Graphical Models 116 (2021) 101110

Contents lists available at ScienceDirect

Graphical Models

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

Combining convex hull and directed graph for fast and accurate ellipse detection

Zeyu Shen [a,b,c], Mingyang Zhao [d], Xiaohong Jia [d], Yuan Liang [e], Lubin Fan [e], Dong-Ming Yan [*,a,c,b]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100149, China
[c] State Key Laboratory of Hydro-Science and Engineering, Tsinghua University, Beijing 100084, China
[d] KLMM, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
[e] Alibaba Group, Hangzhou 311121, China

A R T I C L E I N F O

Keywords:
Ellipse detection
Edge following
Hough transform
RANSAC

A B S T R A C T

Detecting ellipses from images is a fundamental task in many computer vision applications. However, due to the complexity of real-world scenarios, it is still a challenge to detect ellipses accurately and efficiently. In this paper, we propose a novel method to tackle this problem based on the fast computation of *convex hull* and *directed graph*, which achieves promising results on both accuracy and efficiency. We use Depth-First-Search to extract branch-free curves after adaptive edge detection. Line segments are used to represent the curvature characteristic of the curves, followed by splitting at sharp corners and inflection points to attain smooth arcs. Then the convex hull is constructed, together with the distance, length, and direction constraints, to find co-elliptic arc pairs. Arcs and their connectivity are encoded into a sparse directed graph, and then ellipses are generated via a fast access of the adjacency list. Finally, salient ellipses are selected subject to strict verification and weighted clustering. Extensive experiments are conducted on eight real-world datasets (six publicly available and two built by ourselves), as well as five synthetic datasets. Our method achieves the overall highest F-measure with competitive speed compared to representative state-of-the-art methods.

1. Introduction

As one of the most common geometric primitives, ellipses often appear in natural and artificial scenes. In particular, 3D circular or elliptic objects are usually projected as ellipses on the image. Therefore, accurate detection and localization of ellipses from images provides us with a powerful tool for pattern recognition and visual understanding [1]. Actually, ellipse detection is broadly applied in the fields of camera calibration [2,3], industrial component inspection [4,5], traffic sign detection [6,47], cell segmentation [7], pupil tracking [8], object localization for the robotic platform [9], and so on. See Fig. 1 as a reference.

Although ellipse detection problem has gained a lot of attention in literature, it is still very challenging. The major difficulties are the presence of noise, disturbance or occlusion by other objects, image blur or flaw, and varying illuminations. These issues either break the elliptic boundaries as several low-quality arc segments, thus make the differential computations such as tangents inaccurate, or leave the ellipse partially visible, which degrades the ellipse fitting quality. Besides, the requirement of fast detection for real-time scenarios further brings the difficulty.

As a well-known geometric primitive detector, *Hough transform* (HT) is explored for ellipse detection by numerous work [10–15]. However, due to the five-dimensional (5D) parameter space of an ellipse, HT consumes a noticeable amount of storage and time [16,17], which seriously prevents its applications, especially for complicated images needing high-speed processing. Besides, HT suffers from the careful tuning of bin size and peak threshold, hence it may detect false ellipses or lose positive ones if the model parameters are not optimal.

The recent methods based on the *edge following* technique exhibit promising detection performance, in which the connectivity between edge pixels, continuity of arcs are used [18]. Candidate ellipses are generated by incremental least-squares fitting or arc grouping. However, direct ellipse fitting for short arcs inevitably results in errors [19].

* Corresponding author.
E-mail address: yandongming@gmail.com (D.-M. Yan).

https://doi.org/10.1016/j.gmod.2021.101110
Received 25 February 2021; Accepted 5 May 2021
Available online 15 May 2021

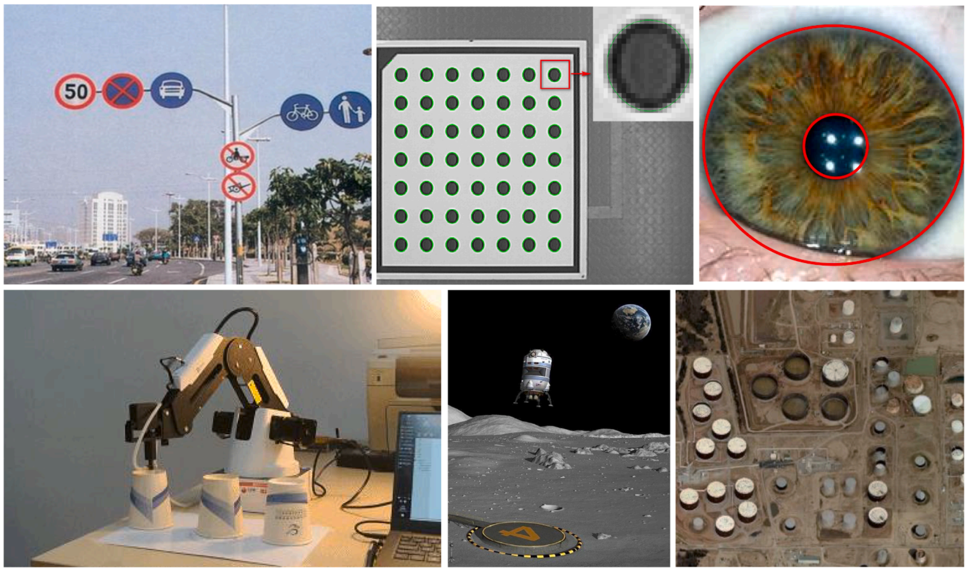

Fig. 1. A wide variety of applications of ellipse detection in the real world, which provides us with a powerful tool for multiple visual understanding tasks.

Although other methods first group arcs together, complex arc grouping strategies are usually designed, where differential calculations or HT are invoked, hence they are more sensitive to noise or less efficient.

Different from aforementioned methods, in this paper, we introduce a new ellipse detector by a more effective arc grouping scheme, aiming to improve the detection ability in both accuracy and efficiency. We use Depth-First-Search (DFS) to extract continuous edge curves, followed by the identification of sharp corners and inflection points to attain smooth arcs. Then, the convex hull is first introduced to distinguish the convexity of arc pairs, along with the fast computation of arc distance, length, and directions. Due to the avoidance of calculations of gradients and tangents for the edge pixels, our method is more robust to noise. Based on these constraints, a sparse directed graph is built, by which arc pairs and their connectivity can be fast accessed to generate candidate ellipses. Finally, a stringent verification and a discriminative clustering are applied to further improve the detection accuracy. In a nutshell, the contributions of this work are as follows:

- a fast and accurate ellipse detector competent of detecting complicated real-world images, as well as occluded, overlapping, concentric, and concurrent ellipses;
- a novel arc grouping scheme based on the efficient computation of the convex hull and sparse directed graph, together with a more discriminative clustering criterion to depress repetitive ellipses, and
- the superior performance with less time consumption on a series of datasets compared with the representative state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, we briefly review the most related work from the perspective of ellipse generation and verification. The detailed steps of our method are presented in Section 3. Then we describe the datasets, experimental results, and performance of the proposed approach in Section 4. A general conclusion and future work are given in Section 5.

2. Related work

The significance of ellipse detection is witnessed by the large amount of work presented in the literature. In general, they can be classified as Hough transform based methods and edge following techniques.

2.1. Hough transform

Most of the traditional methods for ellipse detection rely on HT [20] to estimate the parameters, which casts the detection problem into a peak finding process. The basic principle of HT is voting each edge pixel to a 5D parameter space, and then the local peak exceeding a certain threshold is selected out as an ellipse. Although simple for implementation, it is usually unpractical to directly apply HT to ellipse detection in real images, due to the expensive storage and time load, which are $O(m^5)$ and $O(n^5)$ [17], respectively. To reduce the memory consumption, accelerate the detection, and improve the accuracy of the standard HT, a great number of variants are put forward. *Randomized HT* (RHT) [21] and *probability HT* (PHT) [22] sample subset of pixels rather than all pixels for voting, and thus a many-to-one scheme is built to replace the primary one-to-many scheme. McLaughlin [13] extends RHT to detect ellipses by randomly selecting three non co-linear points, but it is sensitive to occlusion and overlapping ellipses. Lu et al. [23] propose the iterative RHT to circumvent the noise susceptibility of RHT, but it has to divide an image into sub-images for multiple ellipse detection. On the other hand, some methods combine geometric properties of ellipses with HT to lower the voting space. Xie and Ji [24] estimate the semi-axis length of the hypothetical ellipses to reduce the 5D space to 1D. Similarly, Chia et al. [25] use the foci feature to realize the same effect. Geometric symmetry is also explored to decompose the voting space, by which elliptic centers are first located and then the remaining parameters are solved [26,27]. However, these methods are easily deteriorated by occluded or semi ellipses. Besides, we point out that HT based methods are still inefficient in practice, prone to generate false detection with the number of ellipses increasing, suffer from noise and background clutter, and take much effort to tune the required parameters such as the bin size and peak threshold [18].

2.2. Edge following

Different from HT working on the pixel level, edge following methods utilize continuous arcs for ellipse detection, in which edge curves are extracted and geometric characteristics such as convexity or tangents are explored. Compared with HT, edge following methods are more efficient, and currently are the benchmark among the ellipse detection field. For instances, Kim et al. [29] first extract arcs approximated by short line segments, and then frequently use the least-squares fitting to estimate elliptic parameters. Libuda et al. [30] improve the

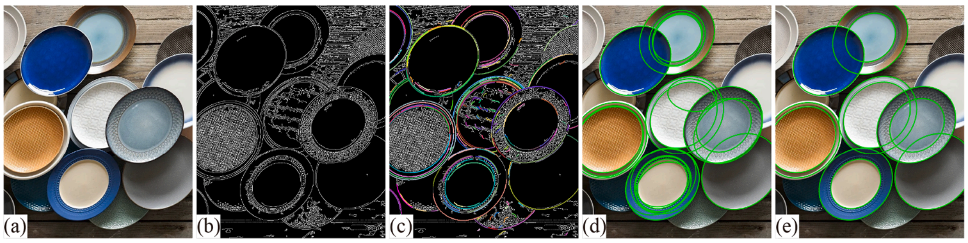

Fig. 2. The workflow of our proposed method. (a) Input image; (b) edge detection by adaptive Canny detector [28]; (c) arc extraction via the identification of sharp corners and infection points; (d) candidate ellipse generation after arc grouping; (e) finally detected ellipses after validation and clustering. The proposed method is competent to detect ellipses in complex real-world images.

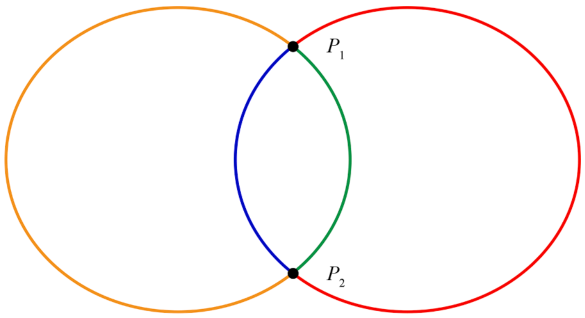

Fig. 3. The two bifurcation points P_1 and P_2 separate the edge curve into four branch-free curves indicated by different colors.

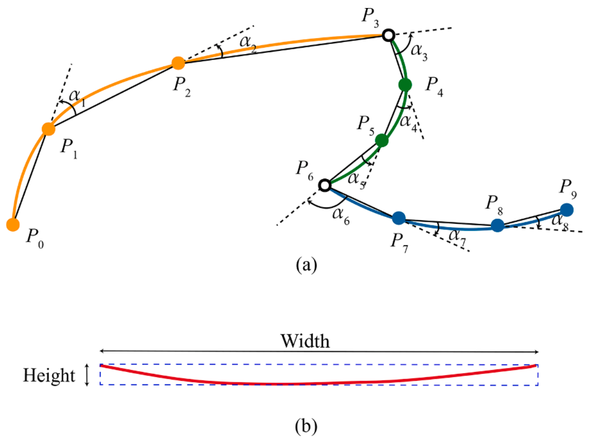

Fig. 4. (a) An edge curve is approximated by nine line segments. From the inner and cross products computation, we find that α_3 is a sharp corner while P_6 is an inflection point. (b) The aspect ratio of the minimum area bounding box (dashed rectangle) is used to remove straight segments for fast detection.

performance of [29] with less memory consumption. Mai et al. [31] inherit the idea of [29], but further link line segments to form arcs based on the adjacency and curvature constraints. However, due to the out of consideration for validating candidate ellipses, there are multiple false detection. Chia et al. [32] adopt a split and merge scheme for arcs, where co-elliptic arc pairs are grouped as an alignment problem. Nevertheless, the complex and iterative optimization process hinders its real-time usage in practice. The detector proposed by Prasad et al. [1] makes use of the information of edge convexity and curvatures for arc grouping, in which the search region is first determined, followed by a line segment length judgment. Due to the incorporation of tangents and line-arc intersecting, this method [1] is more complicated than our proposed method. Although overall improvements are attained, it suffers from long computational time. Fornaciari et al. [33] propose a fast ellipse detector for the embedded vision system, in which arcs are classified into four quadrants based on the gradient computation, and then parameters are estimated by the parallel chord theorem and 2D HT voting. Jia et al. [34] promote the performance of [33] by introducing a projective invariant to prune line segments and group arcs. However, both [33,34] encounter the same problem, that is the number of arcs for grouping must be at least three, which is impractical for occluded or semi ellipses. Dong et al. [35] take the similar scheme of [33] and incorporate the gradient analysis, but also divide the arcs into four quadrants, hence inevitably break the integrity of complete ellipses. Recently, Lu et al. [18] revisit the line detection method proposed by Pătrăucean et al. [36] to attain a high-quality ellipse detector, because of the iterative linking of line segments and voting for arcs, the method is much slower than [34]. Meng et al. [37] design an arc adjacency matrix (AAM) to represent the arc pair relationship, in which curvatures and tangents are computed to make AAM sparse. However, as [24,38] pointed, curvatures and tangents are more sensitive to noise than edge points.

3. Methodology

Our method adopts a standard edge following framework, which contains three main steps: (1) edge detection and elliptic arc extraction; (2) arc grouping and candidate ellipse generation; (3) ellipse validation and clustering. The workflow of our method is shown in Fig. 2. We explain the details of each step in the following.

3.1. Edge detection and elliptic arc extraction

Given an input image, the very first step is to extract the edge map. Here, we implement an adaptive Canny detector [28] for this purpose, because of the efficiency and avoidance of parameter tuning. The higher threshold ensures that only 10% of the image pixels are marked as edge pixels, while the lower threshold equates 0.3 times of the higher threshold. To attain branch-free curves as shown in Fig. 3, given a seed point, we use the Depth-First-Search (DFS) to expand continuous curves according to the 8-connected domain of the edge points.

After the attainment of branch-free curves, we continue to extract smooth arcs. To this end, a parameter free method [39] improved from Ramer-Douglas-Peucker (RDP) algorithm [40] is first applied to simplify curves via a series of line segments $\{l_i = P_{i-1}P_i | P_i \in \mathbb{R}^2\}_{i=1}^{n}$, by which we can effectively compute both the magnitude and direction of edge curvatures, as illustrated in Fig. 4(a). An angle α_i is a sharp corner, indicating the major variation in curvature magnitude, if

$$\cos\alpha_i = \frac{\overrightarrow{l}_i}{\|\overrightarrow{l}_i\|_2} \cdot \frac{\overrightarrow{l}_{i+1}}{\|\overrightarrow{l}_{i+1}\|_2} \leq \cos Th_\theta, \tag{1}$$

where $\overrightarrow{l}_*$ is the directional vector of the line segment l_*, and Th_θ is the angle threshold. Further, a point between l_i and l_{i+1} is an infection point, indicating the variation in curvature direction, if

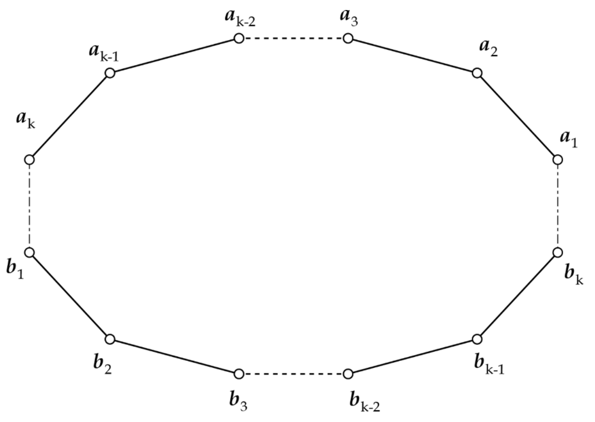

Fig. 5. Convex hull test with k points on each of the two arcs.

$$\left(\frac{\overrightarrow{l}_{i-1}}{\|\overrightarrow{l}_{i-1}\|_2}\times\frac{\overrightarrow{l}_i}{\|\overrightarrow{l}_i\|_2}\right)\cdot\left(\frac{\overrightarrow{l}_i}{\|\overrightarrow{l}_i\|_2}\times\frac{\overrightarrow{l}_{i+1}}{\|\overrightarrow{l}_{i+1}\|_2}\right)=-1. \tag{2}$$

Hence, α_3 is a sharp corner while P_6 is an inflection point in Fig. 4(a). Then we split curves at these points to obtain arc segments.

To speed up the following processing, we further remove straight segments based on the minimum area bounding box as illustrated in Fig. 4(b). We remove the segment if its aspect ratio

$$\frac{\max\{\text{Height},\text{Width}\}}{\min\{\text{Height},\text{Width}\}}>Th_r.$$

Since arc quality is critical for arc grouping, we further access each arc Arc_i by computing its inlier ratio via ellipse fitting, which is defined as

$$I(\text{Arc}_i)=\frac{1}{|\text{Arc}_i|}\sum_{p\in\text{Arc}_i}1\{dist(p,e)<\varepsilon\}, \tag{3}$$

where 1 is the indicator function and equates to one if and only if the distance from the edge pixel p to the ellipse e is less than ε equal to one pixel in default. Arcs with low inlier ratio, *i.e.*, $I(\text{Arc}_i)<Th_{ir}$, where Th_{ir} is the threshold, are regarded as non-elliptic arcs thus are deleted. To keep consistency between different arcs, edge points of each arc are stored in the counter-clockwise order.

3.2. Arc grouping and candidate ellipse generation

Since short arcs may result in major fitting errors, we first group them from the same ellipse together by a local to global scheme. The local search aims to link adjacent arc pairs caused by noise interference, while the global process elaborates to group distant ones.

We introduce convex hull to represent the ellipse convexity, as illustrated in Fig. 5. For every arc pair, supposing k points are sampled on each arc, referred to as $\{a_1,a_2,\cdots,a_k\}$ and $\{b_1,b_2,\cdots,b_k\}$, then there are totally $2k$ points involved for convex test, thereby $2k$ cross products are calculated. Given that each arc is convex, actually we merely need to check the following four cross products:

$$\left\{\overrightarrow{a_{k-1}a_k}\times\overrightarrow{a_kb_1},\ \overrightarrow{a_kb_1}\times\overrightarrow{b_1b_2}\overrightarrow{b_{k-1}b_k}\times\overrightarrow{b_ka_1},\ \overrightarrow{b_ka_1}\times\overrightarrow{a_1a_2}.\right. \tag{4}$$

In our method, for efficiency, we directly sample the endpoints and

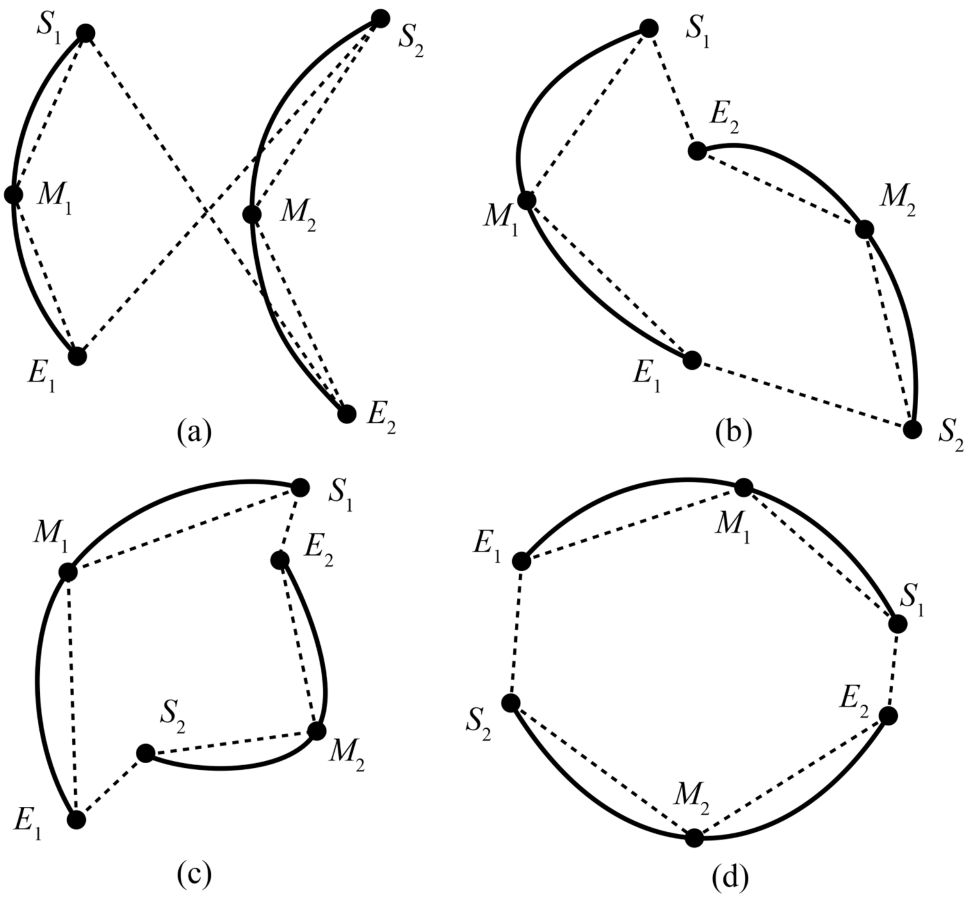

Fig. 6. Grouping arc pairs based on the convex hull computation, where S_i, E_i, and M_i are the endpoints and midpoints of Arc_i. The arcs in Case (d) form a convex hull, thereby they can be grouped together, while the others cannot.

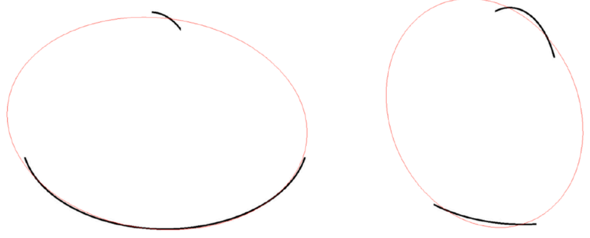

Fig. 7. Fitting an ellipse for two arcs. Left: Arc pair with large difference in length, where the short arc plays very limited impact on the fitting; Right: Arc pair with the similar length, the fitted ellipse is influenced by them simultaneously.

midpoints of two arcs to define the convex hull, as shown in Fig. 6, thereby there are totally six points. To check whether the polygon formed by these six points is convex, we simply judge whether the sign of the cross product of adjacent line segments are all positive, and two arcs Arc_i, Arc_j are said constituting a convex hull if

$$\left\{ \mathrm{sgn}\left(\overrightarrow{M_1E_1} \times \overrightarrow{E_1S_2}\right) > 0,\quad \mathrm{sgn}\left(\overrightarrow{E_1S_2} \times \overrightarrow{S_2M_2}\right) > 0 \mathrm{sgn}\left(\overrightarrow{M_2E_2} \times \overrightarrow{E_2S_1}\right) > 0,\quad \mathrm{sgn}\left(\overrightarrow{E_2S_1} \times \overrightarrow{S_1M_1}\right) > 0 \right. \tag{5}$$

From the local perspective, adjacent arcs tend to come from the same ellipse. We find arc pairs whose end-point distance is no more than one pixel, and merge them together if (1) the endpoints and midpoints of them constitute a convex hull and (2) the inlier ratio of them is larger than each arc after merging. Local grouping significantly reduces the number of arcs participating in the global grouping, hence accelerating the detection process. Actually, noise causes many adjacent arcs, and most of them can be merged, while other invalid arc pairs are directly skipped in the subsequent processing. When two arcs Arc_i and Arc_j are not adjacent enough, we try to group them again by four global constraints.***Arc length constraint.*** If the length ratio of Arc_i and Arc_j satisfies

$$1/Th_{lr} < |\mathrm{Arc}_i|/|\mathrm{Arc}_j| < Th_{lr},$$

then they are checked by subsequent constraints. Otherwise, the arc pair is invalid and ignored. When two arcs with large length difference are fitted to an ellipse, the result ellipse will fit the longer arc better, whereas the shorter one will have very limited impact on the fitting, as shown in Fig. 7. Besides, arc length constraint merely involves a simple comparison, and can effectively reduce the subsequent computation. Hence, we adopt it to accelerate the detection.

Distance constraint. Although global constraints aim to group distant arcs, two arcs apart largely are also less likely from the same ellipse. Arc_i and Arc_j are said satisfying the distance constraint if

$$\frac{dist(M_i, M_j)}{\max\{|\mathrm{Arc}_i|, |\mathrm{Arc}_j|\}} < Th_d,$$

where M_* is the middle point of Arc_*, and $dist(M_i, M_j)$ is the distance between two middle points.

Convex hull constraint. According to the convexity of ellipses, Arc_i and Arc_j can be grouped if their endpoints and midpoints form a convex hull.***Direction constraint.*** Arc pair $\langle \mathrm{Arc}_i, \mathrm{Arc}_j \rangle$ satisfying the above criterion are called co-elliptic, referred to as $\mathrm{Arc}_i \rightarrow \mathrm{Arc}_j$. It should also note that the arcs are connected in order, that is, $\mathrm{Arc}_i \rightarrow \mathrm{Arc}_j$ and $\mathrm{Arc}_j \rightarrow \mathrm{Arc}_i$ are two different situations. Arcs should be connected counter-clockwise, as shown in Fig. 8 (a), C is the center of the corresponding ellipse, and θ_i represents the rotation angle from the positive x-axis to the vector $\overrightarrow{CM_i}$. For example, M_1 can be connected to M_2 if M_1 is co-linear with C and M_2 after a rotation with the angle no more than 180°,

$$fmod(\theta_2 - \theta_1 + 360^\circ, 360^\circ) \langle 180^\circ, \tag{6}$$

where $fmod(x, y)$ stands for the floating point remainder of the division operation x/y. In practice, we approximate the rotation angles $\{\theta_i\}$ based on the pre-fitted ellipse in the inlier ratio step to speed up detection.

Local and global grouping discovers the relationship between any

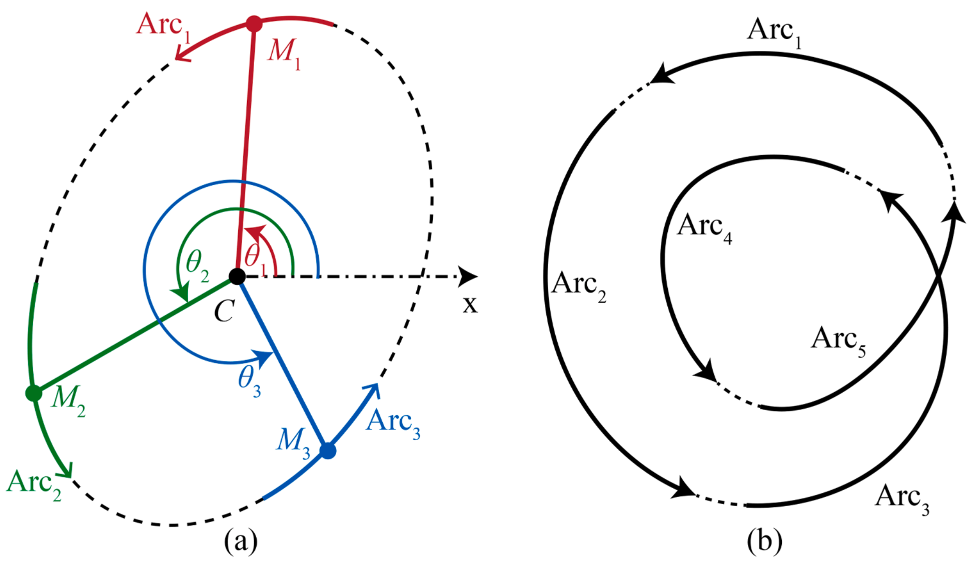

Fig. 8. (a) Connection of counter-clockwise arcs and computation of rotation angles represented by θ_i. (b) A path with self-intersection, which can be effectively removed by our method.

```
Require: $I$: Input image, $Th_{ss}$: Salient score threshold
Ensure: $\{e_j\}_{j=1}^{N}$: detected ellipses
 1: Compute edge map $I_e$ using adaptive Canny operator with Gaussian kernel
 2: Collect continuous curve set $C$ from $I_e$ by depth-first-search
 3: Initialize arc set $A := \emptyset$
 4: for each curve $c_i \in C$ do
 5:     Use RDP to approximate $c_i$ by segments $\{l_i = P_{i-1}P_i | P_i \in \mathbb{R}^2\}_{i=1}^{n}$
 6:     Calculate angles between adjacent line segments $l_i$ and $l_{i+1}$
 7:     Split $c_i$ at sharp corners and inflection points to get arcs $\{a_i\}_{i=1}^{m}$
 8:     Remove arcs that are too short or straight
 9:     Sort the edge points of $a_i$ in counter clockwise to get arc set $A_i$
10:     $A := A \cup A_i$
11: end for
12: Initialize the directed graph $D :=< V, E >$, where $V = A, E = \emptyset$
13: for each arc pair $< a_i, a_j >\in A \times A$ do
14:     if $< a_i, a_j >$ satisfies the four grouping constraints then
15:         $E := E \cup < a_i, a_j >$
16:     end if
17: end for
18: Initialize result ellipse set $Ells := \emptyset$
19: while exist unvisited vertex $u^* \in V$ do
20:     Adopt depth-first-search on $u^*$ to find simple loops
21:     Fit ellipse to the arcs to get candidate $Ell^*$
22:     if inlier ratio of $Ell^* \geq T_{ss}$ then
23:         $Ells := Ells \cup Ell^*$
24:     end if
25: end while
26: Cluster the ellipses according to the weighted Euclidean distance
27: Output the detected ellipses $\{e_j\}_{j=1}^{N}$
```

Algorithm 1. Fast and accurate ellipse detection

two arcs, by which we construct a directed graph to encode the relationship of all arcs. In the graph, vertices stand for arcs, and directed edges represent the connected co-elliptic arc pairs in counter-clockwise direction. Because of the above strict pairing constraints, the graph is usually sparse, thereby we use the adjacency list to reduce the memory usage.

By depth-first searching the directed graph, we can obtain a path

$$\mathrm{Arc}_{k_1} \rightarrow \mathrm{Arc}_{k_2} \rightarrow \cdots \rightarrow \mathrm{Arc}_{k_n},$$

which represents a group of arcs where any two adjacent arcs are co-elliptic. Thanks to the data structure of adjacency list, we can merely visit the neighbors of a vertex without traversing the other vertices, hence greatly reduces the access time consumption. Note that there may exist complex paths with self-intersection as illustrated in Fig. 8 (b). In this case, we use the following criteria

$$R = \frac{1}{360^\circ} \sum_{i=1}^{n} \Delta\theta_i$$

to filter out self-intersection paths if $R \neq 1$, where $\Delta\theta_i$ is defined as

$$\Delta\theta_i = fmod(\theta_{i+1 \bmod n} - \theta_i + 360^\circ, 360^\circ).$$

Intuitively, R represents the number of circles around the center when a virtual point moves along the path. For valid paths, R is always equal to one. Through the searching process, all co-elliptic arc groups are found, and then a direct least-squares-based ellipse fitting [41] is applied to attain candidate ellipses.

3.3. *Ellipse validation and clustering*

Due to the discrete properties of edge pixels, there may exist false ellipses among candidates. To further improve the detection accuracy, we execute an ellipse validation and compute the salient score $S(e)$ for each candidate ellipse e formed by the arc group G, which is defined as

$$S(e) = \frac{1}{\sum_{\mathrm{Arc} \in G} |\mathrm{Arc}|} \sum_{\mathrm{Arc} \in G} \sum_{p \in \mathrm{Arc}} 1\{dist(p, e) < \varepsilon\}, \quad (7)$$

where p is the edge pixel from the corresponding Arc in the same group. A candidate ellipse is validated to be true if $S(e) \geq Th_{ss}$, otherwise we remove it because of the unreliability. Let $e = (a, b, x_c, y_c, \theta)$ be the ellipse parameters, where a, b are the semi-axis length, (x_c, y_c) is the elliptic center, and θ is the rotation angle along the horizontal axis. Then, we use a weighted clustering scheme based on the Euclidean distance to evaluate the distinctiveness of two ellipses e_i and e_j

$$D(e_i, e_j) = \sqrt{\sum_{\lambda=1}^{5} k_\lambda \cdot (e_{i\lambda} - e_{j\lambda})^2}. \quad (8)$$

Ellipses e_i and e_j are clustered together if $D(e_i, e_j) < 20$ (suggested by [14]). The weight k_λ is equal to one except for the rotation angle θ that is defined as

$$k_\theta = \min\left\{\frac{a_i - b_i}{a_i + b_i}, \frac{a_j - b_j}{a_j + b_j}\right\}.$$

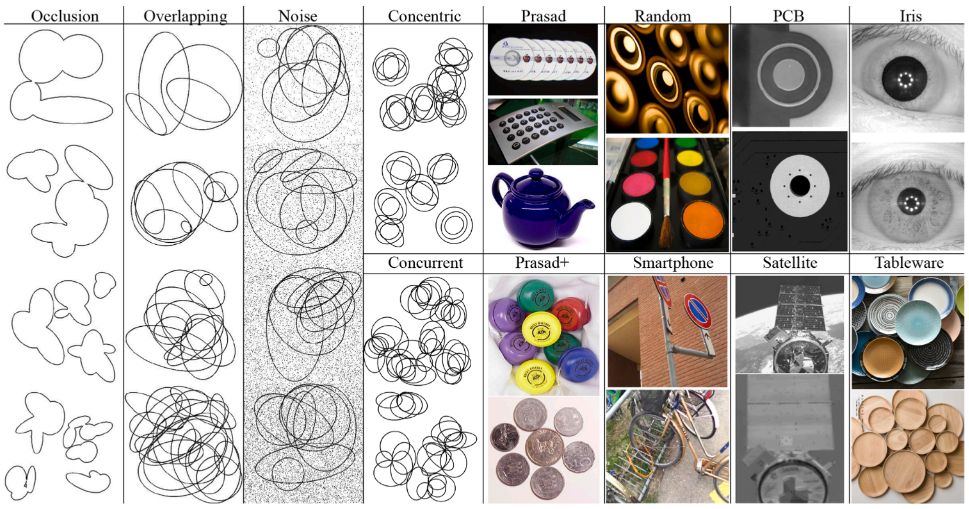

Fig. 9. Example images in the test datasets. Column 1-3 show the synthetic ellipses with occlusion, overlapping, and noise, respectively. Column 4 includes synthetically concentric and concurrent ellipses. Column 5-7 are the images from datasets Prasad, Random and Smartphone, PCB and Satellite, respectively. The last column contains images from our new datasets named Iris and Tableware.

Note that this weighting scheme effectively eliminates the angle influence caused by the rotation symmetry of circles.

We summarize the above steps in Algorithm 1 for easier understanding.

4. Experimental results

In this section, the performance of the proposed method is comprehensively evaluated by a series of experiments including (1) parameter discussion, (2) comparison with six representative state-of-the-art methods regarding synthetic and real-world images, (3) robustness against ellipse variations, and (4) robustness against the intersection over union (IoU) variations. All experiments are executed on a desktop computer with Intel Core I7-7700K CPU @4.20 GHz and 32 GB RAM.

4.1. Datasets

We use five synthetic datasets and eight real-world datasets to verify the general capability of the proposed ellipse detector. Fig. 9 illustrates several images from these datasets, which have different characteristics as described in the following. Our code and all datasets will be available at https://github.com/meiyy/EllDet.

Synthetic datasets. Synthetic ellipses involving occlusion, overlapping, noise, concentric, and concurrent are tested. There are 300 images with occluded ellipses and 300 images with overlapping ones [1], with the resolution of 300 $\times$ 300. Each image has $\beta \in \{4, 8, 12, 16, 20, 24\}$ ellipses under the constraint that they must overlap with at least one ellipse. The complex occlusion or overlapping, especially with the number of ellipses increasing, make the detection tough enough. To test the robustness of the ellipse detector, we use the function *imnoise(img, 'salt & pepper', density)* in Matlab with density ranging from 4% to 24% at the step 4% to add salt-and-pepper noise in the images with 8 overlapping ellipses. Besides, we further test 720 images with concentric ellipses and 1,200 images with concurrent ones [37] under the resolution 600 $\times$ 600. These images are challenging enough because of the multiple cracked arcs for grouping.

Real-world datasets. Dataset Prasad et al. [1] has 400 images sampled from 48 categories in Caltech256 dataset [42]. However, there are only 198 images available online, and we complement the missing part named Dataset Prasad+ according to the file provided by the authors. The varying image size with cluttered background is the major challenge. Dataset Random [33] also contains 400 images up to 1,280 $\times$ 960 from MIRFlickr and LabelMe repositories [43,44]. The high resolution and noisy interference dramatically degrade the detection speed and effectiveness. Dataset Smartphone [33] has 629 images collected from a video. The existence of image blur and perspective transformation is the main difficulty. Dataset PCB [45] has 100 industrial printed circuit board images. The concentric structure and substantial white noise adversely impact the detection performance. The satellite dataset [37] contains 757 optical images and 440 infrared images, which are captured by the OEDMS and NextSat spacecraft infrared cameras, respectively. The space light, camera noise, and the far small ellipses are hard to detect. Furthermore, we provide two new datasets named Iris and Tableware containing 100 images, respectively. Dataset Iris is used to test the detection capability for small ellipses, which are selected from CASIA Iris Database [46], while Tableware aims to simulate the robotic manipulation of elliptical objects. All ground truth images are labeled by ourselves manually and precisely.

4.2. Evaluation metrics

To quantitatively evaluate the performance of the proposed method, three well-known metrics from information retrieval are utilized, *i.e., precision, recall*, and *F-measure*, which are defined as

$$\text{Precision} = \frac{|\text{TP}|}{|\text{TP} + \text{FP}|}, \quad \text{Recall} = \frac{|\text{TP}|}{|\text{TP} + \text{FN}|}.$$

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

Here, TP, FP, and FN represent the true positives, false positives, and false negatives, respectively. A detected ellipse e_d is considered to be a true positive if its *intersection over union* (IoU) regarding the ground truth e_t is no less than γ ($\gamma = 0.95$ for synthetic images and 0.8 for real images, as suggested in [33]). Otherwise, it is a false positive, and a ground truth not rightly recognized is seen as a false negative. Note that F-measure is a comprehensive performance metric. IoU is defined as

$$\text{IoU}(e_d, e_t) = \frac{area(e_d) \cap area(e_t)}{area(e_d) \cup area(e_t)},$$

where $area(e_*)$ denotes the number of pixels inside the ellipse e_*. The proposed ellipse detector is compared with six representative state-of-

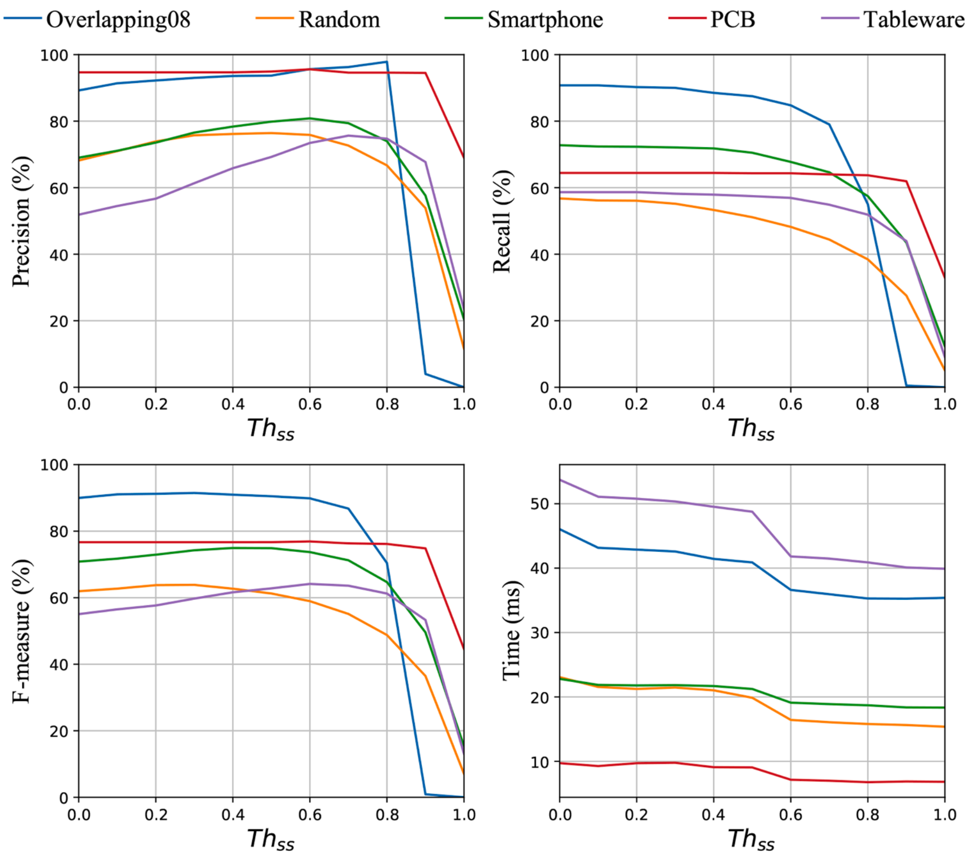

Fig. 10. Investigation of the salient score parameter Th_{ss} on five datasets listed on top. A better choice of Th_{ss} falls in $[0.5, 0.7]$, considering the F-measure and time consumption.

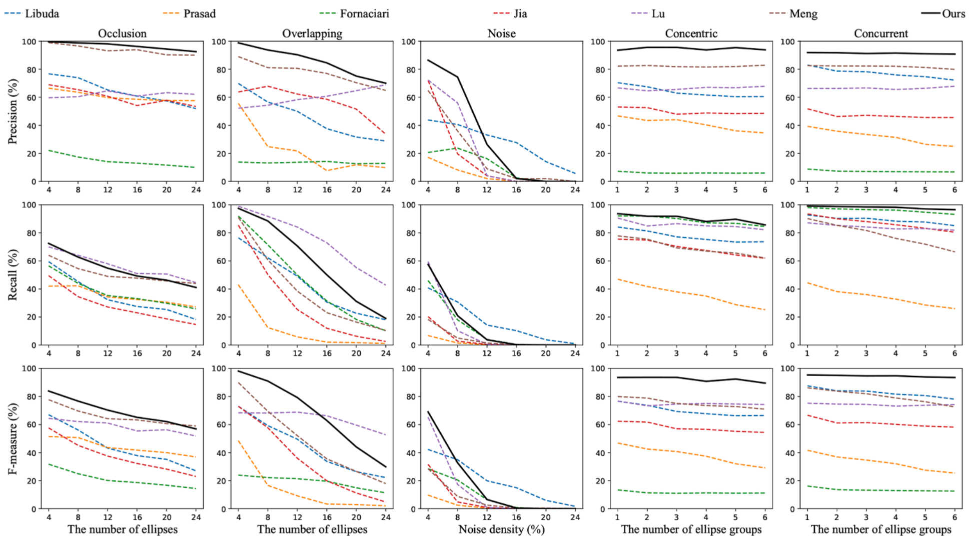

Fig. 11. Ellipse detection results on synthetic datasets. Our method achieves the overall highest F-measure with superior precision.

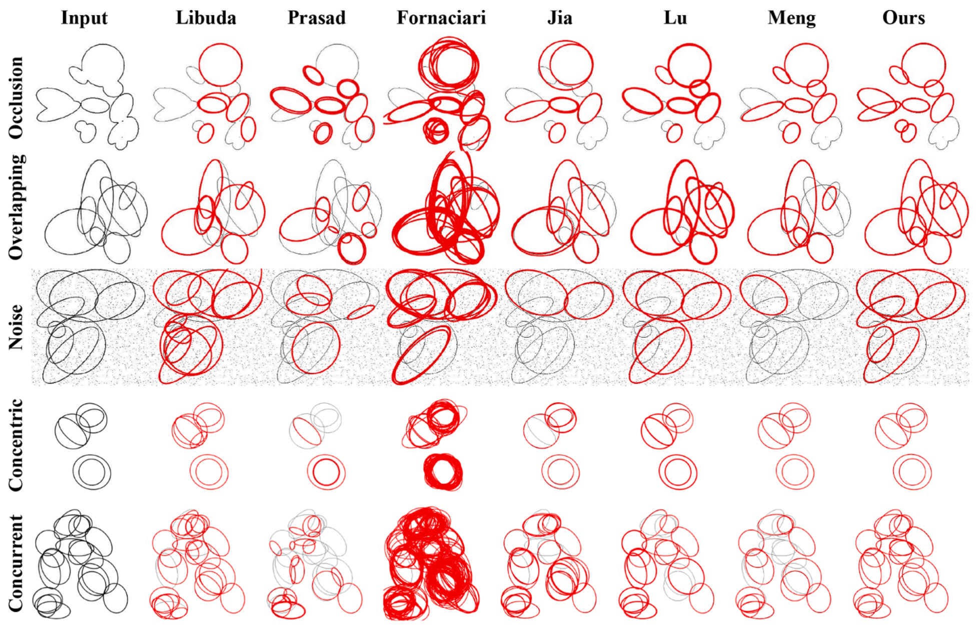

Fig. 12. Ellipse detection examples on synthetic images with occlusion, overlapping, noise, concentric, and concurrent. Our method detects most of the true positives while making less false positives.

the-art methods including Libuda [30], Prasad[1] [1], Fornaciari [33], Jia [34], Lu [18], and Meng [37]. The source code of these methods are publicly available online, and Prasad and Lu are implemented in Matlab, while the others and ours are in C++.

4.3. Parameter discussion

Our method mainly involves six parameters, which are discussed in the following. (1) The angle threshold Th_θ is used to discover sharp corners, and larger value will tolerate curves with larger curvature. Based on the elliptic curvature, we fix $Th_\theta = 46^\circ$ as it performs well for general images. (2) Th_r is the aspect ratio of bounding box to remove straight segments, and we can speed up the detection process by setting relatively small ones. However, more arcs will also be deleted. Extensive experiments suggest that $Th_r = 10$ is a better balance between the effectiveness and efficiency. (3) Inlier ratio threshold Th_{ir} is used to attain high-quality arcs. Admittedly, larger threshold will keep better arcs, but considering the discrete pixels, we choose $Th_{ir} = 0.7$ for use. (4) In the arc grouping step, Th_{lr} is the length ratio tolerance of two arcs. Because too short arcs hardly provide rich information, we let Th_{lr} be equal to 6 to find similar arc pairs. (5) Th_d is used to evaluate the distance between two arcs, due to the image size limit, big values less likely emerge, we set $Th_d = 10$ to incorporate as many arc pairs as possible. Since the fine and stable performance of these parameters for hundreds of images, we fixing them as intrinsic ones without user tuning. (6) The last parameter Th_{ss} in the validation step is used to select salient ellipses. We open it as an adjustable parameter according to the practical requirement. Additionally, to reveal the performance variation regarding different Th_{ss}, we test five datasets as illustrated in Fig. 10. As observed, with Th_{ss} increasing, precision first goes up and then decreases after $Th_{ss} > 0.8$, while recall starts to reduce when $Th_{ss} > 0.5$ and decreases significantly when $Th_{ss} > 0.7$, thereby strict verification will lower down the metric recall. In practice, we can slightly relax Th_{ss} to let more candidate ellipses be true positives. Taking time consumption into consideration, we suggest $Th_{ss} \in [0.5, 0.7]$ in practice.

[1] The implementation online for Prasad is incomplete. We re-implement the validation part based on the Section 4 in the original paper [1], as faithfully as possible.

4.4. Test on synthetic datasets

We report the detection results of synthetic images including occlusion, overlapping, noise, concentric, and concurrent in Fig. 11. As observed, the proposed detector attains the highest F-measure on datasets occlusion, concentric and concurrent, as well as the highest precision with the value more than 80%, which demonstrates its superior localization accuracy. Methods Lu and Meng share the similar performance and are lower than ours. Besides, Fornaciari has the lowest F-measure and precision on these three datasets, whereas Jia is better than Fornaciari, indicating the effectiveness of the added projective invariant. However, the performance of Jia and Prasad are still unsatisfactory and are lower than Libuda. Except the occlusion case, our method also achieves the highest recall on concentric and concurrent cases. For overlapping ellipses, the proposed detector has the highest F-measure when the number of ellipses is less than 20. With more ellipses, although the F-measure is lower than Lu, we still achieve the second highest one, together with the second highest recall, and Lu embraces the best recall. Nevertheless, we remain the highest precision. Note that as the number of overlapping ellipses increases, the F-measure and recall of Prasad and Jia tend to zero, which indicates that they are subject to complex scenes. For noisy test, our method returns acceptable results when the noise level is no more than 8%. With the noise level increasing, the performance of all methods decreases rapidly, it is because heavy noise breaks continuous arcs as small fragments, which adversely influences the arc grouping process. Therefore, a simple denoising step is helpful. Several detection examples and more noisy images are presented in Fig. 12 and Fig. 13, respectively.

4.5. Test on real-world datasets

Besides synthetic test, we further report the test results on eight real-world datasets. The F-measure and time consumption are given in Table 1 and 2, respectively, where the red and blue colors indicate the two best F-measure. From Table 1, we can see that the proposed method attains the highest F-measure on five datasets and achieves the best detection effectiveness in general. Lu achieves the second best F-measure, but its detection speed is much slower than ours as shown in Table 2. Meng gets the third place along with the fastest speed, which benefits from its optimization operation. Jia also has the relatively small

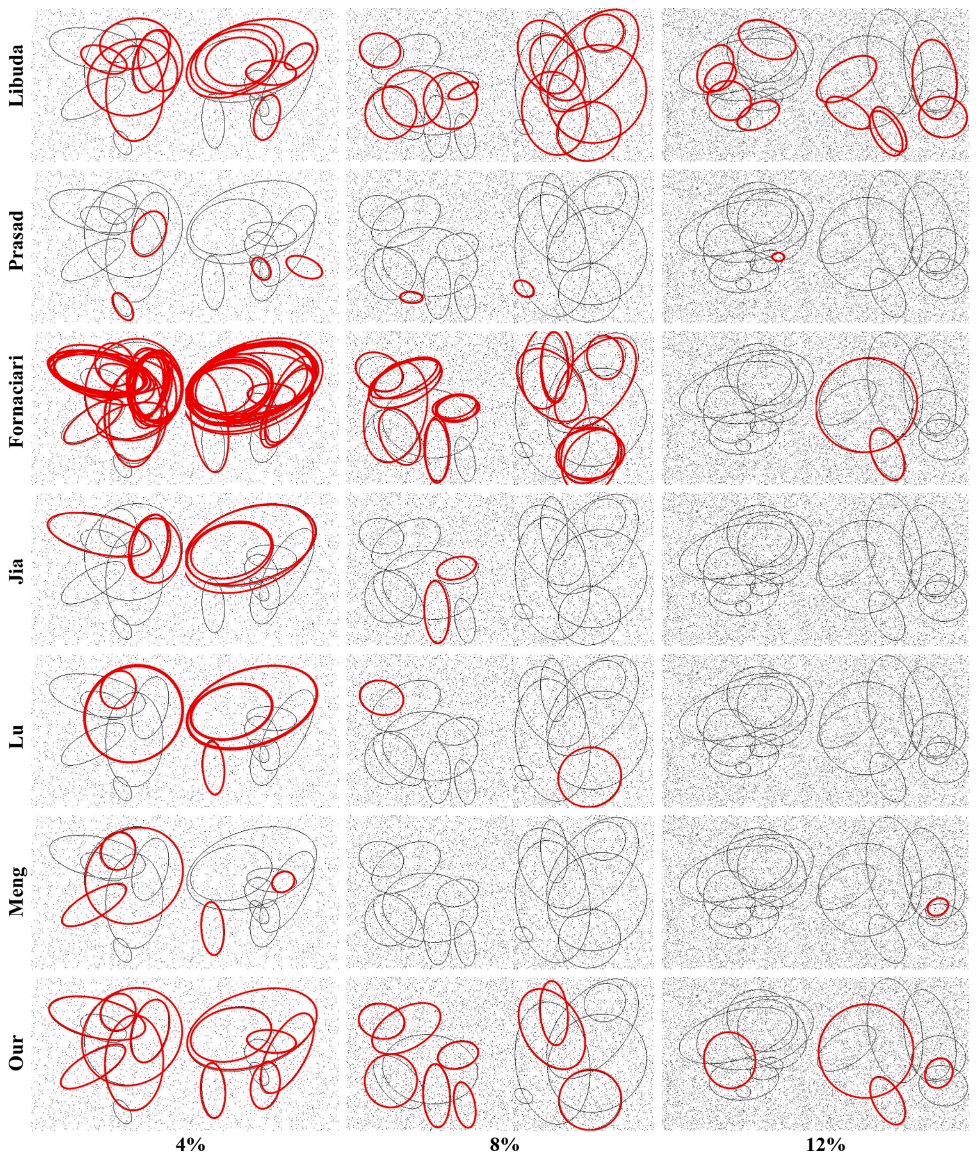

Fig. 13. Ellipse detection examples on noisy images. The proposed method has the most true positives.

Table 1
Comparison on the eight real-world datasets of six different methods in terms of F-measure (%). Red and blue colors indicate the best two performance, respectively. Our method achieves the overall highest F-measure.

Method	Prasad	Prasad+	Random	Smartphone	PCB	Satellite	Iris	Tableware
Libuda	30.82	40.86	37.49	40.09	61.22	31.74	64.81	16.65
Prasad	28.78	21.35	29.1	22.25	56.11	6.81	55.52	33.07
Fornaciari	28.88	31.34	30.62	19.18	55.89	28.79	57.44	15.74
Jia	33.42	48.96	50.15	52.21	74.84	22.21	58.57	54.74
Lu	50.91	65.39	60.02	64.02	80.22	45.03	66.37	54.59
Meng	43.81	54.67	50.05	56.5	70.79	56.65	66.25	53.06
Ours	45.58	66.78	61.12	74.89	79.46	47.76	75.36	62.9

execution time, but the F-measure is a little low. Although Libuda has the fifth highest F-measure on the whole, it performs well on small ellipses, which can be concluded from the dataset Iris. But the time consumption of Libuda is very expensive and is much more than ours. Methods Fornaciari and Prasad share the similar F-measure, but Prasad takes significantly long time, even 100 times than ours, which suffers from the process of complex arc grouping and HT voting. However, the F-measure of both Fornaciari and Prasad are far from satisfactory, especially for complicated images with occlusion or noise, such as the images in datasets Tableware and Satellite. As a whole, the proposed method embraces the highest F-measure with fairly well competitive running time. Several ellipse detection examples are presented in Fig. 16. Note that the execution time of our method suggests that we can work on general camera video with 30Hz rate.

Table 2
Time (ms) comparison on the eight real-world datasets of six different methods. The proposed method can be used for camera video processing of 30 fps.

Method	Prasad	Prasad+	Random	Smartphone	PCB	Satellite	Iris	Tableware
Libuda	12.38	20.36	32.04	52.07	26.94	8.92	14.88	95.11
Prasad	1870.99	5222.84	5153.76	11743	533.97	1074.05	1451.36	16294.7
Fornaciari	3.88	10.9	11.73	16.84	5.08	2.77	2.92	74.93
Jia	3.47	7.18	9.6	12.6	4.87	2.44	2.87	40.02
Lu	78.67	277.92	334.1	618.25	54.53	17.19	27.77	4607.49
Meng	3.19	5.25	8.24	11.55	3.33	2.61	3.01	26.35
Ours	7.94	13.15	16.38	19.16	7.1	4.67	4.89	40.98

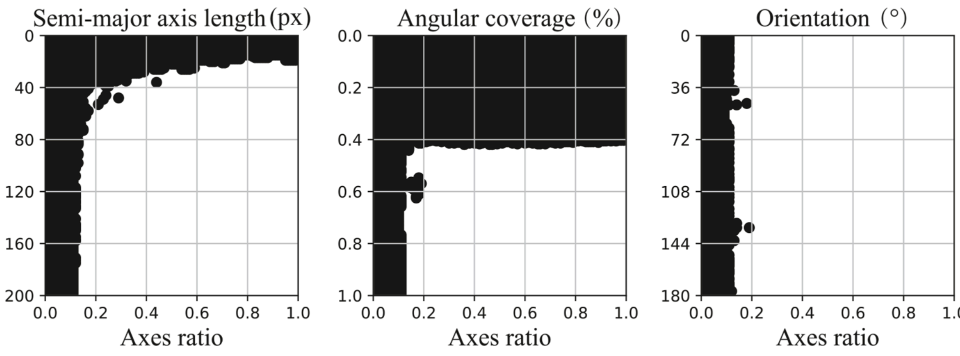

Fig. 14. Robustness test results under different ellipse variations. The horizontal axis indicates the axes ratio of semi-minor axis to semi-major one, ranging from 0.01 to 1 at the step 0.01. The vertical axes are the semi-major axis length in pixel, angular coverage of ellipse arc, and ellipse orientation, respectively. Our method embraces a wide range of successful area indicated by the white region.

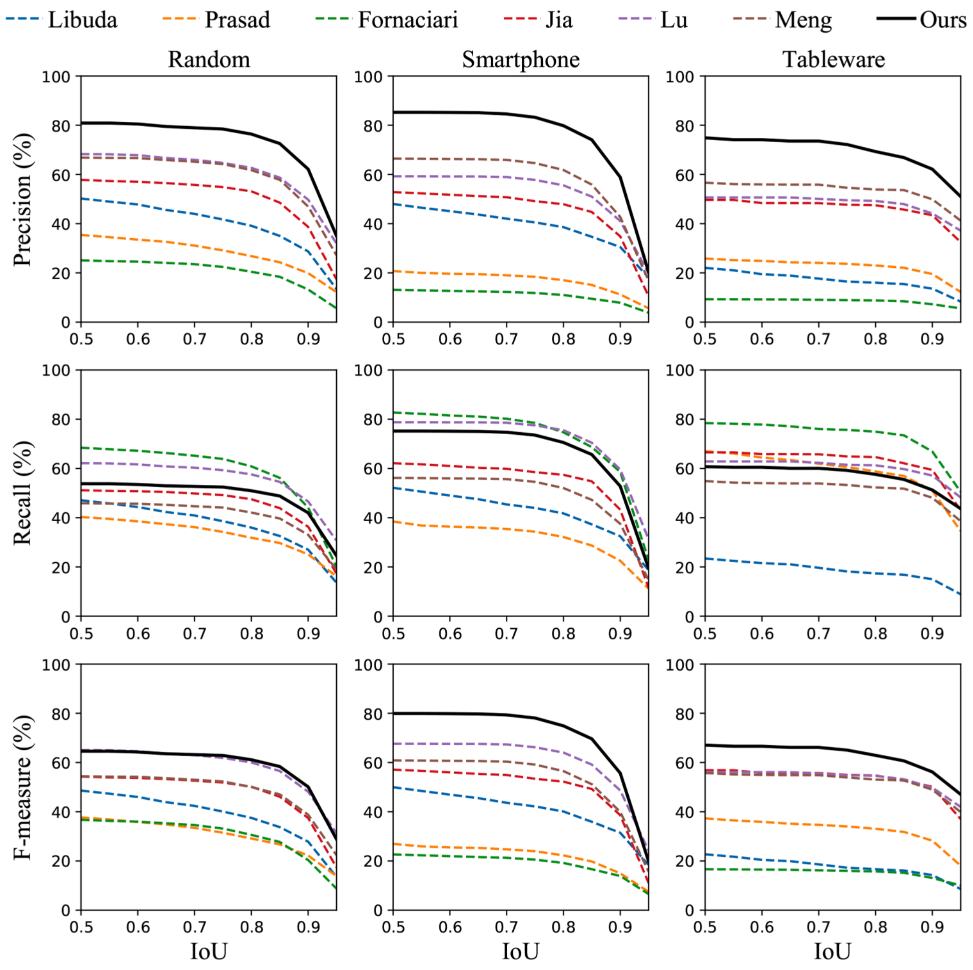

Fig. 15. Robustness test results by varying different IoU values. The proposed method achieves the highest F-measure.

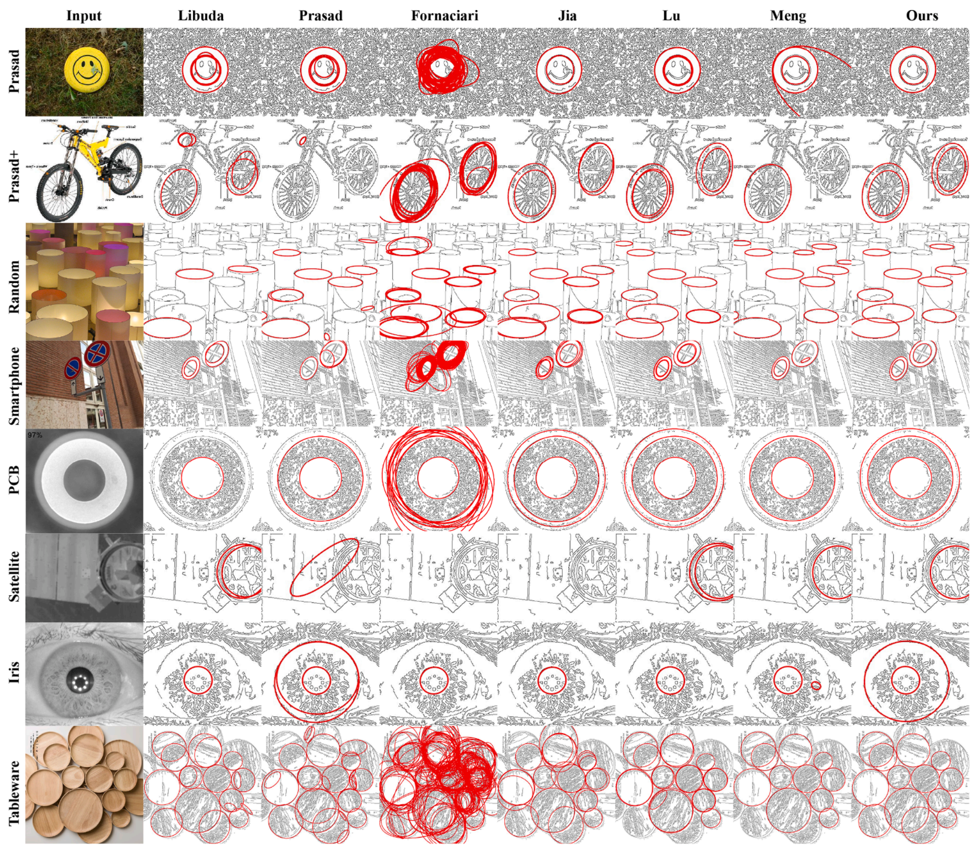

Fig. 16. Sampled ellipse detection results on real-world images. The first column presents the input images from the eight datasets, and detection results of different methods are presented in the second to last columns. The proposed method detects the most true positives without false positives.

4.6. Robustness to ellipse variations

To further investigate the robustness of our method for ellipse variations regarding size, orientation, and incompleteness, we generate three datasets with image size 512 $\times$ 512. The first dataset has 20,000 images with the semi-major axis length varying from 1 to 200 pixels, meanwhile, the axis ratio increases from 0.01 to 1 at the step 0.01. To evaluate the robustness against rotation angles, we build the second dataset by rotating the ellipse from 1° to 180° at the step 1°, fixing the semi-major axis equal to 200 pixels and varying the axis ratio from 0.01 to 1 at the step 0.01, hence there are 18,000 images for test. The last dataset involving 36,000 images aims to check the capacity for incomplete ellipses, where the angular coverage is from 1° to 360° at the step 1° and the axis ratio ranges from 0.01 to 1 at the step 0.01.

The results of ellipse variations are reported in Fig. 14, where the white region indicates the correctly detected ellipses and the black region means the failure cases. From Fig. 14(a), we conclude that our detector has a wide range of successful area and can detect small ellipses with the semi-major axis around 25 pixels and axis ratio slightly below 0.2. Fig. 14(b) shows that our method is able to detect incomplete ellipses with angular coverage about 150°. Furthermore, we can improve the robustness to incomplete ellipses by slightly lowering down the salient score in the validation step. The black region distributes vertically in Fig. 14(c), indicating that our method is very robust to ellipse orientation, which is a basic nature for high-quality ellipse detector.

4.7. Robustness to IoU variations

We also test the robustness of different methods against IoU. To this end, we vary IoU from 0.5 to 0.95 at the step of 0.05 on three datasets. Admittedly, higher IoU brings more stricter constraint of an ellipse being regarded as a true positive. The detection results are reported in Fig. 15. From which, we can see that our method achieves the highest precision on all datasets. Although our recall is slightly lower, we still has the best comprehensive metric F-measure, which demonstrates the high quality performance of our detector. In contrast, Fornaciari attains the highest recall, however, due to the lowest precision, its F-measure is far from satisfactory. With the value of IoU increasing, all methods show descending trend, whereas our method keeps the F-measure higher than 60% when IoU $<=$ 0.8. When IoU $=$ 0.95, although the F-measure of some methods drop below 10% such as Prasad and Fornaciari on dataset Smartphone, we still has the F-measure more than 20%, which indicates the robustness of the proposed detector to IoU variations.

5. Conclusions

In this paper, we have presented a novel ellipse detection method by introducing the convex hull and directed graph, which performs accurately and efficiently for versatile synthetic and real-world images. We have made innovative improvements compared with previous ones. Smooth arcs are extracted by the identification of sharp corners and inflection points based on the immediate computation of inner and cross products. According to the ellipse convexity, we use convex hull to judge the convexity between arc pairs, since merely four cross products are needed, the computation is fast. By incorporating other constraints, a local to global arc grouping strategy is established. The relationship between arc pairs is encoded in a directed graph, by which all arcs from the same ellipse are found to generate candidate ellipses. Moreover, a rigorous verification and weighted clustering further enhance the accuracy by rejecting false positives and repetitive ones.

Extensive experiments on 13 datasets compared with 6 representative state-of-the-art methods demonstrate the superior performance of our method, which also has a good potential for video stream processing. In the future, we plan to apply our detector to more dedicated tasks such as camera calibration and robotic grasping.

Declaration of Competing Interest

There is no conflict of Interest for this submission.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (12022117, 61772523 and 61802406), the Beijing Natural Science Foundation (Z190004), the Beijing Advanced Discipline Fund (115200S001), the Open Research Fund Program of State key Laboratory of Hydroscience and Engineering, Tsinghua University (sklhse-2020-D-07), and Alibaba Group through Alibaba Innovative Research Program.

References

[1] D.K. Prasad, M.K. Leung, S.-Y. Cho, Edge curvature and convexity based ellipse detection method, Pattern Recognit. 45 (9) (2012) 3204–3221.
[2] J. Heikkila, Geometric camera calibration using circular control points, IEEE Trans. Pattern Anal. Mach. Intell. 22 (10) (2000) 1066–1077.
[3] H. Huang, H. Zhang, Y.-m. Cheung, The common self-polar triangle of concentric circles and its application to camera calibration. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4065–4072.
[4] Z.-Y. Liu, H. Qiao, Multiple ellipses detection in noisy environments: a hierarchical approach, Pattern Recognit. 42 (11) (2009) 2421–2433.
[5] 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.
[6] C.-C. Chiang, M.-C. Ho, H.-S. Liao, A. Pratama, W.-C. Syu, Detecting and recognizing traffic lights by genetic approximate ellipse detection and spatial texture layouts, Int. J. Innov. Comput. Inf. Control 7 (12) (2011) 6919–6934.
[7] X. Bai, C. Sun, F. Zhou, Splitting touching cells based on concave points and ellipse fitting, Pattern Recognit. 42 (11) (2009) 2434–2446.
[8] L. Świrski, A. Bulling, N. Dodgson, Robust real-time pupil tracking in highly off-axis images. Proceedings of the Symposium on Eye Tracking Research and Applications, 2012, pp. 173–176.
[9] H. Dong, E. Asadi, G. Sun, D.K. Prasad, I.-M. Chen, Real-time robotic manipulation of cylindrical objects in dynamic scenarios through elliptic shape primitives, IEEE Trans. Robot. 35 (1) (2018) 95–113.
[10] R.O. Duda, P.E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Commun. ACM 15 (1) (1972) 11–15.
[11] S. Tsuji, F. Matsumoto, Detection of ellipses by a modified Hough transformation, IEEE Trans. Comput. (1978) 777–781.
[12] D.H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognit. 13 (2) (1981) 111–122.
[13] R.A. McLaughlin, Randomized Hough transform: improved ellipse detection with comparison, Pattern Recognit. Lett. 19 (3-4) (1998) 299–305.
[14] C.A. Basca, M. Talos, R. Brad, Randomized Hough transform for ellipse detection with result clustering. Proceedings of the EUROCON 2005-The International Conference on" Computer as a Tool" 2, IEEE, 2005, pp. 1397–1400.
[15] Y. Chen, Y. Yang, Two improved algorithms for ellipse detection based on Hough transform, Semicond. Optoelectron. 38 (5) (2017) 745–750.
[16] Z. Zhang, Parameter estimation techniques: a tutorial with application to conic fitting, Image Vis. Comput. 15 (1) (1997) 59–76.
[17] P. Mukhopadhyay, B.B. Chaudhuri, A survey of Hough transform, Pattern Recognit. 48 (3) (2015) 993–1010.
[18] C. Lu, S. Xia, M. Shao, Y. Fu, Arc-support line segments revisited: an efficient high-quality ellipse detection, IEEE Trans. Image Process. 29 (2019) 768–781.
[19] K. Kanatani, Y. Sugaya, Y. Kanazawa, Ellipse fitting for computer vision: implementation and applications, Synth. Lect. Comput. Vis. 6 (1) (2016) 1–141.
[20] P.V. Hough, Method and means for recognizing complex patterns, 1962, US Patent 3,069,654.
[21] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (rht), Pattern Recognit. Lett. 11 (5) (1990) 331–338.
[22] N. Kiryati, Y. Eldar, A.M. Bruckstein, A probabilistic Hough transform, Pattern Recognit. 24 (4) (1991) 303–316.
[23] W. Lu, J. Tan, R. Floyd, Automated fetal head detection and measurement in ultrasound images by iterative randomized Hough transform, Ultrasound Med. Biol. 31 (7) (2005) 929–936.
[24] Y. Xie, Q. Ji, A new efficient ellipse detection method. Object Recognition Supported by User Interaction for Service Robots 2, IEEE, 2002, pp. 957–960.
[25] A.Y.S. Chia, M.K.H. Leung, H. Eng, S. Rahardja, Ellipse detection with hough transform in one dimensional parametric space. Proceedings of the IEEE International Conference on Image Processing 5, 2007.
[26] C.-T. Ho, L.-H. Chen, A fast ellipse/circle detector using geometric symmetry, Pattern Recognit. 28 (1) (1995) 117–124.
[27] Y. Lei, K.C. Wong, Ellipse detection based on symmetry, Pattern Recognit. Lett. 20 (1) (1999) 41–47.
[28] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (6) (1986) 679–698.
[29] E. Kim, M. Haseyama, H. Kitajima, Fast and robust ellipse extraction from complicated images. Proceedings of the IEEE Information Technology and Applications, Citeseer, 2002, pp. 1–6.
[30] L. Libuda, I. Grothues, K.-F. Kraiss, Ellipse detection in digital image data using geometric features. Advances in Computer Graphics and Computer Vision, Springer, 2007, pp. 229–239.
[31] F. Mai, Y. Hung, H. Zhong, W. Sze, A hierarchical approach for fast and robust ellipse extraction, Pattern Recognit. 41 (8) (2008) 2512–2524.
[32] 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.s 20 (7) (2010) 1991–2006.
[33] M. Fornaciari, A. Prati, R. Cucchiara, A fast and effective ellipse detector for embedded vision applications, Pattern Recognit. 47 (11) (2014) 3693–3708.
[34] Q. Jia, X. Fan, Z. Luo, L. Song, T. Qiu, A fast ellipse detector using projective invariant pruning, IEEE Trans. Image Process. 26 (8) (2017) 3665–3679.
[35] H. Dong, D.K. Prasad, I.-M. Chen, Accurate detection of ellipses with false detection control at video rates using a gradient analysis, Pattern Recognit. 81 (2018) 112–130.
[36] V. Pătrăucean, P. Gurdjos, R.G. von Gioi, Jointa contrarioellipse and line detection, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) (2016) 788–802.
[37] C. Meng, Z. Li, X. Bai, F. Zhou, Arc adjacency matrix-based fast ellipse detection, IEEE Trans. Image Process. 29 (2020) 4406–4420.
[38] M. Zhao, X. Jia, D.-M. Yan, An occlusion-resistant circle detector using inscribed triangles, Pattern Recognit. 109 (2020) 107588.
[39] D.K. Prasad, C. Quek, M.K. Leung, S.-Y. Cho, A parameter independent line fitting method. Proceedings of the First Asian Conference on Pattern Recognition, IEEE, 2011, pp. 441–445.
[40] U. Ramer, An iterative procedure for the polygonal approximation of plane curves, Comput Graph. Image Process. 1 (3) (1972) 244–256.
[41] A. Fitzgibbon, M. Pilu, R.B. Fisher, Direct least square fitting of ellipses, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 476–480.
[42] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, 2007.CalTech Report
[43] M.J. Huiskes, M.S. Lew, The Mir Flickr retrieval evaluation. Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, 2008, pp. 39–43.
[44] B. Russell, Labeme:a database and web-based tool for image annotation, IJCV 77 (2008).
[45] C. Lu, S. Xia, W. Huang, M. Shao, Y. Fu, Circle detection by arc-support line segments. Proceedings of the IEEE International Conference on Image Processing (ICIP), IEEE, 2017, pp. 76–80.
[46] CASIA iris image database, http://biometrics.idealtest.org/Accessed May 2020.
[47] Yifan Lu, Jiaming Lu, Songhai Zhang, Peter Hall, Traffic signal detection and classification in street views using an attention model, Computational Visual Media 4 (2018) 253–266, https://doi.org/10.1007/s41095-018-0116-x. In this issue.