Enhancing the Randomized Hough Transform with K-Means Clustering to Detect Mutually-Occluded Ellipses

Tinghui Zhou and Nikolaos Papanikolopoulos

Department of Computer Science and Engineering, University of Minnesota

***Abstract*— In the attempts to resolve the problem of ellipse detection, the Randomized Hough Transform (RHT) serves as a powerful variant of the standard Hough transform that exploits the geometric properties of ellipses in order to speed up the detection process. Despite its simplicity and efficiency, the RHT performs poorly if the target ellipses are overlapped (or mutually-occluded) with each other. We present a novel method that utilizes k-means clustering to boost the performance of the RHT in detecting mutually-occluded ellipses, and test its effectiveness for both synthetic and real-world images. However, as a result of using k-means clustering, this method is susceptible to being stuck at a local optima.**

I. INTRODUCTION

Ellipse detection is an important problem in image processing applications, and has been studied extensively in the community since the introduction of Hough transform [1]. Among the many variants of standard Hough transform (SHT), Randomized Hough Transform (RHT) proposed by Xu et al. [2] stands out because of its simplicity, high speed, and small storage. However, these nice properties of RHT come at the cost of its susceptibility to outliers in noisy images and proneness to fail in detecting mutually-occluded ellipses, which essentially results from its blindness in sampling the point set used to fit an ellipse. A detailed analysis of the cause of these disadvantages is provided in Section III.

This paper introduces a novel ellipse detection method that incorporates the basic idea of *k*-means clustering [3] into the setting of RHT to enhance its performance if the processed image contains mutually-occluded ellipses. The overall flow of this method is organized in analogy to the *k*-means algorithm. First, pixels in the processed image are organized into different clusters based on their distances to the candidate ellipses – current estimation of target ellipses; Second, fit a new set of ellipses using RHT within each pixel cluster; Third, update the candidate ellipses with the newly fitted ones if they are a better fit to the processed image. This extension enables RHT to work effectively even when target ellipses are mutually-occluded, and the computational complexity only scales linearly with the image size.

T. Zhou, Computer Science and Engineering, University of Minnesota-Twin Cities, 200 Union Street, SE, Minneapolis, MN 55455, United States, email: zhoux439@umn.edu

N. Papanikolopoulos, Computer Science and Engineering, University of Minnesota-Twin Cities, 200 Union Street, SE, Minneapolis, MN 55455, United States, email: npapas@cs.umn.edu

However, due to the nature of *k*-means clustering, our method is not guaranteed to converge to the globally optimal solution at termination. In other words, the results it returns might not reflect the best fit to the target ellipses. One way to overcome this problem is to run the algorithm with different initial configurations and pick the best one as the final result. Another way is to allow some error tolerance in determining the fitness of a ellipse set to the processed image as described in Section IV-F.

The rest of this paper is organized as follows. We give a brief review of existing ellipse detection approaches in Section II. An analysis of the limitations of RHT and their cause is provided in Section III. We present in detail our method in Section IV, and illustrate its effectiveness and advantages over RHT through the experimental results in Section V. We conclude our work in Section VI.

II. RELATED WORKS

The standard Hough transform (SHT) [1] has been used in various kinds of shape detection problems and proved to perform robustly even in situations where target shapes are occluded [4] [5] [6]. However, a major drawback of SHT lies in its relatively high computational cost and large storage requirement, especially for high-dimensional shapes (e.g. ellipses) [2].

The Randomized Hough Transform (RHT), proposed by Xu *et al.* [2], exploits the fact that a geometric shape can be uniquely determined with a small set of points sampled from the target shape. RHT is considered to be much more efficient and requires much less storage than SHT, but it alone is inapplicable to images where multiple target ellipses are overlapping with each other, since in this case there's a high probability that the points sampled from the image do not lie on the same ellipse, and thus results in false fitting. Various extensions to RHT have been suggested, such as Iterative RHT (IRHT) [7], Connective RHT (CRHT) [8] and Dynamic RHT (DRHT) [9]. But to the best of our knowledge, none of them is effective when the target ellipses are occluded with each other.

Another popular approach in ellipse detection is random sample consensus (RANSAC) proposed in [10], which is originally an iterative method for estimating parameters of a certain model using a set of data that might be contaminated by outliers. Cai *et al.* [11] proposed a contour-based approach that applied RANSAC algorithm to a set of candidate ellipses extracted from the image contours. Song *et al.* [12] improved Cai's method by exploiting the tangent directions of candidate points and introducing a parallel thinning algorithm. However, RANSAC-based approaches also have difficulty in detecting mutually-occluded ellipses because a *minimal sample set* needs to be randomly selected from the image, and for the case above, the sampled pixels will have high probability of coming from different ellipses.

III. Limitations of Randomized Hough Transform

A key difference between RHT and SHT is the voting scheme each uses to accumulate votes in the parameter space. SHT uses a *one-to-many* scheme meaning that each point in the image votes for multiple ellipses in the accumulator. In contrast, the way RHT works is to randomly sample a minimal set of points from the image, and then reconstruct the target shape using the point set based on the geometric properties of the shape. In other words, RHT uses a *many-to-one* voting scheme.

Thanks to the *many-to-one* voting scheme, RHT has some nice properties such as low computational complexity and small storage requirement compared with SHT [13]. But this particular voting scheme, on the other hand, also results in some limitations of RHT. First, its accuracy can be heavily reduced by outliers in noisy images. This issue has been addressed by Lu and Tan [7] and we refer readers to their paper for detailed discussion. Second, RHT will be problematic if applied directly to images where there exist multiple ellipses spreading out but without mutual occlusion. This issue however can be easily resolved by first applying connected components labeling to the image and then applying RHT to each single component. Third, RHT can be very inefficient or even fail for images where target ellipses are mutually-occluded, which is caused by the strict condition that an *effective* vote (here the effectiveness means that the vote actually goes to a target ellipse in the image) in RHT requires all three points in the sampled point set to come from the same target ellipse. Consider the following example. Suppose there are K ellipses in the image that are mutually-occluded. Let N_k denote the number of points that belong to the k-th ellipse. Then in each iteration, the probability that the three points are all sampled from the k-th ellipse is

$$P_k = \frac{\binom{N_k}{3}}{\binom{N}{3}} = \frac{N_k!(N-3)!}{N!(N_k-3)!} = \prod_{i=0}^{2} \frac{N_k - i}{N - i} \approx (\frac{N_k}{N})^3 \qquad (1)$$

where $N = \sum_{k=1}^{K} N_k$ denotes the total number of points. Furthermore, let M denote the maximum number of iterations allowed, and V denote the least number of votes needed for one ellipse to be detected in the accumulator, then the probability that RHT succeeds in detecting all K ellipses after M iterations is

$$Q \leq \binom{M}{VK} (\prod_{k=1}^{K} P_k^V)(1 - \sum_{k=1}^{K} P_k)^{M-VK} \qquad (2)$$

where the inequality holds because some ellipses are likely to get more than V votes out of the VK effective votes, which means we need more effective votes in order to detect the rest, thus the probability of successful detection becomes smaller. We can see that the magnitude of the right-hand side of the inequality is dominated by $\prod_{k=1}^{K} P_k^V$, which decreases drastically as K increases. In fact, from our experiments, we found that RHT is prone to fail even if K is small.

IV. Ellipse Detection with K-Means Clustering

In this section, we introduce a novel ellipse detection method that incorporates the basic idea of k-means clustering into Randomized Hough Transform in order to overcome its difficulty in detecting mutually-occluded ellipses (this method can be applied to detect non-occluded ellipses as well, but other existing methods might be a better choice in that case). The intuition is simple. As discussed in the previous section, the fundamental cause of RHT's failure in detecting mutually-occluded ellipses is that the point set it uses to fit an ellipse in each iteration is sampled blindly from the image. If instead, each point set is sampled from the same ellipse, then RHT will have no problem in detecting mutually-occluded ellipses. To this end, we introduce the use of k-means clustering algorithm.

In analogy to the conventional k-means clustering, our approach also consists of an initialization step followed by two iterative steps – clustering and parameter update. Additionally, we insert a step for measuring the fitness of our estimation right before the parameter update to ensure the convergence of our algorithm (see Section IV-D and IV-F for details). The overall flow chart of our algorithm is shown in Fig. 1.

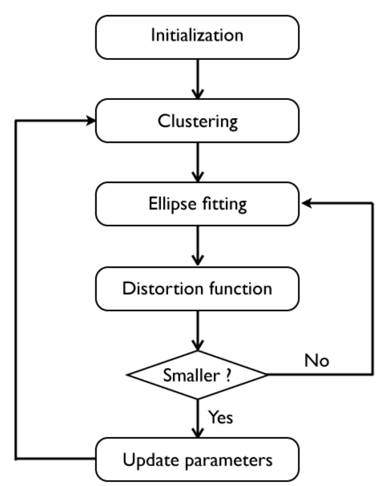

Fig. 1. Flow chart of our algorithm

A. Initialization

Parameters that need to be initialized include: 1) number of candidate ellipses K (similar to the number of clusters in k-means); and 2) Initial parameters for each candidate ellipse, including the length of major axis and minor axis, orientation and centroid coordinates[1]. Generally, all the ellipse parameters can be initialized with random values, but when it comes to implementation, we recommend to initialize the axis lengths with small values and ellipse centroids to be far away from each other in order to achieve better performance.

B. Clustering

Given K candidate ellipses, the clustering process proceeds as follows: for each pixel[2] in the image, (1) calculate its distance to each candidate ellipse; (2) label it as a member of the closest ellipse(s).

1) Distance Calculation: There can be many different ways to define the distance between a pixel and an ellipse. Here we present a simple yet effective method for the distance calculation:

1) Construct a line segment connecting the pixel and the ellipse centroid;
2) Find the intersection point of the line segment and the ellipse; and
3) The distance is defined as the Euclidean distance between the pixel and the intersection point.

A graphical example is shown in Fig. 2, where the distance between the pixel located in (p_x, p_y) and the ellipse centered at (c_x, c_y) is defined as:

$$d = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \tag{3}$$

where (q_x, q_y) is the intersection point.

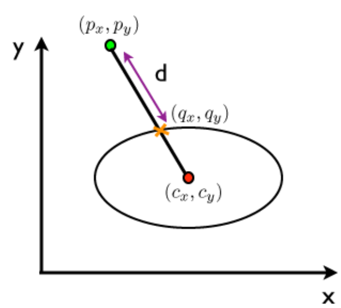

Fig. 2. Point-to-ellipse distance calculation

Note that the distance calculation should be performed for every pair of pixels and candidate ellipses. So if there are N pixels in the image, then we can construct a $K \times N$ matrix with each entry in the matrix indicating the distance between the pixel in its column and the ellipse in its row. To illustrate, consider a toy example that there are five pixels and three candidate ellipses in the image. The corresponding 3×5 distance matrix is shown in Fig. 3, with $d_{i,j}$ in each cell encoding the distance from the j-th pixel to the i-th candidate ellipse.

[1]For simplicity, we assume the orientation of target ellipses to be either 0 or 90 degrees. In other words, they are aligned either along the horizontal axis or along the vertical axis.

[2]We refer pixel to a nonzero or foreground pixel throughout the paper.

	P1	P2	P3	P4	P5
CE 1	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	$d_{1,4}$	$d_{1,5}$
CE 2	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	$d_{2,4}$	$d_{2,5}$
CE 3	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	$d_{3,4}$	$d_{3,5}$

Fig. 3. A 3 by 5 matrix storing the distance between each pair of the 5 pixels and 3 candidate ellipses.

2) Labeling: In this step, we assign each pixel to one candidate ellipse. Intuitively, a pixel is more likely to be sampled from the ellipse close by. Thus, we define the probability that the j-th pixel is sampled from the i-th candidate ellipse as

$$P_{ij} = \mathbf{N}(\frac{1}{d_{ij}}|0, \sigma^2) \tag{4}$$

where $\mathbf{N}(x|\mu, \sigma^2)$ denotes the probability density function of the Gaussian distribution with mean μ and variance σ^2.

Therefore, given the distance matrix computed from the previous step, one can simply assign each pixel to the ellipse with maximum sampling probability, or equivalently, the smallest distance. However, it's a good practice to assign the pixel to other ellipses as well if the maximum sampling probability is not significantly larger than the rest. For instance, in the case of two candidate ellipses, if $P_{11} = 0.55$ and $P_{21} = 0.45$, it is also very likely that pixel 1 is sampled from the second candidate ellipse, thus it should be assigned to both ellipses.

C. Ellipse Fitting with RHT

Up to this step, all the pixels have been clustered (labeled) based on their distances to the candidate ellipses. Now we can fit an ellipse for each pixel cluster using RHT by sampling the point set exclusively from the same cluster. Note that there's still no guarantee that pixels in the same cluster actually lie in the same ellipse, but in comparison with the original RHT, the likelihood has been significantly increased due to the k-means algorithm.

For completeness, here we briefly describe the general framework of RHT and refer the readers to [2] for more details. To fit an ellipse using RHT, the first step is to randomly sample a point set of three points from the image (in our case, the same pixel cluster). The ellipse centroid can be determined by the following procedure (see Fig. 4):

1) Find the tangents of the three points (X_1, X_2 and X_3) by performing least square fitting among a small neighborhood of pixels;
2) Determine the intersection points (T_1 and T_2) of the tangents;
3) Determine the midpoints of line segments $\overline{X_1X_2}$ and $\overline{X_2X_3}$ denoted by M_1 and M_2;
4) The ellipse centroid then lies on the intersection of $\overline{T_1M_1}$ and $\overline{T_2M_2}$

With the centroid found, we can now determine the length of the major and minor axes by substituting the coordinates

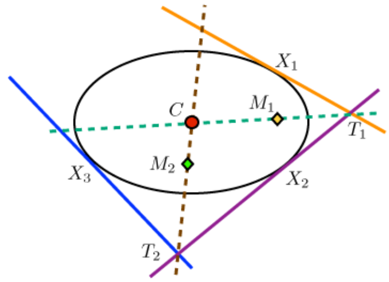

Fig. 4. Ellipse fitting with three points

of the three selected points X_1, X_2 and X_3 to the ellipse equation below and solve for a and b (again, note that we assumed before that the target ellipses align along either the horizontal or vertical axis):

$$\frac{(x-p)^2}{a^2} + \frac{(y-q)^2}{b^2} = 1 \tag{5}$$

where (p,q) is the centroid coordinate, and a and b are the axis lengths. These found parameters are then accumulated in the parameter space indicating the end of one iteration. The iterative process terminates when a stopping criterion is met or the maximum number of iterations allowed is reached. The algorithm then outputs the ellipses with most votes in the accumulator.

D. Distortion Function

In k-means clustering, the next step would be to update the candidate ellipses with the ones fitted from the previous step. However, it might happen that the newly fitted ellipses do not represent the target image better than the old candidate ellipses. Therefore, a measure for the fitness, which is called distortion function, is utilized to determine whether the update step should be performed. The distortion function D is defined as

$$D = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{kn} d_{kn} \tag{6}$$

where z_{kn} is an indicator variable that evaluates to 1 if the n-th pixel belongs to the k-th ellipse, and d_{kn} is the distance function defined in Section IV-B.1.

If the distortion function for the newly fitted ellipse set is larger than that of the old one, the following update step should not be performed. Instead, a new iteration cycle should start. But note that there's no need to repeat the clustering step if the candidate ellipses are not updated since the resulting pixel clusters will remain exactly the same; the new iteration should instead begin from the ellipse fitting step.

E. Update and Termination

Once a new set of ellipses is found to be of lower distortion to the target image, the candidate ellipses should then be replaced with the new ones.

So far, we've finished a complete cycle of one iteration. The next step is to determine whether to terminate the algorithm or start a new iteration. The algorithm terminates when either of the two criteria is met: (1) the maximum number of iterations allowed is reached; and (2) the algorithm has converged meaning that the candidate ellipses have not been updated for a certain number of iterations. Otherwise, if neither of the criteria is met, we start a new iteration from the clustering step.

F. Convergence

Since the candidate ellipses are updated only when the new estimation is a better fit for the target image, the distortion function will be monotonically decreasing during the iteration process, which guarantees the convergence of the algorithm after a certain number of iterations.

However, similar to the conventional k-means clustering, this approach is not guaranteed to converge to the globally optimal solution. Some initial configurations of the candidate ellipses might lead the algorithm to a local optima at termination. One way to alleviate this problem is to allow the candidate ellipses to update even if the distortion function is increased by some amount within a tolerance bound defined by D_{N-1}/N, where N is the number of iterations executed and D_{N-1} is the distortion of the previous iteration. In this way, the algorithm is less likely to get stuck at the local optima while the convergence is still guaranteed since as N increases, the tolerance bound will become so small that it is equivalent to the zero tolerance as before.

G. Analysis of the time complexity

To analyze the time complexity of the proposed approach, we adopt the following notations:

- T_{init} – time for initialization
- M – number of iterations
- N – number of pixels in the image
- K – number of candidate ellipses
- T_{dist} – time for calculating the distance from one point to one ellipse
- T_{RHT} – time for running the RHT in one cluster of pixels

Besides initialization, the time cost comes from the iterative process, where the major computational cost is comprised of two parts: (1) the distance calculation for every pair of pixel and candidate ellipse (remember that the calculation is carried out twice for each iteration: one for clustering the candidate ellipse set, which can be stored in memory for calculating distortion function later, and the other for calculating the distortion function for the newly fitted ellipse set); and (2) fitting ellipses with RHT. Thus we find the total time cost of the proposed approach to be upper bounded by

$$T_{total} \le T_{init} + 2MNKT_{dist} + MKT_{RHT} \tag{7}$$

The second term on the right-hand side of the inequality counts for the distance calculation, while the third term counts for the ellipse fitting with RHT. The equality holds only if the update step is carried out for every iteration, which

TABLE I

PERFORMANCE COMPARISON OF RHT AND OUR METHOD (K-RHT)

Image	# Ellipses		Accuracy	Time(s)
1	2	RHT	99.6%	2.81
		k-RHT	99.0%	6.24
2	3	RHT	81.5%	8.68
		k-RHT	98.8%	8.60
3	4	RHT	67.6%	20.82
		k-RHT	99.2%	13.25
4	4	RHT	69.3%	37.28
		k-RHT	96.4%	14.63
5	4	RHT	77.1%	40.64
		k-RHT	97.9%	13.82
6	5	RHT	45.4%	64.55
		k-RHT	92.7%	16.29

means that the distortion function of the newly fitted ellipse set is smaller than that of the candidate ellipse set for every iteration.

Since the initialization step costs a relatively small amount of time, and $T_{RHT} \ll NT_{dist}$ for most cases, the right-hand side of the inequality is dominated by the second term. Furthermore, with N, K being small integers and T_{dist} being a constant, we can therefore draw the conclusion that the overall time complexity is $O(N)$. In other words, the time complexity of the proposed approach scales linearly with the image size.

V. EXPERIMENTAL RESULTS

We implemented the proposed approach in Matlab, and compared its performance against standard RHT on a set of synthetic images containing mutually-occluded ellipses (Fig. 6). In the experiments, we initialized the minimum required peak votes V in RHT as 10 and the maximum number of iterations allowed as 1000. We also imposed a maximum of 30 iterations for our method. The final results are obtained by running each algorithm for 5 epochs and from which pick the one with the best performance (in the sense of a tradeoff between the accuracy and time cost) to compare (Table I). The accuracy in Table I was obtained by comparing the parameter values found by each algorithm with the ground-truth values. Figure 5 shows how the distortion function for the candidate and new ellipse sets changed over each iteration.

From the results, we can see that the performance of RHT both in accuracy and time efficiency drops drastically if the target ellipses are complexly occluded with each other, whereas our method performs quite well overall regardless of the way and the number of ellipses are occluded as long as a significant portion of each elliptical contour appears in the image.

We also applied our method to the bicycle images from the Graz-02 dataset [14] [15] in detecting the wheels of bicycles. Note that in real-world applications, the number of target ellipses K is not known beforehand. So instead of using a single value for K, we ran our algorithm with K ranging from 1 to 6, and from all the experiments pick the one with the smallest distortion function. The results are shown in Fig. 7.

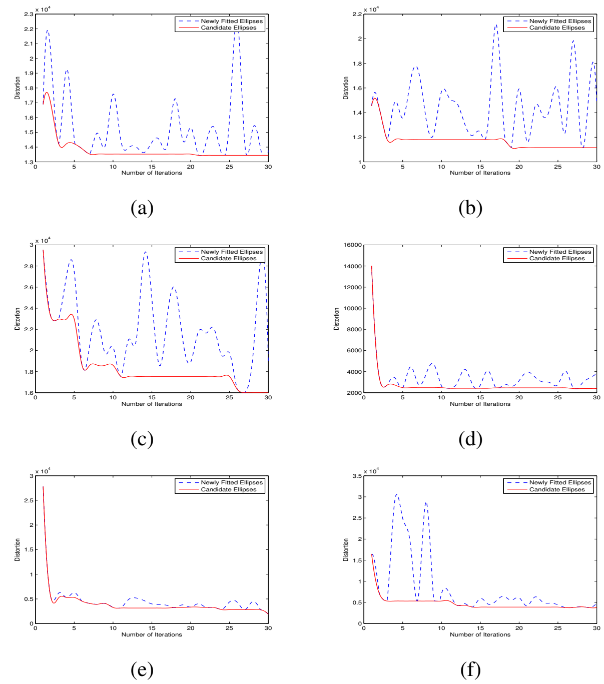

Fig. 5. (a) – (f) illustrates the corresponding distortion function change of test images 1 – 6. The blue dashed lines indicate the distortion of the new ellipse set fitted in every iteration, and red solid lines indicate the distortion of the candidate ellipse set.

VI. CONCLUDING REMARKS

We present in this paper a novel method that enhances the performance of Randomized Hough Transform in detecting mutually-occluded ellipses based on k-means clustering. Experiments with a set of synthetic images with mutually-occluded ellipses showed that our method outperforms RHT by a large margin in the aspects of accuracy and time efficiency. However, one weakness of our method is its susceptibility to being stuck at a local optima, which results from the nature of k-means clustering. Since k-means is essentially a particular case of the $Expectation-Maximization$ algorithm with "hard" labels, it might be an interesting research direction in the future to extend the proposed method to the more general EM framework. But some difficulties need to be addressed first: 1) If a soft label is used to assign the membership of each point, how do we determine the point set that is used to fit each candidate ellipse? If we simply use the point set for which an ellipse is most "responsible", then the result will be similar to our method; 2) Since applying the EM algorithm in general will lead to very high computational complexity, is there a way to implement it efficiently in the problem setting of ellipse detection? Whether there's a way to resolve these issues is open to question.

VII. ACKNOWLEDGEMENTS

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract #911NF-08-1-0463 (Proposal 55111-CI), the Center for Transportation Studies at UMN, and the

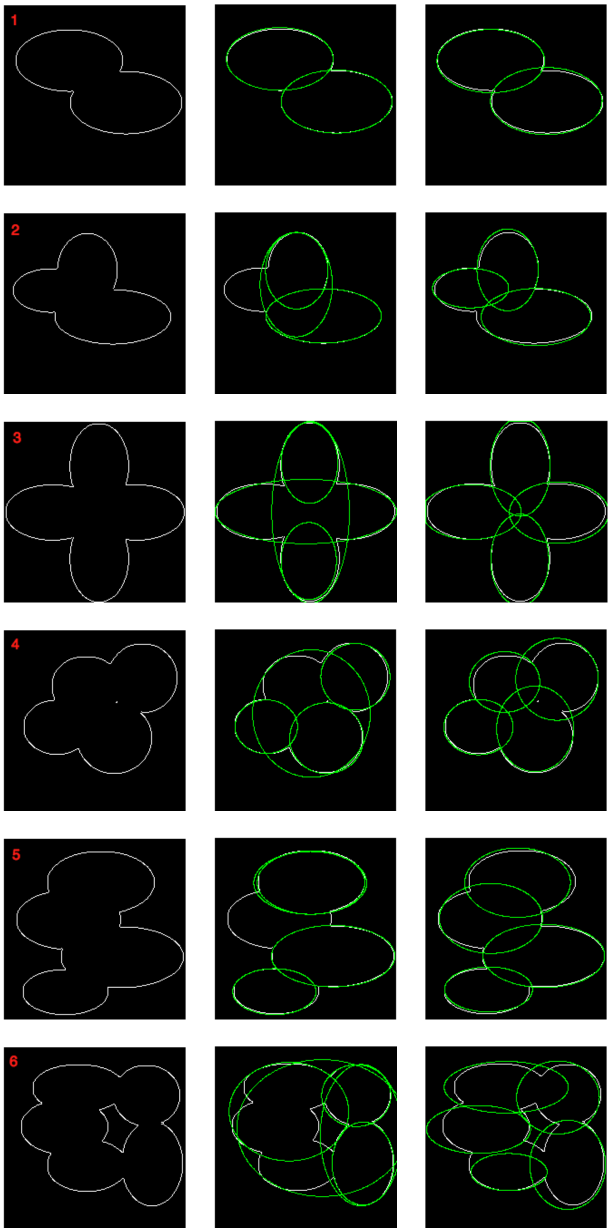

Fig. 6. Synthetic images used for comparison between RHT and our method. The first column are the input images. The second and third columns are detection results obtained by RHT and our method respectively.

National Science Foundation through grants #IIP-0443945, #CNS-0821474, #IIP-0934327, #CNS-1039741, and #SMA-1028076.

REFERENCES

[1] P. V. C. Hough, "Method and means for recognizing complex patterns," *U.S. Patent 3069654*, 1962.

[2] L. Xu, E. Oja, and P. Kultanan, "A new curve detection method: Randomized Hough Transform (RHT)," *Pattern Recog. Lett.*, vol. 11, pp. 331–338, 1990.

[3] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, August 2006.

[4] J. Illingworth and J. Kittler, "Survey of the hough transoform," *CVGIP*, vol. 44, pp. 87–116, 1988.

[5] Y. Zhang and R. Webber, "Windowing approach to detecting line segments using hough transform," *Pattern Recognition*, vol. 29, pp. 255–265, 1996.

Fig. 7. Application examples on bicycle images. The first column are input images from the Graz-02 dataset; The second and third columns are resulting images after performing segmentation (provided in the dataset) and edge detection on the original images; The last column are the results of ellipse detection by our method.

[6] A. Kesidis and N. Papamarkos, "Window-based inverse hough transform," *Pattern Recognition*, vol. 33, pp. 1105–1117, 2000.

[7] W. Lu and J. Tan, "Detection of incomplete ellipse in images with strong noise by iterative randomized Hugh transform (IRHT)," *Pattern Recognition*, vol. 41, pp. 1268–1279, 2008.

[8] H. Kalviainen and P. Hirvonen, "An extension of the randomized Hough transform exploiting connectivity," *Pattern Recognition Letters*, vol. 18, no. 1, pp. 77–85, 1997.

[9] H. Kalviainen, P. Hirvonen, L. Xu, and E. Oja, "Probabilistic and non-probabilistic Hough transforms: overview and comparisons." *Image and Vision Computing*, vol. 13, no. 4, pp. 239–252, 1995.

[10] M. A. Fischler and R. C. Bolles, "A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, pp. 381–395, 1981.

[11] W. Cai, Q. Yu, and H. Wang, "A fast contour-based approach to circle and ellipse detection," *Proceedings of the 5th World Congress on Intelligent Control and Automation*, June 2004.

[12] G. Song and H. Wang, "A fast and robust ellipse detection algorithm based on pseudo-random sample consensus," *Lecture Notes in Computer Science*, vol. 4673, pp. 669–676, 2007.

[13] L. Xu and E. Oja, "Randomized hough transform (rht): basic mechanisms, algorithms, and computational complexities," *CVGIP: Image Understanding*, vol. 57, pp. 131–154, 1993.

[14] M. Marszalek and C. Schmid, "Accurate object localization with shape masks," *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, "Generic object recognition with boosting," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 28, no. 3, March 2006.