Elliptical Object Detection by a Modified RANSAC with Sampling Constraint from Boundary Curves’ Clustering

Yingdi XIE†(a), Student Member and Jun OHYA†, Member

SUMMARY This paper proposes a method for detecting ellipses from an image despite (1) multiple colors within the ellipses, (2) partially occluded ellipses’ boundaries, (3) noisy, locally deformed boundaries of ellipses, (4) presence of multiple objects other than the ellipses in the image, and (5) combinations of (1) through (4). After boundary curves are obtained by edge detection, by utilizing the first-order difference curves of the edge orientation of each pixel in the boundary curves, a segment-reconnect method obtains boundary clusters. Then, a modified RANSAC detects ellipses by choosing five pixels randomly from the boundary clusters, where overlapped ellipses are merged. Experimental results using synthesized images and real images demonstrate the effectiveness of the proposed method together with comparison with the Randomized Hough Transform, a well-known conventional method.

key words: ellipse detection, a modified RANSAC

1. Introduction

Among shape primitives, the circle has drawn special interest throughout history due to its perfectness as a morphological feature. Many natural objects have a circular appearance, from an astronomical scale, such as planets, to a micro scale, such as biological cells. Due to not only the appearance but also the philosophical beauty of circles, numerous artificial objects are designed to have an exterior that is circular or, in variation, elliptical. Spherical objects are appearance-invariant with respect to different viewpoints, while 2D circular objects deform with affine transformation so that their contours are observed as ellipses. Other objects with either partially or entirely intrinsic elliptical boundaries are also very common; thus, detecting ellipses from an image is of fundamental importance in pattern recognition topics.

The ellipse-fitting problem has drawn research interest for several decades. As computing hardware and theories have improved, in recent years a wide range of ellipse-estimation applications, including defect detection for industrial products [24], ellipse extraction from remote sensing images [5], [14], [39], biological and biomedical analysis [2], [12], [18], intelligent transportation systems [25], [32], human pupil monitoring [17], human face recognition and tracking [7], [10], [22], [38], [40], video surveillance [27], [35], [36], and others [37], has been achieved. In addition, since the ellipse is a fundamental shape primitive, ellipse detection can facilitate more advanced object recognition tasks by providing significant appearance features. Fruitful applications as well as the long-term research interest prove not only the practical importance but also the challenging nature of ellipse detection research. Major difficulties lie in both the high parametric dimension of ellipse fitting and the interference from either imagery noise or complex edge spatial distribution with respect to different scene structures.

Of the previous literature on ellipse fitting, the least square-based fitting method and the Hough transform family are the most commonly researched. The least square method is susceptible to the influence of outliers or noise and extraneous pixels. Aiming to improve the extraction accuracy, W. Zou [40] proposed a quadrant arc (against the boundary gradient) combination method that achieves real-time ellipse tracking. A.Y.S. Chia et al. [4] improved the curve segmenting and merging steps by exploring the theories of Gestalt psychology before applying the least square method. This proposal can achieve accurate ellipse fitting even with severe partial occlusion of the ellipses by each other. However, the potential inaccuracy during edge curve merging will result in false ellipse detection. The Hough transform (HT) family, including standard HT, combinatorial HT (CHT) [28], Randomized HT (RHT) [33], Probabilistic HT (PHT) [15], and Iterative Randomized HT (IRHT) [18], achieves model fitting through voting and seeking the local maximal in elliptical parameter space according to the combination of edge pixels. However, an ellipse-fitting problem requires in general a 5D parameter space since an elliptical shape can be specified by five parameters; this results in an extreme computational load both in running time and storage space. Although methods of HT extension such as those in Refs. [11], [30],[34] have been proposed to decrease the sampling size, no satisfactory results have been achieved with a good balance between computation cost and detection accuracy.

To facilitate efficient computation, methods such as “Uprite” [21] and “arc finding” [13],[19],[26] have been proposed. Nevertheless, both methods have their individual defects. Specifically, “Uprite” can only detect ellipses with no occlusion, while “arc finding” is dependent on the preciseness of the detected contours, which results in low detection accuracy when applying to the image where object contours are complicatedly crossed. In Ref. [16], Z. Liu and H. Qiao proposed a hierarchical detection framework to address the accuracy problem for noisy images. However, the proposed algorithm can only handle elliptical contours with...
no seamless connections to other edges. A distinct method of using the multi-population genetic algorithm (MPGA) was proposed by N. Kharma to detect ellipses in biomedical images [12]. However, it fails to detect ellipses when the boundaries of the target objects are broken due to either occlusion by other objects or to vague edges within the ellipses.

In recent years, some works on geometrical analysis of the boundary curves that could correspond to borders between objects can be seen. K. Hahn et al. proposed a contour segment and merge framework to cluster edge pixels so that the random sampling process in RHT is constrained to each cluster for efficient ellipse fitting [9]. However, since their curve segmentation and merge steps cannot avoid generating noisy and false connections of multiple clusters, the ellipse fitting by RHT is inaccurate and inefficient.

In this paper, we further develop the geometrical analysis of boundary curves to pursue better robustness and lower computational load. The purpose of this work is to detect ellipses from an image despite (1) multiple colors within the ellipses, (2) partially occluded ellipse boundaries, (3) local deformations of ellipse boundaries, (4) the presence of multiple objects other than the ellipses in the image, and (5) combinations of (1) through (4). The proposed method is based on applying a modified random sample consensus (RANSAC) [31] to estimate the five parameters of the ellipse. Since the modified RANSAC randomly chooses five points (edge pixels), a choice from different ellipses causes incorrect and/or inefficient estimation. To solve this problem, our proposed method clusters boundary pixels by segmenting raw boundary curves obtained by computing edge orientations into curvelets and by reconnecting the curvelets in accordance with geometrical and morphological criterions. The modified RANSAC is then performed on each boundary cluster (reconnected curves).

The rest of this paper is organized as follows. Section 2 describes the basic concept of the proposed method. Section 3 elaborates on the algorithm for obtaining boundary clusters from raw boundary pixels obtained by edge detection. Section 4 explains the modified RANSAC-based procedure for detecting ellipses. Section 5 presents the experimental results and discussion. Section 6 concludes this paper.

2. Basic Idea

The motivation of the proposed method is to solve major difficult issues in ellipse detection. Specifically, the difficult issues are classified into the five cases shown in Fig. 1: (1) there are multiple-colored areas or textured regions within one elliptical object. This case results in detecting too many noisy edge pixels, which interferes with accurate ellipse detection. (2) An ellipse contour is partially occluded. In this case, the ellipse contour is segmented into multiple parts. (3) An ellipse contour is locally deformed, which causes miss detection. (4) Other than ellipses, there are many objects with different shapes. This situation is a difficult case for random sampling based methods such as the randomized Hough transform. (5) The cases (1) through (4) are combined. Most of the conventional methods cannot deal with this case.

The proposed method aims at solving all the above-mentioned five issues. The proposed method is outlined in Fig. 2. We first apply canny operator [2] to the in-
put image. The proposed method is based on a modified RANSAC [7], which constructs ellipse models by randomly choosing five boundary (edge) pixels. However the random choice does not always pick up five edge pixels from a same elliptical segment in case of (1), (2), (4) and (5) in Fig. 1. This not only lowers the efficiency of the detection process, but also causes detection errors. To avoid this problem, we constrain the random choice to a boundary cluster, which corresponds to an elliptical arc.

As shown in Fig. 2, after detecting edge pixels, we obtain boundary clusters by the processes described in Sect. 3. Then, ellipses are detected from the boundary clusters by a modified RANSAC based processes explained in Sect. 4.

3. Boundary Cluster

3.1 Basic Strategy

As shown in Fig. 3, we cluster the boundary (edge) pixels so that the random choice of five edge pixels for a modified RANSAC is constrained to a boundary cluster. The first row of Fig. 3 indicates raw boundary curves obtained by the edge detection process. In the second row, we eliminate the intersection points before computing edge orientation at each boundary pixel, so that the computational load of edge orientation can be decreased. By computing the orientation of each edge pixel in a non-intersection curve, we obtain one edge orientation curve (in the following, “EOC”) for that non-intersection curve as its morphological feature. Then, as shown in the third row, we segment a non-intersection curve into curvelets at the pixels that correspond to discontinuities in the EOC, because an ideal ellipse’s contour is continuous with respect to its first-order derivative. Finally, in the fourth row, the curvelets are reconnected according to geometrical and morphological criterions so that a boundary cluster is obtained. The rest of this section elaborates on the calculation of EOC (Sect. 3.2), segmentation of non-intersection curves into curvelets (Sect. 3.3), and reconnection of curvelets (Sect. 3.4).

3.2 Edge Orientation Curve

Suppose that the x and y denote the horizontal (x) and vertical (y) coordinates of the image’s x-y coordinate system, respectively. At each boundary pixel \( p_i \) in a non-intersection curve, edge orientation \( \theta_i \) can be simply calculated from pixel based first order derivatives (gray-level differences between two adjacent pixels) for the x and y directions, but this computation method is not so robust against noise components that sufficient accuracies for the subsequent EOC based computations are not obtained. Therefore, \( \theta_i \) should be calculated from a block \( \phi_i \) centered at the pixel \( p_i \) by the following equation:

\[
\theta_i = \arctan \left( \frac{\sum_{(x,y) \in \phi_i} \frac{\partial I(x,y)}{\partial y}}{\sum_{(x,y) \in \phi_i} \frac{\partial I(x,y)}{\partial x}} \right)
\]  

(1)

where, \( I(x,y) \) is the intensity value at \((x,y)\). This computation is often achieved within the gray scale image, against which we can obtain not only the edge orientation but also the normal’s magnitude. However, as shown in Fig. 4, its sensitivity to intensity value in the block \( \phi_i \) causes undesirable detected orientations’ variation for \( \theta_i \); that is, in Fig. 4, although the shape of the boundaries is the same, the computation results of the edge orientation are different, depending on the gray-scale distribution.

We thus pursue an intensity invariant edge orientation by the following adaptive block size scheme. The basic strategy is to adapt the block size so that the following conditions are satisfied.

1. \( p_i \) is not an endpoint.
2. The block contains one boundary curve that passes through \( p_i \), where the length of the boundary curve is not very short.
3. The boundary curve described in (2) touches the block border at two points \( t_1 \) and \( t_2 \).
4. The lengths between \( t_1 \) and \( t_2 \) along the block border in the clockwise and counterclockwise directions are almost equal.
5. The two points \( t_1 \) and \( t_2 \) are found by tracing \( p_i \)'s boundary curve from \( p_i \) in the clockwise (counterclockwise) and counterclockwise (clockwise) directions, respectively, without encountering \( t_2 \) during the search for \( t_1 \) and \( t_1 \) during the search for \( t_2 \).

In case that all of the conditions (1) through (5) are satisfied
as shown in Fig. 5 (a), the block is divided into two areas by the boundary curve. Then, the smaller area in the block is filled by the same value as that of \( p_i \) (Fig. 5 (b)), and the edge orientation \( \theta_i \) is computed by Eq. (1).

Since all the boundary pixels do not satisfy the five conditions, such a case also needs to be dealt with. Therefore, the specific algorithm for computing the edge orientation \( \theta_i \) is as follows. In the initial block with the largest size (in this paper 11 by 11 pixels), if boundary curves that do not pass through \( p_i \) exist in the block (e.g. the upper most curve in Fig. 5 (c)), such curves are eliminated. If the boundary curve to which \( p_i \) belongs is shorter than a predefined threshold (2 pixels in this paper), that curve is eliminated, and “\( NaN \)” (Not a Number) is returned as the value of \( \theta_i \); otherwise, whether \( p_i \) is an endpoint or not is checked. If \( p_i \) is an endpoint, \( \theta_i \) is computed by the following equation:

\[
\theta_i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \tag{2}
\]

where \( x_i \) and \( y_i \) denote the \( x \) and \( y \) coordinates of \( p_i \), respectively, and \( x_{i-1} \) and \( y_{i-1} \) denote the \( x \) and \( y \) coordinates of the adjacent boundary pixel of \( p_i \), respectively. If \( p_i \) is not an endpoint, whether all of the conditions (3) through (5) are satisfied or not is checked. If it is satisfied, the computation of the edge orientation \( \theta_i \) is same as the above-mentioned computation method described just after the conditions (1) through (5); otherwise, the block size is reduced (in this paper, if the block size is greater than 3 pixels, it is reduced by 2 pixels; otherwise, \( NaN \) is returned.).

The dotted block in Fig. 5 (d) is an example against condition (5). That is, \( t'_2 \) is found by the clockwise search from \( p_i \) along \( p_i \)'s boundary curve, but \( t'_2 \) is not found by the counterclockwise search from \( p_i \), because the counterclockwise search from \( p_i \) encounters the discontinuity of \( p_i \)'s curve. In case of the smaller blocks than the initial, largest block, even if (a) part(s) of \( p_i \)'s curve get(s) out of the block, our proposed method preserves that curve’s connectivity or disconnectivity in the initial block. Therefore, in Fig. 5 (d)'s dotted block, \( t'_1 \) is found also by the clockwise search from \( p_i \) only after the search encounters \( t'_2 \); so that this case does not satisfy condition (5).

To keep the obtained edge orientation consistency over the adjacent boundary pixels, we assign the edge orientation value to either \( \pi \) or \(-\pi \) if the orientation difference between adjacent pixels is larger than \( 3\pi/4 \) or smaller than \(-3\pi/4 \), respectively. As a result of computing the edge orientations at each pixel in each non-intersection curves by the algorithm described in Sect. 3.3, EOC is obtained.

### 3.3 Segment the Raw EOC

As described in Sect. 3.1, non-intersection curves are segmented into curvelets by utilizing the EOC obtained from the non-intersection curves. We can consider that at every point of ellipse contours the C-1 continuous property holds; in other words, we should segment non-intersection curves at the discontinuous points, at which the C-1 property breaks. Discontinuous points can be detected by utilizing EOC for the non-intersection curves to be segmented; more specifically, discontinuities are detected as peaks in the absolute value of the first order difference’s (derivative’s) curve of the EOC. Here, before detecting peaks we adjust the raw EOC so that the orientation difference between neighbor pixels whose angels are near \( \pi \) and \(-\pi \) does not cause false detection. The EOC and its absolute value of the first order difference curve for the non-intersection curve in Fig. 6 (a) are shown in Fig. 6 (c) and (d), respectively. As shown in Fig. 6 (d), the discontinuities are detected as the peaks that exceed the predefined threshold; more accurately, the peak is defined as the point with the largest value in the range that exceeds the threshold. The range between two adjacent detected discontinuities corresponds to a curvelet.

In Fig. 6, each curvelet is given a color different from the other curvelets’ colors.

As shown in Fig. 6 (b), (e) and (f), different values for the threshold for the first-order difference’s curve give different segmentation results. In particular, over-segmentation could be problematic for the subsequent processes, but in case of our method, as described in Sect. 3.4, over-segmented curvelets could be reconnected. Therefore, our method’s robustness against the choice of the threshold value can be proved.
Fig. 6 Non-intersection curve segmentation into curvelets. (a) and (b): Original curves and segmentation results, (c) and (e): EOC, (d) and (f): the absolute value of EOC’s first-order difference curve.

3.4 Reconnecting Curvelets

3.4.1 Strategy

Segmented curvelets tend to be spatially separated due to different causes such as partial occlusion or over segmentation even though the curvelets belong to the same elliptical arc. This lowers the efficiency and accuracy for the modified RANSAC based ellipse detection process. Therefore, our method reconnects curvelets by the following algorithm.

All the endpoints of curvelets are candidates to be reconnected. Assume that there are $T$ endpoints, a similarity matrix $A_{ij}$ ($i = 1, \ldots, T, j = 1, \ldots, T$) is constructed for seeking the reconnection pair of the endpoints. The similarity matrix’s element $a_{ij}$ is computed from the following four weights between the terminal points $i$ and $j$: (1) distance weight; (2) morphological similarity weight; (3) terminal’s edge orientation similarity; (4) similarity of average EOC difference. The specific computation of each element $a_{ij}$ is described in Sect. 3.4.3 after the four weights are explained. In the following, we call the similarity matrix’s vertical and horizontal elements “primary” and “candidate”, respectively, because for each primary, the connectable candidate(s) is (are) sought.

The weights (1) through (3) are calculated with respect to the primary’s inscribed circle, which are treated as an extrapolation to the primary. This strategy is based on the fact that spatially close endpoint pairs should not always be connected: e.g. in Fig. 7, the curvelet $\Gamma_1$’s endpoint (primary) $P$ is closer to candidate $C_2$ than $C_1$, but $C_1$ is more coherent with $C_2$ than $C_1$ with respect to elliptical connectivity. The connectivity is predicted by inscribed circle instead of ellipse because of considering the computational efficiency. Thus, $C_1$ is connected to $P$.

The inscribed circle center $(u, v)$ and radius $r$ are computed from the following equations:

\[
\begin{align*}
    r_{\Gamma_1} &= \frac{1}{S'} \sum_{s \in S_{\Gamma_1}'} f'_{\Gamma_1}(s) \\
    u_{\Gamma_1} &= x_{\Gamma_1}^1 - r_{\Gamma_1} \cdot \cos (f'_{\Gamma_1}(x_{\Gamma_1}^1, y_{\Gamma_1}^1)) \\
    v_{\Gamma_1} &= y_{\Gamma_1}^1 - r_{\Gamma_1} \cdot \sin (f'_{\Gamma_1}(x_{\Gamma_1}^1, y_{\Gamma_1}^1))
\end{align*}
\]

where, $f'(s)$ the first derivative of the edge orientation at $s$; $S'$ the curve length of $\Gamma_1'$; and $\Gamma_1'$ is the C-0 continuous curvelet part of $\Gamma_1$, which starts from the primary endpoint and ends at the point where the continuity of the edge orientation’s first difference breaks, typically in our experiment we set a threshold for judging the C-0 discontinuity as $\pi/128$.

3.4.2 Four Weights

(1) Distance Weight:

To evaluate how much the candidate fits to the inscribed circle, the Distance Weight is computed by the following equation:

\[
    w_{dist} = \begin{cases} 
        0 & \text{if } 1 - \frac{d}{min(D, A + B)} < 0 \\
        1 & \text{else} 
    \end{cases}
\]
where $d$ is the shortest distance from the candidate to the inscribed circle, $D$ is a preset maximum distance, $l$ is the distance from the candidate to the primary along the inscribed circle, and $A$ and $B$ are parameters, which are explained below. The denominator of the second term of the right side of Eq. (6) manifests that if the candidate is connected to the primary, then the candidate should fall in the range that is gradually widened as the distance between the primary and candidate gets large, where $AI + B$ corresponds to the range’s width, and $D$ is the range’s maximal width.

(2) Morphological Similarity Weight:
The morphological similarity between the primary curvelet $Γ_1$ and the candidate curvelet $Γ_2$ can be computed accurately according to some shape descriptors, such as Fourier descriptor. However, to reduce the computation load, the morphological similarity is computed with respect to $Γ_2$ and the inscribed circle by the following equation:

$$w_{\text{morphSim}} = \frac{1}{s-1} \sum_{s \in Γ_2} \left(1 - \frac{2 \cdot \theta_s}{\pi}\right)$$

(7)

where, $s$ is a pixel of the candidate curvelet $Γ_2$, $S$ is the length of $Γ_2$, $θ_s$ is defined as:

$$θ_s = \arctan\left(\frac{d_{s+1} - d_s}{l_{s+1} - l_s}\right)$$

(8)

where $d$ and $l$ are defined in the distance weight. The greater the weight is, the higher the similarity between the candidate curvelet and the inscribed circle is.

(3) Terminal’s Edge Orientation Weight:
This weight measures the orientation similarity between the terminal points of the primary and candidate, because considering the procedure for obtaining curvelets, terminal points might have a distinct edge orientation. Thus, this weight is computed as the average orientation of the curvelets’ ranges near the primary and candidate terminal points by the following equation.

$$w_{\text{eosSim}} = 1 - \frac{1}{\pi} \left(\frac{1}{s_{Γ_1}} \sum_{s \in Γ_1} f'(s) + \frac{1}{r_{Γ_2}} \sum_{r \in Γ_2} f'(r)\right)$$

(9)

where, $f(s)$ is the edge orientation at $s$, $Γ_1'$ and $Γ_1''$ are the ranges near the terminal points in the primary and candidate curvelets, respectively, $S_{Γ_1}$ and $S_{Γ_2}$ are the lengths of $Γ_1'$ and $Γ_2''$, and $r_{Γ_1}$ is the radius of $Γ_1$’s inscribed circle.

(4) Similarity Weight of the Average EOC Difference:
This weight measures the similarity between the first order difference of each curvelets’ EOC; more specifically, the difference between the average EOC difference of the primary and candidate curvelets is computed by the following equation.

$$w_{\text{diffEOC}} = 1 - \frac{1}{\pi} \left(\frac{1}{s_{Γ_1}} \sum_{s \in Γ_1} f'_{Γ_1}(s) - \frac{1}{s_{Γ_2}} \sum_{s \in Γ_2} f'_{Γ_2}(s)\right)$$

(10)

where, $f'(s)$ is the first order difference of EOC. In Eq. (10), the maximum of the average EOC difference as well as the terminal edge orientation difference is assumed as $π$.

3.4.3 Reconnection
Among the four weights (1) through (4), the distance weight (1) is considered to be the absolute condition for reconnection; that is, the curvelets far away from the primary curvelet are eliminated. Thus the overall reconnection similarity is computed as:

$$w = \begin{cases} \sum_{\alpha_{ii}} w_i & \text{for } w_{\text{dist}} \neq 0 \\ 0 & \text{for } w_{\text{dist}} = 0 \end{cases}$$

(11)

where, $w_i (i \in [1, 4])$ is the four weight components; $α_i$ is each weight component’s importance (in our experiment we set 0.4, 0.3, 0.15, and 0.15 respectively).

From the computed weight for each terminal point pair, we can obtain the value for each element of a reconnection similarity matrix $α_{ij}$. Notice that the original reconnection similarity matrix is not symmetrical; i.e. if the primary and candidate are swapped; the computed weights are generally different, because the inscribed circles are generally different. In order to achieve the bilateral relationship for the two terminal points, we add the transposed matrix to the original matrix. In the symmetrical similarity matrix, a local maximum is pursued for each row and column to obtain the reconnection candidate pairs. If the weight value exceeds a predefined threshold, the candidate pairs are reconnected.

3.5 Ellipse Candidate
The segment-reconnect processing improves the boundary curves’ morphological feature. We further perform ellipse fitting to the ellipse candidates rather than all the reconnected curves, because in natural images some boundaries are obviously non-elliptical shape. In particular, straight lines should be eliminated. In general, the variations of the orientations of boundary pixels on a straight line tend to be smaller than those for elliptical arcs. Thus, ellipse candidates are detected as the curves whose edge orientations’ range is wider than a pre-defined threshold: in this paper, the threshold is 0.2, of the entire angle ranging from $-π$ to $π$, which is the orientation angles’ range for a full circle and ellipse. The procedure of selecting the ellipse candidate filters out other shaped curvelets such as straight lines, but it is insufficient to conclude the shape of ellipse by only measuring the angle range of the curves’ edge orientation. Thus we apply a modified RANSAC process to pursue precise ellipse estimation.
4. Ellipse Parameterization via a Modified RANSAC

4.1 Estimation of Ellipse Parameters

As detailed below, the ellipse is described by five parameters, which means the five parameters can be determined by five pixels’ positions. The processes described in Sect. 3 constrain the choice of the five pixels to each boundary cluster. However, those processes cannot fully avoid the over-connection of boundary curves, which should be separated. The over-connection causes areas including many complicatedly distributed curves. In such an area, the modified RANSAC that we propose can detect multiple ellipses from a boundary curve cluster accurately and efficiently as opposed to the traditional RANSAC. The algorithm for estimating the parameters are shown in Fig. 8.

Concerning the ellipse’s parameters, we utilize the general conic equation in Eq. (12) rather than the standard ellipse equation in Eq. (13) for computational convenience.

\[ Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \]  \hspace{1cm} (12)

\[ \left(\frac{x \cos \theta + y \sin \theta - h}{u}\right)^2 + \left(\frac{-x \sin \theta + y \cos \theta - k}{v}\right)^2 = 1 \]  \hspace{1cm} (13)

where, \( u \) the long axis, \( v \) the short axis, \((h,k)\) the center position, and \( \theta \) the rotation angle.

The ellipse parameters are obtained as the coefficients in the general conic form in Eq. (12), and then the coefficients are converted to the parameters in the standard form in Eq. (13).

Figure 8 illustrates the modified RANSAC based process for estimating the ellipse parameters from one boundary cluster. From the boundary cluster, five pixels are randomly chosen, and the ellipse model (parameters) is constructed. Then, the validity of the obtained model is evaluated by the modified RANSAC. Traditional RANSAC counts the number of inliers to verify the model’s validity. However, thresholding the inlier number does not always give a reasonable result. That is, the traditional RANSAC’s ellipse fitting is unstable in that multiple ellipse models could be fitted to same inliers; thereby, the accuracy of the ellipse fitting is not very high. The traditional RANSAC’s another unstable case could occur in areas in which boundary pixels are densely distributed, because within such an area, rotations and/or translations of the ellipse do not result in significant changes in the number of inliers; thereby, multiple ellipse models could fit to those boundary pixels. Our modified RANSAC adopts a weight function indicated by Eq. (14), as opposed to the inlier counting, to represent how well the model fits to the boundary pixels.

\[ w^i_{\text{MRANSAC}} = \frac{1}{N} \sum_{i=1}^{N} k\left(f_d(p_i, \hat{\psi}^j)\right) \]  \hspace{1cm} (14)

where \( j \) denotes the ellipse model’s index, the function \( f_d(\cdot) \) computes the minimal distance from the

...
that the ellipse is deformed to a unit circle.

\[
\begin{bmatrix}
x_T \\
y_T \\
1
\end{bmatrix} = 
\begin{bmatrix}
\cos \theta^i - \sin \theta^i & 0 & 1/\mu^i & 0 & 0 & 1 & 0 & -h^i & x \\
\sin \theta^i & \cos \theta^i & 0 & 1/\nu^i & 0 & 0 & 1 & -k^i & y \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\] (17)

On the unit circle, the point “p_c” that is closest to point \( p \) is found. Then, the minimal distance can be calculated as the distance between \( p \) and the reversed affine transformed \( p_c \).

Equations (13) and (14) indicate that the boundary pixels far away from the ellipse (model) do not affect the weight in Eq. (14). That is, if enough number of boundary pixels is close to the model, the value of the weight gets large and exceeds a pre-defined threshold. Then the model is recorded. Given a large value for the pre-defined threshold for the weight in Eq. (14), the modified RANSAC outputs an ellipse model that fits to the boundary pixels more accurately than the traditional RANSAC. The inliers, which are the boundary pixels close to the model, are masked out, because the inliers do not need to be used for another ellipse’ parameter estimation. More specifically, a boundary pixel is judged as an inlier if that pixel’s \( f_d(\cdot) \) value is smaller than a pre-defined threshold. If the random choice of five pixels is repeated more times than a pre-defined threshold, we terminate the computation for the current cluster by outputting the detected ellipses; otherwise, the process returns to the random choice of five pixels.

Note that the process in Fig. 8 is performed for all the boundary clusters’ boundary pixels that have not yet been masked out. Therefore, after the final result of detecting ellipses from a boundary cluster is obtained, the process in Fig. 8 is performed for another boundary cluster, which has not yet been processed.

4.2 Detection Optimization

All the valid ellipse models are optimized before obtaining the final detection result, because multiple valid models could be duplicated. Causes for this duplication include multiple sets of five pixels (1) from a same boundary cluster and (2) from different boundary clusters that correspond to a same elliptical arc. Thus, valid models with small parameter differences should be merged into one model. More specifically, two ellipses whose parameter value difference is smaller than the threshold are merged, where the specific merge conditions for each parameter are listed in Table 1.

In each of the four inequalities in Table 1, the right hand side indicates the adaptive threshold, which is determined according to the specified parameters of the two ellipses to be checked. If all of the four inequalities are satisfied, the two ellipses are integrated; otherwise, not integrated. The constant such as 0.02 and 0.05 in each adaptive threshold is empirically determined, but those constants are effective for all the images shown in the experimental results presented in Sect. 5.

5. Experimental Results and Discussion

5.1 Experimental Condition

The proposed method is applied to synthesized and real images that contain ellipses under different situations. As a whole, 100 synthesized images and 50 real images are used for the experiments. Some of the images are shown in Fig. 9. The specific values of the thresholds used by the proposed method are listed in Table 2.

The threshold for boundary curve segmentation is set to a small value, because although a small value might result in over-segmentation, the subsequent reconnection step can cover this issue. However, a large value for this threshold might cause over-connections, which result in either wrong estimation or non-detection.

Concerning the threshold for curvelet reconnection, a small value might cause over reconnection, while a large value might cause absence in reconnection and result in absence in detection or inaccurate estimation. As an optimal value, we use 0.5 for this threshold.

The iteration number threshold is selected based on the balance between the computational cost and accuracy. A large number of iterations for the modified RANSAC process could raise the detection accuracy at a large cost of computation load, while a small value lowers the detection accuracy. As an optimal value, we use \( n \sim 1.2 \) for this threshold.

In the rest of this Section, the effectiveness and efficiency of the ellipse detection by the proposed method despite the difficult issues shown in Fig. 1 is demonstrated experimentally, together with comparison with a typical conventional method, Randomized Hough Transform.

5.2 Ellipse Detection

Experiments on synthesized images and natural images are carried out, as shown in Fig. 9. The ellipse detection’s difficulties shown in Fig. 1 (1) through (3), except for many objects’ cases (Fig. 1’s (4) and (5)), are furthermore classified into the seven cases shown in Fig. 9: (a) multiple ellipses

| Parameter               | Merge Conditions                                                                 |
|-------------------------|----------------------------------------------------------------------------------|
| Ellipse center          | \( \sqrt{(h_1 + h_2)^2 + (k_1 + k_2)^2} \leq 0.02 \cdot \sqrt{uv} \)            |
| Major Axis \( u \)      | \(|u_1 - u_2| \leq 0.05 \cdot u\)                                               |
| Minor Axis \( v \)      | \(|v_1 - v_2| \leq 0.05 \cdot v\)                                               |
| Rotation Angle          | \(|\theta_1 - \theta_2| \leq \pi e^{-20(u-v)}\)                                |

Note: \( h_i, k_i, u_i, v_i, \theta_i \) are the parameters defined in Eq.(13) for the \( i \)th ellipse; \( u \) and \( v \) are the average lengths of the major and minor axes of the two ellipses to be checked, respectively.
are separated, (b) connected to each other, (c) an ellipse is connected with other shape, (d) an ellipse is composed of different colors with boundary noise, (e) and (f) partially occluded ellipse that is composed of different colors, and (g) multiple ellipses with different colors inside are overlapped. Experiments on detecting ellipses from synthesized images that correspond to the seven cases shown in Fig. 9’s (a) through (g) are first carried out. Boundaries obtained by edge detection, and boundary clusters obtained by the segment and reconnect processes are shown in the second and third rows in Fig. 9, respectively. In the fourth row in Fig. 9, detected ellipses are indicated by red lines. These results demonstrate that the proposed method can accurately detect ellipses despite each difficulty.

As shown in Fig. 10 and Fig. 11, experiments on real images are conducted. In Figs. 10 and 11, the panel (a)’s first and second rows show the original and edge images, respectively; the red lines in the panel (b)’s first and second rows show ellipse candidates obtained by the segment-reconnect procedure described in Sect. 3, and the final results of the ellipse fitting by the procedure described in Sect. 4, respectively.

In Fig. 10, (1) and (2) are simple cases; (3) to (6) correspond to the difficult issue (1) in Fig. 1; (7) to (9) correspond to (2) in Fig. 1; (10) and (11) correspond to (3) in Fig. 1; (12) to (14) correspond to (4) in Fig. 1; (15) to (17) correspond to (5) in Fig. 1. The detection results achieve satisfying detection accuracy against human being’s perceptual understanding. In Fig. 11, very many ellipses with complicated edge distributions are included. Although most of the conventional methods such as RHT cannot detect any ellipse from those images, the proposed method detects ellipses successfully.

**5.3 Performance Comparison and Evaluation**

This paper experimentally compares the proposed method with the standard Randomized Hough Transform (RHT), one of the most well-known conventional methods, as well as manual detection results. RHT’s algorithm we implemented is as follows.

1. Randomly select five boundary pixels and estimate the ellipse’ parameter.
2. Vote the estimated ellipse model in the five-dimensional parameter space.
3. Iterate step (1) and (2) until satisfying a pre-defined condition.
4. Output all the ellipse models with enough votes.

Detection results by RHT and manual detections are shown by red lines in the first and second rows in the panel (c) in Fig. 10 and Fig. 11, respectively. Facing complicated query images, RHT tends to result in fail detection because of the difficulty in vote threshold selection as well as the extreme computational inefficiency. Overall, the detection accuracy of our method is proved to outperform the RHT.

The results by manual detection are the ground truths for each image. As can be seen in Figs. 10 and 11, the ellipse detection results reasonably fit well to the manual detections, despite the small proportion of detection failure, which is reasoned by improper thresholding for the curvelet reconnection.
Fig. 10  Experimental results against synthesized images and comparison with the result by RHT based method, and human labeling result. As demonstrated, along with the increment of edge complexity, the RHT based method tends to achieve failure detection, either missing or wrong positioning, while our proposal remains preferable robustness.
6. Conclusion

This paper has proposed a method for detecting ellipses from an image under difficult conditions, which the conventional methods cannot deal with. More specifically, the proposed method can deal with the following five difficult issues: (1) multiple colors within the ellipses, (2) partially occluded ellipses' boundaries, (3) noisy, locally deformed boundaries of ellipses, (4) presence of multiple objects other than the ellipses in the image, and (5) combinations of (1) through (4). The proposed method first obtains raw boundary curves by edge detection. By utilizing the first-order difference curves of the edge orientation of each pixel in boundary curves, a segment-reconnect method obtains boundary clusters, which constrain the random sampling by the modified RANSAC. Then, ellipses are detected by the modified RANSAC from five pixels chosen from the boundary clusters randomly, and overlapped ellipses are merged.

Experiments using synthesized images show the validity of the proposed method against the above-mentioned issues (1) through (3). Experiments using real images demonstrate the proposed method's better performances than a well-known conventional method, Randomized Hough Transform, with respect to the above-mentioned five issues. The proposed method's ellipse detection fits well to the manually detected ellipses.
Although the proposed method is advanced in detection performance, one of its major constraints lies in edge detection. The performance of edge detection significantly affects the performance of the subsequent processes. Thus in our future work, we will research better edge detection method, e.g. [1], [20], seeking also for efficiency of the computation.

References

[1] N. Adluru and L.J. Latecki, “Contour grouping based on contour-skeleton duality,” JICV, no.83, pp.12–29, 2009.
[2] X.Z. Bai, C.M. Sun, and F.G. Zhou, “Splitting touching cells based on concave points and ellipse fitting,” Pattern Recognit., vol.42, no.11, pp.2434–2446, Nov. 2009.
[3] J. Canny, “A computational approach to edge detection,” IEEE TPAMI, vol.8, no.6, pp.679–714, 1986.
[4] A.Y.S. Chia, D. Rajan, M.K.H. Leung, and S. Rahardja, “A split and merge based ellipse detector,” Proc. Int’l Conf. on Image Processing, pp.3212–3215, 2008.
[5] M. Ding, Y.F. Cao, and Q.X. Wu, “Autonomous crater detection from planetary images.” Proc. Int’l Conf. on Innovative Computing Information and Control, p.443, June 2008.
[6] D. Eberly, “Distance from a point to an ellipse in 2D,” Geometric Tools, LLC, March 2008.
[7] F. Faux and F. Luthon, “Robust face tracking using color Dempster-Shafer fusion and particle filter,” Proc. Int’l Conf. on Information Fusion, pp.1–7, July 2006.
[8] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” CACM, vol.24, no.6, pp.381–395, June 1981.
[9] K. Hahn, S. Jung, Y. Han, and H. Hahn, “A new algorithm for ellipse detection by curve segments,” Pattern Recognition Letters, vol.29, no.13, pp.1836–1841, 2008.
[10] M.B. Hmidi and Y.B. Jemma, “Fuzzy classification, image segmentation and shape analysis for human face detection,” Proc. Int’l Conf. on Electronics, Circuits and Systems, pp.640–643, Dec. 2006.
[11] K. Kanatani and N. Ohta, “Automatic detection of circular objects by ellipse growing,” Int. J. Image Graphics, vol.4, no.1, pp.35–50, Jan. 2004.
[12] N. Kharma, H. Moghnieh, J. Yao, Y.P. Guo, A. Abu-Baker, J. Laganiere, G. Rouleau, and M. Chret, “Automatic segmentation of cells from microscopic imagery using ellipse detection,” IET Image Processing, vol.1, no.1, pp.39–47, March 2007.
[13] E. Kim, M. Haseyama, and H. Kitajima, “Fast and robust ellipse extraction from complicated images,” Proc. Int’l Conf. on Information Technology and Applications, pp.138–145, 2002.
[14] B. Leroy, G. Medioni, E. Johnson, and L. Matthies, “Crater detection for autonomous landing and astoroids,” Image Vis. Comput., vol.19, pp.787–792, Sept. 2001.
[15] V.F. Levers, “The dynamic generalized hough transform: Its relationship to the probabilistic hough transforms and an application to the current detection of circles and ellipse,” CVGIP: Image Understanding, vol.36, no.3, pp.381–398, 1992.
[16] Z.Y. Liu and H. Qiao, “Multiple ellipses detection in noisy environments: A hierarchical approach,” Pattern Recognit., vol.42, no.11, pp.2421–2433, Nov. 2009.
[17] X.D. Long, O.K. Tonguz, and A. Kiderman, “Real time pupil size monitoring as a screening method for diabetic retinopathy,” Proc. Biomedical Circuits and Systems Conference, pp.215–218, Nov. 2007.
[18] W. Lu and J.L. Tan, “Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT),” Pattern Recognit., vol.41, no.4, pp.1268–1279, April 2008.
[19] F. Mai, Y.S. Hung, H. Zhong, and W.F. Sze, “A hierarchical approach for fast and robust ellipse extraction,” Pattern Recognit., vol.8, no.41, pp.2512–2524, Aug. 2008.
[20] D.R. Martin, C.C. Fowlkes, and J. Malik, “Learning to detect natural image boundaries using local brightness, color, and texture cues,” IEEE TPAMI, vol.26, no.5, pp.530–549, May 2004.
[21] R. McLaughlin, Intelligent algorithms for finding curves and surfaces in real world data, PhD. Thesis, The University of Western Australia, 2000.
[22] H. Nanda and K. Fujimura, “A robust elliptical head tracker,” Proc. Int’l Conf. on Automatic Face and Gesture Recognition 2004, pp.469–474, May 2004.
[23] P.L. Rosin, “Analysing error of fit functions for ellipses,” Pattern Recognition Letters, vol.17, no.14, pp.1461–1470, 1996.
[24] D.C.H. Schleicher and B.G. Zagar, “Image processing to estimate the ellipticity of steel coils using a concentric ellipse fitting algorithm,” Proc. Int’l Conf. on Signal Processing, pp.884–890, Oct. 2008.
[25] A. Soetedjo and K. Yamada, “Fast and robust traffic sign detection,” Proc. Int’l Conf. on Systems, Man and Cybernetics, pp.1341–1346, Oct. 2005.
[26] C. Teutsch, D. Berndt, E. Frotsmann, and M. Weber, “Real-time detection of elliptical shapes for automated object recognition and object tracking,” Proc. Machine Vision Applications in Industrial Inspection XIV, vol.6070, pp.171–179, 2006.
[27] A.L. Troter, R. Bulot, J.M. Boi, and J. Sequeira, “Arc of ellipse detection for video image registration,” Proc. Workshop on Signal Processing Systems Design and Implementation, pp.363–367, Nov. 2005.
[28] D.B. Trvi and M.B. Sandler, “A combinatorial hough transform,” Pattern Recognition Letters, vol.11, no.3, pp.167–174, 1990.
[29] J.P. Wu, “Robust real-time ellipse detection by direct least-square-fitting,” Proc. Int’l Conf. on Computer Science and Software Engineering, vol.1, pp.923–927, Dec. 2008.
[30] Y.J. Xie and Q. Ji, “A new efficient ellipse detection method,” Proc. Int’l Conf. on Pattern Recognition 2002, vol.2, pp.957–960, 2002.
[31] Y.D. Xie and J. Ohyama, “Efficient detection of ellipses from an image by a guided modified RANSAC,” Proc. SPIE Electronic Imaging, 7245A-31, Jan. 2009.
[32] G. Xu, J. Su, H.D. Pan, and D.L. Zhang, “A novel method for wheel rim recognition,” Proc. Int’l Symp. on Electronic Commerce and Security, pp.1051–1054, 2008.
[33] L. Xu and E. Oja, “Further developments on RHT: Basic mechanims, algorithms, and computational complexities,” Proc. Int’l Conf. on Pattern Recognition, vol.1, pp.125–128, Sept. 1992.
[34] H.B. Yu, J.B. Liu, and Z.W. He, “A modified RHT method for ellipse detection based on geometric constraints and perceptual grouping,” Proc. Int’l Conf. on Innovative Computing Information and Control, p.444, 2008.
[35] X.G. Yu, C.S. Xu, Q. Tian, X. Yan, K. Wan, and Z. Jiang, “Estimation of the ball size in broadcast soccer video using salient objects,” Proc. Int’l Joint Conf. on Information, Communications and Signal Processing, and Pacific Rim Conf. on Multimedia, vol.2, pp.15–18, Dec. 2003.
[36] X.G. Yu, H.W. Leong, C.S. Xue, and Q. Tian, “A robust hough-based algorithm for partial ellipse detection in broadcast soccer video,” Proc. Int’l Conf. on Multimedia and Expo 2004, vol.3, pp.1555–1558, June 2004.
[37] G. Zhang, D.S. Jayas, and N.D.G. White, “Separation of touching grain kernels in an image by ellipse fitting algorithm,” Biosystems Engineering, vol.92, no.2, pp.135–142, Oct. 2005.
[38] S.C. Zhang and Z.Q. Liu, “A real-time face detector using ellipse-like features,” Proc. Int’l Conf. on Signal Processing, vol.2, pp.1227–1230, Sept. 2004.
[39] C.R. Zhu and R.S. Wang, “A fast automatic extraction algorithm of elliptic object groups from remote sensing images,” Pattern Recognition Letters, vol.25, no.13, pp.1471–1478, Oct. 2004.
[40] W. Zou, Y. Li, K. Yuan, and D. Xu, “Real-time elliptical head contour detection under arbitrary pose and wide distance range,” J. Vis. Commun. Image Represent., vol.20, no.3, pp.217–228, April 2009.
[41] K. Zimmermann and T. Svoboda, “Approximation of euclidean distance of point from ellipse,” Proc. European Medical & Biological Engineering Conf., Nov. 2005.

Yingdi Xie received the B.S. in Electrical Engineering from Shanghai Maritime University in 2003, and M.S. degrees from Waseda University in 2006. He is now a PhD candidate in Graduate School of Global Information and Telecommunication Studies, Waseda University. His research interests include pattern recognition and image processing.

Jun Ohya received B.S., M.S. and Ph.D. degrees in precision machinery engineering from the University of Tokyo, Japan, in 1977, 1979 and 1988, respectively. He joined NTT Research Laboratories in 1979 and engaged in research on color printers and image processing. He was a visiting research associate at the Computer Vision Laboratory, the University of Maryland, USA, from 1988 to 1989. He transferred to ATR, Kyoto, Japan, in 1992 and became a department head in 1996. Since 2000, he has been a professor at Graduate School of Global Information and Telecommunication Studies, Waseda University, Japan. He was a guest professor at the University of Karlsruhe, Germany, in 2005. His research areas include computer vision, computer graphics and virtual reality. Dr. Ohya is a member of IEEE, IPSJ and VRSJ.