Three-dimensional shape descriptors and matching procedures

H. Foulds
School of Computer, Statistical and Mathematical Sciences
North-West University
Potchefstroom, 2531,
South Africa
Henry.Foulds@nwu.ac.za

G.R. Drevin
School of Computer, Statistical and Mathematical Sciences
North-West University
Potchefstroom, 2531,
South Africa
Gunther.Drevin@nwu.ac.za

ABSTRACT

Shape descriptors are used to identify objects in the same way that human fingerprints are used to identify people. Features of an object are extracted by applying functions to the digital representation of the object. These features are structured as a vector which is known as the shape descriptor (feature vector) of that object. The objective when constructing a shape descriptor is to find functions that will yield shape descriptors that can be used to uniquely identify or at least classify an object. A measure of similarity is required to identify or classify an object. The similarity between two objects is computed by applying a distance function to the shape descriptors of the two objects.

The objective of this paper is to examine two of the possible techniques in three-dimensional shape descriptor construction based on Fourier analysis, and to find a descriptor that is able to not only classify, but also identify objects.

Keywords

Shape descriptor, Fourier transform.

1. INTRODUCTION

The Fourier transform has long been used in image and signal processing to convert data from the spatial domain to the frequency domain. Applying the Fourier transform to spatial data results in a set of coefficients that represent different frequency variations in the data. Lower order coefficients represent low frequency variations that normally have large amplitudes while higher order coefficients represent high frequency variations that normally have small amplitudes. The Fourier transform is normally used on data consisting of one dimension (signal processing) or two dimensions (image processing), but can also be applied to three dimensions as seen in Zhang & Chen [Zha01a] and Vranic & Saupe [Vra01a].

Fourier coefficients form conjugate pairs, except for the lowest order coefficient. To create a descriptor from Fourier coefficients the magnitudes of the coefficients are calculated. The first K of these values, corresponding to the K lowest Fourier coefficients, are used. K is a threshold value to establish the smallest number of coefficients needed to identify each object.

In this paper two methods are developed that can be used with the Fourier transform to identify objects. In both methods a $\theta\varphi$-matrix is created to which a Fourier transform is applied to create feature vectors.

The data used in this project are in the form of three-dimensional triangle mesh models. In a triangle mesh model each object is approximated by a collection of structured triangles. The triangles represent the faces of the mesh model and each face has three vertices. Some of the faces share common vertices or sides. The faces are represented by an $N\times3$ matrix with N the number of faces. The vertices are represented by an $M\times3$ matrix with M the number of unique vertices. The three values in each row in the faces matrix represent the three vertices of a face. Each value is the row number of a vertex in the vertices matrix. The first step in calculating the descriptor of an object is to obtain the N×3 matrix representing the N faces and the M×3 matrix representing the M vertices that approximate the object.

Matching objects when they do not have the same pose causes a serious problem. To solve this problem the descriptor has to be translation and rotation invariant. The most frequent used method to solve this problem is to apply Principle Component Analysis (PCA, Hotelling transform, Karhunen-Loeve transform) to the triangle mesh of an object before the descriptor is calculated [Zha01a, Vra01a, Vra03a, Pap06a]. The result of doing PCA is that all objects that are similar will have similar orientations and the descriptors that are calculated will have a smaller match distance. PCA can be used as a second step in calculating a descriptor to ensure rotation and translation invariance when the descriptor technique itself is not invariant to rotation and translation. An example of PCA can be seen in Figure 1. When classifying objects scale invariance is required. The objective of this project was to solve an exact matching problem, therefore scale variance was not considered.

The PCA method uses the eigenvalues and eigenvectors, calculated from the covariance matrix of the vertices matrix, to rotate an object in such a way that the first principal axis aligns with the x-axis, the second principal axis with the y-axis and the third principal axis with the z-axis. On completion of PCA the largest variance of mass is in the x-axis direction. For highly symmetrical objects the eigenvalues are very similar. The small differences in symmetrical objects cause errors during PCA because the eigenvalues are similar and a small change in the eigenvalues causes the eigenvectors to change dramatically. Figure 2 shows the result of applying PCA on symmetrical objects. Using PCA to normalize pose when objects are symmetrical is not very successful.

2. REPRESENTATION

Centroid distance is defined as the distance from the centroid of an object to the surface of that object. The object is centred and orientated using PCA. After PCA the centroid distance is calculated for angles θ and φ with $\theta \in [0,\pi]$ and $\varphi \in [0,2\pi]$. θ is the polar angle from the z-axis and φ the azimuthal angle in the xy-plane from the x-axis. The centroid of the object is located at the origin of the coordinate framework. A two-dimensional θφ-matrix is created where the rows represent θ-values and the columns represent φ-values. The value located at index [θ,φ] of the matrix is the distance from the centroid, in the direction (θ,φ), to the surface of the object. If the surface of the object is intersected more than once, the maximum value is used. In the result for $\theta \in [0,2\pi]$ the elements in the matrix for $\theta \in [\pi,2\pi]$ will be a mirror of the values for the elements with $\theta \in [0,\pi]$. For this reason $\theta \in [0,\pi]$ is used.

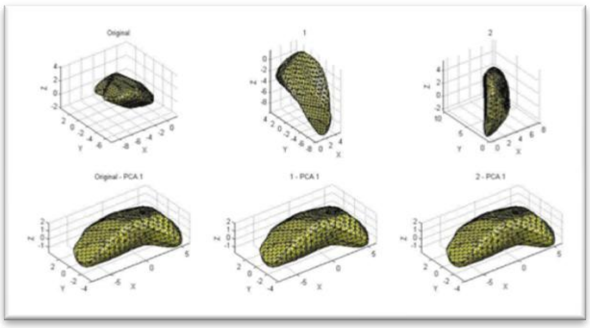

Figure 1. The top row shows three different orientations of the same object. The bottom row shows the result of applying PCA to the top row.

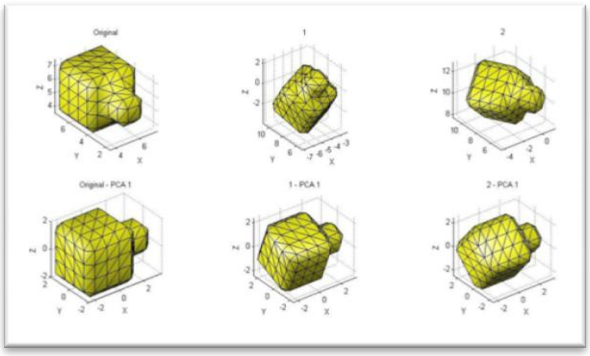

Figure 2. The top row shows three different orientations of a symmetrical object. The bottom row shows the result of applying PCA to the top row.

The discrete Fourier transform (DFT) of the θφ-matrix is calculated and the result used to create the feature vector. The feature vector is created by applying a method, similar to the process used by Vranic & Saupe [Vra01a], to the Fourier coefficients. For a chosen integer value K, the (2K+1)×(2K+1) matrix centred on the lowest frequency coefficient is returned. Except for the lowest frequency coefficient, all other coefficients form conjugate pairs. The feature vector is defined as the magnitudes of the Fourier coefficients. The coefficients are ordered according to distance from the lowest frequency coefficient.

Algorithm:

1. Get faces matrix and vertices matrix
2. Do PCA
3. Centroid is located at origin after PCA.

4. Get θφ-matrix by calculating distance from centroid to last triangle intersected for each [θ,φ] (Two different methods are discussed in paragraph 2.1 and 2.2)
5. Calculate 2D Fourier transform of θφ-matrix
6. Set the number of Fourier coefficients by choosing value for K
7. Get the feature matrix by calculating the absolute value of the elements in the (2K+1)×(2K+1) matrix in the centre of the Fourier transformed θφ-matrix (centred on the lowest frequency coefficient)
8. Calculate the 2D Euclidean distance of each element in the feature matrix from the centre
9. Reorder the elements in the feature matrix into a one-dimensional array, sorting them according to the distance calculated in step 8 (For elements at the same distance, the same order is used for each object as they are orientated with PCA)
10. The result is the feature vector.

Two techniques to calculate the distance from the centroid to the surface of an object were evaluated.

2.1. Method 1

In this technique four θφ-matrices are created with $\theta \in [0,\pi]$ and $\varphi \in [0,2\pi]$. Increments of 1, 4, 9 and 18 degrees for θ and φ are used for each of the four matrices respectively. Larger increments will result in smaller matrices. For each of the directions (θ,φ) in the θφ-matrix the distance is calculated from the centroid to an intersection with a face. If more than one face is intersected, the maximum distance is used. Each face may be intersected multiple times for different directions (θ,φ).

2.2. Method 2

In the second technique four θφ-matrices are also created with $\theta \in [0,\pi]$ and $\varphi \in [0,2\pi]$. Increments of 1, 4, 9 and 18 degrees for θ and φ are used for each of the four matrices respectively. For each of the faces a range of θ and φ values is determined to create a region surrounding the face. Depending on the increment length, the number of intervals for θ and φ are calculated.

This number of intervals is used to create a set of points on the face defined by intervals of λ and β in $\lambda E + \beta F + (1 - \lambda - \beta) G$ with $0 \leq \lambda \leq 1$, $0 \leq \beta \leq 1$ and $0 \leq \lambda + \beta \leq 1$. E, F and G are the vertices of the face. The distance from the centroid to the plane defined by the vertices of the face is calculated for the directions (θ,φ) for all the points in the set. This procedure ensures that distances are calculated only for the set of points on a face.

Using method 2 speeds up the process of calculating the θφ-matrix considerably as there is no need to repeat the calculations for all the directions of all the faces. The process is applied to each face and the results obtained for all the faces are combined. The results obtained with these two techniques are given in section 4.1.

3. MATCHING

3.1. Distance functions

Distance functions are used to determine the distance between two feature vectors f_1 and f_2. A number of distance functions are defined, among others, by Vranić [Vra03a], Osada *et al.* [Osa01a] and Long *et al.* [Lon02a]. Distance functions used in this project are:

- l_1 norm

$$d_1(\mathbf{f}_1, \mathbf{f}_2) = \|\mathbf{f}_1 - \mathbf{f}_2\|_1 = \sum_{i=1}^{N} |\mathrm{f}_{1i} - \mathrm{f}_{2i}| \qquad (1)$$

This function defines the distance between two feature vectors as the sum of the absolute values of the differences between each set of corresponding elements.

- l_2 norm

$$d_2(\mathbf{f}_1, \mathbf{f}_2) = \|\mathbf{f}_1 - \mathbf{f}_2\|_2 = \sqrt{\sum_{i=1}^{N} (\mathrm{f}_{1i} - \mathrm{f}_{2i})^2} \qquad (2)$$

With this function the distance between two feature vectors is the square root of the sum of the squares of the differences between each set of corresponding elements.

- Minkowski with p=0.8

The next two distance measures use the Minkowski norm given in Long *et al.* [Lon02a] as

$$d(\mathbf{f}_1, \mathbf{f}_2) = \left(\sum_{i=1}^{N} |\mathrm{f}_{1i} - \mathrm{f}_{2i}|^p \right)^{\frac{1}{p}}. \qquad (3)$$

The value of p was chosen as 0.8 for this norm.

- Minkowski with p=1.2

This norm uses the same form as in Equation (3), but with p chosen as 1.2.

- l_{max} norm

The l_{max} norm is the maximum of the absolute values of the differences between the sets of corresponding elements.

$$d_\infty(\mathbf{f}_1, \mathbf{f}_2) = \|\mathbf{f}_1 - \mathbf{f}_2\|_\infty = \max_{1 \leq i \leq N} |\mathrm{f}_{1i} - \mathrm{f}_{2i}| \qquad (4)$$

3.2. The matching process

Matching is done by first applying the same descriptor method to two different objects after which the distance between the two feature vectors is calculated. This distance is used as a measure of the similarity of the two objects.

When working with sets of objects, a matrix of distance values is created to match two sets of objects. The rows represent the objects of the first set and the columns represent the objects of the second set. Each value in the matrix is used as a measure of the similarity between two objects, an object from the first set as defined by the row number and an object from the second set as defined by the column number. For each object in the first set, the closest match is found from the second set. With this technique an object can be identified from a set of possible objects.

The elements in a similarity matrix M are calculated using one of the distance functions, therefore $M(i,j)=d(f_i,f_j)$ where f_i is the feature vector of object i on row i of matrix M and f_j is the feature vector of object j on row j of matrix M. d is the chosen distance function.

In this study three sets of 60 objects are used to test the usability of the descriptors for identification. Examples of these objects are given in Figure 3. Each of the three datasets used in the matching process consist of 600 triangle mesh models representing the 60 objects. Each object is digitized 10 times resulting in 10 representations of each object. Due to the nature of the digitization process there are small variations in the representations of the objects. The 600 models in each dataset are divided into two sets with five models representing each object in a set. During the matching process the 300 models representing 60 objects from the first set are matched to the 300 models representing the same 60 objects from the second set. The matching process creates a similarity matrix with the models from the first set represented by 300 rows and the models from the second set represented by 300 columns. Each element in the matrix is the result of a distance function applied to the object represented by the row from the first set and the object represented by the column from the second set. Five different distance functions are used to create five different similarity matrices. Because multiple rows and columns are used to represent feature vectors calculated from different representations of the same object, aggregates of these columns and rows are taken. The aggregate functions used are minimum, mean, maximum and sum. This results in a matrix with 60 rows and 60 columns. The objective is to identify and match objects in a set using similar representations of the same objects in another set. Because both sets contain models of all 60 objects, an object in a specific row must be matched with an object in a specific column. This is done by finding the object with the smallest distance function result.

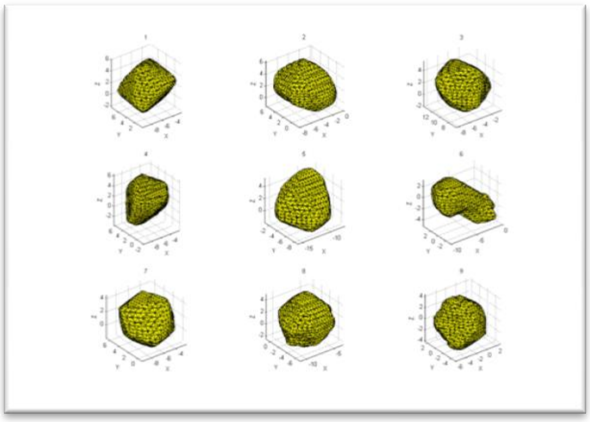

Figure 3. Representations of the first nine objects in dataset 1 used during matching.

4. RESULTS

4.1. Feature vector creation

The project was implemented using Matlab 2007 on an Intel Core2 Quad 2.4GHz PC. The first tests are to show the results obtained when applying the two techniques to calculate the θφ-matrix as discussed in section 2.1. A sphere with radius 3 consisting of 88 faces is used to evaluate the two techniques.

The results when calculating the θφ-matrix of the sphere using method 1 are shown in Table 1. The first column lists the sizes of increments used for θ and φ and the second column lists the time it took to create the θφ-matrix using method 1. The third column show the time it took to create the θφ-matrix using method 2.

Increments (degrees)	Time (s) Method 1	Time (s) Method 2
1	630	506
4	40	0.9
9	9	0.2
18	2.3	0.06

Table 1. Time needed to calculate θφ-matrix using method 1 and 2.

Calculating the θφ-matrix is considerably faster when using method 2 and incrementing θ and φ with 4 degrees or more as seen in Table 1.

Figure 4 shows the θφ-matrix for increments of 1 degree obtained using method 2. The next step is to test the technique used for the creation of the Fourier descriptors. Figure 5 shows 3 representations of objects used to construct the feature vectors. Object A is a sphere consisting of 88 faces. Object B is a rounded cube consisting of 188 faces. Objects C and D are irregular objects consisting of 2476 and 1372 faces respectively. The results of constructing the feature vectors are listed in Table 2. The θφ-matrix is calculated using the second technique with increments of 4 degrees. The feature vectors are calculated with threshold K=10. Figure 6, 7 and 8 shows the θφ-matrix, Fourier coefficients and feature vector for object C.

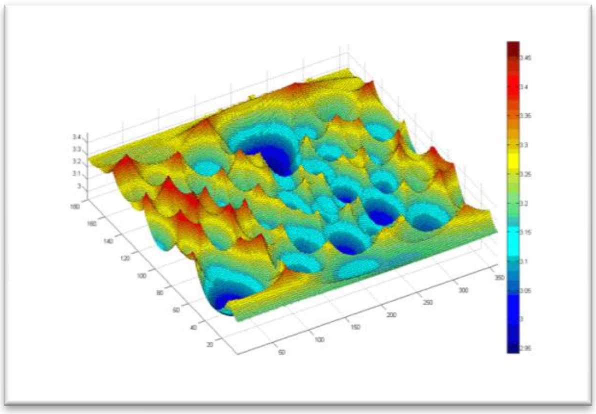

Figure 4. θφ-matrix using method 2 with increments of 1 degree.

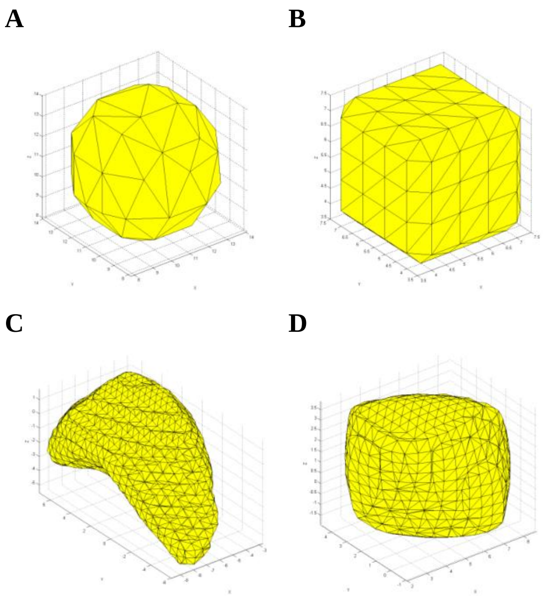

Figure 5. (A) A sphere with radius 3 consisting of 88 faces. (B) A rounded cube consisting of 188 faces. (C) An object consisting 2476 faces. (D) An object consisting of 1372 faces

Object	Time (s) Method 2
A	0.7
B	0.9
C	4.4
D	3.5

Table 2. θφ-matrix obtained by using method 2 with increments of 4 degrees.

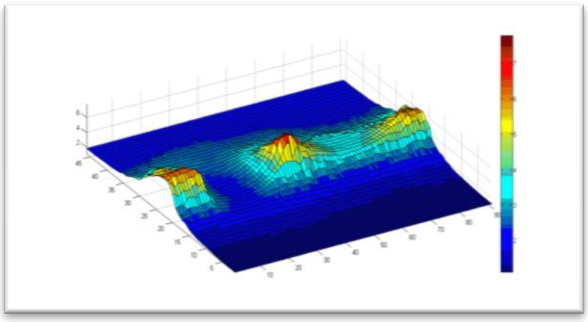

Figure 6. θφ-matrix for object C using method 2 with increments of 4 degrees.

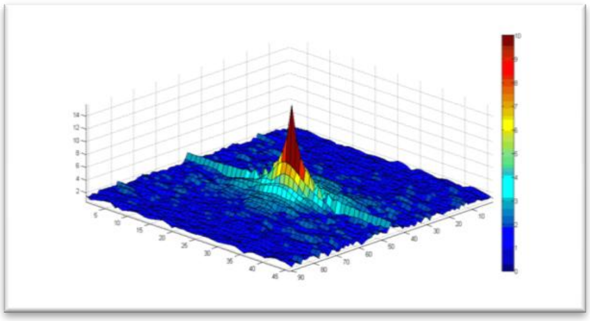

Figure 7. Fourier coefficients obtained from the DFT of the θφ-matrix for object C.

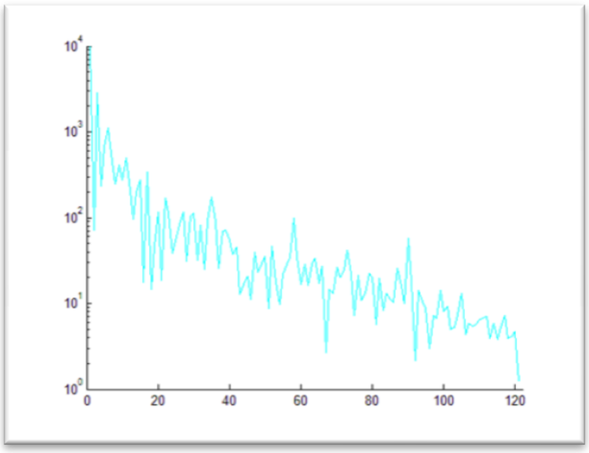

Figure 8. Feature vector constructed from Fourier coefficients for object C

Figure 9 shows the feature vectors of 9 objects. Object A is the sphere (object A) consisting of 88 faces in Figure 5. Object B is a sphere consisting of 66 faces. Object C is a rounded cube consisting of 188 faces. Objects D through G are different representations of object C in Figure 5, consisting of 2384, 2268, 2476 and 2444 faces. Object H and I are very similar objects consisting of 1372 and 1932 faces respectively. Different colours are used to indicate the different objects. Below is a list of colours used.

- Object A - (green).
- Object B - (dark green).
- Object C - (red).
- Object D - (blue).
- Object E - (dark blue).
- Object F - (cyan).
- Object G - (dark cyan).
- Object H - (purple).
- Object I - (magenta).

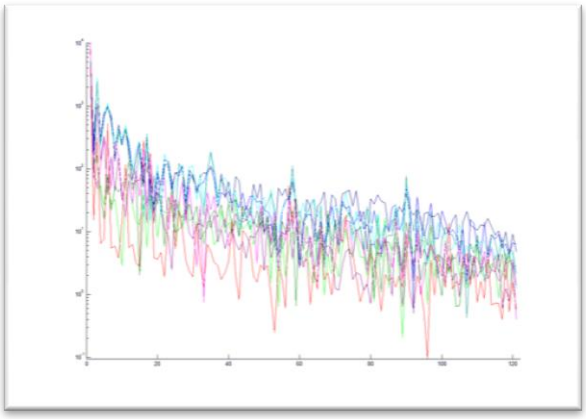

Figure 9. Feature vectors of 9 objects, also represented in the Figure 10.

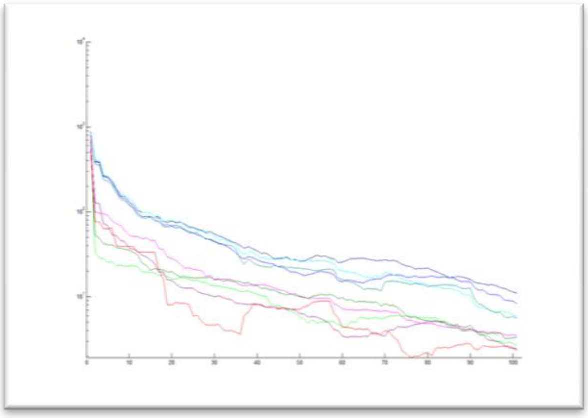

Figure 10. Moving average of length 21 of the feature vectors in Figure 9.

Smoothing of feature vectors as in Figure 10 is done only to improve the visualization of the feature vectors for printing. The original feature vectors are used in the matching process as smoothing causes a loss of feature information. Smoothing is done using a moving average of length 21.

When comparing the feature vectors in Figure 10 it is clear that the four feature vectors at the top are grouped together. These feature vectors are for objects D, E, F and G. When comparing the four triangle mesh models, it is clear that they are different representations of the same object, hence the similarity in their feature vectors. Objects A, B, H and I are very rounded in shape, and their feature vectors are also very close together.

For method 2 the θφ-matrix is calculated with increments of 4, 6 and 10 degrees for angles θ and φ giving 4050, 1800 and 648 elements for each of the matrices. Using increments of 1 degree result in excessive processing time requirements. Using large increment sizes produce feature vectors that result in inaccurate matching. For this reason increments of 4, 6 and 10 degrees where chosen.

The time needed to calculate the θφ-matrices for the different increments are compared in Figure 11. In Figure 11 the number of faces is not the only influence on processing time. The rightmost four objects in the figure are very similar, and variations in the complexity and orientation of the faces influences the processing time. Increasing the increment sizes for angles θ and φ decreases calculation speed, but causes a loss of accuracy.

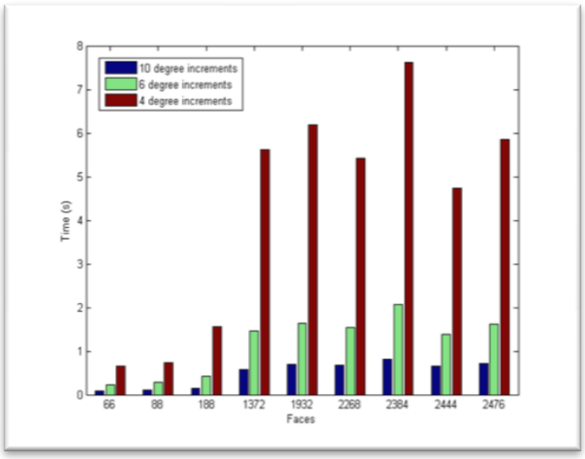

Figure 11. Processing time required to calculate the θφ-matrix for the 9 objects of Figure 9 using method 2 with increments of 10, 6 and 4 degrees.

4.2. Matching

The feature vectors are calculated with K=5 resulting in 36 elements in each feature vector. These feature vectors are calculated using increments of 10 degrees for angles θ and φ. It takes 350 seconds to calculate the feature vectors of all 600 mesh models in a dataset using increments of 10 degrees. For increments of 6 degrees it takes 940 seconds, and 2650 seconds for increments of 4 degrees. The process of calculating the θφ-matrix takes up most of the time during feature vector generation. Four aggregate functions are applied to the results of each of the distance functions, as discussed in section 3.2. The results for each of these functions are listed in the columns marked "Min", "Mean", "Max" and "Sum". The results are the number of errors made in matching the two sets of mesh models. Figure 12 contains four similarity matrices for the l_1 norm distance function calculated from the results of the four aggregate functions. Colours closer to blue indicate similar objects, while colours closer to red indicate dissimilar objects.

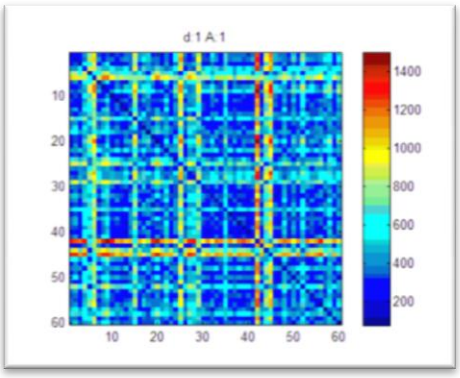

Figure 12. A similarity matrix for the l_1 norm distance function obtained from Fourier method 1 using Min aggregate. Colours closer to blue indicate similar objects, while colours closer to red indicate dissimilar objects.

The first elements of the feature vectors correspond to the lower order Fourier coefficients. The lower order Fourier coefficients relate to large changes in shape while the higher order coefficients relate to smaller variations in shape. For this reason only a number of coefficients surrounding the first coefficient is needed to create a feature vector. This approach will also reduce the influence of noise, which is usually associated with small variations. As the first elements have very large values, they will have a greater influence during the distance calculations between feature vectors. Therefore large variations in shape will result in bigger differences between feature vectors. From the results it is clear that this feature vector creation method will produce feature vectors that can be used to identify objects.

The results of the matching process are given in Tables 3 to 5. These results show that the four distance measures performed well during the matching process. Table 5 shows that without PCA the results are inferior. The errors made during identification were of objects that are exceptionally similar.

A good method for identification should have a low probability of false acceptance and a low probability of false rejection. The area closer to the origin in the ROC graph will reveal the more accurate method. For this reason the ROC graphs are displayed for a probability of false acceptance and false rejection up to 50%. The ROC graphs in Figure 13 illustrate that the distance functions yield very similar results. The two graphs in Figure 13 yielding poor results are the Kullback-Leibler divergence and Jeffrey divergence.

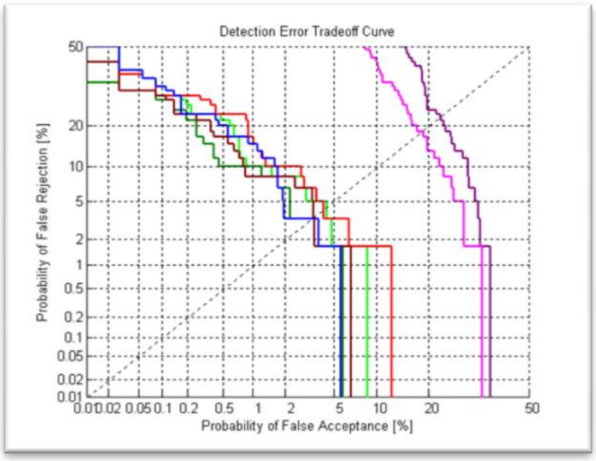

Figure 13. ROC graph of matching results for l_1 norm distance function applied to feature vectors generated using method 2.

These two distance functions gave poor results in all the preliminary tests and where excluded from the rest of the study. Even though using smaller increments result in better matches, the differences between the results observed when incrementing angles θ and φ by 10, 6 and 4 degrees are very small. The time needed for processing make increments smaller than 10 ineffective. With all techniques the "minimum" aggregate gave better results and "maximum" aggregate gave inferior results.

Distance function	Min	Mean	Max	Sum
$l_1$norm	1	2	22	2
l_2 norm	3	10	31	10
Minkowski with $p = 0.8$	1	2	18	2
Minkowski with $p = 1.2$	2	2	24	2
$l_{\max}$ norm	4	18	36	18

Table 3. Results for dataset 3 created with method 2 and increments of 10 degrees.

Distance function	Min	Mean	Max	Sum
$l_1$norm	1	3	12	3
l_2 norm	0	7	23	7
Minkowski with $p = 0.8$	2	3	13	3
Minkowski with $p = 1.2$	0	4	16	4
$l_{\max}$ norm	5	11	27	11

Table 4. Results for dataset 3 created with method 2 and increments of 4 degrees.

Distance function	Min	Mean	Max	Sum
$l_1$norm	30	49	52	49
l_2 norm	31	52	55	52
Minkowski with $p = 0.8$	30	47	51	47
Minkowski with $p = 1.2$	29	49	52	49
$l_{\max}$ norm	39	50	53	50

Table 5. Results for dataset 1 created with method 2 and increments of 10 degrees. No PCA is applied.

5. CONCLUSION

Principal Component Analysis (PCA) plays a vital part in normalizing the orientation of objects during identification. Symmetrical objects result in errors during PCA. Two methods to obtain feature vectors, using the Fourier transform, are described in this paper. Fourier methods are effective in identifying objects as they are fast and accurate. Their only drawback is that accuracy decreases when large numbers of objects in the datasets are very similar. Of the five distance measures evaluated the l_1 norm, l_2 norm, Minkowski with p=0.8 and Minkowski with p=1.2 distance measures are best suited for identification.

6. FUTURE WORK

Alternative methods to normalize orientation can also be explored to improve results.

The CSS descriptor method [Dre05a, Zha01b] is another technique that could possibly be adapted for three-dimensional identification in future research.

7. ACKNOWLEDGMENTS

The authors would also like to thank Keith Forbes and Colin Andrew for their aid during this project.

8. REFERENCES

[Dre05a] Drew, M.S., Lee, T.K. and Rova, A. Shape Retrieval with Eigen-CSS Search. Technical Report, TR 2005-07, Simon Fraser University, Vancouver, B.C., Canada, School of Computing Science, Vancouver, B.C., Canada. 2005.

[Lon02a] Long, F., Zhang, H., and Feng, D.D. Fundamentals Of Content-based Image retrieval. In D. Feng, W. Siu, & H. Zhang, Multimedia Information Retrieval and Management - Technological Fundamentals And Applications (pp. 1-26). Springer, 2002.

[Osa01a] Osada, R., Funkhouser, T., Chazelle B. and Dobkin, D. Matching 3D Models with Shape Distributions. Proceedings of the International Conference on Shape Modeling & Applications, 2001.

[Pap06a] Papadakis, P., Pratikakis, I., Perantonis, S. and Theoharis, T. A Concrete Radialized Spherical Projection Descriptor for 3d Shape Retrieval. IEEE International Conference on Shape Modeling and Applications (SMI'06), 2006.

[Vra01a] Vranić, D.V. and Saupe, D. 3D Shape Descriptor Based on 3D Fourier Transform. Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001), 2001.

[Vra03a] Vranić, D.V. 3D Model Retrieval. PhD Dissertation, Universität Leipzig, Institut für Informatik, 2003.

[Zha01a] Zhang, C. and Chen, T. Efficient Feature Extraction for 2d/3d Objects in Mesh Representation. International Conference on Image Processing, 2001.

[Zha01b] Zhang, S. and Lu, G. Content-Based Shape Retrieval Using Different Shape Descriptors: A Comparative Study. IEEE ICME. Tokyo, Japan. 2001.