

## Research Article

# Fast Parabola Detection Using Estimation of Distribution Algorithms

**Jose de Jesus Guerrero-Turrubiates,<sup>1</sup> Ivan Cruz-Aceves,<sup>2</sup>  
Sergio Ledesma,<sup>1</sup> Juan Manuel Sierra-Hernandez,<sup>1</sup> Jonas Velasco,<sup>3</sup>  
Juan Gabriel Avina-Cervantes,<sup>1</sup> Maria Susana Avila-Garcia,<sup>1</sup>  
Horacio Rostro-Gonzalez,<sup>1</sup> and Roberto Rojas-Laguna<sup>1</sup>**

<sup>1</sup>Division de Ingenierias, Campus Irapuato-Salamanca (DICIS), Universidad de Guanajuato, Carr. Salamanca-Valle Km 3.5+1.8, Palo Blanco, 36885 Salamanca, GTO, Mexico

<sup>2</sup>CONACYT, Centro de Investigacion en Matematicas (CIMAT), A.C., Jalisco S/N, Col. Valenciana, 36000 Guanajuato, GTO, Mexico

<sup>3</sup>CONACYT, Centro de Investigacion en Matematicas (CIMAT), A.C., Fray Bartolome de las Casas 314, Barrio La Estacion, 20259 Aguascalientes, AGS, Mexico

Correspondence should be addressed to Juan Gabriel Avina-Cervantes; avina@ugto.mx

Received 24 September 2016; Revised 4 January 2017; Accepted 15 January 2017; Published 21 February 2017

Academic Editor: Amparo Alonso-Betanzos

Copyright © 2017 Jose de Jesus Guerrero-Turrubiates et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a new method based on Estimation of Distribution Algorithms (EDAs) to detect parabolic shapes in synthetic and medical images. The method computes a virtual parabola using three random boundary pixels to calculate the constant values of the generic parabola equation. The resulting parabola is evaluated by matching it with the parabolic shape in the input image by using the Hadamard product as fitness function. This proposed method is evaluated in terms of computational time and compared with two implementations of the generalized Hough transform and RANSAC method for parabola detection. Experimental results show that the proposed method outperforms the comparative methods in terms of execution time about 93.61% on synthetic images and 89% on retinal fundus and human plantar arch images. In addition, experimental results have also shown that the proposed method can be highly suitable for different medical applications.

## 1. Introduction

In the pattern recognition field, detection of curves in natural or medical images is a significant and challenging problem since relevant information about an object is highly related to the shape of its boundary. Any curve can be detected by using the Hough transform (HT), if this curve can be represented by a parametric equation [1, 2].

Circular Hough transform (CHT) is based on the Hough transform principle and it has been adapted for the detection of circles [3]. For this particular problem, three different parameters  $(x_0, y_0, r)$  that define a circle have to be determined. The parameters  $(x_0, y_0)$  represent the coordinate of the center and  $r$  represents the radius [4]. Although the CHT is able to detect suitable approximation to circles in different

type of images, the computational time to detect a single curve is high. This disadvantage is due to the fact that each single pixel represents a potential center  $(x_0, y_0)$ , and a range of possible radii have to be tested for each particular pixel. On the other hand, the voting accumulator [5] that indicates the parameters of a curve that exists in the image is computed on a matrix where each cell has the number of intersections of the circles formed by taking a single pixel as the center; if the matrix is plotted in a three-dimensional graph, sharp peaks will be visualized in the cells that have more intersections. The parameters of the circle are computed by taking the highest peak in the accumulator. Since the representation of an accumulator is a matrix, two of the three parameters of the circle are represented by the rows and columns; in order to cover the third parameter, a three-dimensional matrix is

needed. Although it is possible that several sharp peaks will be formed in the accumulator, when the radius of the tested circle is close to the radius of the circle, these peaks do not represent a real circle. Therefore, once the accumulator has been computed, the next challenge is to find the peak that actually represents the real circle. Due to the nature of the CHT, it would be virtually impossible to find multiple circles in an image; therefore, some studies have proposed different techniques to solve this issue by using techniques such as genetic algorithms [6] or Harmony Search (HSA) [7]. Ayala-Ramirez et al. [6] optimize the computational time of the CHT, by applying a strategy consisting in selecting three random pixels from the edge image as the chromosomes for the genetic algorithm. These three points are used to calculate a center and radius of a circle, and the fitness function evaluates how many pixels of a virtual circle are actually present in the real edge image.

The problem of detecting parabolas can be accomplished by finding the basic parameters of the general equation for parabolic shapes, which are the vertex in  $(x_0, y_0)$ , the angle, and its aperture. In literature, different methods have been proposed to detect parabolas using the Hough transform as a baseline. Maalmi et al. [8] proposed a method that applies genetic algorithms to perform the voting process of the Hough transform. The detected curves are horizontal and the aperture is fixed. This implementation was reported for the detection of crack defects in B-scan images. Oloumi and Rangayyan [9] applied the Hough transform for parabola detection using three parameters:  $(x_0, y_0)$  that represent the vertex and  $a$  to describe the latus rectum or aperture. The main application reported of this method

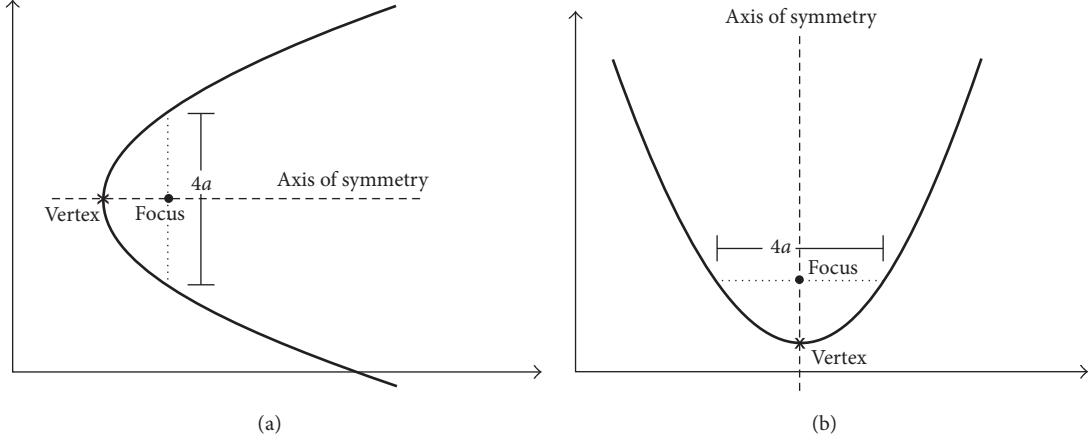


FIGURE 1: Parametric curves described by (2) and (3), respectively.

shapes are explained in detail. Experimental results are presented and discussed in Section 3, and conclusions are given in Section 4.

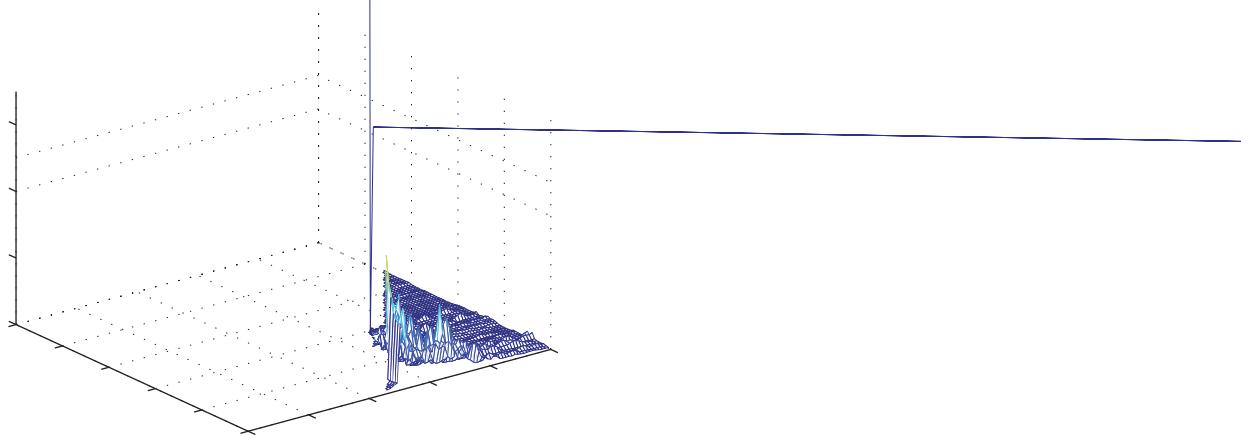
## 2. Methods

**2.1. Parabola Detection.** The detection of curves can be achieved by exploring the duality between points on a curve and the parameters representing that curve; this method is known as Hough transform (HT) [28]. Geometric curves such as circles, ellipses, parabolas, and hyperbolas can be parameterized in a polar coordinate system  $(r, \beta)$ , if the equation that represents the curve satisfies one of the following two equations:

$$\begin{aligned} r &= \frac{de}{1 \pm e \cos \beta}, \\ r &= \frac{de}{1 \pm e \sin \beta}, \end{aligned} \tag{1}$$

where  $e$  represents the eccentricity and  $d$  is the shortest distance between the focal points and the directrix of the curve. Since the present work is focused on the detection of parabolic curves, in these equations, the  $e$  parameter must have a value of 1. The most common parameters to identify a parabola are the vertex  $(x_0, y_0)$ , the coordinates  $(f_x, f_y)$  of the focus, the orientation  $\theta_0$  of the axis of symmetry with respect to the coordinates axes, and the coefficients of (1).

On the other hand, the equation in the Euclidean space



```

Input: PopulationSize, GenerationSize, SelectionPercentage, nbits
Output: Sbest
(1) Population ← InitializePopulation(PopulationSize, nbits)
(2) EvaluatePopulation (Population)
(3) Sbest ← GetBestSolution(Population * SelectionPercentage)
(4) for gen to GenerationSize do
(5)   Selected ← SelectFitSolutions(Population, SelectionSize)
(6)   V ← CalculateFrequencyOfComponents(Selected)
(7)   Offspring ← ∅
(8)   for i to PopulationSize do
(9)     Offspring ← ProbabilisticallyConstructSolution(V)
(10)    Entr ← EntropyMeasurements(Offspring, Model)
(11)   end
(12)   EvaluatePopulation(Offspring) Sbest ← GetBestSolution(Offspring)
(13)   Probs ← ProbabilityMeasurements(Sbest)
(14)   Population ← GetElitePopulation(Probs, Offspring, Population)
(15)   if StopCondition(Entr) then
(16)     Break
(17)   end
(18) end
(19) return (Sbest)

```

ALGORITHM 1: Fast parabola detection method by UMDA.

TABLE 1: Example of an individual with 3 indexes where each index represents an 8-bit pixel position.

Individual																	
Index i						Index j						Index k					
0	0	0	1	0	1	1	0	0	0	1	1	1	0	1	0	0	1

Finally, the coordinate values of the vertex and the aperture of the parabola can be computed by using the following:

$$\begin{aligned}
 x_{\text{vertex}} &= -\frac{B}{(2A)}, \\
 y_{\text{vertex}} &= C - \frac{B^2}{(4A)}, \\
 4p &= \frac{1}{A}.
 \end{aligned} \tag{7}$$

To illustrate the representation of a potential solution in an optimization model fitting process, Table 1 shows an example of an individual formed with the integer indexes {22, 58, 39} represented in base-2 numeral system that can be coded by 8-bits string. This individual is formed by concatenating the binary string indexes {i, j, k} of the three selected pixels in a single vector as it was used by Ayala-Ramirez et al. [6]. The number of genes for each individual directly depends on the number of potential pixels N in the spatial image domain.

Since the proposed method uses the indexes to form all the individuals, the UMDA method can easily eliminate unfeasible solutions by using a function to measure the quality of an individual.

**2.3.2. Fitness Function.** To evaluate the fitness of an individual, a binary image  $I_{\text{VS}}$  (virtual shape) of the same size that

the input image is generated. The value of all the pixels on the virtual image are set to zero, and this leads to a black color image. Then, the vertex and aperture  $4a$  of the parabola represented by (5) are computed using the values A, B, and C obtained from the individual representation. Subsequently, the edge points of the parabola are computed and stored in  $I_{\text{VS}}$  by setting to 1 the corresponding pixels. Finally, the Hadamard product [36] between the binary form  $f_{\text{binary}}$  of the input image and the virtual image is calculated as follows:

$$\text{Hd} = f_{\text{binary}} \odot I_{\text{VS}}. \tag{8}$$

The resulting image Hd contains only those pixels where  $f_{\text{binary}}$  and  $I_{\text{VS}}$  match. Figure 3(a) illustrates an example with the three points taken from an individual to form a parabola in the virtual shape. It can be seen that the Hadamard product obtains partial information of the real parabola. Figure 3(b) shows an example where the three points taken from the individual are part of a parabola that fits perfectly with the parabolic shape of the input image.

The fitness function used to assess the quality of potential solutions is the number of pixels resulting of the Hadamard product. Given that the binary image of the virtual shape is initialized with all pixels on black (intensity value of zero), with more matching pixels in Hd more white pixels will appear.

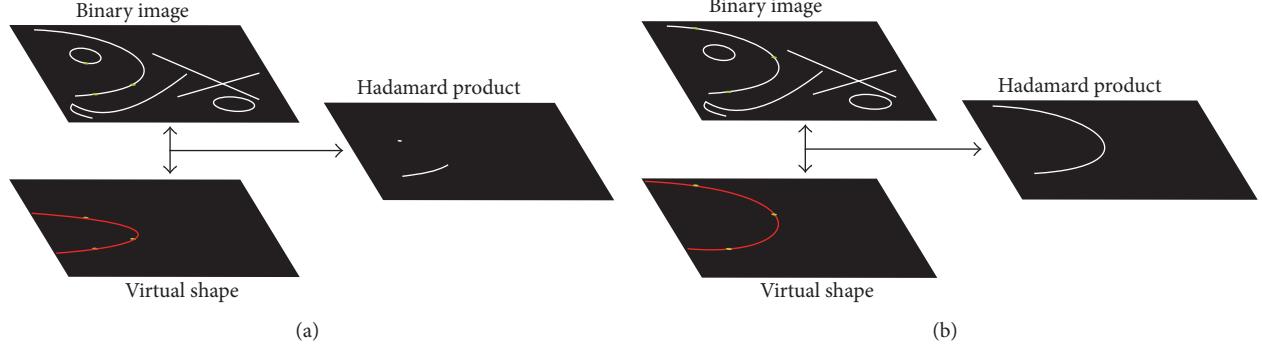


FIGURE 3: Hadamard product between virtual and input images.

TABLE 2: UMDA parameters for all the computational experiments.

Parameter	Value
Number of individuals	10
Selection rate	0.6
Maximum number of generations	30

### **3. Results and Discussion**

In this section, the proposed method for parabola detection is applied on synthetic images and medical images of the retina and human plantar arch. In order to assess the proposed method, it is compared with the Hough transform by applying the algorithm of Sanchez found in the MATLAB central [25] and by running the Hough transform for parabolic shapes algorithm provided by the MIPAV [26] software, that can be downloaded from the website [37]. However, considering that MIPAV and MATLAB parabola detection implementations are based on Hough transform, RANSAC algorithm [21] was implemented to analyze the performance of the proposed algorithm (cf. Tables 3 and 4).

The computational implementations are performed by using the MATLAB version 2013b, on a computer with an Intel Core i5, 4 GB of RAM, and 2.4 GHz processor. Moreover, computational experiments using UMDA were performed using 30 runs in order to perform a statistical analysis of the stochastic process applying the parameter values presented in Table 2.

These parameter settings were determined based on the solutions that give the best tradeoff between precision (lowest RMSE) and computational time using 30 trials to determine the best set of parameters. Moreover, different works were taken into account such as [23].

**3.1. Application on Synthetic Images.** The first experiment was performed by using synthetic images like the one in Figure 4(a). This synthetic image was generated by drawing randomly located parametric objects such as lines, circles, and a parabola in order to evaluate the algorithm in a controlled way.

Table 3 presents a comparative analysis using Figure 4(a). The method presented in [25] shows the largest execution time, given that the result is the execution time per pixel and this image contains 125,899 pixels. Hence, this method

TABLE 3: Comparative analysis of execution time using the binary image (Figure 4(a)).

Method	Execution time (s)		Number of iterations	
Proposed method	Minimum	1.7141	Minimum	15
	Maximum	3.5928	Maximum	28
	Mean	2.3756	Mean	19.55
	Median	2.1934	Median	19
Hough transform [25]	8.8205 per pixel		—	
MIPAV [26]	37.68		—	
RANSAC [27]	16.35		5000	

TABLE 4: Comparative analysis of execution time using the skeleton of the binary image (Figure 4(c)).

Method	Execution time (s)		Number of iterations	
Proposed method	Minimum	0.2820	Minimum	13
	Maximum	3.2988	Maximum	29
	Mean	1.4163	Mean	17.33
	Median	1.4326	Median	18
Hough transform [25]	8.8205 per pixel		—	
MIPAV [26]	36.47		—	
RANSAC [27]	14.86		5000	

depends entirely on the number of potential pixels. The proposed method performs a reduction of 93.61% of the execution time achieved with the MIPAV software, given that the UMDA method has 10 individuals and taking into account the mean of the number of iterations, only 195 function evaluations are required to obtain the best result as it is shown in Figure 4(b).

To ensure that the algorithm is robust and results are consistent with different input conditions, the complexity of the image was increased by adding “salt and pepper” noise. The test was performed by obtaining the skeleton of Figure 4(a) and adding a 10% of “salt and pepper” noise, as it is shown in Figure 4(d). In Table 4, a comparison of the three methods using Figure 4(d) is illustrated. Since the method in [25] works over a single pixel, the result is consistent with Table 3. The proposed method achieves a reduction of 96.14% in comparison with the MIPAV software. Considering that the amount of pixels is lower than the binary image, the number of iterations is congruent with the reduction in the execution

TABLE 5: Comparative RMSE for synthetic and real images shown in Figure 7.

Algorithm	Parameters $(h, k) P$	Time (s)	RMSD	uRMSD*	Matched points
Proposed method	$(468, 276) -40.92$	4.464	9.330	64.230	1439
RANSAC [27]	$(468, 301) -44.55$	12.993	8.453	211.137	1625
Proposed method	$(502, 512) 135.2$	15.490	13.895	426.277	2752
Proposed method	$(464, 278) -43.74$	4.022	9.641	77.995	1939
RANSAC [27]	$(431, 259) -54.18$	27.099	13.932	340.910	1258
Proposed method	$(508, 522) 130.6$	4.357	9.712	182.037	4178

\* Unnormalized RMSD/RMSE.

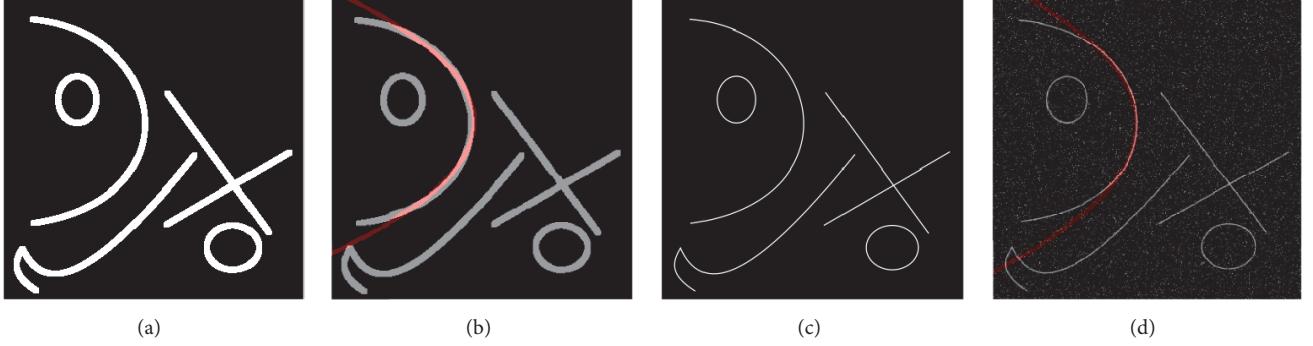


FIGURE 4: Parabola detection on synthetic image using the proposed method.

time. In this test, the average number of function evaluations to obtain the best result using UMDA was 173 (Figure 4(d)).

In order to quantify the fitness of detected parabolas, ten synthetic images with parabolas were generated using (5) with random values of  $A$ ,  $B$ , and  $C$ . *Salt and pepper* noise in the regular range of  $[0, 25\%]$  was added to the images giving a total of 260 test images. Normalized root-mean-square deviation (RMSD) or root-mean-square error (RMSE) was computed to compare original parabolas to the one detected by the proposed algorithm, according to the following:

$$\text{RMSD} = \sqrt{\frac{1}{n} \sum_{t=1}^n (x_{\text{real}} - x_{\text{detected}})^2}. \quad (9)$$

To obtain a true normalized version, (9) can be divided by  $\max(x_{\text{real}})$ ,  $\text{mean}(x_{\text{real}})$ , or  $\text{Cardinality}(x_{\text{real}} == x)$ . In our experiments, the Cardinality was used to prioritize the fitting curves having maximum number of matched points.

A test by adding “salt and pepper” noise over the range  $[1, 25]$  percent to Figure 4(c) (skeleton image) was carried out in order to evaluate the performance of the proposed method in terms of computational time with different amounts of noise. Figure 5(a) shows the average execution time of the obtained results. Since the image has fewer potential pixels than the binary image, the execution time with 1% of noise was low in comparison with previous results. Considering that the UMDA method is a stochastic algorithm, its performance is not shown as a straight line; however, the mean time tends to increase when the amount of noise is also increased. Figure 5(b) shows the standard deviation of the performed tests; since the graph tends to a constant mean over all the percentages of noise, it can be assumed that the algorithm is stable and robust in the presence of noise over the time (i.e., noise may affect quality results). This time stability is

originated by the normalized RMSD metric, because more noisy pixels are considered as part of the detected parabola while the global error is also increasing. However, it is observed that most methods (RANSAC, MIPAV, and Hough transform) fail to detect correctly the parabola parameters on images having more than 25% of *salt and pepper* noise, and the proposed method could detect parabolas on synthetic and real images contaminated until 30%.

Figure 6(a) is a synthetic image for fitness testing, and Figures 6(b) and 6(c) present detected parabolas using our proposed algorithm and MIPAV software, respectively. RMSD values for both times of individual detection are 0.0026 and 0.0089, demonstrating the validity of our method. In Figure 7, comparative results between the proposed method and RANSAC are depicted, and Table 5 shows the normalized errors in parabola detection on original images and with 15% of noise. The fast and stable performance of the proposed algorithm compared to RANSAC is noticed; besides, RANSAC has many parameters to tune and is computationally expensive and there is no warranty for reproducing the same results in a new evaluation.

**3.2. Retinal Fundus Images.** The detection of parametric objects has been applied in different areas of engineering. In medical imaging, the detection of parabolas in retinal fundus images has a particular importance, since the form of the retinal vessels can be approximated to the parabola parametric form. In the previous work, the standard model, the Hough transform, has been used to find the parabola that fits the best on the retinal images. For instance, in the work reported by Oloumi et al. [10, 11], the form of the retina vessels is approximated to a parabola with the aim of monitoring the measurements of the openness of the major temporal arcade (MTA). This study facilitates the quantitative analysis of the

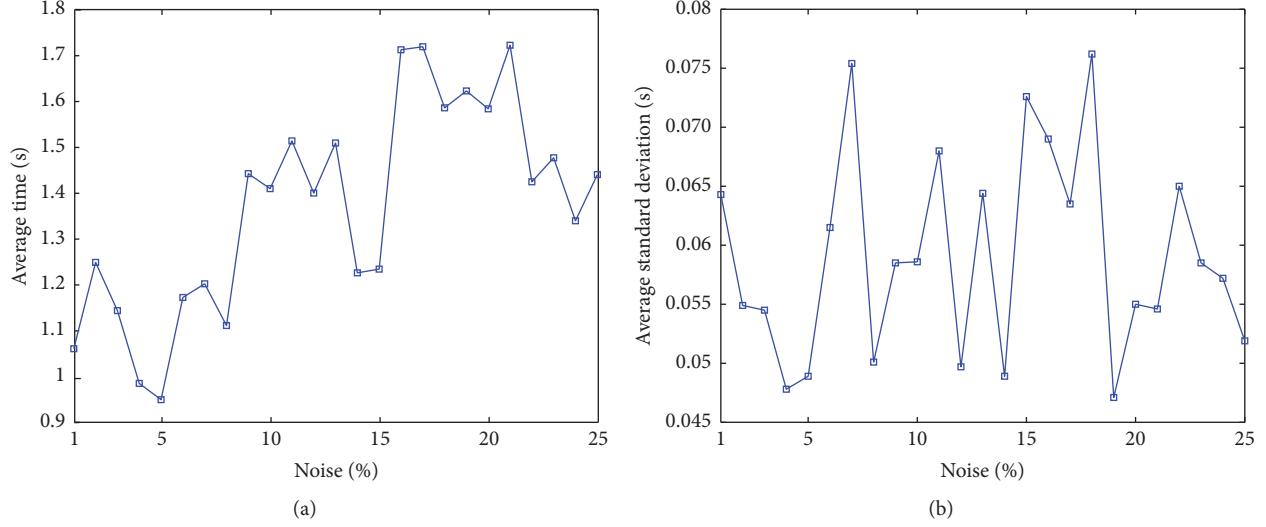


FIGURE 5: Average computational time and standard deviation (in seconds) over 30 runs of UMDA per percentage noise for the skeleton image (Figure 4(c)).

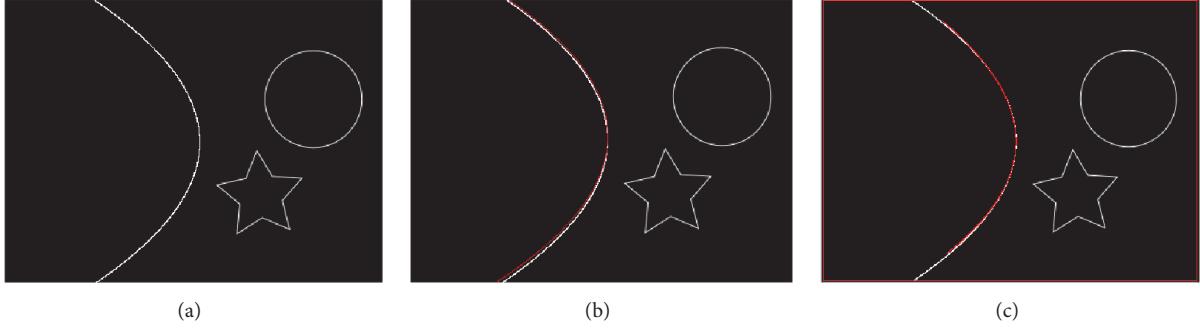


FIGURE 6: Parabola detection performed by the proposed approach and MIPAV software.

MTA and overcomes the limitations associated with manual analysis. Another application is reported by Yu et al. [38], where the retinal images are used to find the vertex of all the vessels; this vertex is known as fovea [39]. This study is significant because the position of the vertex can give the grade of diabetic retinopathy.

In the tests reported in the present work, the proposed method was used to approximate the retinal vessels to a parabola. Figure 8 shows the results obtained, where the best fit for the parabola in the image is shown in red color. As in the work reported by Yu et al. [38], the public database DRIVE [40] that contains a set of 20 retinal fundus images was used. Images were preprocessed by applying Gaussian matched filters for vessel detection.

Table 6 presents the statistical analysis of the results obtained by applying the proposed and comparative methods to the DRIVE database. The method presented in [25] remains constant with the execution time per pixel; then, this execution time for this algorithm is the largest time. Since the proposed method accomplishes the best result with only 165 function evaluations, it is possible to achieve a reduction in terms of computational time of 89% in comparison with the one performed on the MIPAV software.

TABLE 6: Comparative analysis of the average execution time using the DRIVE database of retinal fundus images.

Method	Execution time (s)		Number of iterations	
	Minimum	Maximum	Minimum	Maximum
Proposed method	1.2280	6.3285	5	22
	Mean	4.5838	Mean	16.28
	Median	5.1588	Median	19
	Hough transform [25]	8.8205 per pixel	—	—
MIPAV [26]	43.65	—	—	—
RANSAC [21]	16.66	5000	—	—

Given that the skeleton of an image passes through the center section of a set of pixels, if the proposed algorithm is applied to the skeleton of the retinal images, it is then expected for the algorithm to achieve a lower execution time than in the binary image. In Table 7, the statistics of the execution time for the skeleton images are shown. The results are consistent with those obtained for the synthetic image, and as it can be seen, the proposed method clearly outperforms the execution time of HT algorithm from Sanchez and the MIPAV software.

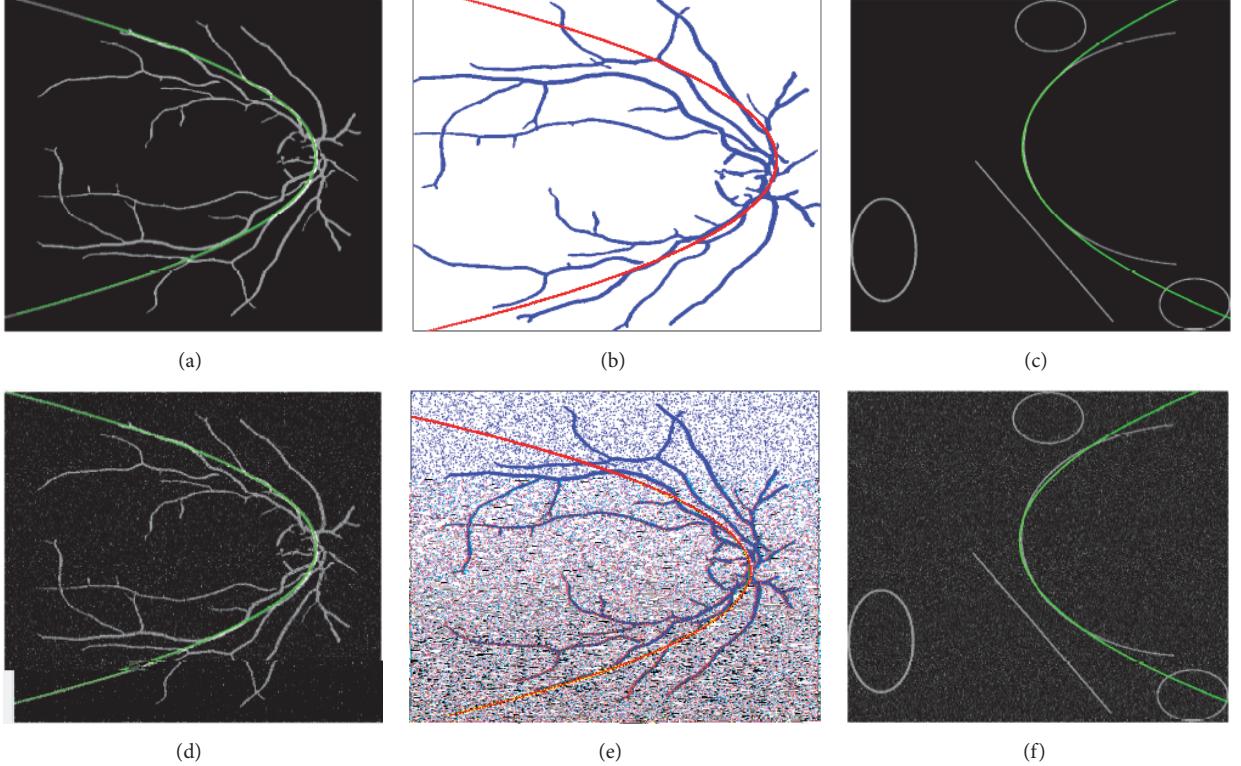


FIGURE 7: Comparative images between RANSAC and the proposed method (noise added). (a) Retinal image process by the proposed algorithm, (b) retinal image processed by RANSAC, (c) synthetic image processed by the proposed method, and (d, e, f) images from (a, b, c) with 15% of *salt and pepper* noise, respectively.

TABLE 7: Comparative analysis of execution time using the skeleton of the retinal fundus images (DRIVE database).

Method	Execution time (s)	Number of iterations
Proposed method	Minimum 0.9550	Minimum 4
	Maximum 5.7105	Maximum 20
	Mean 3.3706	Mean 12
	Median 2.7343	Median 10
Hough transform [25]	8.8205 per pixel	—
MIPAV [26]	41.87	—
RANSAC [21]	12.99	5000

Since the proposed method takes three pixels to compute a parabola, it can be assumed that the number of iterations to achieve the best result is not related to the size of the image. This assumption is validated with the results shown in Tables 6 and 7, where the average number of iterations for both the binary image and the skeleton image is less than 30.

**3.3. Plantar Arch Images.** The form of the plantar arch in humans can conduct to several conditions if the contact area with the floor, given by the plantar arch, is not of the correct size. In clinical practice, the procedure for providing a diagnosis of flatfoot degree is by visual inspection of the specialist. Therefore, the proposed method in this work can be applied to plantar arch images to approximate the heel and the plantar arch to parabolas; hence, these parameters could be used to assist a medical diagnostic. The image database used in this test was created by the authors and approved by

a specialist (Dr. Carlos Reséndiz Ramírez). This database is composed of 80 images of size  $285 \times 707$  pixels in RGB color space of left and right foot collected from different patients. Images were first preprocessed, to reduce the amount of data that was not useful for this study, as described below.

Let  $I$  be a grayscale image of the foot to be analyzed, with  $M$  and  $N$  as the number of pixels for the width and height, respectively; then,

$$I = f(x, y), \quad x \in [0, M - 1], \quad y \in [0, N - 1]. \quad (10)$$

The first step of the method consists in smoothing the image by applying a mean filter, that is, convolving the image with a filter mask:

$$I_{\text{smooth}} = I * f_{\text{smooth}}, \quad (11)$$

where  $f_{\text{smooth}}$  is defined by

$$f_{\text{smooth}} = \begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix}. \quad (12)$$

This filter removes part of the spurious pixels, improving the probability of finding the curves of interest. After the smoothing step, gradients are computed on the image in order to find the edges of the footprint. Canny edge detection is the algorithm used to perform this step. The resulting

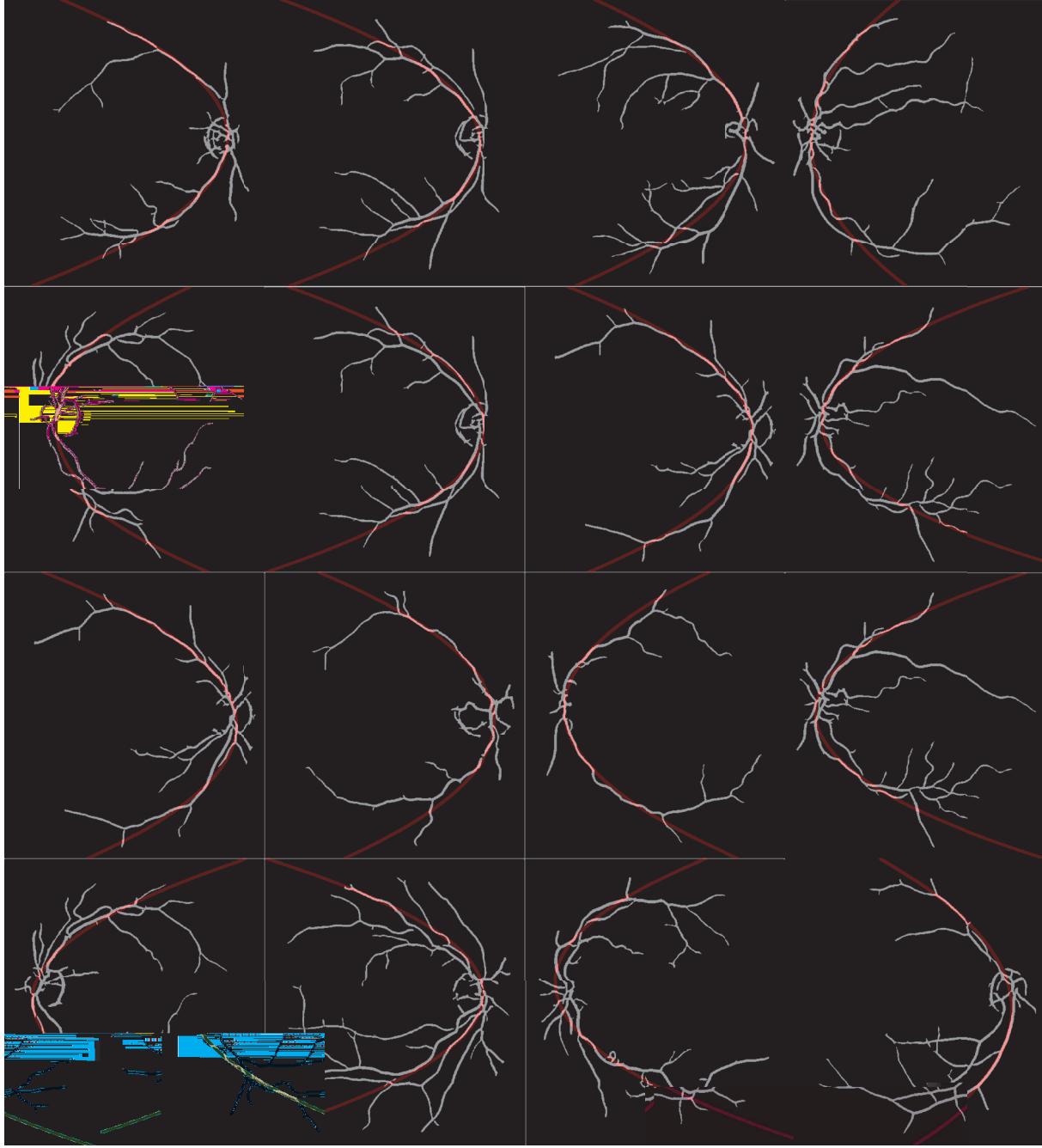


FIGURE 8: Results of parabola detection using the proposed method over a subset of retinal fundus images from the DRIVE database.

contour image  $f_{\text{edge}}$  is then dilated (13) by using a Structuring Element (SE) of radius 3, where the dilation operation closes slightly open contours and makes borders thicker.

$$f_{\text{binary}} = f_{\text{edge}} \oplus \text{SE}. \quad (13)$$

The  $f_{\text{binary}}$  image is the result of the preprocessing stage in the proposed method. This image is referred to as the *edge image*.

In order to address the plantar arch issue, the foot image was divided into three sections from the fingers to the heel, where each section contains 33.33% of the image. The sections

of interest for this study are only those two that include the plantar arch and the heel.

Table 8 presents a comparative analysis of the execution time of the algorithms over the database of the human plantar arch images. As it can be observed, the method developed in [25] maintains the average time per pixel, and so, it takes the largest execution time. The proposed method performs a reduction of 89.96% of the execution time achieved on the MIPAV software; this is because the algorithm requires only an average of 135 function evaluations to obtain the best result. Figure 9 shows a subset of human plantar arch images,

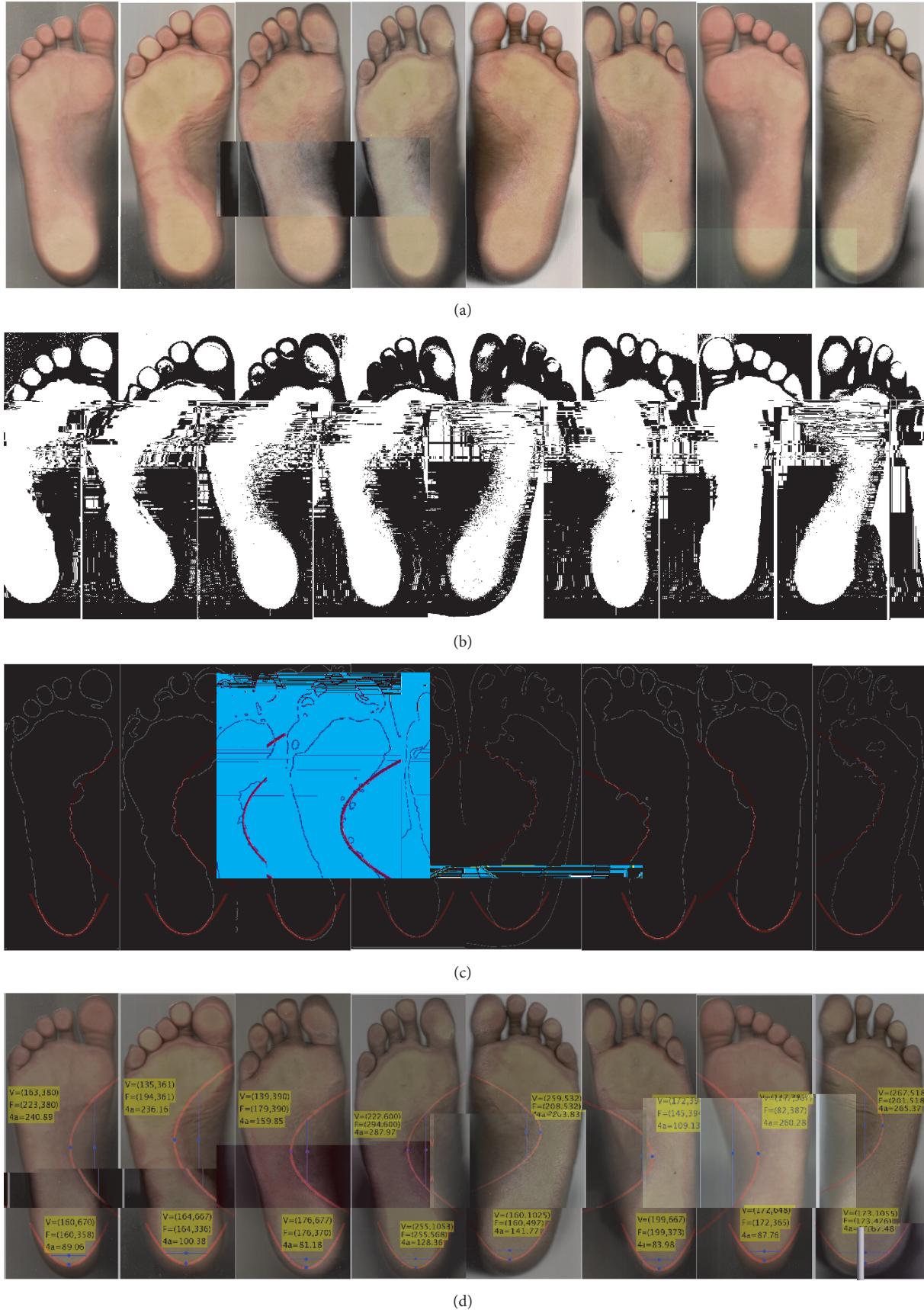


FIGURE 9: (a) Subset of human plantar arch images. (b) Preprocessing step applied to the images in (a). (c) Results of parabola detection (plantar arch and heel) over the edge images. (d) Quantitative analysis of the detected parabolas from the images in (a).

TABLE 8: Comparative analysis of execution time using the database of human plantar arch images.

Method	Execution time (s)	Number of iterations
Proposed method	Minimum 4.2118	Minimum 3
	Maximum 7.1688	Maximum 26
	Mean 6.18	Mean 13.51
	Median 6.40	Median 14.15
Hough transform [25]	8.8205 per pixel	—
MIPAV [26]	63.78	—
RANSAC [21]	22.54	5000

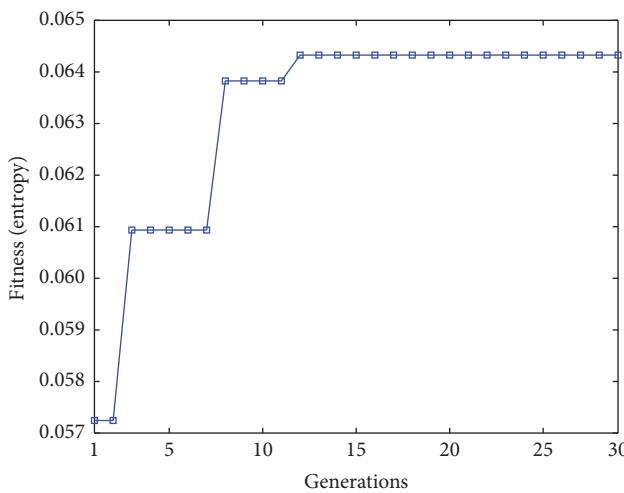


FIGURE 10: Evolution of the average fitness through generations using the database of human plantar arch images.

where the heel and plantar arch were parameterized with the proposed method.

In addition, the comparative results in Table 8 show that the proposed method outperforms the HT algorithm from Sanchez and the algorithm from the MIPAV software in terms of computational time. Figure 9 presents the results of the proposed method in different plantar arch images, where the rows, from (a) to (d), show the real images, the binary images, the edge images with the detected parabolas, and the output of the proposed method.

Finally, Figure 10 shows the average fitness value through the generations of UMDA over the 30 runs of the proposed method. This graph represents the iterations that were required by the UMDA method to find the two parabolas in the image.

## 4. Conclusion

In this paper, a new method based on the Estimation of Distribution Algorithms (EDAs) has been proposed to detect parabolic shapes. The method computes the constant values of the generic parabola equation by selecting three random pixels from the input image. The proposed method was evaluated in terms of computational time and compared with freely available implementations of the parabola Hough

transform. According to experimental results, the average time of the proposed method is significantly better (0.0036 seconds) compared to those obtained by the Hough transform, since this method takes an average of 6.14 seconds to compute the accumulator for one single boundary pixel. In addition, experimental results have also shown that the proposed method outperforms the two comparative methods in terms of execution time saving 93.61% on synthetic images and 89% on the medical images of retinal fundus and human plantar arch.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This work has been supported by the Mexican National Council on Science and Technology under Grant no. 429450/265881 and Cátedras-CONACYT no. 3150-3097. The authors would like to thank the doctors Carlos Reséndiz Ramírez and Hilda Alejandra Sierra-Hernandez for providing their clinical advice and expertise. Moreover, a special recognition is due to the bachelor student Joao Manuel Calvillo-Rodriguez for his support to collect the database of images.

## References

- [1] R. O. Duda and P. E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.
- [2] J. Illingworth and J. Kittler, “A survey of the hough transform,” *Computer Vision, Graphics and Image Processing*, vol. 44, no. 1, pp. 87–116, 1988.
- [3] T. J. Atherton and D. J. Kerbyson, “Using phase to represent radius in the coherent circle Hough transform,” in *Proceedings of the IEE Colloquium on Hough Transforms*, pp. 5/1–5/4, IET, London, UK, May 1993.
- [4] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, “Comparative study of Hough Transform methods for circle finding,” *Image and Vision Computing*, vol. 8, no. 1, pp. 71–77, 1990.
- [5] P. Mukhopadhyay and B. B. Chaudhuri, “A survey of hough transform,” *Pattern Recognition*, vol. 48, no. 3, pp. 993–1010, 2015.
- [6] V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, and R. E. Sanchez-Yanez, “Circle detection on images using genetic algorithms,” *Pattern Recognition Letters*, vol. 27, no. 6, pp. 652–657, 2006.
- [7] E. Cuevas, N. Ortega-Sánchez, D. Zaldivar, and M. Pérez-Cisneros, “Circle detection by harmony search optimization,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 66, no. 3, pp. 359–376, 2012.
- [8] K. Maalmi, A. El-Ouaazizi, R. Benslimane, L. F. C. Lew Yan, A. Voon, and P. Gorria, “Crack defect detection and localization using genetic-based inverse voting hough transform,” in *Proceedings of the 16th International Conference on Pattern Recognition (ICPR ’02)*, vol. 3, pp. 257–260, Institute of Electrical & Electronics Engineers (IEEE), Quebec, Canada, August 2002.

- [9] F. Oloumi and R. M. Rangayyan, "Detection of the temporal arcade in fundus images of the retina using the Hough transform," *Conference proceedings IEEE Engineering in Medicine and Biology Society*, pp. 3585–3588, 2009.
- [10] F. Oloumi, R. M. Rangayyan, and A. L. Ells, "Computer-aided diagnosis of proliferative diabetic retinopathy," in *Proceedings of the 34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS '12)*, pp. 1438–1441, San Diego, Calif, USA, September 2012.
- [11] F. Oloumi, R. M. Rangayyan, and A. L. Ells, "Parabolic modeling of the major temporal arcade in retinal fundus images," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 7, pp. 1825–1838, 2012.
- [12] P. Cai and C. Wang, "An eyelid detection algorithm for the iris recognition," *International Journal of Security and its Applications*, vol. 9, no. 5, pp. 105–112, 2015.
- [13] F. Nabi, H. Yousefi, and H. Soltanian-Zadeh, "Segmentation of major temporal arcade in angiography images of retina using generalized hough transform and graph analysis," in *Proceedings of the 22nd Iranian Conference on Biomedical Engineering (ICBME '15)*, pp. 287–292, Tehran, Iran, November 2015.
- [14] G. Lee and F. E. Pollo, "Technology overview: the gait analysis laboratory," *Journal of Clinical Engineering*, vol. 26, no. 2, pp. 129–135, 2001.
- [15] R. G. Frykberg, L. A. Lavery, H. Pham, C. Harvey, L. Harkless, and A. Veves, "Role of neuropathy and high foot pressures in diabetic foot ulceration," *Diabetes Care*, vol. 21, no. 10, pp. 1714–1719, 1998.
- [16] P. S. Kulkarni and V. B. Kulkarni, "Human footprint classification using image parameters," in *Proceedings of the International Conference on Pervasive Computing (ICPC '15)*, 5, 1 pages, January 2015.
- [17] W. C. Chu, S. H. Lee, W. Chu, T.-J. Wang, and M.-C. Lee, "The use of arch index to characterize arch height: a digital image processing approach," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 11, pp. 1088–1093, 1995.
- [18] S. Zheng, K. Huang, T. Tan, and D. Tao, "A cascade fusion scheme for gait and cumulative foot pressure image recognition," *Pattern Recognition*, vol. 45, no. 10, pp. 3603–3610, 2012.
- [19] M. Salehin, L. Zheng, and J. Gao, "Conics detection method based on pascal's theorem," in *Proceedings of the 10th International Conference on Computer Vision Theory and Applications (VISAPP '15)*, vol. 1, pp. 491–497, VISIGRAPP, 2015.
- [20] T. Merazi-Meksen, M. Boudraa, and B. Boudraa, "Mathematical morphology for TOFD image analysis and automatic crack detection," *Ultrasonics*, vol. 54, no. 6, pp. 1642–1648, 2014.
- [21] P. C. Niedfeldt and R. W. Beard, "Recursive ransac: multiple signal estimation with outliers," *IFAC Proceedings Volumes*, vol. 46, no. 23, pp. 430–435, 2013.
- [22] T. Hassner, L. Assif, and L. Wolf, "When standard RANSAC is not enough: cross-media visual matching with hypothesis relevancy," *Machine Vision and Applications*, vol. 25, no. 4, pp. 971–983, 2014.
- [23] I. Cruz-Aceves, A. Hernandez-Aguirre, and S. I. Valdez, "On the performance of nature inspired algorithms for the automatic segmentation of coronary arteries using Gaussian matched filters," *Applied Soft Computing Journal*, vol. 46, pp. 665–676, 2016.
- [24] R. Armañanzas, I. Inza, R. Santana et al., "A review of estimation of distribution algorithms in bioinformatics," *BioData Mining*, vol. 1, 2008.
- [25] C. Sanchez, *Parabola Detection Using Hough Transform*, 2007, <http://www.mathworks.com/matlabcentral/fileexchange/15841-parabola-detection-using-hough-transform>.
- [26] M. J. McAuliffe, F. M. Lalonde, D. McGarry, W. Gandler, K. Csaky, and B. L. Trus, "Medical image processing, analysis and visualization in clinical research," in *Proceedings of the 14th IEEE Symposium on Computer-Based Medical Systems, (CBMS '01)*, pp. 381–388, July 2001.
- [27] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the Association for Computing Machinery*, vol. 24, no. 6, pp. 381–395, 1981.
- [28] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
- [29] M. Hauschild and M. Pelikan, "An introduction and survey of estimation of distribution algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 3, pp. 111–128, 2011.
- [30] P. Larrañaga and J. A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*, vol. 2, Springer Science & Business Media, 2002.
- [31] P. Larrañaga, "A review on estimation of distribution algorithms," in *Estimation of Distribution Algorithms*, pp. 57–100, Springer, 2002.
- [32] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks. Part 1 (of 6)*, vol. 6, pp. 1942–1948, Perth, Australia, December 1995.
- [33] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, pp. 1–13, American Institute of Aeronautics and Astronautics, Austin, Tex, USA, April 2005.
- [34] M. Pelikan, D. E. Goldberg, and F. G. Lobo, "A survey of optimization by building and using probabilistic models," *Computational Optimization and Applications. An International Journal*, vol. 21, no. 1, pp. 5–20, 2002.
- [35] H. Mühlenbein and G. Paß, "From recombination of genes to the estimation of distributions I. Binary parameters," in *Parallel Problem Solving from Nature—PPSN IV*, vol. 1141 of *Lecture Notes in Computer Science*, pp. 178–187, Springer, Berlin, Germany, 1996.
- [36] E. Million, *The Hadamard Product*, vol. 3 of *Course Notes*, 2007.
- [37] National Institutes of Health Center for Information Technology, *Medical Image Processing, Analysis and Visualization*, 2015, <http://mipav.cit.nih.gov/index.php>.
- [38] C.-Y. Yu, C.-C. Liu, and S.-S. Yu, "A fovea localization scheme using vessel origin-based parabolic model," *Algorithms*, vol. 7, no. 3, pp. 456–470, 2014.
- [39] M. E. Gegundez-Arias, D. Marin, J. M. Bravo, and A. Suero, "Locating the fovea center position in digital fundus images using thresholding and feature extraction techniques," *Computerized Medical Imaging and Graphics*, vol. 37, no. 5–6, pp. 386–393, 2013.
- [40] J. Staal, M. D. Abràmoff, M. Niemeijer, M. A. Viergever, and B. Van Ginneken, "Ridge-based vessel segmentation in color images of the retina," *IEEE Transactions on Medical Imaging*, vol. 23, no. 4, pp. 501–509, 2004.

