Tema
Tendências em Matemática Aplicada e Computacional, **20**, N. 2 (2019), 331-342

www.scielo.br/tema
doi: 10.5540/tema.2019.020.02.0331

On the Improvement of Multiple Circles Detection from Images using Hough Transform

W.O. BARBOSA and A.W. VIEIRA*

Received on April 17, 2018 / Accepted on February 22, 2019

ABSTRACT. The automatic detection of lines and curves from color images is a very important task in many applications, such as object recognition and scene reconstruction. Although there are closed formulation for curve fitting to a set of points, if the point set describes more than one instance of the object, as two circles for example, there is no closed formulation for obtaining the individual set of parameters without a priori information of which points belong to each object. However, it is usual the presence of multiple instances of objects such as lines and circles on an image. The well known Hough transform is an efficient tool for recovering multiple objects from images using a voting process where the usual presence of false positives is an issue. In our work, we present an improvement on the voting process to detect multiple circles using *Hough transform* in order to avoid false positives. Our experiments show that our voting process leads to a more robust detection, reducing the number of false positive and providing a more accurate detection even with large number of circles.

Keywords: computer vision, hough transform, circle detection.

1 INTRODUCTION

The object recognition from images is a fundamental step of most decision-making systems, but recognition requires a prior step of detecting known patterns from objects, such as lines, circles, ellipses, and so on. Patterns in images are usually found in the form of edges, with discontinuities and with noise inclusion, so that the techniques for detection must be robust to obtain results in imprecision and absence of information. The edge points of objects in the image can be extracted using a *high-pass* frequency filter. Once a set of edge points associated with a given curve is detected, the most common techniques use least squares to make the adjustment and obtain the curve parameters that best fit the points.

In the case of circle detection, given the points set $P = \{(x_i, y_i), i = 1, \dots, n\}$, the implicit form $(x_i - a)^2 + (y_i - b)^2 - r^2 = 0$ leads to a closed formula to find the center (a, b) and radius r of the

*Corresponding author: Antonio Wilson Vieira – E-mail: awilsondcc@gmail.com – `https://orcid.org/0000-0001-8079-1671`
Departamento de Ciências Exatas, CCET, Universidade Estadual de Montes Claros, Campus Professor Darcy Ribeiro, 39401-089, Montes Claros, MG, Brasil. E-mails: wesleyafram@gmail.com, awilsondcc@gmail.com

best fit circle that adjusts to the set of points. The same strategy applies to straight lines, ellipses, and conics in general, as long as the point set describes a single instance of the object. However, in most cases, the edge points present in the image describe a variety of distinct instance of curves so that, before using a closed formula, it is necessary to partition the set of points into individual subsets for each object. But this task is a greater challenge than the adjustment of a simple curve, because partitioning implies the identification of individual instances of unknown objects a priori.

One technique that proved to be very efficient in detecting curves, especially straight line, in images is the Hough transform (HT) [6]. This transformation constructs a representation of the image points in a new space called the parameter space. A voting strategy is then adopted so that each edge point in the image determines votes, in the parameter space, for the curves that pass through the point. In this way, points that belong to the same curve determine, with their votes, a local maximum in the parameter space. Finally, the search for the local maximums in this space provides the parameter lists of the objects present in the image. Theoretically, the method can be extended to circles, ellipses, conics in general and curves with any number of parameters. However, the parameter space grows exponentially with the number of parameters, making the technique impractical in terms of storage and time complexity when the curve requires more than 4 or 5 parameters. In the case of line and circle detection the method can be implemented efficiently and is widely used in the literature.

A drawback in detecting circles using HT is the compromise between avoiding False Positive (FP) without creating False Negative (FN). A large number of FP is usually associated with the fact that a circle arc intersects several points in the image, even if such a circle is not an object present in the image. On the other hand, attempting to filter such FP by requiring a minimal amount of votes, eventually creates False Negative (FN) by deleting smaller radius circles that are real and present in the image.

In our work, we sought to detect multiple circles in images using HT by using a strategy that improve the compromise between FP and FN with consequent improvement in the accuracy of the method. Different from the standard approach, where each point in the image space corresponds to a same weight vote in the parameter space, we use a strategy where the weight is inversely proportional to the radius in the parameter space. The advantage of this approach is that the same threshold that eliminates FP associated with large-radius circles, also avoids the FN associated with the disappearance of small radius circles.

2 RELATED WORKS

The Hough transform (HT) [7] was a landmark for the detection of curves in images, especially for straight lines. Using this strategy, several other approaches focused on detecting circles, ellipses and other curves. Many modifications are proposed to improve the performance of circle detection in terms of processing time. While Davies [3] proposed an implementation of the Hough transform in which the circles are parametrized by (x,y,r), where (x,y) refer to the center position and r to the radius, using the usual voting strategy, Matsumoto and Tsuji [10] decompose the parameter space to obtain a more efficient implementation in two stages. Another common approach is the use of edge orientation, as suggested by Kimme et al. [9], observing that the orientation of the edges at the border of a circle points to the center of the circle.

Several other methods use random point selection to reduce complexity, as in [12]. In the work, from Yip et al. [13], a method is proposed to reduce the size of the parameter space by estimating the circle parameters based on local geometric properties. This approach, however, suffers from quantization error problems, which is minimized in [2].

The work presented by Pan et al. [11] proposes a fast and robust algorithm for detecting circular objects based on an extension of the Hough transform. It is based on a probabilistic polling scheme by pairs to find circular objects under occlusion in noisy images with moderate deformations and uses a search algorithm to find several circular objects. A modified method based on the basic algorithm of the Hough transform for circles, but without the use of trigonometry, was proposed by Djekoune et al. [5] in order to improve the computational performance of the voting process for a good precision and robustness of circle detection in a binary image. More recently, Djekoune et al. [4] used the incremental and parallel properties to reduce the resources requirement and the computation time of HT for circles. Despite improvements for memory storage and time processing, these methods use the same standard voting process where each point correspond to one integer vote on the parameter space.

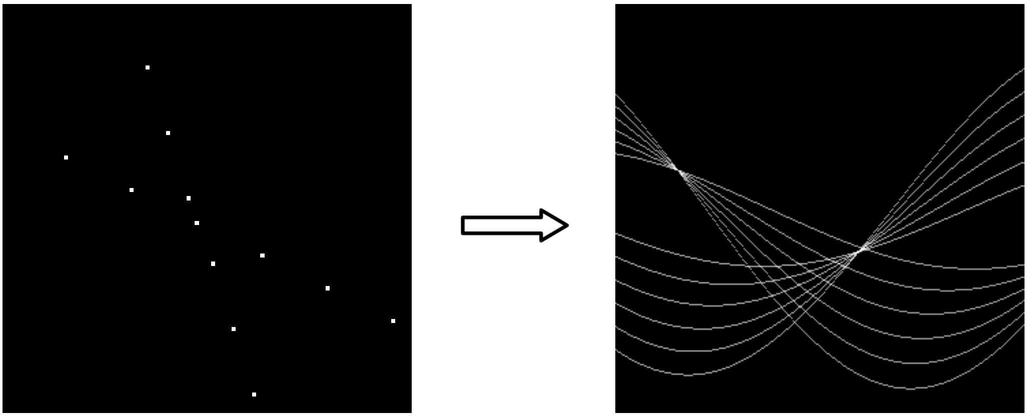
Outside the scope of the Hough transform, an approach was proposed by Cuevas et al. [8] presenting a method that uses differential evolution optimization for the detection of circles. The author takes into account that the detection process is considered similar to a combinatorial optimization problem. They propose an algorithm that uses the combination of three points of the edge as parameters to determine candidate circles in the scene and uses an objective function that determines if some candidate circles are actually present in the image.

Differently from the usual approaches whose goal is to improve performance in terms of processing time or data storage, our work deals with the precision of detection, in order to reduce the appearance of false circles and, at the same time, ensure the maximum detection of true circles in the image.

3 HOUGH TRANSFORM FOR LINES

Developed by Paul Hough in 1962, the Hough transform can detect, in digital images, curves that are easily parameterized, that is, curves that can be parametrized with a known parametric equation, such as lines, circles, ellipses, etc. In the case of line detection, the problem is transformed into a problem of curve intersection in the parameter space. Each line, $y = ax + b$, is identified by a single pair of parameters (a, b), which are angular and linear coefficients, respectively. In the parameter space, each line is then represented by the point (a, b). Likewise, each point $\mathbf{p} = (x, y)$ in the image corresponds to a line $b = x(-a) + y$ in the parameter space so that, varying a and b in this space, all lines in the image space that intersect p are represented. Hence, a line defined by k collinear points $\{(x_i, y_i), i = 1, \dots, k\}$ in the image is identified by the intersection of the k lines associated with the points in the parameter space.

Since a and b assume unlimited values, the parameter space would be unlimited. Alternatively, the polar representation $\rho = x\cos(\theta) + y\sin(\theta)$ for lines, where θ represents the orientation of the line and ρ the distance from the line to the origin, is used to define the parameter space. Fig. 1 illustrates the mapping of a set of points in the image domain to the parameter space. In (a) we have 12 image points along two lines and, in (b), each point is represented by a curve in the parameter space. Notice that each set of 6 curves defines an intersection point. The coordinates of these points of intersection, in the parameter space, are the parameters of the curves that best fit the points in the image.



(a) $p_i = (x_i, y_i)$

(b) $\rho = x_i\cos(\theta) + y_i\sin(\theta)$

Figure 1: Mapping from a set of points in the image domain to the parameters domain. In (a) we have 12 points along two lines and, in (b), each point is represented by a curve in the parameter space. Notice that each set of 6 curves defines an intersection point. The coordinates of these points of intersection are the parameters of the curves that best fit the points in the image

Each point in the image determines one vote for the points of the corresponding curve in the parameter space. Thus, the intersection points will accumulate more votes, determining local maximums in the parameter space. These maximums occur in the parameters of lines that best fit to the points of the image. To construct the parameter space a discretization is adopted in an array of integers, where each position of the matrix corresponds to an interval in the parameter space.

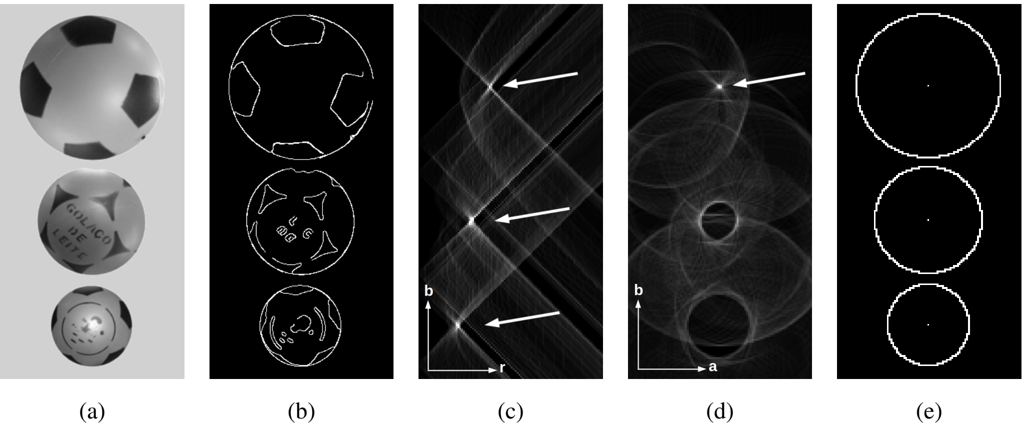
4 MULTIPLE CIRCLE DETECTION WITH HOUGH TRANSFORM

When a set of given points $P = \{(x_i, y_i), i = 1, \ldots, k\}$ describes one circle in the image, the implicit form $(x_i - a)^2 + (y_i - b)^2 - r^2 = 0$ leads to a closed formula to determine the center (a, b) and radius r of the best fit circle to the set of points using least squares. This closed formula is effective as long as the set of points describes a single circle. However, most of the time, we are trying to detect multiple circles at the edge points present in the image. Hence, before using a closed formula, we need to partition the set of edge points into individual circles, which is a more complex task than the curve fitting itself.

The Hough transform is then adapted to map the points (x_i, y_i) of the image to the points (a, b, r) in the $3D$ parameter space, such that $(x_i - a)^2 + (y_i - b)^2 - r^2 = 0$.

Each point (x_i, y_i) determines a surface in the $3D$ parameter space so that the intersections of these surfaces determine points that are the parameters of the circles that best fit to the points of the image. The voting strategy is the same for the case of straight lines, so the search for local maximums in this space provides the parameter lists of the circles present in the image.

The usual implementation of the Hough transform for circles uses the following steps: image smoothing with low-pass filter, edge detection with high-pass filter, application of the Hough transform by voting and identification of local maximums in the parameter space. To perform these steps, the algorithm should receive as input an image I and a threshold for edge detection.



(a) (b) (c) (d) (e)

Figure 2: Steps for detecting circles using the Hough transform. (a) The image is smoothed using Gaussian filter. (b) Edges are detected using Canny filter. (c-d) Accumulated votes in $3D$ parameter space are shown in planar sections with arrows highlighting local maximums. (e) The parameters indicated by the local maximums are used to reconstruct the circles.

Fig. 2 illustrates the steps for detecting circles using the Hough transform. In (a), the image is shown after the smoothing process using Gaussian filter to reduce noise. In (b), edges are detected using Canny filter to enhance edges [1]. After the voting process, each point define a surface in the parameter space. In (c) a planar section of the $3D$ parameter space, for a fixed center coordinate a, is shown. In (d) a planar section for a fixed radius coordinate r is shown. Finally, in (e), the parameters indicated by the local maximums are used to reconstruct the circles.

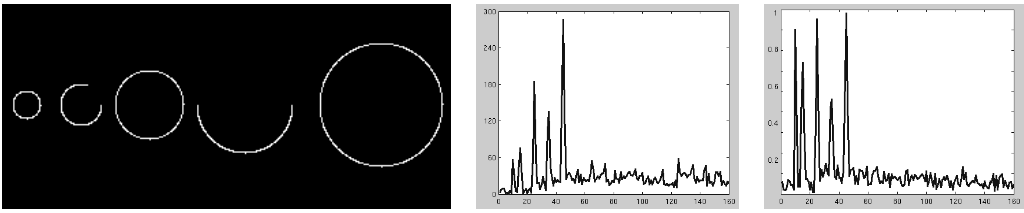
In order to implement the voting step, the parameter space is represented by a $3D$ coordinate system with axes a,b and r, indicating possible values for x center, y center and radius, respectively. A discretization of this space is represented by a matrix $P_{m\times n\times p}$, where m,n and p indicate the number of points sampled uniformly along the axes, a,b and r, respectively. Given an image $I_{N\times M}$ for circle detection, we take $m = M$, $n = N$ and $p = \sqrt{M^2+N^2}$, so that all the points of the image are candidates for center of a circle and the maximum radius is given by the image diagonal size. A set $B = \{(x_i,y_i), i = 1,\dots,k\}$ with k edge points extracted from I will be the entry for the usual implementation of the voting process. The standard voting process is detailed in the Algorithm 1.

Algorithm 1: P=Hough(B,m,n,p)

```
1 $P = zeros(m,n,p)$
2 **for** *each* $(x_i,y_i) \in B$ **do**
3     **for** *each a from* 1 *to m* **do**
4         **for** *each b from* 1 *to n* **do**
5             $r = \sqrt{(x_i-a)^2+(y_i-b)^2}$
6             $P(a,b,r) \leftarrow P(a,b,r)+1$
7         **end for**
8     **end for**
9 **end for**
```

After the voting process, candidates are selected according to a minimum threshold number of votes in the matrix P, that represents the discretized parameter space. Notice that, according to that direct voting process, in line 6 of the algorithm, each candidate circle receives one integer vote, regardless of the radius size. Of course, if a circle have a radius with size of r pixels, the circumference is expected to be described with a number of about $2\pi r$ pixels in the image space, so that in the parameter space, that circle should receive about $2\pi r$ votes. Hence, a circle will have an amount of votes proportional to the radius. Thus, larger circles determine more votes, while smaller circles determine less votes in the parameter space. This becomes a problem when choosing a minimum threshold number of votes to accept the detection of circles of varying radius.

Fig. 3 shows, in (a), an image example with circles of radius $10, 15, 25, 35$ and 45 pixels with full or partial circumferences. Notice in (b) that for direct voting, each circle accumulates votes according to the number of edge points describing it, ignoring the radius and fraction of the circumference that is described by the points. This direct vote causes the fourth circle, partially de-



(a) (b) (c)

Figure 3: Examples of voting in the parameter space. (a) Border image detecting circles of various radius with full or partial circumferences described. (b) The 5 peaks in the graph show that, in direct voting, each circle accumulates votes proportionally to the number of edge points describing it, regardless of the radius size or the portion of the circumference that is described by the points set. (c) In proportional voting, full circles receive more votes, while partial circles accumulate less points, regardless of the size of the radius.

scribed, to have more votes than the first circle, fully described by the edge points. No minimum threshold will be sufficient to select only the first, third, and fifth circles, which are complete.

In our implementation, we propose to modify the algorithm to ensure that circles receive votes so that the same minimum threshold will appropriately select circles with a more complete circumference described, regardless of the radius, by changing line 6 of the 1 Algorithm to

$$P(a,b,r) \leftarrow P(a,b,r) + \frac{1}{2\pi r},$$

so that the accumulated votes express the portion of the circle that is described by the edge points. This strategy, obtained with a simple adaptation in line 6 to the Algorithm, prevents each candidate circle from receiving direct voting equal to 1, making each vote inversely proportional to the radius.

This method of inversely proportional voting makes the complete circle more relevant than the partial circle, independent of the radius, as shown in Fig. 3-(c). Notice that the three complete circles accumulate more votes, while the partial circles accumulate votes proportionally to the portion of the arc that the points describe. While direct voting uses integers to count number of image points, the proposed approach uses scalar values between 0 and 1 to express the portion of the arc described by the points. In our experiments, we showed that this voting strategy allowed a better compromise between avoiding False Positive (FP) or detection of false circles, without creating False Negative (FN) or elimination of true circles.

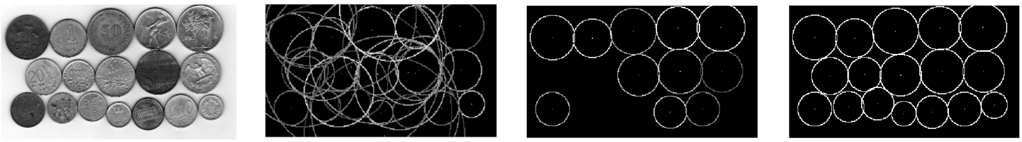
Notice that the preprocessing steps of image smoothing with Gaussian filter and edge detection with Canny filter remains the same for circle detection using the standard voting process or our proposed approach. These parameters are adjusted according to each image in order to best detect edge points and, then, the same set of edge points are used in both approaches for circle detection. Hence, the parameter adjustment for Gaussian filter and Canny filter wont favor any of the approaches.

Recent improvements on Hough transform for circles mostly focus on the time performance and memory storage, as in [4], where standard voting process is used. On the other hand, our method uses a modified voting process to improve the accuracy, reducing FP and FN, without change in processing time or memory storage.

5 EXPERIMENTS AND RESULTS

Our experiments use real images containing circles to show the advantage of our modified voting process in comparison with standard voting process, as used in [4]. In the first experiment, we used a real image to illustrate the difficulty in choosing the minimum amount of votes needed to detect the circles present in the image.

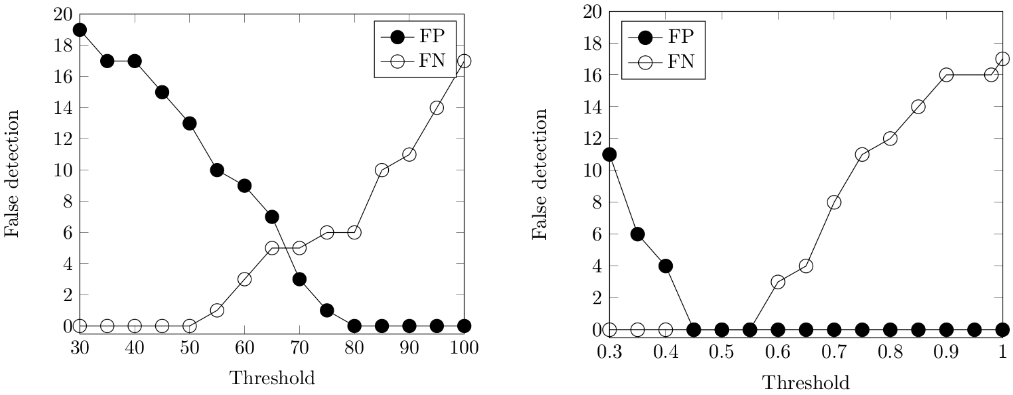
Fig. 4 shows, in (a), a set of 17 coins of varying sizes and colors. For the standard voting process, we found experimentally that a threshold of 50 or less votes is necessary to detect all true circles, avoiding FN (b). At the same time, a threshold of 80 or more votes is necessary to prevent false circles, avoiding FP (c). As its impossible to set a value that is less than 50 and greater than 80, it is impossible to detect all true circles while avoiding false circles using the standard voting process. Using our proposed voting process, a threshold of 0.5 is sufficient to detect all true circles without the appearance of false circles (d).



(a) (b) (c) (d)

Figure 4: Comparison on the multiple circles detection. (a) Real image with 17 coins. (b) Using standard voting and threshold 50, some false circles are detected. (c) Using threshold 80 some true circles are removed. (d) All true circles are detected without the appearance of false circles using the proposed voting process.

Considering the numbers of FP and FN we adopted as the best results for each image the minimum of the number FP+FN. Fig. 5 considers the same image from Fig. 4 to show a graphic with the evolution on the performance of the standard and proposed approach while the threshold is increased. From Fig. 5 notice, in (a), that the standard approach reaches best result for the threshold 80 where we have FP+FN = 6. For the same image, the proposed approach reaches best results for any threshold between 0.45 and 0.55, where FP+FN=0, as shown in (b). This criterion is followed to present best results with each method in the next experiments.



(a) Standard approach.

(b) Proposed Approach.

Figure 5: Evolution on the performance of the standard and proposed approach while the threshold is increased for the image from Fig. 4.

In our second experiment, a series of 7 real images are treated to detect multiple circles using direct voting and our inversely proportional radius voting to compare results. In Fig. 6, the first column shows the actual images, the second column shows the detected edge points, in the third column we have the circles detected with the standard direct voting process and, in the last column, the circles detected with our voting process inversely proportional to the radius. The best results, considering the minimum value of $FP+FN$, obtained with each method are presented. Notice that, with the direct vote, only two of the images have the correctly detected circles and, in the others, occurred the appearance of false circles (lines 1 and 5) or the disappearance of true circles (lines 4, 6 and 7). On the other hand, using our voting strategy all circles were correctly detected without the appearance of false circles and without the disappearance of true circles. Notice also that, in lines 2 and 3, standard direct voting was able to correctly detect circles. In these cases, unlike the others, there is no overlap or circles tangency, making the identification and detection of distinct circles simpler.

6 CONCLUSIONS AND FUTURE WORK

The proposed algorithm presented an improvement of the Hough transform to detect multiple circles in images. Unlike the direct voting process where each vote has weight one in the parameter space, we adopted a strategy where the voting weight is inversely proportional to the radius of the candidate circle. This strategy proved to be more robust in detecting multiple circles by reducing the detection of false circles, as well as avoiding the disappearance of true circles as usually occurs in direct voting. Experiments considering a set of test images allowed to confirm the advantages of the proposed strategy. With direct voting, it's not always possible to choose the appropriate minimum number of votes to adequately detect the circles present in the image, since the candidates accumulate votes proportionally to the radius, making the task of parameter

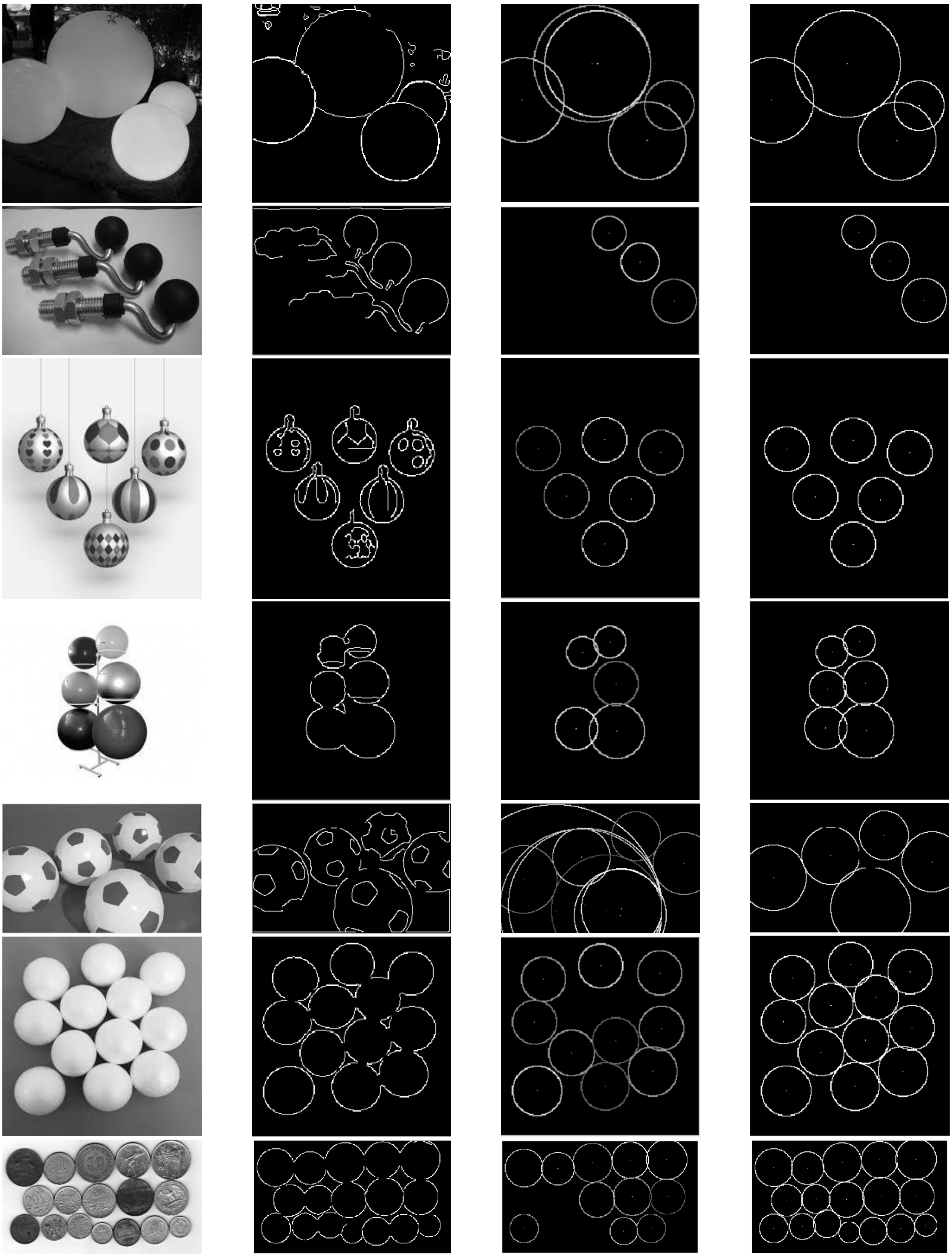


Figure 6: Comparison of results. The first column shows a set of 7 real images. In the second column, the edge points are detected. In the third column, the circles detected with the standard direct voting process. In the last column, the circles detected with our voting process inversely proportional to the radius.

adjustment an exhaustive and not always successful work, since compromise between removing false circles without removing true circles is not always possible. On the other hand, with our strategy of votes inversely proportional to the radius, all true circles accumulate the same amount of votes regardless of radius size, making it more realistic to choose the minimum value of votes that only selects true circles. In spite of the greater efficiency of the method, the automatic selection of the optimal threshold parameters remains a challenge for both approaches, and must be independently adjusted according to each image. In the continuity of this work, we intend to develop strategies for such parameters to be adjusted automatically according to the characteristics of the image to be processed.

RESUMO. A detecção de retas e curvas em imagens é uma tarefa de grande importância em diversas aplicações, como no reconhecimento de objetos e reconstrução de cenas. Apesar de haver fórmulas fechadas para o ajuste de curvas a um conjunto de pontos dados, se os pontos descrevem mais de uma instância do objeto, como dois círculos por exemplo, não há uma fórmula fechada para obtenção dos parâmetros individuais sem a informação a priori de quais pontos pertencem a cada objeto. Como é recorrente a existência de múltiplos objetos em imagens, esse trabalho apresenta uma abordagem, baseada na *Transformada de Hough*, para detecção de múltiplos círculos em imagens. Em nossa implementação, uma adaptação no processo de votação tornou a detecção mais robusta, com redução do número de falso positivos e maior precisão na detecção conforme apresentamos nos experimentos.

Palavras-chave: visão computacional, transformada de hough, detecção de círculos.

REFERENCES

[1] J. Canny. A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, **8**(6) (1986), 679–698.

[2] T.C. Chen & K.L. Chung. An Efficient Randomized Algorithm for Detecting Circles. *Computer Vision and Image Understanding*, **83**(2) (2001), 172 – 191.

[3] E.R. Davies. Image Space Transforms for Detecting Straight Edges in Industrial Images. *Pattern Recognition Letters*, **4**(3) (1986), 185–192.

[4] A.O. Djekoune, K. Messaoudi & K. Amara. Incremental circle hough transform: An improved method for circle detection. *Optik - International Journal for Light and Electron Optics*, **133** (2017), 17 – 31.

[5] A.O. Djekoune, K. Messaoudi & M. Belhocine. A New Modified Hough Transform Method for Circle Detection. In "IJCCI 2013 - Proceedings of the 5th International Joint Conference on Computational Intelligence, Vilamoura, Algarve, Portugal, 20-22 September, 2013" (2013), pp. 5–12.

[6] R.O. Duda & P.E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications ACM*, **15**(1) (1972), 11–15.

[7] P. Hough. Method and Means for Recognizing Complex Patterns. U.S. Patent 3.069.654 (1962).

[8] E.V.C. Jiménez, D. Zaldivar, M.A.P. Cisneros & M.A. Ramírez-Ortegón. Circle detection using discrete differential evolution optimization. *Pattern Analysis and Applications*, **14** (2010), 93–107.

[9] C. Kimme, D. Ballard & J. Sklansky. Finding Circles by an Array of Accumulators. *Commun. ACM*, **18**(2) (1975), 120–122.

[10] F. Matsumoto & S. Tsuji. Detection of Ellipses by a Modified Hough Transformation. *IEEE Transactions on Computers*, **27** (1978), 777–781.

[11] L. Pan, W.S. Chu, J.M. Saragih, F.D. la Torre & M. Xie. Fast and Robust Circular Object Detection With Probabilistic Pairwise Voting. *IEEE Signal Processing Letters*, **18**(11) (2011), 639–642.

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

[13] R.K. Yip, P.K. Tam & D.N. Leung. Modification of hough transform for circles and ellipses detection using a 2-dimensional array. *Pattern Recognition*, **25**(9) (1992), 1007 – 1022.