FINGERPRINT IDENTIFICATION USING THE FINITE ELEMENT TECHNIQUE

Magdy Saeb, Sherif Hegazy
Arab Academy for Science, Technology & Maritime Transport
School of Engineering, Computer Department
Alexandria, EGYPT

ABSTRACT

In this work, we present an approach for fingerprint identification based on the Finite Element Technique (FIFE). The basic notion behind this approach is to generate a feature matrix of the fingerprint employing the standard Galerkin weighted residual method. The approach is based on judiciously discretizing the fingerprint image using first-order triangular elements. The resulting matrix holds the space information regarding the mesh vertices, which, in turn, are relevant to the fingerprint shape features. This matrix is then used to build a content-based image retrieval database system. It is well known that this method is a generalization of Fourier Transform. However, the Galerkin method provides significant saving in memory storage requirements and execution times due to the banded nature of the resulting matrix.
Testing, using various alterations of fingerprints, has shown that the method is successful in general, even in cases of deformed fingerprints, with appreciably rapid search times.

KEYWORDS:
Fingerprint, Finite Element, Image Retrieval, Content-based, Database.

1. INTRODUCTION

In a previous work [1], Saeb and Hegazy have developed a method for shape recognition using the finite element technique. In this work, we employ the same method to identify fingerprints. Some modifications had to be performed on the method in order to apply it on fingerprint identification. The fundamental idea in any fingerprint identifier is to extract some numerical descriptor of the fingerprint. This descriptor is then used soon after for comparison in the identification process. This, in turn, necessitates that the descriptor would be simple enough to be used in millions of comparisons efficiently. The descriptor also has to describe each fingerprint uniquely according to its features.
In the subsequent three sections, we provide a background on fingerprint identification, a summary of the algorithm and the details of the implementation. The results are presented in section four. Section 5 concludes our work along with the promising future extensions.

2. BACKGROUND

The problem of fingerprint identification has drawn a lot of attention due to its critical importance in security work. The problem mainly includes two main phases:
1. Extracting the correct features that uniquely describe each fingerprint,
2. These features are subsequently used to construct a numerical descriptor of the fingerprint.

The feature extraction problem has been solved by physicians efficiently over a century ago. Each fingerprint contains two types of features depending on the ridges of the print itself [2], as shown in Figures 1,2. These are:
1. Ridge endings,
2. Ridge bifurcations.

These two features completely describe the fingerprint. The features are extracted using several existing image processing techniques as outlined in [2]. Eventually the fingerprint is represented by a set of points as demonstrated in Figure 3.

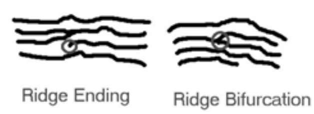


Figure 1: Features of a Fingerprint

The main idea in most of these edge detection algorithms is to use the eight-rule. That is scanning the image point by point and applying the eight-mask to find the positions of Ridges and Bifurcations. This of course requires a preprocessing phase of the image itself to reduce the noise and obtain clear contours of the fingerprint. It also ensures that the contours have similar width, e.g. one pixel, which is required to apply the 8-mask, as shown in Figure 2.
The second task is to use this set of points to generate a descriptor that describes this fingerprint. Several techniques exist for this undertaking, e.g. Fourier Transform descriptor [3]. In this work, present a new algorithm for generating the descriptor.

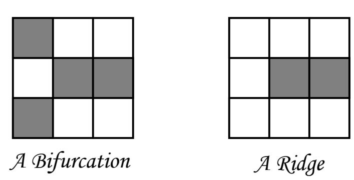


Figure 2: Using the eight-mask to detect features of a fingerprint, (The shaded cells represent black pixels).

Figure 3: The Fingerprint after extracting the features. (The features are indicated in red)

3. THE FINGERPRINT IDENTIFICATION USING FINITE ELEMENT TECHNIQUE (FIFE)

The proposed approach uses the finger print features to construct the finite element mesh [1]. The algorithm consists mainly of two phases: the mesh construction phase and the descriptor calculation phase.

In the mesh construction phase, each feature is represented by a point. The x and y coordinates of this point are then recorded. The set of these target points are used to construct an admissible finite element mesh. To ensure the admissibility of the mesh; we have used the Delaunay method when we form the triangles from the given set of points, i.e. the feature points. No more points are inserted to refine the mesh. The outcome of this phase is a set of triangular elements represented by their vertices, as shown in Figure 4. We have chosen the triangular element shape because it is effortless in the case of constructing the mesh on a predefined set of points. Triangular elements are easily fit on the existing points rather than square or higher-order elements.

Figure 4: The Mesh constructed using the features' points on the fingerprint

In the second phase, we construct the Galerkin matrix as in [1] and set the color property of the shape to one (we don't use colors in fingerprints, so we assume they are all the same color). This matrix is based upon standard Galerkin weighted residual approach [4]. The approach was used mainly for approximating the solution of partial differential equations. That was used to solve boundary value problems whose solution eventually describes the boundary Ω of the problem [5]. Given a system of partial differential equations of the form [4]:

$$L(u0)=f \quad \text{in } \Omega \qquad (1)$$

With boundary conditions:

$$G(u0)=p \qquad \text{on } \Gamma 1$$
$$H(u0)=q \qquad \text{on } \Gamma 2$$

Γ1 ,Γ2 represent boundary portions of the domain Ω, and Γ is the overall boundary, given by:

$$\Gamma 1 + \Gamma 2 = \Gamma$$

Assume u is an approximate solution of the boundary value problem. This approximation produces an error or residual (έ) in equation (1), such that

$$\acute{\varepsilon} = L(u) - f \neq 0 \qquad (2)$$

To minimize the residual έ, we orthogonalize this error with respect to a weighting function ψ over certain locations in Ω. This orthogonalization process is described by the following integral:

$$\int \acute{\varepsilon}\, \psi_i \, d\Omega = 0 \qquad i = 1,2\ldots n \qquad (3)$$

In the standard Galerkin method, the weighting functions ψi are chosen to be identical to the basis functions φi used to approximate the solution u in the set of sub domains or finite elements. This yields:

$$\int_{\Omega} [L(\sum_{i=1}^{n} \alpha_i \varphi_i) - f]\, \varphi_i \, d\Omega = 0 \qquad (4)$$

The method was demonstrated by Saeb [4] to describe field problems with anisotropic surface properties. The global reluctivity matrix R had n elements on the form:

$$r_{ij} = \int_{\Omega} [\upsilon_y \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \upsilon_x \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y}] d\Omega \qquad (5)$$

where:

$$N_i = [(a_i + b_i x + c_i y)/2s(e)]$$

and:

$$N_j = [(a_j + b_j x + c_j y)/2s(e)]$$

The elements are now of the form:

$$r_{ij} = \int_{s(e)} \frac{[\upsilon_y b_i b_j + \upsilon_x c_i c_j] ds}{(2s(e))^2} \qquad (6)$$

Thus, the resultant matrix of each element is of the form

$$r(e) = \frac{1}{4s(e)} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \qquad (7)$$

Where rij is given by:

$$rij = \upsilon_y \; bi \; bj + \upsilon_x \; c_i \; c_j \qquad (8)$$

This elemental matrix r(e) actually describes the field distribution shape of an element using the coordinates of the triangle representing it. The matrix is symmetric and υy and υx represent the surface properties. These properties we have set to one in our first form of the algorithm. It is possible though to construct a more selective descriptor, by setting the values of υy and υx to indicate the positions of the features of the fingerprint (ridges and bifurcations). This leads to a more precise description of the fingerprint by discriminating between the two different types of features when constructing the descriptor. This is achieved as follows: let R be the number of ridges and B the number of bifurcations in a given element of the mesh. Now we set $\upsilon y = \upsilon x = \upsilon$ where υ is given by the following table:

Table 1: The value of υ according to the number of Ridges (R) and Bifurcations (B).

B \ R	0	1	2	3
0	--	--	--	4
1	--	--	3	--
2	--	2	--	--
3	1	--	--	--

In other words, we can describe υ using the following formula:

$$\upsilon = R + 1$$

or

$$\upsilon = 4 - B$$

Thus, if two fingerprints have identical meshes but with different locations of bifurcations and ridges on each mesh, the algorithm will be able to distinguish them from each other. This is accurate since they are biologically very different. This could be an optional parameter to be used to perform more filtering of the results of the original algorithm in case of large databases (1-n matches), or it could be used directly to obtain exact matches (1-1 matches).

The generalized transformed co-ordinates bi and ci are given by:

$$b_1 = y_2 - y_3 \quad , \quad c_1 = x_3 - x_2$$
$$b_2 = y_3 - y_1 \quad , \quad c_2 = x_1 - x_3$$
$$b_3 = y_1 - y_2 \quad , \quad c_3 = x_2 - x_1,$$

The area of the each element s (e) is given by:

$$s(e) = 0.50 \begin{vmatrix} 1 & x1 & y1 \\ 1 & x2 & y2 \\ 1 & x3 & y3 \end{vmatrix} \qquad (9)$$

The matrix, given by equation 7, contains embedded information about the shape of the element using the generalized coordinates, as mentioned before. By assembling all the elements, the resulting global matrix G, takes the form:

$$G = \begin{pmatrix} R(1) & 0 & 0 & 0 \\ 0 & R(2) & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & R(n) \end{pmatrix} \qquad (10)$$

The global matrix G is an n-by-n nonsingular matrix, where n is the total number of elements in the mesh. This global matrix is the numerical descriptor that we have applied to describe the fingerprint features. The Galerkin global matrix implicitly holds the information regarding the relative positions of the vertices of the triangular elements, which in turn represents the fingerprint features. The elemental matrix r (e), given by equation (7), actually describes the shape based on a set of difference equations that are independent of the coordinate system rotation or translation processes.

Each fingerprint in the fingerprint database has a unique descriptor that is constructed when it is included in the database. This is called the feature extraction phase. This descriptor is then used in the comparison process or the query phase. If two fingerprints are similar; then the distance between their describing descriptor matrices will be small as compared with other distances. Thus, the process of the comparison or search of fingerprints leads eventually to a simple matrix subtraction operation. Base on extensive testing epochs, one concludes that the descriptor is both unique and straightforward. The original fingerprint can also be compressed, and the search can still be done on the descriptor, which is not influenced by the compression process at all. The search process is found to be exceptionally fast and efficient at the same time. The storage requirements are minimal since the resulting matrix is, it is well known, is both banded and symmetric. The case of fingerprints with deformations, due to inaccurate scanning or image distortion, can also be detected accurately. The absence of one or more features in the fingerprint still keeps the deformed fingerprint on the top of the list of least distances. This arises from the fact that the whole shape of the fingerprint is nonetheless similar to the original one. This is a property in the finite element method itself [1]. The details involved in the calculation of the descriptor are described in the next section.

4. ANALYSIS OF FPIFE

We now demonstrate the efficiency of the algorithm. The overhead involved in the processing can be divided into two parts:

1. Calculation of the numerical descriptor.
2. Comparison and query.

The first computational overhead is present only in the process of adding a new fingerprint to the database, i.e. only once in the lifetime of a fingerprint. This is called, as mentioned before, the feature extraction phase. This phase only consists of simple multiplications and additions to compute the Galerkin matrix [1] See section III. The second computational overhead is present at each comparison operation of two fingerprints in the search process. This could be carried out millions of times according to the size of the database. The comparison operation is a subtraction of two matrices in order to find the distance. We conducted several tests using several distance measures; the best results were obtained using the "Absolute Distance Measure." Given two matrices, A and B, the absolute distance between the two matrices is given by:

$$d = \sum_{i=1}^{n} |\, A(i) - B(i)\,| \qquad (11)$$

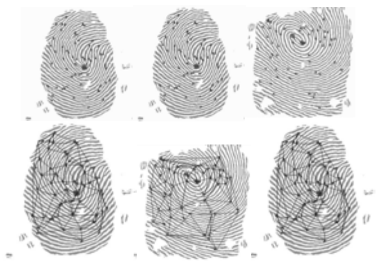
This means that, if we have N elements in the mesh of a fingerprint, we will have a 3N by 3N matrix that describes this fingerprint [1], [2]. This means a complexity of O (N^2) for each comparison process. Since the Galerkin matrix is both symmetric and banded [1] [5], the overhead falls to 12N, [1]. See section III. This is of order (N), which is quite acceptable. The normalized version of equation (7) is given by:

$$d_{normalized} = \sum_{i=1}^{n} | A(i) - B(i) | / \max \ d_j \quad (12)$$

$\forall j \in M$, where M is the number of fingerprint samples.

5. RESULTS

We have developed the FE Recognizer [1], using MS Windows as a platform, to test the proposed method. The obtained test results show consistent outcomes. A test sample of the results is shown in Figure 5 and Table 2.



(1) (2) (3)

Figure 5: Three fingerprint samples: 1 and 2 are closer than 3 (FE Meshes are shown)

As shown in Figure 5, we compare three sample fingerprints: the first two are for the same person with some deformation in one of them. The third fingerprint is for a different person, as shown in Figure 5. As shown in Table 1, the distance between the first and the second is less than the distance between the third and the rest, as shown in figure 5. This complies with the shape of the fingerprints.

Table 2:Comparison results (1 and 2 are similar and the closest in the set)

Sample Number	1	2	3
1	0	**414.568**	496.648
2	**414.568**	0	428.058
3	496.648	428.058	0

Table 3: Comparison based on normalized results (1 and 2 are similar and the closest in the set)

Fingerprint Sample Number	1	2	3
1	0	**83.5**	100.0
2	**83.5**	0	86.2
3	100.0	86.2	0

6. CONCLUSION AND FUTURE WORK

The proposed algorithm FIFE employs the Finite Element method to construct a numerical descriptor that articulates each fingerprint uniquely. The approach is exceptionally efficient in its time complexity since it consists mainly of simple arithmetic operations. The search or comparison procedure involves only addition and subtraction operations, which provides relatively small execution times. On the other hand, the proposed method provides remarkably low storage requirements since the resulting matrix is both banded and symmetric. In other words, the matrix can be stored as a single-dimension vector.

The discretized matrix is thought of as a transformation matrix that stores spatial relations regarding the finger print shape. This matrix, based on a set of difference equations, is both independent of the rotation and translation of the coordinate axes.

Various matrix distance norms were considered. However, the absolute distance that has been adopted in this work was found to be exceptionally consistent in identifying similar cases.

Future work includes a hardware realization of the algorithm, which is a straightforward process using VHDL due to the simplicity of the method.

REFERENCES

[1] S. HEGAZY, MS. Thesis, Arab Academy for Science, Technology, & Maritime Transport, Computer Engineering Dept., "Shape Identification using the Finite Element Method,"2002.

[2] Neal R. Wagner " Fingerprinting," 1983 IEEE. Symposium on Security and Privacy, Oakland, California, pp.18-22, April 25-27 1983.

[3]A. Jain, L. Hong, R. Bolle, "on-line fingerprint verification," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 4, 1997.

[4] Magdy Saeb, Ph.D. Dissertation, The Finite Element Analysis of Devices with Anisotropic Materials, University of California, Irvine, 1985.

[5] O. Axelsson and V. A. Barker, Finite Element Solution of Boundary Value Problems: Theory and Computation, Academic Press Inc., 1984.