Signatures of 3D Models for Retrieval[1]

George Leifman, Sagi Katz, Ayellet Tal, Ron Meir
Department of Electrical Engineering – Technion

ABSTRACT

This paper examines the problem of searching a database of three-dimensional objects for objects similar to a given object. We present two novel signatures for 3D retrieval. We also introduce several measures that can be used for comparing the quality of signatures. Finally, we describe an experimental study where various signatures are compared, and show that one of the proposed signatures outperforms other signatures discussed in the past.

CR Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.3.8 [**Computer Graphics**]: Applications.

INTRODUCTION

Recent progress in the field of digital data storage, the increase of computing power and the developments of modelling techniques have made the construction of 3D models easier and their storage feasible. As a result, large repositories of digital 3D objects are becoming increasingly common in many fields, including e-commerce, medicine, entertainment, molecular biology, CAD and manufacturing. The need for efficient techniques of shape-based retrieval of 3D models from large databases has arisen.

Usually the problem of content-based retrieval is divided into two sub-problems. First, each object in the database is compactly represented by a *signature*. Second, a *retrieval algorithm* that compares signatures is needed. In this paper we focus on developing signatures and use simple retrieval algorithms.

In the past couple of years, there have been a few papers dealing with of 3D object retrieval. In [1] moments are used as signatures, whereas in [2] shape distributions are used. In [6], a topological matching method which uses Reeb graphs is proposed. In [7] descriptors based on cords, moments and wavelets are described.

[1]This research was supported in part by the Ollendorff foundation.

This paper proposes two novel signatures: *Sphere Projection* and *Octrees*. The *Sphere Projection* method tries to capture the global characteristics of a 3D object by computing the amount of "energy" required to deform it into a sphere. The idea behind the *Octrees* method is hierarchical object representation.

A major question is how to measure the quality of a signature. Typically, each technique is tested on a different database using different criteria, which makes it hard to compare the different signatures. In this paper we propose a few general measures for comparing the quality of signatures. These measures are based on the work done in the field of information retrieval [3, 4].

An important aspect of this paper is a comparative study of signatures. In order to make our comparison comprehensive we have created a database that contains 1068 random objects retrieved from the Internet. We compare our proposed signatures to other signatures discussed in the literature [1, 2].

The rest of this paper is organized as follows. The next section we introduce two novel signatures: *Sphere projection* and *Octrees*. Section 3 presents the quality criteria. Section 4 presents our experimental results. Section 5 concludes this paper.

SIGNATURES

In order to achieve invariance to translation, scale and rotation we normalize the objects prior to the signature generation using moments based method that is described in [1].

Sphere Projection

This method tries to capture the global characteristics of an object by computing the amount of "energy" required to deform it into a predefined 3D shape, which is in our case a sphere whose origin is located in the object's center of mass. In order to calculate this energy we use $E = \int_{dist} \vec{F} \cdot d\vec{r}$, where $\vec{F}$ is the applied force and *dist* is the distance between the enclosing sphere and the object surface. Here we assume that the force is constant along this distance and also constant on the different points on the object's surface. Therefore, the energy is proportional to the average distance between the sphere and the object.

The signature consists of two parts. The first part is a bi-variate function, which represents the minimal distances from the sphere to the object's surface: $D_1(\alpha, \theta) = \min_{o \in O}(\text{dist}((\alpha, \theta, R), o))$, where R is the radius of the enclosing sphere, α and θ are angles of spherical coordinates, O is the set of points on the object surface and *dist* is the Euclidian distance.

The second part of the signature is also a bi-variate function, which represents the object's surface in spherical coordinates, (α, θ, r). For objects having more than one surface point with the same angles α and θ, but a different r, the average r is calculated. If we denote this set of points by $G(\alpha, \theta)$ then

$$D_2(\alpha, \theta) = \frac{\sum_{r \in G(\alpha,\theta)}(R - r)}{| G(\alpha, \theta) |}, \tag{1}$$

where $| G |$ is the size of the set G. Note that we average *(R-r)* instead of r in order to be consistent with the definition of the first part.

An important point to make is that the second part of the signature is vital. Consider a sphere with a cylindrical hole from one pole to another. Had the signature consisted only of the first part, the signature of this object would be almost identical to the signature of the ball with small dents on the poles. However, using both parts leads to different signatures.

In practice, in order to calculate the distances from the enclosing sphere, the sphere's surface is first sampled. The sphere's surface is presented by polar coordinates and the sample points are on

$$(\alpha, \theta, r) = \left(\frac{2\pi(i - 0.5)}{m}, -0.5\pi + \frac{\pi(j - 0.5)}{n}, R\right), \tag{2}$$

where $1 \leq i \leq m, 1 \leq j \leq n, m = 2n$ and R is the radius of the enclosing sphere. The set of points are distributed uniformly over the object's surface. This sampling process produces a 2D mesh, *M*.

The distance between each sample point and the object's surface results in a matrix of distances D. Each matrix entry D^{ij} consists of two distances: the first is the distance from the sphere to the object, defined as follows: $D_1^{ij} = \min_{o \in O}(\text{dist}(M_{ij}, o))$, where O is the set of points on the object's surface and dist is the Euclidian distance.

The second distance is the distance from the object to the sphere, calculated according to the following algorithm. For each point $o \in O$ its polar coordinates, (α, θ, r), are calculated. Then, a sample point on the sphere having the most similar angles α and θ is found. Going over all the points in O produces for each sample point a corresponding set of points, G_{ij}, where each point is described by its distance, r, to the origin (0,0,0). More formally, $D_2^{ij} = \frac{\sum_{r \in G_{ij}}(R-r)}{|G_{ij}|}$. The final signature matrix, D, is calculated by averaging or concatenating D_1 and D_2.

We compare between two signature matrices by transforming them into a vector and then we use Euclidean norm to compare between the vectors.

Octrees

The idea behind the Octrees method is to represent an object hierarchically, so that a coarse to fine comparison can be applied to determine the similarity.

Recall that an Octree is a tree in which every internal node has 8 children. The root of the tree represents the axis aligned bounding box of the object. Each node is recursively divided into 8 equal sub-boxes. A leaf of the Octree represents a sub-space which is either entirely inside or entirely outside the object.

Signature comparison: Let D be the difference between the volumes of the root nodes. It is calculated recursively bottom-up. At each step of the recursion the difference between the filled volumes of every two corresponding nodes (one from each octree) is calculated. Let S be a scalar value between 0 and 1 which measures the similarity between two octrees. For two identical octrees S=1 as the octrees become less similar S decreases towards zero. This is defined by the following equation: $S = (1 - D)R$, where R is a scalar multiplication between the two diagonal vectors of the object bounding boxes, given the vectors from the bounding box center to a corner.

COMPARISON CRITERIA

In this section we define several criteria for measuring the quality of signatures of 3D objects. These criteria are based on the large body of work done in the field of information retrieval [3, 4]. We assume that the database consists of several classes and we expect that given an object, other objects from its class will be retrieved. The criteria we propose are: nearest neighbor, precision / recall, first tier, second tier and cumulated gain-based measurements.

Nearest neighbor: For each object O_i in the database, check whether the second result (assuming that the first result is the object itself) belongs to the same class O_i belongs to. The final result is an average over all the objects in database.

Precision / Recall: A standard measure for classification problems is the *Precision / Recall* measure. Let C be the group of the elements that belong to the same class and S - the group of the retrieved elements. Let us define $\gamma = |C \cap S|$, $\alpha = |S| - \gamma$ and $\beta = |C| - \gamma$.

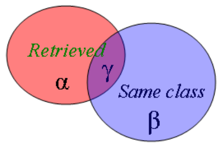


Figure 1: Precision / Recall

Recall is defined as the proportion of the relevant material actually retrieved: $R = \frac{\gamma}{\gamma+\beta}$. Precision is defined as the proportion of the retrieved material actually relevant: $P = \frac{\gamma}{\gamma+\alpha}$. Recall and Precision are calculated for each element and the final result is the average on all results over the elements in the database.

Recall and Precision measure the effectiveness of a classification algorithm. Our problem, however, is slightly different. Retrieval applications just rank the results and order them. Therefore, the number of retrieved elements is not defined. In our tests we assume that the number of retrieved objects is the size of the first screen presented to the user. This assumption is acceptable since we are not interested in the absolute values of P and R but rather want to compare their values for different signatures.

Dissatisfaction with methods of measuring effectiveness by a pair of numbers which may co-vary in a loosely specified way has led to attempts to come up with composite measures. The simplest example of this kind of measure was suggested by Borko [5] $BK = P + R - 1$. Other examples are the *F-Measure* and the *E-Measure*:

$$F = \frac{2}{1/P + 1/R}, \qquad E = \frac{b^2PR + PR}{b^2P + R}, \tag{3}$$

where b measures the relative importance of P or R [3].

First/Second Tier: For each element, test the success (i.e. objects in the same class) percentage for the first k elements, where k is the size of the class the query element belongs to. The final result is the average over all the objects in the database. Second Tier criterion is similar to the First Tier, but works with $k = 2*$*(size of the element's class)* for each element.

Cumulated gain-based measurements: The list of ranked retrieved objects are turned into a gained value list by replacing objects' IDs by their relevance values [4]. We assume that the relevance values are either 0 or 1 (1 - the object is in the same class, 0 - in a different class). For example: $G = \langle 1, 1, 0, 0, 1 \rangle$.

The cumulated gain is defined as follows. Denote the value of the i^{th} position in the gain vector G by G_i. The cumulated gain vector CG is defined recursively by:

$$CG_i = \begin{cases} G_1, & i = 1 \\ CG_{i-1} + G_i, & otherwise. \end{cases} \tag{4}$$

The cumulated gain vector with a discount factor, DCG, is defined recursively by:

$$DCG_i = \begin{cases} G_1, & i = 1 \\ DCG_{i-1} + G_i / \lg_2 i, & otherwise. \end{cases} \tag{5}$$

The length of the vectors $|CG|$ and $|DCG|$ is the number of retrieved objects in each query.

After summing and normalizing $|DCG|$ by the best possible result we get for each query object:

$$F_{DCG} = \frac{DCG_k}{1 + \sum_{j=2}^{|C_l|} \frac{1}{\lg_2(j)}}, \tag{6}$$

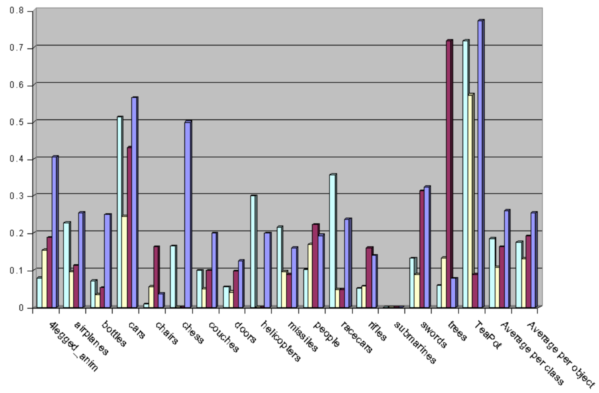
where $k = |DCG|$ and $|C_l|$ is the size of the l^{th} class. The overall result is the average over all objects. The value of each correct result is estimated according to its position in the search. Objects located further down the list are less relevant. The assumption behind it is that users are "impatient" and are less likely to examine results far down the list.

EXPERIMENTAL RESULTS

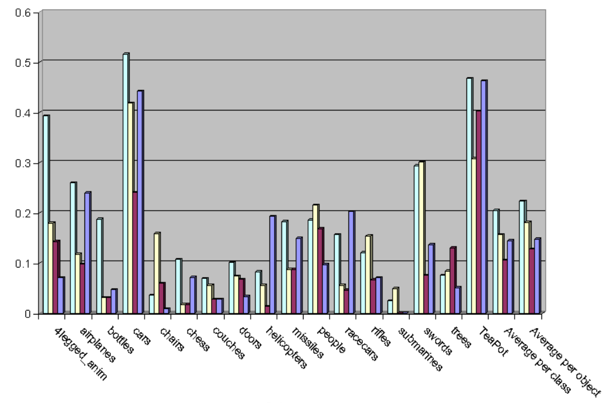
This section describes our comparative study. We have created a database containing 1068 random objects retrieved from the Internet. In this database, 258 objects were manually categorized into 17 different classes. Each of the classes contains objects that are geometrically similar, e.g. 4-legged animals (20 objects), planes (19 objects), people (24 objects), etc. The rest of the objects were not classified and thus were not used as query objects but serve as "background noise".

Four signatures were compared using the five measures described above. The first signature is a vector of *shape moments* [1]. For object D and its surface ∂D the (p,q,r)-th moment is given by: $m_{pqr} = \int_{\partial D} x^p y^q z^r dxdydz$. The second signature is *shape distribution* [2] where an object is represented as a probability distribution sampled from a shape function measuring global geometric properties, i.e. the distribution of Euclidean distances between pairs of randomly selected points on the surface of a 3D model. The third and the fourth signatures are *Sphere projection* and *Octrees* described in the second section.

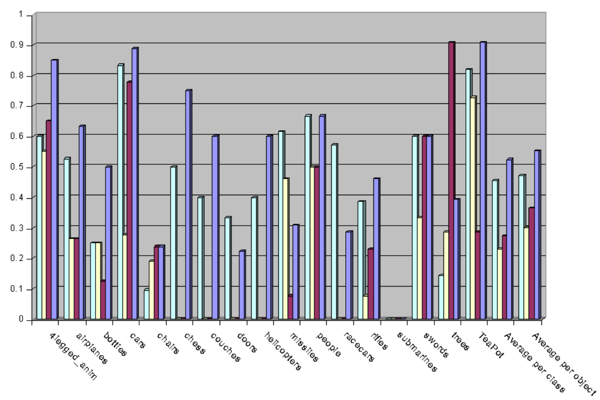
For example, Figure 3 shows the objects contained in the *4-legged animals* class (20 objects) after normalization. Object 10 (a cat) was used as a query object. Figure 4 shows the results of the retrieval. The best results are achieved using the *Sphere* signature where 8 out of 9 (closest) objects retrieved belong to the class. The graphs in Figure 2 show the results of our benchmarks. The first 16×4 bars show the results for each class and the last 2×4 bars show the averages. The *average per class* is the average of all the objects within each class and then the average over all the classes, ignoring the different sizes of the classes. The *average*



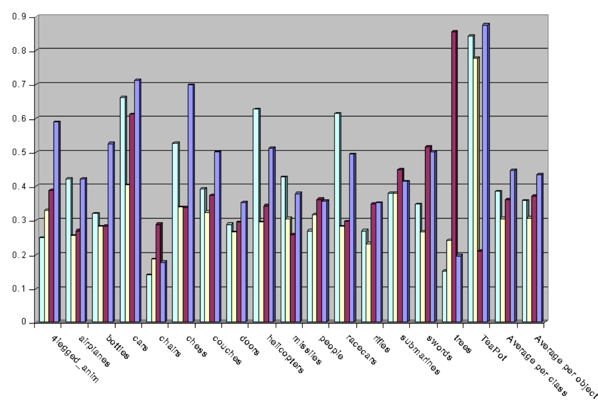
(a)



(b)



(c)



(d)

Figure 2: Performance comparison: (a) 1st Tier, (b) F-Measure, (c) Nearest Neighbor, (d) CGain. Legend: cyan - ocrtrees, yellow - moments, red - probabilities, blue - spheres

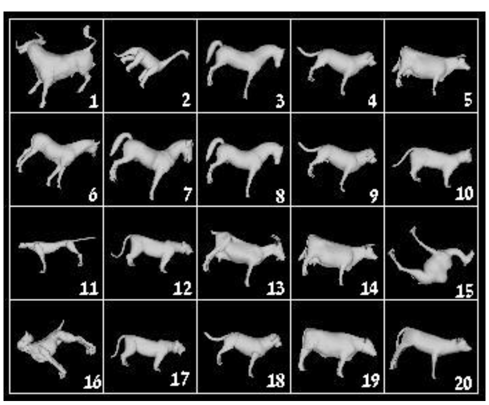


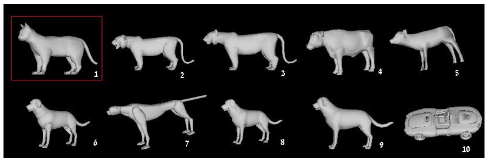
Figure 3: Objects in the *4-legged animals* class after automatic normalization

per object takes into account the different sizes of the classes.

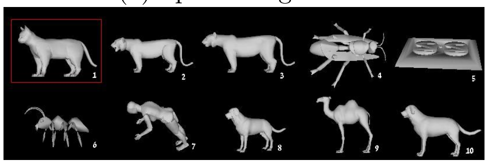
Note that for most classes, the *Sphere* signature gives the best results. However for some classes such as *people*, *trees* and *chairs* the *Probability* signature gives better results. The reason for this is that the objects in these classes do not have a common global shape but rather similar local properties.

To test the robustness to level of details of the signatures we used the *Teapots* class, which contains 11 different teapots, out of which 8 are full teapots with different number of faces (256–40000) and 3 are partial teapots (w/o handle, w/o lid and body only) (Fig. 5). Comparing the performance of the algorithms according to the E-Measure and the F-Measure, the most robust algorithms are the *Sphere* and the *Octree*, while the most sensitive algorithm is the *Probabilities* (Fig. 6). The reason for these results is that the Spheres and Octrees capture the global shape of the teapot while the other signatures are based on local characteristics such as angles and curvature.

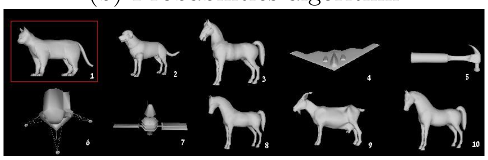
Time and space complexity: Table 1 compares the signature size and the time to generate the signatures. The algorithms were ran on Pentium 4 1.6Ghz. The time to compare a pair of signatures is similar for all the retrieval algorithms with the exception of the *Octree* which is slower. This is so because



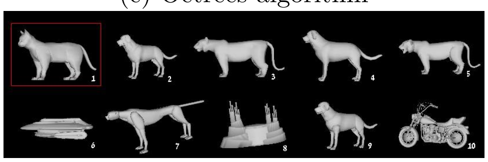
(a) Spheres algorithm



(b) Probabilities algorithm



(c) Octrees algorithm



(d) Moments algorithm

Figure 4: Example of retrieval for 4 signatures

Figure 5: Various Teapots

in this algorithm two trees are compared recursively, while the other algorithms simply compare vectors.

CONCLUSIONS

This paper has introduced a couple of novel signatures of 3D objects for content-based retrieval: a *Sphere projection* and an *Octree*. It has also defined a set of measures that let us compare the quality of various signatures. Finally, the results of a comparative study between signatures were presented. The *Sphere* signature was shown to be better than the other signatures.

In the future we intend to run the methods on larger databases. We also intend to add to the study other algorithms, such as the Reeb graph [6]. Finally, we are currently working on augmenting *Sphere* signatures with new relevance feedback schemes.

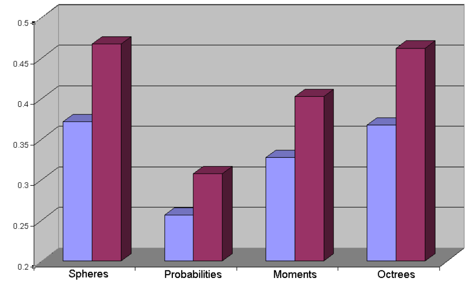


Figure 6: Robustness comparison: blue - E-Measure, red - F-Measure

Algorithm	Signature size	Signature generation time
Spheres	2.3k	1.3sec
Probability	1.3k	1.8sec
Octrees	8k	1.8sec
Moments	0.5k	0.7sec

Table 1: Time and space complexity

REFERENCES

[1] M. Elad, A. Tal, S. Ar, "Content Based Retrieval of VRML Objects - An Iterative and Interactive Approach", EG Multimedia, Sep. 2001, 97-108.

[2] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, "Matching 3D Models with Shape Distributions", International Conference on Shape Modeling and Applications, 154-166, 2001.

[3] C.J. Keith van Rijsbergen, "Information retrieval", London: Butterworths, 1975.

[4] K. Jarvelin, J. Kekalainen, "IR evaluation methods for retrieving highly relevant documents", Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2000.

[5] H. Borko and M. Bernick. "Automatic document classification", J. of the ACM 9, 512-521, 1962.

[6] M. Hilaga, Y. Shinagawa, T. Kohmura, and TL Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes", SIGGRAPH, 203-212, 2001.

[7] E. Paquet, A. Murching, T. Naveen, A. Tabatabai and M. Rioux, "Description of Shape Information for 2-D and 3-D Objects", Signal Processing: Image Communication: 103-122, 2000.

[8] H.Y. Shum, M. Helbert, K. Ikeuchi, "On 3D Shape Similarity", Proceedings of IEEE Comuter Vision and Pattern Recognition, 526-531, 1996.

[9] Y. Sako and K. "Fujimura, Shape similarity using homotopic deformation", The Visual Computer 16, 47-61, 2000.