ROBUST ELLIPSE DETECTION VIA ARC SEGMENTATION AND CLASSIFICATION

Huixu Dong, I-Ming Chen, Fellow, IEEE

Robotics Research Centre
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798

Dilip K. Prasad

School of Computer Science and Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798

ABSTRACT

In this paper, we propose a novel ellipse detection algorithm for synthetic and real images. Existing ellipse detection methods are too slow when used with limited hardware resources. The proposed method demonstrates the capability of detecting ellipses with an excellent accuracy at an acceptable speed level in three public datasets. The excellent performance is attributed to the novel combination of classification of arcs into different quadrants of a candidate ellipse, edge curvature and convexity-concavity analysis, and an elliptic geometry constraint.

Index Terms— Ellipse detection, Edge curvature, Convexity-concavity, Arc classification

1. INTRODUCTION

Ellipse detection is applied in real life widely such as medical diagnosis, recognition of traffic signs and security [1-3]. Most existing approaches use clustering/voting techniques such as Hough transform [4] and its variants [5-9]. The processing speeds of these methods suffer due to the voting process among a great deal of candidate points. On the other hand, arc accumulation techniques exploit local information at the level of arcs rather than the global information at the level of edge pixels [10-14]. These methods achieve high accuracy in ellipse detection, but still suffer from high computational costs. Relatively faster methods employ algebraic or geometric constraints to select candidates for fitting an ellipse [15-17]. These approaches are reasonably accurate for images with simple backgrounds but there performance quickly deteriorates in complex background. Considering the drawbacks of these methods, we propose a method of the consecutive different-angle to split any curve to class arcs correctly so that this method has a good performance comparing with other recent approaches in three datasets.

2. PREPROCESSING

This section describes the pre-processing steps before ellipses are fit on digital edge curves of an image.

2.1. Arc extraction

Several geometric constraints are employed in the pre-processing itself to select the arcs that potentially represent ellipses. Performing this selection in pre-processing translates to computational efficiency in subsequent processing. Canny edge detector [18] with automatic thresholding in the image is followed by identification of connected and unbranched edge segments, referred to as arcs. We adopt the Ramer-Douglas-Peucker algorithm [19, 20] to approximate an arc into line segments. If the number of line segments fitted on an arc is less than a threshold, the arc is concluded to be a line and omitted from further processing. Arc with a large radius may correspond to the center located outside the image while arc with a small radius may correspond to noise. In order to filter away such unsuitable arcs which consume huge computation resources uselessly, we use the concept of oriented bounding box (OBB), i.e. the oriented minimum area rectangle that encloses an arc [21]. Unsuitable arcs are readily identified as the arcs with OBBs outside a range of thresholds. Remaining arcs are considered for further processing. The effect of the threshold values is investigated in the experiments.

2.2. Arc segmentation by three critical conditions

As it is well known the curvature change over an elliptic arc is continuous and smooth, we select edge curves with such property using three geometric constraints, which we discuss in the following sub-sections. We use the coordinates of the pixels in arc specified as (x_i, y_i) and the phase of the gradient η_i for analysing the edge properties.

2.2.1. Three types of critical points

Since the Sobel derivatives dx and dy are zero along the vertical and horizontal gradient directions respectively, the values of their gradients are infinite and zero. The edge points with the vertical gradient $dx = 0$ or horizontal gradient $dy = 0$ are used as the critical points, and referred to as gradient boundary points. With reference to Figure 1, we define that a point of large change of curvature as a

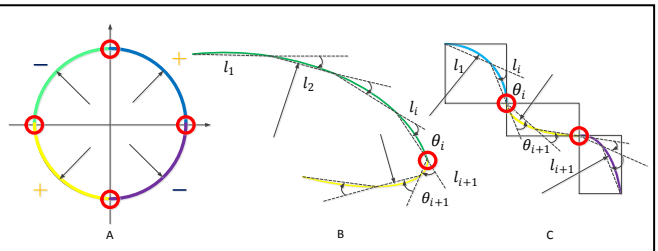

Fig.1.Segmenting an arc by (A)the gradient boundary points,(B)the turning corner and (C) the inflexion points. Specifically, the red circles represent the critical points and the line segments with arrows indicate the direction of the gradient through the point. The arcs with different colours presents the split arcs based on the critical points. The curves are fit by the dash lines. An angle θ_i is formed by the line segments l_i and l_{i+1}.

turning corner and the point of the change of direction of curvature as an inflexion point.

2.2.2. Approach for detecting the critical points

Accuracy of the values of η_i is compromised due to digitization in the image but the problem is alleviated since the sign and not the exact value of η_i is sufficient in detecting the gradient boundary points [14]. Thus, we use the gradient sign function Ω defined as follows

$$\Omega(p_i) = \text{sign}(\tan(\eta_i)) = \text{sign}(dx) \cdot \text{sign}(dy). \quad (1)$$

The curves are split at the boundary gradient points such that arcs with the positive gradient direction rests on the first and third quadrants while other arcs belong to the second and fourth quadrants, as shown in Figure 2.

The Ramer-Douglas-Peucker algorithm is adopted to fit line segments on each arc to gain an approximate representation of the curvature after extracting curves. Therefore, we just focus on some points of arc rather than all of edge points in order to improve the computational efficiency. The line segments fit on the edge curves are applied to extract the smooth edge curves that are potential elliptic arcs. Beginning from one end of the edge curves, the points where the curvatures becomes irregular can be explored such that each edge curve formed out of this process is divided into some smooth curves that may consist of ellipses.

Here we present the approach to find the turning corner and inflexion points. Corresponding to a line segment approximation of a cruve $c: \{l_1, l_2, \dots, l_N\}$, the angles between pairs of the consecutive line segments are denoted as $\{\theta_1, \theta_2, \dots, \theta_{N-1}\}$, where θ_i is the vector angle from l_i to l_{i+1} in the range $\theta_i \in [-\pi, \pi]$, as shown in Figure 1(B,C). The turning points are identified through the following constraint:

$$|\theta_i - \theta_{i+1}| > Thre_a.$$

where, we set the threshold value $Thre_a$ empirically to be sufficiently high to conclude a large change in the curvature. Regarding the inflexion points, the change of the sign (positive and negative) between the consecutive angles

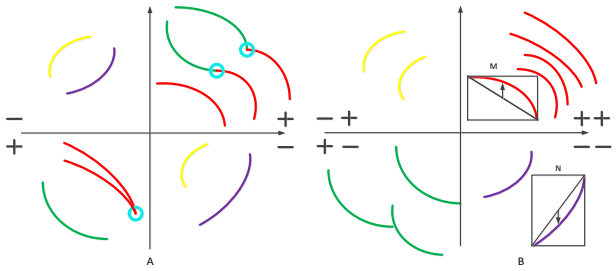

Fig.2. Arc classification depending on the sign of the phase of the gradient (A); arcs classed further by the convexity-concavity(B).The blue circles represent the turning corners and inflexion points. M and N indicate the bounding boxes used in exploring the convexity-concavity of the arc. The line segments with arrows in a bounding box denote the different vector. The positive or negative signs indicate the sign of the phase of the gradient for each point on an arc and the convexity or concavity.

(θ_i, θ_{i+1}) indicates the change for the direction of the curvature.

2.3. Arc classification by the convexity-concavity

After splitting at critical points, each arc has an unique convex or concave property while being smooth and continuous. Thus, according to the convexity-concavity, all the arcs can be classified further in the four quadrants. An arc a is enclosed by a bounding box and its two end points form a line segment. If the direction of the difference vector $\boldsymbol{\delta}$ between the two middle points of the arc and the line segment is positive or negative, the arc is convex or concave, respectively (Figure 2). The convexity-concavity can be confirmed by the following function

$$\Gamma(a) = \begin{cases} +, \boldsymbol{\delta} > 0; \\ -, \boldsymbol{\delta} < 0. \end{cases} \quad (2)$$

The difference vector $\boldsymbol{\delta}$ is zero implies that the convexity-concavity cannot be determined and such arc is omitted. Therefore, based on the function Ω and Γ, each arc a can be classed to the corresponding quadrant by the function Φ

$$\Phi(a) = \begin{cases} \text{I}, & \langle (\Omega(p_i), \Gamma(a)) \rangle = \langle +, + \rangle \\ \text{II}, & \langle (\Omega(p_i), \Gamma(a)) \rangle = \langle -, + \rangle \\ \text{III}, & \langle (\Omega(p_i), \Gamma(a)) \rangle = \langle +, - \rangle \\ \text{IV}, & \langle (\Omega(p_i), \Gamma(a)) \rangle = \langle -, - \rangle \end{cases} \quad (3)$$

3. ELLIPSE DETECTION

3.1. Selection of arcs consisting of the potential ellipse

We choose a set of three arcs $\tau_{abc} = (a_a, a_b, a_c)$ as a triplet that satisfy the grouping requirements discussed below as an indicator of the arcs likely belonging to the same ellipse. We adopt the grouping method based on relative position from [14] to select potential arcs sets at small computational cost. For example, the horizontal coordinate of the arc in the first or fourth quadrant is bigger than that of the arcs in the second or third quadrant while the vertical coordinate of the arc in the first or second quadrant is smaller than that of arc from the third or fourth quadrant. A pair of such arcs is

defined as $\mathcal{E}_{ab} = (a_a, a_b)$, and thus two pairs of arcs with a common arc between them form one set $\tau_{abc} = \{(\mathcal{E}_{ab}, \mathcal{E}_{bc}) | \, a_b \equiv a_b \}$ for a candidate ellipse by sharing the same arc. We impose an additional constraint on the locations of centers of ellipses corresponding to such arcs. If the centers of arcs in four quadrants lie in the same area with a range, they are likely to belong to the same ellipse. The center of the ellipse is estimated by the intersection points of lines through the midpoints of parallel chords. The sets thus selected satisfy the following constrains,① convexity-concavity,②the relative position and ③the same ellipse center. For these sets, we use the method of [7, 22, 23] to determine the remaining parameters of the ellipse by decomposing the parameter space.

3.2. The validation of the candidate ellipse

Even though the arcs satisfy the above three constraints, these constraints are insufficient to ensure that the fitted ellipse is a valid ellipse. We use a false detection control to reject invalid ellipses as described below.

3.3.1. Confirmation by the ratio of the circumference

We compute the differential distance of the positions of a point (x_i, y_i) on an arc and an ellipse arc as follows:

$$X = \frac{[(x_i - x_c)\cos\rho + (y_i - y_c)\sin\rho]^2}{a^2},$$
$$Y = \frac{[(y_i - y_c)\cos\rho - (x_i - x_c)\sin\rho]^2}{b^2}. \quad (4)$$

where (x_c, y_c),a,b,and ρ represent the centre,major and minor semi-axes and orientation angle of the ellipse,respectively. If the position of a point satisfies the inequality $|X + Y - 1| < threshold$, it implies that this point is close to the boundary of the ellipse. We define a set of points that satisfy this inequality and compute a function $\psi(\tau_{abc})$ defined as follows:

$$\psi(\tau_{abc}) = \frac{\mathcal{B}}{a_a + a_b + a_c} \quad (5)$$

where $\mathcal{B}$ is the number of points in the above defined set and a_a, a_b, a_c are the number of pixels in three arcs used for fitting the ellipse. We note that $\psi(\tau_{abc})$ indicates the relative number of good pixels in an arc set $\psi(\tau_{abc})$. If $\psi(a)$ is more than a threshold, we consider this ellipse is valid, otherwise this ellipse is discarded.

3.3.2. Clustering by the difference of real and potential arcs

Since more than one arc sets may correspond to the same 'real ellipses', multiple candidate ellipses may be fit for the same 'real ellipse'. Such candidate ellipses should be clustered [17]. All valid ellipses with the same centre are sorted in the decreasing order of the score $\mathrm{K}(\tau_{abc})$. The ellipse with the highest score is regarded as the reference of a cluster. We obtain the distance between the points on the arc set τ_{abc} corresponding to another ellipse:

$$X_{ref} = \frac{[(x_i - x_{c,ref})\cos\rho_{ref} + (y_i - y_{c,ref})\sin\rho_{ref}]^2}{{a_{ref}}^2},$$
$$Y_{ref} = \frac{[(y_i - y_{c,ref})\cos\rho_{ref} - (x_i - x_{c,ref})\sin\rho_{ref}]^2}{{b_{ref}}^2},$$

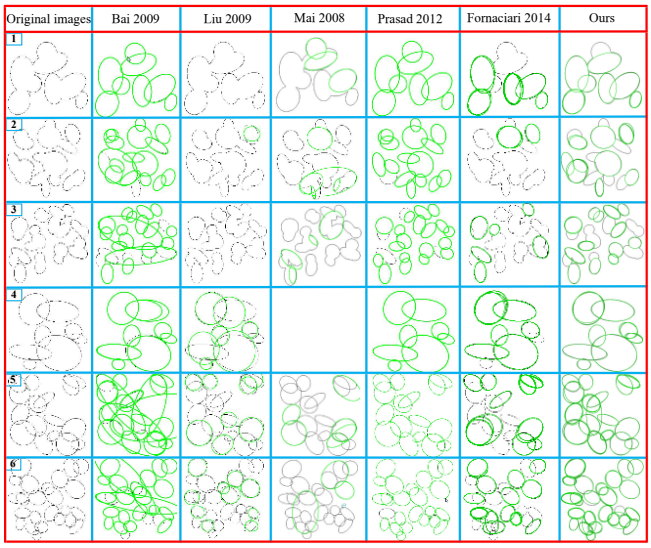

Fig.3. The detection results of the synthetic images. 1,2,3 represents the input images including 8,16,24 occluded ellipses while 4,5,6 illustrates the images including 8,16,24 overlapping ellipses, respectively. There is no any result in Mai(4).

and form a set of points satisfying $X_1 + Y_1 - 1 < threshold$. We define a function $\chi(\tau_{abc})$ to assess the similarity among ellipses:

$$\chi(\tau_{abc}) = \frac{\mathrm{T}}{a_a + a_b + a_c}, \quad (6)$$

where T is the number of elements in the set defined above. If the value of $\chi(\tau_{abc})$ is more than the threshold, it is considered to be similar to the reference one. If the current ellipse does not belong to any cluster, it becomes a reference of a new cluster.

4. EXPERIMENTAL RESULTS

The proposed methods are tested in Dataset of synthetic images with occluded or overlapping ellipses [17]. We also use the real images of Dataset Prasad [24] and dataset #1 [14] with complex and varied backgrounds to demonstrate the performance of the proposed method. The evaluation metrics, *F*-measure and the execution time [17] to assess the performance of the proposed method compared against five recent ellipse detection methods proposed by Fornaciari et. al [14], Prasad et. al [17], Mai et. al [25], Bai et. al [26], Liu et. al [27]. Computation times for executions on a PC with 4GB of RAM and an Intel Core i5 processor are reported.

The detection cases of chosen synthetic images are presented in Figure 3. As illustrated in Figure 5(occluded ellipses) and 6(overlapping ellipses), the proposed method provides the F-measure closest to the method of Prasad et. al. However,other methods reveal poorer performances in the detection effectiveness. As the execution time taken is quite small for Dataset of synthetic images[17], we do not need to present the comparison results regarding these methods.

Fig.4. The detection results of the real images. The image 1-5 from Dataset Prasad, the image 6-10 from Dataset#1. There is no any result in Mai(4).

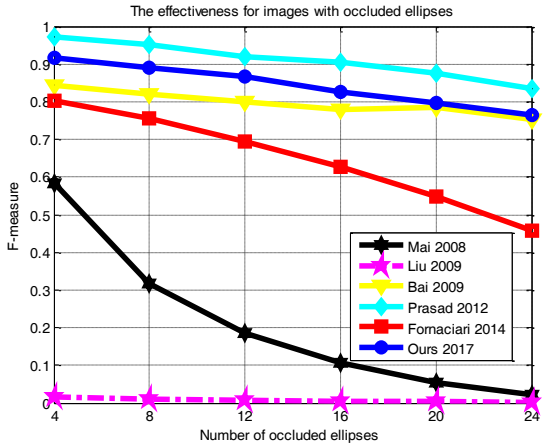

Fig.5. $F\text{-}measure$ for images with γ occluded ellipses, $4 \leq \gamma \leq 24$.

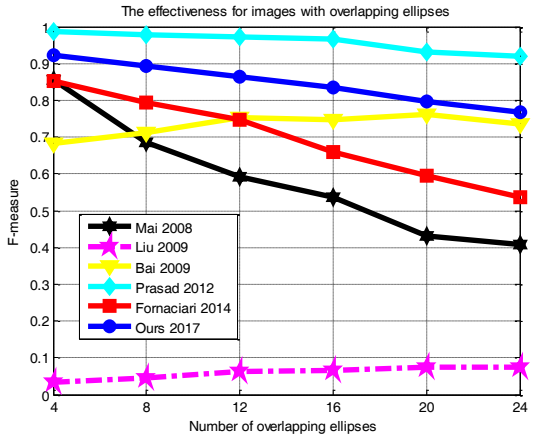

Fig.6. $F\text{-}measure$ for images with γ overlapping ellipses, $4 \leq \gamma \leq 24$.

	Dataset Prasad	
	F-measure(%)	Average time(ms)
Bai 2009	16.85	23.107
Liu 2009	8.08	20.06
Mai 2008	18.31	28.53
Prasad 2012	44.18	158.32
Fornaciari 2014	43.70	5.56
Ours	47.56	18.17

Table.1. The average time and F-measure comparisons for the real images.

The practical cases of detecting real images in Dataset Prasad and Dataset #1 are illustrated in Fig. 4. For comparing the performances based on Dataset Prasad, we adopt the results of five detection methods reported in [14, 17]. Mean values of the metrics for the real images of Dataset Prasad are listed in Table 1. It is evident that the presented method outperforms other methods in terms of F-measure. Moreover, Table 1 shows that the proposed method is more than 30 times faster than Prasad et. al for real images. The proposed method is also the fastest among all the methods except the method of Fornaciari et. al.

The performance of our method outweighs the method of Fornaciari et al in terms of F-measure, especially in detecting small, occluded, or overlapping ellipses. This advantage is attributed to splitting of curves based on the three geometric constraints described in section 3. Nevertheless, our method may discard some small arcs in the pre-processing steps, which may result into false detections and hence poorer performance in comparison to the method of Prasad et. al. We also note that the classification of arcs costs additional computation time in comparison to the method of Fornaciari et. al. In our opinion, the proposed method provides the most balanced trade-off between the conflicting nature F-measure and the computation time t, making it suitable for practical applications.

5. CONCLUSION

In this paper, we propose a method that realizes fast and accurate ellipse detection. Splitting a curve into arcs and classifying arcs into the four quadrants based on the three geometric constraints based on curvature, concavity-convexity, and consensus of ellipse centers not only improves the detection accuracy but also reduces the detection time. Compared with five state-of-the-art methods, our method has significant advantages in terms of both the F-measure and the computation time. It shows good performance on complex synthetic images and real images while providing a balanced trade-off between detection accuracy and execution time.

6. REFERENCES

[1] S. Zafari, T. Eerola, J. Sampo, H. Kälviäinen, and H. Haario, "Segmentation of overlapping elliptical objects in silhouette images," IEEE Transactions on Image Processing, vol. 24, pp. 5942-5952, 2015.

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

[3] D. S. Barwick, "Very fast best-fit circular and elliptical boundaries by chord data," IEEE transactions on pattern analysis and machine intelligence, vol. 31, pp. 1147-1152, 2009.

[4] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," Communications of the ACM, vol. 15, pp. 11-15, 1972.

[5] N. Guil and E. L. Zapata, "Lower order circle and ellipse Hough transform," Pattern Recognition, vol. 30, pp. 1729-1744, 1997.

[6] A. A. Sewisy and F. Leberl, "Detection ellipses by finding lines of symmetry in the images via an hough transform applied to straight lines," Image and Vision computing, vol. 19, pp. 857-866, 2001.

[7] S.-C. Zhang and Z.-Q. Liu, "A robust, real-time ellipse detector," Pattern Recognition, vol. 38, pp. 273-287, 2005.

[8] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: randomized Hough transform (RHT)," Pattern recognition letters, vol. 11, pp. 331-338, 1990.

[9] N. Kiryati, Y. Eldar, and A. M. Bruckstein, "A probabilistic Hough transform," Pattern recognition, vol. 24, pp. 303-316, 1991.

[10] Q. Ji and R. M. Haralick, "A statistically efficient method for ellipse detection," in Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, 1999, pp. 730-734.

[11] P. L. Rosin and G. A. W. West, "Nonparametric segmentation of curves into various representations," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, pp. 1140-1153, 1995.

[12] E. Kim, M. Haseyama, and H. Kitajima, "Fast and robust ellipse extraction from complicated images," in Proceedings of IEEE information technology and applications, 2002.

[13] D. K. Prasad, M. K. Leung, and C. Quek, "ElliFit: An unconstrained, non-iterative, least squares based geometric Ellipse Fitting method," Pattern Recognition, vol. 46, pp. 1449-1465, 2013.

[14] M. Fornaciari, A. Prati, and R. Cucchiara, "A fast and effective ellipse detector for embedded vision applications," Pattern Recognition, vol. 47, pp. 3693-3708, 2014.

[15] Y. Xie and Q. Ji, "A new efficient ellipse detection method," in Pattern Recognition, 2002. Proceedings. 16th International Conference on, 2002, pp. 957-960.

[16] S. Mulleti and C. S. Seelamantula, "Ellipse fitting using the finite rate of innovation sampling principle," IEEE Transactions on Image Processing, vol. 25, pp. 1451-1464, 2016.

[17] D. K. Prasad, M. K. Leung, and S.-Y. Cho, "Edge curvature and convexity based ellipse detection method," Pattern Recognition, vol. 45, pp. 3204-3221, 2012.

[18] J. Canny, "A computational approach to edge detection," IEEE Transactions on pattern analysis and machine intelligence, pp. 679-698, 1986.

[19] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," Computer graphics and image processing, vol. 1, pp. 244-256, 1972.

[20] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," Cartographica: The International Journal for Geographic Information and Geovisualization, vol. 10, pp. 112-122, 1973.

[21] H. Freeman and R. Shapira, "Determining the minimum-area encasing rectangle for an arbitrary closed curve," Communications of the ACM, vol. 18, pp. 409-413, 1975.

[22] A. S. Aguado, M. E. Montiel, and M. S. Nixon, "On using directional information for parameter space decomposition in ellipse detection," Pattern Recognition, vol. 29, pp. 369-381, 1996.

[23] A. Fernandes, "A correct set of equations for the real-time ellipse hough transform algorithm," Technical Report2009.

[24] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category database," ed: Technical Report, Available:{http://authors. library. caltech. edu/7694}, 2016.

[25] F. Mai, Y. Hung, H. Zhong, and W. Sze, "A hierarchical approach for fast and robust ellipse extraction," Pattern Recognition, vol. 41, pp. 2512-2524, 2008.

[26] X. Bai, C. Sun, and F. Zhou, "Splitting touching cells based on concave points and ellipse fitting," Pattern recognition, vol. 42, pp. 2434-2446, 2009.

[27] Z.-Y. Liu and H. Qiao, "Multiple ellipses detection in noisy environments: A hierarchical approach," Pattern Recognition, vol. 42, pp. 2421-2433, 2009.