# Feature Combination and Relevance Feedback for 3D Model Retrieval

Indriyati Atmosukarto, Wee Kheng Leow, Zhiyong Huang
Dept. of Computer Science, National University of Singapore
3 Science Drive 2, Singapore 117543
indri, leowwk, huangzy@comp.nus.edu.sg

## Abstract

*Retrieval of 3D models have attracted much research interest, and many types of shape features have been proposed. In this paper, we describe a novel approach of combining the feature types for 3D model retrieval and relevance feedback processing.Our approach performs query processing using pre-computed pairwise distances between objects measured according to various feature types. Experimental tests show that this approach performs better than retrieval by individual feature type.*

## 1. Introduction

Retrieval of 3D models have attracted much research interest. Many shape features have been proposed for matching 3D objects [1, 5, 6, 7, 11, 13, 14, 18]. Different features are good for retrieving different objects under different contexts, and no one feature performs well under all situations. A natural approach to improve retrieval performance is to combine various feature types. This paper will show that feature combination for 3D model retrieval is non-trivial.

This paper introduces a novel approach of combining various feature types for 3D model retrieval and relevance feedback processing. It distinguishes itself from existing methods in the following way:

- Unlike most traditional image or video retrieval methods, a query is defined in terms of a set of at least one object instead of some shape features.
- Concatenating the values of various feature types into a long feature vector or computing the weighted sum of the distances measured by various feature types are not appropriate for 3D model retrieval. Our query processing is based on pairwise rankings of the objects, which are monotonically related to the pairwise distances.
- Shape features that are individually effective for object retrieval have very high dimensionality and occupy large amounts of storage space. Dimensionality reduction is not appropriate because it may corrupt the spatial information captured in the features. By computing with pairwise rankings of objects instead of feature vectors, we avoid explicit computation in high-dimensional feature spaces.
- Storing pairwise distances (or rankings) between the objects actually uses less space than storing various features of the objects (section 6). In this way, query processing becomes more efficient because there is no need to compute the distances between objects during query processing.
- Unlike existing methods, the contributions of known relevant and irrelevant objects are not combined using weighted sum ,instead, our method ensures that known relevant objects always rank at the top and known irrelevant objects always rank at the bottom.
- The weights of various feature types are computed by approximating the probability that the feature type is effective in retrieving relevant objects.

Extensive tests show that this approach can retrieve many relevant objects in just a small number of relevance feedback iterations, similar to the method proposed by Chang et al. for image retrieval [3, 15]. Furthermore, it has a better performance than retrieval by individual feature type.

## 2. Related Work

Existing methods for 3D model retrieval extract features from the 3D objects and represent the objects in one of four main schemes: (1) histograms, (2) 2D maps, (3) 3D grids, and (4) abstract representations.

Abstract representations encode objects as the coefficients of specialized functions or transforms of the objects' points. Examples include parabolic and trigonometric functions [7], Fourier transform [18], spherical harmonics [5], and rotation invariant spherical harmonics [6].

Most works on relevance feedback have been done for content-based image retrieval. The methods of Rui et al. [12], Elad et al. [4], and Nakazato et al. [10] all computed variations of the quadratic-form distance measure of features. The weights were updated according to the inverse of the variance of each feature among the query examples.

In particular, [4] used distance thresholds to create a margin between relevant and irrelevant objects, and the weights were adapted to bring the relevant objects closer and to push the irrelevant objects further. [10] allowed the user to indicate multiple groups of positive and negative image examples. It then clustered each positive classes while scattering the negative examples away from the positive classes.

Muller et al. [9] focused on negative example feedback in image retrieval. They showed that too much negative feedback can destroy a query. Hence, positive and negative components were weighted differently.

The method of Chang et al. [3, 15] chose the most informative instances for learning by employing MEGA (Maximizing Expected Generalization Algorithm) and SVMActive algorithms. In addition, dimensionality reduction was employed to reduce the number of low-level features.

Yin et al. [17] proposed a reinforcement learning model for integrating existing relevance feedback techniques. They showed that the integration of multiple relevance feedback approaches gives better retrieval performance than single relevance feedback technique alone, and that the sharing of relevance knowledge between multiple query sessions significantly improves retrieval performance.

All the above query and feedback processing methods work in the feature space, which has a high dimensionality. In contrast, our method does not perform query and feedback processing explicitly in the high-dimensional feature space. Another shortcoming of most of the above methods is that the positive and negative components of the relevant and irrelevant objects are combined using variations of weighted sum of features. Our test results show that this method is not reliable because weighting of the feature dimensions does not make sense for the features that we have explored (section 4).

## 3. Object Matching Criteria

Consider two objects $O$ and $O'$ represented by point-sets $\{\mathbf{p}_i\}$ and $\{\mathbf{p}'_j\}$, respectively, with possibly different number of points. One criterion of 3D shape similarity can be expressed succinctly by the well-known principle of *rigid object registration* [2]. The registration of these objects involves a rigid-body transformation $T$ of the 3D points of object $O$ to the transformed points. The transformation $T$ includes scaling, rotation, and translation. Since the two objects can have different number of 3D points sampled at different locations, the correspondence or mapping $f$ from the transformed points to the 3D points of object $O'$ must be determined. Then, the error of registration $E(O, O'|T, f)$ under $T$ and $f$ can be defined as

$$E(O, O'|T, f) = \frac{1}{|O|} \sum_i \|f(T(\mathbf{p}_i)) - T(\mathbf{p}_i)\|^2 . \quad (1)$$

The best registration between the objects is obtained by finding the transformation $T$ and mapping $f$ that minimize the error $E(O, O'|T, f)$. The minimum registration error can then be regarded as the *dissimilarity* between the objects. This criterion applies to the matching of the 3D shapes of rigid objects, disregarding the difference in scale, position, and orientation.

The standard method of normalizing the difference in position and scale is to move the objects' centroids to the origin of the 3D coordinate system, and scale the objects by the mean of the distances of the points from the origin or the square-root of the squared distances.

## 4. 3D Object Representation Schemes

There are two aspects in the representation of objects by shape features: (1) the shape feature and (2) the method of representing objects using shape feature. This section will first discuss three common methods of representing objects for 3D model retrieval. The issues of shape features will be discussed in section 5.

### 4.1. 3D Grids

3D-grid representation of an object is obtained by quantizing the minimum bounding cube of a 3D object into rectangular cells $C(x, y, z)$ at fixed, quantized locations $(x, y, z)$. Feature value $G(x, y, z)$ of cell $C(x, y, z)$ is obtained by averaging the feature values $f_i$ of 3D points $p_i$ within the cell:

$$G(x, y, z) = \frac{1}{|C(x, y, z)|} \sum_{f_i \in C(x,y,x)} f_i . \quad (2)$$

Given two objects $O$ and $O'$ with grid cell values $G(x, y, z)$ and $G'(x, y, z)$, their difference can be computed as the normalized (squared) Euclidean distance which has a bounded range of values:

$$d(O, O') = \sum_{x,y,z} (\overline{G}(x, y, z) - \overline{G'}(x, y, z))^2 \quad (3)$$

where $\overline{G}$ and $\overline{G'}$ are the values of $G$ and $G'$, respectively, normalized over the whole 3D-grid representation.

### 4.2. 2D Spherical Maps

The 3D points of a 3D objects can also be represented in spherical coordinates $(\rho, \theta, \phi)$. Then, the angles $\theta$ and $\phi$ are quantized into discrete intervals, thus dividing the object's bounding sphere into pyramidal sections $\mathcal{S}(\theta, \phi)$ at quantized angles $(\theta, \phi)$. Feature value $M(\theta, \phi)$ in section $\mathcal{S}(\theta, \phi)$ can be derived as:

$$M(\theta, \phi) = \frac{1}{|S(\theta, \phi)|} \sum_{f_i \in S(\theta,\phi)} f_i \quad (4)$$

where $f_i$ is the feature value of point $p_i$ in section $\mathcal{S}(\theta, \phi)$.

One easy way to handle rotation in $\theta$ and $\phi$ angles is to perform Fourier Transform (FT) of the 2D map. The difference between objects $O$ and $O'$ can be computed as:

$$d(O, O') = \sum_{\mu,\nu} (\overline{\mathcal{M}}(\mu, \nu) - \overline{\mathcal{M}'}(\mu, \nu))^2 \qquad (5)$$

where $\overline{\mathcal{M}}$ and $\overline{\mathcal{M}'}$ are the normalized values of the magnitudes of $\mathcal{M}$ and $\mathcal{M}'$.

### 4.3. Histograms

The object's feature values can also be discretized into intervals $I_j$ to derive the frequency or probability distribution of the feature values:

$$H(j) = \frac{1}{|O|} \sum_{f_i \in I_j} 1 \, . \qquad (6)$$

In general, histogram representations are invariant to rotations of objects. However, they loss all spatial information about the objects' features.

## 5. 3D Shape Features

This section describes some 3D shape features that are useful for object retrieval. Let $\mathcal{N}(\mathbf{p}_i)$ denote the set that contains the 3D point $\mathbf{p}_i$ and its connected neighbors in the mesh that forms the surface of an object.

1. Distance of the points from the object's centroid $\|\mathbf{p}_i\|$.
2. Local elongation:
   Perform PCA on the points in $\mathcal{N}(\mathbf{p}_i)$ to obtain the eigenvalues $\lambda_i$, $i = 1, 2, 3$, in decreasing order. The ratio $\lambda_2/\lambda_1$ is inversely related to the local elongation at $\mathbf{p}_i$.
3. Bumpiness:
   The ratio $\lambda_3/\lambda_1$ gives a measure of the bumpiness at $\mathbf{p}_i$.
4. Total curvature [8]:
   Denote the connected neighbors of $\mathbf{p}_i$ by $\mathbf{q}_1, \ldots, \mathbf{q}_m$ in sequence, with $\mathbf{q}_{m+1} = \mathbf{q}_1$. Also denote by $\omega_j$ the angle subtended by the edge connecting $\mathbf{q}_j$ and $\mathbf{p}_i$ and the edge connecting $\mathbf{q}_{j+1}$ and $\mathbf{p}_i$. Then, the total curvature $T(\mathbf{p}_i)$ can be approximated by the angle deficit:
   $$T(\mathbf{p}_i) = 2\pi - \sum_j \omega_j \, . \qquad (7)$$
5. Gaussian curvature [8]:
   Let $A_j$ denote the area of the triangle made by points $\mathbf{p}_i$, $\mathbf{q}_j$, and $\mathbf{q}_{j+1}$. Then, the Gaussian curvature $K(\mathbf{p}_i)$ can be approximated by:
   $$K(\mathbf{p}_i) = T(\mathbf{p}_i) / \sum_j A_j \, . \qquad (8)$$
6. Random distance [11]:
   The distance between two randomly selected points on the object's surface.
7. Random angle [11]:
   The angle between three randomly selected points on the object's surface.
8. Random area [11]:
   The square-root of the area of the triangle formed by three randomly selected points on the object's surface.

All the features can be captured in 3D grids, 2D maps, and histograms to represent objects in different forms. Various combinations of object representations and feature types have been used in existing methods

- [1] used histograms of number of points; and [11] used histograms of random distance, angle, and area.
- [6] used 2D maps of mean, standard deviation, and maximum distance of points to the object's centroid, surface normal, and surface area; and [18] used 2D maps of number of surfaces and mean distance of points from bounding sphere.
- [13] used 3D grids of point density; [14] used 3D grids of Gaussian curvature, normal variation, and mid point; and [6] used 3D grids of area.

## 6. Retrieval by Feature Combination

We have discussed eight 3D shape features, each of which can be captured in the three object representation schemes. In total, there are 24 different combinations. Our preliminary tests show that not all combinations work well. Out of the 24 combinations, 13 yield average or good retrieval performance:

- Histograms of random distance (RDH), random angle (RAH), random area (RRH), and Gaussian curvature (KH). Each histogram has 200 bins.
- 2D spherical maps of distance (DM), elongation (EM), bumpiness (BM), total curvature (TM), Gaussian curvature (KM), and random distance (RDM). Each map contains $64 \times 64$ entries.
- 3D grids of elongation (EG), bumpiness (BG), and Gaussian curvature (KG). Each 3D grid contains $25 \times 25 \times 25$ entries.

The memory space required to store these 13 feature representations of an object is $4 \times 200 + 6 \times 64 \times 64 + 3 \times 25 \times 25 = 72.3 \times 10^3$ units. The total memory space for storing the feature representations of $N$ objects would be $N \times 72.3 \times 10^3$ units. Suppose we pre-compute the pairwise distances between two objects measured according to each of the feature representations, then the total memory space required is $13N^2$. The break-even point is then $5.5 \times 10^3$. That is, storing the pairwise distances of up to 5500 objects actually requires less memory space than storing the 13 feature representations of the objects. Query processing time is also significantly reduced because there is no need to compute the distances and to fiddle with high-dimensional feature spaces during retrieval.

A simple approach for combining the distances measured by different feature representations is to compute the weighted sum of the distances. However, this is not a good approach. Our test results show that combining the feature representations by any form of weighted sum of distances do not yield good retrieval performance.

Our method of handling the above problem is to compute the integer rank $r_k(O_i|O_j)$ of object $O_i$ with respect to object $O_j$ in increasing order of the distance $d_k(O_i, O_j)$ between the objects measured according to representation $k$. The rank of an object with respect to itself is defined as 0: $r_k(O_i|O_i) = 0$. The largest rank is equal to the number $N$ of objects in the database. In this way, the ranks $r_k$ measured by different feature representations can be directly compared. For each object $O_j$, the ranks $r_k(O_i|O_j)$ of other objects $O_i$ with respect to $O_j$ are stored.

A query $Q$ is defined in terms of a set of at least one relevant object $\mathcal{R} = \{R_j\}$ and a set of zero or more irrelevant object $\mathcal{I} = \{I_j\}$ specified by the user. The query processing task is to determine the similarity between an object $O_i$ in the database and the query $Q$.

First, let us consider the case in which no irrelevant object is specified, i.e., $\mathcal{I} = \emptyset$. In this case, the similarity $s_{ik}$ between $O_i$ and the query $Q$ measured according to feature representation $k$ is defined as

$$s_{ik} = 1 - \frac{1}{N} \min_j \, r_k(O_i|R_j) \,. \tag{9}$$

Equation 9 is a reasonable similarity measure. In particular, the similarity between any relevant object $R_i$ and the query $Q$ is always equal to 1. So, relevant objects in the query set are always retrieved and placed at the top of the list of retrieved objects. Furthermore, an object that is very similar to any relevant object will have a small rank and, thus, a large similarity, and is also retrieved. This allows the system to retrieve other relevant objects in the database that are not in the current query set. So, the number of relevant objects retrieved is guaranteed to increase monotonically with the number of feedback iterations.

The similarity $s_{ik}$ with respect to each representation $k$ can be combined to obtained an overall similarity $s_i$:

$$s_i = \frac{1}{C} \sum_k w_k s_{ik} \,, \quad C = \sum_k w_k \tag{10}$$

where $w_k$ is the weight of feature representation $k$.

The weight $w_k$ should reflect the probability that representation $k$ is effective in retrieving objects that match the query set. The probability can be estimated as the ratio of the number of known relevant objects over the number of objects within the hypersphere in the feature space spanned by the known relevant objects. The larger the ratio, the more effective is a representation in retrieving the desired objects.

The probability is estimated as follows. For each relevant object $R_j \in Q$, the furthest known relevant object has the largest rank with respect to $R_j$ measured according to representation $k$, which is equal to the number $n_{jk}$ of objects whose distances are smaller than or equal to the furthest relevant object:

$$n_{jk} = \max_i \, r_k(R_i|R_j) \,. \tag{11}$$

So, the ratio with respect to each relevant object $O_j$ in feature space $k$ is $|\mathcal{R}|/n_{jk}$. Then, the probability or weight $w_k$ can be estimated as the mean ratio:

$$w_k = \sum_j |\mathcal{R}|/n_{jk} \,. \tag{12}$$

Consider the case in which the query set contains one or more irrelevant object, i.e., $\mathcal{I} \neq \emptyset$. Our method determines the possibly irrelevant objects based on the known irrelevant objects $I_j$ in the query set. These objects are near to the known irrelevant objects, which are accumulated over successive feedback iterations. The radius $\rho_{jk}$ of the hypersphere that contains possibly irrelevant objects can be estimated as a ratio $\alpha$ of the distance between a known irrelevant object $I_j$ and its nearest known relevant object:

$$\rho_{jk} = \alpha \min_i r_k(R_i|I_j) \,. \tag{13}$$

Then, an object $O_i$ is possibly irrelevant if it is contained in the hypersphere of its nearest known irrelevant object. The set $\chi_k$ that contains the possibly irrelevant objects is called the *exclusion set*:

$$\chi_k = \{O_i \mid r_k(O_i|I_j) \leq \rho_{jk}, j = \arg\min_t r_k(O_i|I_t)\} \,. \tag{14}$$

The exclusion sets are defined as empty sets if no irrelevant object is specified by the user. Now, we only have to modify Eq. 9 to take into account the exclusion sets:

$$s_{ik} = \begin{cases} 0 & \text{if } O_i \in \chi_k \\ 1 - \dfrac{1}{N} \min\limits_j r_k(O_i|R_j) & \text{otherwise.} \end{cases} \tag{15}$$

Equation 10, together with Eq. 15, guarantees that a known relevant object $R_j$ always has a similarity $s_j$ of 1 and a known irrelevant object $I_j$ always has a similarity $s_j$ of 0. Furthermore, an object that is regarded as near to a known irrelevant object by many feature representation types will have $s_{ik} = 0$ for many $k$. So, they will also have close to 0 similarity. By moving these possibly irrelevant objects away from the known relevant objects, the possibly relevant objects in the database are moved closer to the known relevant objects. Therefore, provided the ratio $\alpha$ is not so large that relevant objects are included in $\chi_k$, this approach (using Eq. 15) can always retrieve more relevant objects than the one with empty irrelevant set $\mathcal{I}$ (using Eq. 9).

# 7. Experiments and Discussions

## 7.1. Test Database

The database used to performed the tests was created by merging three existing sets of objects. The first dataset contained 52 objects from 34 categories. Among these 52 objects, 6 of them were manually articulated with the help of 3D Studio Max to produce a total of 110 articulated objects.

The second set, the *Utrecht Database*, contained 512 aircrafts in six categories: delta jets, conventional airplanes, multifuselages, biplanes, helicopters, and other aircrafts. The third set is a subset of the *Princeton Database*. It contained 1236 objects in 52 categories. In total, our test database contained 1910 objects.

The scale, position, and orientation of the objects were normalized using the method described in section 3. Various features were extracted from the objects, which were then represented in histograms, 2D maps, and 3D grids. Each object was represented using 13 different feature representations. Pairwise distances between the objects were computed. Then, the ranks of the objects with respect to each other were computed and stored in the database.

## 7.2. Test Procedure and Results

The tests were conducted as follows. First, a user selected one relevant object to form the query set. Next, the system retrieved and displayed the top 48 objects, and the user selected the relevant objects. Then, this retrieval and feedback process was repeated until no new relevant object was retrieved. For retrieval by single features, no irrelevant object were selected. These tests were used to obtain the baseline results. For the feature combination method, two types of tests were performed: with and without irrelevant objects. In the tests where irrelevant objects were used, all objects displayed on the GUI that were not marked as relevant by the user were regarded as irrelevant.

The type of selected relevant objects reflects the user's query context (Table 1). For example, retrieving humans in a fixed posture (fHu) requires rigid object matching criterion, whereas retrieving humans in any posture (Hu) requires articulated object matching criterion. Other rigid objects include head (He), guitar (Gu), computer monitor (Mo), rifle (Ri), and pistol (Pi). Other articulated objects include hand (Ha), ant (An), eagle (Ea), and shark (Sh).

The total number of relevant objects in the database for each query is shown in Table 1. Some queries have more than 48 relevant objects in the database. In these cases, the total number of relevant objects are taken as 48 since only 48 objects are displayed in the GUI.

The above retrieval test was performed using each of the 13 feature representations as well as the feature combination method. Nine versions of the feature combination method were tested: I$\emptyset$: without irrelevant objects, and In.nn: with irrelevant objects and $\alpha$ = n.nn.

Table 1 shows the percentage of relevant objects in the database retrieved at the last feedback iteration. The individual feature representations are effective in some but not all queries. The 3D grids capture the most complete spatial information. So, they are generally better for retrieving rigid objects. The histograms and 2D maps can match articulated versions of objects and are, thus, better for retrieving articulated objects. The feature combination method without irrelevant objects I$\emptyset$ yields better overall performance than individual feature representations. Retrieval performance is improved significantly with irrelevant objects. In particular, the one with $\alpha = 0.06$ has the best overall retrieval performance. When $\alpha$ is too large, the retrieval performance decreases for some of the queries.

# 8. Conclusion

This paper presented a novel method of combining various feature representations for 3D model retrieval and relevance feedback processing. It performs query processing based on known relevant and irrelevant objects in the query, and computes the similarity of an object with the query using pre-computed rankings of the objects without computing in high-dimensional feature spaces. Storing the pre-computed rankings uses less space than storing the feature representations of the objects. Query processing is very efficient because there is no need to compute the distances of the objects during query processing. Extensive test results show that the feature combination method significantly improves the retrieval performance of individual feature types.

# Acknowledgment

This research is supported by ARF R-252-000-137-112.

# References

[1] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Proc. 6th Int. Symp. on Spatial Databases*, 1999.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. on PAMI*, 14(2):239–256, 1992.

[3] E. Chang, B. Li, G. Wu, and K.-S. Goh. Statistical learning for effective visual information retrieval. In *Proc. ICIP*, 2003.

[4] M. Elad, A. Tal, and S. Ar. Content based retrieval of vrml objects: An iterative and interactive approach. In *Proc. EG Multimedia*, 2001.

| | fHu | Hu | He | Gu | Mo | Ri | Pi | Ha | An | Ea | Sh | 4Ch | sCh | Fi | Pl |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| total | 15 | 48 | 34 | 13 | 18 | 17 | 22 | 15 | 21 | 21 | 14 | 22 | 16 | 48 | 48 |
| RDH | 0.60 | 0.98 | 0.26 | 0.69 | 0.17 | 0.53 | 0.23 | 0.47 | 0.95 | **1.00** | 0.79 | 0.32 | 0.44 | 0.15 | 0.13 |
| RAH | 0.53 | 0.94 | 0.56 | 0.46 | 0.28 | 0.47 | 0.14 | 0.13 | **1.00** | **1.00** | 0.71 | **0.68** | 0.50 | 0.25 | 0.31 |
| RRH | 0.20 | 0.75 | **0.71** | 0.23 | 0.22 | 0.29 | 0.23 | 0.33 | **1.00** | 0.95 | **1.00** | 0.64 | 0.19 | 0.21 | 0.21 |
| KH | 0.07 | 0.48 | 0.18 | 0.46 | 0.06 | 0.12 | 0.05 | 0.27 | **1.00** | 0.57 | 0.93 | 0.09 | 0.06 | 0.19 | 0.06 |
| DM | 0.47 | 0.81 | 0.03 | 0.08 | 0.33 | 0.06 | 0.05 | 0.07 | **1.00** | 0.90 | 0.43 | 0.05 | 0.13 | **0.88** | 0.02 |
| EM | 0.60 | 0.96 | 0.29 | 0.08 | 0.28 | 0.24 | 0.23 | 0.33 | **1.00** | 0.95 | 0.93 | 0.36 | 0.13 | 0.71 | 0.15 |
| BM | 0.53 | 0.98 | 0.15 | 0.31 | 0.06 | 0.18 | 0.09 | 0.40 | **1.00** | **1.00** | 0.36 | 0.05 | 0.13 | 0.42 | 0.27 |
| TM | 0.53 | 0.88 | 0.35 | 0.23 | **0.56** | 0.35 | 0.36 | 0.20 | **1.00** | 0.86 | 0.50 | **0.68** | 0.13 | 0.83 | 0.31 |
| KM | 0.07 | 0.52 | 0.06 | 0.38 | 0.06 | 0.06 | 0.05 | 0.27 | 0.95 | 0.57 | 0.29 | 0.05 | 0.06 | 0.15 | 0.02 |
| RDM | 0.40 | 0.77 | 0.03 | 0.69 | 0.06 | 0.35 | 0.23 | 0.07 | 0.76 | 0.86 | 0.36 | 0.55 | 0.13 | 0.15 | 0.31 |
| EG | **0.93** | **1.00** | 0.56 | 0.92 | 0.17 | 0.76 | 0.41 | 0.53 | **1.00** | 0.76 | 0.93 | 0.59 | 0.44 | 0.73 | **0.54** |
| BG | 0.87 | 0.88 | 0.44 | 0.85 | 0.06 | 0.65 | 0.36 | 0.20 | 0.67 | 0.86 | 0.93 | 0.59 | 0.13 | 0.58 | 0.38 |
| KG | 0.80 | **1.00** | 0.59 | **1.00** | 0.33 | **0.82** | **0.50** | **0.60** | 0.71 | 0.57 | 0.79 | 0.14 | **0.56** | 0.71 | 0.46 |
| $I\emptyset$ | 0.87 | **1.00** | 0.62 | 0.92 | 0.61 | 0.76 | 0.55 | 0.67 | **1.00** | **1.00** | 0.93 | 0.82 | 0.56 | 0.94 | 0.40 |
| I0.00 | 0.93 | **1.00** | 0.62 | **1.00** | 0.78 | 0.82 | **0.73** | 0.87 | **1.00** | **1.00** | 0.93 | 0.82 | 0.69 | **0.98** | 0.46 |
| I0.04 | **1.00** | **1.00** | 0.85 | **1.00** | 0.72 | **0.94** | 0.64 | 0.87 | **1.00** | **1.00** | 0.93 | **1.00** | **0.75** | **0.98** | 0.71 |
| I0.08 | **1.00** | **1.00** | 0.91 | **1.00** | 0.89 | **0.94** | 0.50 | 0.53 | **1.00** | **1.00** | **1.00** | 0.91 | **0.75** | **0.98** | 0.85 |
| I0.10 | **1.00** | **1.00** | 0.88 | **1.00** | 0.83 | 0.88 | 0.59 | 0.87 | **1.00** | **1.00** | 0.93 | 0.95 | **0.75** | **0.98** | **0.88** |
| I0.15 | **1.00** | **1.00** | 0.74 | **1.00** | 0.72 | 0.88 | 0.36 | 0.47 | **1.00** | **1.00** | 0.93 | **1.00** | **0.75** | **0.98** | 0.77 |
| I0.20 | **1.00** | **1.00** | 0.82 | **1.00** | **0.94** | 0.88 | 0.41 | 0.53 | **1.00** | **1.00** | 0.86 | 0.95 | **0.75** | **0.98** | 0.77 |

**Table 1. Percentage of relevant objects in the database retrieved at the last feedback iteration. The values in bold face are the highest percentages for each query (column) in each section.**


[5] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, and D. Dobkin. A search engine for 3D models. *ACM Trans. on Graphics*, 2003.

[6] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Proc. Eurographics Symp. on Geometry Processing*, 2003.

[7] H.-P. Kriegel, T. Schmidt, and T. Seidl. 3D similarity search by shape approximation. In *Proc. 5th Int. Symp. on Large Spatial Databases*, volume 1262, pages 11–28, 1997.

[8] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. Int. Workshop on Visualization and Mathematics*, 2002.

[9] H. Mueller, W. Mueller, S. Marchand-Maillet, and T. Pun. Strategies for positive and negative relevance feedback in image retrieval. In *Proc. ICPR*, 2000.

[10] M. Nakazato, C. Dagli, and T. Huang. Evaluating group-based relevance feedback for content-based image retrieval. In *Proc. ICIP*, 2003.

[11] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distribution. In *Proc. SMI*, 2001.

[12] Y. Rui, T. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retrieval. In *Proc. SPIE*, 1998.

[13] M. T. Suzuki, T. Kato, and N. Otsu. A similarity retrieval of 3D polygonal models using rotation invariant shape descriptors. In *Proc. IEEE Conf. SMC*, pages 2946–2952, 2000.

[14] J. W. H. Tangelder and R. C. Veltkamp. Polyhedral model retrieval using weighted point sets. In *Proc. SMI*, 2003.

[15] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proc. ACM Multimedia*, 2001.

[16] Utrecht Univeristy Object Database, http://www.cs.uu.nl/centers/give/imaging/3drecog/3dmatching.html.

[17] P.-Y. Yin, B. Bhanu, K.-C. Chang, and A. Dong. Reinforcement learning for combining relevance feedback techniques. In *Proc. ICCV*, 2003.

[18] M. Yu, I. Atmosukarto, W. K. Leow, Z. Huang, and R. Xu. 3D model retrieval with morphing-based geometric and topological feature maps. In *Proc. IEEE CVPR*, pages II–656–II–661, 2003.