Visual Information Retrieval
Employing the Galerkin Finite Element Technique
(VIRGFET)

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

Abstract

In this work, we present our approach for indexing images in multimedia databases, in order to facilitate the visual information retrieval process. We propose a new shape descriptor that can incorporate color information as well. The descriptor is based on the Galerkin Weighted-residual Finite Element Technique. It can be easily extended to the three-dimensional case, in case of describing three-dimensional images (objects). The technique provides appreciable savings in storage and memory, which allows the search in highly compressed image. Testing in several applications and types of image databases has shown encouraging results for this type of shape descriptor.

Keywords:
Multimedia Databases, Finite Element Method, Galerkin's Method, Compressed images, Content-Based Image Retrieval, Shape description, image indexing.

1. Introduction

Recently, substantial advancements in the field of multimedia databases have taken place [1,2]. The need to search and organize this form of databases has become quite necessary in many fields of research. It became necessary to be able to search huge amounts of multimedia data, more efficiently and intelligently. The user demands are now beyond simple queries, they are requesting more complex queries to be carried out on the multimedia database. Indexing in multimedia databases has become essential to aid achieving this particular goal [3]. With a good indexing technique, users will be able to obtain the desired results accurately and efficiently. In order to be able to deal with the contents of images, several Content-Based Image Retrieval (CBIR) techniques were developed as shown in [4]. In this work, we present a novel shape recognition technique that is based on the Galerkin Finite Element Technique.

The paper is organized as follows: Background and mathematical basis then our methodology and its complexity analysis and finally the results and conclusion.

2. Background

In order to be able to retrieve images containing certain shapes, these shapes need to be indexed or described using a certain descriptor [5,6]. This is usually a matrix or a vector that is constructed based on the features of the shape [7,8]. Users enter query image or sketch, and the system returns similar images by comparing the descriptors in both the database and the query image. This comparison is performed using some similarity measures [9].

3. Mathematical Foundation

In this section we illustrate, in brief, the standard Galerkin method [10], which is the foundation of our work. Galerkin's formulation is the generalization of Fourier Transform with more added advanced features. This is our motivation of using such a method as a basis for formulating a shape feature descriptor.
The result of this deduction is a set of algebraic equations. Solving these equations will lead to the solution of the problem and obtaining the field equation that *describes* the given field completely [12].
Consider a the following field equation:

$$\underline{\nabla . B} = 0 \qquad (1)$$

$$\underline{\nabla} \times \underline{H} = \underline{J} \qquad (2)$$

$$\underline{H} = [\upsilon]\underline{B} \qquad (3)$$

where,

$$[\upsilon]=\begin{bmatrix}\upsilon_x & 0 & 0\\ 0 & \upsilon_y & 0\\ 0 & 0 & \upsilon_z\end{bmatrix}$$

$$\underline{B}=\underline{\nabla}\times\underline{A} \qquad (4)$$

Where $\underline{\text{B}}$ is the flux density, $\underline{\text{H}}$ is the magnetic field intensity, $[\upsilon]$ is the reluctivity tensor, and $\underline{\text{A}}$ is the magnetic vector potential.

If $\underline{a}_x$, $\underline{a}_y, \underline{a}_z$ are unit vectors in the direction of coordinate axes x,y,z respectively then by substituting from the above in equation 4:

$$\begin{vmatrix}\underline{a}_x & \underline{a}_y & \underline{a}_z\\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z}\\ \upsilon_x B_x & \upsilon_y B_y & \upsilon_z B_z\end{vmatrix}=J_x\underline{a}_x+J_y\underline{a}_y+J_z\underline{a}_z \qquad (5)$$

The above equation describes the system in the rectangular coordinates system.[10]

The Galerkin method can be applied to find an approximate solution to a boundary value problem. To minimize the error associated with such an approximate solution, the method requires the error must be orthogonal to the functions used in the approximation. This is the basis of what is known as the "weighted – residual" Finite Element Methods.

After reduction and mathematical manipulation, the resulting Galerkin's approximations are of the form:

$$\int_V [N_K]\left[\frac{\partial}{\partial y}(\upsilon_z B_z)-\frac{\partial}{\partial z}(\upsilon_y B_y)-J_x\right]dV=0 \qquad (6)$$

$$\int_V [N_K]\left[\frac{\partial}{\partial z}(\upsilon_x B_x)-\frac{\partial}{\partial x}(\upsilon_z B_z)-J_y\right]dV=0 \qquad (7)$$

$$\int_V [N_K]\left[\frac{\partial}{\partial x}(\upsilon_y B_y)-\frac{\partial}{\partial y}(\upsilon_x B_x)-J_z\right]dV=0 \qquad (8)$$

where $[N_K]$ are the approximating functions, K=1,2,….,n.

By continuing the derivations, we can obtain the solution of the system in the three-dimensional case.

In this work, we will concentrate on the two-dimensional case, since it is less complex and is more suitable for two-dimensional image databases. The extension to the three-dimensional case is straight forward. With the availability of scanning techniques that will be able to represent three-dimensional objects, this will give more power to our proposed technique. The ***three-dimensional-images descriptor*** *will be handled in a similar manner to the two-dimensional case presented in this work. This is one of the powerful features of our technique: the three-dimensional image indexing.*

The three-dimensional finite element analysis, mentioned previously, can be reduced into two-dimensional analysis, if the energy storage in the end regions is not of interest. The resulting equations for the magnetic vector potential and flux density vectors (by reduction and substitution from 5) will be of the form:

$$\frac{\partial}{\partial x}(\upsilon_y\frac{\partial A_z}{\partial x})+\frac{\partial}{\partial y}(\upsilon_x\frac{\partial A_z}{\partial y})=-J_z \qquad (8)$$

The boundary conditions usually encountered are:

a) Dirichlet boundary conditions: $A_z=0$ on Γ_1
b) Neuman boundary conditions: $\upsilon_y\frac{\partial A_z}{\partial x}\underline{a}_x+\upsilon_x\frac{\partial A_z}{\partial y}\underline{a}_y=0$ on Γ_2

Applying Galerkin's method we obtain:

$$\int_\Omega[\frac{\partial}{\partial x}(\upsilon_y\frac{\partial A_z}{\partial x})+\frac{\partial}{\partial y}(\upsilon_x\frac{\partial A_z}{\partial y})+J_z]N_i d\Omega=0 \qquad (9)$$

where N_i are the approximating functions for i=1,2,…,n. By reducing the order of the highest order derivatives and then applying Gauss's theorem to reduce the integral into a surface integral, we get:

$$\int_S[\upsilon_y N_i\frac{\partial A_z}{\partial x}\underline{a}_x+\upsilon_x N_i\frac{\partial A_z}{\partial y}\underline{a}_y]dS-$$
$$\int_\Omega[\upsilon_y\frac{\partial N_i}{\partial x}\frac{\partial A_z}{\partial x}+\upsilon_x\frac{\partial N_i}{\partial y}\frac{\partial A_z}{\partial y}]d\Omega+\int_\Omega J_z N_i d\Omega=0 \qquad (10)$$

To solve in two-dimensions by descritizing into finite elements, the use of triangles is convenient. The advantage of using such elements lies in reducing the computational cost [10]. (as shown in Figure 1).

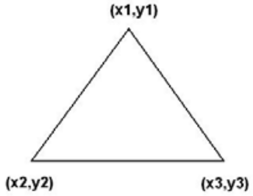


***Figure 1**: A triangular element of the mesh*

Integration is then performed over each element in the domain. The discretized equations are given by:

$$\sum_{j=1}^{n}[\int_{\Omega}(\upsilon_y \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \upsilon_x \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y})A_j]d\Omega =$$
$$\sum_{j=1}^{n}\int_{\Omega} JN_i d\Omega + \sum_{j=1}^{n}\int_{S}[N_i(\upsilon_y \frac{\partial N_j}{\partial x}\underline{a}_x + \upsilon_x \frac{\partial N_j}{\partial y}\underline{a}_y)A_j]dS \tag{11}$$

i=1,2,…,n

Where n is the total number of nodes in the discretized domain. These equations can be written in matrix form:

$$\text{R A =I} \tag{12}$$

Where R is the global reluctivity matrix of dimensions n x n elements, A is the global vector representing the magnetic vector potential of dimensions n x 1 and I is the current vector of dimension n x 1.

R is a non-singular matrix of elements on the following form:

$$r_{ij} = \int_{\Omega}[\upsilon_y \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \upsilon_x \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}]d\Omega \tag{13}$$

Having decided upon using the triangular elements, the next step is to choose the type of the approximating functions for the magnetic vector potential $A_z(x,y)$ inside each element. Polynomials are most commonly used for this step.

An approximating polynomial would look like this:

$$A_z^{(e)}(x,y) = \alpha_1 + \alpha_2 x + \alpha_3 y \tag{14}$$

solving for α and substituting we get:

$$A_z^{(e)} = \frac{1}{2S^{(e)}}((a_1 + b_1 x + c_1 y)A_{z_1}^{(e)} +$$
$$(a_2 + b_2 x + c_2 y)A_{z_2}^{(e)} + (a_3 + b_3 x + c_3 y)A_{z_3}^{(e)}) \tag{15}$$

where:

$$a_1 = x_2 y_3 - x_3 y_2$$
$$a_2 = x_3 y_1 - x_1 y_3$$
$$a_3 = x_1 y_2 - x_2 y_1$$

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

and:

$$2S^{(e)} = \begin{vmatrix} 1 & x_1^{(e)} & y_1^{(e)} \\ 1 & x_2^{(e)} & y_2^{(e)} \\ 1 & x_3^{(e)} & y_3^{(e)} \end{vmatrix}$$

The flux density can then be subsequently calculated [10].

The elements of the reluctivity matrix can be found by substituting for N_i, N_j, by the approximating polynomials. Performing the differentiation, we get:

$$r_{ij} = \int_{S^{(e)}} \frac{[\upsilon_y b_i b_j + \upsilon_x c_i c_j]dS}{(2S^{(e)})^2} \tag{16}$$

It can be proven that the flux densities in both the x and y directions are constant and thus υ_x, υ_y are also constants within the element [10]. This leads to the fact that all the quantities within the integration are constant. So equation 16 becomes:

$$r_{ij} = \frac{\upsilon_y b_i b_j + \upsilon_x c_i c_j}{4S^{(e)}} \tag{17}$$

and R(e), the elemental reluctivity matrix is then given by:

$$R(e) =$$

$$\frac{1}{4S^{(e)}}\begin{bmatrix} \upsilon_y b_1 b_1 + \upsilon_x c_1 c_1 & \upsilon_y b_2 b_1 + \upsilon_x c_2 c_1 & \upsilon_y b_3 b_1 + \upsilon_x c_3 c_1 \\ \upsilon_y b_1 b_2 + \upsilon_x c_1 c_2 & \upsilon_y b_2 b_2 + \upsilon_x c_2 c_2 & \upsilon_y b_2 b_3 + \upsilon_x c_2 c_3 \\ \upsilon_y b_1 b_3 + \upsilon_x c_1 c_3 & \upsilon_y b_1 b_3 + \upsilon_x c_1 c_3 & \upsilon_y b_3 b_3 + \upsilon_x c_3 c_3 \end{bmatrix} \tag{18}$$

This matrix is obviously symmetric. The global reluctivity matrix is then assembled as:

$$R_{Global} = \begin{bmatrix} R_{e1} & 0 & 0 & 0 \\ 0 & R_{e2} & 0 & 0 \\ 0 & 0 & ... & 0 \\ 0 & 0 & 0 & R_{en} \end{bmatrix} \tag{19}$$

This is a banded and symmetric matrix. This matrix completely describes the system through its properties [10]. The matrix band width can be minimized with proper choice of element node numbers. The magnetic field potential can also be calculated [10], but this is of no interest to us in this work.

4. Methodology

In the previous section, we have calculated the field equation of the given boundary value field problem. In general, we have several forces (flux density B, Magnetic field intensity H, magnetic vector potential A) that are applied on some material with given properties (the reluctivity tensor υ). The resultant field is a function of all of the above factors. *That is, the equation of the field is actually a representation of the total effect of the above factors on the given material.*

Now, consider the elemental reluctivity matrix presented above. This matrix actually relates several factors together:

1. The coordinates of the given material element boundary.
2. The reluctivity tensors, which in turn represent the material properties.

The coordinates of the boundary result from the relation between the element of the discretized mesh that is constructed on the material. Each element is described by its coordinates in the space.

The reluctivity tensor [υ] determines the behavior of the field according to the material under consideration (Figure 2). This implies that two different materials (i.e., υ_1 <> υ_2) having the same shape, would still result in different fields around them even if the applied forces (A,B and H) are the same.

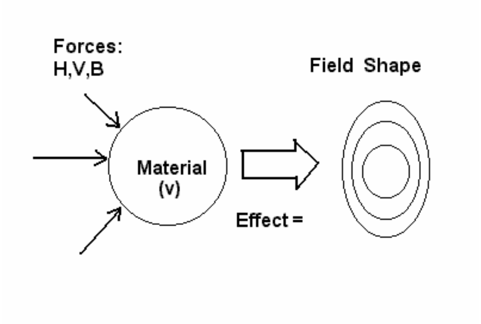


***Figure 2**: Factors affecting Field Shape*

The most important part is, the reverse fact: If we apply the same forces on two materials having the same properties (i.e., $\upsilon_1 = \upsilon_2$) and the two resultant fields turn out to be different, then the reluctivity matrix will actually be representing the shape of the material (physical dimensions and coordinates of the material). That is possible to happen if the two pieces of the same material are different in size (which is very natural, e.g., a big magnet will generate a stronger field that a smaller magnet of the same material and in the same environment).

This leads to the following conclusion: The elemental reluctivity matrix actually describes the physical dimensions of the element. It also depends on the material properties. If all the elements have the same properties (i.e., same material), then the elemental reluctivity matrix will represent the shape of the element purely.

The above discussion leads to the following hypothesis: The global reluctivity matrix of the assembled system (all the elements in the mesh), under the same assumptions as discussed in the previous paragraph, will indeed represent the shape of the overall field or system. Now for our basic goal in shape description, for any shape in a given image, the hypothesis we used in this work is that the shape is resulting from a field distribution. We also assume that all the shapes have the same material properties, for simplicity we set:

$$\upsilon_y = \upsilon_x = 1 \tag{20}$$

Then the finite element mesh is constructed on the shape under consideration (the details of constructing the mesh are presented in [11]) and the elements are defined (triangular elements are chosen for the ease of calculation, which is a goal, also they produce precise result [10]).

It is also independent of the applied forces on the shape, which in the case of an image, do not exist, or may be considered constant to all the shapes (unity) for the sake of mathematical reasoning.

Therefore, the shape descriptor that we need is the Galerkin matrix itself. We call this the Graphical Galerkin (GG) shape descriptor.

The Galerkin matrix as in equation (19) is a banded matrix. In our case of triangular elements the bandwidth is 3 (as shown in equation (18)). The calculation of the matrix is basically simple additions and multiplications.

The GG shape descriptor is translation-invariant by nature. From equation (18) we clearly notice that the calculation of the descriptor actually relies on the differences between the vertices of elements, rather than their absolute coordinates.

Originally in the Galerkin method, the reluctivity matrix contains the material properties (υ). This is a very powerful feature that represents the solid body shape and its elasticity using Maxwell's equations. This is the actual power of the finite element method. These characteristics are inherited in our Galerkin matrix model through the material properties (υ).

We can rephrase the elemental reluctivity matrix in equation (18) to the following:

$$r(e) = \frac{\nu_y}{4s(e)}\begin{bmatrix} b_1b_1 & b_1b_2 & b_1b_3 \\ b_1b_2 & b_2b_2 & b_3b_2 \\ b_1b_3 & b_3b_2 & b_3b_3 \end{bmatrix} + \frac{\nu_x}{4s(e)}\begin{bmatrix} c_1c_1 & c_1c_2 & c_1c_3 \\ c_1c_2 & c_2c_2 & c_3c_2 \\ c_1c_3 & c_3c_2 & c_3c_3 \end{bmatrix} \tag{21}$$

The last equation shows that we have two material properties that we can use in our descriptor to describe any feature we want beside the shape itself. In the case of pure shape descriptor we set $\upsilon_x= \upsilon_y=1$.

If we desire to incorporate the color in our description of the shape, we would set $\upsilon_x= \upsilon_y=$ the color value of that element of the shape. Thus each element of the shape is described (through its elemental matrix) by its shape and its color as well. So, if we have three identical shapes, i.e., their Galerkin matrices are similar (see Figure 3), but differ in their colors, the two with the nearest color property υ will be considered similar (r(e) is proportional to the color υ).

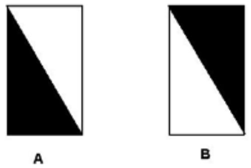


***Figure 3**: These two boxes are identical in shape but different in color.*

Other extensions to our descriptor may include texture: taking $\upsilon_x= \upsilon_y=$ the texture property of the element, the descriptor is capable to describe the elements texture as well as its shape

5. Complexity Analysis

We consider the basic calculations required in the calculation of the descriptor. The Galerkin matrix is clear to be a banded matrix. The band width is to be 3 (equation 18). Thus having n elements in our finite element mesh, we'll have a Galerkin matrix of dimension n*n. The number of elements on the diagonal of a square matrix is $\sqrt{2}$ n. The total number of elements will become about $3\sqrt{2}$ n (by multiplying the band by the no. of elements on the diagonal). Each of the elements of this matrix will require a constant time to calculate. Thus the complexity of the total calculation will be $3\sqrt{2}$ n. Which is of O(n). This is quite acceptable compared to some well known much less powerful image descriptors.

6. Results

In this section we present samples of the results we have obtained using our FERecognizer Ver 1.1. The original test sample included 450 shapes. Here we present only a subset of the work as a demonstration. We have implemented this software tool to construct the GG Shape descriptor matrix of a given image entered by the user. The search process involves calculating the similarity between the GG shape descriptors of images. The tool allows the use of several similarity measures as a parameter for the user to choose and for us as a research area to determine the most efficient similarity measure to use. The similarity measures we included are listed as follows.

Euclidean Distance: The Euclidean distance is the simplest form of distance measures. Given two matrices, A and B, the distance is calculated using:

$$D(e) = \sum_{i=1}^{n}\sum_{j=1}^{m}(A_{i,j} - B_{i,j}) \qquad (22)$$

It is clear that this distance measure is not symmetric [11].

Absolute Distance: It is similar to the above measure, and it is given by:

$$D(a) = \sum_{i=1}^{n}\sum_{j=1}^{m}\left| A_{i,j} - B_{i,j} \right| \qquad (23)$$

It is clearly symmetric [11]. It is similar to calculating the angle between two vectors.

Least Squares: A variation of the Euclidean distance is the least squares distance, defined as:

$$D(l) = \sum_{i=1}^{n}\sum_{j=1}^{m}(A_{i,j} - B_{i,j})^2 \qquad (24)$$

This measure is symmetric, but it emphasizes the errors or deviations (squares them).

QBIC Method: It is a quadratic-form distance measure that is used successfully in the IBM QBIC (Query by Image Content) system [12]. It compares each element of a matrix with all the elements of the other matrix. It is given by:

$$D(q) = (A - B)^T S(A - B) \qquad (25)$$

S is a similarity matrix calculated using the Euclidean distance for example.

The method produces better results, since it gives a global measure of the distance rather than a local element-by-element operation. The disadvantage of the method is the large number of calculation.

Cosine Distance: The cosine distance metric computes the difference in direction [13]. It is calculated using the following:

$$D(c) = Cos(\theta) = \frac{A^T B}{|A||B|} \qquad (26)$$

We have chosen simple shapes to test the descriptor in order to demonstrate the efficiency of our algorithm (Figure 4). More complex shapes will be handled the same way, except that they will require more preprocessing using edge detection algorithms; which is out of the scope of our study. Simple shapes also contain less information than complex one, thus it is better to test the new descriptor with such shapes. It is much more difficult to distinguish simple shapes than complex ones due to the large similarity between the simple shapes. So if the descriptor could distinguish the simple shapes, combinations of these simple shapes will be much easier to distinguish.

Table 1 shows a sample of the results on the given shapes. We indicate the query image name underlined, then the distance between this image and the rest of the images in the sample space. The closest distance is highlighted. It is clear from the results in table 1, that the distances between similar shapes, e.g. large triangle and small triangle, is always the smallest. The similarity measure used depends on the application. However, it is clear, as shown in table 1, that absolute distance measure and the least squares distance measure produce the best results. A dynamically-weighted voting technique between three of the measures could be used to decide the similarity. This could be done by taking the minimum distance of each measure, and checking the number of votes to decide the closer image. Note that some measures like the QBIC and the Euclidean are not symmetric, so we take the absolute value in comparison.

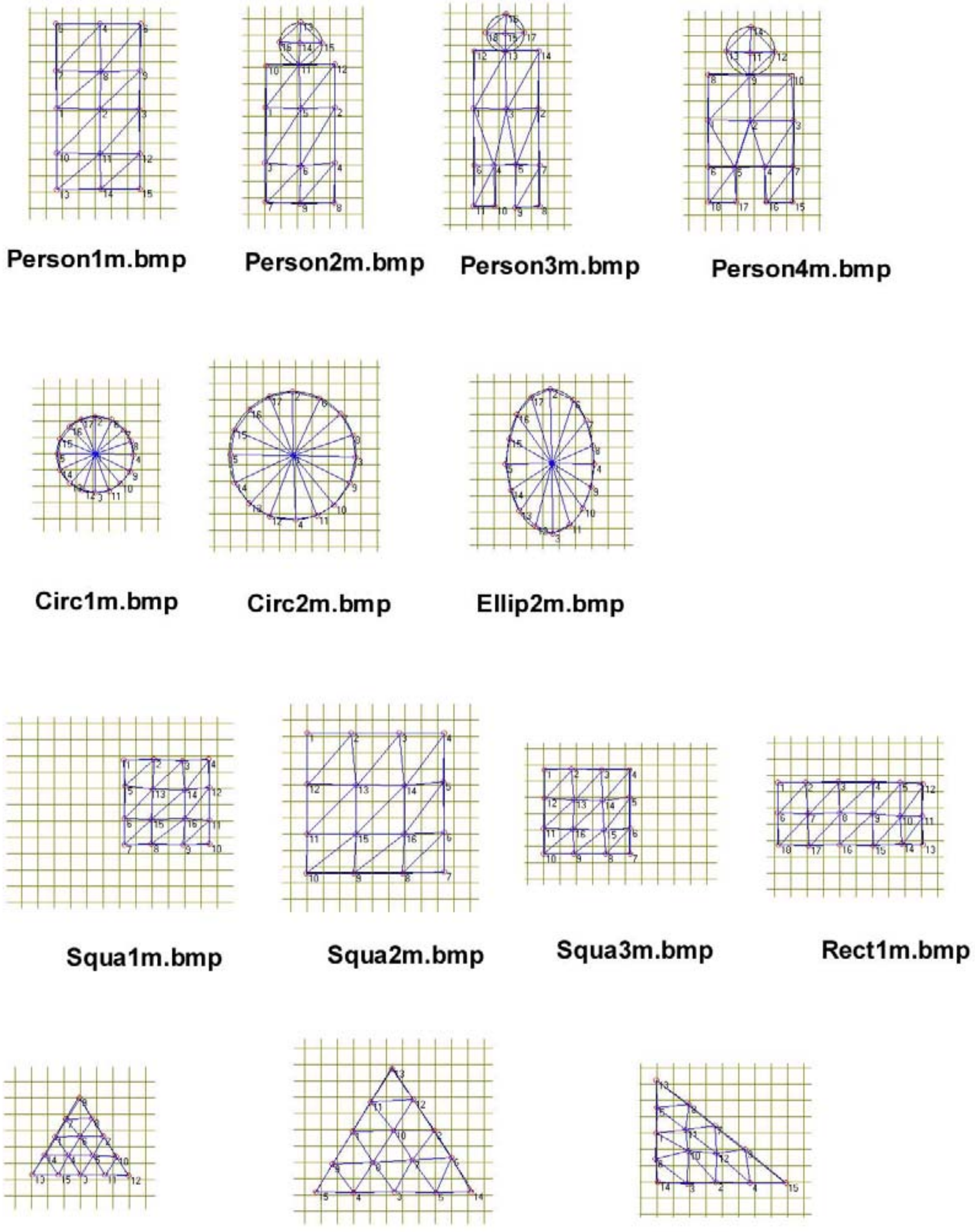
The QBIC measure produced false results with more complex shapes, as shown in table 1. The COS measure will be disregarded in our analysis.

7. Summary and Conclusion

In this work we have presented a novel shape descriptor, called the Graphical Galerkin shape descriptor. It is based on the Galerkin weighted residual finite element method.

There are several features regarding the use of our proposed descriptor, these are:

1. Simple calculations.
2. Incorporates shape and color (as well as other properties) in a single descriptor.
3. Translation invariant, since it is calculated through differences of coordinates (see equations in this chapter).
4. Preserves spatial information about the color (or any other property used) as it is stored in the elemental matrix.
5. Storage efficient as all we need to store is a banded sparse matrix.
6. Allows manipulating data of compressed images as all image features are stored in the Galerkin matrix, so the original image can be highly compressed using lossy algorithms.
7. Describes an image uniquely and accurately.
8. Can be easily extended to describe three-dimensional images as was discussed earlier.



Person1m.bmp Person2m.bmp Person3m.bmp Person4m.bmp

Circ1m.bmp Circ2m.bmp Ellip2m.bmp

Squa1m.bmp Squa2m.bmp Squa3m.bmp Rect1m.bmp

Tria1m.bmp Tria2m.bmp Tria3m.bmp

***Figure 4**: Discretized sample shapes (discretized using FERecognizer Ver. 1.1)*

***Table 1**: Distances report (closest distances are highlighted)*

```
"Welcome..."
This is the Distances Report..
Generated by FERecognizer Ver. 1.1
--------------------------------

-->             Euclidean    LeastSQ      Absolute     COS          QBIC
------------------------------------------------------------------------------------------------------------
circ1m.bmp
----------------
circ1m.bmp      0            0            0            1.22966      0
circ2m.bmp      .279220      23.0990      4.72473      6.13996      .164344
person3m.bmp    1.24798      76.3638      90.1457      3.09302      166.618
person4m.bmp    -1.27031     71.8726      71.5443      2.17774      -110.946
squa1m.bmp      .203781      74.7624      92.0041      1.45173      27.3521
squa2m.bmp      .296905      75.2618      92.5390      1.01576      40.0939
tria1m.bmp      1.96546      74.1692      91.2702      7.57434      245.412
tria2m.bmp      -3.29237     74.5954      73.6692      6.56010      -287.562


person3m.bmp
----------------
circ1m.bmp      -1.24798     76.3638      90.1457      2.83729      -166.618
circ2m.bmp      -.968762     74.8585      90.0626      2.47824      -129.995
person3m.bmp    0            0            0            2.65162      0
person4m.bmp    -2.64232     50.3805      30.8103      2.69653      -58.5722
squa1m.bmp      -1.21220     66.6802      62.0029      2.60514      -74.0396
squa2m.bmp      -1.07414     67.6272      62.5504      2.51944      -65.8674
tria1m.bmp      .717479      55.5231      56.0145      2.45456      37.0733
tria2m.bmp      -4.54035     55.2603      57.2671      2.36133      -289.067


squa1m.bmp
----------------
circ1m.bmp      -.203781     74.7624      92.0041      1.78496      -27.3520
circ2m.bmp      7.54383      74.6251      91.8314      1.63827      10.1236
person3m.bmp    1.21220      66.6802      62.0029      1.59962      74.0395
person4m.bmp    -1.43012     56.8550      57.5838      1.56684      -73.6017
squa1m.bmp      0            0            0            1.60489      0
squa2m.bmp      .136806      20.2657      4.73629      1.63887      4.70659
tria1m.bmp      1.76168      37.7644      42.1963      1.61874      62.5271
tria2m.bmp      -3.49615     38.4295      37.4982      1.60562      -105.304


tria1m.bmp
----------------
circ1m.bmp      -1.96546     74.1692      91.2702      1.28791      -245.411
circ2m.bmp      -1.68624     74.1451      91.3006      1.21571      -208.977
person3m.bmp    -.717479     55.5231      56.0145      1.19704      -37.0732
person4m.bmp    -3.23577     39.9037      47.3805      1.18017      -138.514
squa1m.bmp      -1.76168     37.7644      42.1963      1.16917      -62.5271
squa2m.bmp      -1.66855     38.0777      42.7259      1.15835      -59.6202
tria1m.bmp      0            0            0            1.17025      0
tria2m.bmp      -5.25783     21.7697      41.0367      1.15525      -194.040
```

We believe that this approach in the area of visual information retrieval is accurate, computation and storage efficient as compared to other standard techniques such as discrete Fourier transform.

Future work includes the extension of the descriptor to the three dimensional case, and incorporating other image features into the descriptor.

References

[1] V.S Subrahmanian, "Principles of Multimedia Database Systems," 1998.

[2] Y. Rui, S.F. Chang, "Image Retrieval: Current techniques, promising directions, and open issues," Journal of Visual Communications and Image Representation, 10, pp. 39-62, 1999.

[3] A. Chu, "Survey on Indexing Techniques in image DMBS," CMPUT 509- Final Report. Dept. of Computing Science, University of Alberta, 1999.

[4] A. Yoshitaka, T. Ichikawa, "A survey on Content-based retrieval for multimedia databases," IEEE Transactions on Knowledge and Data Engineering, vol. 11, January/February, 1999.

[5] J.K. Wu, "Content-based Indexing of multimedia databases," IEEE Transactions on Knowledge and Data Engineering, vol. 9, November/December, 1997.

[6] D. M. Squire, H. Mueller, W. Mueller and T. Pun, "Design and Evaluation of a content-based image retrieval system," Design and Management of Multimedia Information Systems: Opportunities and Challenges, Idea Group Publishing, 2001.

[7] G. Lu, "On image retrieval based on color," Storage and retrieval for image and video databases volume 2670, SPIE, San Jose,1996.

[8] J. Smith, S.F. Chang, "Tools and techniques for color image retrieval," SPIE proceedings, 2670, pp. 1630-1693, February 1996.

[9] Mehrotra. R., J. E. Gary, "Similar-Shape retrieval in shape data management," IEEE Computer, pp. 57-62,September 1995.

[10] G. Pass, R. Zabeh, "Histogram refinement for content-based image retrieval," IEEE Workshop on Applications of Computer Vision, pp. 96-102, December 1996.

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

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

[12] M. Flickner, H. Sawheney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petrovic, D. Steele, P. Yanker, "Query by image and video content: The QBIC system," IEEE Computer, pp. 23-32, September, 1995.

[13] M. Sticker, M. Orengo, "Similarity of color images," Storage and Retrieval for Image Databases, SPIE Proceedings, vol.2420, February 1995.