - s_3 - s_4)$.

Thus,

$$K = \pm\sqrt{1 - \frac{\beta}{\alpha}}. \tag{18}$$

We continuously update the N-K accumulator that consists of weighted voting values for each point pairs on arcs depending on Eq. (15)–(16). For two pairs of points, the values of N and ρ are the highest peaks, which are the local maxima, in two one-dimensional accumulators. After the ellipse centre (x_c, y_c) and the two parameters (N, K) are estimated by the clustering algorithm and two one-dimensional accumulators respectively, we calculate the two semi-axes A and B. Here, the polar equation of ellipse and coordinate transformation are applied. The coordinate (x_i, y_i) of a point on an arc is transformed to the coordinate (x_o, y_o) in the ellipse coordinate system (the centre of the ellipse as the origin and the semi-major axis as the horizontal axis) as,

$$\begin{bmatrix} x_o \\ y_o \end{bmatrix} = \begin{bmatrix} \cos\rho & \sin\rho \\ -\sin\rho & \cos\rho \end{bmatrix} \begin{bmatrix} x_i - x_c \\ y_i - y_c \end{bmatrix}. \tag{19}$$

Based on Eq. (13)–(16), we obtain the following equations,

$$x_o = \frac{(x_i - x_c) + (y_i - y_c)K}{\sqrt{K^2+1}},$$
$$y_o = \frac{(y_i - y_c) - (x_i - x_c)K}{\sqrt{K^2+1}}. \tag{20}$$

Referring to [55], we estimate A_x as follows.

$$A_x = \sqrt{\frac{x_o^2 N^2 + y_o^2}{N^2\left(1+K^2\right)}}. \tag{21}$$

Finally, combining $A_x = A\cos\rho$ and $N = \frac{B}{A}$, we identify an ellipse candidate with a parameter set (x_c, y_c, N, K, A).

4. Validation and clustering

4.1. The validation of the candidate ellipse

An arc set that satisfies the three constraints above still may consist of an invalid ellipse. Therefore, a scheme that filters false detections by identification of invalid ellipses is incorporated in the proposed method.

4.1.1. Confirmation by the ratio of the length

We adopt the ratio of half of the circumference of the bounding box enclosing an arc and the sum of the semi-axes lengths to be a measurement of an ellipse circumference. Assuming that an ellipse candidate is fit by an arc pair $\tau = (a_1, a_2)$ enclosed by bounding boxes with the size of $m_i \times n_i$ $(i = 1, 2)$, we define a function $\mathrm{K}(\tau)$ as follows

$$\mathrm{K}(\tau) = \frac{\sum_{i=1}^{2}(m_i + n_i)}{2(A+B)}. \tag{22}$$

A higher value of $\mathrm{K}(\tau)$ implies that the arc pair has a larger integrity for an ellipse candidate. If $\mathrm{K}(\tau)$ exceeds a threshold $(thre_l)$, the ellipse candidate can be verified further by the below constraint, otherwise it is regarded as a false detection and is gave up. The corresponding score $s(\mathrm{K}(\tau))$, which is equvalent to $\mathrm{K}(\tau)$, is given to the candidate ellipse.

4.1.2. Confirmation via the ratio of the circumference

Supposing that the coordinate of a pixel on an arc is (x_i, y_i), we put this coordinate into the follows equations,

$$X = \frac{[(x_i - x_c)\cos\rho + (y_i - y_c)\sin\rho]^2}{A^2},$$
$$Y = \frac{[(y_i - y_c)\cos\rho - (x_i - x_c)\sin\rho]^2}{B^2} \tag{23}$$

where x_c and y_c represent the coordinate of the ellipse centre and ρ donates the orientation angle of a detected ellipse. The differential distance d between a chosen elliptic arc and an arc of a real ellipse arc is provided as follows,

$$d = |X + Y - 1|. \tag{24}$$

If d is less than a threshold $(thre_d)$, it implies that the point (x_i, y_i) on an arc pair $\tau = (a_1, a_2)$ is close enough to the edge of the ellipse candidate. In this case, the point (x_i, y_i) can be put a set of points $\mathcal{B}$. We define a function $\psi(\tau)$ to represent the ratio of the number of elements in $\mathcal{B}$ to the total number of points on the two arcs in the arc set τ used for detecting the ellipse,

$$\psi(\tau) = \frac{N_{\mathcal{B}}}{N_a + N_b} \tag{25}$$

where N_a and N_b are the numbers of points on the two arcs, respectively and $N_{\mathcal{B}}$ is the number of points in $\mathcal{B}$. The larger the ratio $\psi(\tau)$ is, the better the ellipse-fitting is. If $\psi(\tau)$ is more than a threshold $(thre_\psi)$, the detected ellipse is considered as a valid one, otherwise this ellipse is discarded. When satisfying the constraint, the ellipse candidate is given a score $s(\psi(\tau))$ whose value is the ratio $\psi(\tau)$.

4.2. Ellipse clustering

Through the verifications above, multiple arc sets may be fit to the same 'real ellipse' as they may belong to the same 'real ellipse'. We adopt the clustering approach [57,58] to eliminate duplicated ellipses by the order of the total scores $s(\mathrm{K}(\tau)) + s(\psi(\tau))$ [21]. We first allow the ellipse with the highest score to be the centre of a given cluster and then, we discard other ellipses that belong to this cluster. If an ellipse does not belong to any cluster, it becomes the centre of a new cluster.

5. Geometric property for false detection control

Since the intrinsic attribute of the ellipse is expressed by mathematical model, a geometric feedback loop is developed by us to sieve out false detections significantly. Here we use the following geometric constraint to describe an elliptical geometric property: if a point Q_M is the midpoint of elliptic chord Q_1Q_2, the product of the slope $k_{Q_1Q_2}$ of chord Q_1Q_2 and the slope k_{OQ_M} of line segment formed by the origin O of ellipse and Q_M is equal to the constant $-\frac{B^2}{A^2}$, as illustrated in Fig. 11. This property remains valid

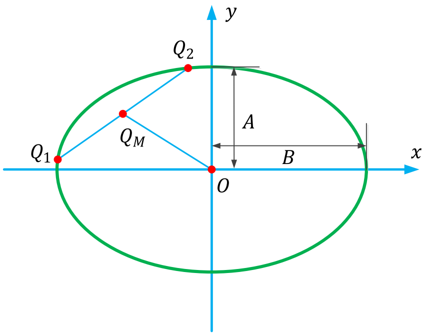

Fig. 11. Schematic of geometric property of ellipse. A and B represent the major and minor semi-axes lengths; Q_1 and Q_2 denote two points on an ellipse. O is the origin of ellipse.

for any form of linear coordinate transformation. For simplification, we consider an ellipse with centre (0,0), and orientation angle $\rho = 0$. As $Q_1(x_1, y_1)$ and $Q_2(x_2, y_2)$ satisfy the ellipse equation $\frac{x^2}{A^2} + \frac{y^2}{B^2} = 1$, we can obtain as,

$$\frac{{x_1}^2}{A^2} + \frac{{y_1}^2}{B^2} = 1, \ \frac{{x_2}^2}{A^2} + \frac{{y_2}^2}{B^2} = 1. \tag{26}$$

Then the difference between the two equations is provided as,

$$\frac{(x_1 - x_2)(x_1 + x_2)}{A^2} + \frac{(y_1 - y_2)(y_1 + y_2)}{B^2} = 0, \tag{27}$$

$$\begin{aligned} \text{Thus}, \ k_{Q_1Q_2} &= \frac{y_1 - y_2}{x_1 - x_2} = -\frac{B^2(x_1 + x_2)}{A^2(y_1 + y_2)} \\ &= -\frac{B^2}{A^2 k_{OQ_M}} \text{ with } Q_M\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2}\right). \end{aligned} \tag{28}$$

$$\text{Finally}, \ k_{Q_1Q_2} \cdot k_{OQ_M} = -\frac{B^2}{A^2}. \tag{29}$$

Since the discrete character of digital image influences the precision of calculation, it is preferable to choose a suitable threshold for the following inequality,

$$\left| k_{Q_1Q_2} \cdot k_{OQ_M} + \frac{B^2}{A^2} \right| < Thre_p.$$

A chosen pair of points are from two different arcs consisting of an ellipse. If the x or y values of the chosen points are the same, a slope k_{OQ_M} or $k_{Q_1Q_2}$ is absent, respectively. We capture three pairs of such points that rest on randomly for verifying a detection result depending on the constraint above. If these points satisfy the geometric constraint, we accept the detection, otherwise discard it. In fact, this procedure spends a small computational cost, there are just a small number of ellipse candidates in each image after validation and clustering of ellipse.

6. Experimental results and discussions

We present various experimental results to verify the performance and advantages of the proposed method. We use four publicly available datasets, namely, Dataset of synthetic images [20,21] supplemented by us, Dataset Prasad [21], dataset #1 [28], Dataset HX created by us. We perform a series of experiments to evaluate the performances of ellipse detection approaches. These datasets have different characteristics and are discussed in subsection 6.1. We compare the performance of our method against six state-of-the-art ellipse detection methods, namely Jia et al. [42], Fornaciari et al. [28], Prasad et al. [21], Bai et al. [29], Liu et al. [33], and Mai et al. [22]. Source codes of these methods in C++ or MATLAB are available online. The execution time of the programs in MATLAB is scaled down by 50 to make it comparable to a more computationally efficient implementation in C++ [28]. The performance metrics are presented in subsection 6.2. All the experiments are executed without code parallelization on a PC with 8GB RAM and an Intel Core i7 processor. The threshold parameters that determine the performance of the proposed ellipse detection are explored in subsection 6.3. The performance comparison with six state-of-the-art methods and the corresponding discussions are presented in subsection 6.4.

6.1. Four public datasets

6.1.1. Dataset of synthetic images

We first evaluate the performance of the proposed method using the dataset of synthetic images. Originally, there are 600 images with occluded ellipses and 600 images with overlapping ellipses of the size of 300 × 300 in synthetic dataset [20]. The $\tau \in \{4, 8, 12, 16, 20, 24\}$ occluded or overlapping ellipses are distributed randomly within the region of image. For each value of γ, we have 100 images with occluded ellipses and 100 other images with overlapping ellipses. We create synthetic images with arcs that just lie in two quadrants to make up dataset of synthetic images [20]. Specifically, the size of such images is also 300 × 300. All the images with arcs resting on two quadrants can be divided into two categories such as images including arcs that lie in the first and second quadrants and images with arcs resting on the first and third quadrants. The $\tau \in \{2, 4, 6\}$ arcs consisting of ellipses scatter randomly within the region of image. 50 images with arcs in first and second quadrants and 50 other images with arcs in first and third quadrants are created for every τ.

6.1.2. Dataset Prasad

The ellipses are detected in the images of Dataset Prasad [21] to demonstrate the performance of the presented method in real scenarios. The dataset is composed of 198 images that are randomly chosen from 48 categories in Caltech256 dataset [59] and still available online. In Dataset Prasad, the number of small ellipses (i.e. with semi-major axis length shorter than 20 pixels) is high compared to Dataset #1. Moreover, in Dataset Prasad the number of ellipses in each image is more uniform across the images than in Dataset #1.

6.1.3. Dataset #1

Dataset#1 consists of 400 real images with elliptic shapes, which are collected from MIRFlickr and LabelMe repositories [28]. Dataset #1 contain fewer ellipses than Dataset Prasad. The images in Dataset #1 are classified into two main categories, such as the high-definition image with a single object and lower-resolution images containing many objects with more complex scenes than Dataset Prasad.

6.1.4. Dataset HX

We construct a new dataset named Dataset HX, which allows evaluation of the proposed method for application to real industrial scenarios. For instance, a robot can manipulate an object by a real-time ellipse detection. We randomly collect 300 images with elliptic shapes from 16 categories in ImageNet repository [60] to form the Dataset HX. We generate the ground truth for each image using the annotation tool presented in [21]. All the images are from real scenarios like households, industries, and roads. For each scenario, we choose 100 images for building Dataset HX. The images in industrial scenarios are taken in dim light conditions and are degraded due to partial occlusions and illumination variations. In household and road environments, the quality of the images is poor because of spectral reflections. In industrial images, the presence of reflective and dense pits on surfaces of mechanical parts corrupt the elliptic contours seriously, which makes the elliptic edge contours appear as non-elliptic and splits the continuous elliptic curve into short edge contours. Thus, there exist more challenges for detecting ellipses in images of Dataset HX than Dataset Prasad and Dataset #1.

6.2. Evaluation metrics

We evaluate the performance of the presented method in terms of the detection effectiveness and execution time. The detection effectiveness of the algorithm is assessed according to the approach proposed in [61]. Specifically, we determine the correctness of a detected ellipse E_d based on the following overlap ratio between the detected and the ground-truth ellipses,

$$r_{overlap} = \frac{area(E_d) \cap area(E_g)}{area(E_d) \cup area(E_g)} \tag{30}$$

where $area(E_i)$ denotes the cardinality of the set of pixels within ellipse E_i; E_d and E_g are a valid detection and a ground-truth ellipse, respectively. An ellipse is considered detected if the overlap ratio is more than 0.95 for synthetic images and 0.8 [28,30] for real images in Dataset Prasad, Dataset #1 and Dataset HX. A common index *F-measure* can evaluate the detection effectiveness. *F*-measure combines the precision and the recall of the detector, which are defined as follows,

$$Precision = \frac{Number\ of\ correctly\ detected\ ellipses}{Total\ number\ of\ ellipses\ detected} \tag{31}$$

$$Recall = \frac{Number\ of\ correctly\ detected\ ellipses}{Total\ number\ of\ ellipses\ present\ in\ test\ images} \tag{32}$$

$$F - measure = \frac{2\ \times Precision \times Recall}{Precision + Recall}. \tag{33}$$

Besides the effectiveness, we also use an average execution time t (ms/image) to assess the execution speed of the presented method in the four datasets.

6.3. Parameter tuning and sensitivity analysis

The performance of ellipse detection method depends on the choice of threshold parameters, including $thre_r$, $thre_a$, $thre_\theta$, $thre_{center}$, $thre_l$, $thre_d$, and $thre_\psi$, $thre_p$. However, we cannot ensure suitability of a chosen set of values for the threshold parameters in all images for achieving the best performance of ellipse detection. On the other hand, it is quite difficult to build an accurate mathematical model of the multi-objective optimization problem of identifying suitable values of control parameters for each image. Thus, we use the statistical data of *F-measure* and the excitation time (t) obtained by tuning one threshold parameter at a time while fixing other threshold parameters.

During the pre-processing, the straight curves and short curves are removed through the threshold parameters $thre_r$ and $thre_a$, respectively. The influences of the threshold $thre_r$ and $thre_a$ on the effectiveness and execution time are explored in Fig. 12(A). Regarding $thre_r$, it is quite unambiguous that the detection effectiveness does not show significant changes for $thre_r \geq 10$, while the execution time increases monotonically. Thus, we choose 10 as the value of $thre_r$. Regarding $thre_a$, it is observed that although the execution time decreases monotonically with increasing values of $thre_a$, the F-measure decreases for $thre_a > 8$. Thus, we adopt $thre_a = 8$. The threshold $thre_\theta$ is used for arc extraction and its impact on the ellipse detection performance is studied in Fig. 12(B). The execution time monotonically decreases with increasing values of $thre_\theta$. On the other hand, the best value of F-measure is obtained in the range $30 \leq thre_\theta \leq 40$ across all the datasets of real images. As a result, we set $thre_\theta = 35^o$. The threshold $thre_{center}$ is used for the geometric constraint pertaining selections of arc pairs based on the proximity of the estimated centres of ellipses. Impact of its value on the performance of the proposed methods is investigated in Fig. 12(C). The execution time increases monotonically with increase in the value of $thre_{center}$. However, the F-measure reaches the best values for $25 \leq thre_{center} \leq 30$. In Fig. 12(D), we choose $thre_{center} = 25$ as a good trade-off to gain a good effectiveness and keep an execution time at the minimum.

The threshold $thre_l$ is applied to the ratio of the semi-axis lengths to the side lengths of bounding box for reducing false positive detections. As illustrated in Fig. 13(A), the execution time decreases monotonically with increasing values of $thre_l$. However, the F-measure decreases very fast for values of $thre_l$ larger than 0.6. This implies a more strict selection of valid ellipses, which results in poorer recall. The peak F-measure is achieved in the range $0.4 \leq thre_l \leq 0.5$. We set $thre_l = 0.5$ in favour of a balance between F-measure and execution time. The threshold $thre_d$ on the distance between a pixel on a detection arc and corresponding pixel on a real arc of ellipse is another parameter used of suppressing false positive detections. Fig. 13(B) illustrates that F-measure, peaking in the range $0.6 \leq thre_d \leq 1.0$, but monotonically increasing execution times. Thus, we choose $thre_d = 0.8$. Threshold $thre_\psi$ pertains the reduction of false positives through the ratio of the circumference. Results for $thre_\psi$ are shown in Fig. 13(C). The execution time decreases monotonically with $thre_\psi$ while the F-measure dramatically decreases when the threshold $thre_\psi$ is greater than 0.3 because of the reduction in recall. Thus, we choose $thre_\psi = 0.3$. As the fluctuation of execution times is relatively stable, we choose $thre_p = 10$ to realize the best detection performance, as illustrated in Fig. 13(D).

6.4. Performance comparisons

6.4.1. Experiments on synthetic images

We first test the performance of the proposed ellipse detection method in synthetic images in which each image contains γ occluded or overlapping ellipses, $4 \leq \gamma \leq 24$. γ represents the number of occluded or overlapping in an image. The synthetic images with arcs resting on two quadrants that consist of $\tau \in \{2, 4, 6\}$ ellipses are also explored to evaluate the performance of the presented detector.

As illustrated in Fig. 14(A), the F-measure of the proposed approach exceeds five reference methods and is comparable to Prasad's method. For images with between four to twelve occluded ellipses, the F-measure is more than 0.90. As γ increases to twenty-four, *F-measure* still maintains a relatively high value for the presented approach. The reference methods of Jia, Fornaciari and Mai shows a deteriorating F-measure with increasing number of occluded in an image. Specifically, at $\gamma = 24$, the presented method can obtain F-measure of approximately 0.8, a significantly better value than the methods of Jia (0.45), Fornaciari (0.50), and Mai (0.05). Even though the F-measure of Bai's method does not deteriorate significantly with increasing γ, it performs poorer as a whole compared with the proposed method, with the exception of $\gamma = 24$. Liu's method is ineffective for detecting occluded ellipses because there is no scheme for dealing with the inflexion points in Liu's method. We show examples of the synthetic test images in Fig. 15.

We further assess the robustness of the proposed method by detecting overlapping ellipses. It is evident that the proposed method significantly outperforms the reference methods except for Prasad's method in terms of F-measure, as shown in Fig. 14(B). As for images containing less than 20 overlapping ellipses, our approach achieves F-measure of more than 0.9. The detection effec-

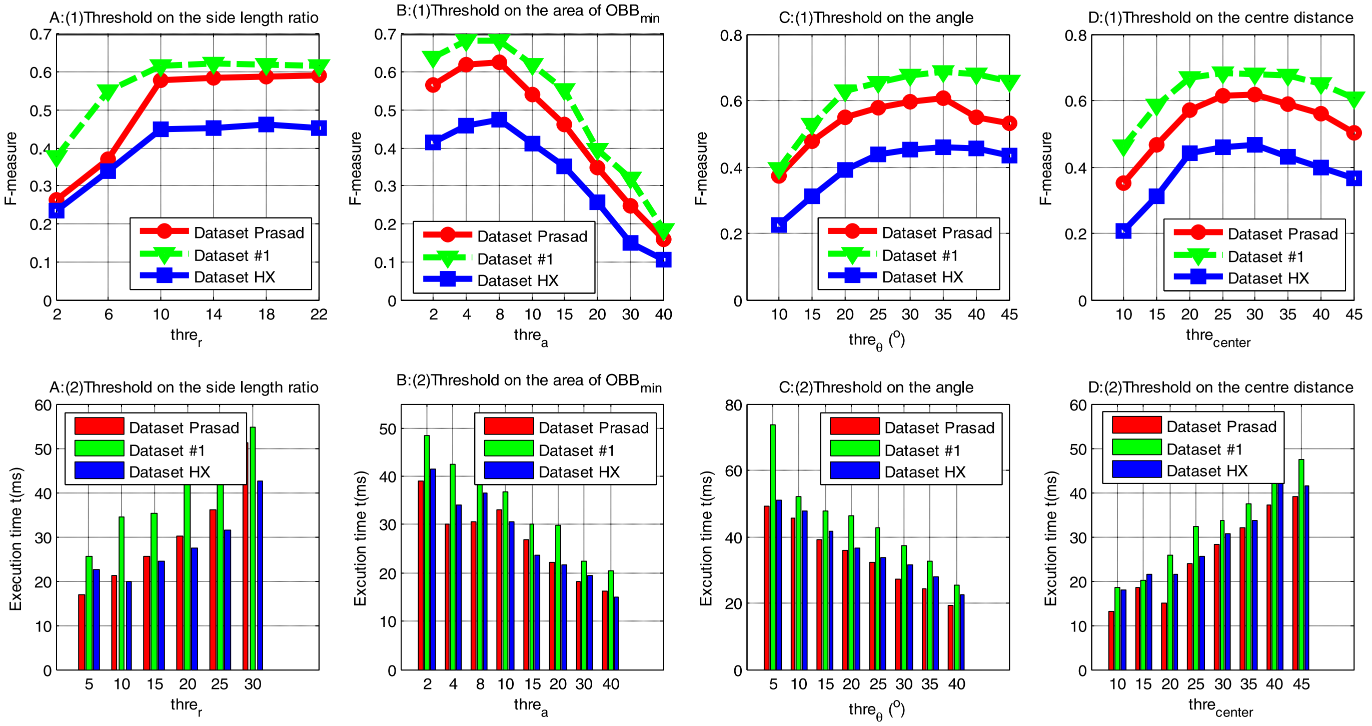

Fig. 12. *F-measure* (1) and execution time *t* values (2) for $thre_r$(A), $thre_a$(B), $thre_\theta$(C), $thre_{center}$(D).

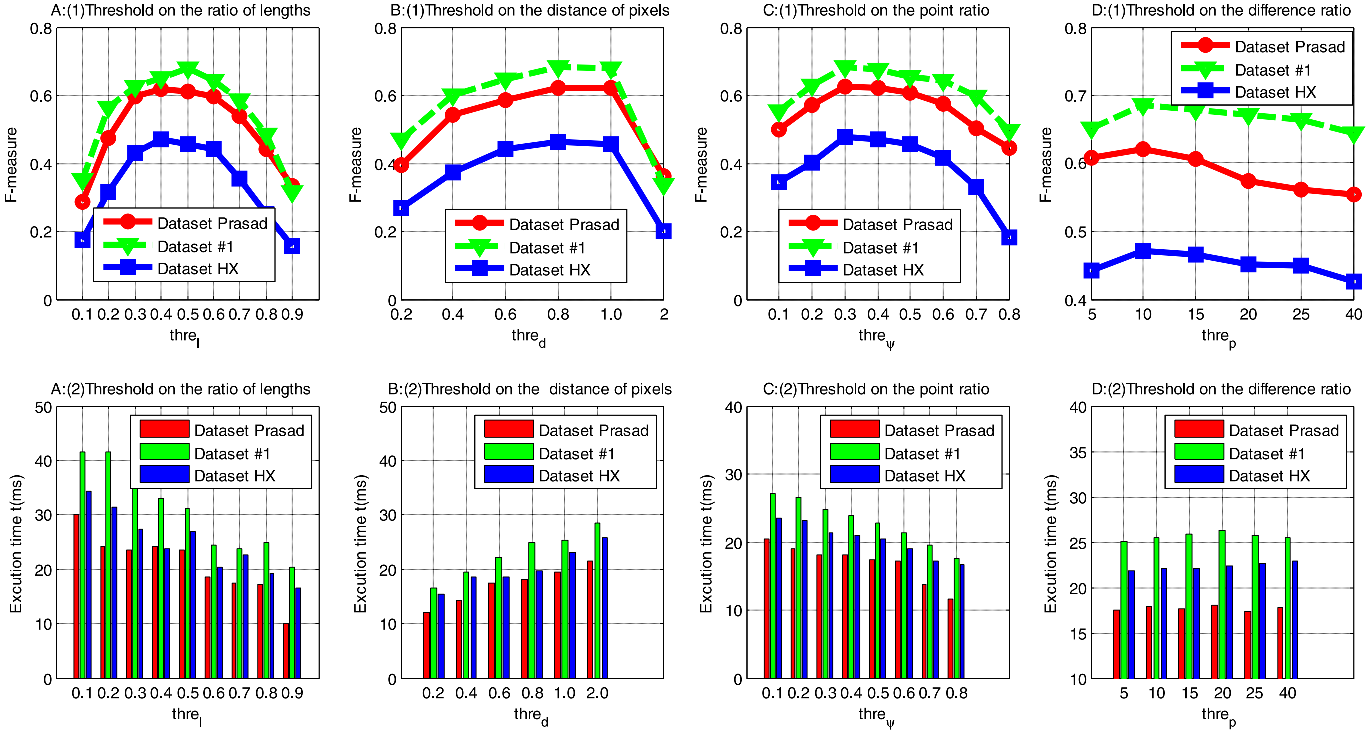

Fig. 13. *F-measure* (1) and execution time *t* (2) for $thre_l$(A), $thre_d$(B), $thre_\psi$(C), $thre_p$(D).

tiveness of the proposed method and Prasad's method are relatively stable, deteriorating only slightly with increasing γ. The performances of Jia's, Fornaciari's and Mai's methods show deterioration in detecting overlapping ellipses with increasing γ. The significant difference in performance is apparent for = 24, for example, we obtain F-measure of approximately 0.87, whereas the F-measures of Jia, Fornaciari, and Mai are approximately 0.56, 0.53, and 0.39, respectively. We note that the performances of Bai's and Liu's methods improve slightly as the number of overlapping ellipse increases. The specific cases of detecting overlapping ellipses are shown in Fig. 16.

Detection of multiple fragmentary ellipses that are made up of arcs in two quadrants is a very challenging task. We simulate this situation by considering ellipses with arcs spanning only two

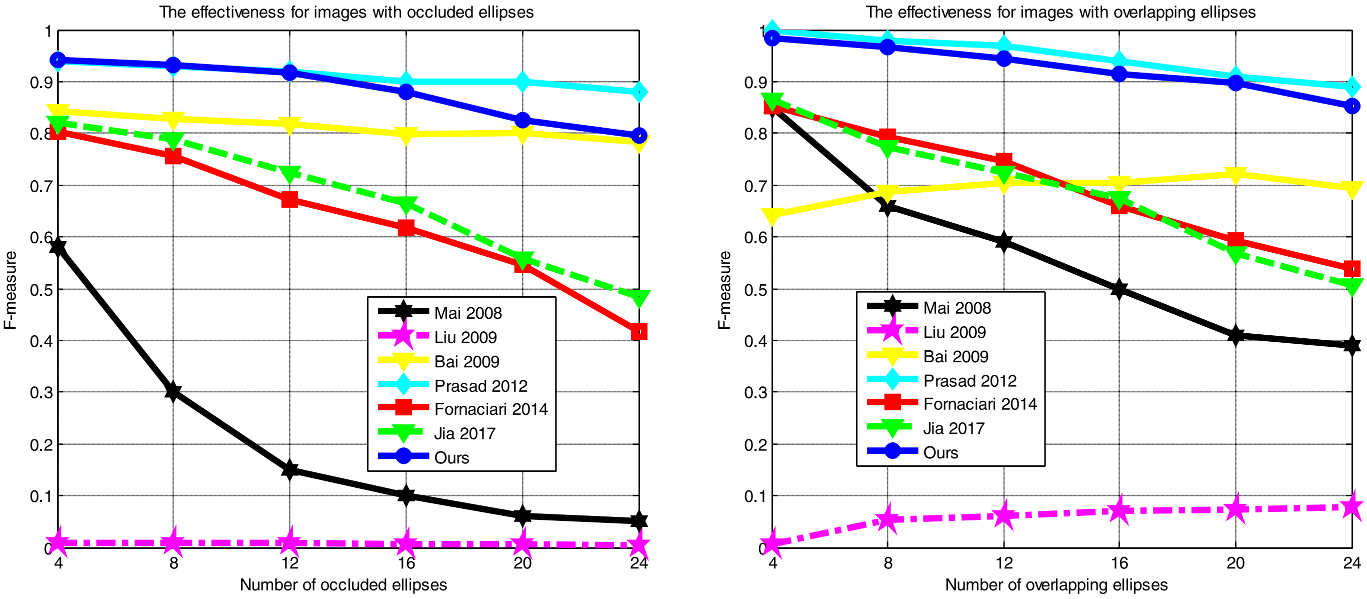

Fig. 14. *F-measure* for images with γ occluded ellipses (A) or overlapping ellipses (B), $4 \le \gamma \le 24$.

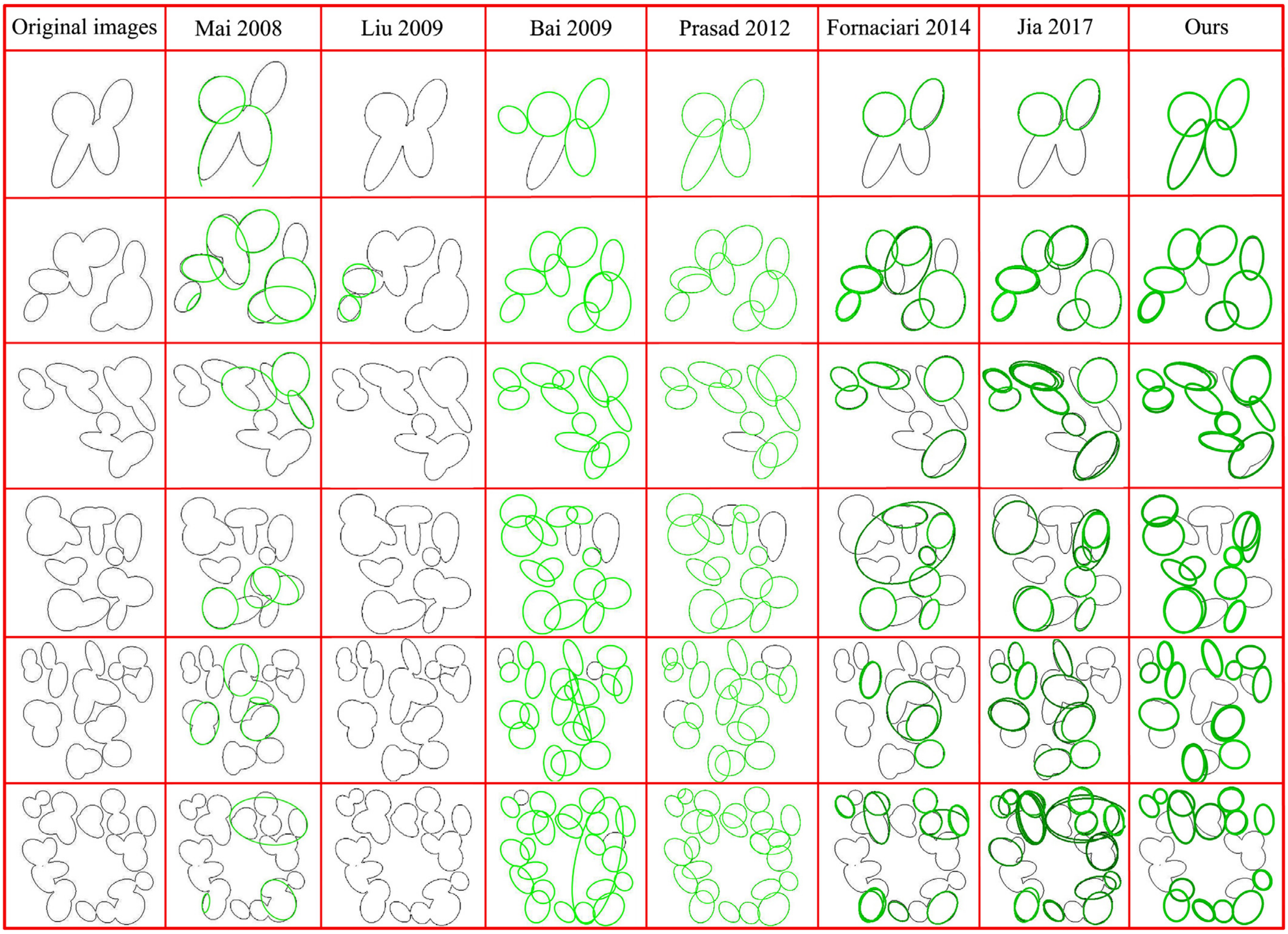

Fig. 15. The practical cases of detecting γ occluded ellipses in synthetic images, $4 \le \gamma \le 24$.

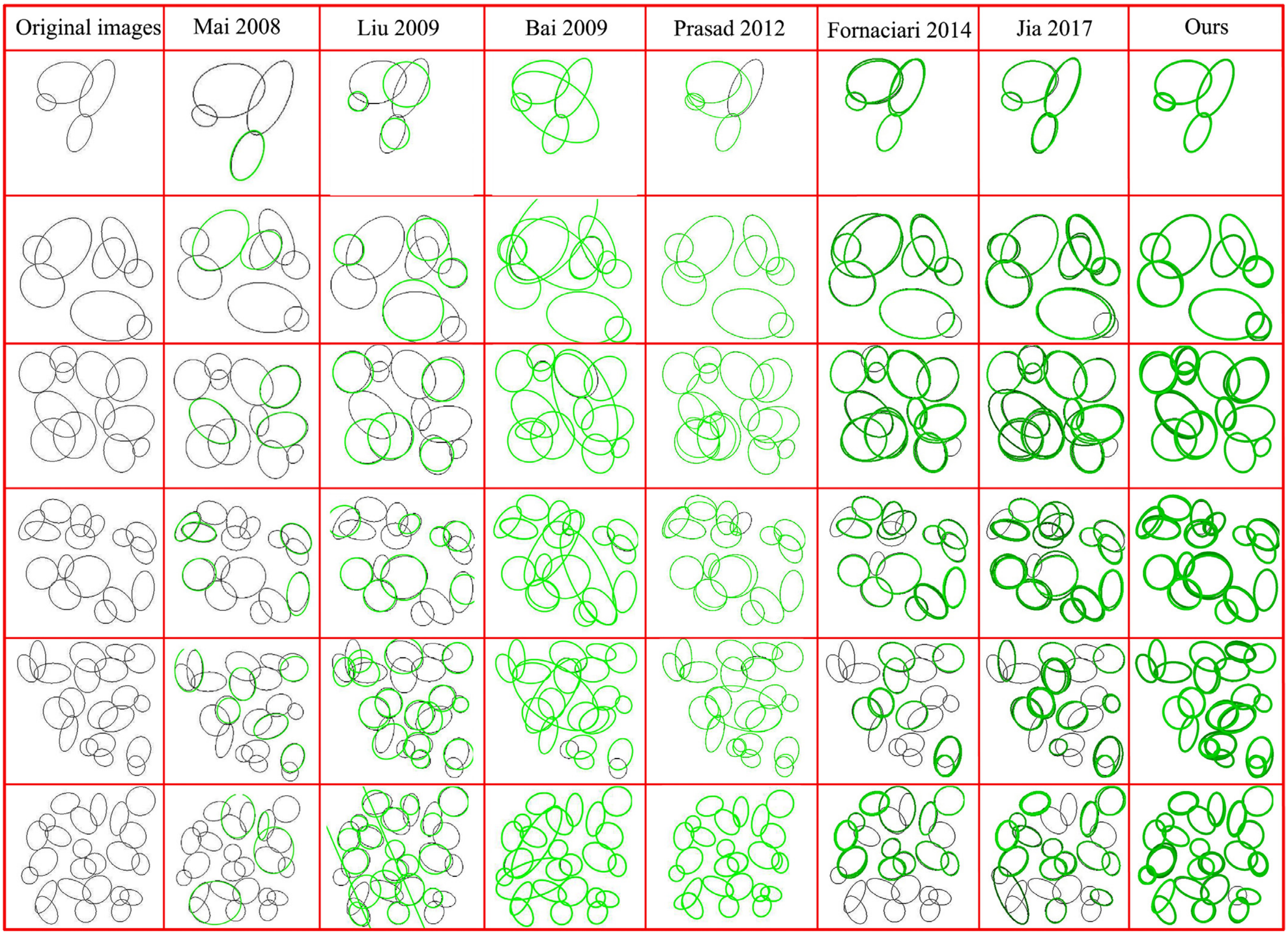

Fig. 16. The practical cases of detecting γ overlapping ellipses in synthetic images, $4 \le \gamma \le 24$.

Table 2
The detection results for Dataset of synthetic images (arcs resting on two quadrants).

	Dataset of synthetic images (arcs resting on two quadrants)						
	Mai 2008	Liu 2009	Bai 2009	Prasad 2012	Fornaciari 2014	Jia 2017	Ours
Precision	0.0000	0.0400	0.4503	0.5237	0.0000	0.0000	0.7820
Recall	0.0000	0.0119	0.6056	0.8228	0.0000	0.0000	0.9064
F-measure	0.0000	0.0184	0.5165	0.6400	0.0000	0.0000	0.8396

quadrants. In order to make the problem even more challenging, we consider presence of multiple such ellipses (2 ellipses, 4 elipses, or 6 ellipses), see Fig. 17. As illustrated in Table 2, the proposed method shows the best performances in the detection effectiveness against other six methods on such images. Obviously, Fornaciari's and Jia's detectors depend on three arcs that must lie in three different quadrants respectively. So, they cannot detect such ellipses. Compared to the results of Bai's and Prasad's methods, the proposed method significantly improves algorithm efficiency by the means of geometric properties. Mai's and Liu's detectors fail to work when the elliptical arc is broken and fragmented. The detection examples of the presented detector are shown in Fig. 17. As the execution time taken is quite small for Dataset of synthetic images [21], we do not present the comparison of execution times for this dataset.

6.4.2. Experiments on real-world images

For comparing the performances in Dataset Prasad, the comparison results in terms of *F-measure* and average execution times are reported in Table 3. For Dataset Prasad, the method proposed by Prasad et al. [21] performs the best in terms of F-measure but reports quite high execution time. The presented method is in the second place, very close to Prasad's method in terms of *F-measure*, however requiring significantly smaller execution time. Jia and Fornaciari's methods reveal poorer performances since their algorithms are not suitable for detecting small ellipses and occluded ellipses in two quadrants, which are numerous in Dataset Prasad. They have smaller execution times as compared to our method, however at the cost of reduction in F-measure. Liu's method performs the worst (*F-measure* < 0.1) because the selection strategy based on random points is ineffective in the presence of noise in real images. The methods of Bai et al. and Mai et al. result into similar F-measures with values 0.2395 and 0.1831, respectively. Liu's, Bai's, and Prasad's method perform poorer than ours in terms of execution times, and that our method can support detection rates of up to 60 Hz due to mean execution time of ~16 ms. Based on the trade-off consideration, our method is superior to

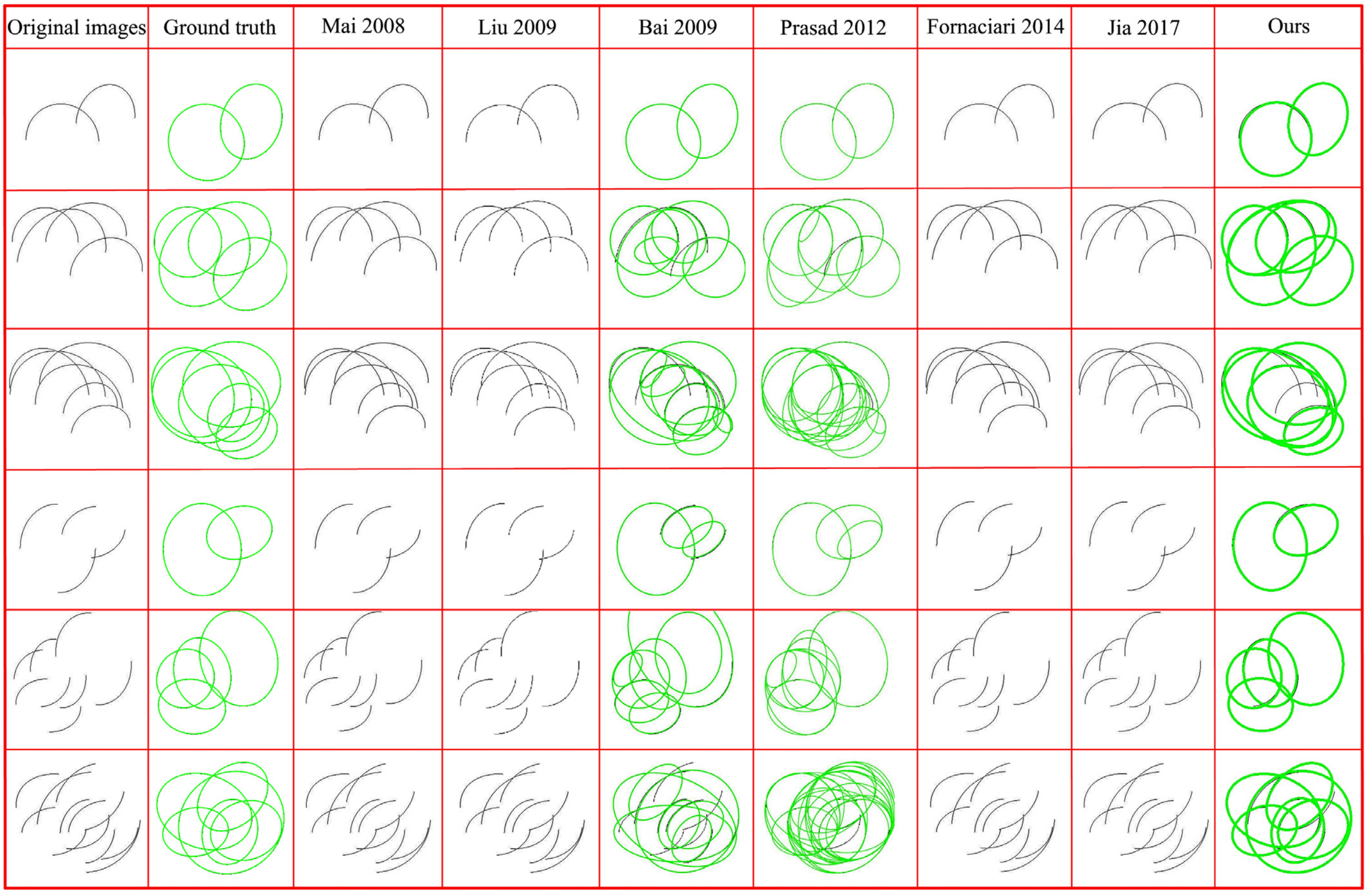

Fig. 17. The example cases of detecting $\tau \in \{2, 4, 6\}$ partial two quadrant ellipses in synthetic images.

Table 3
The detection results for Dataset Prasad, Dataset #1 and Dataset HX.

	Dataset Prasad				Dataset #1				Dataset HX			
	F-measure	Time(ms)	Precision	Recall	F-measure	Time(ms)	Precision	Recall	F-measure	Time(ms)	Precision	Recall
Mai 2008	0.2535	962.2	0.4135	0.2269	0.2604	1979.5	0.3299	0.2463	0.2573	597.8	0.3109	0.2431
Liu 2009	0.0950 0.0950	1049	0.07	0.1505	0.1170	3960	0.1248	0.1415	0.0534	1560	0.0653	0.0575
Bai 2009	0.2395	3470	0.2195	0.2890	0.2121	136,085	0.2420	0.1909	0.1028	4140	0.0830	0.2181
Prasad 2012	**0.7428**	8300	0.8512	0.6813	0.4512	17,130	0.4425	0.4610	0.3910	3260	0.3364	0.4669
Fornaciari 2014	0.3661	12.61	0.8541	0.2330	0.5716	16.55	0.7117	0.4777	0.3472	19.35	0.8825	0.2161
Jia 2017	0.4332	**8.42**	0.7390	0.3064	0.5733	**12.61**	0.6161	0.5361	0.4032	**15.69**	0.8165	0.2677
Ours	**0.6278**	**16.72**	0.8763	0.4891	**0.6861**	**24.82**	0.8635	0.5691	**0.4756**	**22.14**	0.8965	0.3237

Table 4
The average execution time of datasets of real images and average F-measure for datasets of real images and synthetic images.

	Mai 2008	Liu 2009	Bai 2009	Prasad 2012	Fornaciari 2014	Jia 2017	Ours
Average time (ms) for datasets of real images	1179.8	2189.7	47,898.3	9563.3	16.17	**12.24**	21.23
Average F-measure for datasets of real images	0.2535	0.0885	0.1848	0.5283	0.4282	0.4699	**0.5964**
Average F-measure for dataset of synthetic images	0.2578	0.0274	0.6735	0.8333	0.4441	0.4579	**0.8832**

other methods in the detection performance. The detection results are illustrated for several examples in Fig. 18.

Now, we compare the performances for Dataset #1. The average results of the performance metrics, including the average F-measure and the average execution time, are listed in Table 4. Our method not only outperforms the other methods in terms of the F-measure, but also illustrates an acceptable level on the execution time for practical applications. At average execution time of ~24 ms, it can support detection rates of approximately 40 Hz. Prasad's method is very demanding in terms of execution times and but does not reach the F-measure to the same levels as in synthetic scenarios or their own dataset. Fornaciari's and Jia's methods demonstrate excellent execution speeds but fail to detect several small ellipses and do not reach the F-measure of our method. Liu's, Bai's and Mai's methods provide low F-measures even though they take small execution time. Examples of ellipses detected by all the methods for images from Dataset#1 are shown in Fig. 19.

Table 4 also shows the performance results for the more challenging and practical Dataset HX. Examples of detected ellipses are illustrated in Fig. 20. As observed in Fig. 20, our method can be ability to detect most of ellipses in images while most reference methods may be ineffective for detecting such ellipses. The pre-

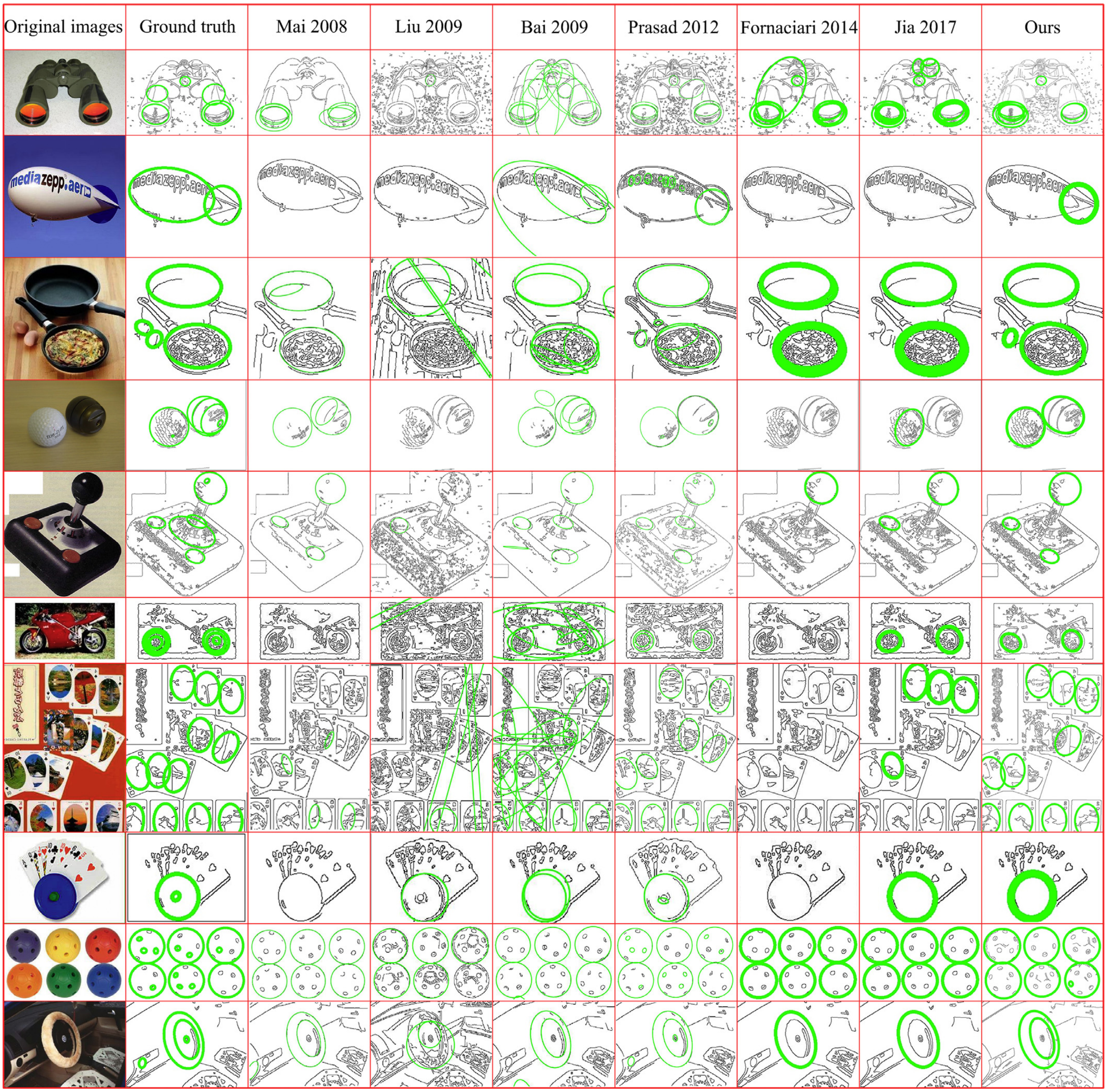

Fig. 18. The detection results of the real images in Dataset Prasad.

sented method not only outweighs the other methods in terms of F-measure, but also has practically small execution time. At average execution time of 22.14 ms, it can support detection rate of approximately 45 Hz. In contrast, Prasad's method obtains F-measure of 0.391, but suffers for images with industrial backgrounds in terms of the execution time. Though the time consumed by the methods of Jia and Fornaciari is little, they are clearly inferior to the proposed method in terms of detection effectiveness. Liu's, Bai's, and Mai's methods provide impractically small F-measures, especially missing overlapping or occluded ellipses in images with complex backgrounds.

6.5. Discussions

Statistical results regarding the average F-measure and average execution times for datasets are shown in Table 4, respectively. As seen in Table 4, the presented method is in the first place in terms of the average F-measure for all four datasets. Further, the low average values of execution time indicate that our detector can be applied to practical scenarios in real time. We mention that the poorest detection rate for our method is approximately 40 Hz, which easily supports video rate ellipse detection. Thus, it can be concluded that our algorithm realizes an excellent balance between the detection effectiveness and the execution time. In this section, we focus on the methods of Prasad, Fornaciari and Jia as they have relatively better performances than Bai's, Liu's and Mai's

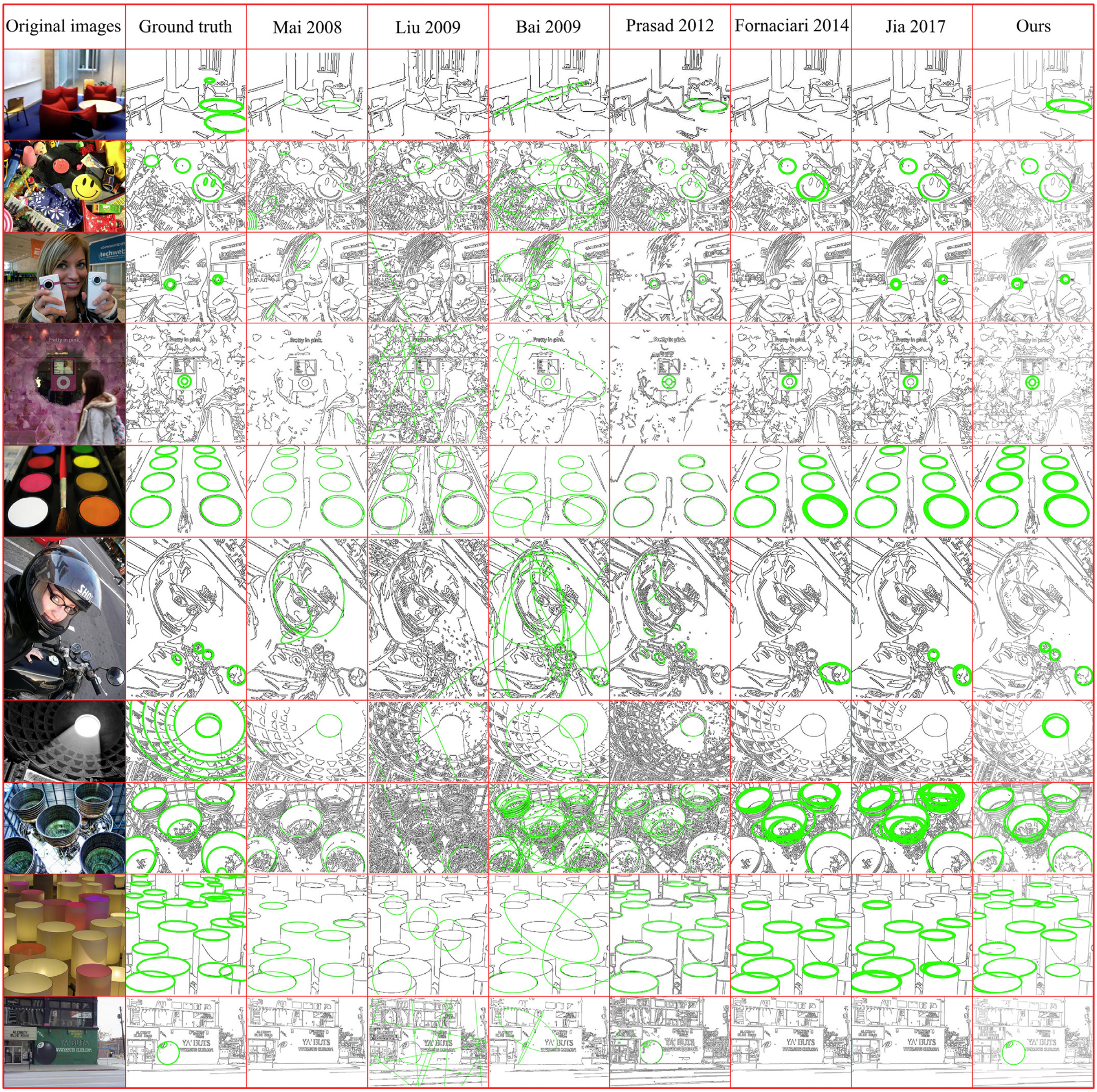

Fig. 19. The detection results of the real images in Dataset #1.

methods. Prasad's detector has an apparent disadvantage in the execution time since its clustering procedure is very time consuming. For arc grouping, they explore matching all arcs within the search region of an arc to form a set of arcs that potentially belong to the same ellipse. In this sense, it takes an exhaustive approach, which results in high computational costs and limited applicability for real time or video rate ellipse detections. The approach of Fornaciari's and Jia's methods to classify arcs into quadrants by the arc convexity-concavity based on sizes of areas is not suitable for small arcs. This results for poor performance of these methods in detecting overlapping or occluded ellipses. Moreover, though the selection strategies of Fornaciari's and Jia's methods, similar to ours, significantly speed up the grouping procedure, they impose the condition that a set of arcs selected for fitting an ellipse must rest on at least three different quadrants. Consequently, highly occluded ellipses, such as semi-ellipses, cannot be detected by these methods. Further, Fornaciari's and Jia's methods use parallel chords to estimate ellipse's centres. However, as mentioned in Subsection 3.2, this approach of centre estimation is inaccurate for short arcs, subsequently decreasing the detection effectiveness of overlapping or occluded ellipses. In general, our method aims for overcomes the disadvantages of Prasad's and Fornaciari's methods by employing superior strategies of arc extraction and arc grouping.

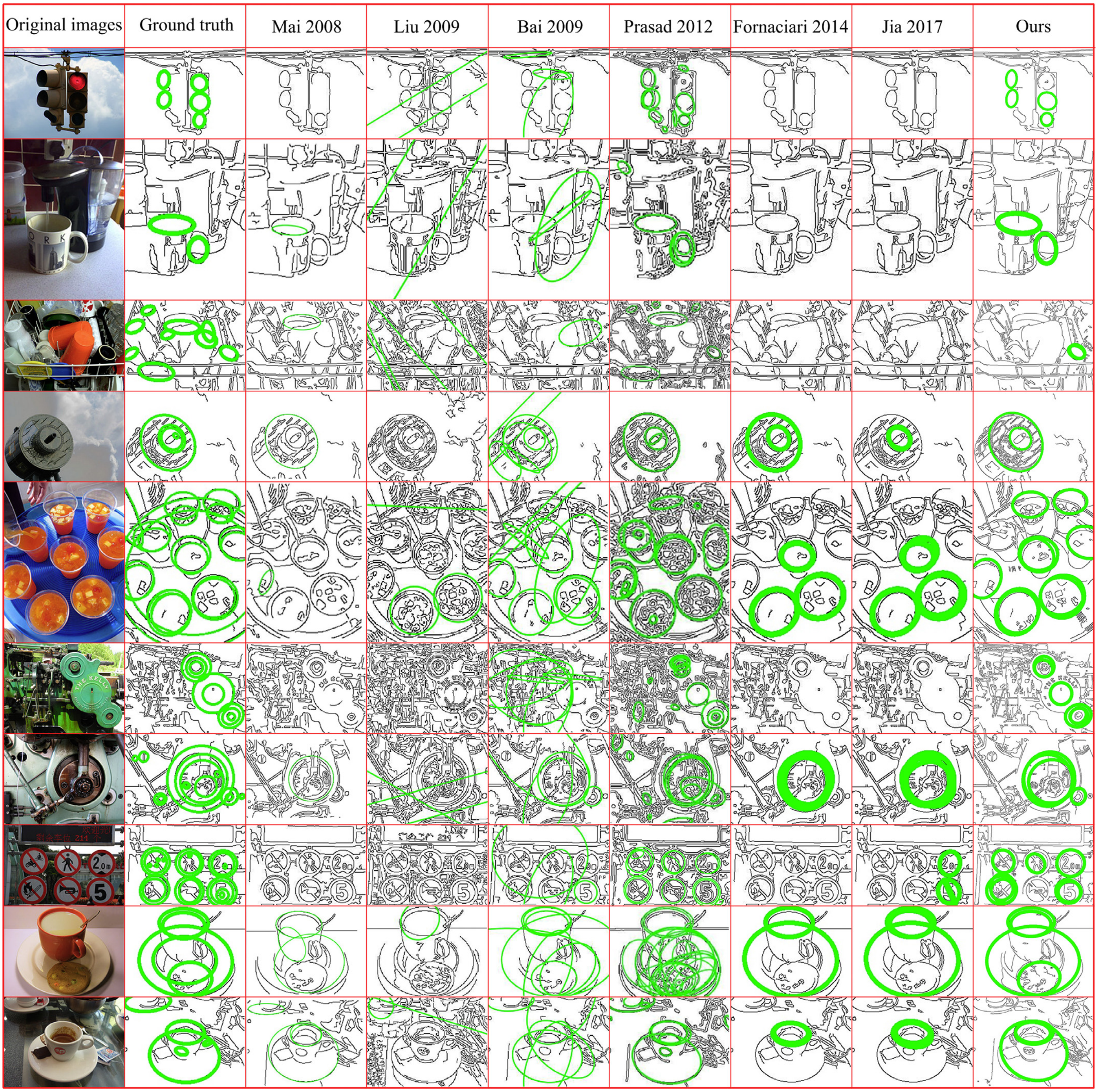

Fig. 20. The detection results of the real images in Dataset HX.

7. Conclusion and future works

In this paper, we propose a geometric approach based on gradient analysis to detecting ellipses in practical scenarios with a high detection effectiveness and low execution time. The balance of low execution time and detection effectiveness is obtained through (a) effective combination of rejecting curves that do not contribute to ellipse detection, (b) smart grouping of arcs potentially suitable for detecting ellipses, and (c) utilizing pre-computed gradient information powerfully at multiple steps to alleviate computation needs of the conventional approaches that do not exploit gradients directly. Experimental results demonstrate that the proposed method performs better than six state-of-the art methods in terms of overall performance. Further, it can be used in practical scenarios such as industrial fields and automatic unmanned drive. Furthermore, we can achieve this performance at video rates, with average detection rates of 40 images per second or better.

Future works will update this algorithm to detect even more heavily occluded ellipses, which allows a robot to sort multiple circle and elliptic mechanical components in cluttered environments by ellipse detection in real time.

References

[1] J. Liang, Y. Wang, X. Zeng, Robust ellipse fitting via half-quadratic and semidefinite relaxation optimization, IEEE Trans. Image Process. 24 (2015) 4276–4286.

[2] S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, H. Haario, Segmentation of overlapping elliptical objects in silhouette images, IEEE Trans. Image Process. 24 (2015) 5942–5952.

[3] Y. Garcés, A. Guerrero, P. Hidalgo, R.E. López, C.D. Wood, R.A. Gonzalez, et al., Automatic detection and measurement of viral replication compartments by ellipse adjustment, Sci. Rep. 6 (2016).

[4] A. Soetedjo, K. Yamada, Fast and robust traffic sign detection, in: 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005, pp. 1341–1346.

[5] D.S. Barwick, Very fast best-fit circular and elliptical boundaries by chord data, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 1147–1152.

[6] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.

[7] X. Zhu, D. Ramanan, Face detection, pose estimation, and landmark localization in the wild, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, 2012, pp. 2879–2886.

[8] N. Greggio, L. Manfredi, C. Laschi, P. Dario, M.C. Carrozza, Robotcub implementation of real-time least-square fitting of ellipses, in: *8th* IEEE-RAS International Conference on Humanoid Robots, 2008. Humanoids 2008, 2008, pp. 174–181.

[9] S.S.M. Salehian, M. Khoramshahi, A. Billard, A dynamical system approach for softly catching a flying object: theory and experiment, IEEE Trans. Rob. 32 (2016) 462–471.

[10] K. Ono, T. Ogawa, Y. Maeda, S. Nakatani, G. Nagayasu, R. Shimizu, et al., Detection, localization and picking up of coil springs from a pile, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 3477–3482.

[11] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1972) 11–15.

[12] S.-C. Zhang, Z.-Q. Liu, A robust, real-time ellipse detector, Pattern Recognit. 38 (2005) 273–287.

[13] A.A. Sewisy, F. Leberl, Detection ellipses by finding lines of symmetry in the images via an hough transform applied to straight lines, Image Vision Comput. 19 (2001) 857–866.

[14] N. Guil, E.L. Zapata, Lower order circle and ellipse Hough transform, Pattern Recognit. 30 (1997) 1729–1744.

[15] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (RHT), Pattern Recognit. Lett. 11 (1990) 331–338.

[16] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognit. 24 (1991) 303–316.

[17] H.I. Cakir, B. Benligiray, C. Topal, Combining feature-based and model-based approaches for robust ellipse detection, in: 2016 24th European Signal Processing Conference (EUSIPCO), 2016, pp. 2430–2434.

[18] S. Chen, R. Xia, J. Zhao, Y. Chen, M. Hu, A hybrid method for ellipse detection in industrial images, Pattern Recognit. 68 (2017) 82–98.

[19] R. Grbić, D. Grahovac, R. Scitovski, A method for solving the multiple ellipses detection problem, Pattern Recognit. 60 (2016) 824–834.

[20] A.Y.-S. Chia, S. Rahardja, D. Rajan, M.K. Leung, A split and merge based ellipse detector with self-correcting capability, IEEE Trans. Image Process. 20 (2011) 1991–2006.

[21] D.K. Prasad, M.K. Leung, S.-Y. Cho, Edge curvature and convexity based ellipse detection method, Pattern Recognit. 45 (2012) 3204–3221.

[22] F. Mai, Y. Hung, H. Zhong, W. Sze, A hierarchical approach for fast and robust ellipse extraction, Pattern Recognit. 41 (2008) 2512–2524.

[23] E. Kim, M. Haseyama, H. Kitajima, Fast and robust ellipse extraction from complicated images, in: Proceedings of IEEE Information Technology and Applications, 2002.

[24] L. Libuda, I. Grothues, K.-F. Kraiss, Ellipse detection in digital image data using geometric features, Adv. Comput. Graphics Comput. Vision (2007) 229–239.

[25] D.K. Prasad, M.K. Leung, C. Quek, ElliFit: an unconstrained, non-iterative, least squares based geometric ellipse fitting method, Pattern Recognit. 46 (2013) 1449–1465.

[26] H. Dong, I.-M. Chen, D.K. Prasad, Robust ellipse detection via arc segmentation and classification, in: 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 66–70.

[27] Y.-B. Lin, C.-P. Young, High-precision bicycle detection on single side-view image based on the geometric relationship, Pattern Recognit. 63 (2017) 334–354.

[28] M. Fornaciari, A. Prati, R. Cucchiara, A fast and effective ellipse detector for embedded vision applications, Pattern Recognit. 47 (2014) 3693–3708.

[29] X. Bai, C. Sun, F. Zhou, Splitting touching cells based on concave points and ellipse fitting, Pattern Recognit. 42 (2009) 2434–2446.

[30] H.I. Cakir, C. Topal, C. Akinlar, An occlusion-resistant ellipse detection method by joining coelliptic arcs, in: European Conference on Computer Vision, 2016, pp. 492–507.

[31] K. Hahn, S. Jung, Y. Han, H. Hahn, A new algorithm for ellipse detection by curve segments, Pattern Recognit. Lett. 29 (2008) 1836–1841.

[32] B.-K. Kwon, Z. Teng, T.-J. Roh, D.-J. Kang, Fast ellipse detection based on three point algorithm with edge angle information, Int. J. Control Autom. Syst. 14 (2016) 804.

[33] Z.-Y. Liu, H. Qiao, Multiple ellipses detection in noisy environments: a hierarchical approach, Pattern Recognit. 42 (2009) 2421–2433.

[34] Y. Xie, Q. Ji, A new efficient ellipse detection method, in: Proceedings 16th International Conference on Pattern Recognition, 2002, 2002, pp. 957–960.

[35] J. Ni, M.K. Singh, C. Bahlmann, Fast radial symmetry detection under affine transformations, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 932–939.

[36] B. Jian, B.C. Vemuri, Robust point set registration using Gaussian mixture models, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 1633–1645.

[37] C. Liu, W. Hu, Effective method for ellipse extraction and integration for spacecraft images, Opt. Eng. 52 (2013) 057002.

[38] C. Arellano, R. Dahyot, Robust ellipse detection with Gaussian mixture models, Pattern Recognit. 58 (2016) 12–26.

[39] S. Mulleti, C.S. Seelamantula, Ellipse fitting using the finite rate of innovation sampling principle, IEEE Trans. Image Process. 25 (2016) 1451–1464.

[40] V. Pătrăucean, P. Gurdjos, R.G. von Gioi, Joint A Contrario ellipse and line detection, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 788–802.

[41] C. Liu, W. Hu, Ellipse fitting for imaged cross sections of a surface of revolution, Pattern Recognit. 48 (2015) 1440–1454.

[42] Q. Jia, X. Fan, Z. Luo, L. Song, T. Qiu, A fast ellipse detector using projective invariant pruning, IEEE Trans. Image Process. (2017).

[43] M. Cicconet, K. Gunsalus, D. Geiger, M. Werman, Ellipses from triangles, in: 2014 IEEE International Conference on Image Processing (ICIP), 2014, pp. 3626–3630.

[44] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. (1986) 679–698.

[45] D.K. Prasad, M.K.H. Leung, C. Quek, M.S. Brown, DEB: definite error bounded tangent estimator for digital curves, IEEE Trans. Image Process. 23 (2014) 4297–4310.

[46] H. Freeman, R. Shapira, Determining the minimum-area encasing rectangle for an arbitrary closed curve, Commun. ACM 18 (1975) 409–413.

[47] T.M. Nguyen, S. Ahuja, Q.J. Wu, A real-time ellipse detection based on edge grouping, in: IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009., 2009, pp. 3280–3286.

[48] C. Akinlar, C. Topal, EDCircles: a real-time circle detector with a false detection control, Pattern Recognit. 46 (2013) 725–740.

[49] K.C. Chen, N. Bouguila, D. Ziou, Quantization-free parameter space reduction in ellipse detection, Expert Syst. Appl. 38 (2011) 7622–7632.

[50] Y. Lei, K.C. Wong, Ellipse detection based on symmetry, Pattern Recognit. Lett. 20 (1999) 41–47.

[51] H. Liu, Z. Wang, Geometric property based ellipse detection method, J. Visual Commun. Image Represent. 24 (2013) 1075–1086.

[52] D.K. Prasad, M.K. Leung, An ellipse detection method for real images, in: 2010 25th International Conference of Image and Vision Computing New Zealand (IVCNZ), 2010, pp. 1–8.

[53] P.-K. Ser, W.-C. Siu, Novel detection of conics using 2-D Hough planes, IEE Proc.-Vision, Image Signal Process. 142 (1995) 262–270.

[54] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans. Pattern Anal. Mach. Intell. 17 (1995) 790–799.

[55] A.S. Aguado, M.E. Montiel, M.S. Nixon, On using directional information for parameter space decomposition in ellipse detection, Pattern Recognit. 29 (1996) 369–381.

[56] A. Fernandes, "A correct set of equations for the real-time ellipse hough transform algorithm," Technical Report 2009.

[57] C. Basca, M. Talos, R. Brad, Randomized Hough transform for ellipse detection with result clustering, in: The International Conference on Computer as a Tool, 2005. EUROCON 2005., 2005, pp. 1397–1400.

[58] D.K. Prasad, M.K. Leung, Clustering of ellipses based on their distinctiveness: an aid to ellipse detection algorithms, in: 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010, pp. 292–297.

[59] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009, 2009, pp. 248–255.

[61] A.Y. Chia, D. Rajan, M.K. Leung, S. Rahardja, A split and merge based ellipse detector, in: 15th IEEE International Conference on Image Processing, 2008. ICIP 2008., 2008, pp. 3212–3215.

Huixu Dong received the B.Sc. degree in mechatronics engineering from Harbin Institute of Technology in China, in 2013. Currently, he is pursuing Ph.D. at Robotics Research Centre of Nanyang Technological University, Singapore. His current research interests include perception based on robotic grasp, robot-oriented image processing and robot-oriented artificial intelligence.

Dilip K. Prasad received the B. Tech. degree in computer science and engineering from IIT Dhanbad, Dhanbad, India, in 2003 and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2013. He is currently a Senior Research Fellow at Nanyang Technological University, Singapore. He has authored over 55 internationally peer-reviewed research articles. His research interests include image processing, pattern recognition, and computer vision.

I-Ming Chen received the B.S. degree from National Taiwan University in 1986, and M.S. and Ph.D. degrees from California Institute of Technology, Pasadena, CA in 1989 and 1994 respectively. He is a full professor of the School of Mechanical and Aerospace Engineering, directions of Robotics Research Centre and Intelligent System Centre in Nanyang Technological University, Singapore. He is Fellow of ASME and Fellow of IEEE, General Chairman of 2017 IEEE International Conference on Robotics and Automation (ICRA2017). He is a senior editor of IEEE transaction on robotics. He also acts as the Deputy Program Manager of A*STAR SERC Industrial Robotics Program to coordinate project and activities under this multi-institutional program involving NTU, NUS, SIMTech, A*STAR I2R and SUTD. He is a member of the Robotics Task Force 2014 under the National Research Foundation which is responsible for Singapore's strategic R&D plan in future robotics. He works on many different topics in robotics, such as mechanism, actuator, human-robot interaction, perception and grasp, and industrial automation.