EDSF: FAST AND ACCURATE ELLIPSE DETECTION VIA DISJOINT-SET FOREST

Jingen Jiang[*1,2], *Mingyang Zhao*[*1,3], *Zeyu Shen*[4], *Dong-Ming Yan*[1,2,5]
[1]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Beijing Academy of Artificial Intelligence, Beijing, China
[4]State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
[5]State Key Laboratory of Hydro-Science and Engineering, Tsinghua University, Beijing, China

ABSTRACT

We present a novel yet effective method for detecting elliptical primitives in cluttered, occluded images, which has versatile applications in computer vision and multimedia processing fields. We begin by the fast extraction of smooth arcs from the edge map, followed by the construction of a *directed graph* and a *disjoint-set forest*, whereby the arc relationships are effectively encoded to enhance the arc grouping process. Compared with representative approaches such as the depth-first search, the disjoint-set forest enables complete grouping of arcs to generate candidate ellipses. Moreover, it merely has *linear memory complexity* and *constant access time*, hence guarantees fast detection. To boost precision and remove false positives, we propose to project the candidate ellipses onto the original image, to align the gradients of ellipses and the image pixels. We also vectorize the elliptical parameters to depress duplicated candidates. We perform extensive experiments on both synthetic and challenging real-world datasets, to show that our detector is accurate and efficient, as well as versatile in many practical tasks. The source code and datasets are available at https://github.com/xiaowuga/EDSF.

Index Terms— Ellipse detection, disjoint-set forest, geometric primitives, geometric camera calibration

1. INTRODUCTION

The detection of elliptical primitives from images has various applications in a wide variety of computer vision and multimedia processing tasks, such as traffic sign location [1], eye or head tracking [2, 3], industrial inspection [4], robotic grasping [5, 6], camera calibration [7], and so on. However, designing a fast and reliable general-purpose ellipse detector is still a challenging mission [8]. The crucial aspects comprise the noise contamination, sophisticated backgrounds, varying illuminations and concentric structure. For instance, multiple concentric ellipses typically incur erroneous arc grouping, resulting in *false negatives* or *duplicated detections* (Fig. 1).

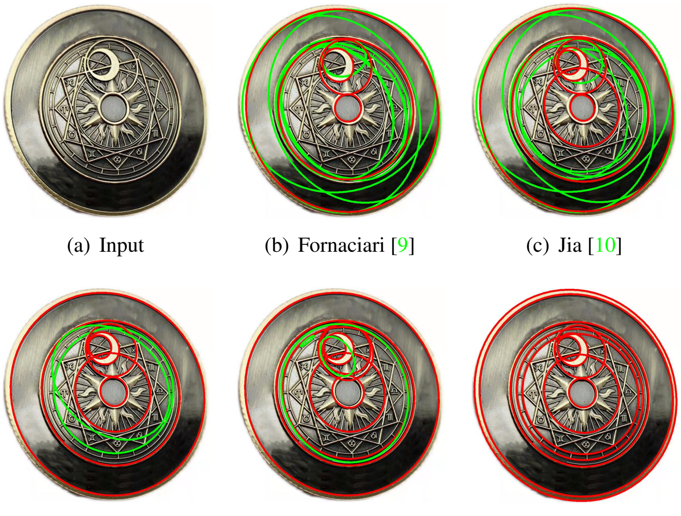

(a) Input
(b) Fornaciari [9]
(c) Jia [10]
(d) Meng [11]
(e) Shen [12]
(f) Ours

Fig. 1. Comparison of state-of-the-art approaches and ours on images with multiple concentric ellipses. Green ellipses indicate false detections due to incorrect or insufficient arc grouping, whereas our method successfully detects all ellipses.

The most classical geometric primitive detector is the *Hough Transform* (HT) [13], which transforms the detection task as a peak finding process in a parametric voting space. Nevertheless, due to the five-dimensional elliptical parameters (x_c, y_c, a, b, θ), where (x_c, y_c) is the center position, (a, b) are the semi-axis lengths, and θ is the rotation angle along the $x-$axis, HT typically has long computing time and large memory consumption. Moreover, it suffers from images with complicated backgrounds and concentric structure [14]. Later, many variants of HT are proposed to either speed up the detection or decrease the memory usage, such as the *randomized* and *probabilistic HT* [15, 16] and methods that combine HT with the elliptical geometric properties [17]. Despite improvements have been attained, these methods are still less efficient, particularly for complicated images. Additionally, they are prone to incur *false positives* due to the lack of effective verification and the inappropriate tuning of the bin size.

Latterly, *edge chaining* methods are given more prominence and have been proved quite efficient for ellipse detection [9, 18, 10, 19, 11]. These approaches utilize *continuous*

[*] The authors contribute equally to this paper.

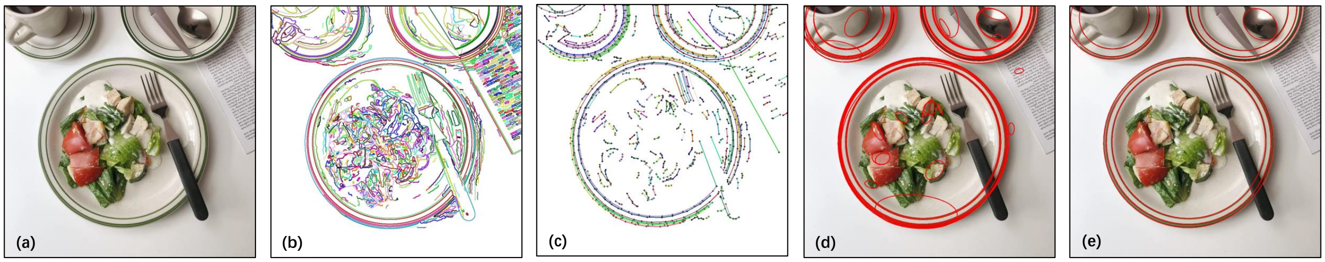


Fig. 2. **Overview of the proposed method**. (a) Input image. (b) Edge detection using an adaptive edge detector. (c) Arc extraction by splitting at sharp corners and inflection points. (d) Candidate ellipse generation by the disjoint-set forest. (e) Finally detected ellipses after gradient validation and clustering. It has fast detection benefit from the parallel implementation.

arcs rather than discrete pixels like HT to generate candidate ellipses. For instance, Chia *et al.* [20] formalize arc grouping as a pairwise matching task and attain optimal ellipses based on a feedback iteration. Prasad *et al.* [21] design a searching region based on ellipse curvature to combine arcs together, however, due to the incorporation of 2D HT, the method is computationally expensive. Fornaciari *et al.* [9] and Jia *et al.* [10] adopt the similar paradigm to split ellipses into four quadrants by gradient computation, and re-group arcs through differential such as tangents. Although they are efficient, the number of grouping arcs must be at least three, hence these methods are susceptible to occluded or semi ellipses. Recently, Meng *et al.* [11] propose an adjacency-matrix-based arc grouping method by curvatures, while Shen *et al.* [12] leverage convex hull to generate ellipses. Compared with previous approaches, these two methods are more accurate and efficient. Nevertheless, since the brute-force *depth-first search strategy* is invoked in both methods, they are prone to consume much memory or losing ellipses with concentric configuration.

In this work, our goal is to propose a fast and high-precision detector allowing to reliably recognize elliptical primitives, regardless of the image size and content, which has a spectrum of applications in practice. Concretely, based on the *convexity* and *smoothness* of ellipses, we use vector computations to efficiently recognize sharp corners and inflection points to attain smooth arcs, then we design four *geometric constraints* to judge arcs that if they are potentially from the same ellipse, followed by the modeling and encoding of arc relationships into a *disjoint-set forest* (DSF), by which all arc grouping cases are enumerated. DSF only has a linear memory complexity $O(n)$ (n is the number of arc elements), and is one of the most efficient data structure that can be accessed with nearly constant time complexity [22], hence paves the way for fast detection. To depress false positives and duplicated candidates, we propose a *gradient projection* method to discriminate the ellipse geometry with the pixel gradients of the original input image. From such operation, we can reliably determine inliers on ellipses, thereby enhance the detection accuracy. We perform extensive experiments to assess the proposed ellipse detector and compare it with representative state-of-the-art approaches in both synthetic and real-world images. Results evidence that our detector is fast and outperforms competitors with higher *F-measure*, as well as being more robust against occlusion.

Our main contributions are: (1) We propose effective geometric constrains with disjoint-set forest for fast and accurate ellipse detection, capable of handling complicated real-world images. (2) We introduce the disjoint-set forest to encode arc relationships for fast access, and ensure complete arc grouping without omissions. (3) Our approach achieves higher F-measure and is faster than most competitors on both synthetic and real-world benchmark datasets.

2. METHODOLOGY

Our proposed method consists of three main steps, as shown in Fig. 2: (1) adaptive edge detection and elliptical arc extraction; (2) candidate ellipse generation by construction of the directed graph and disjoint-set forest; (3) false positives and duplicated candidate depression via gradient validation and clustering. We present each step in the following.

2.1. Adaptive edge detection and arc extraction

Given an input image (Fig. 2(a)), the adaptive edge extraction method [23] is adopted to detect edge segments due to its efficiency and the avoidance of parameter tuning (Fig. 2(b)). Then we use fast vector computations to extract smooth arcs. First, the *Ramer-Douglas-Peuker* (RDP) algorithm [24] is applied to approximate edges via a series of line segments $\{l_i = P_{i-1}P_i | P_i \in \mathbb{R}^2\}_{i=1}^{n}$, where P_* is referred as the *dominant point*, and the *direction vector* of l_i is $\overrightarrow{l}_i$. We use *parallel computation* to speed up the RDP process. According to the smoothness and convexity of ellipses, we split edge curves at sharp corners (where curvatures vary significantly) and inflection points (indicating the change of convexity and concavity) to attain smooth arcs, in which the inner and cross products of direction vectors are used. As illustrated in the left of Fig. 3, an angle α_i between l_i and l_{i+1} indicates a sharp corner if

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

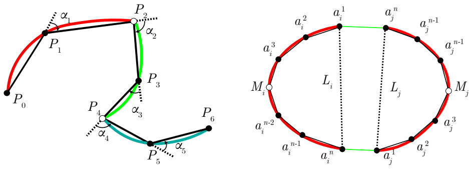


Fig. 3. Left: Recognition of sharp corners (α_2) and inflection points (P_4). Right: Arc grouping via the position and the continuity constraints, where a_*^k, M_* and L_* denote the k^{th} dominant point, midpoint and chord equation of the arc a_*.

where T_α is the angle threshold. Inflection points are identified by means of the cross product, *i.e.*, if

$$(\frac{\overrightarrow{l}_{i-1}}{\|\overrightarrow{l}_{i-1}\|_2} \times \frac{\overrightarrow{l}_{i}}{\|\overrightarrow{l}_{i}\|_2}) \cdot (\frac{\overrightarrow{l}_{i}}{\|\overrightarrow{l}_{i}\|_2} \times \frac{\overrightarrow{l}_{i+1}}{\|\overrightarrow{l}_{i+1}\|_2}) = -1, \qquad (2)$$

then the start point of $\overrightarrow{l}_i$ denotes an inflection point. Edges are split at these sharp corners and inflection points to attain smooth elliptical arcs $\{a_j \in \mathbb{R}^2\}_{j=1}^m$ (indicated by different colors in the left of Fig. 3).

2.2. Candidate ellipse generation

We use nodes of a directed graph to represent arcs, and the directed edges are pointed from the longer arc to the shorter one. Two nodes are connected by a directed edge if they satisfy the following geometric constraints, meaning that they may come from the same ellipse.

Position constraint. As shown in the right of Fig. 3, let the chord equation of a_i be $L_i : A_i x + B_i y + C_i = 0$ (A_i, B_i, C_i are coefficients), and a_i, a_j are said satisfying the position constraint if

$$L_i(M_i) \cdot L_i(M_j) < 0 \wedge L_j(M_i) \cdot L_j(M_j) < 0, \qquad (3)$$

where M_i is the midpoint of a_i. This constraint indeed reflects the convexity of two elliptical arcs in the two half planes.

Arc length constraint. If the length of a_i and a_j satisfies

$$1/T_{lr} < \|a_i\|_2/\|a_j\|_2 < T_{lr}, \qquad (4)$$

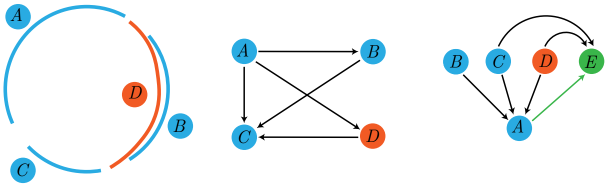
then they will be checked by the subsequent constraints. When the length difference of two arcs are too large, the long arc will play a dominant role among the fitting, which is almost the same as the long arc fitting alone, hence such a directed edge has little effect on the detection result.

Distance constraint. Due to the limited object shape in images, two arcs with large distance are less likely from the same ellipse. If a_i and a_j fail to satisfy

$$\|M_i M_j\|_2 < T_d \cdot (\|a_i\|_2 + \|a_j\|_2), \qquad (5)$$

then we remove them to accelerate detection.

Blending constraint. After satisfying the above constraints, we further consider the blending constraint. As shown in the right of Fig. 3, two arcs a_i and a_j can be blended together if there is no inflection point among the segments generated by $\{a_i^2, a_i^1, a_j^n, a_j^{n-1}\}$ and $\{a_i^{n-1}, a_i^n, a_j^1, a_j^2\}$. This constraint can reduce the disadvantage of concentric arcs for grouping.



(a) Arc map (b) Directed graph (c) Disjoint-set forest

Fig. 4. **Candidate ellipse generation**. Each node of the directed graph (b) stands for an arc in the arc map (a). (c) is the disjoint-set forest of the left directed graph, in which green and black arrows indicate the brotherhood and the parent-child relationship of two nodes, respectively.

Arc grouping by the disjoint-set forest. After the construction of directed graph (Fig. 4(b)), since there will be several candidate ellipses sharing with the same a_i such as the concentric cases, we adopt the disjoint-set forest for arc grouping. As shown in Fig. 4(c), we copy the node A to be the root of a tree and the brother of A is denoted as E. Different from the primary disjoint-set forest, we allow a node has several father nodes, such as node C has two fathers A and E, which means that C belongs to two candidate ellipses at the same time. Besides, two nodes connected by a directed edge may already belong to an existent ellipse such as node B and C in Fig. 4(c), thereby we can skip them to accelerate the arc grouping process. We allow for an arc to be a node of an existent candidate ellipse $\mathcal{CE}$ if its integrity increases. The integrity I of an ellipse is defined as

$$I = \frac{\sum_{p \in arc} \mathbb{1}(dist(p, \mathcal{CE}) < \epsilon)}{L_{\mathcal{CE}}}, \qquad (6)$$

where $\mathbb{1}$ is an indicator function and equates to one if and only if the distance from the arc pixel p to $\mathcal{CE}$ is less than ϵ, and $L_{\mathcal{CE}}$ is the approximation circumference of ellipse calculated by the *Ramanujun's formula*.To further improve detection precision and depress duplicate ellipses, we perform the subsequent validation and clustering.

2.3. Gradient projection and feature clustering

If the detected ellipse is correct, then it will match well with the original ellipse in the image. To utilize this prior feature, we project $\mathcal{CE} = (x_c, y_c, a, b, \theta)$ to the image by uniformly sampling n points from the elliptical parameter equation:

$$\begin{cases} x_i = a \cdot \cos\theta \cdot \cos\alpha + b \cdot \sin\theta \cdot sin\alpha + x_c \\ y_i = -a \cdot \cos\theta \cdot \sin\alpha + b \cdot \sin\theta \cdot cos\alpha + y_c \end{cases} \qquad (7)$$

where $\alpha \in [0, 2\pi)$. By means of the elliptical gradient $(\frac{\partial f}{\partial x_i}, \frac{\partial f}{\partial y_i})^T = (2Ax_i + 2By_i + 2D, 2Bx_i + 2Cy_i + 2E)^T$ (deduced in the supplemental material), we compute the *match score* S of the candidate ellipse $\mathcal{CE}$ as

$$S(\mathcal{CE}) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(v_d \cdot v_g \geq \cos T_{grad}), \qquad (8)$$

where v_d denotes the normalized vector $(\frac{\partial f}{\partial x_i}, \frac{\partial f}{\partial y_i})^T$, $v_g = (g_x, g_y)^T$ is the gradient of pixel (x_i, y_i) calculated by the *Sobel operator* and T_{grad} is an angle threshold. Candidate ellipses with matching score less than a preset tolerance will be discarded. We further vectorize the five elliptical parameters and calculate the Euclidean distance to cluster similar ellipses or depress duplicated ones. If the distance

$$Dist(\mathcal{CE}_i, \mathcal{CE}_j) = \sqrt{\sum_{\lambda=1}^{5} \|\mathcal{CE}_{i\lambda} - \mathcal{CE}_{j\lambda}\|_2} \tag{9}$$

between two candidate ellipses $\mathcal{CE}_i$ and $\mathcal{CE}_j$ is small enough, then we generate a directed edge pointing from the node with higher matching score (Eq. 8) to the one with lower score. Thereby we finally keep the candidate ellipses corresponding to the node with zero *in-degree* in the directed graph, which stands for the salient detections.

3. EXPERIMENTS AND DISCUSSIONS

Implementation details. To assess the effectiveness and efficiency of the proposed method, we perform a series of experiments and compare our method with representative state-of-the-art approaches including Fornaciari *et al.* [9], Jia *et al.* [10], Meng *et al.* [11] and Shen *et al.* [12]. Our algorithm is implemented in C++ with the support library OpenCV 4.5.3. The source codes of the others are available online also implemented in C++. We use the default or fine-tuning hyper-parameters of each method to achieve their best performance, and their parameter setting details are reported in the supplemental material. All experiments are conducted in Intel(R) Core i7-7700K CPU@4.2GHz with 4 cores and 48GB RAM.

Evaluation metrics. We adopt *precision, recall* and *F-measure* to quantitatively evaluate the experimental results, which are defined as Precision $= |TP|/|DE|$, Recall $= |TP|/|GT|$, and F-measure $= 2 \times$ Precision $\times$ Recall/(Precision + Recall), where $|TP|$, $|DE|$ and $|GT|$ stand for the number of true positives, detected and ground truth ellipses, respectively. An ellipse $\mathcal{E}_d$ is to be a true positive if the *Intersection over Union* (IoU) between it and a ground truth ellipse $\mathcal{E}_t$ is larger than 0.8. We use $\frac{area(\mathcal{E}_d) \cap area(\mathcal{E}_t)}{area(\mathcal{E}_d) \cup area(\mathcal{E}_t)}$ to compute the IoU, and $area(\mathcal{E}_*)$ is the number of pixels inside the ellipse $\mathcal{E}_*$.

Occlusion test. Incomplete ellipses are quite common in practical scenarios, which is also one of the challenges in ellipse detection. To demonstrate the robustness of the proposed method against occlusion, we adopt a synthetic dataset [21] comprising of six sub-datasets for test, and each sub-dataset has 100 images with $\{4, 8, 12, 16, 20, 24\}$ occluded ellipses. The constraint is that each ellipse must completely lie in the image and overlap with at least another one ellipse. Fig. 5 shows the statistic results of precision, recall, F-measure and executing time in millisecond. As observed,

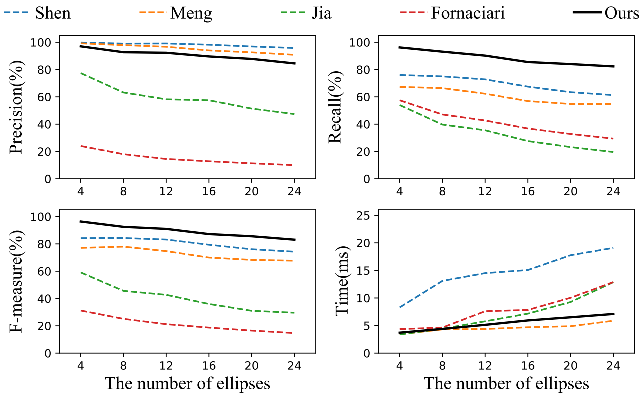


Fig. 5. **Results on synthetic occlusion datasets**. Our method achieves the highest F-measure with high speed.

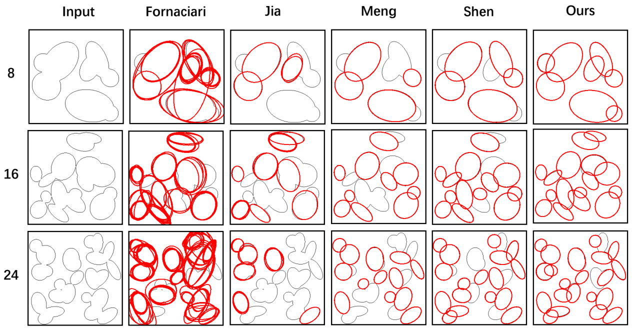


Fig. 6. **Detection samples of occlusion ellipses**. The proposed method has the most true positives than competitors.

with the number of occlusion ellipses increasing from 4 to 24, the precision, recall and F-measure decreases, and the executing time increases. The proposed method attains comparable precision with Shen and Meng, but has the highest recall and F-measure, indicating that our method can find as many correct combinations of elliptic arcs as possible. In contrast, the performance of Jia and Fornaciari is relatively unsatisfactory. As for efficiency, our detection speed is fast enough and is comparable to that of Meng, as well as being three times faster than Shen. It is worth mentioning that the method Meng requires a time-consuming initialization to execute the ellipse fitting procedure, which is not counted in the algorithm test. We present several detection samples in Fig. 6, where our method recognizes the most true positives among all the compared approaches, demonstrating its high recall and F-measure.

Real-world images. We also test the performance of our method on four cluttered real-world datasets, including Prasad+ [21], Random [9], Smartphone [9] and Tableware [12]. Prasad+ has 193 low-resolution images from the Caltech256 dataset, while Random is a multi-class dataset contains 400 images collected from the MIRFlickr and LabelMe repositories. Dataset Smartphone has 629 frames with the resolution 640 × 480 attained from several videos. Due to the varying illuminations, skew perspectives and the image blur,

Table 1. Performance comparison of different methods on the four real-world datasets and our collected concentric ellipses, where **green** and **red** font indicate the first and the second best F-measure.

Dataset	Metric	[9]	[10]	[11]	[12]	Ours
Prasad+	F-measure	34.0	43.9	58.5	43.6	59.7
	Time(ms)	17.22	16.92	19.17	17.17	12.88
	Precision	48.6	80.0	76.6	85.4	76.5
	Recall	52.8	44.1	55.6	34.2	60.6
Random	F-measure	31.6	45.6	56.9	49.2	61.3
	Time(ms)	11.30	12.15	10.98	17.00	15.23
	Precision	50.0	75.0	77.5	86.2	81.6
	Recall	49.1	47.6	55.7	43.5	59.2
Smartphone	F-measure	25.4	48.0	74.1	68.3	73.6
	Time(ms)	17.70	14.66	17.43	23.81	18.73
	Precision	43.2	63.0	82.0	86.1	81.6
	Recall	52.3	63.8	80.0	67.2	76.7
Tableware	F-measure	30.0	50.0	52.3	49.4	58.8
	Time(ms)	138.38	88.23	45.80	67.68	31.11
	Precision	40.9	56.6	58.4	56.6	76.3
	Recall	55.7	66.4	61.2	61.4	64.2
Concentric	F-measure	44.5	57.6	64.9	64.8	75.8
	Time(ms)	227.33	101.06	121.2	154.8	88.3
	Precision	35.8	54.4	67.1	57.5	75.3
	Recall	77.0	74.0	72.2	80.4	81.1

this dataset is challenging enough for detection [9]. The last dataset Tableware has 100 images and there are many concentric elliptical objects in it, which is first designed for robot grasping of elliptical objects on a table.

We report the average quantitative results in Table. 1, where our method achieves the overall best F-measure on the four datasets, which is a comprehensive metric of precision and recall. Meng has the second best performance, followed by Shen. Nevertheless, Fornaciari attains the unsatisfactory F-measure and is lower than Jia. The detection speed of our method is also fast enough, and is comparable to the fastest one. We present several detection results of the four real-world datasets in Fig. 7. The first two rows of Fig. 7 show that our method can successfully detect ellipses which are similar or close to each other, while the last two rows of Fig. 7 show that our method is robust against images with skew perspectives and cluttered textures on the background or the objects.

Concentric structure. We also test the proposed method on images with multiple concentric ellipses to verify the effect of arc grouping. We collect a number of images that have concentric ellipse structure, as illustrated in Fig. 8. The detection results are reported in Table 1, where our method attains the highest F-measure with the lowest runtime, whereby competitors consumes more time.

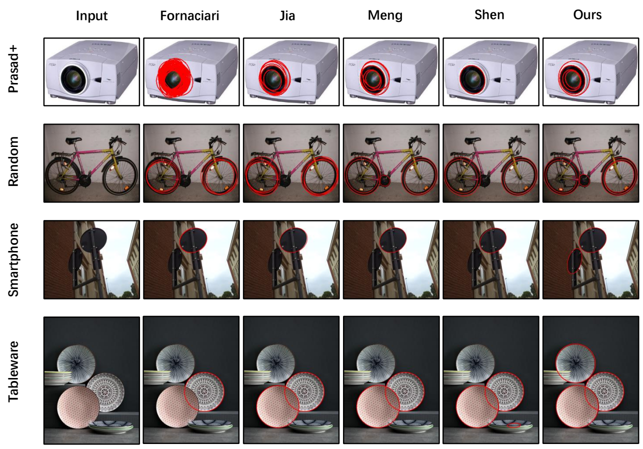


Fig. 7. **Detection samples of real-world images**. Our method has the most true positives without false detection.

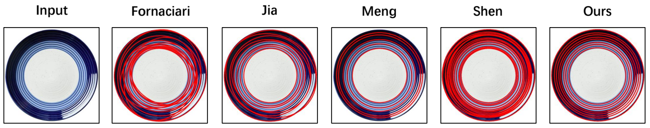


Fig. 8. **Detection results with multiple concentric ellipses**. Our method has the most correct detections.

Application to camera calibration. Thanks to the accurate and fast detection of the proposed method, we also employ it for camera calibration patterns, from which the camera intrinsic parameters can be solved by the detected elliptical centers. We use a 3D linear laser camera AT03LL020-400GM-070 capturing 65 images with resolution $1,920 \times 1,080$ from different perspectives and light conditions. Despite the skew perspectives and noise impact, our method still attains high-precision localization, and the average detection time is 57.55 ms. We present detection samples in Fig. 9, and more results are reported in the supplemental material.

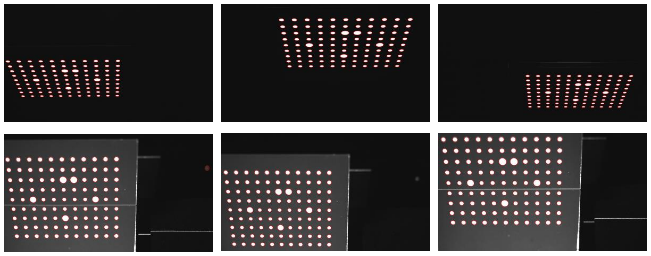


Fig. 9. **Application to camera calibration patterns**. Our method has high-precision localization under the influence of skew perspectives and noise.

4. CONCLUSIONS

We presented a fast and accurate pipeline for elliptical primitive detection in cluttered images. To combine arcs together, we first propose new geometric constraints to judge the relative location and potential of two arcs that come from the same ellipse. Then the arc relationships are encoded into a disjoint-set forest, by which we generate all candidate ellipses without omissions, especially for images with multiple concentric structure. We project the candidate ellipses to the original images, and compare the gradient differences between ellipses and image pixels, to further validate the detection. Experimental results on synthetic occlusion and real-world images demonstrate the salient advantages of the proposed approach in both accuracy and efficiency. Benefit from our fast implementation, in the future, we will apply the proposed detector to real-time elliptical object tracking.

Acknowledgements

This work was partially supported by the the National Natural Science Foundation of China (No. 62172415) and the Open Research Fund Program of State Key Laboratory of Hydroscience and Engineering (sklhse-2022-D-04).

5. REFERENCES

[1] Aryuanto Soetedjo and Koichi Yamada, “Fast and robust traffic sign detection,” in *Int. Conf. Syst., Man Cybern.*, 2005, vol. 2, pp. 1341–1346.

[2] Ronda Venkateswarlu et al., “Eye gaze estimation from a single image of one eye,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, pp. 136–143.

[3] Bogdan Kwolek, “Stereovision-based head tracking using color and ellipse fitting in a particle filter,” in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 192–204.

[4] Junli Liang, Yunlong Wang, and Xianju Zeng, “Robust ellipse fitting via half-quadratic and semidefinite relaxation optimization,” *IEEE Trans. Image. Process*, vol. 24, no. 11, pp. 4276–4286, 2015.

[5] Huixu Dong, Ehsan Asadi, Guangbin Sun, Dilip K Prasad, and I-Ming Chen, “Real-time robotic manipulation of cylindrical objects in dynamic scenarios through elliptic shape primitives,” *IEEE Trans. Robot*, vol. 35, no. 1, pp. 95–113, 2018.

[6] Zining Wang and Masayoshi Tomizuka, “Precise 3D calibration of wafer handling robot by visual detection and tracking of elliptic-shape wafers,” in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 4977–4982.

[7] Janne Heikkila, “Geometric camera calibration using circular control points,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 10, pp. 1066–1077, 2000.

[8] Viorica Pătrăucean, Pierre Gurdjos, and Rafael Grompone von Gioi, “Joint a contrario ellipse and line detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 788–802, 2016.

[9] Michele Fornaciari, Andrea Prati, and Rita Cucchiara, “A fast and effective ellipse detector for embedded vision applications,” *Pattern Recogn.*, vol. 47, no. 11, pp. 3693–3708, 2014.

[10] Qi Jia, Xin Fan, Zhongxuan Luo, Lianbo Song, and Tie Qiu, “A fast ellipse detector using projective invariant pruning,” *IEEE Trans. Image. Process*, vol. 26, no. 8, pp. 3665–3679, 2017.

[11] Cai Meng, Zhaoxi Li, Xiangzhi Bai, and Fugen Zhou, “Arc adjacency matrix-based fast ellipse detection,” *IEEE Trans. Image. Process*, vol. 29, pp. 4406–4420, 2020.

[12] Zeyu Shen, Mingyang Zhao, Xiaohong Jia, Yuan Liang, Lubin Fan, and Dong-Ming Yan, “Combining convex hull and directed graph for fast and accurate ellipse detection,” *Graph. Models*, vol. 116, pp. 101110, 2021.

[13] Paul VC Hough, “Method and means for recognizing complex patterns,” Dec. 18 1962, US Patent 3,069,654.

[14] Mingyang Zhao, Xiaohong Jia, and Dong-Ming Yan, “An occlusion-resistant circle detector using inscribed triangles,” *Pattern Recogn.*, vol. 109, pp. 107588, 2021.

[15] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein, “A probabilistic Hough transform,” *Pattern Recogn.*, vol. 24, no. 4, pp. 303–316, 1991.

[16] Robert A McLaughlin, “Randomized Hough transform: improved ellipse detection with comparison,” *Pattern Recogn. Lett.*, vol. 19, no. 3-4, pp. 299–305, 1998.

[17] Yonghong Xie and Qiang Ji, “A new efficient ellipse detection method,” in *Object recognition supported by user interaction for service robots*, 2002, vol. 2, pp. 957–960.

[18] Halil Ibrahim Cakir, Burak Benligiray, and Cihan Topal, “Combining feature-based and model-based approaches for robust ellipse detection,” in *Eur. Sign. Process. Conf.*, 2016, pp. 2430–2434.

[19] Changsheng Lu, Siyu Xia, Ming Shao, and Yun Fu, “Arc-support line segments revisited: An efficient high-quality ellipse detection,” *IEEE Trans. Image. Process*, vol. 29, pp. 768–781, 2019.

[20] Alex Yong-Sang Chia, Susanto Rahardja, Deepu Rajan, and Maylor Karhang Leung, “A split and merge based ellipse detector with self-correcting capability,” *IEEE Trans. Image. Process*, vol. 20, no. 7, pp. 1991–2006, 2010.

[21] Dilip K Prasad, Maylor KH Leung, and Siu-Yeung Cho, “Edge curvature and convexity based ellipse detection method,” *Pattern Recogn.*, vol. 45, no. 9, pp. 3204–3221, 2012.

[22] Bernard A Galler and Michael J Fisher, “An improved equivalence algorithm,” *Commun. ACM*, vol. 7, no. 5, pp. 301–303, 1964.

[23] Cihan Topal and Cuneyt Akinlar, “Edge drawing: a combined real-time edge and segment detector,” *J. Vis. Commun. Image Represent.*, vol. 23, no. 6, pp. 862–872, 2012.

[24] Urs Ramer, “An iterative procedure for the polygonal approximation of plane curves,” *Comput. Graph. Image Process.*, vol. 1, no. 3, pp. 244–256, 1972.