Available online at www.sciencedirect.com

SCIENCE DIRECT®

Pattern Recognition Letters 27 (2006) 652–657

Pattern Recognition Letters

www.elsevier.com/locate/patrec

# Circle detection on images using genetic algorithms

Victor Ayala-Ramirez *, Carlos H. Garcia-Capulin, Arturo Perez-Garcia, Raul E. Sanchez-Yanez

*Universidad de Guanajuato FIMEE, Electronics and Communications Engineering Department, Tampico 912, Colonia Bellavista, Guanajuato, 36730 Salamanca, Mexico*

Received 23 December 2004; received in revised form 6 October 2005
Available online 28 November 2005

Communicated by G. Borgefors

**Abstract**

In this paper, we present a circle detection method based on genetic algorithms. Our genetic algorithm uses the encoding of three edge points as the chromosome of candidate circles $(x, y, r)$ in the edge image of the scene. Fitness function evaluates if these candidate circles are really present in the edge image. Our encoding scheme reduces the search space by avoiding trying unfeasible individuals, this results in a fast circle detector. Our approach detects circles with sub-pixellic accuracy on synthetic images. Our method can also detect circles on natural images with sub-pixellic precision. Partially occluded circles can be located in both synthetic and natural images. Examples of the application of our method to the recognition of hand-drawn circles are also shown. Detection of several circles in a single image is also handled by our method.


*Keywords:* Object recognition; Circle detection; Genetic algorithms; Hand-drawn shape location

## 1. Introduction

Shape detection is needed in many computer vision tasks because shape is an important cue for modelling objects in scenes (da Fontoura Costa and Marcondes Cesar Jr., 2001). Object location problems are mainly solved by two types of techniques: In one hand, deterministic techniques include application of Hough transform based methods (for example, Yuen et al., 1990), geometric hashing and template or model matching techniques (Iivarinen et al., 1997; Jones et al., 1990). In the other hand, stochastic techniques include random sample consensus techniques (Fischer and Bolles, 1981), simulated annealing and genetic algorithms (GA) (Roth and Levine, 1994).

Template and model matching techniques were the first approaches to shape detection. Shape coding techniques and combination of shape properties were used to represent objects. Plenty of methods have been developed to solve the shape detection problem (Peura and Iivarinen, 1997). Main drawback of these techniques is related to the contour extraction step from real images. Additionally, it is difficult for models to deal with pose invariance except maybe for very simple objects.

Hough transform based methods (Muammar and Nixon, 1989) have shown to be robust enough to deal with noisy images. However, Hough transform techniques are not efficient for high dimensional problems because they are computationally or memory intensive even if they have a robust behavior. In order to overcome these problems, some other researchers have proposed new approaches to the Hough transform. We can find in the literature, for example, the probabilistic Hough transform (Shaked et al., 1996), the randomized Hough transform (RHT) (Xu et al., 1990), the fuzzy Hough transform (Han et al., 1993). We can even find alternative transforms as the proposed by Becker et al. (2002).

* Corresponding author. Tel.: +52 464 6480911; fax: +52 464 6472400.
*E-mail address:* ayalav@salamanca.ugto.mx (V. Ayala-Ramirez).

0167-8655/$ - see front matter 
doi:10.1016/j.patrec.2005.10.003

Stochastic techniques can attain a good solution quickly without the needing of high memory or computation power requirements. Nevertheless, they can not guarantee to find an optimal solution. Genetic programming, in particular GA, are used to optimize a fitness function by mimicking natural evolution for organisms. Individuals for this evolution are computational representations of potential solutions for the problem to be solved. Each individual is represented as a binary string also known as a computational chromosome. The entire set of individuals examined at a time is called the population. In particular, genetic algorithms have been recently used for some shape detection tasks: Roth and Levine (1993, 1994) has proposed to use GA for primitive extraction on images. A further improvement on Roth approach has been presented by Lutton et al. (1994). Yuen and Ma (2000) uses GA for template matching when the pattern has been the subject of an unknown affine transformation.

Circle and ellipse detection problem has attracted many researchers. Most of them use Hough transform based techniques. For example, Lam and Yuen (1996) propose an approach based on hypothesis filtering and Hough transforms to detect circles. Yuen and Lo (1994) propose to use a multi-resolution approach for the circle detection problem. Mainzer (2002a,b) uses circle detection in a traffic sign detector. Rosin and Nyongesa (2000) use a soft computing approach to shape classification. For the ellipses, Rosin proposes an ellipse fitting algorithm that uses five points (Rosin et al., 1997) and he has extended to fit data on superellipses (Zhang and Rosin, 2003). Yao et al. (2004) propose to use a multiple population GA to detect ellipses.

In our work, we present a GA-based circle detector. Our system uses a three edge point circle representation that lets the system to reduce the search space by eliminating unfeasible circle locations in our image. This approach results in a sub-pixellic circle detector that can detect circles in real images even when the circular object has a significative occluded portion. We present the results of the application of our method to synthetic images, real-world images and hand-drawn circles.

This paper is organized as follows: In Section 2, we formulate our approach and we also present the main characteristics of the GA method used to detect circles in images. Section 3 shows our test protocol and the results of applying our method to synthetic and real-world images. Our conclusions are presented in Section 4, we also discuss here some future work to be done.

## 2. Circle detection using GA

### 2.1. Our approach

We represent circles by using the parameters of the second degree equation passing through three edge points in the edge space of the image. Images are preprocessed by using an edge detection step. We store locations for all the edge points. The detected edge points are the only potential candidates to define circles in our image, taken in triplets. We need an edge detection method that obtains a single pixel contour for objects in the image. We have used a classical Sobel detector as implemented by Matlab software. All the edge points in the image are stored in a vector array $V = \{v_0, v_1, \ldots, v_{N_e}\}$ with $N_e$ the total number of edge points in the image. We store the $(x_i, y_i)$ coordinates for each edge pixel $v_i$ in the edge vector. We encode each individual as the concatenation of the binary codes of the indexes $i_1$, $i_2$ and $i_3$ of three edge points. We consider the circle passing through the points $v_{i_1}, v_{i_2}, v_{i_3}$ as a potential solution to the detection of circles in images. A number of these candidate solutions are generated for the initial population of the GA. We evolve this solution by using genetic operators in the simple GA approach. When a given threshold is obtained in the fitness evolution of the population, best individual is taken as the solution of the circle detection problem.

Even if RHT for circle detection uses also three edge points to cast a vote for the corresponding point in the parameter space, to obtain a subpixel resolution (as our method is capable to detect for circle parameters), RHT would require huge amounts of memory and it would spend a lot of computational time. The evidence collecting step used in RHT is also performed in our method but it is done along the evolution process, so fitness function improves at each generation by discriminating non plausible circles.

In the following paragraphs, we detail the different steps used to formulate the circle detection problem as a GA optimization problem.

### 2.2. Individual representation

Each individual $C$ uses three edge points as chromosomes. In this representation, edge points are stored as an index to their relative position in the edge array $V$ of the image. That will encode an individual as the circle that passes through the three points $v_i$, $v_j$ and $v_k$.

We represent each circle $C$ by the three parameters $x_0$, $y_0$ and $r$. With $(x_0, y_0)$ being the $(x, y)$ coordinates of the center of the circle and $r$ being its radius. We can compute the equation of the circle passing through the three edge points as:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \tag{1}$$

with:

$$x_0 = \frac{\begin{vmatrix} x_j^2 + y_j^2 - (x_i^2 + y_i^2) & 2(y_j - y_i) \\ x_k^2 + y_k^2 - (x_i^2 + y_i^2) & 2(y_k - y_i) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \tag{2}$$

$$y_0 = \frac{\begin{vmatrix} 2(x_j - x_i) & x_j^2 + y_j^2 - (x_i^2 + y_i^2) \\ 2(x_k - x_i) & x_k^2 + y_k^2 - (x_i^2 + y_i^2) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))} \tag{3}$$

and

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{4}$$

We can then represent the shape parameters (for the circle, $[x_0, y_0, r]$) as a transformation $T$ of the edge vector indexes $i$, $j$, $k$.

$$[x_0, y_0, r] = T(i, j, k) \tag{5}$$

with $T$ being the transformation composed of the previous computations for $x_0$, $y_0$ and $r$.

Using the indexes as computational chromosomes, we can sweep a continuous space for the shape parameters but keeping a binary string representation of the GA individual. This approach reduces the search space by eliminating unfeasible solutions.

### 2.3. Fitness evaluation

In order to compute the fitness value of an individual $C$, we compute the coordinates of the edge points in a virtual sampled shape. We then validate if they actually exist in the feature space. The test set for these points is $S = \{s_1, s_2, \ldots, s_{N_s}\}$ with $N_s$, the number of test points where the existence of an edge border will be seek.

The test point set $S$ is generated by the uniform sampling of the shape boundary. In our case, $N_s$ test points are generated around the circumference of the candidate circle. Each point $s_i$ is a 2D-point where its coordinates $(x_i, y_i)$ are computed as follows:

$$x_i = x_0 + r \cdot \cos \frac{2\pi i}{N_s} \tag{6}$$

$$x_i = x_0 + r \cdot \sin \frac{2\pi i}{N_s} \tag{7}$$

Fitness function $F(C)$ accumulates the number of expected edge points (i.e. the points in the set $S$) that actually are present in the edge image. That is:

$$F(C) = \frac{\sum_{i=0}^{N_s-1} E(x_i, y_i)}{N_s} \tag{8}$$

with $E(x_i, y_i)$ being the evaluation of edge features in the image coordinate $(x_i, y_i)$ and $N_s$ being the number of pixels in the perimeter of the circle corresponding to the individual $C$ under test. As the perimeter is a function of the radius, this serves as a normalization function with respect to the radius. That is, $F(C)$ measures completeness of the candidate circle encoded by the computational chromosome. Our objective is then to maximize $F(C)$ because a larger value implies a better response to this circularity operator. We can stop the optimization process by fixing a number of epochs to evolve the chromosomes or by satisfying a completeness threshold. Each of these procedures implies some a priori knowledge about the specific application context of the circle detector. In our tests, we have fixed a number of epochs for population evolution.

Table 1
List of parameters for the genetic algorithms used in the circle detection problem

| Parameter | Value |
|---|---|
| Population size | 70 |
| Crossover probability | 0.55 |
| Mutation probability | 0.10 |
| Number of elite individuals | 2 |
| Selection method | Roulette wheel |
| Crossover method | 1-point crossover |

There is also a penalty cost for small radius circles in order to favor detection of larger circles. This penalty cost is applied by multiplying $F(C)$ by a factor $f(r)$ defined as follows:

$$f(r) = \begin{cases} \frac{k}{r^*} r & \text{if } r < r^* \\ 1.0 & \text{otherwise} \end{cases} \tag{9}$$

with $r*$ a threshold depending on the specific application of the circle detector.

### 2.4. Genetic operators and GA parameters tuning

We have developed a generic library for genetic algorithms implementation. Functions included in this tool are described in (Perez-Garcia et al., 2004). Our implementation uses three genetic operators: selection, crossover and mutation.

For the selection operator we have used a roulette wheel implementation. In this method, each individual is assigned a slice in the roulette wheel. The size of the slice is proportional to its normalized fitness value. This selection strategy favors best fitted individuals but it gives a chance to the less fitted individuals to survive. A steady state policy is also used, this implies to have elite individuals that will be kept in the next population.

For the crossover operator, we have used a 1-point crossover implementation. This method uses a crossover probability value to determine where we will divide the genetic material of the two parents to recombine into a new individual.

A bit mutation operator is used in our implementation. In this method, a bit is inverted or not depending on a mutation probability.

We have tuned the population size by running our circle detection algorithm on an image with a circle and salt and pepper noise. We have chosen the population sizes that minimizes detection time. After experimental tuning, GA parameters were configured as shown in Table 1.

## 3. Experimental work

Experimental tests have been developed in order to evaluate the performance of the circle detection environment. Test platform was implemented in C language, on a PC using an AMD XP processor running at 2.07 GHz. These experiments mainly address tasks such as

(1) circle localization,
(2) shape discrimination,
(3) hand written circle detection and
(4) multiple circle localization.

A set containing more than an hundred of synthetic and natural images has been used. Error analysis and elapsed time during experiments are also discussed for the system.

### 3.1. Circle detection tests

#### 3.1.1. Synthetic images

The experimental setup includes the use of 20 synthetic images of $256 \times 256$ pixels. We have generated images with only one randomly located circle in them. Parameters of interest are the $(x, y)$ center of the circle position and its radius $r$.

The algorithm has been run 100 times for each test image. We can say that the average error for localization is 0.06 pixels and the worst case (maximum error) is 0.31 pixels. That is, the proposed algorithm finds the circle parameters in a sub-pixel level and with an error lower than a tenth of pixel. The detection is robust to translation and scale and the elapsed time is reasonably low (typically under 5 ms).

#### 3.1.2. Natural images

Circle detection on natural images is discussed now. Twenty images of $640 \times 480$ pixels form the test set. The 8 bits depth monochromatic images are captured using a digital camera. Each natural scene include a circle shape among other objects. Images are preprocessed using an edge detection algorithm and each output image is fed into the genetic algorithm-based detection. A particular case among these 20 is shown on Fig. 1.

Due to the absence of ground truth reference data, results on detection have been statistically analyzed for comparison purposes. We have computed averages and standard deviation for the detected parameters for a 100 run test for each of a 10 images set. Standard deviation is in the 0.3 pixels range for the tests. These are good results, considering the existing noise, irregularities in the shape and inclusive, the partial occlusion of circles in some cases. Again, the system is robust to translation and scale in the field of view.

Fig. 1. Circle detection on a natural image. Detected circle is shown as an overlay.

### 3.2. Circle discrimination tests

The ability of the system on the circle detection from a set of diverse shapes is discussed here. Firstly, five synthetic images of $256 \times 256$ pixels are considered. For each test image, the algorithm in run 100 times and both, the number of right detections and the elapsed time are registered. A maximum of 500 generations is the limit for the detection process. A number of different shapes are present in the images, as indicated on Table 2. A sample image is the one in Fig. 2 (labeled as 5 in Table 2).

The same experiment was repeated using natural images. An example of such an image is Fig. 3. Table 3 resumes the system performance.

From Tables 2 and 3 we can say that circles are detected an 100% in most cases and we have a 92% as the worst detection percent. That is, circle discrimination from other shapes is feasible in natural images. Translation and change of scale are well handled by the system.

Table 2
Discrimination results on synthetic images

| Image | Shapes in image | Time (s) | Accuracy rate (%) |
|---|---|---|---|
| 1 | 2 | 1.64 | 100 |
| 2 | 3 | 1.56 | 100 |
| 3 | 4 | 0.28 | 100 |
| 4 | 12 | 3.81 | 92 |
| 5 | 11 | 2.39 | 100 |

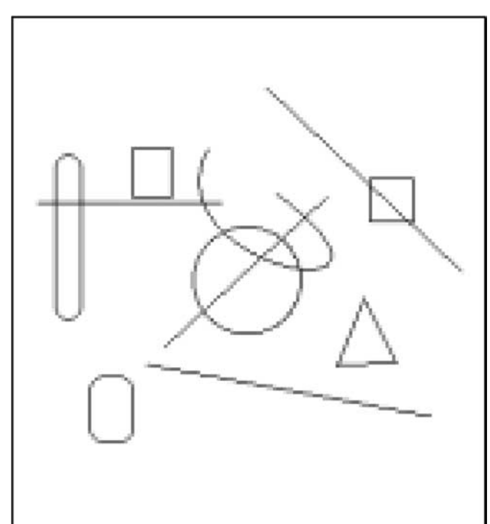

Fig. 2. Sample synthetic image with 11 different shapes.

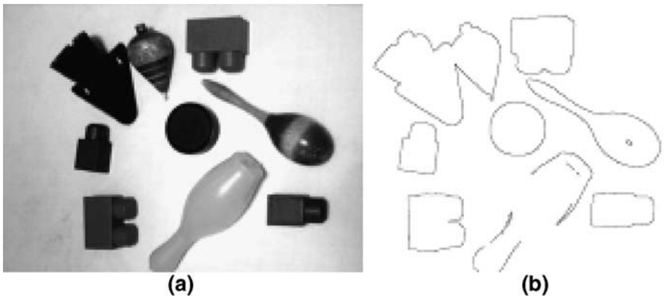


Fig. 3. Sample natural image with a variety of different shapes. (a) A sample input and (b) the corresponding edges.

Table 3
Discrimination results on natural images

| Image | Shapes in image | Time (s) | Accuracy rate (%) |
|---|---|---|---|
| 1 | 2 | 0.93 | 92 |
| 2 | 3 | 0.34 | 100 |
| 3 | 4 | 0.38 | 100 |
| 4 | 9 | 0.43 | 100 |
| 5 | 9 | 5.05 | 98 |

### 3.3. Hand-drawn sketch recognition

One of the main characteristics offered by the environment is the capability of detecting shapes deformed to a certain degree. This can be shown when processing images such as those in Fig. 4. We can say the system performance is adequate to both, discriminate the hand-drawn circle from the other shapes present in the image and determine the best approximation for this circle.

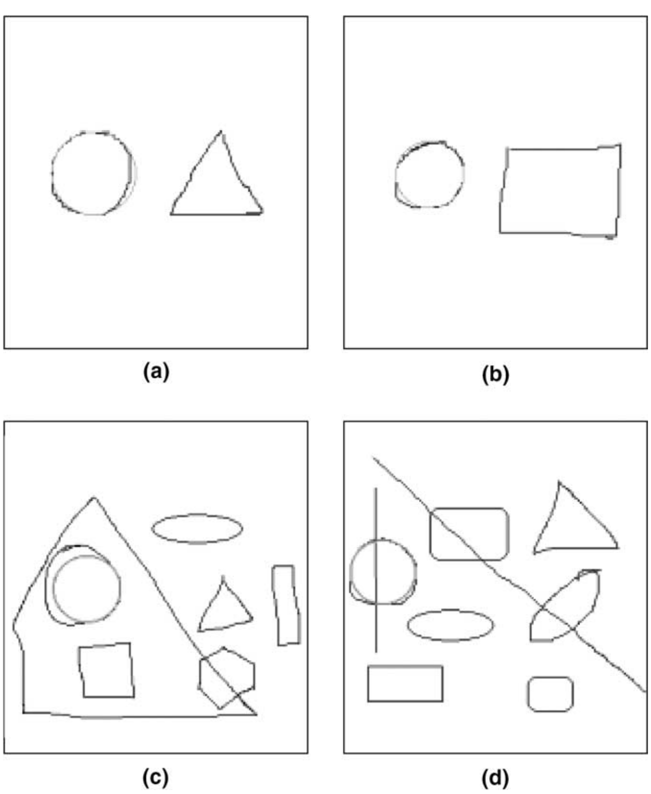


Fig. 4. Sample images for hand-drawn circles detection.

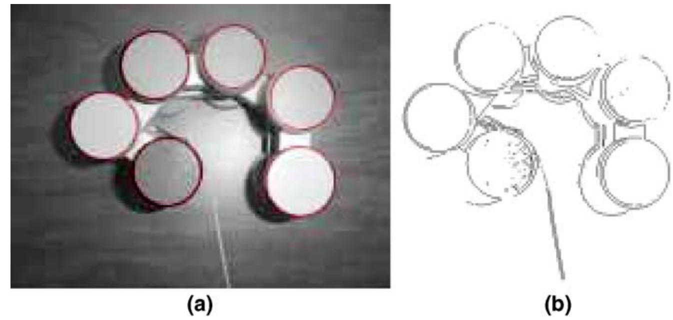


Fig. 5. Multiple circle detection on real-world images: (a) the original image with an overlay of the detected circles and (b) edge image obtained by the application of the Sobel operator.

The observed flexibility is due to the fact that the environment requires a minimum information for shape reconstruction. In this case, a minimum of edge points. Once a circle has been proposed, this shape is evaluated in order to see the concordance with the edge image. This leads to always finding the best approximation for a circle, be it present or not. This feature is the one that enables a successful detection when dealing with noisy, incomplete, or partially occluded circular shapes.

### 3.4. Multiple circle detection

Our method can also detect several circles on a real image. We need to set a maximal number of circular shapes to find. The method works in the original edge image until convergence for the first circle detection. This shape is then masked on the edge image and the GA circle detector is run on the modified image. This procedure is repeated until the maximum number of detection is attained. A validation of all the detected circles is then performed by analyzing continuity of the circumference segments detected as proposed by Kelly and Levine (1997). This procedure is necessary because we could ask to detect more circular shapes than the number of circles actually present in the image. The system can then provide a no circle response if none of the detected shapes satisfies circular completeness or also to detect the best circular shapes if there are more circular shapes in the image than the required number.

Fig. 5(a) shows a real-world image including several circular shapes on it. Detection results are shown overlaid on the image. In that example, we asked the algorithm to find the 6 best circular shapes. The edge image obtained after application of the Sobel mask used by our algorithm is shown in Fig. 5(b).

## 4. Conclusions and perspectives

In this paper, we have presented a GA-based circle detector. Our system is capable of detecting circles with sub-pixellic accuracy in synthetic images. We can also detect circles in real-world images with sub-pixellic precision. The encoding scheme used in this method encodes circles by using the equation for the ideal circle passing through three edge points. The search space is reduced because we only try the circles that can pass through the actual edge points in the scene. This results in a fast detector as compared to recent results in literature. Our circle detector can reliably detect circles even if they present significative occlusions, discontinuities or incompleteness. We show also that we can apply the same method to detect hand-drawn circles. Our system is also capable of detecting multiple circles on real images. The main drawback of our system arises when it deals with small circles. This problem is derived of the ability of our system to deal with imperfect circles, that is, when some circular sectors can be inscribed in a small shape, it will be detected as an imperfect but plausible circle. Behavior of our system with respect to small circles can be tuned according to the specific application context.

Given the performance characteristics of the method presented here, our future work will be directed towards applying this detector for part inspection in manufacturing and for artificial landmark detection in robotics.

## Acknowledgements

This work has been partially funded by the France–Mexico LAFMI Project "Concepción de funciones de percepción y planificación para la navegación topológica de un robot móvil en ambiente semi-estructurado interior o natural" and the Mexico PROMEP project "Navegación topológica de robots usando análisis difuso de textura".

## References

Becker J., Grousson S., Coltuc D., 2002. From Hough transforms to integral transforms. In: Proc. Int. Geoscience and Remote Sensing Symp., 2002 IGARSS'02, vol. 3, pp. 1444–1446.

da Fontoura Costa, L., Marcondes Cesar Jr., R., 2001. Shape Analysis and Classification. CRC Press, Boca Raton FL.

Fischer, M., Bolles, R., 1981. Random sample consensus: A paradigm to model fitting with applications to image analysis and automated cartography. CACM 24 (6), 381–395.

Han, J.H., Koczy, L.T., Poston, T., 1993. Fuzzy Hough transform. In: Proc. 2nd Int. Conf. on Fuzzy Systems, vol. 2, pp. 803–808.

Iivarinen, J., Peura, M., Sarela, J., Visa, A., 1997. Comparison of combined shape descriptors for irregular objects. In: Proc. 8th British Machine Vision Conf., Cochester, UK, pp. 430–439.

Jones, G., Princen, J., Illingworth, J., Kittler, J., 1990. Robust estimation of shape parameters. In: Proc. British Machine Vision Conf., pp. 43–48.

Kelly, M., Levine, M., 1997. Advances in Image Understanding. Finding and Describing Objects in Complex Images. IEEE Computer Society Press, pp. 209–225.

Lam, W., Yuen, S., 1996. Efficient techniques for circle detection using hypothesis filtering and Hough transform. IEEE Proc. Visual Image Signal Process. 143 (5), 292–300.

Lutton, E., Martinez, P., 1994. A genetic algorithm for the detection 2-D geometric primitives on images. In: Proc. of the 12th Int. Conf. on Pattern Recognition (ICPR'94), vol. 1, Jerusalem, Israel, pp. 526–528.

Mainzer, T., 2002a. Genetic algorithm for shape detection, Technical Report DCSE/TR-2002-06, University of West Bohemia in Pilsen, Pilsen, Czech Republic (March).

Mainzer, T., 2002b. Genetic algorithm for traffic sign detection. In: Proc. Int. Conf. Applied Electronics 2002, Pilsen, Czech Republic, pp. 129–132.

Muammar, H., Nixon, M., 1989. Approaches to extending the Hough transform. In: Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP'89, vol. 3, pp. 1556–1559.

Perez-Garcia, A., Ayala-Ramirez, V., Jaime-Rivas, R., 2004. A genetic algorithms generic library for application in optimization problems (in spanish). In: Proc. of the 5th Mexican Conf. on Robotics, pp. 141–146.

Peura, M., Iivarinen, J., 1997. Efficiency of simple shape descriptors. In: Arcelli, C., Cordella, L.P., di Baja, G.S. (Eds.), Advances in Visual Form Analysis. World Scientific, Singapore, pp. 443–451.

Rosin, P.L., Nyongesa, H.O., 2000. Combining evolutionary, connectionist, and fuzzy classification algorithms for shape analysis. In: Cagnoni, S. et al. (Eds.), Proc. EvoIASP, Real-World Applications of Evolutionary Computing, pp. 87–96.

Rosin, P.L., 1997. Further five point fit ellipse fitting. In: Proc. 8th British Machine Vision Conf., Cochester, UK, pp. 290–299.

Roth, G., Levine, M.D., 1993. Extracting geometric primitives. CVGIP: Image Understanding 58 (1), 1–22.

Roth, G., Levine, M.D., 1994. Geometric primitive extraction using a genetic algorithm. IEEE Trans. Pattern Anal. Machine Intell. 16 (9), 901–905.

Shaked, D., Yaron, O., Kiryati, N., 1996. Deriving stopping rules for the probabilistic Hough transform by sequential analysis. Comput. Vision Image Understanding 63, 512–526.

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

Yao, J., Kharma, N., Grogono, P., 2004. Fast robust GA-based ellipse detection. In: Proc. 17th Int. Conf. on Pattern Recognition ICPR'04, vol. 2, Cambridge, UK, pp. 859–862.

Yuen, S., Ma, C., 2000. Genetic algorithm with competitive image labelling and least square. Pattern Recognition 33, 1949–1966.

Yuen, H., Princen, J., Illingworth, J., Kittler, J., 1990. Comparative study of Hough transform methods for circle finding. Image Vision Comput. 8 (1), 71–77.

Yuen, K.S., Lo, E.K., 1994. A coarse-to-fine approach for circle detection. In: Proc. 1994 Int. Symp. on Speech, Image Processing and Neural Networks, Hong Kong, HK, pp. 523–526.

Zhang, X., Rosin, P.L., 2003. Superellipse fitting to partial data. Pattern Recognition 36, 743–752.