

# Reconstruction of Fingerprint Shape using Fractal Interpolation

Abdullah Bajahzar<sup>1</sup>

Department of Computer Science and Information  
College of Science, Majmaah University  
Zulfi 11932, Saudi Arabia

Hichem Guedri<sup>2</sup>

Electronics and Microelectronics Laboratory  
Physics Department, Faculty of Sciences, Monastir  
University, 5019 Monastir, Tunisia

**Abstract**—One of the severe problems in a fingerprint-based system is retaining the fingerprint images. In this paper, we propose a method to minimize the fingerprint images size and retain the reference points. The method is divided into three parts, the first part is about digital image preprocessing that allows us to eliminate the noise, improve the image, convert it into a binary image, detect the skeleton and locate the reference point. The second part concerns the detection of critical points by the Douglas-Peucker method. The final part presents the methodology for the fingerprint curves reconstruction using the fractal interpolation curves. The experimental result shows the accuracy of this reconstruction method. The relative error (ER) is between 2.007% and 5.627% and the mean squared error (MSE) is between 0.126 and 0.009 at a small iterations number. On the other hand, for a greater number of iterations, the ER is between 0.415% and 1.64% and MSE is between 0.000124 and 0.0167. This clearly indicates that the interpolated curves and the original curves are virtually identical and exceedingly close.

**Keywords**—Fingerprint images; enhancement; thresholding process; thinning algorithms; minutiae extraction; Douglas-Peucker algorithm; fractal interpolation; iterated function system (IFS)

## I. INTRODUCTION

In nowadays, the world has become electronically connected and more dynamic because the technological revolution. The traditional identity representations such as passwords and cards cannot be trusted to identify a person. The cards may be lost or stolen, and passwords or PIN numbers may be compromised. In addition, passwords and cards can be easily shared, thus our personal information cannot be secured [1-3]. There are innovative techniques for identifying individuals called biometric identifiers [1]. The biometric identification uses distinct anatomical properties such as fingerprints, facial, iris and behavior (such as speech). The biometrics offers an effective solution to identify people because biometric identifiers cannot be lost or assumed, and represent essentially the individual's body identity.

The fingerprint is considered one of the most practical to identification persons. The fingerprint recognition requires minimal effort from the user, since it allows us to capture unique information necessary for the recognition process, and provides us with relatively excellent performance. Another reason of the fingerprints popularity is the relative minimum price of fingerprint sensors, which for peaceful integration into many companies. But, the downside of this technique is that

they require too much memory or storage space to save all the fingerprint images.

There are specific techniques to describing, analyzing, understanding, modeling, and controlling complex processes. Fractal modeling makes it possible to manage complex elements by starting with a reduced number of classical geometric shapes. Fractal interpolation can be found in many applications at both the 1D, 2D and 3D levels. Moreover, the use of fractal models, both in terms of the signal and the image, has become commonplace and constitutes an active search, motivated by the plethora of possible applications based on this concept. In this paper, a study is developed to reconstruct fingerprint curves using the fractal interpolation method.

Some pretreatment and enhancement steps are frequently performed to simplify the minutiae extraction task [4-8]. The first step of the algorithm concerns the fingerprint image segmentation; this phase requires the conversion of the gray-scale fingerprints image into a binary image [9-11]. The binary images obtained by the binarization process are generally subjected to a thinning step [12-13] which makes it possible to reduce the thickness of the peak line to one pixel [14-15]. Once a binary skeleton has been obtained, a simple image scan can detect the minutiae [16-18]. The further step is to determine the control points implementing the Douglas-Peucker algorithm [19-24]. In the final step, the reconstruction approach was presented by using the fractal interpolation algorithm and the iterated function system (IFS) [25-32].

The rest of the paper is arranged as follows. In Section 2, brief description of the related work. The proposed method is illustrated in Section 3; Section 4 describes a general overview in the fingerprint image segmentation and detection of the minutiae, the Douglas-Peucker algorithm is presented in Section 5, Section 6 describes the fractal interpolation, the illustrative simulation and experimental results are presented in Sections 7, finally followed by the discussion and the conclusion in Sections 8 and 9, respectively.

## II. RELATED WORK

A brief discussion is presented in this section on related work; fractal reconstruction techniques have been widely studied. Lai et al. [25] have used fractal interpolation for the compression and reconstruction of two-dimensional digital elevation model (2D DEM). In their proposed method, they have used the improved Douglas - Peucker (IDP) technique to extract feature points. They used fractal interpolation for

reconstruction, and they proposed a probability-based method to accelerate the fractal interpolation execution. Li et al. [26] have used the fractal interpolation technique for the seismic data reconstruction. They have examined the fractal interpolation method based on previous work and theoretically analyzed a special type of fractal interpolation function. They have ascertained that the numerical results of the fractal interpolation method have a high accuracy and efficiency, the largest error between the theoretical seismograms and the reconstructed seismograms within limits 3%. Cader and Krupski [27] have introduced a new interpolation method for fractal curves. They have studied curves that have a very irregular character, this type of curve has coarse characteristics and has a complex structure at Different metrics, and they numbered among fractals or stochastic fractures - multifractals. They proposed another alternative method of using fractal curves for the complex curves approximation, his method is better than (FIF) for the multifractal structures interpolation. It bases this on the classical notion of interpolation node and introduces non-local values for its description (the fractal dimension). Guedri et al. [28] have introduced the fractal interpolation technique for the human retina blood vessels reconstruction. They have studied the segmentation technique of the retinal image (such as binarization and skeletonization)

and the line simplification by using the Douglas-Peucker method. And finally, they have used fractal interpolation and IFS for the blood vessel curves reconstruction.

### III. PROPOSED APPROACH

The proposed method considers three necessary phases; preprocessing phase of fingerprint images, phase for the characteristic points determination using the Douglas-Peucker algorithm and the fractal interpolation phase. The structure of these phases discussed in the flowchart is illustrated in Fig. 1.

In the proposed method, the first objective is the enhancement and Binarization the fingerprint image which consists of transforming a multilevel image into a black and white image (two levels only). Subsequently the phase of thinning; at this point, the fingerprint is presented as a set of connected curves, while keeping its original topology. After obtaining the image skeleton, the most interesting minutiae points are detected. Subsequently, a lines simplification step applied to reduce the lines complexity (the control point's detection) by implementing the Douglas-Peucker algorithm. In the final step, the fractal interpolation and iterated function systems (IFS) has been used to visualize the result of the fingerprint image reconstruction.

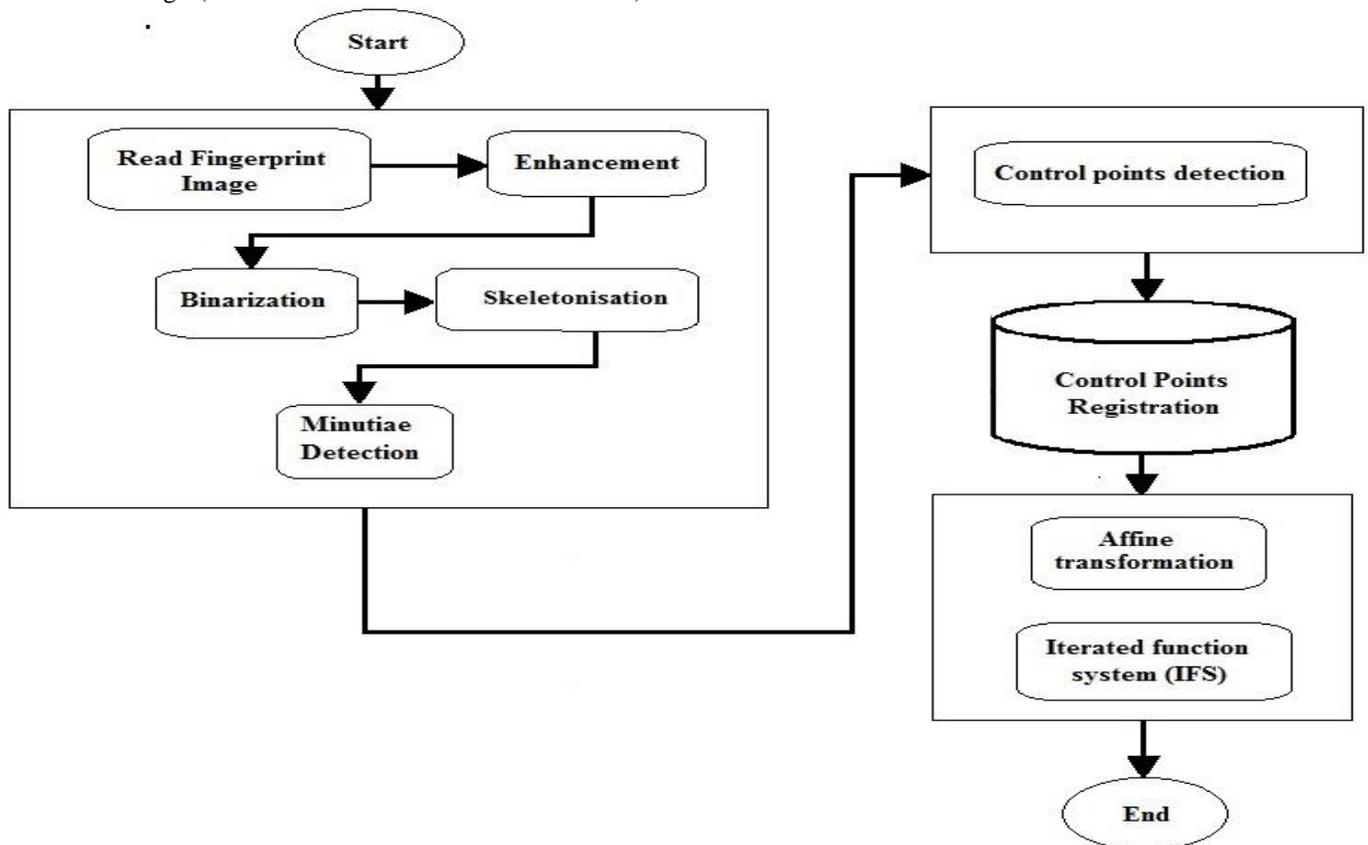


Fig. 1. Schematic Diagram for a Fractal Reconstruction the Fingerprints Image.

#### IV. PRETREATMENT OF FINGERPRINT IMAGES

In this approach, the grayscale image is converted to a binary image before the minutia detection. The proposed method is composed of diverse parts; it includes the following steps:

- **Enhancement:** In the First step, to eliminate unnecessary noise, the Gabor filter algorithm is used, which is based on the technique proposed by Lin Hong [4] to enhance fingerprint images [4-6]. This image is convolved with a Gabor filter with uniform symmetry to improve fingerprints [7-8].
- **Binarization:** the binarization phase is a very important step in image segmentation; the thresholding process is used to create a binary image from grayscale image [9]. In this approach, each pixel is examined and assigned his luminance. If a pixel is in a region where the low luminance (lower luminance to a set ready level value), then it is replaced by the value 0 (black color). However, if a pixel is in a region where the luminance is clearly defined, it is replaced by the value 1 (white color) [10-11].
- **Skeletonisation:** The purpose of skeletonization is to extract a shape feature from an object and present this object through a connected curves set or digital arcs with a reduced data amount or a simplified form, and remain the original object topology [12-15]. There are different categories of skeletonisation methods (Distance Skeleton Algorithms, Critical Points Algorithms, and Thinning Algorithms); in this work, the thinning technique is used to create the skeleton image. This technique based on the thinning process, it peels the object contour until the medial line with a one-pixel width, and the extracted skeleton retains the topology of the original form.
- **Minutiae extraction:** After obtaining the skeleton image, the objective principle is to detect the most interesting minutiae from this image. Rutovitz's approach [16] is implemented to detect minutiae and their end point or bifurcation types [17-18], as shown in Fig. 2(a) and Fig. 2(b), by calculating the Crossing-Number indicator (CN) according to the each pixel neighborhood, according to the CN value the point type is determined.
- The crossing-number of a pixel P is calculated by the following equation (1) [16]:
$$CN = \frac{1}{2} \sum_{i=1}^8 |P_i - P_{i+1}|$$

(1)
- $P_i$  is the value of the pixels in the 3 \* 3 neighborhood of P.
- $P_1, P_2, \dots, P_8$  are the 8 pixels in the neighborhood of P
- Indeed, the value CN allows us to identify the nature of a minutiae according to the result obtained during the computation of CN, as illustrated in the Table 1.

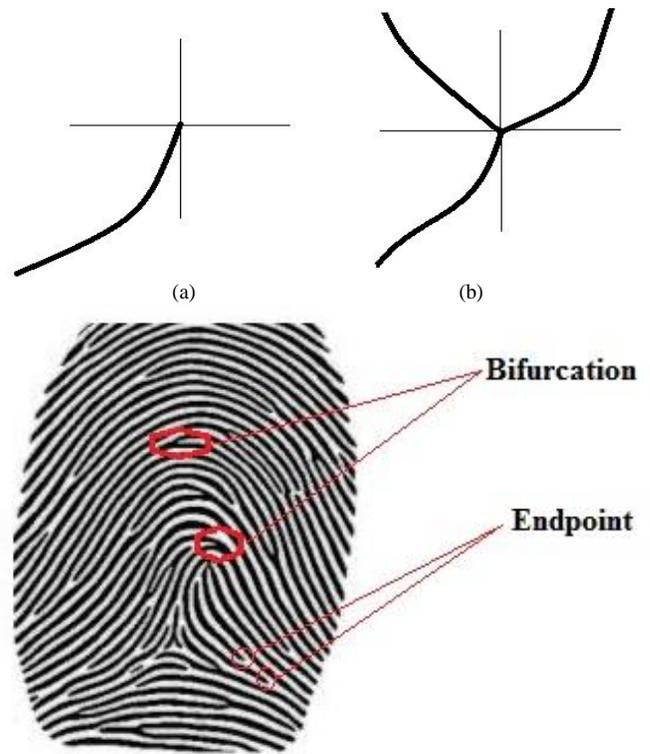


Fig. 2. Example of the Two Most used Minutiae, (a) end Point, (b) Bifurcation.

TABLE I. IDENTIFICATION OF THE MINUTIAE NATURE FROM THE CN VALUE

CN value	Nature of a minutia
0	Isolated point
1	Endpoint
2	Connective point
3	Bifurcation point
>=4	Crossing point

#### V. SIMPLIFICATION ALGORITHM

Initially, a line simplification algorithm is used to reduce the curve point number by a point's series, while preserving its shape. The points types selected should provide valuable indications for utilizing it later in the reconstruction of this curve. There are many line simplification algorithms such as the Area-Based Method, the Douglas-Peucker algorithm, Visvalingam Algorithm,... Depending on White study [22] of three simplification algorithms based on Marino work [23], he showed that line simplification produced by Douglas-Peucker they are the best lines examples with 86% of all tests. Because of these results, the Douglas-Peucker method represents the best of the three suggested methods. Also the McMaster [24] study which showed that the Douglas-Peucker algorithm was mathematically and perceptually superior to other algorithms, this algorithm detected more critical points and kept the original curve topology because it produces the least displacement compared to the original curve. The simplification of a 2D line using the Douglas - Peucker algorithm defined by this algorithm (1):

**Algorithm 1. Douglas-Peucker Algorithm**

```

INPUT → Curve C (an ordered set of points (P1, ..., Pn)) and the threshold value  $\epsilon$  ( $\epsilon > 0$ ).
OUTPUT → ResultPoint (control points)
function DouglasPeucker(C,  $\epsilon$ )
k → length (P1, ..., Pn)
Dmax → NULL // initialization of the maximum orthogonal distance
index → NULL
For i=2 to k-1 do
Dnew = orthogonal distance(P(1),P(i),P(n));
  if Dnew > Dmax
    index = i
    Dmax = Dnew
  End If
End For
if Dmax >  $\epsilon$ 
  C1=P(1) to P(index)
  C2= P(index) to Pn
  ResultPoint1=function DouglasPeucker(C1,  $\epsilon$ )
  ResultPoint2=function DouglasPeucker(C2,  $\epsilon$ )
  ResultPoint={ ResultPoint1, ResultPoint2}
else
  ResultPoint={P1,Pn}
End If

```

VI. FRACTAL INTERPOLATION

In life, they are many complex phenomena and natural forms that cannot be characterized in a classical way. All these complex phenomena may have specific characteristics. They can be complex in the sense of multiscale, scale invariance or self-similarity [30-32].

Mandelbrot gives mathematical models as well as the tools of analysis of these phenomena in his book "fractal geometry of nature" [32]. In addition, there are several studies that are interested in fractal study, such as Barnsley studies [30-31], which found that fractal phenomena can be obtained with deterministic models. And that they are based on sets of affine transformations, are called IFS (Iterated Function Systems) [30-31]. The iterative application of these transformations produces a form that has the property of being similar to it at several scales. Further details on this aspect will be given in the following section of the document.

A. Presentation

Consider the interpolation points:

$$P_i = \{(x_i, y_i) : i = 0, 1, \dots, N\} \in \mathfrak{R}^2 \text{ and } N \in \mathbb{N}$$

A geometric transformation  $W$  can be given by equation (2) the coordinates  $x'_i$  and  $y'_i$  of a point  $P'_i$ , image of  $P_i$  by  $W$ , according to the coordinates  $x_i$  and  $y_i$  of the point  $P_i$  [25-27].

$$W = \begin{cases} x'_i = f(x_i, y_i) \\ y'_i = g(x_i, y_i) \end{cases} \quad (2)$$

Where  $x'_i = f(x_i, y_i)$  and  $y'_i = g(x_i, y_i)$  are 2-variable functions  $x_i$  and  $y_i$

B. Affine Transformations

An affine transformation can scale, distort, rotate, and convert data differentially. a transformation can be described as a function that sends a point  $P_i(x_i, y_i)$  from the plane to another point  $P'_i(x'_i, y'_i)$ , let  $x'_i$  and  $y'_i$  the coordinates of a point  $P'_i$ , according to the coordinates  $x_i$  and  $y_i$  of  $P_i$  [27-29].

The transformation is called affine if it can be written in the following equation (3):

$$W_i = AP_i + T = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} * \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} \quad (3)$$

It is represented from the matrix  $A$  where the sub-matrix 2x2 denotes the combined effects of the transformations, rotation and scale change, while the vector  $T$  denotes the combined effects of the translation transformations.

It can be written as an equation system following:

$$W_i = \begin{cases} x'_i = a_i x_i + b_i y_i + e_i \\ y'_i = c_i x_i + d_i y_i + f_i \end{cases}$$

With a, b, c, d, e and f are real numbers.

Where  $x_i$  and  $y_i$  are the source control points coordinates and  $x'_i$  and  $y'_i$  the transformed coordinates [29].

- The IFS fundamental idea: The IFS fundamental idea: The IFS model is exclusively based on the self-similarity notion. The idea is to encode a fractal object (in a whole or in part with a similar structure) by a transformation set translating this property; it is this similarity, which allows utilizing the iterative method to generate a curve or an image. The transformations set will be called IFS and the corresponding fractal will be the attractor of the IFS.

The self-similarity notion is expressed by the equation (4):

$$K = \bigcup_{i=1}^N w_i(K) \quad (4)$$

In general, the fractal object K is equal to the transformation union of itself.

Let f be a continuous fractal function passing through a certain given points number of the form  $f(P_i, F_i) \in \mathfrak{R}^2, i = 0, \dots, N$  and with  $P_0 < P_1 < \dots < P_N$

In this study, we rely on the fractal interpolation principle studied by Barnsley [31-32]. An interpolation function corresponding to this data set declares above is a continuous function  $f [P_0, P_N] \rightarrow \mathfrak{R}$  passing through the interpolation points  $(P_i, F_i) \in \mathfrak{R}^2$  and checking  $f(P_i) = F_i$  with  $i = 0, 1, \dots, N$ . The function graph f (he is the attractors set) having N transformations of the form:

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = W_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} * \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}$$

The transformations  $W_i$  are defined with the five real numbers  $a_i, c_i, d_i, e_i$  and  $f_i$ .

The resolution of the preceding constraints makes it possible to define the transformations parameters  $W_i$  of the IFS

generating an attractor passing by the points set  $P_i$  with  $i = 0, 1, \dots, N$ . The transformations parameters are thus defined by the following equations (5) [29]:

$$\begin{aligned} a_i &= \frac{x_i - x_{i-1}}{x_N - x_1} \\ c_i &= \frac{y_i - y_{i-1}}{x_N - x_1} - d_i * \frac{y_N - y_1}{x_N - x_1} \\ e_i &= \frac{x_N x_{i-1} - x_1 x_i}{x_N - x_1} \\ f_i &= \frac{x_N y_{i-1} - x_1 y_i}{x_N - x_1} - d_i * \frac{x_N y_1 - x_1 y_N}{x_N - x_1} \end{aligned} \quad (5)$$

There remains therefore a freedom degree symbolized by the parameter  $d_i$  and representing a "vertical scale factor" or "contraction factor"[29]. Its value is independent of the interpolation points and controls the shape and roughness between the interpolation points of the graph. This parameter is calculated using the fractal dimension (DF) [33-34] and is expressed by the following equation (6):

$$d_i = (N-1)^{DF-2} \quad (6)$$

With the fractal dimension (DF) is calculated by the box-counting method [33-34].

### C. Implantation of Fractal Interpolation and IFS

Let us consider a points set  $P_i \{(x_i, y_i)\} i = 0, 1, \dots, N$  that we are trying to interpolate (the control points detected by the Douglas-Peucker algorithm). The IFS theory can be used to interpolate [29-30]. The  $N$  affine transformations are used for this purpose. This  $N$  transformations partition the interval  $[P_0, P_N]$  into  $[P_0, P_1] \cup [P_1, P_2] \cup \dots \cup [P_{N-1}, P_N]$ . We present in the remainder of the present section the fractal interpolation algorithm [28-29].

The implementation of this algorithm is a function called IFS and uses affine transformations in  $\mathfrak{R}^2$ , as you can see in algorithm (2).

#### Algorithm 2. IFS Algorithm

```

INPUT  $\longrightarrow P_1, P_0, N, K$ 
OUTPUT  $\longrightarrow P_{new}$ 
function IFS( $P_i, P_0, N, K$ )
   $h \longrightarrow \text{length}(P_i)$ 
  For  $i=1$  to  $h$  do
     $W_i \longleftarrow$  The parameters  $a_i, c_i, d_i, e_i, f_i$  are calculated using relations (5) - (6)
  End For
  For  $i=1$  to  $h$  do
     $P_{new}(i) \longleftarrow W_i(P_0)$ 
  End For
   $N \longleftarrow \text{length}(P_{new})$ 
  For  $j=1$  to  $K$  do
    For  $i=1$  to  $h$  do
       $N \longleftarrow N+i$ 
       $P_{new}(N) \longleftarrow W_i(P_{new})$ 
    End For
  End For

```

The input parameters are:

-  $P_i$ : The control points set.

-  $P_0$ : The initial points set (a matrix with the coordinates of the initial  $M$  points).

-  $K$ : integer type determines the iterations number.

The output is a point set  $P_{new}$ , in matrix form.

The algorithms begin by calculating the IFS map parameters  $W_i$  with using the data points set or control points  $P_i$ . Subsequently, we initialize the algorithm with a randomly selected points set. In the first step, apply an IFS map to these selected points; a new point set will emerge. At each step, repeat the same technique to the new points set obtained during the previous iteration until we obtain the original curve or image.

#### D. Evaluation of Performances

An interpolation function represent a approximating function, there is no interpolation type can be assumed as fully accurate. It is therefore meaningful to recognize to what extent the measured value may deviate from the initial value, the quantity variation, it is called error analysis [35-39]. To evaluate the interpolation accuracy as well as the efficiency of the fractal interpolation algorithms, There are two kinds of error computation, the first represent the relative error (RE) [35-37] and the second is the mean square error (MSE) [37-39].

Relative error (RE) used as an accuracy measure—represent the ratio of a measurement absolute error to the current measurement. Put differently, the RE (also called fractional error) is obtained by dividing the absolute error in the quantity by the quantity itself. RE is expressed as a percentage and he includes no units [35-37]. The relative error (RE) of the potentials is defined by the following equation (7):

$$RE = \frac{\Delta x}{x} \quad (7)$$

Where  $x$  is the true value of a quantity and  $\Delta x$  is the absolute error.

The mean squared error (MSE) or mean squared deviation (MSD) [37-38] is an estimator measures the errors mean squares, that is, the mean squared difference between the origin and what is interpolated [38-39]. It is described by as follows equation (8):

$$MSE = \frac{1}{n} \sum_{i=1}^n \left( P_i - \hat{P}_i \right)^2 \quad (8)$$

## VII. RESULT

To evaluate the proposed approach performance, this algorithm uses fingerprint images from the Fingerprint Verification Competition database (FVC2006)<sup>1</sup>. This database consists of four different sub-databases (DB1, DB2, DB3, and DB4), the first three acquired with different sensors and the last created with a synthetic generator; each fingerprint image was

captured at a 500 dpi resolution with various sizes. The size of each database to be used in the test was set to 110 fingers wide ( $w$ ) and eight impressions per finger depth ( $d$ ) (880 fingerprints in total). And they are devised in two sets:

Set A: consisting of fingers numbered from one to 100

Set B: made up of fingers numbered 101 to 110 and made available to users.

The results obtained were implemented using the software platform MATLAB (R2015a). The program was assessed on a personal computer running at 2.20 GHz for CPU and 4 GB of RAM.

#### A. Segmentation Steps

Initially, the fingerprint image is extracted from the database, the raw image or original image. The initially, the fingerprint image is extracted from the database, the raw image or initial image. Thereafter, the Gabor filter is used to enhance fingerprint images through eliminating its unnecessary noises. Then the binarization phase to transform the greyscale image into a binary image by the thresholding method. Subsequently, the binarized images are thinned implementing a thinning algorithm to reduce the lines width to one-pixel width of the skeleton. This thinning process does not change the original fingerprint topology that ensures efficient of the minutiae points extraction. Ultimately, scanning the fingerprint skeleton image allows detecting the pixels that correspond to minutiae by calculating the Crossing Number Indicator (CN).

Taking the example of the fingerprint image 101\_1.tif<sup>1</sup>, as you can see in Fig. 3(a) (raw image) taken from DB1 in FVC2006 to verify and applying the same segmentation steps, the results are presented in (Fig. 3), where Fig. 3(b) shows the enhanced version of the fingerprint image. Fig. 3(c) represents its binary image, Fig. 3(d) shows its thinned version, Fig. 3(e) and Fig. 3(f) represent the Endpoints and the bifurcation points, respectively.

The results of six fingerprint images are presented in the table below (Table 2). The first column is for illustration the minutiae number of type 1 (Endpoints) and column 2 for the minutiae number of type 2 (bifurcation). Column 3 and 4 shows the run time in second for each segmentation step.

Table 2 shows the results obtained during the segmentation steps. It should be noted that the execution time doesn't depend on the image size, but depends on another criterion, such as the image quality in the enhancement task, the execution time increases for low quality images. Second criterion, the end and bifurcations points number contained in the image. The average execution time for the enhancement task of the fingerprint images is between 4 and 6 second. On the other hand, the execution time for the minutia extraction is between 0.5 and 0.7 second. As you can see, the enhancement task is extremely costly in time terms. The run time obtained and that found by S. Sojan and R. K. Kulkarni [8] are almost identical (average execution times between 0.4 s and 0.6 s). The minutiae points extracted are visible in column 2 and 3. About 104 up to 173 endpoints and 20 up to 59 bifurcations were detected, as can be seen from Table 2.

<sup>1</sup>Fingerprint Verification Competition 2006 (FVC2006) web site (<http://bias.csr.unibo.it/fvc2006/>)



Fig. 3. (a) Original Fingerprint Image (Raw image) (b) The Enhanced Image (c) Binarized Image (d) Thinned Image (e) Endpoints (Red Points) (f) Bifurcations (Bleu Points).

TABLE II. MINUTIAE EXTRACTION AND THE RUN TIME

images	number of endpoint	number of bifurcation	execution time of enhancement task (Second)	execution time(Second)
101_1.tif	126	29	4.94	0.548
101_2.tif	104	32	5.15	0.577
110_4.tif	138	41	6.319	0.606
102_5.tif	174	28	4.12	0.702
108_3.tif	134	59	4.71	0.638
108_5.tif	173	20	5.79	0.6164

### B. Douglas-Peucker Algorithm

An approach to simplifying data in curves is to use a line simplification algorithm. One of the most used algorithms is the Douglas-Peucker algorithm. This algorithm allows data compression and reduces the data points number by eliminating redundant points, while retaining its shape, which will save storage space and transmission cost. Douglas-Peucker is a recursive algorithm based on perpendicular distance and a given tolerance value  $\epsilon$ .

The image pixel unit is used when it comes to the tolerance value. The figure below (Fig. 4) shows the simplified fingerprint curves with the Douglas-Peucker algorithm at a few different tolerance values between 0.5 to 2 Pixels.

As Fig. 4 shows, even with a low-precision simplification that results in a much smaller set of results, the overall curve shape remains the same. The first figure (Fig. 4(a)) shows a reduction for  $\epsilon = 0.5$  pixels of 75.37%, from 5113 to 1259 points. While, the twelfth figure (Fig. 4(c)) shows a reduction of 91.82% for  $\epsilon = 1$ , from 5113 to 418 points. Whereas, for  $\epsilon = 2$  the Fig. 4(e) shows a reduction of 94.15 %, from 5113 to 299 points.

Table 3 illustrates the Douglas-Peucker algorithm results on the test images set. This table describes for four different tolerance values (between 0.5 and 2 pixels) the reduced points number in the third column, the simplification rate in the next column and the last column, the run time.

The values in the Table 3 clearly show that the control points number decreases if the tolerance value  $\epsilon$  increases, and vice versa. As an example, for the first image test the control points number increases from 299 points (for  $\epsilon = 2$ ) to 1259 points (for  $\epsilon = 0.5$ ), and similarly for the simplification rate, he passes from 75% to 94 % for the same values of  $\epsilon$ . From the

obtained results, the simplification rate is between 75 and 95% according to the value of  $\epsilon$ . In addition, the execution time of the Douglas-Peucker algorithm is between 1s and 2.3s.

### C. Interpolation Fractal Tests

After implementing the Douglas-Peucker algorithm and the characteristic point's detection, the fractal interpolation algorithm already stated in section VI is used to reconstruct the fingerprint curve. The IFS technique result for different iterations number (between 50 and 500 iteration) is shown in Fig. 5. It can be noted that the reconstruction quality is affected according to the iterations number, for the narrow iterations number (between 50 and 300) the reconstruction quality is poor, as you can shows in Fig. 5(a) and Fig. 5(b). On the other hand, for an important iteration number (between 400 and 500), as shown in Fig. 5(c) and Fig. 5(d), the interpolated curve is identical to the original curve.

Fig. 5 shows examples obtained results at various iterations. At 50 iterations, the deformation is weak but visible. At 300 iterations, the final result is almost obtained. On the other hand, at the 500 iterations, the final result is perfectly obtained. The tests models details as well the numerical results are presented in the figure below.

Fig. 6 shows the relative error (RE) as a function of the iterations number. Indeed, there is an average relative error  $REM = 5.48\%$  at the iteration 50,  $REM = 3.165\%$  and  $REM = 2.13\%$  at the iteration 100 and 200, respectively. In addition, the average relative error is between 0.5 and 1.5% at iterations between 500 and 300, which indicates clearly that the interpolated curves and the original curves are practically identical and exceedingly close.

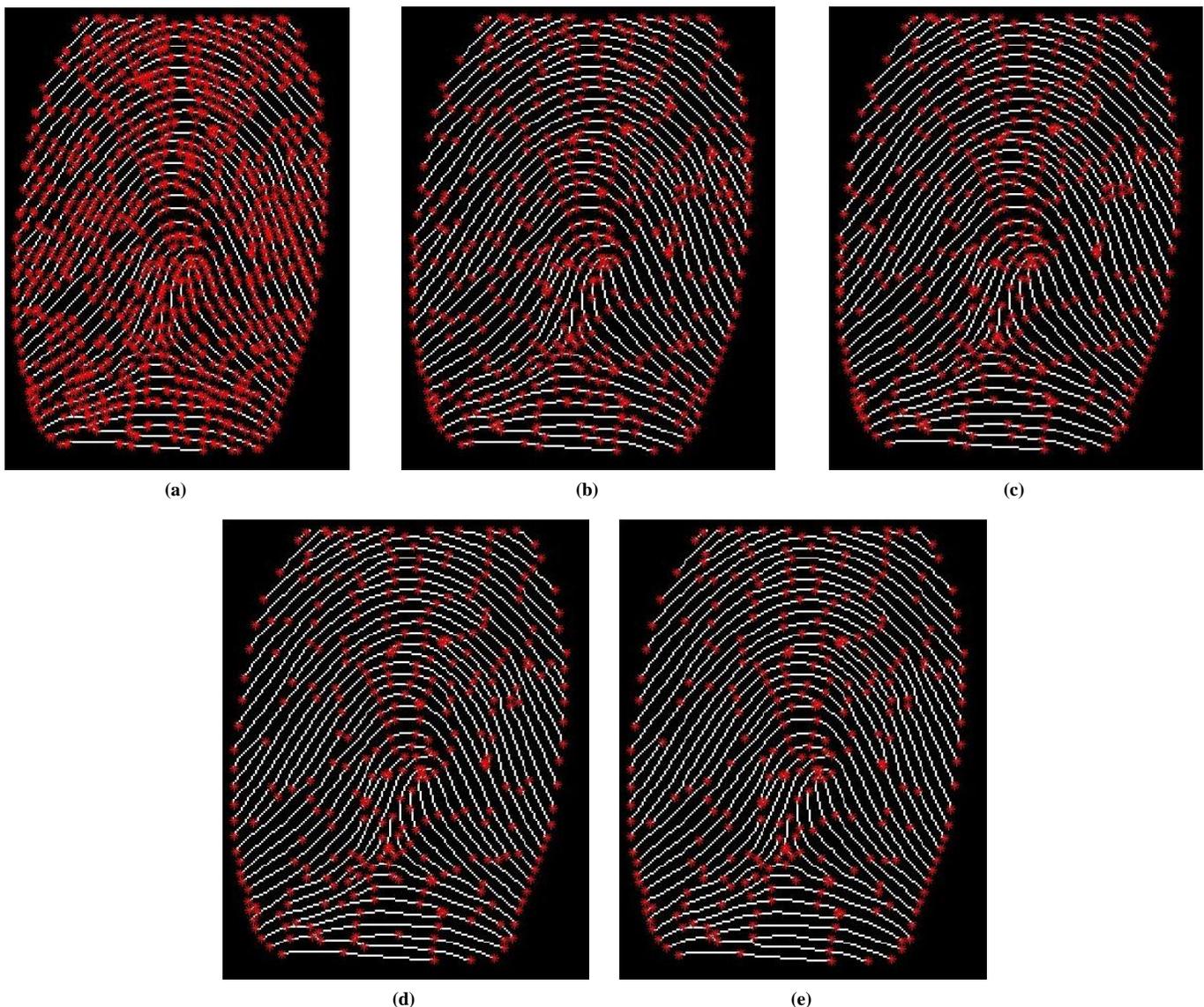


Fig. 4. Simplification Rate for different Tolerance Values  $\epsilon$ . (a)  $\epsilon=0.5$  Pixel (b)  $\epsilon=0.8$  Pixel, (c)  $\epsilon=1$  Pixel, (d)  $\epsilon=1.5$  Pixel, (e)  $\epsilon=2$  Pixel.

TABLE III. SIMPLIFICATION RATE FOR DIFFERENT TOLERANCE VALUES E AND RUN TIME

image	Data points number	$\epsilon$	Reduced points number	Simplification rate in%	Run time
101_1.tif		0.5	1259	75.3765	1.345205
		0.8	511	90.0059	1.231993
		1	418	91.8248	1.196033
		1.5	342	93.3112	1.176604
		2	299	94.1522	1.161165
101_2.tif		0.5	2251	76.4564	2.331704
		0.8	817	91.4549	2.040004
		1	634	93.3689	2.017986
		1.5	494	94.8332	1.933262
		2	415	95.6594	1.932721
110_4.tif		0.5	2278	75.5579	1.763994
		0.8	820	91.2017	1.439569
		1	654	92.9828	1.392574
		1.5	512	94.5064	1.361353
		2	441	95.2682	1.339857
102_5.tif		0.5	2383	74.1400	1.812687
		0.8	701	92.3928	1.542794
		1	552	94.0098	1.467193
		1.5	449	95.1275	1.536463
		2	386	95.8112	1.488915
108_3.tif		0.5	3014	77.0903	2.091803
		0.8	1111	91.5552	1.854681
		1	851	93.5315	1.698356
		1.5	697	94.7020	1.565364
		2	578	95.6066	1.506735
108_5.tif		0.5	2735	74.3457	1.804292
		0.8	804	92.4585	1.712217
		1	606	94.3157	1.651870
		1.5	491	95.3944	1.542717
		2	410	96.1542	1.444087

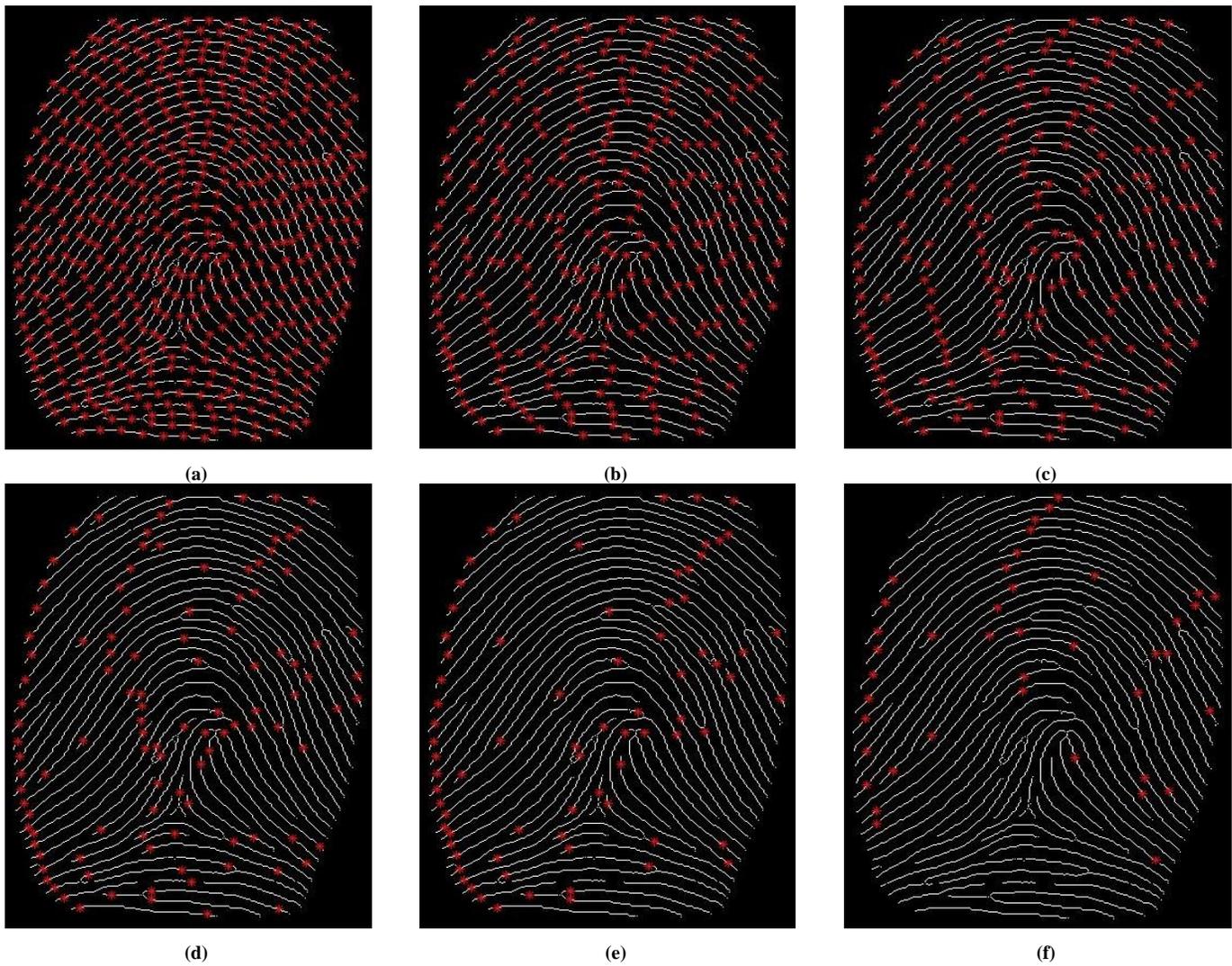


Fig. 5. Fractal Reconstruction for different Iterations Values (White Points: Bullet Points, Red Points: Erroneous Points), (a) Iterations Number= 50, (b) Iterations Number= 100, (c) Iterations Number= 200, (d) Iterations Number= 300, (e) Iterations Number= 400, (f) Iterations Number= 500.

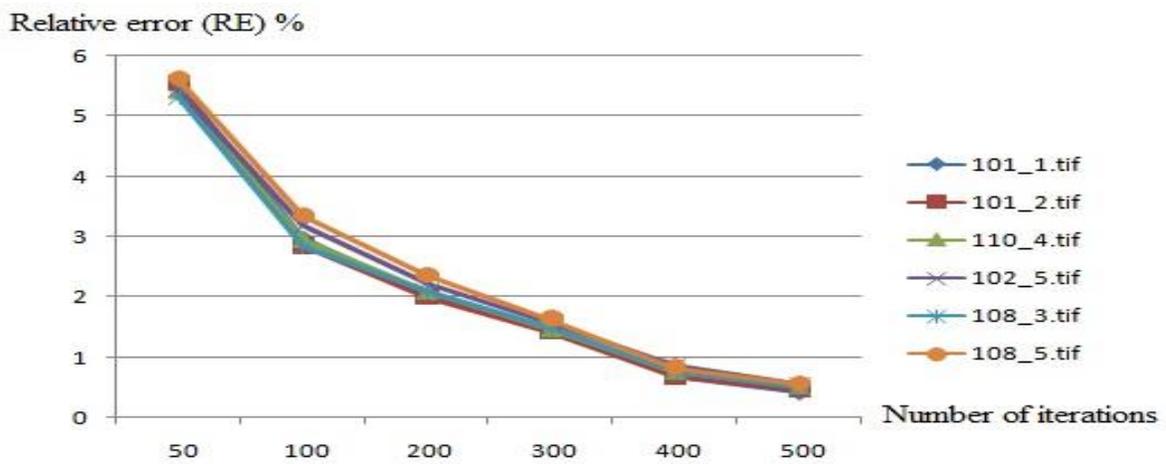


Fig. 6. The Relative Error Curve.

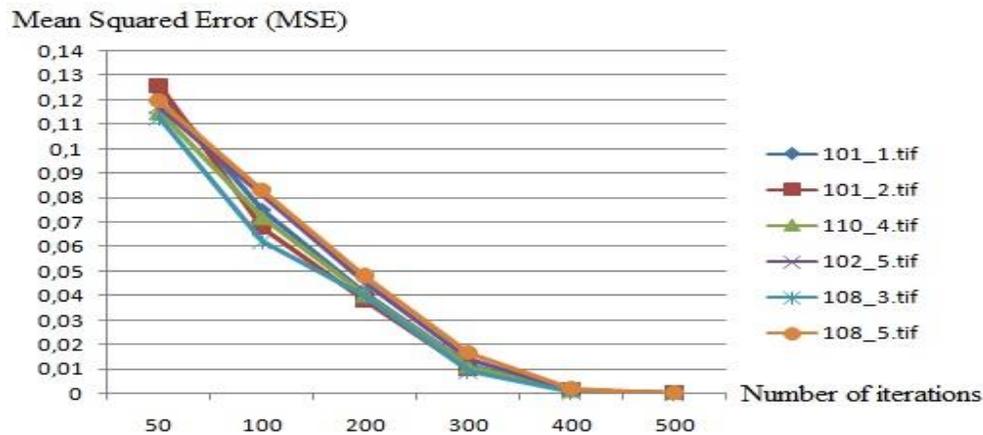


Fig. 7. The Mean Squared Error Curve.

Fig. 7 represents the Mean Squared Error MSE measure that reflects the distance between the original curves points and corresponding interpolated curves points. According to Fig. 7. above, it can be noted that when the iterations numbers are great (between 300 and 500), the mean squared error is small (between 0.000124 and 0.012); on the other hand, when the iterations numbers are small (between 50 and inferior to 300), the mean squared error is relatively large (between 0.041 and 0.126), which corresponds exactly to the same remark in the previous section as the interpolated curves and the original curves are virtually identical and exceedingly close for the high iterations numbers.

#### VIII. DISCUSSION

The results concerning the curves approximation with fractal interpolation is presented in this section. Validation tests were performed on real fingerprint image. With respect to the curvature error evolution using fractal interpolation, the obtained results show that the relative error (ER) and the mean squared error (MSE) are large when the iterations number is small (between 2.007% and 5.627% for RE and 0.126 and 0.009 for MSE). On the other hand, these errors (between 0.415% and 1.64% for RE and 0.000124 and 0.0167 for MSE) are weak when the iterations number is considerable. The method proposed by Jian-Kai et al. [40] for the average relative error between predicted and genuine values was 1.32%. For the method proposed by MACÉNAITÉ et al [41], he calculated two errors types (Mean squared errors (MSE) and relative errors (RE)) between the autocorrelation functions of the real profilogram and its model. They obtained a relative error between 8.068% and 20.382%, mean squared errors (MSE) between 0.134 and 0.339. The method proposed by Guedri et al. [28] the relative errors between the ideal centre line and the simulated blood vessels centre line ranged from 1.341 to 12.608 %.

The results obtained and the comparison shows that the proposed method has succeeded in reconstructing very precisely the fingerprint curves; this study shows that the fingerprint curves can appropriately preserve its particular structure when the iterations numbers exceed 300 iterations.

The advantage of such an approach lies in the high simplification rate (can reach up to 96%), which makes it

possible to reduce the memory size to storing the images of the fingerprint, which allows reduce the costs and the transmission time. In addition, the fractal interpolation can reconstruct the data points with higher resolution than initially model and the reconstructed model has more natural and real details. On the other hand, the proposed algorithm vulnerable point lies in the calculation complexity. This requires a specialized computer or calculator to minimize the run time.

#### IX. CONCLUSION

This paper explains how to used fractal theory to data compression and interpolation of fingerprint curves. This method is divided into three essential parts; the first part is the image segmentation of the fingerprint (enhancement, image binarization, skeletonization and minutiae extraction). Subsequently, the second part integrates the Douglas-Peucker linear simplification method to reduce the images size. In closing, the third part is interested in the fingerprint curves reconstruction by fractal interpolation. This Research has shown that the data reduction rate of the fingerprint curves can reach 96%, it can be noted that the Douglas-Peucker algorithm has a high reduction rate and also a rapid reduction to 2 seconds, moreover, they have retains detailed characteristics for the fingerprint curves. Further, the general structure of reconstructed curves by fractal interpolation can obtain an excellent quality, validation tests were performed. The research shows that the error obtained between 0.415% and 1.64% for RE and between 0.000124 and 0.0167 for MSE with the iterations number larger than 300. This clearly indicates that the interpolated curves and the original curves are virtually identical and extremely close.

In the future work we will implement this method proposed within hardware platform for example on embedded architectures and reconfigurable such as FPGA programmable elements of the type "soft-core" (heart CPU generalist or core DSP).

#### REFERENCES

- [1] N.Soundharadevi and M.Pushparani, "Analysing on multimodal biometric frame work with face, iris and fingerprint images," Shanlax International Journal of Arts, Science & Humanities, Vol. 4 (1), pp.165-172, September 2016.
- [2] V. Conti, C. Militello and S. Vitabile, "Biometric authentication overview: a fingerprint recognition sensor description," Int J Biosen

- Bioelectron, vol. 2(1), pp. 26–31, 2017. DOI: 10.15406/ijbsbe.2017.02.00011.
- [3] A. Mira Saleh, A. Bahaa and A. Wahdan, “Fingerprint Recognition,” *Advanced Biometric Technologies*, InTech, 2011 pp.201- 224, DOI: 10.5772/23476
- [4] L. Hong, Y. Wan, and A. K. Jain, “Fingerprint image enhancement: Algorithm and performance evaluation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 20(8), pp 777-789, 1998.
- [5] A. Nagar, S. Rane and A. Vetro, “Alignment and Bit Extraction for Secure Fingerprint Biometrics,” *Conference: Media Forensics and Security II*, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 18-20, 2010. DOI: 10.1117/12.839130
- [6] T. Vidhya and T. K. Thivakaran, “Fingerprint Image Enhancement Using Wavelet Over Gabor Filters,” *Int. J. Computer Technology & Applications(IJCTA)*, Vol. 3 (3),pp 1049-1054, 2012.
- [7] S. Mohammedsayeemuddin, S. K. Gonsai, and D. Vandra, “Efficient Fingerprint Image Enhancement Algorithm Based On Gabor Filter,” *International Journal of Research in Engineering and Technology*, Vol. 3(4),pp. 809-813, Apr 2014.
- [8] S Sojan and R. K. Kulkarni, “Fingerprint Image Enhancement and Extraction of Minutiae and Orientation,” *International Journal of Computer Applications*, vol 145(4),pp 14-19, 2016.
- [9] W. Hussain, T. Munawar, M. Shahzaib, M. Masood, “Automated Enhancement of Compromised Fingerprint Images,” *Journal of Biochemistry, Biotechnology and Biomaterials ( JBCBB)*, Vol. 1(2), pp. 27-33, 2016.
- [10] A. Vij and A. Namboodiri, “ Learning Minutiae Neighborhoods: A New Binary Representation for Matching Fingerprints,” *IEEE Conference on Computer Vision and Pattern Recognition Workshops 23-28 June 2014*, Columbus, OH, USA DOI: 10.1109/CVPRW.2014.15
- [11] G. A. Bahgat, A. A. Hefnawy, A. H. Khalil, N. S. A. Kader and S. Mashali, “Developed Fingerprint Segmentation Technique based on Mean and Variance Intensity Thresholding,” *Proceedings of the 4th IIAE International Conference on Industrial Application Engineering 2016*, pp. 250- 257. DOI:10.12792/iciae2016.047.
- [12] W. Abu-Ain, S. N. H. S. Abdullah, B. Bataineh, T. Abu-Ain and K. Omar, “Skeletonization Algorithm for Binary Images,” *Procedia Technology*, vol. 11, pp. 704–709, 2013. doi: 10.1016/j.protcy.2013.12.248.
- [13] M. S. Al-Ani, “A Novel Thinning Algorithm for Fingerprint Recognition,” *International Journal of Engineering Sciences*, Vol. 2(2), pp. 43-48, 2013.
- [14] R. F. L. Carneiro and J. A. Bessa, “Techniques of Binarization, Thinning and Feature Extraction Applied to a Fingerprint System”, *International Journal of Computer Applications*, Vol. 103(10), pp. 1-8, 2014.
- [15] V. Jain, A. K. Singh, “An approach for Minutia Extraction in Latent Fingerprint Matching,” *International Journal of Innovations in Engineering and Technology (IJET)*, Vol. 6(1), pp. 51-58, 2015.
- [16] D. Rutovitz, “Pattern recognition,” *Journal of the Royal Statistical Society*, Vol. 129 (4), pp. 504–530, 1966.
- [17] Atul S. Chaudhari, G. K. Patnaik, S. S. Patil, “Implementation of Minutiae Based Fingerprint Identification System using Crossing Number Concept,” *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 8(4), pp. 178-183, 2014.
- [18] I. K. Virdaus, A. Mallak, S.-W. Lee, G. Ha, and M. Kang, “Fingerprint Verification with Crossing Number Extraction and Orientation-Based Matching,” *Proceedings of The International Conference on Next Generation Computing*, Bangkok, Thailand, February 2016, pp. 113-115.
- [19] D. H. Douglas and T. K. Peucker “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *The Canadian Cartographer*, vol. 10 (2), pp. 112-122, 1973.
- [20] Z. Shukai, L. Zhengjiang, Z. Xianku, S. Guoyou and C. Yao, “A method for AIS track data compression based on Douglas-Peucker algorithm,” *Journal of Harbin Engineering University* , 5, pp. 595-599, 2015.
- [21] T. Tienaaah, E. Stefanakis, D. Coleman, “Contextual Douglas-Peucker Simplification,” *Geomatica*, Vol. 69(3), pp. 327-338, 2015. DOI: 10.5623/cig2015-306
- [22] E.R. White, “Assessment of line-generalization algorithms using characteristic points,” *Am. Cartogr*, vol. 12, pp. 17–28, 1985.
- [23] J.S. Marino, “Identification of characteristic points along naturally occurring lines,” *Cartogr. Int. J. Geogr. Inf. Geovisualization*, vol. 16, pp. 70–80, 1979.
- [24] R. B. McMaster, “Automated Line Generalisation”, *Cartographica*, vol. 24 (2) , pp.74 -111, 1987.
- [25] S. Ri, “A new idea to construct the fractal interpolation function,” *Indagationes Mathematicae*, Vol. 29(3), pp. 962-971, 2018. DOI: <https://doi.org/10.1016/j.indag.2018.03.001>
- [26] J. Tang, Z. Feng, Y. Guo, “The Structure of Fractal Interpolation Curve in Plane,” *International Journal of Nonlinear Science*, Vol. 26(1),pp.34-40, 2018.
- [27] S. Ri, “A new nonlinear fractal interpolation function,” *Fractals*, Vol. 25(6), pp. 1750063-1- 1750063-12, 2017 DOI: 10.1142/S0218348X17500633
- [28] H. GUEDRI, J. MALEK and H. Belmabrouk, “Reconstruction of the human retinal blood vessels by fractal interpolation,” *Journal of Theoretical and Applied Information Technology (JATIT)*, Vol. 83(2), pp. 227-233, 2016.
- [29] C. Chen, T. Lee, Y. M. Huang and F. Lai, “Extraction of Characteristic Points and Its Fractal Reconstruction for Terrain Profile Data,” *Chaos, Solitons Fractals Extraction of Characteristic Points and Its Fractal Reconstruction for Terrain Profile Data*, vol. 39(4), pp. 1732-1743, 2009.
- [30] M.F. Barnsley, “Fractal Functions and Interpolation,” *Constructive Approximation* , vol. 2(1), pp. 303–332, 1986.
- [31] M.F. Barnsley, *Fractals Everywhere*, 2nd edn. Academic Press, Inc., Boston , 1993.
- [32] B. Mandelbrot, *The Fractal Geometry of Nature*, Freeman, San Francisco, 1982.
- [33] S. B. Sujata, V. Khushbu, M. P. Jaymin and J. B. Subhash, “Finger Print Fractal Dimension as a Supplementary Quantitative Measure Distinguishing Fingerprints and Gender,” *Biostat Biometrics Open Acc J*. vol.2(3), pp. 555-587, 2017. DOI: 10.19080/BBOAJ.2017.02.555587.
- [34] C. Sahu, V. Jain, “A Novel Approach to Fractal Dimension based Fingerprint Recognition System,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 3(4), pp.67-71 ,2016.
- [35] K. Chen, S. Guo, Y. Lin and Z. Ying, “Least Absolute Relative Error Estimation,” *Journal of the American Statistical Association*, vol. 105:491, pp. 1104-1112, 2010.
- [36] W. Ford, “Floating Point Arithmetic,” *Numerical Linear Algebra with Applications Using MATLAB*, pp. 145-162, Elsevier 2015.
- [37] C. Chen, J. Twycross and J. M. Garibaldi, “A new accuracy measure based on bounded relative error for time series forecasting,” *PLoS ONE*, vol. 12(3): e0174202, 2017. Doi:<https://doi.org/10.1371/journal.pone.0174202>
- [38] J. Rougier, “Ensemble averaging and mean squared error,” *Journal of Climate*, vol. 29(4), pp. 8865-8870, 2016. <https://doi.org/10.1175/JCLI-D-16-0012.1>
- [39] E. Holst and P. Thyregod, “A statistical test for the mean squared error,” *Journal of Statistical Computation and Simulation*, vol. 63(4), pp. 321-347, 1999. DOI: 10.1080/00949659908811960
- [40] J-K. Liang, C. Cattani and W-Q. Song, “Power Load Prediction Based on Fractal Theory,” *Advances in Mathematical Physics*, vol. 2015, Article ID 827238, 6 pages. <http://dx.doi.org/10.1155/2015/827238>.
- [41] L. MAČĚNAITĚ , M. LANDAUSKAS and V. P. PEKARSKAS, “Surface Roughness Simulation Using Fractal Interpolation to the Profilogram,” *MATERIALS SCIENCE (MEDŽIAGOTYRA)*. Vol. 18(2), pp. 138-144, 2012. <http://dx.doi.org/10.5755/j01.ms.18.2.1916>.