AN APPROACH FOR GENERIC DETECTION OF CONIC FORM

Maysa G. MACEDO[1,2] and Aura CONCI[1]
[1] Fluminense Federal University, Brazil
[2] Augusto Motta University Centre, Brazil

ABSTRACT: In this paper we introduce a unique methodology to detect any conic form by using Hough transform which is an established method for shape detection. This technique works based on features extraction for each conic separately and intend to unify all type of Hough detection in only one. Once we have the conic's parameters it is possible to build an accumulator array named parameter space where the most voted set of parameters is the chosen form. To build this accumulator, a range of values is set up and each value is a real parameter candidate. The length of which cell depends on the accuracy desirable and the dimension depends on the type of conic. This transform maps points from digital space to parameters space in order to match the point of the chosen form to the form in the real image. In this way it is much easier to reconstruct the conic form whenever it is necessary and to detect a form from artificial images or handmade drawing. Although each form has one different equation and number of parameters, we realized it follows a pattern and concluded that is possible to elaborate a generic equation and a methodology to detect any conic. The use of polar coordinates makes the generic detection easier because of the rotations variety detection. In addition, the order of search is important. We observed that the detection result is better if the searching for the correct rotation is the first step. Another relevant thing is the chosen length of bins in the accumulator which really defines a better match. This paper involves analysis about each conic equation, their similarities, number of parameters and how they can be introduced by a unique expression for Hough transform.

Keywords: Feature Extraction, Geometry, Hough Transform, Conics.

1. INTRODUCTION

Detecting objects in an image is very useful in many applications [13] [17]. The Hough transform is one way of doing this. Its idea is transform an image in a digital space (x,y) to represent it in a space of parameters (ρ, β, x0, y0, ...). One point of the edge images produces an increasing of n cells of the accumulator array, so all points in the same curve are mapped in an intercession cell at the parameter space. The biggest value will define the aimed curve. This method was proposed by Hough in [1] and since then many papers about this issue has been written. Firstly, Duda and Hart present an improvement of line detection using polar form. Then Kimme et al. show an approach of circles detection using gradient information. In [4] is proved through circles and parabolas detection it is possible outline a generalized method. Ellipses detection was a target of [7], [9], and [10] in different approaches. Moreover, in 1981 Ballard was pioneer in a development of generalized Hough transform and the followings [4], [15], [16] and [18] improve the generalized transform for arbitrary set of points. However, in such generalization the main objective is more related to the pattern recognition than to shape detection because no more an equation is searched.

This paper extends the study of detection of each form separately to a generic detection of conic equations, so this algorithm is able to detect all conic forms. Firstly, next section shows what form can be represented by generic second-order polynomials. It analyzes a set of parameters with theirs ranges and discrete intervals respectfully, and finally the accumulator array. In section 3, the new integrated approach is presented. Section 4 demonstrates by examples the approach efficiency on conic shapes detection and definition. Then, some conclusions are considered in last section.

2. CONIC FORMS

Conic curves are resulted from a plane and a cone intersection (figure 1) [20]. If the plane is perpendicular to the axis and passes in vertex intersection results a line. If the perpendicular plane does not pass in the vertex, intersection results a circumference. If the cutting plane is parallel to the generatrix of cone without passing in the vertex it will obtain a parabola. If the cutting plane is parallel to axis but it do not pass in vertex it will define a hyperbole. If the plane is not parallel to axis or cone generatrix and it do not pass in the vertex, the obtained curve by the intersection is an ellipse [11][19].

In order to develop a unified generic approach for conic detection, in the following lines the application of Hough transform for each second-order curve is discussed.

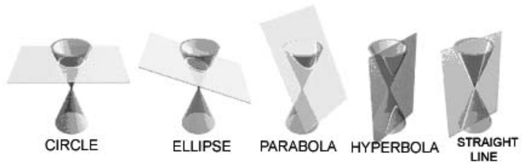

Figure 1: Conics.

2.1 Straight Lines

Although several papers have already shown straight line detections, it is important to mention that straight lines are also a conic form. Lines in polar form are represented as:

$$\rho = x\cos\theta + y\sin\theta$$

This equation is the main point in computing the accumulator matrix for straight line. It describes a line having orientation θ at distance ρ from the origin, clockwise is considered. In this work *(x,y)* are the coordinates of any form searched. The range of parameters to be used is given by:

$$-\sqrt{N_1^2 + N_2^2} \le \rho \le \sqrt{N_1^2 + N_2^2}$$

$$-\pi/2 \le \theta < \pi/2$$

Applying the Hough transform to straight lines means to have a 2-D accumulator array $M(\rho,\theta)$.

2.2 Circles

The circumference is the locus of point of an

equidistant plane from a fixed point, called center. The distance between the center and a point on the circumference is called radius.
The implicit form of circles is:

$$(x-x_0)^2+(y-y_0)^2=r^2,$$

where x_0 e y_0 are the Cartesian coordinates of circle center and r is its radius.
Lines detection copes with a 2-D accumulator array but in circles detection is necessary to manipulate a 3-D accumulator because of the needed parameters: x_0, y_0 e r. The computational efficiency in this case becomes worse than the line detection due to this increase on the quantity of parameters. The implicit form has some difficulties to define the coordinates x_0 e y_0 by an expression containing only independent variables, so it is better use the polar form.

$$x_0 = x - \rho\cos\theta$$

$$y_0 = y - \rho\sin\theta$$

where (ρ,θ) are polar coordinates in a circle.
Circles do not have sensibility for rotation. On the other hand the next curves do. Thus this accumulator array has three dimensions $M(x_0, y_0, \rho)$.
As well as line detection, curves detection has the necessity to define a range and discrete interval of all parameters.
The coordinate x_0 should be defined between 0 and N_i (maximum coordinate in the image x axis). y_0 should have a range between 0 and N_j (maximum coordinate in the image y axis). The programmer or user decides the range of the radius because of its flexibility. It means the range can have just a value or a set of different values. The most voted will be the radius of circle in the original image. For each point in an image is considered a range of radius and for each radius, a circle center is computed. Each discovered center associated with a radius is incremented by a unity in the accumulator matrix.

2.3 Parabolas
Parabola is a plane curve (figure 2). It is the locus of points in a plane, which are equidistant from a fixed point called focus and from a fixed line called diretrix. Its general equation is:

$$(y-y_0)^2=4d(x-x_0).$$

A parabola can be better parameterized through the polar form:

$$\rho=\frac{2d}{1-\cos\beta},\ \beta\neq 0$$

Where d is the distance between focus (F) and vertex (V), ρ is the distance from the focus to each parabola point, and β is the angle between ρ and the horizontal axis. So (ρ,β) are the polar coordinates of the parabola (figure 2).

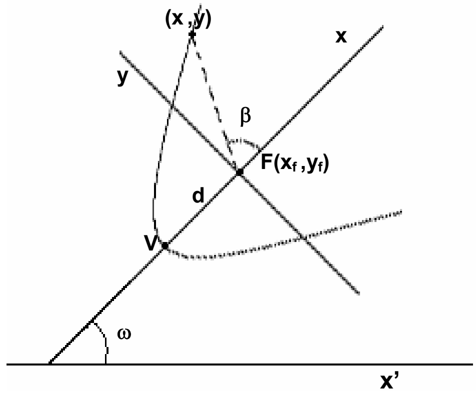

Figure 2. Rotated parabola.

In addition to a generic translation (x_f,y_f) and rotation ω, considerate equations for this kind of transformations are given by:

$$x' = x_f + \rho\cos\beta\cos\omega - \rho\sin\beta\sin\omega$$

$$y' = y_f + \rho\cos\beta\sin\omega + \rho\sin\beta\cos\omega$$

In this case is necessary to obtain a 4-D accumulator array M (x_f, y_f, d, ω).
The parameter space is discretized with focus in the image domain. The parameter presents d in a flexible range between a maximum and a minimum value, depending on the programmer or user. The rotation parameter ω varies

between 0 e 2π radians (excluding 2π).

2.4 Ellipse

An ellipse is a closed plane curve generated by a point moving in such a way that the sum of its distances from two fixed points is a constant. The Cartesian coordinates in an ellipse is expressed as:

$$\frac{x^2}{s^2}+\frac{y^2}{t^2}=1$$

when the focuses are on the x axis. A polar representation of ellipse is:

$$\rho^2=\frac{s^2t^2}{s^2\sin^2\tau+t^2\cos^2\tau}$$

where s is the biggest radius, t is the smallest radius, ρ is the distance between center and an edge point in the ellipse, and τ represents the angle between ρ and horizontal axis. (ρ,τ) are the polar coordinates. Figure 3 represents graphically the polar form where the parameter α is the rotation angle about the axes center.

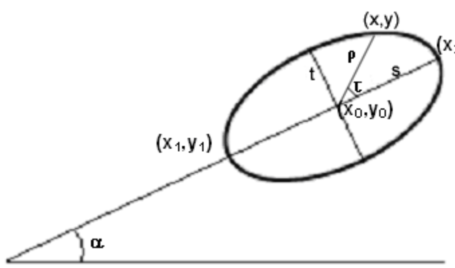

Figure 3. Rotated ellipse.

Each coordinate (x,y) on ellipse if it is rotated of α and translated through (x_0,y_0), is described by the following equation.

$$x^{'}=x_0+s\cos\tau\cos\alpha+t\sin\tau\sin\alpha$$

$$y^{'}=y_0+t\sin\tau\cos\alpha-s\cos\tau\sin\alpha$$

Detecting ellipses without any adaptations is to cope with a 5-D accumulator array $M(x_0, y_0, s, t, \alpha)$.

However, fixing any parameter value it is possible to reduce the high computation level.

In view of N_i x N_j be the image dimension, the biggest axis *2s* can be in the range:

$$2\le 2s\le\sqrt{N_i^2+N_j^2}\ ,$$

the smallest axis *2t* is in the range:

$$1\le 2t\le\sqrt{N_i^2+N_j^2}-1\,,$$

and the rotation angle $\alpha\in(0,2\pi]$.

2.5 Hyperbole

Hyperbole is a locus of points where the difference of focus is *2m,* where *m* is the biggest radius (figure 4). A hyperbole is expressed as:

$$\frac{x^2}{m^2}-\frac{y^2}{n^2}=1,$$

where the biggest radius is on the image *x* axis. A polar representation of hyperbole is:

$$\rho^2=\frac{m^2n^2}{m^2\sin^2\sigma-n^2\cos^2\sigma}$$

The half biggest axis is *m* and *n* is the half of smallest axis in hyperbole, *ρ* is the distance between centre and an edge point in the hyperbole, and *δ* represents the angle between *ρ* and *m* axis. (*ρ, δ*) are the polar coordinates. Figure 4 represents graphically the parameter *σ* that is the rotation angle of the hyperbole *m* axis.

The followings equations compute all points (*x,y*) in hyperbole by using the *σ* rotation and the center translation (x_0,y_0) if it is necessary.

$$x^{'}=x_0+m\cos\delta\cos\sigma-n\sin\delta\sin\sigma$$

$$y^{'}=y_0+n\sin\delta\cos\sigma+m\cos\delta\sin\sigma$$

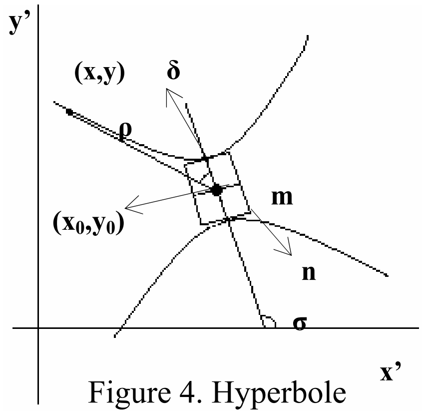

Figure 4. Hyperbole

Similar to ellipses, this curve presents an application of Hough transform, where a 5-D accumulator array M(x_0, y_0, m, n, σ) is necessary. Unless some modification on the formulation are introduced by fixing one of the parameters.

The hyperbole parameters are limited by:

$$2 \le 2m \le \sqrt{N_i^2 + N_j^2}\ ,$$

$$1 \le 2n \le \sqrt{N_i^2 + N_j^2} - 1\ ,$$

$$\sigma \in (0,2\pi].$$

3. GENERIC DETECTION METHOD

The developed integrated formulation approach is presented in this section. The bases of the formulation are: the use of polar coordinates on the description of all curve equations and parameters; the order of search for these parameters; and its respective level of approximation and range. The description of each curve in polar form was presented in the previous section. It was observed that the search sequence is fundamental on the successes of the unified formulation. Its better start the search for the polar coordinates (polar radius, polar angle) of each image pixel, given value intervals of angle, through the Hough methodology, which involves the election of the most parameter voted, the polar radius is chosen. Other simple but relevant consideration is that more important than the number of parameters is the definition of its range in limits and also in number of bins. The accuracy in a conic detection depends on the chosen size of cell, it means, the smaller size of cell is, the more accuracy and slower it is.

Considering the common aspects of all conics, it is possible to establish a unified Hough transform approach. The parameters of a curve are represented as a set P= (p_1, p_2, p_3,…). Then, each element of P should have a value in the adequate range as L= (l_1, l_2, l_3,...) and U=(u_1, u_2, u_3,…), where L is the lower limit and U is the upper limit according to the image domain.

The accumulator matrix has a dimension given by:

$$D = \begin{cases} length(P) + 1 & \textit{if the conic has rotation} \\ length(P) & \textit{otherwise} \end{cases}$$

For each element in P, its polar equation is computed. If there is center or focus, these are calculated as following:

$$x_0 = x - \rho\cos\beta\cos\omega + \rho\sin\beta\sin\omega\,,$$

$$y_0 = y - \rho\cos\beta\sin\omega - \rho\sin\beta\cos\omega\,,$$

where ρ is the polar form of curve point , (x,y) is an image point, β is the inclination between the curve center or focus and a curve edge point, and ω is the curve rotation angle. Afterwards, the cell that corresponds to the wanted parameters is incremented by 1, but the accumulation array has to respect the limits L and U. This algorithm allows detecting conics shapes also in very noisy images. It is also possible to detect approximated conic forms like handwriting drawing. When all parameters of curve are detected its easier compute their geometric features like inclinations, curvatures, focuses, centers and others, which can be useful in pattern recognition or shape analysis [17].

3.1. Final point identification

Only circles and ellipses are closed curves. Considering the other conic forms, it is not possible to define the final points through only the space of parameters. So it is necessary to create a limit point identification mechanism. In this work is proposed as a mechanism a simple comparison between the pixels in the original image and in the detected image to elect the final point. However, if such pixels do not match, the Euclidian distance with a tolerance will be used. In case of lacking points along the curve this distance is also considered. Moreover, if the matching does not occur, the final point will be the last pixel matched successfully with the original image. When a conic form is closed, both initial and final points receive as default the value zero. Furthermore, arcs of circles or ellipses can be determined by the same mechanism as shown

in next section.
Other resource that is possible with final point identification is detecting more than one conic at the same image. The basic idea about that is detecting a conic, then this form is erased and then the next conic could be detected at the same image [12].

4. EXPERIMENTAL RESULTS

In this section, the efficiency of the generic detection of all conic forms is demonstrated. It also illustrated the algorithm possibility on detection of coarse curves. An AMD Athlon 2000xp 1.6 GHz, 256Mb of memory RAM computer was the machine used in the experiments. All images are gray scale. The average of images size is between 100x100 and 300x300 pixels. All the results are presented in pixels except the angle which are in degrees.
In the first test, detection of lines is considered. Figure 5 shows the original image with four straight lines that constitutes a rectangle. Figure 6 presents the four detected straight lines where the blue points are the initial and final detected points (the vertex of the form). Table 1 shows the parameters of the detected straight lines.

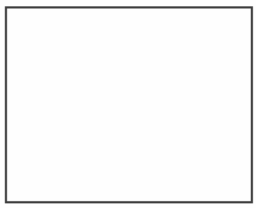

Figure 5. Rectangle

Figure 6. Detected rectangle

Table 1. Extracted features

θ	ρ	Final Point	Initial point
0	26	(26,40)	(26,154)
0	116	(116,40)	(116,154)
90	40	(26,40)	(116,40)
90	154	(26,154)	(116,154)

The next test presents the possibility of detection conic forms in failed or lacking images (figure 9). Although there are fails on the original image, the figure 10 shows a complete detected circle. This test detected a radius circle with 69 pixels and center (95,102).

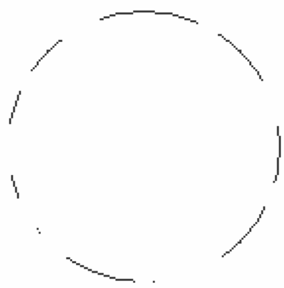

Figure 9. Original failed image.

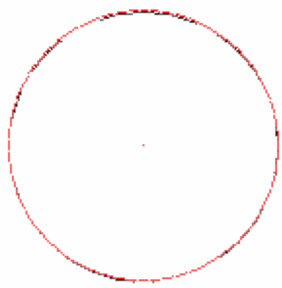

Figure 10. Detected circle.

Figures 11 and 12 show the detection of circle arcs, and their parameters are: center (95,101), radius= 68 pixels, final point (62,18), and initial point (22,69).

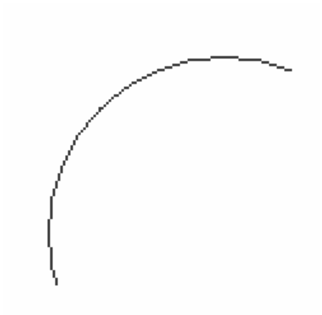

Figure 11. Circle arc.

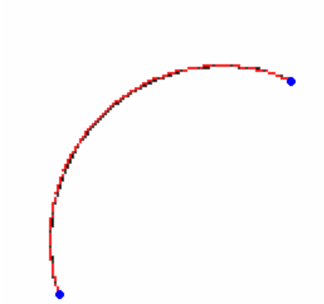

Figure 12. Circle arc detected.

Figures 13 and 14 show the fifth test, parabola detection, and their parameters are: focus coordinates: (67,62), d= 7 pixels, ω= 223°, final point: (62,18) and the initial point: (22,69).

Figure 13. Original image of rotated parabola.

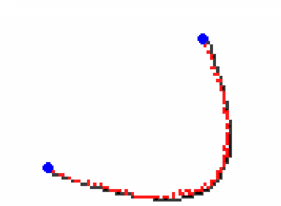

Figure 14. Detected parabola

The test with ellipses also happened successfully. The tests in figures 15 and 16 show one of the tests done, and the detected parameters are: center (64,62), s=86, t=48 and α=138°.

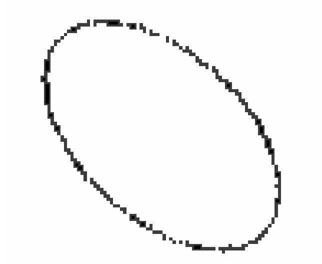

Figure 15. Original rotated ellipse

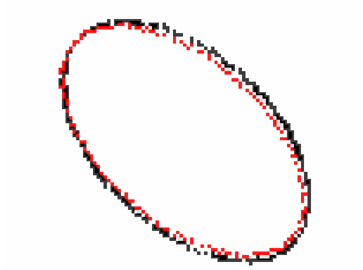

Figure 16. Detected ellipse

The tests include ellipses with fails (figure 17). The idea is found the complete form in the image (figure 18). The founded parameters are: center (70,94), s=124, t=60 and α=0°

Figure 17. Original failed

Figure 18. Complete detected ellipse

Forms drawn by hand can be detected at the same way of the others, which shows the possibility on detection of approximate shapes. Figures 19 and 20 show the process. The extracted parameters are: center (67,55), s=46 pixels, t=26 pixels, and α= 50°.

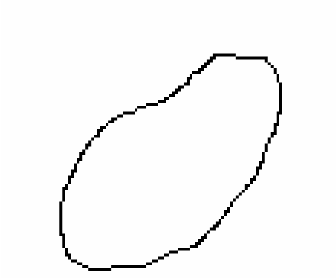

Figure 19. Handwritten form.

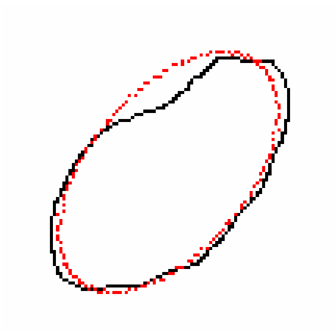

Figure 20. Handwritten detected form.

The last test (figure 21,22) shows more than one form conic detection to the same image. At the following set of images, it is possible see each step of detection. First, circle arc with the smallest radius is detected, after that, the circle arcs with the biggest radius, the superior line, and finally the inferior line is detected. Table 3 presents all features founded. The set of noise pixels in the end of process it is not significant, but if it is necessary or be considered relevant the detection process can be continued.

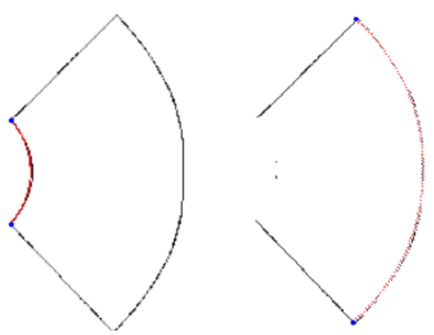

Figure 21. 1°. and 2°. steps to detect a complex form.

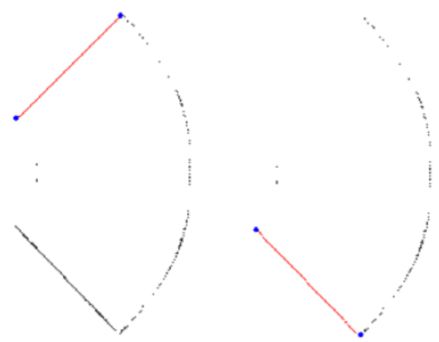

Figure 22. 3°. and 4°. steps to detect a complex form.

Table 3: Parameters founded in arcs and lines

	x_0	y_0	ρ	Initial point	Final point
Circle arcs	109	32	51	(74,70)	(144,69)
	109	46	140	(5,140)	(216,136)
	θ	ρ	-	Initial point	Final point
Lines	45	102	-	(5,141)	(73,71)
	135	53	-	(145,70)	(215,140)

5.CONCLUSIONS

In this paper, an approach for the generic detection of conic forms based on the Hough transform is presented. Several tests illustrated the efficiency on detection of any conic form or any combination of conic shapes. The conic form can be corrected detected in any quantity on the image; with any rotation; for closed or opened curves; with fails or noise; for circle or ellipses arcs; by handwriting, or in a coarse manner when the form is drawn. Finally when several different kind of conics appear in the same image [12]. The features that define the curve equation can also be computed.

Although the generic algorithm works, in some cases it presents a high computation time due to the large accumulator arrays, so in order to go down this computation it is recommended do an association with a optimization technique like in [6] and [19], which proposed to implement Hough transform using an adaptative accumulator array or [8] and [14], whereby a set of n pixels is randomly selected from the edge image.

The union of all conic detection on the same approach it is a new issue. It also detects any kind of conic when these are at the same image.

REFERENCES

[1] Hough, P., “Method and means for recognizing complex patterns”, U.S Patent 3 069 654, 1962.

[2] Duda, R. and. P. Hart, “Use of the Hough transformation to detect lines and curves in the pictures”, Communication Association Computing Machine, Vol. 15, pp. 11-15, Jan, 1972.

[3] Kimme, C., D. Ballard, and J. Sklansky, “Circles by an Array of Accumulators”, Communications of the ACM, Vol.18, pp 120-122, Feb, 1975.

[4] Sklansky, J., “On the Hough Technique for Curve Detection”. IEEE Transaction Computer, No. 10, 1978.

[5] Ballard D., “Generalizing the Hough Transform to Detect Arbitrary Shapes”. Pattern Recognition, Vol. 13, pp.111-122, 1981.

[6] Illingworth, J. and J. Kittler, “The Adaptive Hough Transform”, IEEE Transaction on Pattern Analysis and Machine intelligence, Vol. PAMI-9, No. 5, 1987.

[7] Lei, Y. and K. Wong, “Ellipse Detection Based on Symmetry”. Pattern Recognition, Vol. 20, pp. 41-47, 1999.

[8] Xu, L. and E. Oja, “Randomized Hough transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities”, CVGIP: Image Understanding, 1993.

[9] Aguado, A., M. Montiel and M. Nixon. “On Using Directional Information for Parameter Space Decomposition in Ellipse Detection”. Pattern Recognition, Vol. 29, No. 3, pp. 369 381, 1996.

[10] Xie, Y., “A New Efficient Ellipse Detection Method”, IEEE Transaction Computer, 2002.

[11] Kaplan, W. and D. Lewis. “Cálculo e Álgebra Linear 2”. Livros Técnicos e Científicos. S.A., 1973.

[12] Macedo, M., “Uso da Transformada de Hough na Vetorização de Moldes e Outras Aplicações”. M. Sc. Dissertation, Computer Science Department - Universidade Federal Fluminense, 2005.

[13] Olson, C., “A General Method for Geometric Feature Matching and Model Extraction”. International Journal of Computer Vision. Vol.45, pp.39–54, 2001.

[14] Chen, T. and K. Chung. “An Efficient Randomized Algorithm for Detecting Circles”. Computer Vision and Image Understanding, Vol. 83, pp. 172-191, 2001.

[15] Aguado, A., M. Montiel, and M. Nixon. “Invariant Characterization of the Hough Transform for Pose Estimation of Arbitrary Shapes”. Pattern Recognition, Vol. 35, pp. 1083-1097, 2002.

[16] Chau, C. and W. Siu. “Generalised Hough Transform using Regions with Homogeneous Color”. International Journal of Computer Vision, Vol. 59, pp. 183-199, 2004.

[17] Costa, L. and R., Cesar Junior. “Shape Analysis and Classification:- Theory and Practice”. ISBN: 0849334934, CRC Press,

2000.

[18] Aguado, A., M. Montiel , and M. Nixon. “Bias Error Analysis of the Generalized Hough Transform. Journal of Mathematical Imaging and Vision”, No. 12, pp. 25-42, 2000.

[19] Ecabert, O. and J. Thiran. “Adaptive Hough Transform for the Detection of Natural Shapes under Weak Affine Transformations”. Pattern Recognition, Vol. 25, pp. 1411-1419, 2004.

[20] Azevedo, E. and A. Conci. “Computação Gráfica: teoria e prática”, Campus; 2003.

ABOUT THE AUTHORS

1. Maysa Macedo, Master in Computing by Fluminense Federal University, is an Assistant Professor in the Department of Computing Science, Augusto Motta University Centre. Her interests are in Image Processing. She can be reached by e-mail: maysam@unisuam.edu.br .

2. Aura Conci, Doctor, is a Titular Professor in the Department of Computing Science, Fluminense Federal University. Her research interests are in Computer Graphic and Image Processing. She can be reached by e-mail: aconci@ic.uff.br .