Randomized Hough Transform for Ellipse Detection with Result Clustering

Cosmin A. Başca, Mihai Taloş, and Remus Brad, *Member, IEEE*

Abstract — **Our research is focused on the development of robust machine vision algorithms for pattern recognition. We want to provide robotic systems the ability to understand more on the external real world. In this paper we describe a method for detecting ellipses in real world images using the Randomized Hough Transform with Result Clustering. A preprocessing phase is used in which real world images are transformed – noise reduction, greyscale transform, edge detection and finaly binarization – in order to be processed by the actual ellipse detector. The ellipse detector filters out false ellipses that may interfere with the final results. Due to the fact that usually more "virtual" ellipses are detected for one "real" ellipse, a data clustering scheme is used, the clustering method, classifies all detected "virtual" ellipses into their corresponding "real" ellipses. The post processing phase is VQ similar and it also finds the actual number of classes unknown a priori.**

Keywords — **Adaptive Threshold, Clustering, Ellipse Detection, Hough Transform.**

I. INTRODUCTION

ELLIPSES are the most common features that appear in images, (most often circular shapes appear as ellipses due to the projection transform, only spherical objects remain as circles – e.g. balls – when projecting 3D space to 2D space). The Hough technique is particularly useful for computing a global description of a feature(s) (where the number of solution classes need not be known *a priori*), given (possibly noisy) local measurements. The Hough Transform is widely used for parametric curve detection. A *generalized* Hough (Ballard [1]) transform can be employed in applications where a simple analytic description of a feature(s) is not possible. Due to the computational complexity of the generalized Hough algorithm, this variant of the Hough Transform is rarely used in practice. We will only focus on detecting ellipses using the Hough Transform. The traditional approach for ellipse detection using the Hough technique is similar to line or circle detection (Duda and Hart [3]). The parametric equation of a line is written as follows:

$$x\cos\theta + y\sin\theta = r \qquad (1)$$

C. A. Başca is a Msc. Student at the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania (phone: 40-269-216062/461; e-mail: cosmin.basca@gmail.com).

M. Taloş has graduated the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania (e-mail: mihai.talos@gmail.com).

R. Brad is an Associate Professor at the Faculty of Engineering, "Lucian Blaga" University of Sibiu, Romania (phone: 40-269-216062/207; fax 40-269-214294; e-mail: remus.brad@ulbsibiu.ro).

The Hough Transform is actually a map from *xy* space to the parameter space $r\theta$. In order to find all the lines in a given image we would have to solve the given parametric equation for known x and y coordinates. The process of ellipse detection is similar but evidently more complex and more time consuming as the parametric equation would be of the following form:

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0 \qquad (2)$$

We know have to find all five parameters of the equation in order to detect a valid ellipse. Evidently this approach would require solving the equation for five different points on the ellipse, thus mapping *xy* points to a five dimensional space, (hence having to manage a five dimensional accumulator). This approach is not only memory expensive but also computationally intensive, as the algorithm in its brute search form would have a $O(n^5)$ complexity. Although there are refined variations of this approach (see Ballard[1]) these are still expensive considering memory or computational time.

In order to overcome this inconveniencies we have further developed on the idea of Chellali[2], Fremont[2] and Czervinski[2] who use only one dimensional accumulator for ellipse voting, and reducing algorithm complexity.

II. ALGORITHM DESCRIPTION

A. Preprocessing Phase

The basic information flow is as described in Fig. 1. As it can be seen in Fig. 1 the input for the detector block is a binarized image obtained by applying an adaptive threshold (we use the Maximum Variance Threshold as it varies with luminosity) to the gradient image. This step can be further refined, because the detector is highly dependent on its output. All edge pixels are white (the foreground) and the rest are black (background). All contours (features) hence ellipse contours as well should be well defined and noise should be greatly reduced.

B. Ellipse Filtering and Detection Phase

In order to detect all five parameters of an ellipse only three points are needed, two of which are considered to be the ellipses vertices. We will not detail Chellali[2], Fremont[2] and Czervinski[2] algorithm as it is not the porpoise of this paper, but we will mention the basics of it as it is necessary in order to get an understanding of the modifications we made.

Given the two vertices of the ellipse the determination of four of the five ellipse parameters is straightforward using the following formulas:

$$x_0 = \frac{x_1 + x_2}{2} \quad (3)$$

$$y_0 = \frac{y_1 + y_2}{2} \quad (4)$$

$$a = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{2} \quad (5)$$

$$\alpha = \tan(\frac{y_2 - y_1}{x_2 - x_1}) \quad (6)$$

x_0 and y_0 being the ellipses center coordinates a (please note that a from formula (5) is not the same as a from formula (2)) being half of its major axis and α being the orientation of the ellipse.

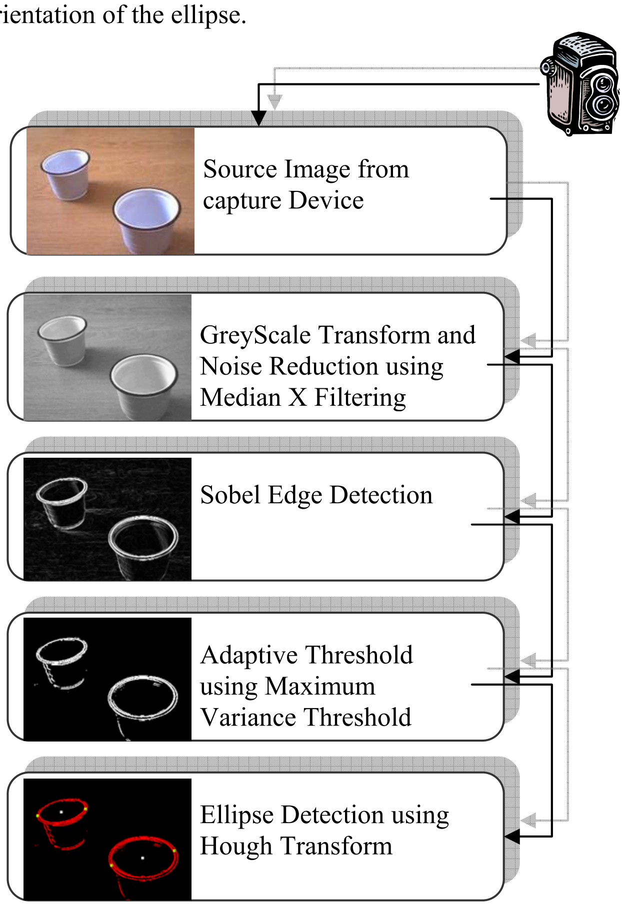


Fig 1. Processing steps involved in the algorithm.

A third point is needed, in order to determine the fifth parameter of the ellipse – the half length of its minor axis - using the following approximation formulas:

$$b = \sqrt{\frac{a^2 d^2 \sin^2 \tau}{a^2 - d^2 \cos^2 \tau}} \quad (7)$$

$$\cos \tau = \frac{a^2 + d^2 - f^2}{2ad} \quad (8)$$

where d, τ and f can be observed in Fig. 2.

Based on formulas (3) to (8) all ellipses in the image can be detected.

The algorithm as proposed by Chellali[2], Fremont[2] and Czervinski[2] has a $O(n^3)$ complexity. The complexity of the algorithm can be further reduced by transforming the method into a Randomized Hough Transform (RHT).

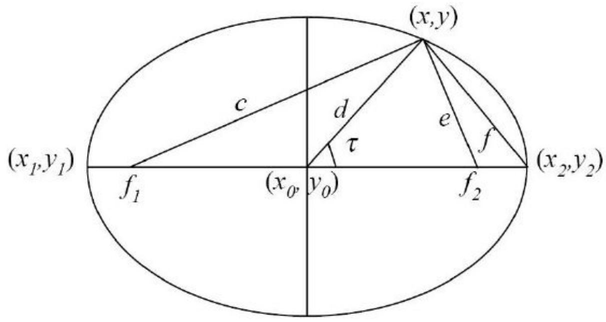


Fig 2. Ellipse geometry, f_1 and f_2 are the ellipses foci.

In the first step we randomly select m pairs of points satisfying the search domain condition (the number of points initially selected is considerably lower than in the original implementation – we only select m = n*C where C = 1, 2, 3,… instead of selecting n*n) thus reducing the complexity to $O(\frac{n^2}{C})$, where as mentioned before C is a constant much smaller than n. The number of selected pairs must be sufficiently large in order to detect all present ellipses, in our current implementation C = 2.

When an ellipse is detected the points located on its contour are not deleted from the search array, because this significantly lowers the performance of the algorithm –e.g. suppose a false ellipse is detected intersecting the real ellipse because of noise, removing its edge points would mean that we would also have to remove edge points that are present on the contour of the real ellipse, thus reducing its quality and its chances of being detected.

Further filtering methods are introduced to filter out false ellipses. We only consider an ellipse to be valid is it has points distributed on both sides of the major axis, furthermore we check to see if the number of points is proportional on either side of the ellipse (when we refer to the side of the ellipse we mean the points located on its one side of its contour).

To improve ellipse detection, the accumulator is quantized, resulting in a thicker ellipse contour. Having more points on its contour, the digitization problem is overcome, because a more complete elliptic contour can be approximated with formulas (3) to (8).

C. Post Processing Phase, Result Clustering

Due to the fact that after thresholding the original image the contours of the ellipses are thick, for one real ellipse many ellipses appear to be in the same place with slightly different parameters, so the result of the detection phase must be clustered, to obtain the real ellipses.

Because we have no a priori knowledge of how many real ellipses are in the source image, standard clustering techniques such as LVQ or K-means can not be applied because they are highly dependent on the initial number of clusters.

Our approach is similar to VQ. We calculate a similarity distance between two ellipses using ellipse feature vectors. The feature vectors consist of the following:

$$V(x_0, y_0, a, b, \alpha) \qquad (9)$$

The measured distance is the Euclidian distance in this five dimensional feature space. The distance is calculated using the following formula:

$$D(V, W) = \sqrt{\sum_{i=1}^{5} (p_{iv} - p_{iw})^2} \qquad (10)$$

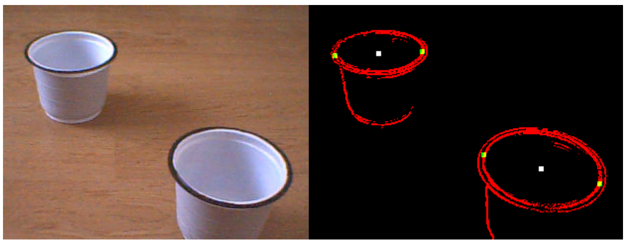
where $V(p_{1v}, p_{2v}, p_{3v}, p_{4v}, p_{5v})$ and W are vectors of the type (9).

If the distance is above a certain similarity threshold – which is experimentally chosen (we used D_T=20) – the ellipse is considered to be different from the compared one, otherwise the two compared ellipses are said to match.

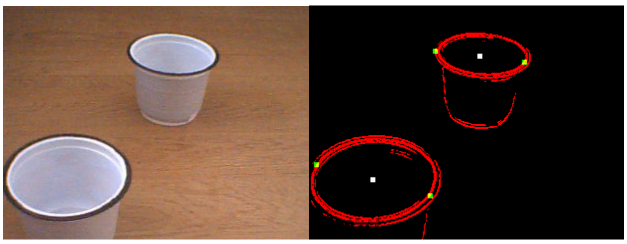
Each time a different ellipse is found a new cluster is formed. Using a single pass algorithm all the detected ellipses are assigned to the clusters to which they belong. In the comparison process the centroid of the cluster is used as the representative ellipse of the group, but when outputting the result the ellipse that is most near to the centroid of the group is considered to be the detected ellipse.

III. RESULTS

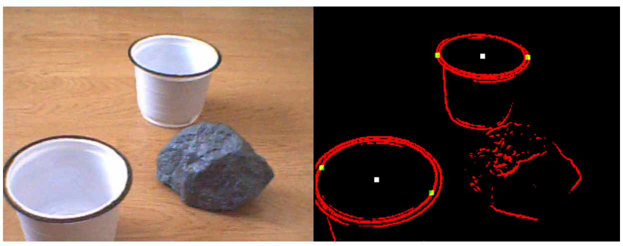
The performance of the algorithm is demonstrated using real world images. We present four cases of real ellipse detection either with or without foreign objects (that represent noise for the ellipse detector).



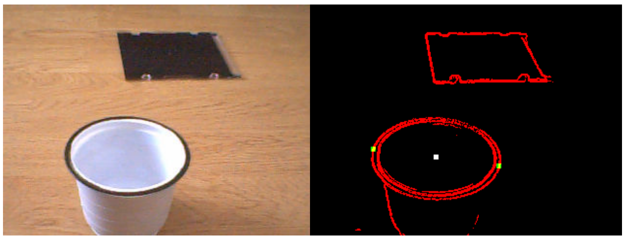
(a)



(b)



(c)



(d)

Fig. 3. Experimental Results.

Output of the ellipse detector can be seen in table 1.

TABLE 1: DETECTED ELLIPSES PARAMETERS

Fig.	Center x (pixels)	Center y (pixels)	Major Axis (pixels)	Minor Axis (pixels)	Alfa (rad)
3(a)	235.5	164.5	60	37	0.253
3(a)	71.0	49.0	44	22	0.045
3(b)	65.5	174.5	59	40	0.276
3(b)	173.0	51.5	45	22	0.122
3(c)	66.0	173.5	58	43	0.222
3(c)	176.0	50.0	47	22	0.021
3(d)	129.0	155.5	64	40	0.133

Output of the post processing phase can be seen in table 2. Ellipse quality represents the threshold of a high pass filter used to filter out false ellipses – we refer to ellipse quality meaning the number of points on its contour.

TABLE 2: DETECTED ELLIPSES STATISTICS

Fig.	Virtual Ellipses	Real Ellipses	Ellipse Quality	Search Point Pairs	Total Edge Points
3(a)	48	2	200	6428	3214
3(b)	42	2	200	6530	3265
3(c)	16	2	230	8198	4099
3(d)	44	1	300	6238	3119

IV. CONCLUSION

The method is applicable in real time applications, native implementations (both in hardware and software) can be very fast.

It is also robust regarding false ellipse detection.

It classifies correctly the detected ellipses as being similar or different.

As further improvements, a more advanced adaptive method for clustering is desired, and to also improve ellipse detection and filtering using context information.

The search step can be pseudo-random, ellipse positions could be predicted using local and global prediction tables or neural networks.

REFERENCES

[1] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.

[2] R. Chellali, V. Fremont, P. J. Czervinski, “Ellipse Detection Using Hough Transform”, Warsaw University of Technology, Ecole des Mines des Nantes, 2003.

[3] R. Duda and P. Hart, “Use of the Hough transform to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11-15, 1972.

[4] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities*. New York: Academic Press, 1990.

[5] A. Rosenfeld, *Picture Processing by Computer*. New York: Academic Press, 1969.

[6] S. D. Shapiro, “Transformations for the noisy computer detection of curves in noisy pictures,” *Computer Graphics and Image Processing*, vol. 4, pp. 328-338, 1975.

[7] S. D. Shapiro, “Properties of transforms for the detection of curves in noisy image,” *Computer Graphics and Image Processing*, vol. 8, pp. 219-236, 1978.

[8] S. D. Shapiro, “Feature space transforms for curve detection,” *Pattern Recognition*, vol. 10, pp. 129-143, 1978.

[9] P. K. Sahoo, S. Soltani, A. C. Wong, and Y. C. Chen, “A survey of thresholding techniques,” *Computer Vision, Graphics, and Image Processing*, vol. 41, pp. 233-260, 1988.

[10] J. Kittler, J. Illingworth, and J. Foglein, “Threshold selection based on a simple image statistic,” *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 125-147, 1985.

[11] S. U. Lee, S. Y. Chung, and R. H. Park, “A comparative performance study of several global thresholding techniques for segmentation,” *Computer Vision, Graphics, and Image Processing*, vol. 52, pp. 171-190, 1990.

[12] I. Sobel, *Camera Models and Machine Perception*. PhD thesis, Stanford University, Stanford, CA, 1970.

[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. New York, NY: John Wiley & Sons, 1973.

[14] W. Pratt, *Digital Image Processing*. New York: John Wiley, 1978.

[15] G. Ritter, J. Wilson, and J. Davidson, “Image algebra: An overview,” *Computer Vision, Graphics, and Image Processing*, vol. 49, pp. 297-331, Mar. 1990