Combining Feature-based and Model-based Approaches For Robust Ellipse Detection

Halil Ibrahim Cakir
Department of Computer Engineering
Dumlupinar University, Kutahya, Turkey
Email: cakirhal@dumlupinar.edu.tr

Burak Benligiray, Cihan Topal
Department of Electrical and Electronics Engineering
Anadolu University, Eskisehir, Turkey
Email: {burakbenligiray, cihant}@anadolu.edu.tr

Abstract—**Fast and robust ellipse detection is a vital step in many image processing and computer vision applications. Two main approaches exist for ellipse detection, i.e., model-based and feature-based. Model-based methods require much more computation, but they can perform better in occlusions. Feature-based approaches are fast but may perform insufficient in cluttered cases. In this study, we propose an hybrid method which combines both approaches to accelerate the process without compromising accuracy. We extract elliptical arcs to narrow down search space by obtaining seeds for prospective ellipses. For each seed arc, we compute a limited search region consisting of hypothetical ellipses that each can be formed with that seed. Later, we vote them on the edge image to determine best hypothesis among the all, if exists. We tested the proposed algorithm on a public dataset and promising results are obtained compare to state of the art methods in the literature.**

I. INTRODUCTION

Apparent geometric features of certain objects are commonly used for their detection in computer vision applications [1], [2], [3]. Edges and corners are the most primitive geometric features. More complex geometric features, such as lines, circles and ellipses can be formed using these primitive features. There are two main approaches for this augmentation, which are model-based methods and feature-based methods. Model-based methods have been long studied in the literature and can be summarized as exhaustively fitting models to lower level features. The second approach taken by feature-based methods forms the desired geometric features as combinations of lower level features. Characteristically, feature-based methods have a lower processing time and are more robust against noise.

Among the geometric shapes, ellipses are one of the most common geometric shapes in the nature. Since a circle's projection onto a camera image plane is also an ellipse, encountering ellipses can be quite frequent in real life applications. Hence they are used as geometric features in applications from a wide array of fields such as intelligent vehicles [4], medicine [5] and biometrics [6], [7]. An ellipse has 5 degrees of freedom (DoF), which stands for the center coordinates x & y, semi-major and semi-minor axes a & b, and rotation angle θ. It is considerably higher than common geometric features. Therefore, it is one of the hardest one to properly detect in many scenarios.

Due to its 5 DoF, various shapes such as rectangles can be represented by an ellipse with a reasonable amount of error. This situation both increases the computation time for the search in high dimensional space and beclouds distinguishing of ellipses from other shapes. Furthermore, for most applications, ellipses should be able to be accurately detected under occlusion and in a reasonable processing time. Due to these reasons, ellipse detection is a difficult problem that requires further improved solutions.

In this study, we propose an hybrid method for ellipse detection. In our method, the lower level features that forms the ellipses are arcs, which themselves are formed by edge segments. Each arc is used as a seed for an ellipse, similar to a feature-based approach. Then, additional arcs from the same ellipse are searched in the image by altering the seed ellipse's parameters, which is similar to a model-based approach. Therefore, we reduce the search space in one dimension and perform voting in a limited area of image. In this way, we avoid performing computationally expensive model-based voting operation in 5-dimensional parameter space of the ellipse for the entire image. Eventually, the combination of model-based and feature-based approaches yields the advantage of former, which is occlusion resistance, and the advantages of latter, which are lower running time and robustness against noise.

II. RELATED WORK

The most common method of model-based ellipse detection is to apply Hough Transform (HT) [8]. However, this is not as viable of an approach as detecting lines with HT, as ellipses have 5 degrees of freedom, while lines have only 2. A large degree of freedom results in a far larger parameter space to be searched, hence proportional processing times. Improvements to HT mainly aim to shrink the search space. Lei and Wong constraint the search to fewer parameters at start, then move on to determine other parameters of the ellipse [9]. Randomized Hough Transform is another method designed to reduce the complexity of HT [10], which is used to detect ellipses [11]. Zhang and Liu take edges' convexity information into account while applying HT, which improves running time by reducing the required computation [6].

This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) and Anadolu University Commission of Scientific Research Projects (BAP) under the grant numbers 115E928 and 1505F319, respectively.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

Fig. 1: The block diagram of the proposed algorithm.

Feature-based models combine lower level features to form ellipses. The lowest level features to be used are edges. Since computing exhaustive combinations of edge pixels is not feasible, RANSAC has been used to find edge pixels that are on the same ellipses [12]. Higher level features such as line segments and arcs are lower in number, hence using them as building blocks of ellipses is preferred by some studies. Libuda et al. propose a hierarchical method, detecting edges, line segments, arcs, extended arcs and ellipses in this particular order [13]. In another work, curves are detected directly from the edges, which are then combined to form ellipses [14]. In methods that use high level features such as arcs, ellipses tend to be fit to the edges that constitute these features [15]. In general, methods that use higher level features report a much lower processing time compared to other methods.

In addition, Genetic Algorithm (GA) and its variants are rarely used for ellipse detection. Yin applies operations which are specific to GA and calculate a fitness value [16]. After some improvements on candidates, grouping and choosing higher values result to ellipses on a local search phase. Multi-population Genetic Algorithm (MPGA) is also used for detecting partial and full ellipses [17]. It works by clustering subpopulations around an ellipse hypothesis. It is compared to Randomized Hough Transform and another variant Sharing Genetic Algorithm.

There are also methods that derive from feature-based methods, yet are essentially model-based. Prasad et al. choose edge pixels from the same ellipse based on their local curvature, followed by a lower dimensional HT search [18]. Fornaciari et al. propose a selection strategy for detected arcs [19]. HT is applied to the selected arcs to detect the ellipses.

III. Proposed Method

The proposed hybrid ellipse detection algorithm basically aims to restrict the five-dimensional search space by the information acquired from primitive arc features and follows the steps shown in Fig. 1. In the first step elliptical arc features are extracted from the image and each of them represented by a parametric ellipse equation. Then, separate search regions are formed for each arc by manipulating their parameters. Search region of an arc is actually comprised of a number of hypothetic ellipses that each of them can be a valid ellipse for the seed arc with a reasonable amount of error. Afterwards, a voting scheme is performed for each arc in their own search regions on the edge image and number of edge pixels are accumulated for each ellipse hypothesis. Finally, the best ellipse hypothesis is found according to the voting results and a threshold value. In the following subsections, we explain steps of the proposed algorithm in fine detail.

A. Arc Detection

In the first step of the algorithm we extract elliptical arc segments from the input image in a three stage method shown in Fig.2. First, we detect edge segments from the image each of them as a connected chain of pixels [20]. A test image and detected edge segments are shown in Fig. 3.a and Fig. 3.b, respectively. Once we obtain the edge segments, we find the corners along each segment with a curvature-based corner detection algorithm (see Fig. 3.c) [21]. Then we fit an ellipse to pixel chains lying between two consecutive corners along the edge segments and compute fitting error to decide whether the pixel chain is an elliptical arc or not [22]. If resulted error is smaller than a reasonable threshold (i.e. < 2 px), we determine that the pixel chain is a valid arc. In Fig. 3.d extracted elliptical arcs and the their corresponding ellipse results are shown. With the fitting operation we obtain an ellipse equation in conic form (see Eq. 1) to represent the arc and be utilized for computation of arc specific search region.

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0 \qquad (1)$$

B. Search Region Computation

At the beginning, all elliptical arcs and their conic equations are extracted from the input image. In the second step, we compute a search region for each seed arc to determine whether that arc can be part of a valid ellipse. A search region is actually a list of ellipse hypotheses each of which can represent the arc with a reasonable amount of fitting error. In this way, instead of performing computationally expensive five-dimensional voting operation on entire image, we perform voting in only one-dimension and on a very limited area in the image.

To compute the search region, we first derive the parametric equation for each seed arc from the conic equation (Eq. 1) obtained from ellipse fitting in the previous step:

$$\frac{(x-x_c)^2}{a^2} + \frac{(y-y_c)^2}{b^2} = 1 \qquad (2)$$

where (x_c, y_c) is center coordinates; a and b are semi-major and semi-minor axes of the ellipse, respectively [23]. The same procedure also yields a θ value as the rotation angle of the ellipse around x axis. As shown in Fig. 4, we tune the center location, semi-major and semi-minor axes so as to derive ellipse hypotheses which are similar to the ellipse of seed arc.

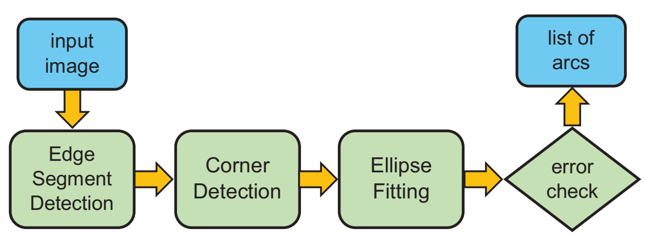
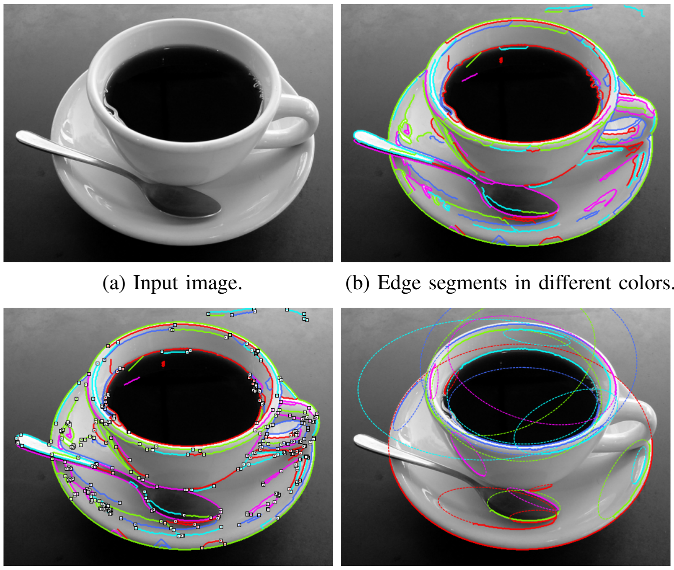


Fig. 2: Block diagram of arc detection method.



(a) Input image. (b) Edge segments in different colors.

(c) Detected corners on edge segments. (d) Arcs with corresponding ellipses.

Fig. 3: Steps of arc segment detection procedure seen in Fig. 2 applied on a test image.

To find the ellipse hypotheses, we need to select one of the axes or both and compute a direction for the search region to span through. For this purpose, we determine the closest axis to seed arc by computing distances between arc's end points $e_i = (x_{e_i}, y_{e_i})$ and each vertices of axes on ellipse $v_i = (x_{v_i}, y_{v_i})$. Each vertice is simply a point which is an intersection of semi-major and semi-minor axes and ellipse contour. For each end point of arc we get a closest vertice and this vertice appearently shows us which axis value is to change in search region. It is also possible to change both axis values and it can be seen as an example in Fig. 5.

After we find out the axis (i.e. semi-major or semi-minor or both) where the search region take place, we need to determine the direction that it spans through. Direction of search region is the direction of vector $\vec{mc}$ where $m = (x_m, y_m)$ is the arc's middle point and $c = (x_c, y_c)$ is the center of ellipse (See Eq. 3). Once we determine the axis and direction for the search region, we derive ellipse hypotheses by tuning chosen semi-major (a) and semi-minor (b) axes lengths with a certain ratio ($\pm 10\%$, i.e. between 90% and 110%) of axes with a 1 pixel step ratio. For each a and b values, we concurrently update the ellipse center through search region direction in order to make sure that the new ellipse still overlaps seed arc.

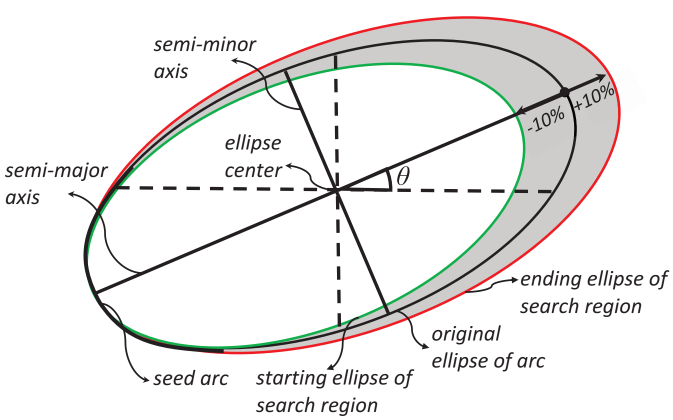


Fig. 4: Search region computed for a single arc.

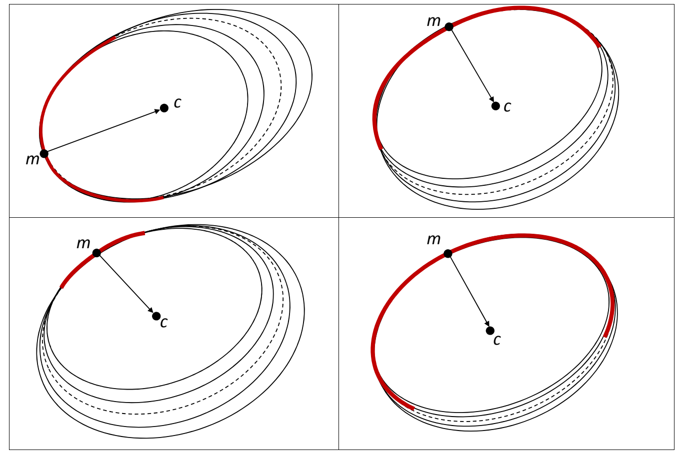


Fig. 5: Different variations of search regions with regard to different arc locations.

$$\alpha_i = \arccos \frac{y_c - y_m}{x_c - x_m} \tag{3}$$

As a result of tuning axis length, center coordinates of ellipse hypotheses must be shifted depending on amount of variation (k). Initial center point starting from first ellipse hypothesis is computed by Eq. 4 and for each hypothesis k value is increased by 1 iteratively.

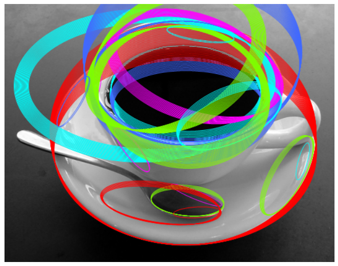
$$x_{c_0} = x_c + k \cos \alpha, \;\; y_{c_0} = y_c + k \sin \alpha \tag{4}$$

Finally, we obtain the ellipse hypotheses which constitutes search region for voting.

C. Detection By Voting

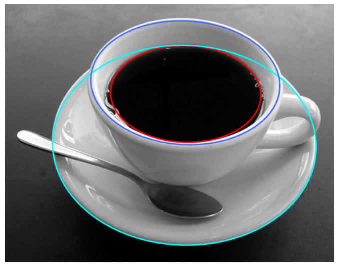
After we obtain all seed arcs and search regions, i.e. ellipse hypotheses, we compute a ratio for each ellipse hypothesis by voting to edge pixels. We compute ellipse contour points for each ellipse hypothesis and utilize those point location to vote on edge image. For each hypothesis, we determine the number of overlapping edge pixels and then compute the overlapping ratio by dividing the value by perimeter of the hypothesis. To test if an ellipse hypothesis is valid, we set an adaptive threshold which varies with respect to the hypothesis' perimeter. In this manner, larger ellipses are to have higher voting ratios to be elected as valid ellipses. In the experiments, we employ threshold values for voting ratio linearly varying between 0.4 and 0.9 for ellipses that are 100 and 1000 pixels in perimeter, respectively.

(a) Detected arc segments.



(b) Separate search regions for arcs.

(c) Voting result for one arc.



(d) Detected ellipses after voting.

Fig. 6: Ellipse detection process for the input image. In (c) a voting result is seen with the color legend for voting ratios.

IV. Experimental Results

In our experiments, we used an annotated real dataset and compared our method with three popular model-based and feature-based methods [18], [13], [24] named Prasad, Libuda and Basca respectively. The dataset is named Smartphone which recently proposed in [19]. It has 629 frames in 640x480 resolution and almost each frame is challenging for detection of real elliptical objects. All of them are used for comparison and the comparison metric to quantify the detection performance is the overlap ratio calculation mentioned in [18]. Each algorithm was tested under fixed ellipse validation threshold value that yields the best performance for this dataset. These threshold values are 0.83 for Prasad, 0.01 for Libuda and 0.93 for Basca. Detection performances as f-measure is shown in Fig. 7. It is obvious that our algorithm outperfoms others quantitatively. Some visual results from the dataset are illustrated in 8. Timing results for each algorithm can be seen in Table. I. Algorithms exhibit various timing results from long running to real-time. As it is seen that our algorithm and Libuda are in a competition by performing very close timing in milliseconds. Note that all experiments are performed on a laptop computer with an Intel i7 2.40 GHz processor and 8 GB of RAM.

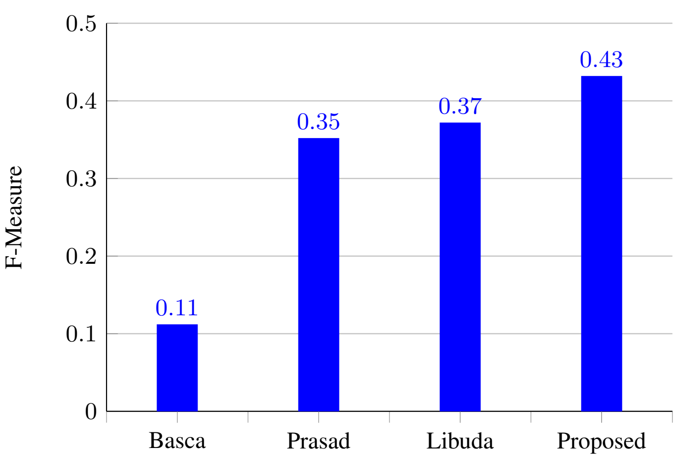


Fig. 7: Quantitative detection performance comparison for each algorithm. Each bar represents average F-Measure value per image.

TABLE I: Average timing results of each algorithm.

Algorithm	Basca	Prasad	Libuda	Proposed
Timing (ms)	533.10	2780.50	27.18	34.79

V. Conclusion

In this study, we propose an ellipse detection algorithm which aims to combine advantages of feature-based and model-based approaches. The algorithm starts in feature-based manner by extracting edge segments, corners and elliptical arcs from the input image, respectively. After the arcs are obtained, they are used as seeds for detection of ellipses and the algorithm computes arc-specific search region based on the information inferred from each arc. A search region is actually a group of ellipse hypotheses each of which can represent the seed arc with a reasonable amount of proximity. Search regions are then utilized to perform voting on the edge image for the arc that the region is derived from. During the voting, edge pixels are accumulated for each ellipse hypothesis to perform comparison. Finally, at most one ellipse is selected among the hypotheses if it can accumulate enough number of pixels.

The proposed method can utilize very primitive features that cannot be form an arc but still belong to an ellipse by using nearby arcs as seeds. In this way the algorithm both saves excessive computation time, and handles cluttered situations where ellipses have clutters. We perform extensive experiments with a publicly available dataset and obtain better accuracy results compare to existing methods in the literature. The proposed method gives also promising results in terms of computation time.

References

[1] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 2, pp. 150–162, 1994.

[2] A.C. Berg, T.L. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 26–33 vol. 1.

[3] F. Jianping, D.K.Y. Yau, A.K. Elmagarmid, and W.G. Aref, "Automatic image segmentation by integrating color-edge extraction and seeded region growing," *Image Processing, IEEE Transactions on*, vol. 10, no. 10, pp. 1454–1466, 2001.

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

[5] W. Lu and J. Tan, "Detection of incomplete ellipse in images with strong noise by iterative randomized hough transform (irht)," *Pattern Recognition*, vol. 41, no. 4, pp. 1268 – 1279, 2008.

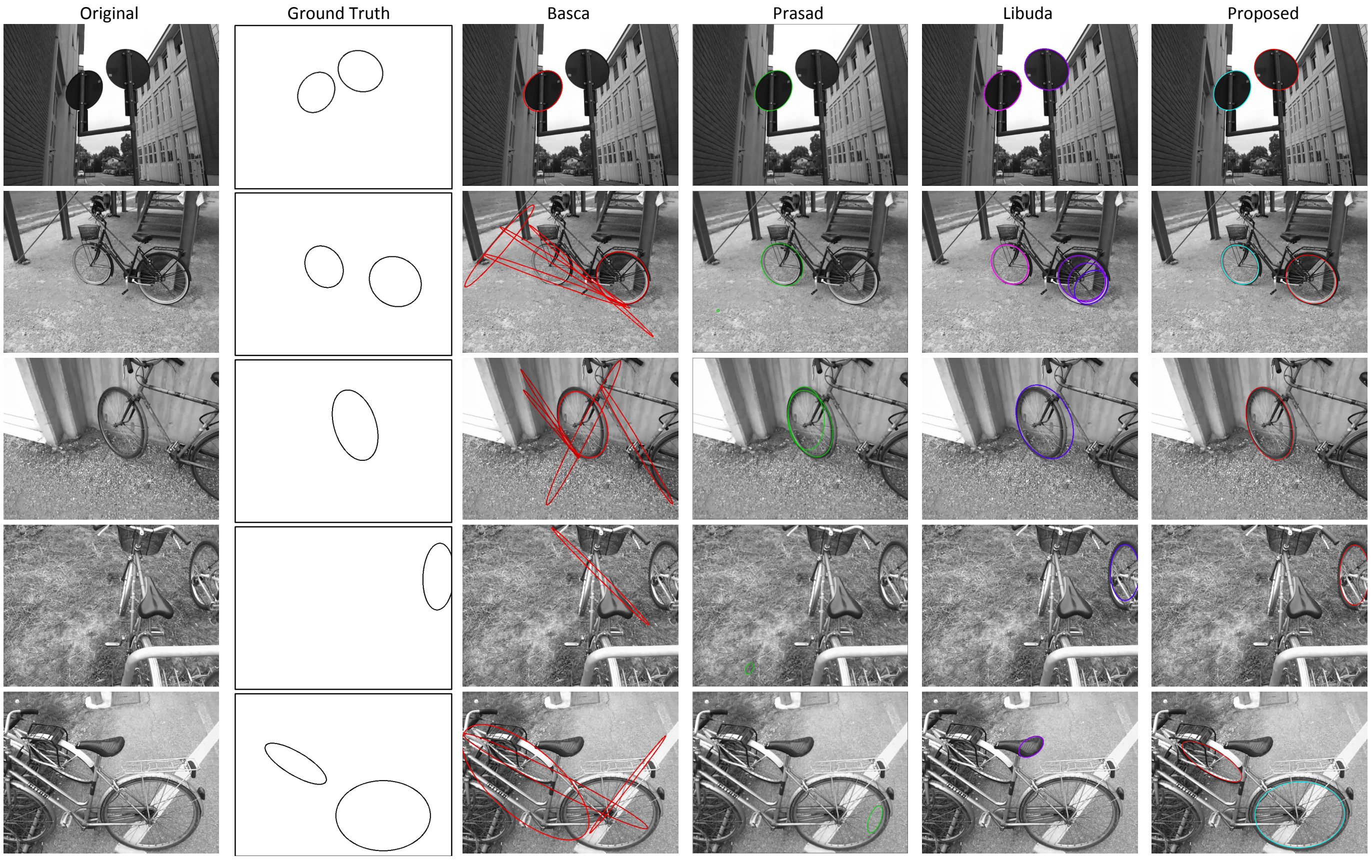


Fig. 8: Qualitative results of each algorithm for various images from the employed dataset.

[6] S. C. Zhang and Z. Q. Liu, "A robust, real-time ellipse detector," *Pattern Recognition*, vol. 38, no. 2, pp. 273 – 287, 2005.

[7] E.M. Arvacheh and H.R. Tizhoosh, "Iris segmentation: Detecting pupil, limbus and eyelids," in *Image Processing, 2006 IEEE International Conference on*, 2006, pp. 2453–2456.

[8] P. V. C. Hough, "Machine analysis of bubble chamber pictures," in *Proceedings, 2nd International Conference on High-Energy Accelerators and Instrumentation, HEACC 1959*, 1959, vol. C590914, pp. 554–558.

[9] Y. Lei and K. C. Wong, "Ellipse detection based on symmetry," *Pattern Recognition Letters*, vol. 20, no. 1, pp. 41 – 47, 1999.

[10] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized hough transform (RHT)," *Pattern Recognition Letters*, vol. 11, no. 5, pp. 331 – 338, 1990.

[11] R. A. McLaughlin, "Randomized hough transform: Improved ellipse detection with comparison," *Pattern Recognition Letters*, vol. 19, no. 34, pp. 299 – 305, 1998.

[12] G. Song and H. Wang, "A fast and robust ellipse detection algorithm based on pseudo-random sample consensus," in *Computer Analysis of Images and Patterns: 12th International Conference, CAIP 2007. Proceedings*, 2007, pp. 669–676.

[13] L. Libuda, I. Grothues, and K. F. Kraiss, "Ellipse detection in digital image data using geometric features," in *Advances in Computer Graphics and Computer Vision: International Conferences VISAPP and GRAPP 2006*, 2007, pp. 229–239.

[14] 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.

[15] T. M. Nguyen, S. Ahuja, and Q.M.J. Wu, "A real-time ellipse detection based on edge grouping," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 3280–3286.

[16] P. Y. Yin, "A new circle/ellipse detector using genetic algorithms," *Pattern Recognition Letters*, vol. 20, no. 7, pp. 731 – 740, 1999.

[17] J. Yao, N. Kharma, and P. Grogono, "Fast robust ga-based ellipse detection," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, vol. 2, pp. 859–862 Vol.2.

[18] D. K. Prasad, M. K. H. Leung, and S. Y. Cho, "Edge curvature and convexity based ellipse detection method," *Pattern Recognition*, vol. 45, no. 9, pp. 3204 – 3221, 2012.

[19] M. Fornaciari, A. Prati, and R. Cucchiara, "A fast and effective ellipse detector for embedded vision applications," *Pattern Recognition*, vol. 47, no. 11, pp. 3693 – 3708, 2014.

[20] C. Topal and C. Akinlar, "Edge drawing: A combined real-time edge and segment detector," *Journal of Visual Communication and Image Representation*, vol. 23, no. 6, pp. 862 – 872, 2012.

[21] C. Topal, K. Özkan, B. Benligiray, and C. Akinlar, "A robust CSS corner detector based on the turning angle curvature of image gradients," in *IEEE Int'l Conf. Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada*, 2013, pp. 1444–1448.

[22] G. Taubin, "Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 11, pp. 1115–1138, 1991.

[23] R. A. Adams, "Calculus: A complete course," *Pearson Addison Wesley*, pp. 441–442, 2006.

[24] C.A. Basca, M. Talos, and R. Brad, "Randomized hough transform for ellipse detection with result clustering," in *Computer as a Tool, 2005. EUROCON 2005.The International Conference on*, 2005, vol. 2, pp. 1397–1400.