ELSEVIER

Pattern Recognition Letters 19 (1998) 299–305

Pattern Recognition Letters

Randomized Hough Transform: Improved ellipse detection with comparison [1]

Robert A. McLaughlin [2]

Centre for Intelligent Information Processing Systems, The University of Western Australia, Nedlands, WA 6907, Australia

Received 30 September 1997

Abstract

We describe an algorithm for the detection of ellipse shapes in images, using the Randomized Hough Transform. The method is compared with three other Hough-based algorithms. Tests are performed using both noise-free and noisy images, and several real-world images. The algorithm was found to give improvements in accuracy, and a reduction in computation time and the number of false alarms detected. A program allowing the reader to experiment with these algorithms can be found at the WWW address http://ciips.ee.uwa.edu.au/Papers/Journal_Papers/1998/01/Index.html.

Keywords: Randomized Hough Transform; Hough Transform; Ellipse detection

1. Introduction

Image processing programs are frequently required to detect ellipses. Examples include cell counting in the detection of breast cancer and particle classification in an industrial setting. A standard algorithm for this detection is the Hough Transform, as implemented by Yuen et al. (1989), or a probabilistic variant using the principles outlined by Kiryati et al. (1991). Both of these algorithms are slow, memory intensive and have a limited accuracy as the number of ellipses in the image increases.

The Hough Transform has also been used for the detection of lines and circles. Here the same flaws are evident but to a lesser degree. To alleviate these problems, the Randomized Hough Transform (RHT) (Xu et al., 1990; Xu and Oja, 1993) was introduced. Unfortunately, RHT becomes impractical when the object to be detected is defined as a nonlinear equation of a set of parameters. Ellipses are one such object.

This paper presents an algorithm which reduces this to a linear problem. This allows the use of RHT for ellipse detection. We test the algorithm against three Hough-based methods, with both synthetic and real-world images. We also make implementations for all algorithms tested in this paper available on the World Wide Web.

In this paper we work with black images on a white background. The term ''pixel'' refers to a black pixel.

[1] Electronic Annexes available. See http://www.elsevier.nl/locate/patrec.

[2] Email: ram@ee.uwa.edu.au.

0167-8655/98/$19.00
PII S0167-8655(98)00010-5

2. Randomized Hough Transform

RHT (Xu et al., 1990) accumulates points in a parameter space by randomly choosing n-tuples of pixels from an image, and computing the parameters of the object which passes through these pixels. It is useful for the detection of objects which are defined as a linear function of a set of parameters. To quote Xu et al. (1990), pp. 333–334,

... for curves expressed by equations which are nonlinear with respect to the parameters, the RHT cannot be directly used.

An ellipse is defined as the solution to the equation

$$a(x-p)^2 + 2b(x-p)(y-q) + c(y-q)^2 = 1 \quad (1)$$

with the restriction that $ac - b^2 > 0$. This is nonlinear with respect to the parameters (p,q,a,b,c). To find the unique ellipse passing through a 5-tuple of pixels in the image would require that we solve a system of 5 simultaneous nonlinear equations. Having to solve this system of nonlinear equations for each 5-tuple of pixels chosen will slow the RHT and render it impractical. The following algorithm reduces this to a linear problem.

We begin by choosing three pixels from the image and obtaining estimates of the tangent at each pixel. This is done by defining a small neighbourhood around the pixel and finding the line of best fit to those pixels within the neighbourhood (using least squares method). The three pixels $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$ and the estimates of tangent form the input to the algorithm. Computing the parameters of the corresponding ellipse will be organised into two sections: finding the centre, and finding the remaining three parameters.

To find the ellipse centre we use a feature of ellipse geometry noted by Yuen et al. (1989). Take two points on an ellipse and find their midpoint $\boldsymbol{m}$, and the intersection of their tangents $\boldsymbol{t}$. The ellipse centre will lie on the line $\overline{\boldsymbol{tm}}$ (Fig. 1). To understand this, consider it for the simple case of a circle. Then note that the transformation mapping a circle onto an ellipse is affine, thus maps lines to lines, tangents to tangents and intersections to intersections.

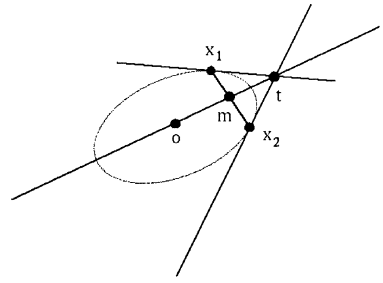

Fig. 1. Ellipse geometry. o = ellipse centre.

We compute this line using pixels $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, then again with pixels $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$. The intersection of the two lines gives us an estimate of the ellipse centre.

We next estimate the remaining three ellipse parameters. First translate the ellipse to be centred on the origin. This reduces Eq. (1) to the following:

$$ax^2 + 2bxy + cy^2 = 1.$$

Substituting in the coordinates of the three pixels $(\boldsymbol{x}_1 = (x_1,y_1),\ \boldsymbol{x}_2 = (x_2,y_2),\ \boldsymbol{x}_3 = (x_3,y_3))$, we obtain the following system of three simultaneous linear equations:

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 \\ x_2^2 & 2x_2y_2 & y_2^2 \\ x_3^2 & 2x_3y_3 & y_3^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}.$$

Solving this for a, b, c gives the remaining three ellipse parameters.

Next, we check the inequality $ac - b^2 > 0$ which is true if the parameters represent a valid ellipse. If false, it implies that either the three pixels do not lie on the same ellipse, or that the estimates of tangent were inaccurate. In either case, we discard the parameters and choose a new pixel triplet.

Finally, we convert from parameters (p,q,a,b,c) to the parameters (p,q,r_1,r_2,θ) where r_1, r_2 are the major and minor radii of the ellipse (respectively) and θ is the angle of the major axis. From experiment, we found this parameter space to be better behaved than (p,q,a,b,c). That is, the points representing a range of ellipses were spread more evenly throughout the parameter space.

In common with Xu et al. (1990), the histogram typically used in Hough algorithms has been replaced with an ordered linked list structure. This list

structure has proven more efficient than the 5-D histogram which would otherwise be required. The list only contains entries for regions of the parameter space which are used by the current image. In contrast, a histogram would contain many entries that are set to zero. Also, when using a histogram, it is necessary to decide ahead of time which region of the parameter space the histogram will cover. The larger this region, the more memory required. The smaller this region, the less likely it is to include those ellipses present in the image. In contrast, the list structure accepts entries from anywhere in the parameter space. Further implementation details are given elsewhere (McLaughlin, 1997).

The algorithm outlined above shares certain features in common with the Dynamic Generalized Hough Transform (DGHT) (Leavers, 1992). The algorithms differ in that the DGHT also requires a segmentation of the image into individual ellipses based on connectivity of horizontal, vertical or diagonal pixels, combined with assumptions of the expected size of the ellipses being detected. In addition, the DGHT requires that a 5-tuple of pixels be used to define a single ellipse. The algorithm used here requires only a triplet of pixels. The required extra information is contained in the estimates of tangent.

3. Experiment

A software package was developed to test the algorithms on randomly generated ellipses, in a similar style to that used in the comparison of line algorithms detailed by Hare and Sandler (1993). The domain of interest for this experimental comparison is accuracy and computation time. We compare the following algorithms:

RHT Randomized Hough Transform.

SHT The Hough-based ellipse detection algorithm outlined by Yuen et al. (1989).

PHT A Probabilistic Hough Transform (Kiryati et al., 1991; Bergen and Shvaytser, 1991) implementation of the algorithm of Yuen et al. (1989).

GSHT Method using Geometric Symmetry (Ho and Chen, 1995).

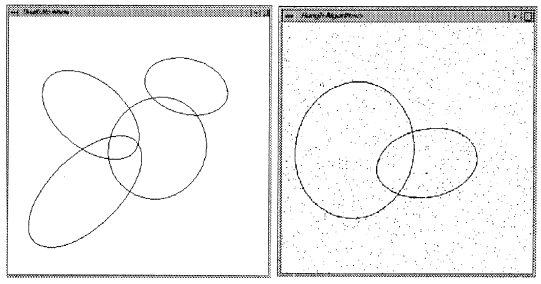

Fig. 2. Test images: Multiple noise-free ellipses (left) and image with 1% speckle noise (right).

3.1. Experiment 1: Multiple ellipses

Four-hundred images were produced, fifty of which contained one ellipse, fifty with two ellipses, etc. up to fifty images with eight ellipses. A typical image is shown in Fig. 2. The ellipses often intersected. We recorded the number of ellipses correctly found, the number of false alarms and the processing time.

3.2. Experiment 2: Speckle noise

Next we tested the tolerance of the algorithms to uniform, speckle noise. Fifty images, each containing two ellipses were produced. We then randomly selected 0.5% of the pixels and inverted them (changing black to white, white to black). This was repeated with 1.0%, 1.5%, ... up to 4% of the pixels. A typical noisy image is shown in Fig. 2.

3.3. Experiment 3: Real-world images

We applied the algorithms to three real-world images. The first image contained several eggs

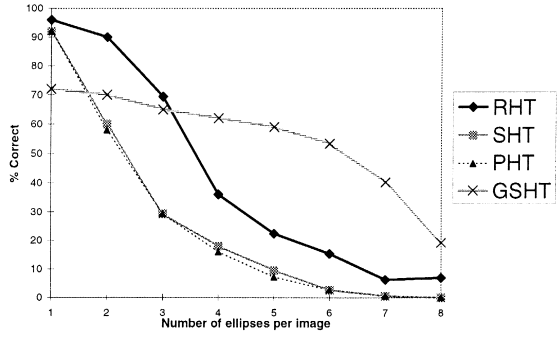

Fig. 3. Percentage accuracy with multiple ellipses.

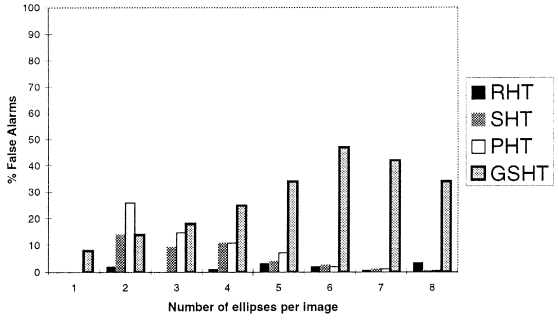

Fig. 4. Percentage of false alarms.

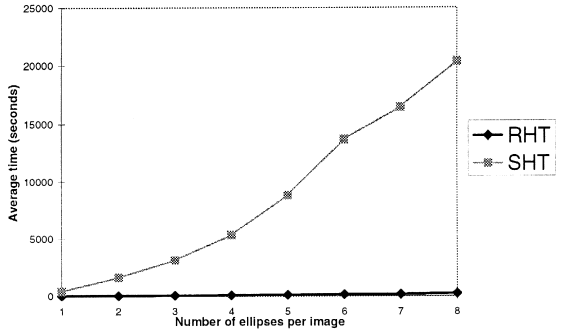

Fig. 6. Average computation time in seconds. RHT and SHT.

(Fig. 8) as may be found on a production line. The second image shows a collection of crockery (Fig. 10) and the third is of cells in a breast cancer sample (Fig. 12).

4. Results

4.1. Experiment 1: Multiple ellipses

Fig. 3 shows the percentage of ellipses correctly found, plotted against the number of ellipses in the image. RHT has the highest accuracy for images containing few ellipses. GSHT proved better with more complicated images.

We plot the total number of false alarms found in each set of images in Fig. 4. A false alarm is recorded when a non-existent ellipse is incorrectly identified as existing. RHT proved to be robust against false alarms.

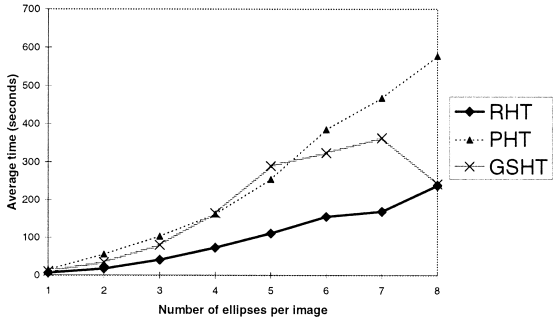

Fig. 5. Average computation time in seconds. RHT, PHT and GSHT.

Figs. 5 and 6 plot the average computation time (in seconds) against the number of ellipses in the image. RHT proved to be the fastest algorithm. Because the computation time for SHT was so much greater than that of the other three algorithms, we have presented this as two graphs. If all four algorithms were plotted on the same graph, the differences between RHT, PHT and GSHT would become obscured.

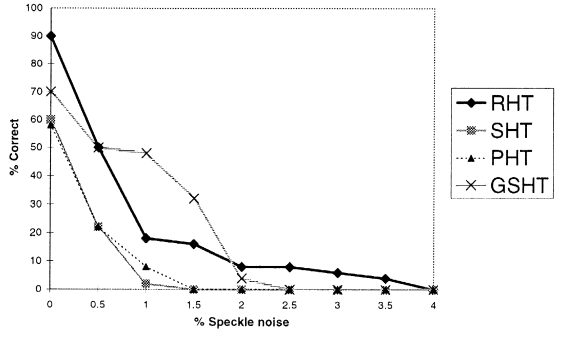

Fig. 7. Accuracy in the presence of speckle noise.

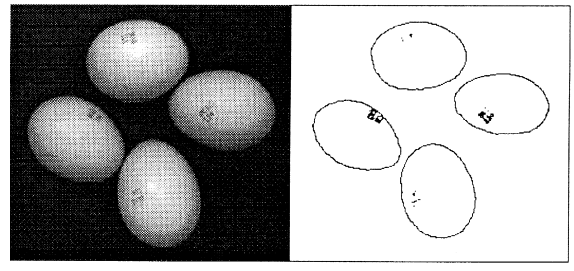

Fig. 8. Eggs: Original image (left), edge detected image (right).

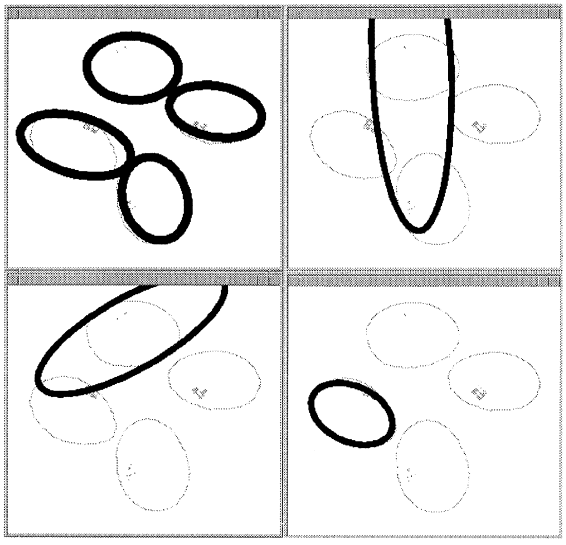

Fig. 9. Eggs. RHT (top left), SHT (top right), PHT (bottom left), GSHT (bottom right).

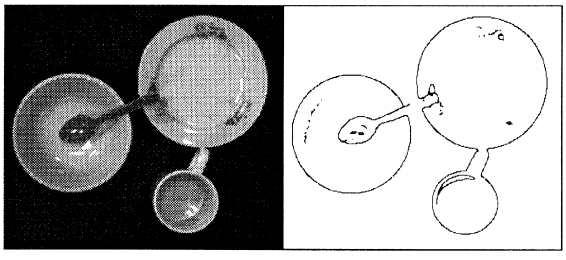

Fig. 10. Plates: Original image (left), edge detected image (right).

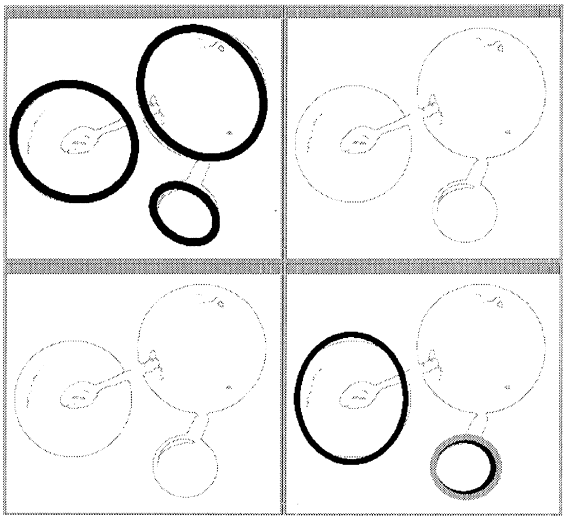

Fig. 11. Plates. RHT (top left), SHT (top right), PHT (bottom left), GSHT (bottom right).

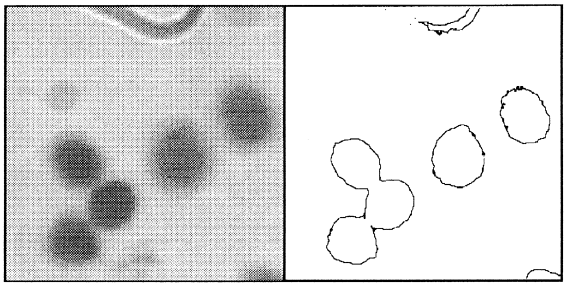

Fig. 12. Breast cancer cells: Original image (left), edge detected image (right).

4.2. Experiment 2: Speckle noise

Fig. 7 graphs the percentage of ellipses correctly found, against the percentage of speckle noise present in the images. All four algorithms degraded rapidly as noise levels increased.

4.3. Experiment 3: Real-world images

Figs. 8, 10 and 12 show three real-world images. On the left of each figure is the original image and on the right is the edge image which was used for ellipse detection. Results of the four algorithms are shown in Figs. 9, 11 and 13 as follows:

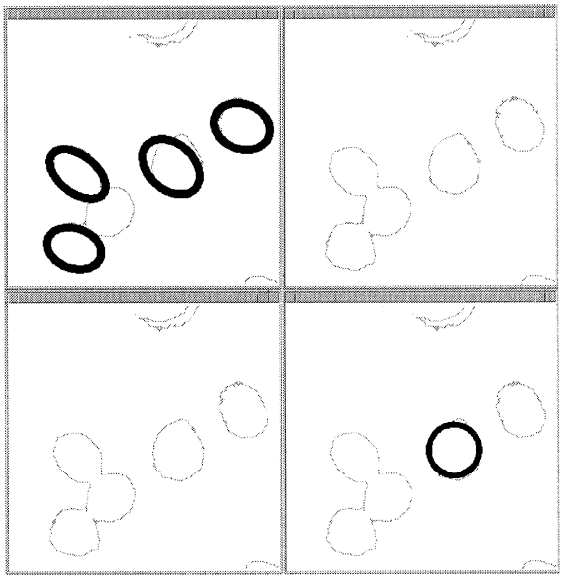

Fig. 13. Breast cancer cells. RHT (top left), SHT (top right), PHT (bottom left), GSHT (bottom right).

- RHT – top left image.
- SHT – top right image.
- PHT – bottom left image.
- GSHT – bottom right image.

Images where no ellipses are drawn indicate that no ellipses were detected. Note that in Fig. 11, GSHT mistakenly found two ellipses centred on the cup.

5. Discussion

RHT was found to have a higher accuracy than both SHT and PHT in both noise-free images with multiple ellipses (Fig. 3) and noisy images (Fig. 7). In images with few ellipses or little noise, it proved superior to GSHT. RHT was also found to be less subject to false alarms (Fig. 4). Moreover, RHT proved to be faster than either SHT, PHT or GSHT (Figs. 5 and 6). Specifically, when compared with SHT, RHT was several hundred times faster.

To explain these results, consider the different mechanisms at work in the algorithms. A standard strategy used to reduce the high computational and memory demands of the Hough transform is to organise the recognition into several simpler stages, each involving a low dimensional Hough Transform. This is the approach adopted in both SHT and PHT. In the first stage, a 2-D Hough Transform is used to produce a set of candidate ellipse centres. This collection will include spurious points incorrectly identified as ellipse centres. It will also include inaccuracies in the estimates of the position of the centres.

In the second stage, an estimate is made of the remaining three ellipse parameters for each candidate ellipse centre. This estimate will be nonsensical where the candidate ellipse centre was an incorrectly chosen spurious point. Note that knowledge gained in estimating the remaining three ellipse parameters is not used to refine the position of the ellipse centre. That is, information from this stage is not fed back to the first stage of processing. Thus, if the candidate ellipse centre was inaccurately positioned, this will lead to inaccuracies in estimates of the remaining three parameters. We have found from experiment that small inaccuracies in the position of an ellipse were magnified in estimates of the remaining three parameters.

These problems in the above method can be avoided by use of a single stage, 5-D Hough Transform. The computational and memory requirements of such an algorithm can be made practical by introducing a method of sub-sampling the parameter space. This is the approach used in RHT.

When sampling from a population (in this case, points in the parameter space), the sample is typically drawn according to a uniform probability density function (pdf) defined over the population. With RHT, we are implementing what could be considered a more intelligent form of sampling. By ''intelligent'', we mean that our sampling is guided by knowledge of the object to be recognised (an ellipse). Rather than explicitly defining a pdf over the parameter space, we define a uniform pdf over the collection of black pixels in the image. Three distinct pixels are randomly chosen, and these are then mapped to a point in the parameter space. In this way, we implicitly define a pdf over the parameter space.

For an image containing few ellipses and little noise, this pdf will favour points in the parameter space which describe the ellipses in the image. Because of this, a small sample is sufficient to generate the appropriate clusters in the parameter space.

As with any sampling technique, under certain conditions the sample will be poorly chosen. For a particular sampled point, this occurs when the three pixels do not come from the same ellipse. Consider an image containing n ellipses of equal size, each consisting of p pixels. Let event A be defined as ''a 3-tuple of randomly chosen pixels come from the same ellipse''. Then the probability of A is given by:

$$\mathrm{P}[A] = n\frac{p(p-1)(p-2)}{np(np-1)(np-2)} \approx \frac{1}{n^2}.$$

As we increase n, the number of ellipses in the image, less of the 3-tuples of pixels will come from a single ellipse. Thus our sample of the parameter space will consist increasingly of spurious points, with less points located near the parameters of ellipses in the image. In *Experiment 1: Multiple ellipses*, this manifested itself as a decreased accuracy as the number of ellipses in the image increased (Fig. 3).

Uniform speckle noise gives rise to the same problems. In *Experiment 2: Speckle noise*, speckle

noise was generated uniformly over the entire image. As only a very small proportion of the total pixels in the image formed ellipses, even a small percentage of noise led to a rapid decrease in accuracy (Fig. 7).

Like SHT and PHT, the GSHT algorithm organises ellipse detection into two stages: centre finding and estimation of the remaining three ellipse parameters. During the first stage, it uses certain features of ellipse geometry to attempt to segment the image into separate ellipses. It then utilises a three dimensional Hough transform to identify the remaining ellipse parameters. Although some tolerance was coded into the implementation used here, the segmentation typically fails with non-ideal ellipses. Successful segmentation with ideal ellipses is reflected in the stable level of accuracy shown in Fig. 3 for images containing multiple ellipses. As with SHT and PHT, inaccuracies in the estimation of an ellipse centre typically lead to failure in the second stage of the algorithm.

6. Conclusions

In this paper, we have proposed an algorithm for ellipse detection using the Randomized Hough Transform (RHT). We considered the task of finding the unique ellipse passing through an n-tuple of pixels in the image. This was reduced to a linear problem by making use of a feature of ellipse geometry.

We then compared the resulting method against three other Hough-based algorithms. RHT was found to have a higher accuracy than the other algorithms with noise-free images containing few ellipses. In was also less subject to false alarms. RHT gave considerable improvements in computation time, especially when compared against SHT. The accuracies of all four algorithms were comparable in the presence of noise. Finally, the algorithms were tested with several real-world images. RHT proved superior with all three images.

We have described RHT as an implementation of a 5-D Hough Transform with sub-sampling of the parameter space. The sampling method is guided by knowledge of the geometry of ellipses. We also demonstrated why the sampling degrades as the number of ellipses present in the image increases. In contrast, the other algorithms organise the recognition task into several stages. The resulting improvements in computation time and memory requirements are offset by a reduced accuracy under certain conditions.

References

Bergen, J.R., Shvaytser, H., 1991. A probabilistic algorithm for computing Hough transforms. J. Algorithms 12, 639–656.

Hare, A.R., Sandler, M.B., 1993. General test framework for straight-line detection by Hough transforms. In: Proc. 1993 IEEE Internat. Symp. on Circuits and Systems, vol. 1. IEEE, pp. 239–242.

Ho, C.-T., Chen, L.-H., 1995. A fast ellipse/circle detector using geometric symmetry. Pattern Recognition 28 (1), 117–124.

Kiryati, N., Eldar, Y., Bruckstein, A.M., 1991. A probabilistic Hough transform. Pattern Recognition 24 (4), 303–316.

Leavers, V.F., 1992. The dynamic generalized Hough transform: Its relationship to the probabilistic Hough transforms and an application to the concurrent detection of circles and ellipses. CVGIP: Image Understanding 56 (3), 381–398.

McLaughlin, R.A., 1997. Randomized Hough Transform: Improved ellipse detection with comparison. Tech. Rept. No. TR97-01. Centre for Intelligent Information Processing Systems, University of Western Australia. (Available online from http://ciips.ee.uwa.edu.au/Papers/)

Xu, L., Oja, E., 1993. Randomized Hough Transform (RHT): Basic mechanisms, algorithms, and computational complexities. CVGIP: Image Understanding 57 (2), 131–154.

Xu, L., Oja, E., Kultanen, P., 1990. A new curve detection method: Randomized Hough Transform (RHT). Pattern Recognition Lett. 11 (5), 331–338.

Yuen, H.K., Illingworth, J., Kittler, J., 1989. Detecting partially occluded ellipses using the Hough transform. Image and Vision Comput. 7 (1), 31–37.