A fast and robust ellipse detector based on top-down least-square fitting

Yongtao Wang
wyt@pku.edu.cn
Zheqi He
philokeys@gmail.com
Xicheng Liu
liuxicheng@pku.edu.cn
Zhi Tang
tangzhi@pku.edu.cn
Luyuan Li
liluyuan@pku.edu.cn

Institute of Computer Science & Technology
Peking University
Beijing, China

Abstract

Ellipse detection is a very important problem in the field of pattern recognition and computer vision. The existing algorithms often use a bottom-up strategy to combine edge points or elliptical arcs into ellipses, hence limit their robustness. In this paper, we propose a fast and robust ellipse detection algorithm which can accurately detect ellipses in the images. The main idea of the proposed algorithm is to exploit a novel top-down fitting strategy to combine edge points into ellipses and use integral chain to speed up the fitting process. Experimental results have demonstrated that our ellipse detection algorithm achieves a better performance than the state-of-the-art methods on the common evaluation measures of F1 score and average execution time.

1 Introduction

Detecting ellipses efficiently and accurately in digital images is a fundamental problem in the field of pattern recognition and computer vision. Ellipse detection can serve a wide range of applications such as traffic sign recognition [7, 20], vehicle safety enhancement [21], rectification of broadcast sports video [8], cell detection and counting [6].

In the past three decades, researchers have proposed a lot of methods to detect ellipses in digital images. The three major concerned issues of ellipse detection are accuracy, robustness and computational performance. Hough transform (HT) can be used to effectively detect ellipses and estimate the parameters of ellipses. Since an ellipse is analytically defined by five parameters, the standard HT needs a five-dimensional (5D) accumulator to estimate the parameters. But the time and space cost to use a 5D accumulator is very high. In order to overcome the shortcoming of the standard HT, one possible way is to use only a subset of the edge points rather than using all of them in the voting process. For instance, the randomized Hough transform (RHT) [16], the probalistic Hough transform(PHT) [10] and the fuzzy

Pages 156.1-156.12
DOI: https://dx.doi.org/10.5244/C.29.156

HT[9] are all devised based on this idea. Splitting the five dimensional space into subspaces with less dimensionality is another possible way to decrease the time and space complexity of the standard HT. For example, the iterative randomized Hough transform(IRHT) [14] uses five one-dimensional accumulators to replace the original five-dimensional accumulator. Chia *et al*. [2] compute four parameters geometrically and then estimate the last parameter by a 1D accumulator.

In addition to HT based methods, some heuristic methods have also been proposed to detect ellipses, such as genetic algorithm (GA) [24] and genetic algorithm with multiple populations (MPGA) [23]. These algorithms can produce good results under certain circumstances but have the risk of finding a suboptimal solution.

The methods mentioned above have a common problem that their detection time is usually very long. Recently, some real-time ellipse detection algorithms are proposed. The framework of these methods are similar: first construct elliptical arcs or regions according to the geometric properties of the edge points, and then use a fitting method, such as least-square fitting [4, 19], RANdom SAmple Consensus(RANSAC) [15] approach and HT, to estimate the parameters of ellipses. Prasad *et al*. [18] propose an edge curvature and convexity based ellipse detection method which first extracts the canny edge map of an image and chains the edge points into edge segments, and then fits line segments on each edge segment to approximately represent the edge segments. After that, they use the convexity of line segments to get elliptic arcs and further combine elliptic arcs into ellipses. Patraucean *et al*. [17] propose an ellipse detection algorithm named ELSD. They first exploit a seed growing scheme to find line-support regions, and further chain the line-support regions into curve regions based on a convexity rule and a smoothness rule. After that, they estimate an ellipse for each curve region with a fitting technique which merges the algebraic distance with the gradient orientation. And finally, false detections are further eliminated by a *contrario* validation technique. More recently, Fornaciari *et al*. [5] propose a very fast method to detect ellipses which first splits the canny edge map into many short elliptic arcs, and then classify these elliptic arcs into four types by their edge direction and convexity. They define a candidate ellipse as a triplet, i.e. a set of three arcs that satisfy a set of criteria, then use HT to determine the parameters of the ellipses. Finally, they validate the candidate ellipses using a criterion based on the algebraic fitting error, and perform a clustering procedure to deal with multiple detections. The method [5] is extensively tested on many datasets, and the experiments results reported in [5] demonstrate that it is very efficient, and outperforms many state-of-the-art methods [11, 16, 18, 25]. In spite of their good performance on many datasets, the preceding methods are not robust enough to handle clutter interruption and noise, due to that they all adopt the bottom-up strategy to find candidate ellipses. Moreover, their efficiency can also be improved.

In this paper, we propose a novel algorithm which can efficiently and accurately detect ellipses in digital images by exploiting a novel top-down scheme. The proposed method is very efficient, which is faster than the most efficient method [5] that has been reported. Our experimental results also demonstrate that our algorithm is more robust than the state-of-the-art methods including [5] and ELSD [17], for handling Gaussian noise. The rest of the paper is organized as follows. In Section 2, we introduce the background. In Section 3, we describe the proposed algorithm in detail. In Section 4, we present the experimental results. Finally, we conclude our work in Section 5.

2 Background

2.1 Method of ellipse fitting

In this work, the conventional least square fitting technique is exploited to estimate the parameters of an ellipse. Without loss of generality, the ellipse equation is given by

$$f(x,y) = ax^2 + by^2 + cxy + dx + ey - 1 = 0, \tag{1}$$

where a,b,c,d,e are the five ellipse parameters. And for a given set of points $\{p_1(x_1,y_1),...,$ $p_n(x_n,y_n)\}$, the aim is to find the optimal estimations of a,b,c,d,e which can minimize the the algebra square error $\sum_{i=1}^{n} f(x_i,y_i)^2$.

Before fitting, we usually perform a translation operation to make the set of points with zero mean, which means that we actually minimize the following term instead:

$$S = \sum_{i=1}^{n} f(u_i,v_i)^2, \tag{2}$$

where

$$u = x - \bar{x}, \quad v = y - \bar{y}, \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i. \tag{3}$$

This problem can be solved by setting the derivation of each parameter $\frac{\partial S}{\partial a}$, $\frac{\partial S}{\partial b}$, $\frac{\partial S}{\partial c}$, $\frac{\partial S}{\partial d}$, $\frac{\partial S}{\partial e}$ to zero and concretely, we can get following linear system:

$$\begin{pmatrix} \sum_{i=1}^n u_i^4 & \sum_{i=1}^n u_i^2 v_i^2 & \sum_{i=1}^n u_i^3 v_i & \sum_{i=1}^n u_i^3 & \sum_{i=1}^n u_i^2 v_i \\ \sum_{i=1}^n u_i^2 v_i^2 & \sum_{i=1}^n v_i^4 & \sum_{i=1}^n u_i v_i^3 & \sum_{i=1}^n u_i v_i^2 & \sum_{i=1}^n v_i^3 \\ \sum_{i=1}^n u_i^3 v_i & \sum_{i=1}^n u_i v_i^3 & \sum_{i=1}^n u_i^2 v_i^2 & \sum_{i=1}^n u_i^2 v_i & \sum_{i=1}^n u_i v_i^2 \\ \sum_{i=1}^n u_i^3 & \sum_{i=1}^n u_i v_i^2 & \sum_{i=1}^n u_i^2 v_i & \sum_{i=1}^n u_i^2 & \sum_{i=1}^n u_i v_i \\ \sum_{i=1}^n u_i^2 v_i & \sum_{i=1}^n v_i^3 & \sum_{i=1}^n u_i v_i^2 & \sum_{i=1}^n u_i v_i & \sum_{i=1}^n v_i^2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n u_i^2 \\ \sum_{i=1}^n v_i^2 \\ \sum_{i=1}^n u_i v_i \\ \sum_{i=1}^n u_i \end{pmatrix}. \tag{4}$$

After getting the estimation of ellipse parameters by solving this linear system, we can further calculate the geometric parameters like center, major/minor axis and rotation from a, b, c, d, e.

2.2 Our criterion for candidate ellipses selection

As what we will introduce in Sec. 3, after obtaining the ellipse parameters through least square fitting, we check whether an edge segment with a set of points $\{p_1,...,p_n\}$ can be selected as a candidate ellipse based on the following criterion:

$$\sum_{i=1}^{n} g(\varepsilon, p_i)/n > th_{score}, \tag{5}$$

$$g(\varepsilon, p_i) = \begin{cases} 1, & \text{if } |f(x_i,y_i)| < 0.1 \\ 0, & \text{if } |f(x_i,y_i)| >= 0.1 \end{cases}, \tag{6}$$

th_{score} is a threshold and we set it to 0.7 in all of the experiments.

3 Proposed algorithm

In this paper, we propose a novel ellipse detection method based on the top-down edge segment analysis. As shown in Fig. 1, our method consists of four major steps. In the first step, we extract edge segments from the input image and then obtain line segments within each edge segment. Secondly, we detect candidate ellipses from each edge segment by the top-down analysis. After that, we cluster candidate ellipses to merge and remove the fractional detections. Finally, we eliminate false alarms using NFA validation. The rest of this section describes each step in more detail.

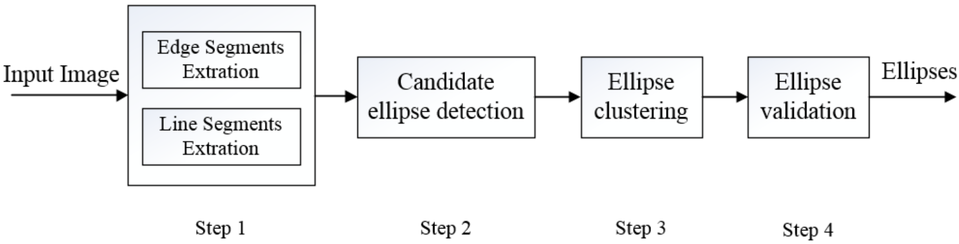


Figure 1: Workflow of the proposed ellipse detection method

3.1 Edge segment and line segment extraction

We directly use the edge segment extraction method and the line segment detection method proposed in [12, 13], respectively, which are very computational efficient. Since they are not our own work, we only briefly introduce these two methods here, for the details please refer to [12, 13].

1. **Edge segment extraction:** In order to extract edge segments(a chain of 8-connected edge pixel points) within the input image, we first obtain the Canny [1] edge points and then chain them to form the edge segments. We calculate the gradient of input image I(smoothed by a $5*5$ Gaussian kernel with $\sigma = 1.3$) by the Sobel operator and determine the high threshold T_h and the low threshold T_l in an adaptive way [1]. After that, the pixels are categorized into three types: 1) p is a *strong edge point* if the gradient magnitude at p is a local maximum and larger than T_h ; 2) p is a *weak edge point* if the gradient magnitude at p is a local maximum and in the range of T_l to T_h ; 3) p is not considered as an edge point. The process of chaining the edge points to form an edge segment consists of two steps: 1) pick the point with the largest gradient magnitude among the strong edge points as the starting point p_0 to trace an edge segment; 2) construct a list and add p_0 to it as the tail and then iteratively add the un-searched edge points from the 8-neighbours of the tail to the list as the new tail, until none of the 8-neighbours of the tail is an un-searched edge point.

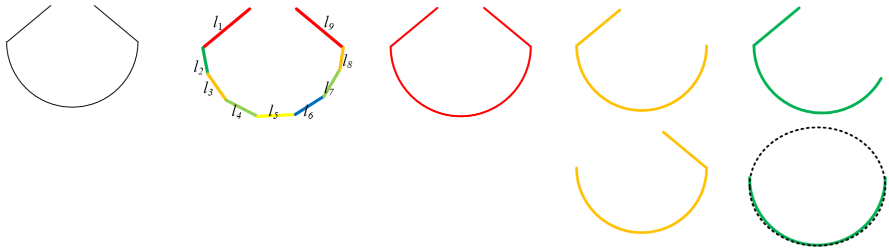
2. **Line segment extraction:** We detect line segments within each edge segment by a top-down scheme. For a given edge segment $E = \{p_1, ..., p_n\}$, we gradually verify its sub-segments from the longer ones to the shorter ones. To be more specific, we first verify the sub-chain with length $l = n$(the edge segment itself). If the sub-chain with $l = n$

satisfies the straightness criterion, we accept it as a line segment. Otherwise, we further verify the sub-chains with length $l = n-1$ and so on. Once a line segment is found, we remove the edge points belonging to it from the original edge segment, and then continue to detect line segments in the rest parts of the edge segment. After obtaining the line segments, we use the obtained line segments to represent their corresponding edge segment, i.e. using a set of line segments $\{l_1,...,l_m\}$ to represent E.

3.2 Candidate ellipse detection

In this section, we describe how to detect candidate ellipses from the edge segments. We fit the edge segments into ellipses using a top-down least-square fitting scheme, and speed up our least-square fitting by integral chain.

3.2.1 Edge segment selection with a top-down scheme



(a) (b) (c) (d) (e)

Figure 2: (a) An edge segment; (b) We use line segments $\{l_1,...,l_9\}$ to roughly represent the edge segment; (c) At the first iteration, we use the chain corresponding to $\{l_1,...,l_9\}$ to fit an ellipse; (d) At the second iteration, we use the chain corresponding to $\{l_1,...,l_8\}$ and $\{l_2,...,l_9\}$ to fit an ellipse; (e) At the third iteration, we use the chain corresponding to $\{l_1,...,l_7\}$, $\{l_2,...,l_8\}$ and $\{l_3,...,l_9\}$ to fit an ellipse, and $\{l_2,...,l_8\}$ can be fitted as a valid ellipse(represented by the dashed line), so we stop the iteration.

Different from the conventional bottom-up strategy which iteratively fits line segments or arcs into ellipses from shorter ones to longer ones, we apply a novel top-down edge segment analysis scheme. Such scheme is more robust to background noise and it is more likely to detect complete elliptical arcs.

First, we represent an edge segment by line segments instead of points (as described in Sec. 3.1), because this representation will make the top-down scheme more efficient. For a given edge segment $E = \{l_1,...,l_m\}$, the line segments ($\{l_1,...,l_m\}$) are first ordered by their positions on the edge segment. Then we iteratively detect ellipses among the sub-chains of E. A sub-chain of E consists of a successive subset of line segments in E. We verify the sub-chains from longer ones to shorter ones. For example, we first verify the sub-chain of length $M = m$, i.e., the E itself. If it can not be fitted as an ellipse, we further verify the sub-chains of length $M = m-1$ i.e., $\{l_1,...,l_{m-1}\}$ and $\{l_2,...,l_m\}$ can be fitted as an ellipse or not. Once we find an ellipse, we remove its corresponding line segments from E and continue this process on the left parts of E. Fig. 2 gives an example of this top-down scheme.

3.2.2 Speed up with integral chain

The above top-down analysis process could be very time-consuming. Hence, we use the 1-D case(which is called "integral chain" hereafter) of integral image technique [22] to speed up the proposed ellipse detection algorithm. Specifically, for a sequence $x_1, x_2, ..., x_n$, the corresponding integral chain is:

$$IntChain(k) = \sum_{i=1}^{k} x_i, k = 1, ..n. \tag{7}$$

The recurrent formulas for computing the integral chain are:

$$IntChain(1) = x_1, \quad IntChain(k) = IntChain(k-1) + x_k, \quad k > 1. \tag{8}$$

Within the integral chain, the sum of any sub-sequence can be efficiently obtained by

$$\sum_{i=1}^{k} x_i = IntChain(k), \sum_{i=s}^{k} x_i = IntChain(k) - IntChain(s-1). \tag{9}$$

Noting that we actually use $u = x - \bar{x}$, $v = y - \bar{y}$ to fit ellipse, therefore, we should expand the terms $\sum_{i=l}^{r} u_i^k v_i^h$ where $0 <= k, h <= 4$ to calculate them from the integral chains. For example,

$$\sum_{i=l}^{r} u^2 = \sum_{i=l}^{r} (x - \bar{x})^2 = \sum_{i=l}^{r} x^2 - \frac{1}{(r-l+1)} (\sum_{i=l}^{r} x_i)^2. \tag{10}$$

3.3 Ellipse clustering

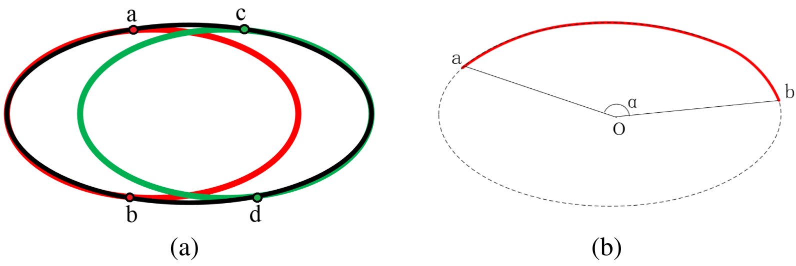


Figure 3: (a) The edge segment belonging to the black ellipse are miss-split into two parts *ab* and *cd*, and fitted as two ellipses: the red one and the green one. We can merge them back into the black one by clustering. (b) The red edge segment *ab* is fitted into the ellipse ε drawn by dashed lines, O is the center of ε, α is the central angle of ε.

As shown in Fig. 3(a), sometimes, especially under noise, a complete ellipse may be split into two or more candidate ellipses due to the incomplete edge segment extraction results. To address this problem, we further cluster the candidate ellipses according to their elliptic arcs (for the reason that, each candidate ellipse corresponding to an elliptic arc) to form new candidates and eliminate the fractional ones. First, the candidate ellipses are sorted according to the central angles (its definition is given in the legend of Fig. 3(b)) of their elliptic arcs. Then, in each iteration, we select the candidate ε_i with maximum central angle as the base and combine it with the rest candidates ε_j to see if a better ellipse can be found. If a better ellipse ε_b is found, we replace ε_i with ε_b and remove ε_j form candidates. After repeating the process for every candidate ellipse, we remove the candidates corresponding to the short elliptic arcs (in our case,short elliptic arcs are the arcs whose length is less than 0.38 portion of the perimeter of the corresponding ellipse).

3.4 Ellipse validation

To further eliminate false alarms, we employ the Desolneux's [3] technique to validate the clustered ellipses, which is similar to ELSD [17]. The ellipse validation method is based on the Helmholtz Principle [3], which means that for a structure to be perceptually meaningful, the expectation of this structure by chance must be very low. To be more specific, for an ellipse with l independent edge points, there are k of these points having gradient directions aligned with the direction of this ellipse, where aligned means that a point p_i is at the direction orthogonal to the tangent of the ellipse on p_i. The expectation of this kind of ellipses, the so-called "number of false alarms"(NFA), should be less than φ , that is:

$$NFA(l,k) = N_t \sum_{i=k}^{l} \binom{l}{i} p^i (1-p)^{l-i} < \varphi, \tag{11}$$

where N_t is the number of all possible ellipses; φ is set to 1 and the precision of the direction alignment p is set to $1/8$ as detailed in [17]. We calculate the gradient direction of each edge point in the original image with $2*2$ mask [3].

ELSD [17] also uses NFA to discard the false detections, but there are some differences between it and our method. First, the number of all possible ellipse N_t is different. As five points can determinate an ellipse, we set N_t to n^5, where n is the number of the edge points in the image, rather than $(NM)^4$(the image size is $N \times M$ and the degree of freedom for elliptical arc is 8) in ELSD. Second, for a elliptical arc $\{p_1,...,p_n\}$, ELSD performs the validation on the whole sequence. Howerer, we perform the validation on its sub-sequences arc$\{p_1, p_3, ..., p_{2i-1}\}$ and $\{p_2, p_4, ..., p_{2i}\}$ to guarantee the independence of the points [3]. If neither of them satisfy the condition (11), we label this candidate ellipse as a false detection.

4 Experimental results

This section presents an evaluation of our method and compares its performance with two state-of-the-art ellipse detection methods: (1) Fornaciari *et al.*'s method (FORN) which is proposed in [5], and (2) ELSD [17] which is proposed by Puatruaucean *et al.*The implementations of FORN and ELSD are downloaded from the authors' websites[1].

4.1 Evaluation metrics

We evaluate the detection effectiveness of the methods according to the evaluation methodology proposed in [18]. We determine the correctness of a detected ellipse ε_1 according to its overlap ratio D with its corresponding ground truth ellipse ε_2. D is defined as follow:

$$D = 1 - \frac{count(XOR(\varepsilon_1, \varepsilon_2))}{count(OR(\varepsilon_1, \varepsilon_2))}. \tag{12}$$

[1] http://sourceforge.net/p/yaed/code/HEAD/tree/; http://ubee.enseeiht.fr/vision/ELSD/

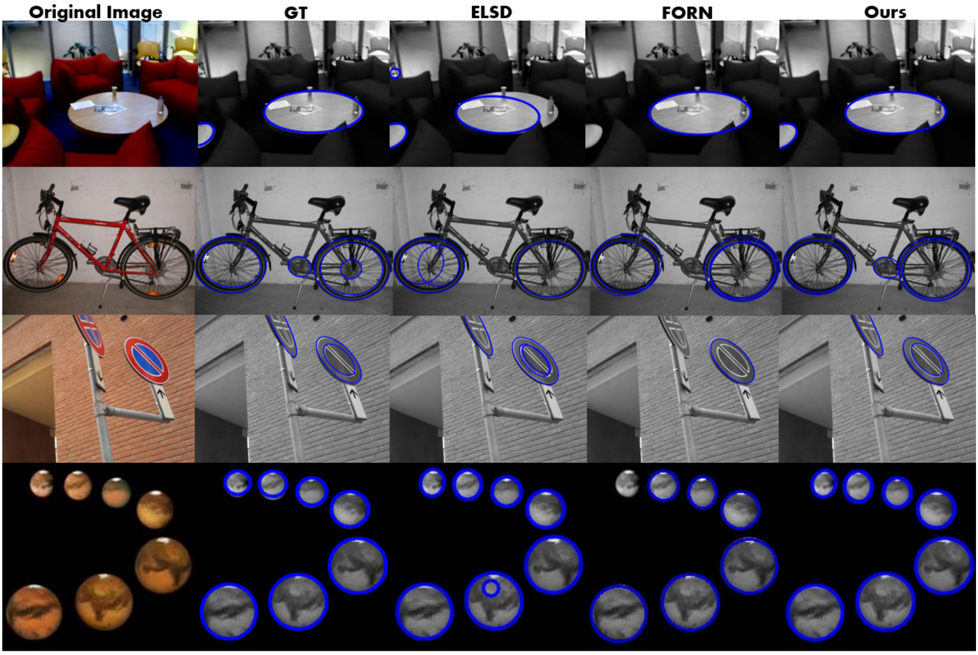


Figure 4: Sample visual results of ELSD, FORN, and our method

A detected ellipse is considered as a true positive(TP) if it has a overlap ratio $D > 0.8$ with a ground truth ellipse. If the detected ellipse does not match with any of the ground truth ellipses, then it is counted as a false positive(FP). A false negative(FN) is found if a ground truth ellipse does not have a match ellipse among the detected ellipses. According to these, the detection effectiveness is calculated in the aspect of F1 score. Besides the effectiveness, we also evaluate the methods' execution speed and their robustness to noise. The execution speed of a method is evaluated by its average execution time (ms/image) on the datasets. And the methods' robustness to noise is evaluated by their detection effectiveness on the datasets with different level of Gaussian noise.

4.2 General experiment

We make experiments on three public real image datasets which are available on-line[2]. Dataset #1 is proposed in [18], which contains 197 images. Dataset #2 and Dataset #3 are proposed in [5]. Dataset #2 consists of 400 real images which are collected from MIR-Flickr and LabelMe repositories. Dataset #3 is composed of 629 frames at the resolution of 640x480 which are selected from several videos. Sample visual results of ELSD, FORN, and our method are illustrated in Fig. 4. From which, it can be seen that our method performs much better than the others: ELSD tends to find some false detections and FORN sometimes misses the true positives, while our method gets the accurate result. The F1 scores of each method on the three datasets are illustrated in Fig. 5(a), and their corresponding average execution time on each dataset is summarized in Table 1. It can be seen that the proposed

[2]http://imagelab.ing.unimore.it/files/ellipse_dataset.zip

Method \ Dataset	Dataset #1	Dataset #2	Dataset #3
Ours	8.3	22.6	41.5
Fornaciari	10.5	36.2	53.1
ELSD	173.4	480.7	1419.3

Table 1: Overall execution time(ms/image)

Step \ Dataset	Dataset #1	Dataset #2	Dataset #3
Edge detection	2.0	7.1	15.1
Line detection	4.1	10.1	17.4
Ellipse fitting	1.9	4.6	8.5
Clustering	0.3	0.6	0.8
Validation	0.04	0.1	0.2

Table 2: Detailed Execution time(ms/image)

method achieves better performance than the other two methods: it runs faster while achieving the highest F1 score. The execution time of each processing step for these datasets is shown in Table. 2. It can be seen that the top-down ellipse fitting scheme is very fast and the bottleneck of the scheme is the line detection process.

4.3 Anti-noise experiment

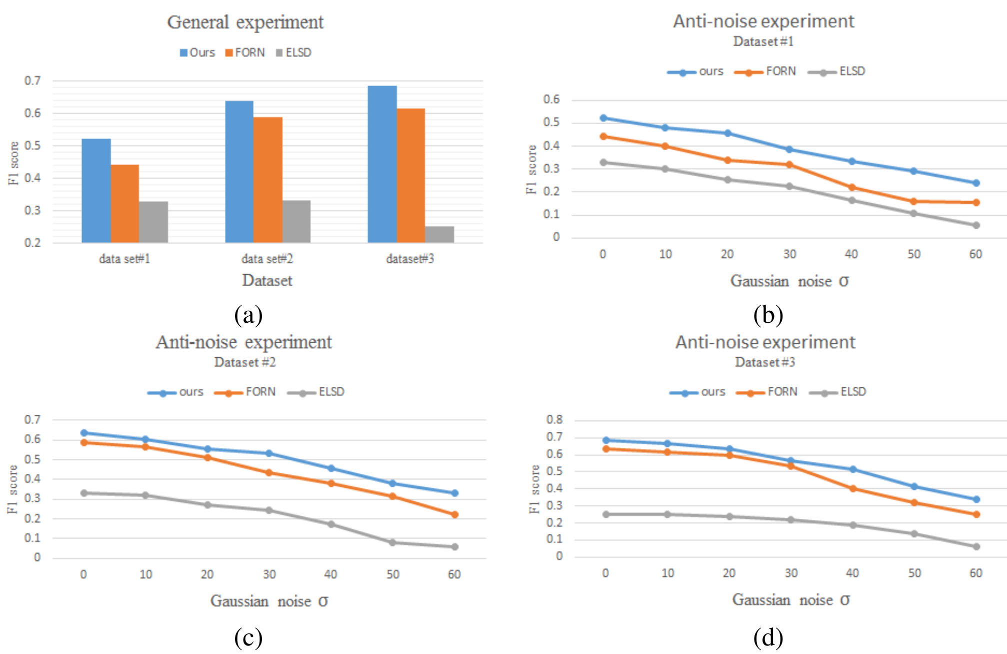


Figure 5: (a) F1 scores of general experiment; (b-d) F1 scores of anti-noise experiment

To evaluate the methods' robustness to noise, we add Gaussian noise with different level (the variance $\sigma^2 = \{0, 10^2, 20^2, 30^2, 40^2, 50^2, 60^2\}$) to the images in the three datasets and run each method again. The F1 scores of the three methods under different levels of Gaussian noise are shown in Fig. 5(b)-(d). The result indicates that our method is more robust than the two other methods for handling Gaussian noise.

5 Conclusion

This paper proposes a novel method for ellipse detection, which is efficient, robust and parameter-free.

Our major contributions are: (1) a top-down least-square fitting analysis scheme for finding candidate ellipses, which is more robust than the conventional bottom-up strategy used by the state-of-the-art methods; (2) a more effective validation process for filtering out the false detections, in which we calculate the number of all possible ellipse through a more reasonable rule and guarantee the independence of the edge points by validating through every other point. Extensive experimental results demonstrate that: (1) the proposed algorithm is indeed more accurate and robust than multiple state-of-the-art methods [5, 17] in terms of F1 score; (2) the proposed algorithm is very fast, even faster than the most efficient method [5] which has ever been proposed; (3) the proposed algorithm is more robust to noise than the methods [5, 17].

6 Acknowledgment

This work is supported by National Natural Science Foundation of China (Grant 61300061 and Grant 61472014) and Beijing Natural Science Foundation (4132033).

References

[1] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[2] Alex YS Chia, Maylor KH Leung, How-Lung Eng, and Susanto Rahardja. Ellipse detection with hough transform in one dimensional parametric space. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V–333. IEEE, 2007.

[3] Agnès Desolneux, Lionel Moisan, and Jean-Michel Morel. Meaningful alignments. *International Journal of Computer Vision*, 40(1):7–23, 2000.

[4] Andrew Fitzgibbon, Maurizio Pilu, and Robert B Fisher. Direct least square fitting of ellipses. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(5): 476–480, 1999.

[5] Michele Fornaciari, Andrea Prati, and Rita Cucchiara. A fast and effective ellipse detector for embedded vision applications. *Pattern Recognition*, 47(11):3693–3708, 2014.

[6] Wesley Nunes Gonçalves and Odemir Martinez Bruno. Automatic system for counting cells with elliptical shape. *arXiv preprint arXiv:1201.3109*, 2012.

[7] Yanlei Gu, Tomohiro Yendo, Mehrdad Panahpour Tehrani, Toshiaki Fujii, and Masayuki Tanimoto. A new vision system for traffic sign recognition. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pages 7–12. IEEE, 2010.

[8] Ankur Gupta, James J Little, and Robert J Woodham. Using line and ellipse features for rectification of broadcast hockey video. In *Computer and Robot Vision (CRV), 2011 Canadian Conference on*, pages 32–39. IEEE, 2011.

[9] Joon H Han, LászlóT Kóczy, and Timothy Poston. Fuzzy hough transform. *Pattern Recognition Letters*, 15(7):649–658, 1994.

[10] Nahum Kiryati, Yuval Eldar, and Alfred M Bruckstein. A probabilistic hough transform. *Pattern recognition*, 24(4):303–316, 1991.

[11] Lars Libuda, Ingo Grothues, and Karl-Friedrich Kraiss. Ellipse detection in digital image data using geometric features. In *Advances in Computer Graphics and Computer Vision*, pages 229–239. Springer, 2007.

[12] Dong Liu, Yongtao Wang, Zhi Tang, and Xiaoqing Lu. A robust circle detection algorithm based on top-down least-square fitting analysis. *Computers & Electrical Engineering*, 40(4):1415–1428, 2014.

[13] Dong Liu, Yongtao Wang, Zhi Tang, and Xiaoqing Lu. A robust and fast line segment detector based on top-down smaller eigenvalue analysis. In *Fifth International Conference on Graphic and Image Processing*, pages 906916–906916. International Society for Optics and Photonics, 2014.

[14] Wei Lu and Jinglu Tan. Detection of incomplete ellipse in images with strong noise by iterative randomized hough transform (irht). *Pattern Recognition*, 41(4):1268–1279, 2008.

[15] Fei Mai, YS Hung, Huang Zhong, and WF Sze. A hierarchical approach for fast and robust ellipse extraction. *Pattern Recognition*, 41(8):2512–2524, 2008.

[16] Robert A McLaughlin. Randomized hough transform: improved ellipse detection with comparison. *Pattern Recognition Letters*, 19(3):299–305, 1998.

[17] Viorica Pătrăucean, Pierre Gurdjos, and Rafael Grompone Von Gioi. A parameterless line segment and elliptical arc detector with enhanced ellipse fitting. In *Computer Vision–ECCV 2012*, pages 572–585. Springer, 2012.

[18] Dilip K Prasad, Maylor KH Leung, and Siu-Yeung Cho. Edge curvature and convexity based ellipse detection method. *Pattern Recognition*, 45(9):3204–3221, 2012.

[19] Dilip K Prasad, Maylor KH Leung, and Chai Quek. Ellifit: An unconstrained, non-iterative, least squares based geometric ellipse fitting method. *Pattern Recognition*, 46 (5):1449–1465, 2013.

[20] Aryuanto Soetedjo and Koichi Yamada. Fast and robust traffic sign detection. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 2, pages 1341–1346. IEEE, 2005.

[21] Mohan M Trivedi, Tarak Gandhi, and Joel McCall. Looking-in and looking-out of a vehicle: Computer-vision-based enhanced vehicle safety. *Intelligent Transportation Systems, IEEE Transactions on*, 8(1):108–120, 2007.

[22] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[23] Jie Yao, Nawwaf Kharma, and Peter Grogono. A multi-population genetic algorithm for robust and fast ellipse detection. *Pattern analysis and applications*, 8(1-2):149–162, 2005.

[24] Peng-Yeng Yin. A new circle/ellipse detector using genetic algorithms. *Pattern Recognition Letters*, 20(7):731–740, 1999.

[25] Si-Cheng Zhang and Zhi-Qiang Liu. A robust, real-time ellipse detector. *Pattern Recognition*, 38(2):273–287, 2005.