CCIT Report #418 March 2003

Geometric and Topological Signatures for 3D Retrieval [1]

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

Abstract

This paper examines the problem of retrieving from a database of three-dimensional objects the most similar objects to a given object. We present two novel geometric signatures for 3D retrieval. We also show how to enrich these signatures with a topological signature. Finally, we describe an experimental study comparing the quality of various signatures using several estimation measures, and draw conclusions.

CR Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling; I.3.8 [**Computer Graphics**]: Applications; H.3.3 [**Information Systems**]: Information Search and Retrieval

1 Introduction

Large repositories of digital 3D objects have become increasingly common in many fields, including e-commerce, medicine, entertainment, molecular biology, CAD and manufacturing. The existence of large databases creates a need for efficient techniques of shape-based retrieval of 3D models.

Shape-based retrieval is usually done in two steps. First, each object in the database is compactly represented by a *signature*. Second, a *retrieval algorithm* compares signatures and ranks objects according to the similarity of their signatures. In this paper we focus on developing representative signatures.

In the last few years, several papers dealing with 3D object retrieval appeared. In [4] a vector of preset order shape moments is used as a signature. In [12] an object is represented by a probability distribution sampled from a shape function measuring global geometric properties, such as the distribution of Euclidean distances between pairs of randomly selected surface points. In [6], a topological matching method is proposed, where a multi-resolution Reeb Graph is computed and considered a signature. In [13] descriptors based on cords, moments and wavelets are described. In [16] spherical harmonics are used, where a spherical function is sampled in many points, yet the map is characterized by a few parameters. In [9] a shape descriptor that represents a measure of reflective symmetry for an arbitrary 3D voxel model for all planes through the model's center of mass, is proposed.

This paper proposes two novel geometric signatures: a *sphere projection* signature and an *Octree* signature. The sphere projection signature attempts to capture the global characteristics of a 3D object by computing the amount of "energy" required to deform it into a pre-defined shape. The Octree signature provides a hierarchical representation of an object. The paper also shows how to enrich the above geometric signatures with a topological signature.

An important issue is how to assess the quality of signatures. Typically, each technique is tested on different databases using different criteria, which makes it hard to compare signatures. In this paper we discuss a few general measures for evaluating signatures. These measures are based on the work done in the field of information retrieval [1, 2, 8, 15].

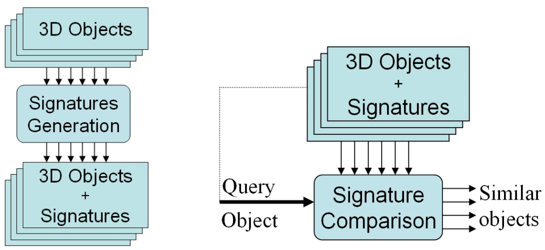
Finally, we describe a comparative study we conducted. We collected from the Internet a database containing 1850 objects. Our proposed signatures were compared to other signatures discussed in the literature [4, 12] using various estimation measures.

The rest of this paper is organized as follows. Section 2 introduces our novel signatures and shows how to enrich them with topological properties. Section 3 describes methods for achieving invariances to resampling, simplification and rigid deformations. Section 4 presents retrieval performance evaluation techniques. Section 5 presents our experimental study. Finally, Section 6 concludes this paper.

[1] This research was supported in part by the Ollendorff foundation and by The Israeli Ministry of Science grant no. 01-01-01509.

2 Signatures

The goal of shape-based object retrieval is to rank objects according to their similarity to a given object. To make this scheme feasible for large databases, instead of comparing the objects themselves, *signatures* associated with each object are compared. Signatures are generated off-line, during a preprocessing step, while a retrieval query is processed on-line, as illustrated in Figure 1.



(a) Preprocessing (b) On-line retrieval query

Figure 1: 3D Object retrieval using signatures

A signature should be a compact representation of an object which suffices to uniquely identify it, yet reflects similarities and dissimilarities between objects.

We assume that objects are given in VRML or in any other format that represents a surfaces of an object by a set of vertices and a set of planar polygonal faces embedded in three dimensions.

We propose below two novel signatures : a *sphere projection* signature and an *Octree* signature. Then we show how to enrich the above geometric signatures with a topological signature.

2.1 Sphere Projection Signature

The *sphere projection* signature attempts to capture the global characteristics of the object by computing the amount of "energy" required to deform it into a predefined three-dimensional shape, in our case a sphere.

Let $\vec{F}$ be the applied force and let *dist* be the distance between the enclosing sphere and the object surface. The energy required to deform an object is given by

$$E = \int_{dist} \vec{F} \cdot d\vec{r}. \tag{1}$$

We assume that the force is constant along this distance and is also constant for all the points on the object's surface. Therefore, the energy is proportional to the average distance between the sphere and the object.

We define the sphere projection signature as a concatenation of three sub-signatures: the distance from the sphere to the object D_1, the distance from the object to the sphere D_2 and the variance of radii D_3.

The first distance, D_1, is a bi-variate function which represents the minimal distance from the enclosing sphere to the object's surface. Let R be the radius of the enclosing sphere, (θ, ϕ, R) be spherical coordinates, $P_{(\theta,\phi,R)}$ be a point on the enclosing sphere, and O be the set of points on the object's surface. Then:

$$D_1(\theta, \phi) = \min_{o \in O}(||P_{(\theta,\phi,R)} - o||). \tag{2}$$

D_1 is not sufficient for describing non star-shaped objects. Consider, for instance, a sphere with a cylindrical hole from one pole to another. Had the signature consisted only of D_1, the signature of this object would be identical to the signature of the ball with dents on the poles.

To solve this problem, we consider D_2, a bi-variate function which represents the distance to a sphere. We denote the set of object points having the same spherical coordinates (θ, ϕ) by $G(\theta, \phi)$, and the size of $G(\theta, \phi)$ by $|G(\theta, \phi)|$. Then

$$D_2(\theta, \phi) = \frac{\sum_{r \in G(\theta,\phi)}(R - r)}{|G(\theta, \phi)|}, \tag{3}$$

where r is the radius of a point in $G(\theta, \phi)$. If the size of $G(\theta, \phi)$ is infinite, the sum is replaced by an integral.

In practice, in order to calculate the distances, the sphere's surface is first sampled, producing a 2D mesh, M, of points distributed on the sphere's surface, as illustrated in Figure 2. Let R be the radius of the sphere and m and n be parameters defining the size of the signature, then for $1 \le i \le m$, $1 \le j \le n$, the ij^{th} sample point is defined as:

$$\left(\frac{2\pi(i - 0.5)}{m}, -0.5\pi + \frac{\pi(j - 0.5)}{n}, R\right). \tag{4}$$

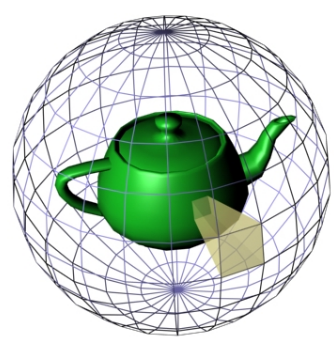


Figure 2: Sampling the sphere

Next, a set of points O, distributed uniformly over the object's surface, is drawn. The number of points drawn from each object's face is proportional to its relative surface area.

The entries of the distance matrix D_1 are defined by:

$$D_1^{ij} = \min_{o \in O}(||M_{ij} - o||). \quad (5)$$

Similarly, for each sampled point $o = (\theta, \phi, r) \in O$, a sample point on the sphere having the most similar angles θ and ϕ is found. Thus, for each sphere sample point, a corresponding set of object points, G_{ij}, is produced. The entries of the distance matrix D_2 are then defined by:

$$D_2^{ij} = \frac{\sum_{r \in G_{ij}}(R - r)}{\mid G_{ij} \mid}. \quad (6)$$

Finally, D_3, the variance of radii, is calculated. While D1 and D2 describe either an average or an extreme property of a spherical angle, D3 describes the local properties of spherical angles. It represents the similarity (or deviation) within a specific angle to the surface of a sphere. Let m_{ij} be the mean of the radii, r, of the points in the set G_{ij}. The entries of the signature matrix D_3 are defined by:

$$D_3^{ij} = \frac{\sum_{r \in G_{ij}}(r - m_{ij})^2}{|G_{ij}|}, \quad (7)$$

The final signature of a given object is defined as a matrix D whose entries D^{ij} each consists of a concatenation of three entries: D_1^{ij}, D_2^{ij} and D_3^{ij}.

Given two signatures, a simple way to compare them is to reorder each signature matrix into a vector and use an L_2 metric (i.e. a Euclidean distance). The advantages of an L_2 metric is its simplicity and its low complexity. Its drawback is that it does not take into consideration the positions of the signature entries. To overcome this limitation a *Quadratic Form Distance* [5, 7] function or an *Earth Mover's Distance* [14] algorithm can be used. Our experimentations revealed, however, that the latter do not achieve drastic improvements in the retrieval results, yet increase the retrieval running time considerably.

2.2 Octree Signature

An Octree is a common way to represent three-dimensional objects. The key idea behind the Octrees signature is to represent an object hierarchically, so that a coarse-to-fine comparison can be applied to determine similarity.

Recall that in an Octree, the root of the tree represents the axis aligned bounding box of the object. Each node is recursively divided into eight equal sub-boxes, until the whole sub-space of the box is either entirely inside or entirely outside the object.

To determine the similarity between Octree signatures, the volumes of the nodes are compared 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 V_i be an Octree internal node, having eight children V_{ij}, $1 \leq j \leq 8$. The *unweighted volume difference* ΔV_i is defined as $\Delta V_i = \Sigma_{j=1}^{8}|\Delta V_{ij}|$. For instance, for the Quadtree in Figure 3, $\Delta V_1 = \Delta V_2 = 0$, $|\Delta V_3| = |\Delta V_4| = \frac{1}{4}$, thus the difference between the parents is defined as $\Delta V = |\Delta V_1| + |\Delta V_2| + |\Delta V_3| + |\Delta V_4| = \frac{1}{2}$.

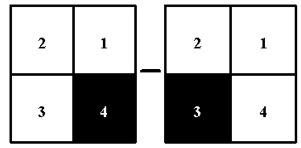
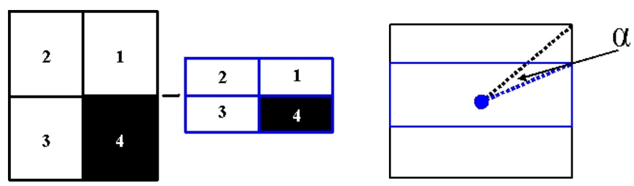


Figure 3: Volume difference calculation

To distinguish between objects having a similar Octree structure but different bounding boxes we define S, the similarity between Octrees nodes to be a function of the weighted volume difference:

$$S = (1 - \triangle V)\cos\alpha, \quad (8)$$

where α is an angle between the two diagonal vectors of the objects' bounding boxes, originating from the box centers to a corners, as illustrated in Figure 4. Note that two identical Octrees have $S = 1$ whereas as the Octrees become less similar, S decreases towards zero.



(a) Unwighted volume difference $\Delta V = 0$ (b) Similarity $S = (1 - \Delta V)\cos\alpha$

Figure 4: Nodes having the same structure but different bounding boxes

To decrease the search time, the comparison of Octree signatures is performed in a coarse-to-fine manner. During the initial search, over the whole database, only a few levels of the Octree are compared. Then, a more precise search, using more levels, is applied to the set of most similar objects of the coarse search.

The main disadvantage of the Octree signature is that even when only a few levels are considered, generating the signature takes a relatively long time and the storage needed is relatively high (as will be discussed in Section 5). Another disadvantage is the difficulty in applying *relevance feedback* techniques which require the signatures to be represented as feature vectors.

2.3 Topological Signature

We describe how to enrich geometric signatures by including topological properties. Given an object in $\mathbb{R}^3$, its Betti numbers provide some important properties [3, 10]. Betti zero, β_0, is the number of connected components; Betti one, β_1, is the number of independent tunnels; Betti two, β_2, is the number of closed regions in space.

We assume that each object consists of one or more two-manifold components. To find the number of connected components β_0, a BFS or a DFS is applied to the *dual graph* of the object. In the dual graph, each vertex represents a face in the model, and there is an arc between two vertices if their corresponding faces are adjacent. Next, each connected component is tested for closedness (β_2) by checking that each triangular face has three neighbors. To calculate β_1, the Euler number, χ, is used.

$$\chi = \beta_0 - \beta_1 + \beta_2, \tag{9}$$

or equivalently:

$$\chi = v - e + f, \tag{10}$$

where v, e, f are the number of vertices, edges and faces respectively. Thus β_1 is given by:

$$\beta_1 = \beta_0 + \beta_2 - \chi = \beta_0 + \beta_2 - v + e - f. \tag{11}$$

The simplest way to compare Betti number signature vectors is to use either an L_1 or an L_2 metric. The problem, however, is that there is a substantial difference between the comparison of small and large Betti numbers. For instance, the difference between 2 and 3 connected components is more significant than the difference between 3298 and 3299. Thus, a logarithm function, which is a slowly increasing function, is used.

Figure 5 compares the results of two queries when using only the sphere projections signature, only the topological signature, and a combination of the two. For a calf, the geometric signature achieves good results, but not the topological signature. This is because all 4-legged animals are similar geometrically, but there are many other objects resembling the calf topologically. Trees, however, are not necessarily similar geometrically. Yet, they are usually modeled similarly by designing one element (e.g., a leaf) and copying it multiple times, thus they all have a large number of components (branches, leafs). Therefore, in this case the topological signature achieves good results. In both examples, the best results are achieved when a combination of topological and geometric signatures is used.

Often, Betti numbers can also help for retrieving objects which are non-rigid deformations of each other, such as people in different motions. This is so because these objects are modeled using the same prototype and changing the position and orientation of different parts. Thus, they have the same Betti numbers.

3 Desirable Properties

This section discusses how to achieve invariance to resampling, simplification and rigid transformations. To accomplish invariance to resampling and simplification, points are distributed uniformly on the surface of the model and the signatures are computed for these points. This is done as follows.

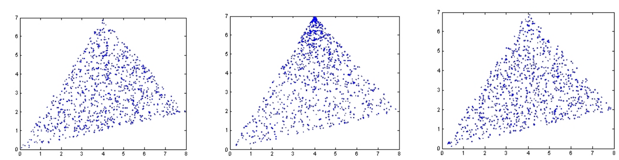
The area of each triangle is first stored in an array along with the cumulative area of triangles visited so far. Next, a triangle is drawn with probability proportional to its area. Finally, a point on this triangle is drawn.

Let (A, B, C) denote the vertices (A,B,C) of the triangle. In [12] it is proposed to generate two random numbers, $r1$ and $r2$, $0 \leq r1, r2 \leq 1$, and to generate a point by evaluating the following equation:

$$P = (1 - \sqrt{r_1})A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C. \tag{12}$$

Another alternative is to use Barycentric coordinates. A random number s, $0 \leq s \leq 1$, is drawn followed by drawing a random number t, $0 \leq t \leq (1 - s)$. The point is defined as $P = A + (B - A)s + (C - A)t$.

We implemented the following method which is less costly than [12], yet distributes the points uniformly, as demonstrated in Figure 6. Two random numbers, s and t, $0 \leq s, t \leq 1$, are drawn. A point above the line $l : s = 1 - t$ is reflected about l, i.e. $s \leftarrow 1-t$, $t \leftarrow 1-s$, for all $s+t>1$.



(a) Osada et al (b) Barycentric (c) Our method

Figure 6: Distributing 1000 points

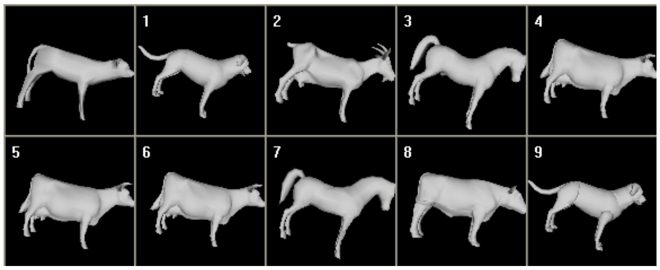
To achieve invariance to rigid transformations, each object is normalized prior to signature computation. This is done using surface moments [4]. The (p, q, r)-th moment is defined as

$$m_{pqr} = \int_{\partial D} x^p y^q z^r \, dxdydz, \tag{13}$$

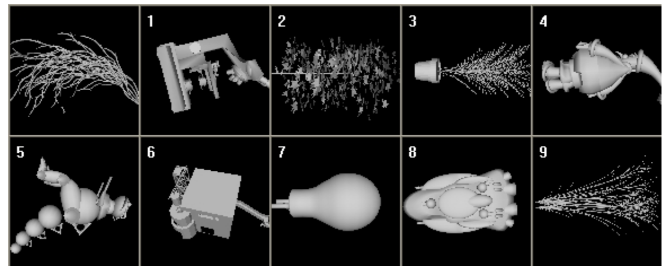
where ∂D is the object's surface. Instead of analytically evaluating this integral, it is approximated by uniformly distributing points over the object's surface, as described above. Given this set of points, $\{x_i, y_i, x_i\}_{i=1}^{N}$, the (p, q, r)-th moment is approximated by:

$$\widehat{m}_{pqr} = \frac{1}{N} \sum_{i=1}^{N} x_i^p y_i^q z_i^r \tag{14}$$

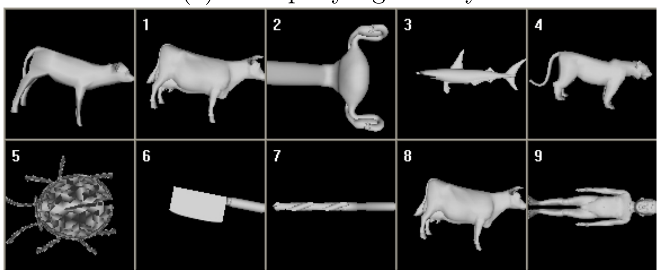
The first order moments $m_{100}, m_{010}, m_{001}$ represent the object's center of mass. Consequently, subtracting



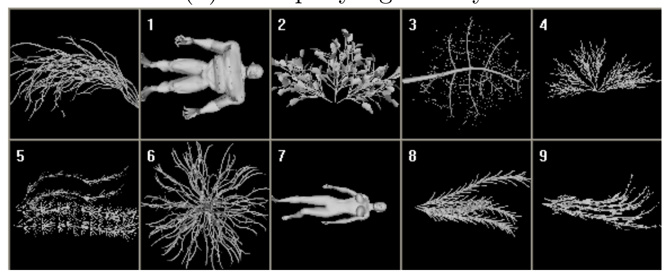
(a) Calf query - geometry



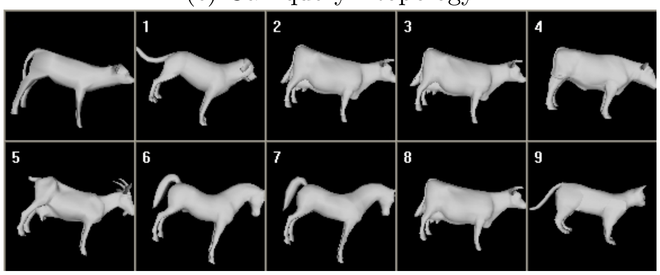
(b) Tree query - geometry



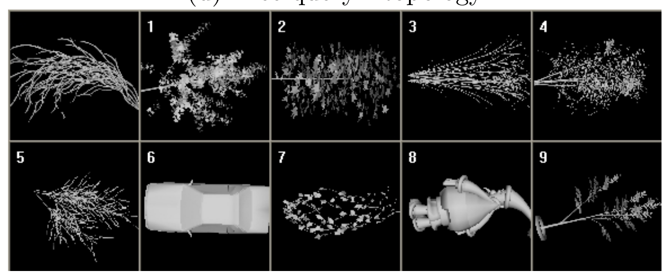
(c) Calf query - topology



(d) Tree query - topology



(e) Calf query - geometry + topology

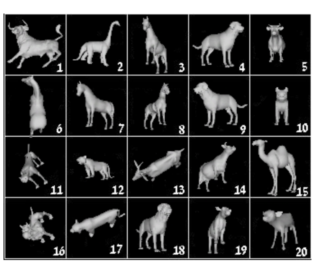


(f) Tree query - geometry + topology

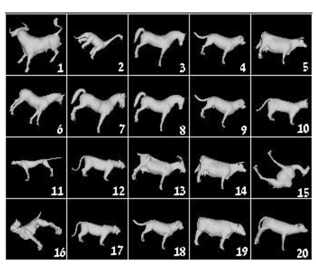
Figure 5: Queries using geometric signatures, topological signatures and combinations

them from each point achieves invariance to translation. After applying a Singular Value Decomposition, $U\nabla U^T = SVD(\widehat{M})$ on the second order moment matrix, $\widehat{M}$, the orthogonal matrix U, represents the rotation and the diagonal matrix ∇ represents the scale in each axis, ordered in decreasing order. Multiplying each surface point by U rotates the object to its canonical position and dividing each surface point by $\nabla_{(1,1)}$ re-scales the object.

As shown in Figure 7, most 4-legged animals are normalized in the same manner.



(a) Before normalization



(b) After normalization

Figure 7: Object Normalization: 4-legged animals

4 Performance Evaluation

A fundamental issue in *Information Retrieval (IR)* is the performance evaluation of retrieval algorithms. An analytical evaluation is difficult since relevancy cannot be described mathematically. Instead, benchmarks are used to evaluate system effectiveness. We describe several evaluation criteria common in information retrieval [1, 2, 8, 15].

We assume that the database consists of several classes and we expect that given an object, other objects belonging to the same class, will be retrieved. The criteria we propose are: nearest neighbor, Precision/Recall based measurements, first tier, second tier and cumulated gain based measurements.

1. Nearest Neighbor [6, 12]: Check whether the second ranked result (assuming the first result is the object itself) belongs to the same class as the query object. The final result is an average over all the object queries.

2. Precision/Recall Based Measurements [1, 11]: Let C be the set of objects that belong to the same class as the query, S be the set of all retrieved objects and $I = C \cap S$ be the set of retrieved objects that belong to the same class as the query, as illustrated in Figure 8.

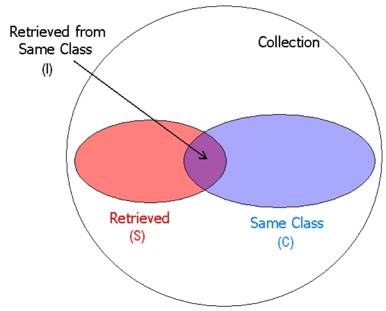


Figure 8: Recall/Precision

Recall is defined as the proportion of the relevant material actually retrieved: $R = \frac{|I|}{|C|}$. *Precision* is defined as the proportion of the retrieved material actually relevant: $P = \frac{|I|}{|S|}$. Recall measures the ability of the search to find all the relevant objects in the database, while precision measures the ability to retrieve top-ranked objects that are mostly relevant.

Unlike classification, for which Recall and Precision were originally defined, the goal of retrieval is to rank the results. Therefore, the number of retrieved objects $|S|$ is not defined a-priori. In our experiments 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 interested not in the absolute Recall/Precision values, but rather in comparing their values for different signatures.

Dissatisfaction with measuring effectiveness by a pair of numbers which may co-vary in a loosely specified way has led to attempts to define composite measures. A common way to do this is to use the harmonic mean *F-Measure* [15]:

$$F = \frac{2PR}{P+R} = \frac{2}{1/P + 1/R}. \tag{15}$$

Unlike the arithmetic mean, the harmonic mean requires both recall and precision values to be high in order for the mean to be high.

The *E-Measure*, a variant of the F-measure, allows different weighting of precision and recall [15]:

$$E = \frac{b^2 PR + PR}{b^2 P + R} = \frac{1+b^2}{b^2/R + 1/P}, \tag{16}$$

where b measures the relative importance of P or R. When $b = 1$ precision and recall weigh equally ($E = F$), when $b > 1$ precision weighs more and when $b < 1$ recall weighs more.

3. First/Second Tier [6, 17]: The *first tier* is the success (i.e. objects in the same class) percentage among the first k retrieved objects, where k is the size of the class the query object belongs to. The *second tier* criterion is similar, but $k = 2*$*(size of the object's class)*. The final result is the average over all the queries.

4. Cumulated Gain Based Measurements: The list of ranked retrieved objects is turned into a *gained value list* by replacing objects' IDs by their relevance values [8].

Denote the value of the i^{th} position in the gain vector G by G_i. $G_i = 1$ if the objects are in the same class as the query object, whereas $G_i = 0$ otherwise. The *cumulated gain* vector CG is defined recursively by:

$$CG_i = \begin{cases} G_1 & i = 1 \\ CG_{i-1} + G_i & \text{otherwise.} \end{cases} \tag{17}$$

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

$$DCG_i = \begin{cases} G_1 & i = 1 \\ DCG_{i-1} + G_i / \log_2 i & \text{otherwise.} \end{cases} \tag{18}$$

Here, objects located further down the list are considered less relevant. This is done in order to accommodate for users who are typically "impatient" and are less likely to examine results far down the list.

Comparing two DCG curves is not always convenient, since the curves may intersect each other. To remedy this problem, we assume that the number of retrieved objects equals the size of the first screen. Then, the values of a predefined entry of CG or DCG vectors are compared. Measuring the algorithm's performance with a single value for each query is done by normalizing by the best possible result:

$$\overline{DCG} = \frac{DCG_k}{1 + \sum_{j=2}^{|C|} \frac{1}{\log_2(j)}}\ , \tag{19}$$

where k is the number of retrieved objects and $|\,C\,|$ is the size of the class the query belongs to. The overall result is the average over all queries.

5 Experimental Results

This section describes our comparative study. We collected a database containing 1850 objects from the Internet. A subset of the database (725 objects) was classified into 25 different classes, each containing similar objects. The remaining objects were not classified and thus were not used as query objects. Table 1 describes the classes of the objects.

We compare four signatures: (1) our sphere projection signature used jointly with the topological signature, (2) the Octree signature, (3) shape moments [4]

Class	Size	Class	Size
4-legged animals	32	Airplanes	95
Bottles	15	Cars	57
Chairs	50	Chess	12
Couches	18	Doors	11
Faces	6	Glasses	5
Guitars (violins)	17	Helicopters	12
Knifes (swords)	38	Missiles	24
People	78	Plants	47
Race cars	20	Rifles	28
Space Ships	55	Submarines	10
Tanks	12	Teapots	11
Trees	53	Vases	9
Zeppelins	10		
		Non-Classified	1125
		TOTAL	1850

Table 1: Database organization

and (4) shape distributions [12]. The moments signature is a vector of the (p,q,r)-th moments. The shape distributions signature represents object by a probability distribution sampled from a shape function such as the distribution of Euclidean distances between pairs of randomly selected surface points.

We used 72 bins for each distance in the sphere projection signature. Thus the length of this signature is $72 \times 3 = 216$. The height of the Octree signature we used is 4. In the shape distributions signature, the number of bins for each histogram for each measure is 100. The maximum moment order in the moments signature is 6.

Figure 9 shows some retrieval results. Visually, the sphere projection signature outperforms the other signatures. For instance, nine 4-legged animals were retrieved among the top ten using the sphere projection signature, while only three, four and six were retrieved using the Octree, shape distribution and moments, respectively.

To compare the average performance, each object from the above classes was used as a query object. A retrieved object is considered relevant, if it belongs to the same class of the query object.

Our results, averaged over all queries, are displayed in Figure 10 where the evaluation measurements discussed in Section 4 are used. The graphs in Figure 10(a)-(d) show the average performance for 9 classes. The first 9×4 bars show the results for each class and the last 4 bars show the total average of all the objects within all 25 classes.

These results indicate that the sphere projection signature performs better than the other signatures. The Octree signature fails to describe some classes, e.g., animals or trees, but overall, it fares reasonably well. Moreover, these figures show that there is a very high correlation between different evaluation criteria.

Figure 11 illustrates the average performance of the topological, the geometric and the combined signatures according to the Discount Cumulated Gain (DCG) measurement. Seven classes are presented, where the first five have a similar geometric structure and the last two (plants and trees) do not. Using a combined signature achieves a high performance for all the classes. Averaging over all the queries (right column) shows that the combined signature is more effective for a wide range of queries.

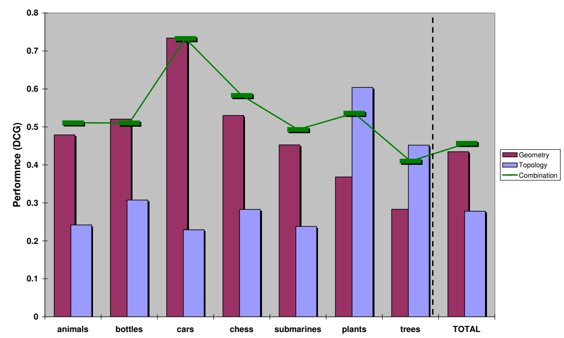
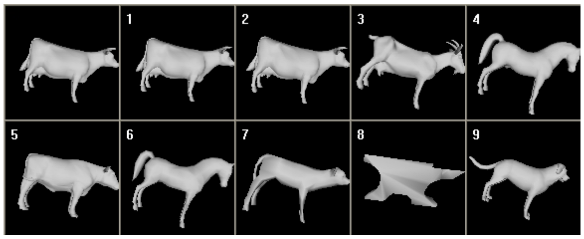


Figure 11: Average performance using topological, geometric and combined signatures

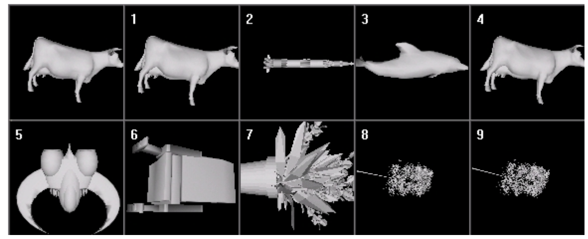
To illustrate the robustness to level of detail, consider the Teapots class (Figure 12), which contains 11 different teapots, out of which 8 are full teapots with different number of faces (256–40000) and 3 are partial teapots (lacking a handle, lacking a lid and containing the body only). Figure 13 compares the performance of the signatures, where it is shown that the most robust signature is the sphere projection. A possible explanation is that this signature captures the global shape of the teapot, and thus fares better than signatures based on local characteristics.

Figure 12: The Teapot class

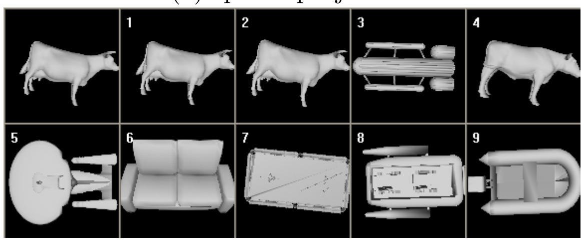
Finally, Table 2 compares the signature size and the time needed to generate the signatures. The results are



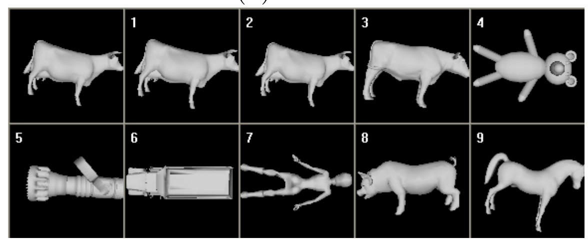
(a) sphere projection



(b) Octree

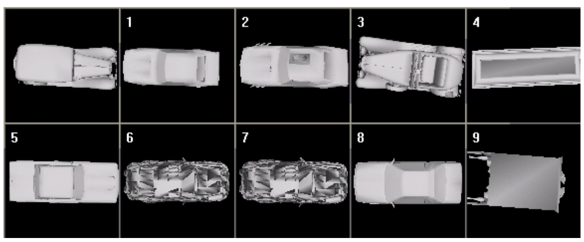


(c) shape distribution

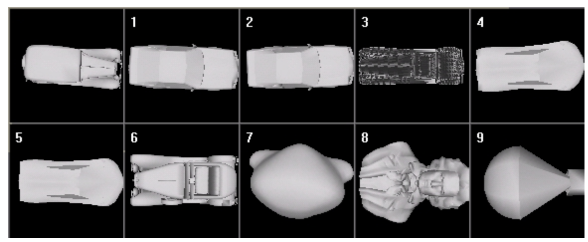


(d) moments

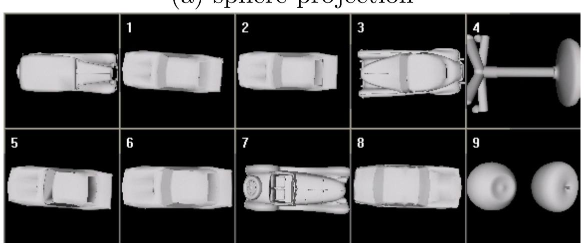
(1) Query object – Cow



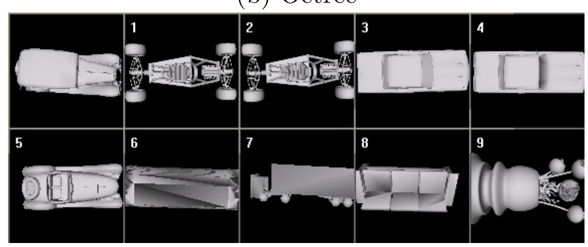
(a) sphere projection



(b) Octree

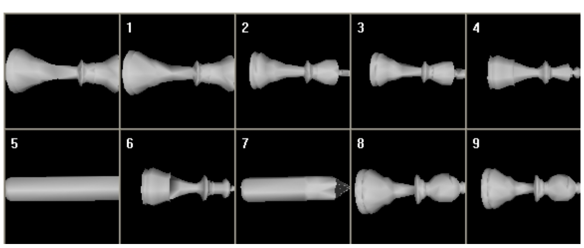


(c) shape distribution

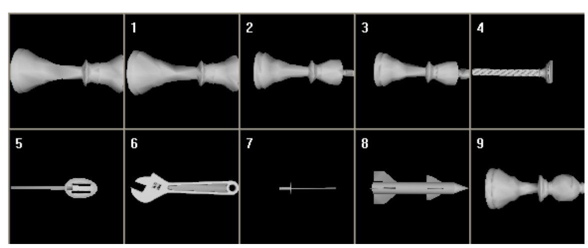


(d) moments

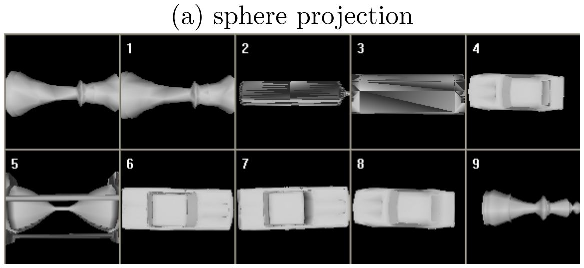
(2) Query object – Car



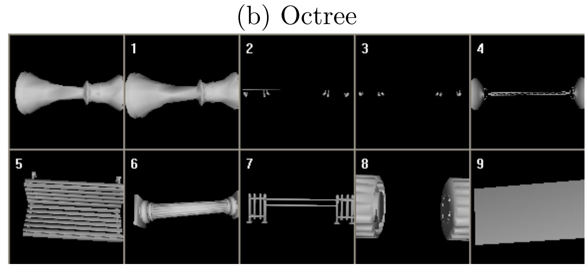
(a) sphere projection



(b) Octree



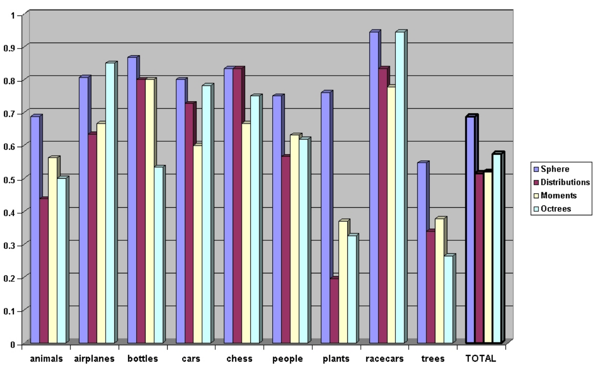
(c) shape distribution



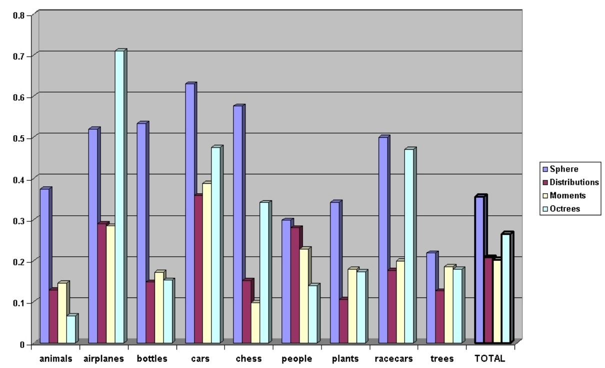
(d) moments

(3) Query object – Chess

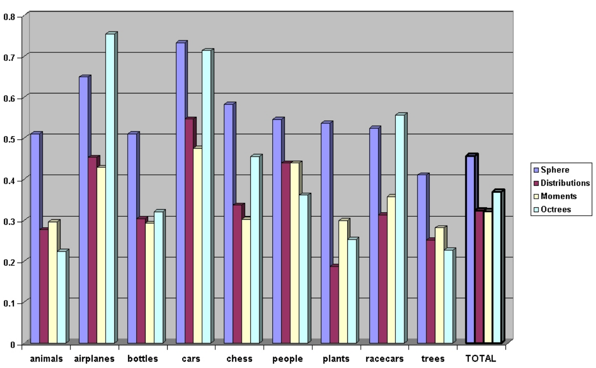
Figure 9: Retrieval results – the query object is at the left upper corner



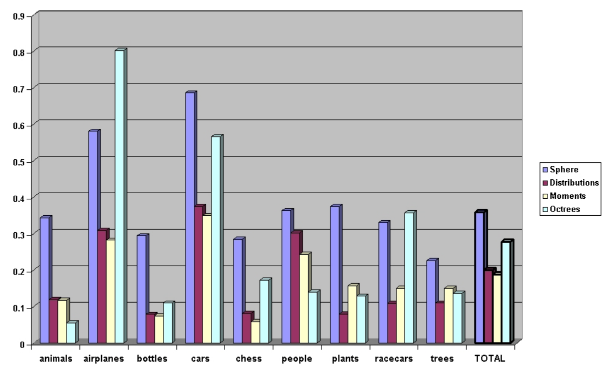
(a) Nearest Neighbor



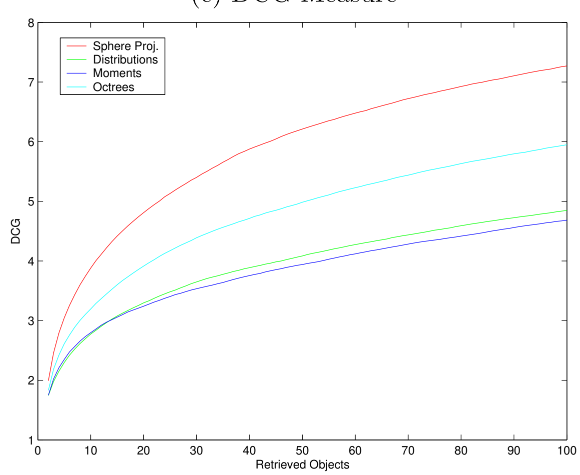
(b) 2^{nd} Tier



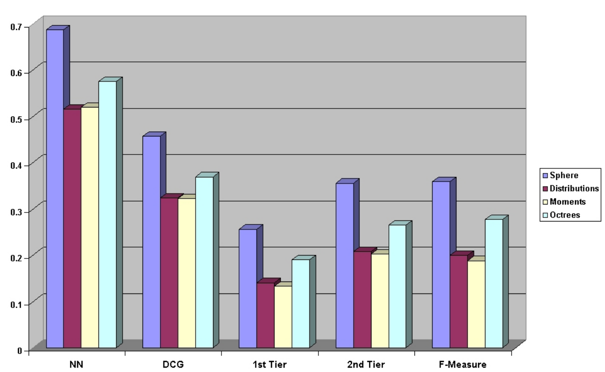
(c) DCG Measure



(d) F-Measure



(e) DCG curves



(f) Average on the five criteria

Figure 10: Average performance using various evaluation criteria

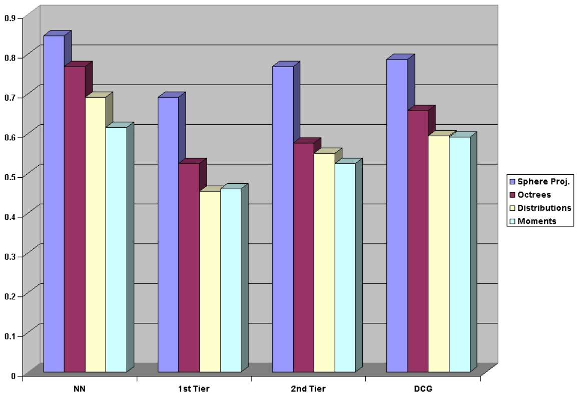


Figure 13: Robustness to level of detail

calculated by averaging over the whole database, running the algorithms on a Pentium 4 1.6GHz, 256MB RAM machine. It can be seen that both the storage and the running-time of the sphere projection and shape distribution are similar. The moments signature is computationally the most efficient signature, while the Octree is the least efficient one.

Signature	Size	Generation Time
Spheres	2.3k	2.1sec
Distributions	2.0k	1.9sec
Octrees	8.2k	2.6sec
Moments	0.4k	0.9sec

Table 2: Time and space complexity

6 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 shown how geometric signatures can be enriched by a simple topological signature. Finally, the paper has presented a comparative study of signatures, using various evaluation measures common in information retrieval. The sphere projection signature was shown to outperform the other signatures considered in this study.

We are currently working on augmenting the sphere projection signature with novel relevance feedback schemes. We also intend to handle invariance to non-rigid transformations

Acknowledgments: We thank Sagi Katz for his help in implementing the Shape Distribution and the Moments signatures. We also thank Omer Katzir and Avihai Mejer for implementing the Octree method, and Michael Gustus and Anat Paskin for collecting the 3D model library. Finally are grateful to Idan Shatz for building a 3D Search Engine based on our signatures.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, 1996.

[2] H. Borko and M. Bernick. Automatic document classification. *Journal of the ACM*, 9:512–521, 1962.

[3] C.A. Delfinado and H. Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12:771–784, 1995.

[4] M. Elad, A. Tal, and S. Ar. Content based retrieval of vrml objects - an iterative and interactive approach. *EG Multimedia*, 39:97–108, September 2001.

[5] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[6] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. *SIGGRAPH*, pages 203–212, 2001.

[7] G.R. Hjaltason and H. Samet. Ranking in spatial databases. In *Symposium on Large Spatial Databases*, pages 83–95, 1995.

[8] K. Jarvelin and J. Kekinen. IR evaluation methods for retrieving highly relevant documents. In *Research and Development in Information Retrieval*, pages 41–48, 2000.

[9] M. Kazhdan, B. Chazelle, D. Dobkin, and T. Funkhouser. A reflective symmetry descriptor for 3d models. *Algorithmica*, page to appear, 2003.

[10] S.F. Konke, P.J. Moran, B. Hamann, and K.I. Joy. Fast methods for computing isosurface topology with betti numbers. In *Proceedings of Dagstuhl Seminar on Scientific Visualization*, pages 14–18, 2002. Data Visualization: The State of the Art.

[11] R. Korfhage. *Information Storage and Retrieval.* John Wiley & Sons, New York, 1997.

[12] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *Proceedings of the International Conference on Shape Modeling and Applications*, pages 154–166, 2001.

[13] 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*, pages 103–122, 2000.

[14] Y. Rubner, L. Guibas, and C. Tomasi. The earth mover's distance, multidimensional scaling, and colorbased image retrieval. In *Proceedings of ARPA Image Understanding Workshop*, New Orleans, LA, 1997.

[15] C. van Rijsbergen. *Information retrieval*. Butterworth, London, 2 edition, 1979.

[16] D.V. Vranic, D. Saupe, and J. Richter. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In *IEEE Workshop Multimedia Signal Processing*, pages 293–298, 2001.

[17] X.S. Zhou and T.S. Hunag. Small sample learning during multimedia retrieval using biasmap. In *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, Hawaii, December 2001.