01 Aug 2009

Subpixel Detection of Circular Objects using Geometric Property

Wen-Yen Wu

Vincent Wen-Bin Yu
Missouri University of Science and Technology, yuwen@mst.edu

Recommended Citation

Wu, W., & Yu, V. W. (2009). Subpixel Detection of Circular Objects using Geometric Property. *International Journal of Mathematical and Computational Sciences, 3*(8), pp. 572-576. World Academy of Science, Engineering and Technology.

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:3, No:8, 2009

Subpixel Detection of Circular Objects Using Geometric Property

Wen-Yen Wu, and Wen-Bin Yu

***Abstract*—**In this paper, we propose a method for detecting circular shapes with subpixel accuracy. First, the geometric properties of circles have been used to find the diameters as well as the circumference pixels. The center and radius are then estimated by the circumference pixels. Both synthetic and real images have been tested by the proposed method. The experimental results show that the new method is efficient.

***Keywords*—**Subpixel, least squares estimation, circle detection, Hough transformation.

I. INTRODUCTION

DETECTING circular objects is an important task in various industrial applications. Most algorithms for detecting and locating circular objects are based on Hough transformation (HT) or its variants [1, 3, 6, 7-19, 22-23]. In general, the HT methods employ a 3D accumulator array by mapping the x-y coordinates to the corresponding (x_c, y_c, r) parameter space. A peak finding procedure is then applied to detect circles.

Some non-HT algorithms have been proposed to solve the problems caused by HT-based algorithms. Wojcik [21] used graph to represent objects and a template matching technique is used to recognize the circular objects. Chen and Lin [4] used the orthogonal circular detector to estimate the parameters of circles. However, only isolated circles were tested in their study. Chen and Lee [5] used the geometric property of a circle to find some pixels on the circle and then fitted these pixels to get the center and radius of the detected circular object.

In this paper, we propose an efficient technique to identify the diameter as well as the pixels on the circumference of a circle by using the basic geometric properties of a circle. The result is of subpixel accuracy. The proposed method is presented in section II. The experimental results of the proposed method to various types of circular objects are illustrated in section III. Discussion and conclusion can be found in the final section.

This paper is partially supported by National Science Council, ROC under grant no. NSC 97-2221-E-214-040 and NSC 97-2918-I-214-002.

W. Y. Wu is with the Department of Industrial Engineering and Management, I-Shou University, Taiwan (phone: 886-7-6577711 ext. 5512; fax: 886-7-6578536; e-mail: wywu@isu.edu.tw).

W. B. Yu is with the Department of Department of Business and Information Technology, Missouri University of Science and Technology (Formerly UMR), Rolla, MO 65401, USA (e-mail: yuwen@mst.edu).

II. THE PROPOSED METHOD

Some basic geometric properties are first discussed. Based on the properties, we develop a simple method for circular object detection.

Theorem 1. Let $\angle BAC=\theta$ (see Fig. 1), then the cosine value of θ can be defined as follow:

$$\cos\theta = \frac{u_1 v_1 + u_2 v_2}{\sqrt{u_1^2 + u_2^2}\sqrt{v_1^2 + v_2^2}}, \quad (1)$$

where $(u_1, u_2) = (x_2-x_1, y_2-y_1)$ and $(v_1, v_2) = (x_3-x_1, y_3-y_1)$

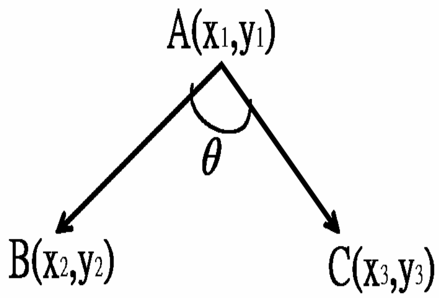

Fig. 1 The computation of cosine: points A, B, and C

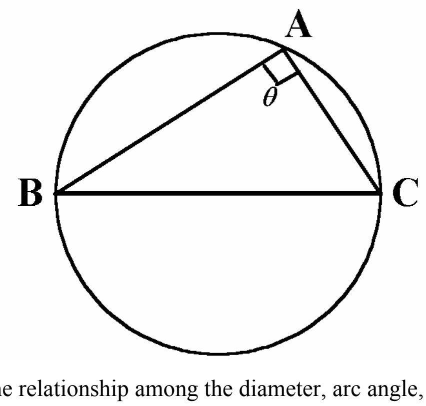

Fig. 2 The relationship among the diameter, arc angle, and chord

Theorem 2. As seen in Fig. 2, the segment $\overline{BC}$ is a diameter of the circle, if and only if, $\angle BAC$ is equal to $\pi/2$.

Theorem 3. Let $\overline{BC}$ be a diameter of a circle (see Fig. 2). A is a point on the circle, if and only if, $\angle BAC$ is equal to $\pi/2$.

Theorem 4. As shown in Fig. 3, $\overline{BC}$ is a diameter of a circle and *h* is the height from A to $\overline{BC}$. Furthermore, if *a*, *b*, and *c* are the chord length of $\overline{BC}$, $\overline{AC}$, and $\overline{AB}$ respectively, then *ah*=*bc*.

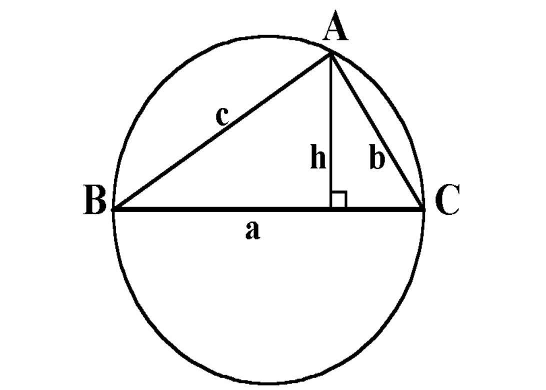

Fig. 3 Illustration of chord lengths, where $\overline{BC}$ is a horizontal diameter

For a horizontal diameter, i.e. $y_2 = y_3$, it implies that $h = y_2-y_1=y_3-y_1$ and $a=x_3-x_2$ (assume that $y_3>y_1$ and $x_3>x_2$), Eq. (1) can be formulated as:

$$\begin{aligned}\cos\theta &= \frac{(x_2-x_1)(x_3-x_1)+(y_2-y_1)(y_3-y_1)}{bc} \\ &= \frac{(x_2-x_1)(x_3-x_1)+(y_2-y_1)^2}{ah} \qquad (2) \\ &= \frac{(x_2-x_1)(x_3-x_1)+(y_2-y_1)^2}{(x_3-x_2)(y_2-y_1)}\end{aligned}$$

For a vertical diameter, that is, $x_2=x_3$, similarly, we have

$$\cos\theta = \frac{(x_2-x_1)^2+(y_2-y_1)(y_3-y_1)}{(x_2-x_1)(y_3-y_2)}. \qquad (3)$$

A. Finding the Diameter

To identify pixels on a circle, it is important to find the diameter correctly. In Fig. 4, for each object we first find two pixels $A_1(x_1, y_1)$ and $A_2(x_4, y_4)$ vertically, and then the two other pixels $B(x_2, y_2)$ and $C(x_3, y_3)$ can be found by the corresponding mid-horizontal chord, i.e. the horizontal line $y=(y_1+y_4)/2$. Therefore, $y_2=y_3= (y_1+y_4)/2$. Because a circle is symmetric, the line segment connected by B and C is the candidate diameter of a circle. Suppose that A_1 or A_2 is one pixel on a circle. Next, we can test whether $\overline{BC}$ is a diameter by Theorem 2. That is, $\overline{BC}$ is a diameter, if and only if $\angle B A_1C$ or $\angle B A_2C$ is equal to $\pi/2$.

Therefore, if $\cos\theta_1$ or $\cos\theta_2$ equals to 0, where $\theta_1 = \angle BA_1C$ and $\theta_2 = \angle BA_2C$, then $\overline{BC}$ will be considered as a diameter of a circle. However, the condition of $\cos\theta_i = 0$ may be never hold due to the digitization error. Thus, the condition $\cos\theta_i = 0$ is replaced by $|\cos\theta_i| < \varepsilon$.

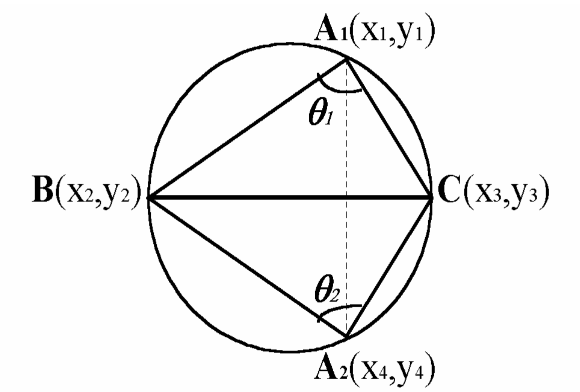

Fig. 4 The arc angles being used for finding the horizontal diameter of a circle

That is, if one of the conditions $|\cos\theta_1| < \varepsilon$ and $|\cos\theta_2| < \varepsilon$ is satisfied, then $\overline{BC}$ is a horizontal diameter. Otherwise, we can scan from line $y=y_1$ to line $y=y_4$ vertically to find whether a horizontal diameter exists. If horizontal diameter cannot be found, we may change the scanning direction to find the vertical diameter similarly.

If both of horizontal and vertical diameters are not found, then we will try to find diagonal diameters in the same way. Overall, four types of diameters, as shown in Fig. 5, have to be checked in the diameter searching process.

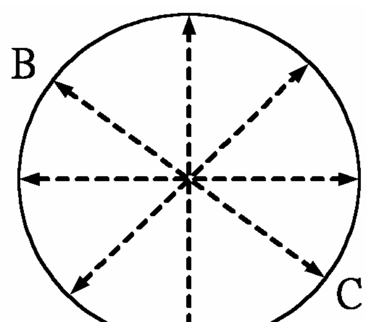

Fig. 5 The four types of diameters of a circle: horizontal, vertical, and two types of diagonal diameters

B. Detecting Pixels on a Circle

Once a diameter has been identified by the above searching strategy, one can subsequently look for all the candidate circular pixels. We first search for these pixels in pairs in the perpendicular direction corresponding to the circular diameter $\overline{BC}$ from B to C. For each pair of pixels, we can determine whether they are pixels on a circular object by Theorem 3 (see Fig. 6). Again, the condition of $\cos\theta_i = 0$ must be relaxed by setting an allowance as described in the last section.

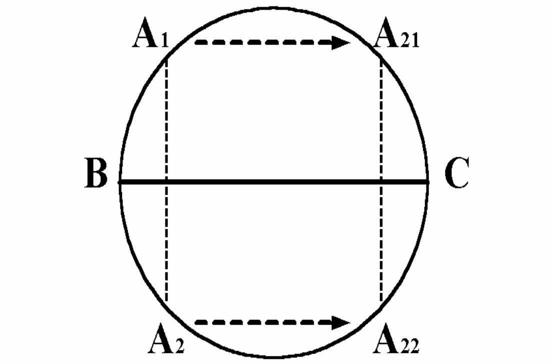

Fig. 6 Looking for circular pixels in pairs from (A_1, A_2) to (A_{21}, A_{22}) with diameter $\overline{BC}$

However, some objects may result in false diameter detection as seen in Fig. 7. In order to avoid false detections, it is necessary to develop a criterion to judge whether the detected object is a circular object or not. Since the number of circular pixels is proportional to the diameter, a threshold n_t depending on the length of the detected diameter is specified. If the number of the detected circular pixels is larger than the threshold, the object is considered as a circle; otherwise, the object is not a circle. For example, only a few of pixels will be identified as the circular pixels in Fig. 7. Therefore, it is not a circle. In addition, for a broken or noisy circular object, the threshold should be decreased, because the number of circular pixels is less than that of a perfect one.

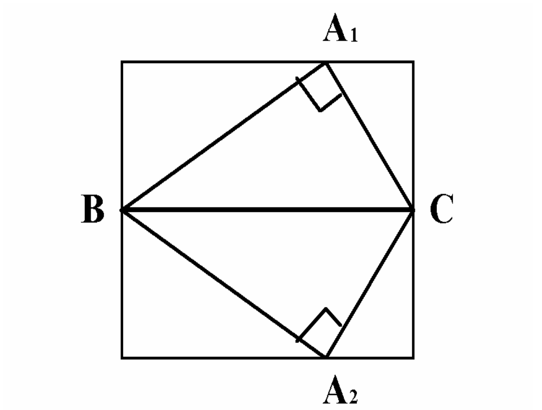

Fig. 7 An example of objects results in false diameter detection

C. Estimating Parameters of the Circular Objects

In this section, the procedure of estimating the parameters of the circle is demonstrated. Once the circular pixels were identified and the number was larger than a preset threshold, it is necessary to locate the circular object. The Thomas and Chan's approach [20] is used to estimate the center (x_c, y_c) and radius *r*. Let $(\hat{x}_c, \hat{y}_c, \hat{r})$ be the estimate of (x_c, y_c, r). For a set of *n* circular pixels, (x_i, y_i) for i=1, 2, …, n, the formula of estimating the center and radius is given as:

$$\hat{x}_c = \frac{c_1 b_2 - c_2 b_1}{a_1 b_2 - a_2 b_1} \tag{4}$$

$$\hat{y}_c = \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1} \tag{5}$$

$$\hat{r} = \left(\frac{1}{n}\left(\Sigma x^2 - 2\hat{x}_c \Sigma x + n\hat{x}_c^2 + \Sigma y^2 - 2\hat{y}_c \Sigma y + n\hat{y}_c^2\right)\right)^{1/2} \tag{6}$$

where

$$a_1 = 2(\Sigma_x^2 - n\Sigma x^2) \tag{7}$$

$$b_1 = 2(\Sigma x \, \Sigma y - n\Sigma xy) \tag{8}$$

$$a_2 = 2(\Sigma x \, \Sigma y - n\Sigma xy) = b_1 \tag{9}$$

$$b_2 = 2(\Sigma_y^2 - n\Sigma y^2) \tag{10}$$

$$c_1 = \Sigma_{x^2} \, \Sigma x - n\Sigma_{x^3} + \Sigma x \, \Sigma_{y^2} - n\Sigma_{xy^2} \tag{11}$$

$$c_2 = \Sigma_{x^2} \, \Sigma y - n\Sigma_{y^3} + \Sigma y \, \Sigma_{x^2} - n\Sigma_{x^2 y} \tag{12}$$

Considering the efficiency of computation, we can reduce the number of circular pixels in the least squares estimation. When locating circular pixels in pairs along the perpendicular direction of the diameter from B to C, instead of moving pixel by pixel, it can be done by jumping α-pixels in each step. The setting of α depends on the length of the diameter. The longer the length of the diameter is, the larger the α can be. Both of the circular pixels identification and the parameter estimation time will be about $1/\alpha$ of the original processing time. Besides, the threshold n_t must be reduced to n_t/α at the same time.

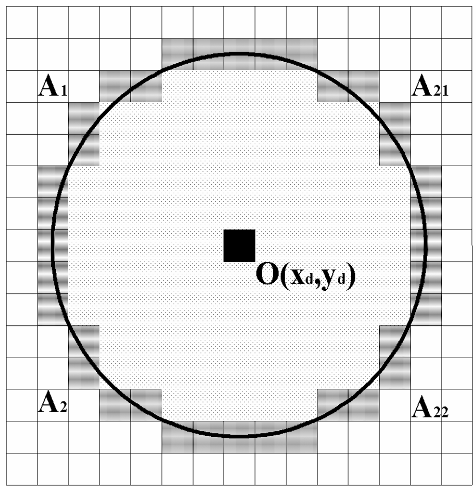

Fig. 8 Circle: continuous vs. digital

The modified coordinates are closer to the true coordinates of a continuous circle than that of the original coordinates. In addition, they are of subpixel accuracy. They are adjusted by the following rules.

$$x_i' = \begin{cases} x_i + 0.5 & \text{if } x_i < x_d \\ x_i & \text{if } x_i = x_d \\ x_i - 0.5 & \text{if } x_i > x_d \end{cases} \tag{15}$$

$$y_i' = \begin{cases} y_i + 0.5 & \text{if } y_i < y_d \\ y_i & \text{if } y_i = y_d \\ y_i - 0.5 & \text{if } y_i > y_d \end{cases} \tag{16}$$

D. Circle Detection Algorithm

In summary, the proposed circular object detection method can be stated explicitly as the following steps.

Step 1. Acquire the image of the object and convert it to binary image.

Step 2. Repeat Steps 3 to 6 until all objects have been processed.

Step 3. Find a horizontal, vertical, or diagonal diameter using the modified version of Theorem 2. If there is no diameter found, go to Step 2.

Step 4. Identify the circular pixels of the detected diameter using the modified version of Theorem 3.

Step 5. If the number of circular pixels is larger than a threshold n_t/α, a circle has been detected. Otherwise, go to Step 2.

Step 6. Modified the x-y coordinates of the circular pixels by Eq. (13) and Eq. (14). Next, estimate the parameters of the circle by Eq. (4) to Eq. (6).

III. Experimental Results

For a perfect circular object, by experience, we can use 160% of the length of the detected diameter as threshold. That is, the value of n_t is set to $1.6\times\left|\overline{BC}\right|$, where $\left|\overline{BC}\right|$ is the length of the diameter $\left|\overline{BC}\right|$. On the other hand, if the object is semi-circular, then the threshold should be reduced to 80% of the diameter; that is, $n_t = 0.8\times\left|\overline{BC}\right|$. In general, it is sufficient to use about 40 pixels in the parameter estimation. In order to use about 40 pixels in the parameter estimation for a perfect circle, the jumping interval α is set to the smallest integer that is larger than or equal to $\left|\overline{BC}\right|/20$.

In order to verify the proposed method, it has been tested on 400 synthetic images. Each of the synthetic images includes five perfect objects and four half-broken circular objects with random radius varying from 3 to 50 pixels. One of the images including five types of circular objects is presented in Fig. 9. Further, in order to access the ability of the proposed method under noisy condition, two additional experiments have been conducted. We repeat the first experiment by adding noise the testing images with signal to noise ratio (S/N) 20 and 50, respectively.

Since the centers and radii of the testing circles are known. The errors can be accessed. Let $(\hat{x}_c, \hat{y}_c, \hat{r})$ be the estimate of the real parameter (x_c, y_c, r). x_e, y_e, and r_e (the errors corresponding to x-coordinate, y-coordinate, and radius, respectively) can be defined as

$$x_e = \hat{x}_c - x_c \tag{17}$$

$$y_e = \hat{y}_c - y_c \tag{18}$$

$$r_e = \hat{r} - r \tag{19}$$

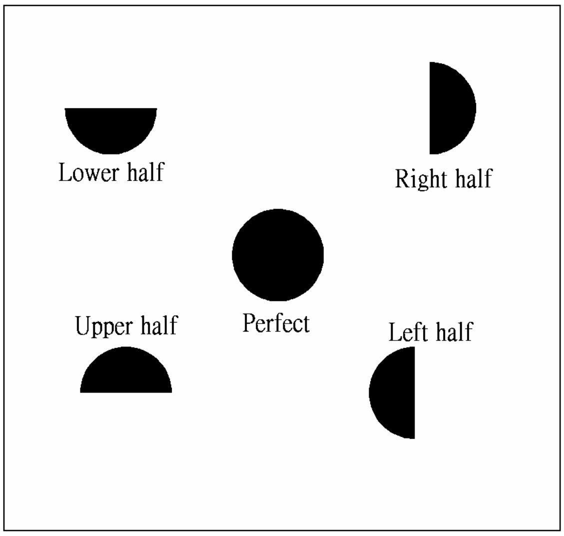

Fig. 9 A synthetic image containing five types of circular objects

TABLE I
Average Estimation Errors for Synthetic Images

Circles	Fig. 9	S/N=20	S/N=50
	(x_e, y_e, r_e)	(x_e, y_e, r_e)	(x_e, y_e, r_e)
Perfect	(0.01, 0.01, 0.05)	(0.05, 0.06, 0.10)	(0.56, 0.67, 0.84)
Upper half	(0.01, 0.10, 0.09)	(0.12, 0.21, 0.25)	(0.98, 1.01, 1.15)
Lower half	(0.01, 0.10, 0.09)	(0.13, 0.18, 0.20)	(0.83, 0.95, 1.01)
Left half	(0.10, 0.01, 0.08)	(0.19, 0.15, 0.16)	(1.26, 0.98, 1.18)
Right half	(0.10, 0.01, 0.08)	(0.15, 0.23, 0.24)	(1.05, 1.16, 1.35)

The experimental results are shown in Table I. All of the circular objects were successfully detected. For noise-free images, the average errors of x_e, y_e, and r_e are very small (within 0.10 pixels) for all the five types of circular objects. Especially for perfect circular objects, the average x_e and y_e are even smaller i.e.- about 0.01 pixels. In addition, the average accuracy level of radius approximates to 0.05 pixels for perfect circular object. For half-broken circular objects under the four directions, the accuracy level can reach 0.10 pixels. Further, the new method can detect very small circular objects. For a synthetic image, it can even detect circular objects with radius of 3 pixels.

For the images with S/N=20, the average errors are all smaller than 0.10 pixels for the perfect circular objects. The average errors of x-coordinate and y-coordinate are less than 0.23 for the semi-circles. In addition, for the images with S/N=50, the average errors of x-coordinate, y-coordinate, and radius are less than 0.90 pixels for the perfect circles. The errors are less than 1.50 pixels for the broken circular objects. The results indicate that the proposed method is robust for noise.

Another testing image that contains various types of objects is shown in Fig. 10. They include some near half-broken, occluded, and perfect circular objects as well as other shaped objects. The detected circles (dash-lined) are superimposed into the original image and the detected centers have been marked as small crosses. As it can be seen, the proposed method can successfully locate all the circular objects.

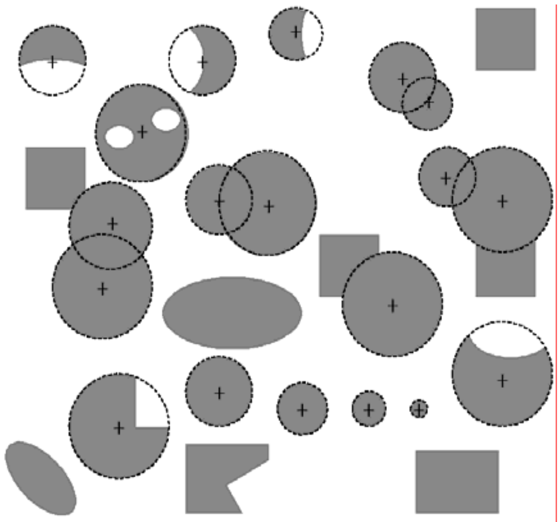

Fig. 10 Results of detecting circular objects, where the detected circles are presented by dash-line and the centers are marked as crosses

IV. CONCLUSION

A new technique for circular object detection and location is proposed. Instead of transforming pixels into its parameter space as suggested by the HT approaches, we make use of the geometric properties of a circle to find the diameter and circumference pixels and estimate the parameters of the circles. In comparison with the HT approaches, the proposed method has the advantages of rather small storage required, high speed, and high accuracy. Further, the new method does not surfer from the problems of clustering and peak finding involved in HT-based methods. The experimental results also indicated that incomplete, occluded, as well as tiny circular objects can also be successfully detected and located by the proposed method. Further, the proposed method is robust to noisy environment. Overall, the proposed is effective and efficient in detecting circular objects.

REFERENCES

[1] D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition, 13, 1981, 111-122.

[2] M. Berman, Large sample bias in least squares estimators of a circular arc center and its radius, Computer Vision, Graphics, and Image Processing, 45, 1989, 126-128.

[3] R. Chan and W. C. Siu, New parallel Hough transformation for circles, IEE Proceedings-E, 138, 1991, 335-344.

[4] F. L. Chen and S. W. Lin, Subpixel estimation of circle parameters using orthogonal circular detector, Computer Vision and Image Understanding, 78, 2000, 206-221.

[5] L. H. Chen and K. L. Lee, A new method for circular object detection and location, Pattern Recognition Letters, 11, 1990, 691-697.

[6] R. S. Conker, A dual plane variation of the Hough transform for detecting non-concentric circles of different radii, Computer Vision, Graphics, and Image Processing, 43, 1988, 115-132.

[7] E. R. Davies, A high speed algorithm for circular object location, Pattern Recognition Letters. 6, 1987, 323-333.

[8] E. R. Davies, A modified Hough scheme for general circle location, Pattern Recognition Letters. 7, 1988, 37-43.

[9] E. R. Davies, A hybrid sequential-parallel approach to accurate circle center location, Pattern Recognition Letters. 7, 1988, 279-290.

[10] R. O. Duda and P. E. Hart, Use of Hough transformation to detect lines and curves in pictures, Communications of the ACM. 15, 1972, 11-15.

[11] J. Y. Goulermas and P. Liatsis, Genetically fine-tuning the Hough transform feature space, for the detection of circular objects, Image and Vision Computing, 16, 1998, 615-625.

[12] N. Guil and E. L. Zapata, Lower order circle and ellipse Hough transform, Pattern Recognition, 30, 1997, 1929-1744.

[13] D. Ioannou, W. Huda, and A. F. Laine, Circle recognition through a 2D Hough transformation and radius histogramming, Image and Vision Computing, 17, 1999, 15-26.

[14] A. Kavianpour and N. Bagherzadeh, Finding circular shapes in an image on a pyramid architecture, Pattern Recognition Letters, 13, 1992, 843-848.

[15] P. Kierkegaard, A method for detection of circular arcs based on the Hough transform, Machine and Vision Applications, 5, 1992, 249-263.

[16] G. Kimme, D. Ballard, and J. Sklansky, Finding circles by an array of accumulators, Communications of the ACM. 18, 1975, 120-122.