Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Real-Time Ellipse Detection Based on Edge Grouping

Thanh Minh Nguyen, Siddhant Ahuja and Q. M. Jonathan Wu
Department of Electrical and Computer Engineering, University of Windsor
401 Sunset Avenue, Windsor, ON, N9B3P4, Canada
{nguyen1j, ahuja5, jwu}@uwindsor.ca

***Abstract*—In this paper, we present a efficient algorithm for real-time ellipse detection. Unlike Hough transform algorithm that is computationally intense and requires a higher dimensional parameter space, our proposed method reduces the computational complexity significantly, and accurately detects ellipses in real-time. We present a new method of detecting arc-segments from the image, based on the properties of ellipse. We then group the arc-segments into elliptical arcs in order to estimate the parameters of the ellipse, which are calculated using the least-square method. Our method has been tested and implemented on synthetic and real-world images containing both complete and incomplete ellipses. The performance is compared to existing ellipse detection algorithms, demonstrating the robustness, accuracy and effectiveness of our algorithm.**

***Index Terms*—Ellipse detection, real-time, curve segments, edge grouping.**

I. INTRODUCTION

Ellipse detection is one of the key problems in machine vision. The results of ellipse detection are used to infer 3D structures in the image, and camera viewpoints [1]. Furthermore, an ellipse or a circle in 3D space is projected to an ellipse in an image. Therefore, ellipse detection has been widely used in various applications in the field of computer vision, including gaze tracking [2], ball tracking in soccer games [3], vehicle detection [4], cell counting in breast cancer cell samples [5] and traffic sign detection [6], among others. Over the last two decades, researchers have developed various approaches for detecting an ellipse, and estimating the related parameters. In general, these algorithms can be divided into two groups.

The first group contains algorithms that relate to the optimization or voting process. In this group, the algorithm based on least square method [7], [8] is an efficient method for fitting an ellipse to scattered data points. This approach can guarantee an ellipse-specific solution with no computational ambiguity. However, this algorithm is sensitive to noise, and fails to detect multiple ellipses in a single image. Among existing techniques, Hough transform [9] is the most popular method for ellipse detection, but it tends to occupy a large amount of memory, and takes a long time to execute. An improved ellipse detection method using randomized Hough transform [10], [11], [12], is based on the principle of a stochastic process. Although this algorithm can reduce the storage requirements needed to implement Hough transform, it has a disadvantage, because tangents at any given point are sensitive to changes in the neighboring pixels selected within a window, leading to uncertainty of parameters and lower accuracy. Other approaches based on genetic algorithm were used in [13], [14], [15] to detect multiple ellipses in the image. These algorithms perform well on images with multiple ellipses, and in the presence of noise. However, they consume a lot of computational time.

The second group is based on edge linking of the image. UpWrite method [16] belongs to this group. This method is robust with respect to noise in an image, but it has difficulties in detecting overlapping ellipses. Other methods belonging to this group are Fast Ellipse Extraction (FEE) [17] and Hierarchical Ellipse Extraction (HEE) [18], [19]. They can extract ellipses from complicated images and are not sensitive to noise in an image. However, they are computationally expensive and could not work in the real-time.

Based on these considerations, we suggest a new ellipse detection algorithm. Our approach differs from those discussed above by the following statements. Firstly, we combine the advantages from both the groups of ellipse detection algorithms. By using the least square method to estimate the parameters with no computational ambiguity, our method has received an advantage from the first group. Moreover, by using the edge linking of image, the proposed method is robust with respect to noise in the image, and detects ellipses accurately. Our system has received this advantage from the second group of ellipse detection algorithms. Secondly, unlike least square method where the detector usually fails in detecting multiple ellipses in an image, our method can detect multiple ellipses in the case of both complete and incomplete ellipses. Thirdly, while other methods from the second group (UpWrite, FEE, and HEE) are computationally expensive, our proposed method can detect ellipses in real-time. Finally, to speed up the process, some geometric properties of an ellipse has been used in our method.

The remainder of this paper is organized as follows: In section II, a curve segmentation algorithm is presented. Section III outlines a new way of grouping arc-segments into elliptical arcs. Ellipse fitting approach is presented in section IV. In section V, we show experimental results, and the conclusions are given in section VI.

II. CURVE ELLIPSE EXTRACTION

To reduce the total execution time, we have to estimate the ellipse parameters at the curve level, not at the individual edge

978-1-4244-2794-9/09/$25.00

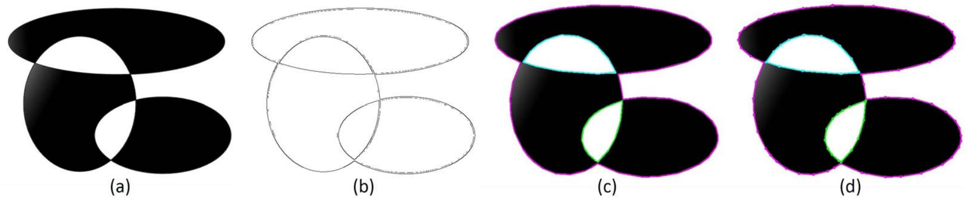

Fig. 1. Line segmentation results, (a): original image, (b): edge image, (c): three edge-contours (M=3: pink, turquoise, green color), (d): line segmentation results for each edge-contour.

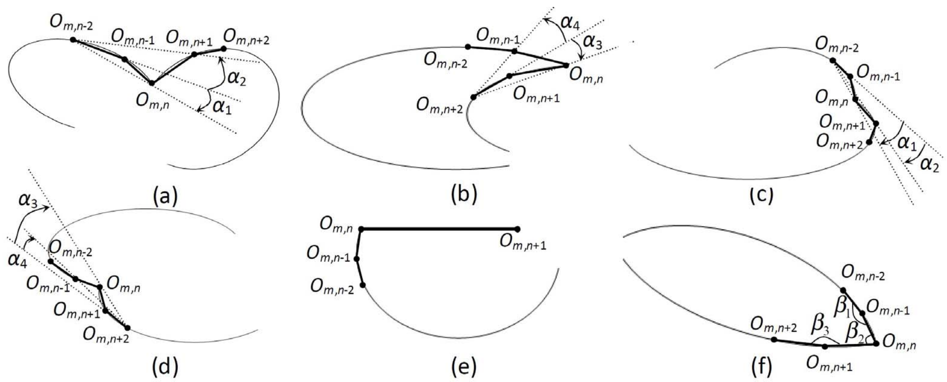

Fig. 2. Curve segmentation conditions, (a): $(\alpha_1 * \alpha_2 < 0)$, (b): $(\alpha_3 * \alpha_4 < 0)$, (c): $(|\alpha_2| - |\alpha_1| < 0))$, (d): $(|\alpha_4| - |\alpha_3| < 0)$, (e): the length condition, (f): the angle condition.

pixel level. This section describes how to extract the elliptical curve from an edge image. We first detect the edge-contour of the objects. After that, we apply a curve segmentation based on geometric properties of ellipses to extract the curvature of a possible ellipse from the edge-contour obtained from the previous step.

A. Line Segmentation Approaches

Over the last thirty years, a number of algorithms have been developed for line segmentation. In this subsection, we describe a simple and efficient algorithm for discriminating the linearity of an edge-contour in an image by slightly modifying Peter Kovesis implementation [20]. Firstly, Canny edge detector [21] is used to find edge-points with a low threshold of 100 and a high threshold of 150 as shown in Fig. 1(b). Secondly, edge pixels are linked together into lists of sequential edge points, one list for each edge-contour. Here, we discard edge-contour less than 10 pixels long. Finally, linear segmentation method is used to form straight-line segments from an edge-contour. Fig. 1 elaborates on these three steps of line segmentation approaches used in this subsection. Fig. 1(c) shows three edge-contours detected from the edge image in Fig. 1(b). The final line segmentation results are shown in Fig. 1(d).

B. Curve Segmentation Approaches

Suppose that we have an edge-contour given from the previous step. However, this contour is not the curve of a possible ellipse as shown in Fig. 1(c). Therefore we want to segment this edge-contour into different curve-segments, such that each curve-segment is the curve of a possible ellipse.

Let M denote the number of edge-contours. In the m-th edge-contour, N_m denotes the number of segment-points. The n-th segment-point of the m-th edge-contour will be denoted by $O_{m,n} = (x_{m,n}, y_{m,n})$, where ($m$=1,...,$M$) and ($n$=1,...,$N_m$). We construct a new graph model to segment each edge-contour into the curve-segments of a possible ellipse. We elaborate on this model as follows: For the m-th edge-contour in the graph, we want to segment it into individual curves at the connection point $O_{m,n}$. To find this connection point, the following conditions are used.

The curvature condition: In an ellipse, a sequence of connected line segments traces an arc segment. In addition, the curvatures of these arcs should be of the same sign throughout the sequence. Based on this idea, we propose here a new way to segment the edge-contour. That is: the m-th edge-contour will be segmented at the connection point $O_{m,n}$, if the following conditions are satisfied:

$$\begin{aligned} & (\alpha_1 * \alpha_2 < 0) or (|\alpha_2| - |\alpha_1| < 0) \\ or \quad & (\alpha_4 * \alpha_3 < 0) or (|\alpha_4| - |\alpha_3| < 0) \end{aligned} \tag{1}$$

where, α_1 is the angle from line $O_{m,n-2}O_{m,n-1}$ to line $O_{m,n-2}O_{m,n}$ and α_2 is the angle from line $O_{m,n-2}O_{m,n-1}$ to line $O_{m,n-2}O_{m,n+1}$ as shown in Fig. 2(a) and Fig. 2(c). Similarly, α_3 and α_4 are angles from line $O_{m,n+2}O_{m,n+1}$ to line $O_{m,n+2}O_{m,n}$ and from line $O_{m,n+2}O_{m,n+1}$ to line

$O_{m,n+2}O_{m,n-1}$ respectively, as shown in Fig. 2(b) and Fig. 2(d).

The length condition: If two neighboring line segments at connection point $O_{m,n}$ of the m-th edge-contour have a large difference in length, we segment this edge-contour into individual curves at this connection point $O_{m,n}$. Fig. 2(e) illustrates the case when $||O_{m,n-1}O_{m,n}|| \ll ||O_{m,n+1}O_{m,n}||$. In this paper, we will segment the edge-contour into individual curves when

$$\begin{array}{rl} & ||O_{m,n-1}O_{m,n}|| > 4||O_{m,n+1}O_{m,n}|| \\ or & ||O_{m,n-1}O_{m,n}|| < 4||O_{m,n+1}O_{m,n}|| \end{array} \qquad (2)$$

The angle condition: Based on the idea that there is not a big difference in gradients of tangents of neighboring line segments on the same ellipse contour, we check if the condition in Eq.(3) is satisfied at every connection point $O_{m,n}$, and segment the edge-contour.

$$||\beta_1|| - ||\beta_2|| > TH_\theta \; and \; ||\beta_3|| - ||\beta_2|| > TH_\theta \qquad (3)$$

where, the β_1, β_2 and β_3 are the angles from line $O_{m,n-1}O_{m,n-2}$ to line $O_{m,n-1}O_{m,n}$, from line $O_{m,n}O_{m,n+1}$ to line $O_{m,n}O_{m,n-1}$, and from line $O_{m,n+1}O_{m,n}$ to line $O_{m,n+1}O_{m,n+2}$ respectively. TH_θ is a threshold (we choose TH_θ = 40 degrees in this paper). Fig. 2(f) shows the results obtained by applying this condition.

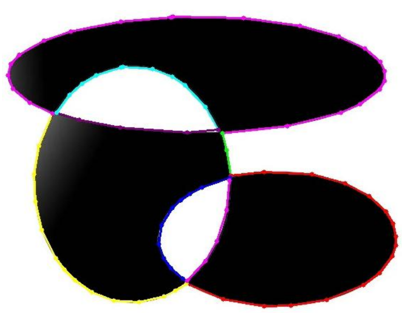

Fig. 3. Eight arc-segments (Eight curves of possible ellipses).

We denote K ($K \geq M$) as the number of arc-segments after finishing the curve segmentation step. Fig. 3 shows eight arc-segments (K=8) after segmentation of the three edge-contours (M=3) in Fig. 1(c). Each arc-segment in Fig. 3 is the curve of a possible ellipse.

III. Curve Grouping Approaches

In this section, we propose a method of grouping arc-segments to form an elliptical arc, if they belong to the same ellipse. This method can reduce the total execution time, as it estimates the parameters of an ellipse at the elliptical arc level, and not at the individual arc-segment level. Firstly, we select an arc-segment from the group of arc-segments in the neighborhood. Finally, we look for global grouping of arc-segments that may not connect, but may come from the same ellipse.

A. Neighborhood Curve Grouping

This step links neighborhood arc-segments that were detected from the previous step into elliptical arcs. The algorithms of our grouping technique for the neighborhood curve can be described as follows.

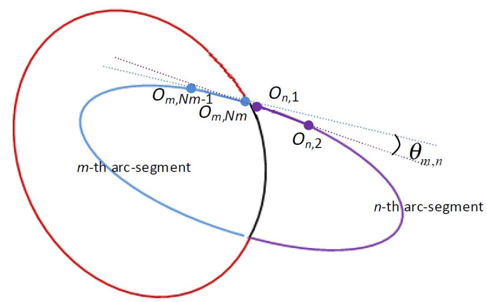

Fig. 4. Neighborhood curve grouping.

Algorithm 1 Neighborhood Curve Grouping Algorithm

1: For each m-th arc-segment (m=1,...,K)
2: We find the n-th arc-segment (n=1,...,K;$n \neq m$) from the arc-segment set
3: Measure the minimum distance from the m-th arc-segment to n-th arc-segment

$$D_{m,n} = \min \left(\begin{array}{c} ||O_{m,1}O_{n,1}||, ||O_{m,N_m}O_{n,1}||, \\ ||O_{m,1}O_{n,N_n}||, ||O_{m,N_m}O_{n,N_n}|| \end{array} \right) \quad (4)$$

4: IF ($D_{m,n} < d_0$) THEN measure the difference in gradients of tangents $\theta_{m,n}$ between m-th arc-segment and n-th arc-segment. Fig. 4 shows the case of $\theta_{m,n}$ where $D_{m,n} = ||O_{m,N_m}O_{m,1}||$. In this step, d_0 is a threshold for determining closeness. We choose (d_0=10 pixels) in this paper.
5: Group the m-th arc-segment with j-th arc-segment that has the minimal value of $\theta_{m,n}$

$$\theta_{m,j} = \min_{1 \leq n \leq K} (\theta_{m,n}) \qquad (5)$$

6: Repeat for every m-th arc-segment.

After finishing the neighborhood curve grouping step, we denote L ($L \leq K$) as the number of elliptical arcs. Fig. 5 shows three elliptical arcs (L=3) grouped from the eight arc-segments from Fig. 3.

B. Global Curve Grouping

Up to now, we have L elliptical arcs to estimate the parameters of L ellipses. However, due to noise and extraneous features in the image, two elliptical arcs may not be connected, but may come from the same ellipse. In this subsection, we propose a simple method to group two elliptical together based on the curvature condition. Our algorithm for global curve grouping is as follows:

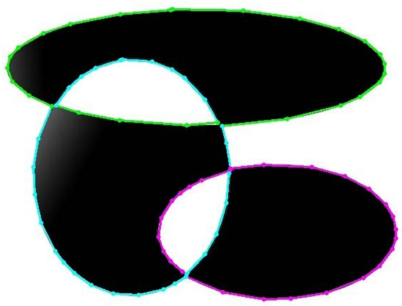

Fig. 5. Neighborhood curve grouping results: three elliptical arcs (L=3: pink, turquoise, green color).

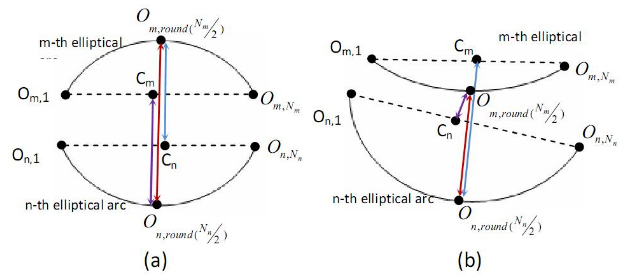

Fig. 6. Global curve grouping results, (a): a pair of elliptical arc that satisfy condition for global curve grouping in Eq.(7), (b): a pair of elliptical arc that do not satisfy condition in Eq.(7).

Algorithm 2 Global Curve Grouping Algorithm

1: For each m-th elliptical arc (m=1,...,L)

2: We find the n-th elliptical arc (n=m+1,...,L) from the elliptical arc set.

3: Measure the minimum distance from the m-th elliptical arc and n-th elliptical arc:

$$D_{m,n} = \min \begin{pmatrix} ||O_{m,1}O_{n,1}||, ||O_{m,N_m}O_{n,1}||, \\ ||O_{m,1}O_{n,N_n}||, ||O_{m,N_m}O_{n,N_n}|| \end{pmatrix} \quad (6)$$

4: IF $(D_{m,n} > d_0)$

We propose a simple criterion to determine whether they should be connected , based on the curvature condition in Eq.(7). IF this condition is satisfied THEN we will group the m-th elliptical arc with n-th elliptical to form a single elliptical arc.

$$\begin{aligned} &\left\| O_{m,round(\frac{N_m}{2})} O_{n,round(\frac{N_n}{2})} \right\| > \left\| C_m O_{n,round(\frac{N_n}{2})} \right\| \\ &\& \left\| O_{m,round(\frac{N_m}{2})} O_{n,round(\frac{N_n}{2})} \right\| > \left\| C_n O_{m,round(\frac{N_m}{2})} \right\| \end{aligned} \quad (7)$$

where, $C_m = \frac{O_{m,1}+O_{m,N_m}}{2}, C_n = \frac{O_{n,1}+O_{n,N_n}}{2}$. Fig. 6(a) shows an example of global curve grouping between two elliptical arcs; while Fig. 6(b) shows the case where a pair of elliptical arcs does not satisfy the condition for global curve grouping in Eq.(7).

5: We repeat for every m-th arc-segment.

After finishing the global curve grouping step, we denote H as the number of pairs of elliptical arcs that are grouped. Therefore, in totally, we have L elliptical arcs that have been detected from neighborhood curve grouping in subsection III-A and H pairs of elliptical arcs that have been detected from global curve grouping in subsection III-B. We use these (H+L) arcs to estimate the parameters of (H+L) ellipses by using the algorithm detailed in the next section.

IV. Ellipse Fitting Approaches

From the previous section, (H+L) candidates for ellipse fitting have been found. For each candidate, in order to estimate the parameters of the ellipse while reducing the total execution time, a direct least square technique [7] is used. This approach will find the best fit to an ellipse from the segment-point set of each candidate.

After detecting the m-th possible ellipse with five parameters, we wish to find out if this possible ellipse is a true ellipse or not, by using the following process.

Let H and W be the height and width of an image. Edge pixel in image will be expressed by $P_{i,j} = (x_{i,j}, y_{i,j})$, where $(1 \leq i \leq H)$ and $(1 \leq j \leq W)$. Let NE_m denotes the number of pixels on the m-th ellipse, (m=1,2,...,H+L). Each pixel on this ellipse will be presented by $PE_{m,k}, (k = 1, 2, ..., NE_m)$. For each pixel, the distance from $PE_{m,k}$ to the nearest edge pixel $P^*_{i,j}$ of the edge image can be obtained. We continue the above process until all the pixels in this ellipse have been examined. Then we calculate the error:

$$E_m = \frac{1}{NE_m b} \sum_{k=1}^{NE_m} (PE_{m,k} - P^*_{i,j}) \quad (8)$$

where, b is the size of the short axis of this ellipse. In this paper, we use $NE_m = 2N_m$. The error in Eq.(8) is used to detect a true ellipse in an image. IF E_m is smaller than the given global threshold TH_E, we claim that the possible ellipse is a true ellipse. Otherwise, we will remove this possible ellipse.

V. Simulation Studies

In this section, the performance of the proposed algorithm has been compared with those of the RHT [10], UpWrite [16], FEE [17] and HEE [18], [19] in terms of computation time and accuracy. The proposed algorithms have been implemented and tested on a PC (Pentium 4 at 3 GHz, 1GB of Ram) using free software libraries (OpenCV libraries for image), and Visual C++ 6.0.

The first experiment illustrates the implementation of our algorithm as compared to other algorithms on real-world images. Fig. 7(a) shows the real image (306x281) that contains both complete and incomplete ellipses. The image shown in Fig. 7(b) is the result of RHT method. The accuracy of this method is very poor. Moreover, the computational cost of this algorithm as shown in Table 1 is also the most expensive. In Fig. 7(c), the UpWrite algorithm reduces the computation time significantly, but the ellipse extraction accuracies remain poor. FEE method in Fig. 7(d) extracts a complete ellipse,

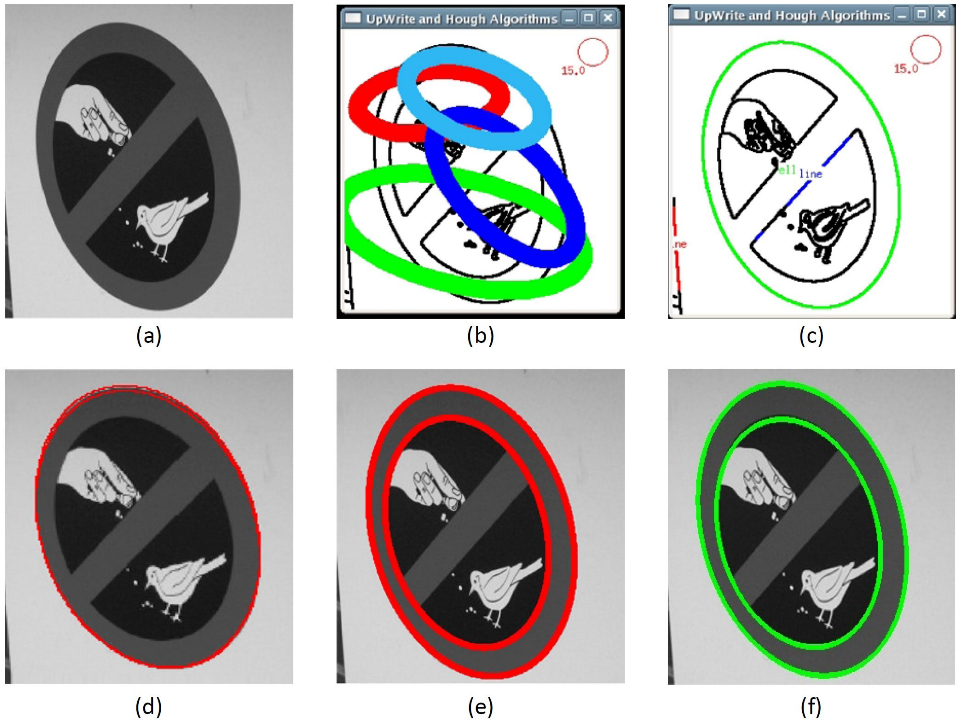

Fig. 7. Ellipse detection results for real image (306x281), (a): original image, (b): RHT, (c): UpWrite, (d): FEE, (e): HEE, (f): The proposed real-time ellipse detection.

TABLE I
TEST RESULTS FOR A REAL IMAGE (306X281).

Algorithm	RHT	UpWrite	FEE	HEE	Ours
Time (ms)	3900	1340	2030	1000	31

TABLE II
TEST RESULTS FOR A REAL IMAGE (480X640).

Algorithm	FEE	HEE	Our method
Time (ms)	6580	3650	219

but performs poorly with an incomplete ellipse. For the result shown in Fig. 7(e), HEE algorithm is capable of extracting all the incomplete and complete ellipses accurately. However, this method is computationally complex, and could not detect the ellipses in real-time, as shown in Table V. Comparing these methods, we find that our proposed method, shown in Fig. 7(f) with global threshold TH_E=0.015, could not only successfully detect all the ellipses, but also reduces the computational complexity significantly, and could detect these ellipses in real-time.

In the second experiment, the real image (480x640) shown in Fig. 8(a) is used for testing. The reference FEE method in Fig. 8(b) has difficulty in determining the ellipse accurately. HEE method in Fig. 8(c) still needs a higher computational time for this example. A comparison of the computational efficiency among FEE, HEE and our proposed methods is presented in Table V, and shows that our method in Fig. 8(d) with global threshold TH_E=0.007 outperforms the reference algorithms not only in terms of accuracy but also speed.

The Fig. 9 shows some other images (240x320) that have been used to test our proposed algorithm. Fig. 9(a)-(c) show the original images. The results produced by proposed algorithm corresponding to these images have been shown in Fig. 9(d)-(f) with global threshold TH_E=0.02, TH_E=0.009 and TH_E=0.015. For each image in this example, the computational time of our algorithm is 31 milliseconds.

In the final experiment, synthetic images (480x640) were generated and tested. Four of which are shown from Fig. 10(a) to Fig. 10(d). The first two images in Fig. 10(a) and Fig. 10(b) contains both complete ellipses and incomplete ellipses. Some of them have overlapping lines and rectangles. The last two images in Fig. 10(c) and Fig. 10(d) are corrupted with salt and pepper noise (at noise rate = 5%). This synthetic image contains incomplete ellipse, complete ellipses, rectangles and triangle. Some of them are overlapping each other. The detected ellipses proposed by our algorithm corresponding to four images with global threshold TH_E=0.01 are shown from Fig. 10(e) to Fig. 10(h), respectively. Clearly, experimental results demonstrate that the proposed algorithm is not only accurate, but also and robust with respect to salt and pepper noise.

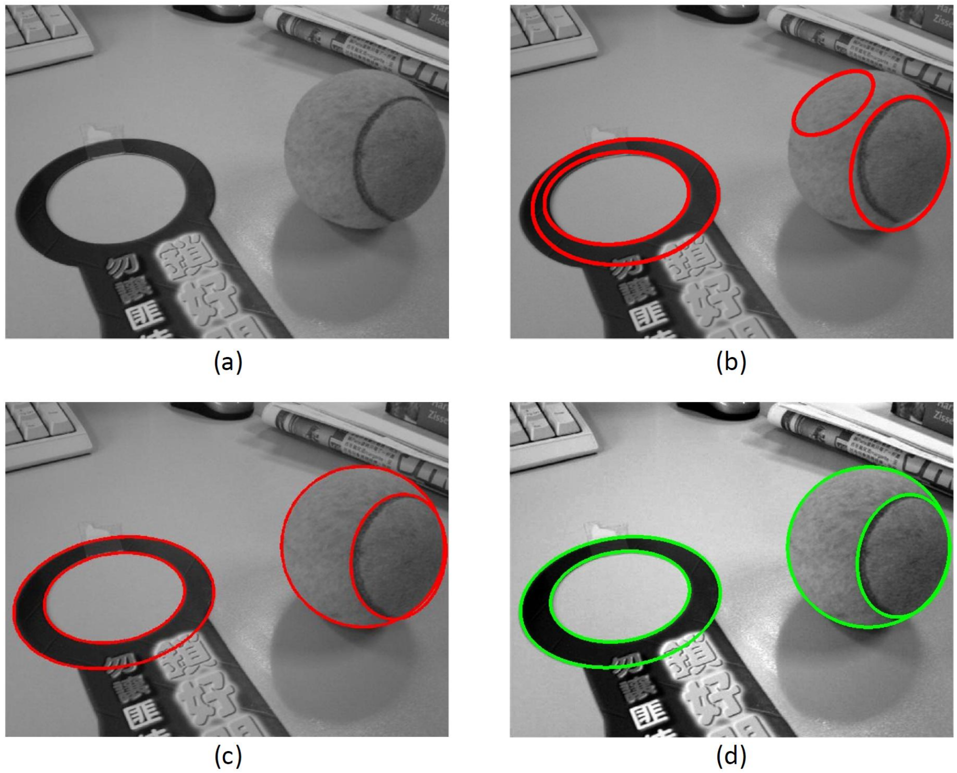

Fig. 8. Ellipse detection results for real image (480x640), (a): original image, (b): FEE, (c): HEE, (d): Our proposed method.

VI. Conclusions

In this paper, we have proposed new edge grouping method for real-time ellipse detection. The performance study of this technique with both real images and synthetic images shows that the proposed detector is capable of detecting ellipse features with an excellent accuracy. Furthermore, the proposed method is easy to implement and could detect the ellipses at a high speed ($>$20 frames per second on an image of size 240x320).

VII. Acknowledgements

This research has been supported in part by the Canada Research Chair Program, AUTO21 NCE, and the NSERC Discovery grant.

References

[1] R. Hu, Q. Ji, Camera Self-Calibration from Ellipse Correspondences, *Proc. of the 2001 IEEE ICRA*, vol.3, pp.2191–2196, May 2001.

[2] Canzler U., Kraiss K.F., Person-adaptive facial feature analysis for an advanced wheelchair user-interface, *IEEE Intl. Conf. on Mechatronics and Robotics*, pp.871–876, 2004.

[3] dOrazio T., Guaragnella C., Leo M., Distante A., A new algorithm for ball recognition using circle hough transform and neural classifier, *Pattern Recognition*, pp.393–408, 2004.

[4] Radford C., Houghton D., Vehicle detection in open-world scenes using a hough transform technique, *Third International Conference on Image Processing and its Applications*, pp.78–82, 1989.

[5] Mclaughlin R., Randomized hough transform: Improved ellipse detection with comparison, *Pattern Recognition Letters*, pp.299–305, 1998.

[6] Piccioli G., Michelli E., Parodi P., Campani M., Robust road sign detection and recognition from image sequences, *Proceedings of the IEEE Symposium on Intelligent Vehicles*, Paris, FR, pp. 278–283. IEEE Computer Society Press, Los Alamitos, 1994.

[7] R. Halir and J. Flusser, Numerically stable direct least squares fitting of ellipses, *6th Int. Conf. in Central Europe on Computer Graphics and Visualization*, pp. 125–132, 1998.

[8] A. Fitzgibbon, M. Pilu, and R.B. Fisher, Direct least square fitting of ellipses, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.476–480, 1999.

[9] Nair P.S., Ssaunders A.T., Hough transform based ellipse detection algorithm, *Pattern Recognition Letters*, pp.777–784, 1996.

[10] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (RHT), *Pattern Recognition*, Vol. 11 (5), 331–338, 1990.

[11] R.A. McLaughlin, Randomized Hough transform: better ellipse detection, *IEEE Proc. on TENCON, Digital Signal Processing Applications*, Vol. 1, pp.409–414, 1996.

[12] Ch. Cheng and Y. Liu, Efficient technique for ellipse detection using restricted randomized Hough transform, *Int. Con. on Information Technology: Coding and Computing*, Vol. 2, pp.714–718, 2004.

[13] S. Procter, J. Illingworth, A comparison of the randomized Hough transform and a genetic algorithm for ellipse detection, *Pattern Recognition*, pp.449–460, 1994.

[14] K.U. Kasemir, K. Betzler, Detecting ellipses of limited eccentricity in images with high noise levels, *Image Vision Comput.*, pp.221–227, 2003.

[15] J. Yao, N. Kharma, P. Grogono, A multi-population genetic algorithm for robust and fast ellipse detection, *Pattern Anal. Appl.*, pp.149–162, 2005.

[16] R.A. McLaughlin, M.D. Alder, The Hough transform versus the UpWrite, *IEEE Trans. Pattern Anal. Mach. Intell.*, pp.396–400, 1998.

[17] E. Kim, M. Haseyama, H. Kitajima, Fast and robust ellipse extraction from complicated images, *Proceedings of IEEE International Conference on Information Technology & Applications*, 2002.

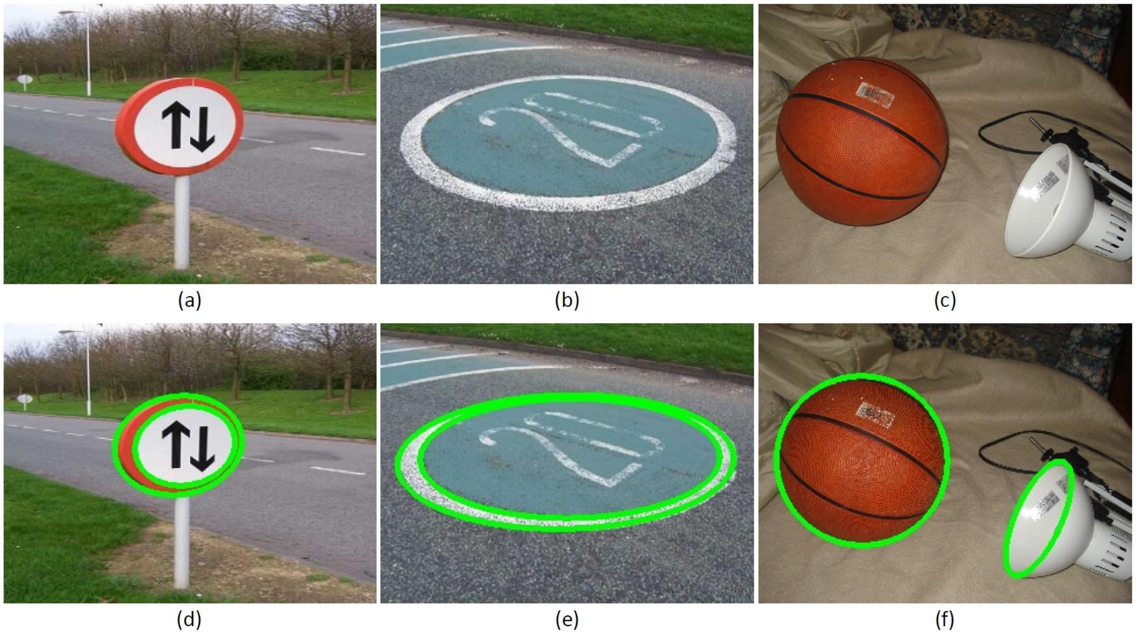

Fig. 9. Testing results for real images.

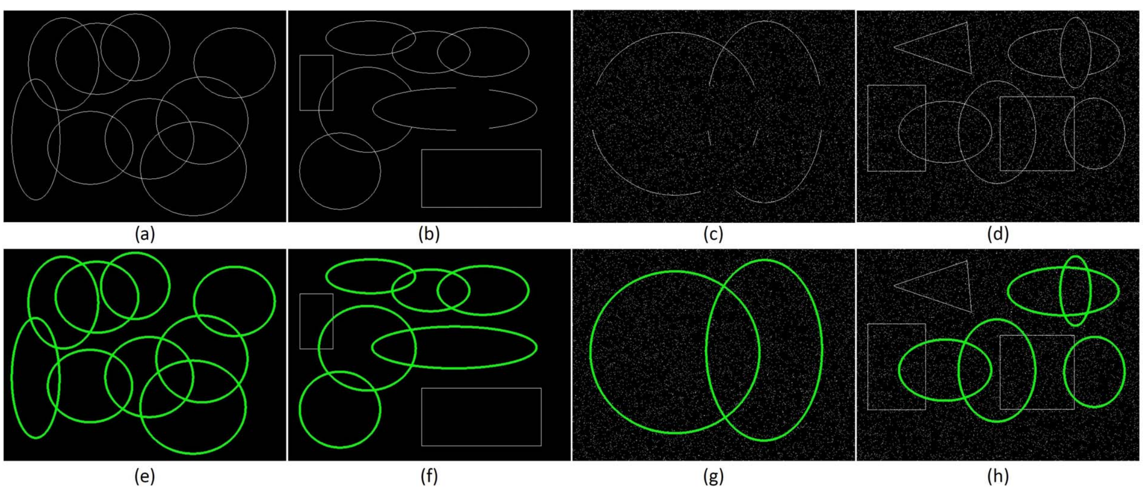

Fig. 10. Testing results for synthetic images.

[18] F. Mai, Y.S. Hung, H. Zhong, W.F. Sze, A hierarchical approach for fast and robust ellipse extraction, *Proceedings of IEEE International Conference on Image Processing*, San Antonio, TX, USA, September 2007.

[19] F. Mai, Y.S. Hung, H. Zhong, W.F. Sze, A hierarchical approach for fast and robust ellipse extraction, *Pattern Recognition*, Vol.41, Issue 8, pp.2512–2524, August 2008.

[20] *http://www.csse.uwa.edu.au/ pk/research/*.

[21] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, pp.679–714, 1986.