A Fast Line and Ellipse Detection on High Resolution Images

Limin Liu[1(✉)], Dingzhe Li[2], Zhaoxi Li[2], and Cai Meng[2,3]

[1] The Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China
2112006139@e.gzhu.edu.cn

[2] Image Processing Center, Beijing University of Aeronautics and Astronautics, Beijing 100191, China
{sy2015208,lizhaoxi}@buaa.edu.cn

[3] Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing 100083, China

Abstract. Line and ellipse are important image features in pattern recognition and computer vision. Many methods have been developed to extract line or ellipse in images separately but few try to detect them simultaneously. In this paper, a novel fast line and ellipse detection (FLED) method is proposed to detect line and ellipse simultaneously, even in high resolution images. At first, a detection framework (Pre-SGV) for high detection speed is proposed, which explicitly decomposes the detection into precalculate, segment, grouping, validation phases. Secondly, a simple but efficient algorithm is designed to segment the edges into line or arc candidates. Thirdly, the grouping constraints and fitting methods are further improved. Finally, validation are conducted to exclude erroneous detection. Experiments on synthetic images and real image dataset show that the proposed method, FLED, can robustly detect lines and ellipses fast and efficiently, especially for high resolution image (e.g. remote sensing image, the scanning image).

Keywords: Line and ellipse detection · Arc grouping · High resolution images

1 Introduction

Line and ellipse detection is one of the classical tasks in computer vision, and play an important part in vision measuring. Line detection is applied to railway detection [16], building line extraction [3], and ellipse detection is applied to robot guidance [24,31], pupil/eye tracking [34], cell segment [17] and industrial applications [30]. Recent years, there are some great algorithms have been proposed. However, most state of the art detections still can only detect lines or ellipses, and can hardly be used in the real-time applications. In case of lines and ellipses required simultaneously, such as Meng et al. [22], the lines and ellipses only can be detected step by step. A line and ellipse detection has great potential to be used in a pretreatment process which need to have low execution time and high detection accuracy.

This work is supported by the Beijing Natural Science Foundation under Grant 7202103.

Y. Wang et al. (Eds.): IGTA 2022, CCIS 1611, pp. 41–54, 2022.
https://doi.org/10.1007/978-981-19-5096-4_4

A large number of algorithms have beed studied for the line or ellipse detection currently, and these methods are illustrated as follows:

- **Line Detection Methods.** Line detection methods have been utilized for many years. The methods based on Hough [12], is first used to detect the lines. When the slope of the line tends to infinity, there is a problem with this method. The methods based on edge linking, mainly use the straightness criterion [6] to segment lines. Brian Burn [1] proposed a new method based on the consistency of the gradient direction. This method is divided into two processes, link and grouping. This method is improved by Kahn et al. [13]. Apart from these, there are some methods based on PCA, Gurn [10] proposed a method based on the minimal eigenvalue, but this method might detect the smoothing curve as lines and is sensitive to noise, and was improved by Yun-Seok Lee [15]. LSD [9] is a linear-time line segment detector giving subpixel accurate results and has high anto-noise capability, but it is easy to divide arcs to some lines.
- **Ellipse Detection Methods.** Hough transform [4] can detect the ellipse effectively, having high detection accuracy, but it requires massive computation and memory. Xu et al. [33] proposed random hough transform(RHT), this method mainly utilize the random sampling and dynamic list storage to reduce the execution time and memory requirements, but its non-sampling will introduce a large number of invalid accumulation, so RHT has been improved by [18,28]. RTED [23] uses lines to approach the edge contours and proposed many conditions to group some arcs which belong to the same ellipse. The difficulty(include accuracy and real-time performance) of ellipse detection is far greater than that of line detection. Prasad [26] proposed a method basd on edge curvature and convexify and has beed improved by Fornaciari [8].

However, in the application of the monocular Position-Pose measurement [22], line and ellipse features are required simultaneously. The traditional method is to detect lines and ellipses respectively, so the speed and efficiency is affected. Aim at the detection of line and ellipse simultaneously, There are Etemadi's detector [5], UpWrite [20], ELSD [27] methods have been realized. Etemadi's detector [5] realized the line and circle arcs detection simultaneous, and is accurate in reporting the correct detections (line and circle arcs), but has no ability in eliminating false positives, reported on parasite contours. Equally, its precision at reduced scale is poor. UpWrite [20] method can detect the line, circle and ellipse features, and first use 'The Spot Algorithm' to compute local models for every edge pixel at a resolution r. All pixels are linked based on the estimated curvature and predicted location. The GMM method is used to judge which feature type the linking pixels belong to. This method is robust with respect to noise in an image, but it has difficulties in detecting overlapping features. Furthermore, it is low efficient, and it is sensitive to the resolution radius. ELSD [27] method can classify the image edge as line and curve arc, but the algorithm stays on the basic of line and curve classification, and doesn't give the fitting information of line and curve, and the local information cann't guarantee the accurate global information. The UpWrite and ELSD methods will be compared with our method.

According to the synchronous detection requirement of the line and ellipse features, we proposed a fast line and ellipse detection method (FLED) with an improved detection framework Precalculate-Segment-Grouping-Validation**Pre-SGV** in this paper.

The framework Pre-SGV with the new fast calculate nodes (FC-nodes) has lower execution time than other compared methods, the Pre-SGV and FC-node will be illustrate in Sect. 2. In the part of the Segment, a fast segment arcs (FSA) matrix is proposed based on curvature and convexity to get line and ellipse ars candidates. Then, candidate arcs which may belong to the same ellipse are merged based on geometric constraint and improved fitting methods in the grouping process. Finally, each candidate ellipse is verified by the location and tangent constrint. Three mainly contributions of this paper are listed as follows:

- A fast framework **Pre-SGV** with FC-Nodes is designed to get low execution detection time.
- Our effective arc grouping and feature parameter estimation are proposed to get candidate combinations faster and effectively.
- A weighted verification equation and two verification constraints are proposed to make the ellipse fitting more accurately.

The rest of the paper is organized as follows: Sect. 2 introduces the Pre-SGV framework, Sect. 3 provide the FSA matrix, Sect. 4 improve the grouping constraints and Sect. 5 introduce a new validation method. Section 6 performs a number of comparative experiments and comments on the implementation of Pre-SGV. Section 7 gives the conclusion and discussion.

2 Precalculate-Segment-Group-Validate

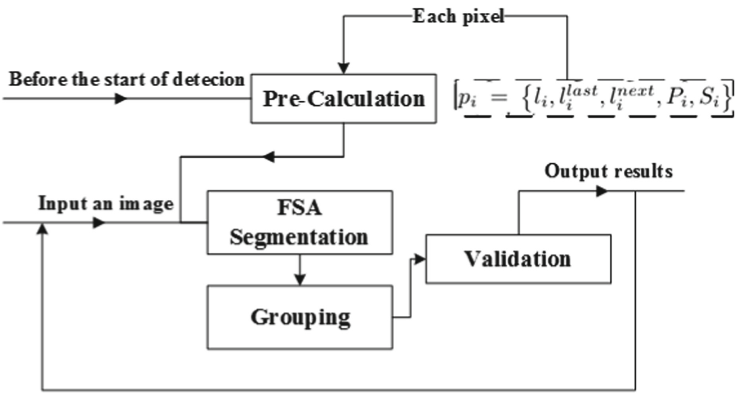

Fig. 1. The block diagram of the Pre-SGV framework

Pre-SGV is a framework designed for long-term detection of line and ellipse in a video stream or a set of images with different sizes. Its block diagram is shown in Fig. 1. The components of the framework are characterized as follows: *Pre-Calculate* allocates a given number $N^{data} = N^{data}_{row} \times N^{data}_{col}$ of nodes to get higher speed, where $N^{data}_{row}(N^{data}_{col})$ are largest than all used images' row(col) number, these nodes are named

FC-Nodes, which are defined as $P_i^{FC} = \left\{Add_i, Add_i^{last}, Add_i^{next}, F_i, F_i^{sum}\right\}, i = 1, .., N^{data}$. Add_i stores the pixel location (x_m, y_m) which $i = x_m \cdot N_{col}^{data} + y_m$, $Add_i^{last}(Add_i^{next})$ stores the last (next) location of Add_i. F_i, F_i^{sum} are real 6-by-6 symmetric matrix, we define $m_i = [x_m^2, 2x_m y_m, y_m^2, 2x_m, 2y_m, 1]$, so $F_i = m_i^T m_i$. $Add_i^{last}, Add_i^{next}, F_i^{sum}$ will be used in Sect. 3. *Segment-Group-Validate* are used based on the initialized FC-Nodes, and will be illustrated as follow sections.

3 Segmentation of Arcs

For line and ellipse feature detection, the approximate polygonal contours of the image edges are acquired. Then these contours are segmented into elliptic arc or line candidates based on the curvature and convexity. A new criterion is given for simple and efficient segmentation here. Inspired by the curvature and convexity of line and ellipse [23,26], we introduce a segment matrix based on the edge approximate contours, i.e. fast segment of arc (FSA) matrix, for efficiently and fast extracting the candidate line and ellipse arc of an image.

To get edge approximate contours, an edge image can be get by using Canny [2]. We provide two search template shown in Fig. 2(a) to get non-branched connected edge contours based on Kovesi's method [14] in an edge image. The search order is from index 1 to index 8. For example, $(x_m, y_m), m = 1, ..., M$ are pixel locations which belong to the same contours by using templates. idx_m is defined $idx_m = x_m \cdot N_{col}^{data} + y_m$. So $Add_{idx_m}^{last}, Add_{idx_m}^{next}$ will be updated by their definitions, and $F_{idx_m}^{sum} = F_{idx_{m-1}}^{sum} + F_{idx_m}, F_{idx_1}^{sum} = F_{idx_1}$. Next, Edge approximate contours are approximated by RDP algorithm [25]. Then, FSA-matrix will be get as follows.

It is known that ellipse arc set E and line set L belong to the approximate contours C, obviously, $E \bigcup L \subset C, E \bigcup L = \varnothing$. So there need a method to segment the approximate contours to get the candidate ellipse arcs fastly, and others of contours belong to the candidate lines. This method is named as FSA. FSA mainly bases on the curvature and convexity of ellipse, and three segment constraints [23,26] (curvature constraint, angle constraint and length constraint) are used to get FSA-matrix. Firstly, three basic points $\widehat{A_1 A_2 A_3}$ which satisfy the angle and length constraint need to calculate the curvature $L_{dir} = sign(\overrightarrow{A_2 A_1} \times \overrightarrow{A_2 A_3})$ where $sign$ is a sign function. For the next points $\widehat{A_{i-1} A_i A_{i+1}}$, a vector model is defined as shown in Eq. 1 and Fig. 2(b).

$$\overrightarrow{A_i A_{i+1}} = t\overrightarrow{A_i P_1} + p\overrightarrow{A_i P_2} \tag{1}$$

where t, p are unknown parameters, $\overrightarrow{A_i P_1} = \overrightarrow{A_{i-1} A_i}$, $|\overrightarrow{A_i P_2}| = |\overrightarrow{A_{i-1} A_i}|$, $\angle P_1 A_i P_2 = \theta_T$, θ_T is a threshold used in the angle constraint. FSA matrix is defined as shown in Eq. 2.

$$FSA_i^{L_{dir}} = \frac{1}{|A_{i-1}A_i|^2} \begin{bmatrix} \overrightarrow{A_{i-1} A_i} \cdot \overrightarrow{A_i A_{i+1}} \\ L_{dir} \overrightarrow{A_{i-1} A_i} \times \overrightarrow{A_i A_{i+1}} \end{bmatrix} \tag{2}$$

A const matrix is defined as $K_{FSA} = \begin{bmatrix} 1 & -cot\theta_T \\ 0 & csc\theta_T \end{bmatrix}$, and t_i, p_i can be calculate by $K_{FSA} FSA_i^{L_{dir}}$. So when $t_i > 0, p_i > 0, R_{min} < |FSA_i^{L_{dir}}| < R_{max}(R_{min}, R_{max}$

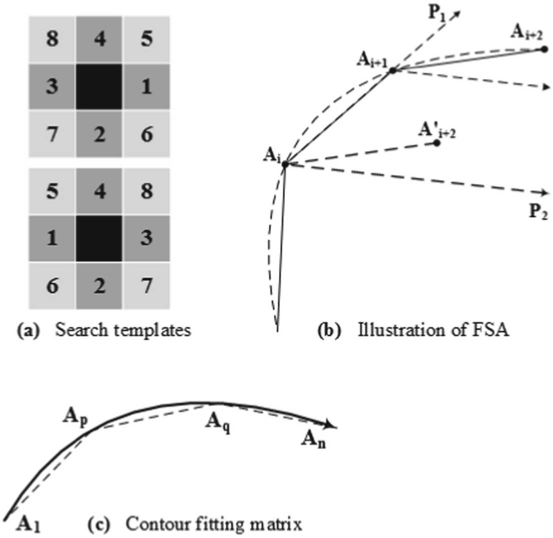

Fig. 2. (a) Shows the clockwise and anticlockwise direction template. (b) Shows the geometric meaning of FSA. (c) Shows how to get the fitting matrix of an edge contour directly.

are used in the length constraint), the point A_{i+1} and $A_{i-1}A_i$ belong to the same candidate ellipse. For some arcs that don't belong to the candidate ellipses, they will be placed in the candidate lines.

4 Arc Grouping and Feature Fitting

After arc segmentation, each elliptic arc is one part of an ellipse. But some elliptic arcs may be from same ellipse which are isolated by overlap, occlusion, or noise. For accurate detection, they should be linked or grouped into one candidate. The following method is used for grouping the edge contours that possibly belong to the same ellipse. This step is conducted in two level: local vicinity and global range. Firstly, the neighboring candidate arcs are judged whether they are from same ellipse. Secondly, the non-adjacent candidate arcs are testified whether they may be from same ellipse. Before the grouping process, all arcs are changed into the same direction.

4.1 Neighborhood Grouping

Suppose arcs A and B meet condition: $|A_nB_1| < T$, then they will be judged whether they need to be merged. Three constraints are employed for the judgement:

FSA Constraint: We get a point O_1 as the midpoint of A_nB_1. So $A_{n-1}O_1B_1$ should meet the three constraints in Sect. 3, we make $A_n = O_1, A_{n+1} = B_1$, and get the fsa matrix FSA_n^{-1}. If FSA_n^{-1} meet the constraints in Sect. 3, arc A and arc B satisfy the FSA constraint.

Curvature Constraint: The curvature of $B_{m-1}\widehat{B_m A_1}A_2$ should be of the same sign, i.e. when $\overrightarrow{B_{m-1}B_m} \times \overrightarrow{B_m A_1} < 0$ and $\overrightarrow{B_m A_1} \times \overrightarrow{A_1 A_2} < 0$, arc A and arc B satisfy the curvature constraint.

Color Constraint: if $\widehat{A_1 A_n}$ and $\widehat{B_1 B_m}$ belong to the same ellipse, in a small neighborhood of A_n and B_1, their pixel gradients should be roughly the same by using Sobel method. We define the gradient p_1 at A_n and p_2 at B_1. When $0 < \dfrac{p_1 \cdot p_2}{|p_1||p_2|} < T_{eps}$, we consider these two arcs satisfying the color constraint.

4.2 Arc Global Grouping

To judge whether two non-adjacent arcs A and B belongs to same ellipse, three constraints are employed: the curvature constraint provided by Nguyen [23], angular and fitting constraints provided by Prasad [26]. Beyond that, we propose the distance constraint. The constraints for global grouping are as follows:

Distance Constraint: If the two arcs belong to the same ellipse, the number of two arcs should be bigger than the threshold $C_{min-ellipse}$. The ellipse circumference can be calculate by $C_{ellipse} = T(\frac{r}{R}) \cdot (R+r)$. R is semi-major axis, r is semi-minor axis, $T(x)$ is elliptic coefficient. According to the experiment, we use the distance $l_{A_n B_m}$ between the end points of the arcs as the semi-major axis, so the min semi-minor axis is $R_{min} l_{A_n B_m}$, R_{min} is the minimum of the l_{major}/l_{minor}. So we get the minimum estimated circumference as shown in Eq. 3.

$$C_{min-ellipse} = (1 + R_{min})\, T\,(R_{min})\, l_{A_n B_m} \tag{3}$$

Curvature Constraint: For arc A and arc B, this constraint have the same defination at Sect. 3. So $\widehat{A_1 A_n}, \widehat{B_1 B_m}$ should satisfy $\overrightarrow{A_{n-1}A_n} \times \overrightarrow{A_n B_1} \leqslant 0, \overrightarrow{A_n B_1} \times \overrightarrow{B_1 B_2} \leqslant 0, \overrightarrow{B_{m-1}B_m} \times \overrightarrow{B_m A_1} \leqslant 0, \overrightarrow{B_m A_1} \times \overrightarrow{A_1 A_2} \leqslant 0$.

Fitting Constraint: Arc A and Arc B are fitted with DLS method, the error of DLS must be lower than a chosen threshold. The ellipse semi-minor and the ratio of semi-minor/semi-major must be higher than chosen thresholds.

$$\begin{cases} ERR_{ellipse} < T_{err} \\ l_{short} \geqslant T_{min\_short}, \dfrac{l_{minor}}{l_{major}} \geqslant R_{ratio} \end{cases} \tag{4}$$

Angular Constraint: The angle θ_i of the arc_i on the fitting ellipse should be higher than a threshold θ_{min} after ellipse fitting. If $\theta_1 + \theta_2 \geqslant \theta_{max}$, we think these two arcs belong to the same ellipse, and don't require validation. There is an arc $\widehat{A_1 A_n}$, and O is fitting center. $\overrightarrow{n_1}, \overrightarrow{n_2}$ are defined as the vectors of $\overrightarrow{OA_1}, \overrightarrow{OA_n}$. $\theta_{\widehat{A_1 A_n}}$ can be calculated as follows.

$$\theta_{\widehat{A_1 A_n}} = \pi - sign\,(\overrightarrow{n_1} \times \overrightarrow{n_2}) \left[\pi - arccos\left(\frac{\overrightarrow{n_1} \cdot \overrightarrow{n_2}}{|\overrightarrow{n_1}||\overrightarrow{n_2}|}\right)\right] \tag{5}$$

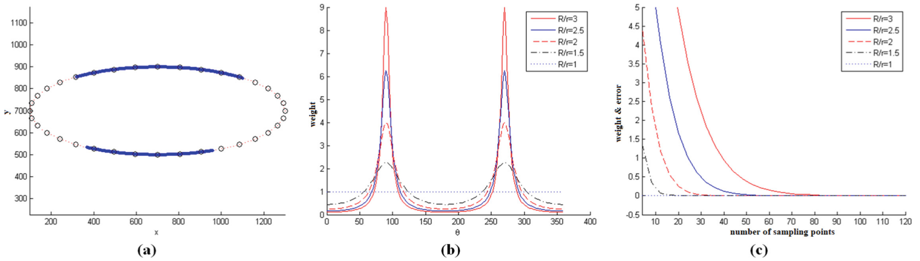

Fig. 3. Validation: (a) the location of validation points; (b) the value of W_i; (c) the discrete error of W_i.

4.3 Feature Parameter Estimation

The least squares methods is used to judge whether the error condition is satisfied for candidate lines and ellipses. The fitting matrix can be got by using FC-nodes. For example, there a arc $\widehat{A_1A_n}$, idx_{A_i}, idx_{A_n} are defined as the index of the FC-nodes. The fitting matrix $F_{A_1A_n} = F^{sum}_{idx_{A_n}} - F^{sum}_{idx_{A_1}}$.

- **Fast Line Fitting Algorithm.** The linear equation is $a_1x + a_2y + a_3 = 0$ with $a_1^2 + a_2^2 = 1$, $d_i = |a_1x_i + a_2y_i + a_3|$ is the distance from (x_i, y_i) to line. The fitting of a general line can be approached by minimizing the sum of d_i^2. S_l is defined as a contour fitting matrix which can be calculated with FC-nodes. So, the data number is $N = S_{6,6}$, the average point is $(\overline{X}, \overline{Y}) = (\frac{S_{4,6}}{N}, \frac{S_{5,6}}{N})$ which can be proved on the line, the linear direction angle is $\theta = arctan\left(\frac{2S^*_{1,2}}{S^*_{1,1} - S^*_{2,2}}\right)$. Finally, $a_1 = cos\theta, a_2 = sin\theta, a_3 = -a_1\overline{X} - a_2\overline{Y}$, the fitting error: $e = sin^2\theta S_{4,4} + cos^2\theta S_{5,5} - sin2\theta S_{4,5} - N(sin\theta\overline{X} - cos\theta\overline{Y})^2$.
- **Fast Ellipse Fitting.** The ellipse general equation is $a_1x^2 + 2a_2xy + a_3y^2 + 2a_4x + 2a_5y + a_6 = 0$, and the fitting matrix can be calculated with FC-nodes. DLS is a great developed by Fitzgibbon [7] because of its performing fit and non-iterative manner. However, DLS suffers from matrix singularity constraints and non-optimal solution will be find when all data points lie on the ideal ellipse curve [32]. Halir et al. [11] addressed this problem using an alternative formulation of the original task based on the block decomposition of matrix. In order to improve the accuracy of numerical solution, we have scaled down the data in pretreatment. Given a contour $\widehat{A_pA_q}$, the fitting result is calculated by Halir's methods [11]. A transformation from fitting result to ellipse parameter is given as follows: the ellipse center $(x_0, y_0), x_0 = (a_2a_5 - a_3a_4) \cdot SCALE, y_0 = (a_2a_4 - a_1a_5) \cdot SCALE$, angle of rotation is $\theta = \frac{arctan(2a_2/(a_1 - a_3))}{2}$. We define $a_{1p} = cos\theta a_4 + sin\theta a_5, a_{2p} = -sin\theta a_4 + cos\theta a_5, a_{11p} = a_1 + tan\theta a_2, a_{22p} = a_3 - tan\theta a_2.C_2 = a_{1p}^2/a_{11p} + a_{2p}^2/a_{22p} - a_6$, so semi-major and semi-minor axis are $l_{major} = C_2/a_{11p}, l_{minor} = C_2/a_{22p}$.

5 Validation

If an ellipse is got after grouping, there must be some edge points around this ellipse. By the same reason, there must be some edge points around lines. There are two validation constraints: (1) The Location Validation; (2) The Pixel Validation. We use the sampling points for each candidate shape as its validation points, and we judge these points whether they satisfy the two validation constraints. We give the validation equation as shown in Eq. 6 where N is the number of the validation points, $I_m(x_i, y_i), m = 1, 2$ is an indication function that if points(x_i, y_i) satisfy the validation m, $I_m(x_i, y_i) = 1, m = 1, 2$. If $R_v > R_{min}$, we consider this fitting ellipse is a real one.

$$R_v = \sum_{i=1}^{N} \frac{W_i I_1(x_i, y_i) I_2(x_i, y_i)}{N} \tag{6}$$

For the candidate lines, we take the validation points uniformly. $W_i = 1, i = 1, 2, ..., N$.

For the candidate ellipse, we define semi-major axis as R, semi-minor axis as r, the center as (x_e, y_e), the angle as θ_e. So, the validation point $[x_i, y_i] = [Rcos\theta_i, rsin\theta_i]\, R\,(\theta_e) + [x_e, y_e]$. As shown in Fig. 3(a), the more concentrated to the end points of the semi-major axis the validation points are, the closer they are. We take the rate of slope change as its weight [21], so $W_i = \dfrac{rR}{R^2cos(\theta_i)^2 + r^2sin(\theta_i)^2}$. W_i is shown in Fig. 3(b), and shown that dispersing points have bigger weight. W_i is easy to certify $\sum_{i=1}^{N} \dfrac{rR\delta\theta}{R^2cos(\theta_i)^2 + r^2sin(\theta_i)^2} \approx \int_0^{2\pi} \dfrac{rR}{R^2cos(\theta)^2 + r^2sin(\theta)^2} d\theta = 2\pi$. Because of $N = \dfrac{2\pi}{\delta\theta}$, we can get $\sum_{i=1}^{N} W_i = N$. It is shown that when N is big enough, W_i does not need to be normalized. As shown in Fig. 3(c), when N is bigger than 80, $|N - \sum_{i=1}^{N} W_i|$ is very close to 0. So N must satisfy $N > 80$.

(1) **The Location Validation.** For each validation point $V_i = [x_i, y_i]$, if there exists a edge point on the V_i 8 neighborhood edge points, V_i pass the location validation. Mark $I_1(x_i, y_i) = 1$.

(2) **The Grad Validation.** For each validation point $V_i = [x_i, y_i]$, the point $V_i^{next_N}, V_i^{last_N}$ is the V_i next and last Nth point. Mark $l_i = \overrightarrow{V_i^{last_N} V_i^{next_N}}$. The gradient g_i at V_i is $(-Rsin\theta_i, rcos\theta_i)$ for ellipse, and $(cos\theta_l, sin\theta_l)$ for line where θ_l is the line slope. GV_i is the score of the slope similarity and can be calculated by $GV_i = \dfrac{g_i \cdot l_i}{|g_i||l_i|}$, if GV_i is larger than the threshold T_{GV}. The point V_i is considered that it pass the grad validation, and mark $I_2(x_i, y_i) = 1$.

Finally, the validation score R_v can be got by the Eq. 6. If the R_v is larger than the threshold T_{RV}. This ellipse (line) can be place the real ellipse (line).

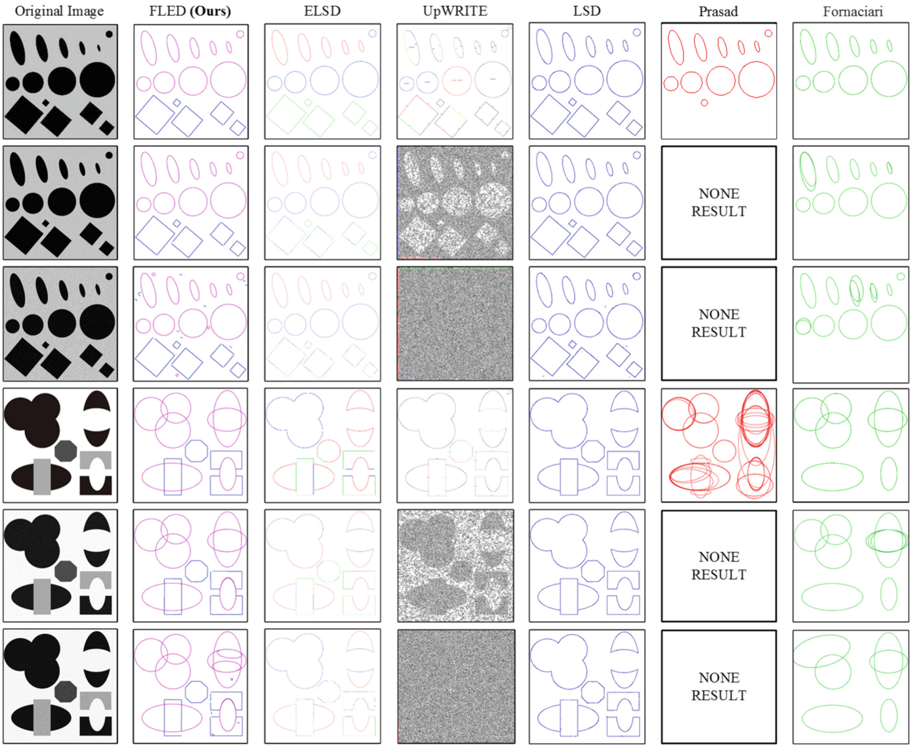

Fig. 4. Results on simulation images (1000 × 1000) of geometric shapes (square, ellipse) for different noise types and different scales. From left to right: Original Image, FLED (Ours), ELSD, UpWRITE, LSD, Prasad, Michele. From top to bottom: noise-free image, 0.05-Gaussian noise, 0.1-Gaussian noise, overlapping shapes, 0.05-Gaussian noise, 0.1-Gaussian noise

Table 1. Comparison of execution time on low resolution images (msec)

Image name	Size	Execution time (ms)				
		FLED (Ours)	ELSD	LSD	Prased	Fornaciari
//LR//1.png	442 × 640	**5.78781**	270	16.4607	821	7.92355
//LR//2.png	480 × 640	**7.75933**	610	19.3336	NONE	16.3411
//LR//3.png	480 × 640	**13.1629**	2930	27.8511	NONE	12.2071
//LR//4.png	488 × 700	**10.9402**	1380	23.2976	NONE	20.6639
//LR//5.png	375 × 500	**7.7169**	1030	17.5384	NONE	10.9937

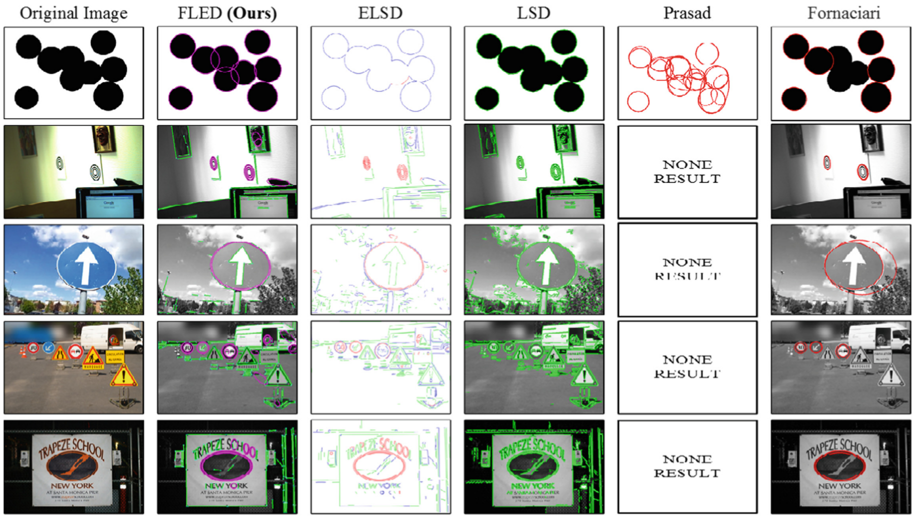

Fig. 5. Results on low resolution images. From left to right: Original Image, FLED (Ours), ELSD, LSD, Prasad, Michele.

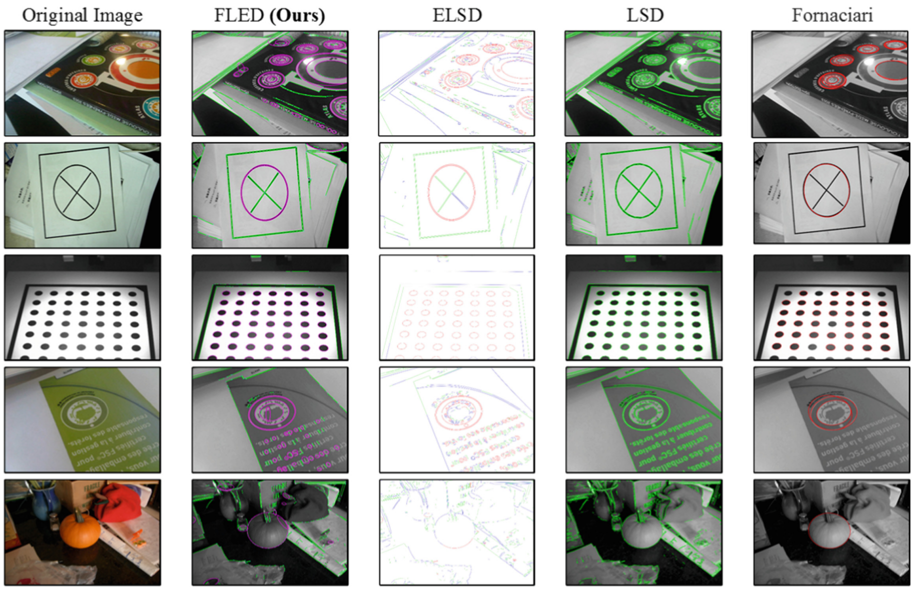

Fig. 6. Results on ordinary resolution images. From left to right: Original Image, FLED (Ours), ELSD, LSD, Prasad, Michele.

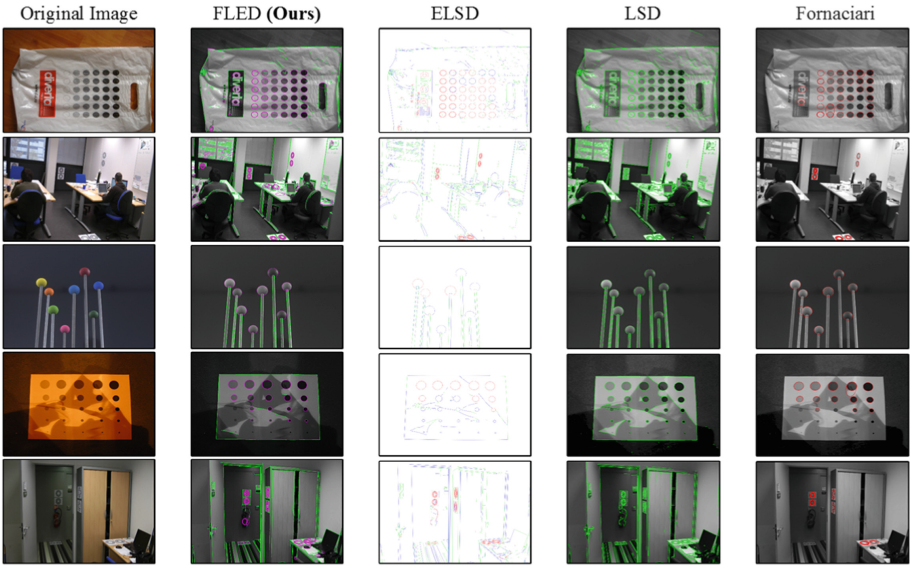

Fig. 7. Results on high resolution images. From left to right: Original Image, FLED (Ours), ELSD, LSD, Prasad, Michele.

Table 2. Comparison of execution time on ordinary and high resolution images (msec)

Size (OR)	Execution Time (ms)				Size (HR)	Execution Time (ms)			
	FLED	ELSD	LSD	Fornaciari		**FLED**	ELSD	LSD	Fornaciari
768 × 1024	**29.6**	3070.0	88.0	72.0	1200 × 1600	**62.1**	4750.0	125.2	1071.0
540 × 960	**10.0**	1180.0	32.6	13.9	1080 × 1920	**142.2**	7350.0	134.9	336.9
1024 × 1280	**15.3**	1890.0	64.8	95.5	1920 × 2560	**76.1**	3970.0	251.5	145.7
768 × 1024	**19.1**	1940.0	46.9	63.5	1200 × 1600	**17.3**	1830.0	92.7	149.2
960 × 1280	**46.0**	4170.0	103.3	129.8	1080 × 1920	**86.3**	2980.0	115.7	168.6

6 Experiments

To verify the proposed method, FLED is tested and compared with other methods. In the comparison, UpWrite [19], ELSD [27], LSD [29], Prasad [26] and Fornaciari [8] are selected, whose codes are available on-line. The experiments can be divided into two parts: With synthetic images and with real images. The synthetic images are simulation images validate methods performance from noise and overlapping. The real images are divided into three categories: (1) The images whose resolution are less than 500 thousand pixels are named Low Resolution Images (LR Image); (2) The images whose resolution are from 500 to 1 500 thousand pixels are named Ordinary Resolution Images (OR Image); (3) The images whose resolution are larger than 1 500 thousand pixels are named High Resolution Images (HR Image). All detectors are tested with their default parameters.

6.1 Simulation Images

The simulation images including both overlapping and non-overlapping geometric shapes, are used to analyse the robustness against noise and shape scales. The result is shown in Fig. 4. We analyze the characteristics of each method as follows.

- **ELSD**: ELSD can detect ellipse arcs effectively on each shape scale and has good anti-noise capability. For overlapping shapes, this method takes on high robustness. But when it comes to line segments, some lines are considered as circles.
- **UpWRITE**: UpWRITE detect geometric shapes by using GMM, and perform well on hand-drawn shape. But UpWRITE has high error detection rate, for flat ellipse and overlapping shapes, UpWRITE performs poorly and has low anti-noise capability.
- **LSD**: LSD is a line detector, it can detect all lines, perform well with high anti-noise capability. But LSD detect an arc as a set of lines.
- **Prased**: Prased's method can detect all ellipse, but have high error detection, while with low anti-noise capability.
- **Fornaciari**: Fornaciari is an ellipse detection method, it has high anti-noise capability and performs well. Fornaciari has low error detection, but has high missed detection.
- **FLED (Ours)**: FLED can detection lines and ellipses at once, performs well and has high anti-noise capability, FLED can detection most all lines and ellipse, for some overlapping shapes, this method might get some error detection.

After evaluating the performance of the method mentioned above, we decide to drop the UpWRITE method in the following real image test section.

6.2 Real Images

In this section, we test our method with other methods in real images. These images are tested in LR Images, OR Images and HR Images respectively, we get results as shown in Fig. 5, Fig. 6 and Fig. 7, the execution time is shown in Tables 1 and 2. The Prased not perform well in LR Images results, so this method won't be used in OR and HR Images.

From the result provided above, ELSD can get lines and ellipses, but its execution time is very high, ELSD can't be used for real-time application. Meanwhile, there are many lines detected as circles in ELSD. LSD can detect all lines in the image, unable to identify that some lines may belong to the same arcs, and Fornaciari's method can detect ellipse, and has low error detection, but has high miss detection. Our method FLED, can detect most lines and ellipse directly, particular, the execution time of FLED is fastest all of these methods. So, no matter in performance or in execution time, our method takes on superior performance.

7 Conclusion

In this paper, we have proposed a fast and robust line segment and elliptical shape detection method. We have made a lot of innovative improvements of the existing methods.

Through a lot of experiments, we adjust and verify the method parameters to achieve the best performance. Experiments show that, compared with the traditional representative methods. Our method (FLED) exhibit absolute advantages such as fast, robust and so on. So it can be widely used in hardware and software of video streamprocessing problems, especially in the processing of high resolution images and video, Our method has a broad application prospect.

Despite its performance, there are some room for improving in our method still has some space for improvement. When elliptic arc segments are divided very small, it is easy to occur false detection and missed detection and miss detection. In the future work, we will continue to improve our method on the basis of FLED from the following aspects: (1) Design a method for grouping muti-arcs to reduce the error detection. (2) Improve the fitting method to get more accurate results. (3) Improve the validation constraints to reduce the error detection. We hope our algorithm will have a better performance in the future.

References

1. Burns, J.B., Hanson, A.R., Riseman, E.M.: Extracting straight lines. Read. Comput. Vis. **8**(4), 180–183 (1987)
2. Canny, J.: A computational approach to edge detection. Read. Comput. Vis. PAMI **8**(6), 184–203 (1986)
3. Ding, W.: Extracting straight lines from building image based on edge orientation image. Acta Optica Sinica **30**(10), 2904–2910 (2010)
4. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. IPSJ Mag. **13**(1), 512–513 (1975)
5. Etemadi, A.: Robust segmentation of edge data. In: International Conference on Image Processing and Its Applications, pp. 311–314 (1992)
6. Faugeras, O.D., Deriche, R., Mathieu, H., Ayache, N., Randall, G.: The depth and motion analysis machine. Int. J. Pattern Recognit. Artif. Intell. **6**(3), 143–175 (1992)
7. Fitzgibbon, A.W., Pilu, M., Fisher, R.B.: Direct least squares fitting of ellipses. IEEE Trans. Pattern Anal. Mach. Intell. **1**(5), 253–257 (1996)
8. Fornaciari, M., Prati, A., Cucchiara, R.: A fast and effective ellipse detector for embedded vision applications. Pattern Recogn. **47**(11), 3693–3708 (2014)
9. Gioi, R.G.V., Jakubowicz, J., Morel, J.M., Randall, G.: LSD: a fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. **32**(4), 722–32 (2010)
10. Guru, D.S., Shekar, B.H., Nagabhushan, P.: A simple and robust line detection algorithm based on small eigenvalue analysis. Pattern Recogn. Lett. **25**(1), 1–13 (2004)
11. Halir, R.: Numerically stable direct least squares fitting of ellipses (1999)
12. Hough, P.V.C.: Method and means for recognizing complex patterns (1962)
13. Kahn, P., Kitchen, L., Riseman, E.M.: A fast line finder for vision-guided robot navigation. IEEE Trans. Pattern Anal. Mach. Intell. **12**(11), 1098–1102 (1990)
14. Kovesi, P.D.: MATLAB and octave functions for computer vision and image processing (2000)
15. Lee, Y.S., Koo, H.S., Jeong, C.S.: A straight line detection using principal component analysis. Pattern Recogn. Lett. **27**(14), 1744–1754 (2006)
16. Li, Q., Shi, J., Li, C.: Fast line detection method for railroad switch machine monitoring system. In: International Conference on Image Analysis and Signal Processing, pp. 61–64 (2009)

17. Li, X., Wang, Y., Deng, Y., Yu, J.: Cell segmentation using ellipse curve segmentation and classification. In: International Conference on Information Science and Engineering, pp. 1187–1190 (2009)
18. Liu, Z.Y., Qiao, H., Xu, L.: Multisets mixture learning-based ellipse detection. Pattern Recogn. **39**(4), 731–735 (2006)
19. Mclaughlin, R.A., Alder, M.: The hough transform versus the upwrite. IEEE Trans. Pattern Anal. Mach. Intell. **20**(4), 396–400 (1997)
20. Mclaughlin, R., Alder, M.: Technical Report - The Hough Transform versus the UpWrite. University of Western Australia (1997)
21. Meng, C., Li, Z., Bai, X., Zhou, F.: Arc adjacency matrix-based fast ellipse detection. IEEE Trans. Image Process. **29**, 4406–4420 (2020). https://doi.org/10.1109/TIP.2020.2967601
22. Meng, C., Xue, J., Hu, Z.: Monocular position-pose measurement based on circular and linear features. In: International Conference on Digital Image Computing: Techniques and Applications (2015)
23. Nguyen, T.M., Ahuja, S., Wu, Q.M.J.: A real-time ellipse detection based on edge grouping, vol. 5, no. 4, pp. 3280–3286 (2009)
24. Park, S., Kim, G.W.: Expanded guide circle-based obstacle avoidance for the remotely operated mobile robot. J. Electr. Eng. Technol. **9**(3), 1034–1042 (2014)
25. Poiker, T., Douglas, D.H.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica Int. J. Geogr. Inf. Geovisualization **10**(2), 112–122 (1973)
26. Prasad, D.K., Leung, M.K.H., Cho, S.Y.: Edge curvature and convexity based ellipse detection method. Pattern Recogn. **45**(9), 3204–3221 (2012)
27. Pătrăucean, V., Gurdjos, P., von Gioi, R.G.: A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7573, pp. 572–585. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33709-3_41
28. Qiao, Y., Ong, S.H.: Arc-based evaluation and detection of ellipses. Pattern Recogn. **40**(7), 1990–2003 (2007)
29. Rafael, G.V.G., Jérémie, J., Jean-Michel, M., Gregory, R.: LSD: a fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Mach. Intell. **32**(4), 722–32 (2010)
30. Schleicher, D.C.H., Zagar, B.G.: Image processing to estimate the ellipticity of steel coils using a concentric ellipse fitting algorithm. In: International Conference on Signal Processing, pp. 884–890 (2008)
31. Thamizharasan, S., Baskaran, J., Ramkumar, S.: A new cascaded multilevel inverter topology with voltage sources arranged in matrix structure. J. Electr. Eng. Technol. **10**(4), 1553–1558 (2015)
32. Wong, C.Y., Lin, S.C.F., Ren, T.R., Kwok, N.M.: A survey on ellipse detection methods, pp. 1105–1110 (2012)
33. Xu, L., Oja, E.: Randomized hough transform (RHT): basic mechanisms, algorithms, and computational complexities. Comput. Vis. Image Underst. **57**(2), 131–154 (1993)
34. Yang, Q., Hu, H., Gui, W., Zhou, S., Zhu, C.: 3-parameter hough ellipse detection algorithm for accurate location of human eyes. J. Multimediad **9**(5), 619–626 (2014)