Journal of Mobile Multimedia, Vol. 3, No.3 (2007) 235-254

MULTI-FEATURE INTEGRATION WITH RELEVANCE FEEDBACK ON 3D MODEL SIMILARITY RETRIEVAL

SAIFUL AKBAR, JOSEF KÜNG, ROLAND WAGNER

Johannes Kepler University of Linz, Austria

{saiful.akbar, jkueng, rrwagner}@faw.uni-linz.ac.at

Received March 9, 2007
Revised July 9, 2007

In this paper, we combine the use of Reduced Feature Vector Integration (RFI) and Distance Integration (DI) with Relevance Feedback (RF) on 3D model similarity retrieval. The RFI outperforms the individual FVs and gives high probability of providing relevant objects, other than the query itself, on the limited-size of display window. Therefore, user may select as many relevant objects as possible just after the initial query for the next RF iteration. In order to deal with the user's feedback, we have used and extended an RF algorithm, which enhances the precision by employing multipoint queries and estimating feature relevance derived from both the variance of the distance of relevant objects and the maximum rank of them. In addition, an Extended Exclusion Set (EES) incorporating with Exclusion Set (ES) is introduced. Using EES and ES, the RF algorithm pushes prospectively irrelevant objects away from the queries. By utilizing both approaches, the small number of RF iterations significantly improves the retrieval precision.

Keywords: 3D model similarity retrieval, feature integration, relevance feedback, multipoint queries

1. Introduction

The research of 3D model similarity retrieval is increasing in the recent years. The 3D model similarity retrieval plays an important role in domain specific applications, such as electronic commerce, virtual reality design application, CAD/CAM application [1][2], medical applications [3][4], protein classification [5][9], pollen-forecasts system [8], forensic purpose application [6], and fitting shoes [7]. Consider an electronic commerce application as an example. Traditionally, an online furniture-shop provides a text based searching facility in order to let the customer select the furniture she plans to buy. She might search the target using a keyword such as "table". When the system retrieved many examples of tables, she has the difficulty to find which kind of table she really wants. Using a keyword of table type, the text based searching engine might help her to retrieve the models. However, this is not an intuitive way, as it requires the customer to have prior knowledge about the model type. The more intuitive way is enabling the user to query the system using some examples, which are retrieved in advance based on the keyword, and let the application find the similar model from the repository database based on the appearance, such as the shape.

Most previous work on 3D model similarity retrieval system focuses on developing new features representing 3D shape. There are many features or descriptors which are considered to represent 3D model shape, such as Shape Histogram [9], Geometry Images [11], Aspect Graph [15], Visual Similarity [16], Skeleton Based Similarity [12], 3D Fourier Transform based descriptor [17], Topology Matching [10], Physical Moment [13], and Cube Based 3D Similarity [14]. A descriptor might be effective for a kind of shape characteristic, but not for another kind of those. Moreover, as long as we know they employ a feature vector (FV) individually, instead of combining them to obtain higher discriminating power.

In order to improve the performance of single-feature based similarity search, in our previous experiment [19], we proposed a multi-feature based 3D model retrieval, which employs a combination of two features. The basic idea is as follows. When one extracts a 3D model into an FV, some information will be lost. Therefore, an extracted FV will not able to represent the characteristic of 3D model shape as a whole. Hence, a combination of some features should be able to describe 3D shape more effectively such that in the end it will provide better retrieval-performance. The experiment highlighted that combining two features, by either merging the features or the distances, using certain weighting factor improves the search performance.

Due to the complexity of representing characteristics of 3D shape and the subjectivity in similarity judgment by human, low level FVs, including multi FVs, do not sufficiently represent the higher level concept/semantic of user. In many cases, a 3D shape retrieval system provides models which are close to each other in their low level FVs, but not visually or semantically similar in user point of view. This problem is known as semantic gap problem. The RF technique provides a way to bridge the gap by allowing user to choose relevant and irrelevant objects given by the system from the previous iteration. Then the system performs the next query based on given feedbacks in order to provide as many user-desired models as possible in the next retrieved result.

In this paper, we present an approach to integrating multiple FVs into an RF technique on 3D model retrieval. In the first shoot, the system provides as many models as possible which are prospectively relevant to what user desire in the limited-size of display window; thereby the user can mark as many relevant models as possible just after getting the result from the initial query. For that, we make use of the integration of FVs, which are considered to be relatively complementing to each other, i.e. FVs which are generated from different abstraction of 3D shape. To deal with an unknown initial query, i.e. a query where the query object is not in the available dataset, we make use Reduced FV Integration (RFI), while for a known one and the next feedback iteration, we make use of Distance Integration (DI). Then, we develop an RF technique based on multipoint queries [35], [36] and the RF technique proposed by [21], and extend the use of exclusion set [21] in order to strengthen the effect of irrelevant feedbacks on object points surrounding them.

The remainder of the paper is organized as the following. In Section 2, we review some related work and present the motivation and contribution of the proposed approach. In Section 3, we describe five 3D shape descriptors extracted from different abstraction of 3D model. Section 4 presents our approach for integrating FVs, including RFI, DI, and the RF algorithm. In Section 5, we present implementation issues related to the prototype of 3D model retrieval. In Section 6, we discuss the experimental results. Section 7 presents the conclusions.

2. Related Work, Motivation, and Contribution

2.1 Combined FVs

As far as we know, the idea of combining multi-feature on 3D shape retrieval was introduced by Vranic [18]. A new hybrid feature vector of dimension 472 is obtained by crossbreeding the depth buffer FV of dimension 186, the silhouette-based FV of dimension 150 and the ray based FV of dimension 136. Instead of employing different choices of weights, they define the weighting factor by the proportion of FV dimension such that the importance of individual FV is specified by its dimension. In [26] we extract individual FVs which come from different abstraction and test with various weighting factors and integration schemas, i.e. PFI, RFI, DI, and RI. The experimental result shows that RFI and DI outperform PFI and RI.

Regarding that an individual FV may have different importance than others w.r.t. different classes of 3D model, a purity-based weight estimation is introduced by [23]. The purity value indicates the maximum number of objects belong to the same model class in the first k-position of each ranking under given individual FV and a query q. The more a purity value of FV, the more weight the FV is assigned to. The purity values are obtained by submitting a query to the training dataset, and the first k-position of the result is used for evaluating the purity value. Then, the query is resubmitted to the real dataset by regarding the weight factor obtained from the purity value. This approach enables the system to assign different weighting factor for different query object.

We observe that there are three drawbacks in either our approach [26] or purity based approach [23], as the followings:

1. The training dataset should be large enough and covers large spectrum of model classification
2. The weighting factor obtained from the training phase is not always the best one for another dataset, and
3. It strictly depends on how models in the data set are classified in advance and therefore is not able to adapt different user semantics.

2.2 Relevance Feedback (RF) Techniques

Relevance feedback has been used in the retrieval of text and images. In the following sections, we review the three de facto standard relevance feedback techniques, i.e.: Query Vector Modification (QVM), Feature Relevance Estimation (FRE), and Multipoint Queries (MPQ) / Multiple Queries (MQ). Fig. 1 illustrates the techniques.

2.2.1 Query Vector Modification (QVM)

The basic idea of the QVM approach is repeatedly reformulating the query vector using user's feedback. For each successive iteration, the query moves toward the relevance objects and away from irrelevant ones. Let $Q^{(k)}$ be the query submitted by the user at the *k*-th iteration, *R* be the set of relevant objects, and *I* be the set of irrelevant objects' FVs at the *k*-th iteration, the modified query vector [30] at the *(k+1)*-th iteration is defined as:

$$Q^{(k+1)} = \alpha Q^{(k)} + \beta \sum_i \frac{X_i}{|R|} - \gamma \sum_j \frac{Y_j}{|I|} \qquad \text{Eq. 1}$$

where X_i are the FVs of relevant objects, Y_j are the FVs of irrelevant objects, and α, β, and γ are the parameters for controlling the relative contributions of each component. The formula in Eq. 1 implies that the query moves from the original one to the new one in the feature space, as shown in Fig. 1 (a).

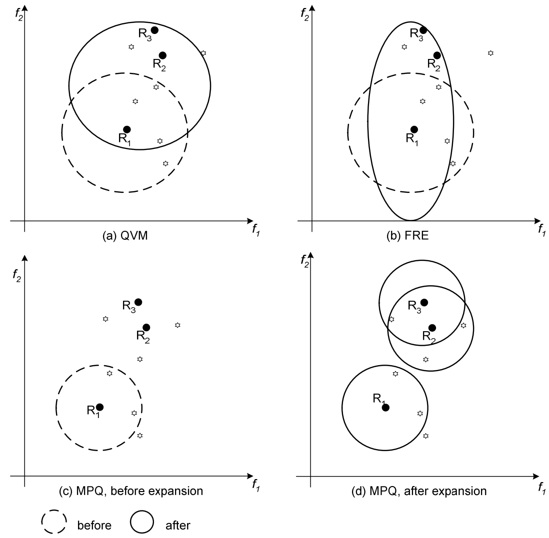


Fig. 1. (a) Query Vector Modification (b) Feature Relevance Estimation, (c) & (d) Multipoint Queries

2.2.2 Feature Relevance Estimation (FRE)

A feature may be considered to be more important than others. Thus, the contributions of a FV in similarity measurement vary in different proportion and should be estimated before measuring the similarity. The more relevant a FV, the more it contributes in the similarity measurement. The weighting factor then represents the relevance of FV, relatively to each other. Let w_i be the weighting factor of the *i*-th FV, and N be the number of individual features, the distance (L1) between object *X* and *Y*, *D(X,Y)*, is defined as:

$$D(X,Y) = \frac{1}{N}\sum_{i=1}^{N} w_i \left| x_i - y_i \right| \qquad \text{Eq. 2}$$

where x_i and y_i are the FVs of object *X* and *Y*, respectively. Fig. 1 (b) shows the situation of the query before and after weighting.

2.2.3 Multipoint Queries (MPQ) / Multiple Queries (MQ)

Unlike the previous approaches, the Multipoint Queries approach [35] or Multiple Queries approach [36] allow multiple objects as queries. Instead of providing a single point as a query, the approach allows multipoint queries based on relevant objects given by the user. Thus, a new distance of objects is determined by calculating the minimum distance between the objects and all query points. The advantage of MPQ approach is that it allows the query to find possible relevant object in different clusters. Fig. 1 (c) and (d) illustrate the situation before and after query expansion by MPQ approach respectively.

2.3 RF on 3D Model Retrieval

As far as we know, RF is relatively new area of research for 3D model similarity retrieval. A very first attempt to incorporating it into 3D model retrieval was introduced by [13]. A learning technique based on Support Vector Machine (SVM) is used for predicting the weight factor for each dimension of FV. Using relevant and irrelevant feedback (positive and negative examples) from the user, SVM divides the input space, i.e FV, into two halfspaces by defining a hyperplane so that the generated halfspaces contain the classes to be distinguished. The experiments done by the author show that after some iteration, the similar objects are convergence on the top list of the result. However, SVM approach works best if two classes have to be separated, while in our case we want to distinguish one class against all others in dataset. The problem is known as "all positive examples are alike; each negative example is negative in its own way" [32]. That is, negative examples are often not in the same class and therefore deteriorate the hyperplane. Besides, we highlight that the approach employ the weighting factor only to the dimension of a single FV, i.e. moments, instead of employing it to multiple FVs.

Bang, et. al. [33] proposes another relevance feedback technique, Feature Space Warping (FSW). The idea behind FSW is "shifting relevance models closer to the query and irrelevant object farther away". Other models, which are not in the set of given feedback, receive such an effect indirectly by moving according their distance into relevant/irrelevant models. Given a query q, and a relevance feedback f_j^*, where the subscript $* \in \{+,-\}$ indicates the relevant and irrelevant examples, respectively, FSW calculates the movement vector of pi, v_{iq}, to move pi closer to or farther away from q. Let v_{ij} be the vector from pi to f_j^*, v_{iq} be the vector from pi to q, and u_i be the strength coefficient given by the user, according to [33] the new vector of point P, v_{pi} is defined as:

$$v_{pi} = \left[\gamma \sum_{j=i}^{M} u_i \exp\left(-c\left|v_{ij}\right|\right)\right] v_{iq} \qquad \text{Eq. 3}$$

The coefficient γ and c are chosen globally such that the final coefficient multiplying v_{iq} is less than one. The constant u is determined within the interval [-2,2] in order for specifying the weight, i.e. how strong the moving effect of a feedback. The approach has a drawback in that it needs to calculate new vectors for each object during feedback sessions, and the approach is considered computationally arduous [33].

An interesting approach was proposed by Atmo et.al. [21]. It is an RF mechanism processing query and feedback based on the pair-wise rankings of objects, instead of the weighting sum of the feature or the distance. The authors argue that in their case the weighting sum of distance measured by various feature types is not appropriate for 3D model retrieval. The algorithm works as follows. Let R_{ijk} be the rank of relevance object f_i^+ obtained by submitting a query f_j^+ to the k-feature, and N be the number of relevant objects, the importance of the k-feature, w_k, is estimated by Eq. 4.

$$w_k = \sum_{j=1}^{N} \frac{N}{\max_i\left(R_{ijk}\right)} \qquad \text{Eq. 4}$$

Then, the new distance for each object except those in the exclusion set, i.e. those which is close to f_i^-, is defined by the minimum distance of objects from the relevance objects. The new distance for objects in the exclusion set is set to the maximum distance such that it will be ranked in the bottom list.

Our approach distinguishes itself from the previous approaches in the followings:

1. We incorporate RFI and DI with an RF technique. RFI enables unknown query processing, but consequently it needs distance calculation. DI accepts only known query but performs retrieval faster than RFI because the distance is calculated in advance.
2. Our previous experiment [26] shows that distance integration outperforms rank integration. Therefore, different from [21] our technique makes use of the distance, instead of the rank, and formulates appropriate weighting estimation based on the combination of the variance of distances and the ranks. More specifically, the multipoint queries approach is adapted for the distance integration rather than the rank integration as done by [21].
3. We expand the idea of the exclusion set [21] by defining an extended exclusion set to increase the coverage area in distance space affected by negative/irrelevant feedback. Then, our algorithm deals with objects in the both sets in different way.
4. Different from [33], which moves objects in feature space (object points are updated and the updates are used for the next feedback iteration), our approach updates the distance during feedback iteration, and use the FV only for the first query. In addition, FSW employs single point query, while our approach employs multipoint queries.

2.4 Motivation of Our Approach

The motivation of our approach is two folds. The first one is related to the approach of integrating multi-feature, while the second is the motivation of the proposed RF technique.

There are two motivations inspiring us to exploit an integration of multi feature. First, an individual descriptor is able to describe only some limited and specific aspects of 3D shape rather than the whole aspect of those. For example, suppose that we make use of the 2D shape of 3D object to deduce similarity, and decide to choose a region-based descriptor. In this case, the descriptor can represent the interior region of the shape, but not the outer boundary or contour. On the other hand, the only use of a contour-based leads to the opposite. Second, for a given 3D model database, an individual FV that brings the best retrievals to a certain class of query object may be inferior to another individual FV for another class of query object; and integration approach could be exploited and the best one can improve retrieval performance.

Most CBIR technology assumes that similar images/shapes lay in a single cluster [36]. Therefore, they employ RF technique by reformulating a single query centred in the cluster. However, this is not the case in 3D model retrieval. Since FV does not sufficiently capture the whole shape characteristics of 3D model, we argue that similar models are often scattered in several clusters. Hence, adapting from [35] and [36], we make use of the Multipoint Queries approach that considers relevant objects as multipoint queries.

In general point of view the RF technique proposed in this paper is the same as the one proposed by [21]. We emphasize on providing the rationale of using positive feedbacks as multipoint queries, integrating DI into the algorithm (rather than RI), defining an appropriate formula of feature relevance estimation, and proposing the use of the extended exclusion set. More specifically, we observe that the exclusion set defined by [21] provides only a single way to deal with objects in the set, i.e. by regarding them as irrelevant objects and setting the distance to the maximum distance. Thus, the use of exclusion set strongly reduces the number of feedback iteration while finding similar objects, but at the same time may lead to wrong filtering result. It is because the similar objects close to the irrelevant objects are penalized by maximum distance; therefore, they will be ignored in the next iteration. In order to minimize the problem, in our approach we define the extended exclusion set that expands the exclusion set to make the filtering strategy more flexible.

2.5 Contribution of the Paper

The contribution of this paper includes the following aspects:

1. Selecting FVs, which are considered to be complementing to each other. In order for that, we choose FVs extracted from different abstractions of 3D model, i.e. 2D image, surface, and volume, as described in Section 3. By using relatively-complement FVs, the system may provide the similar objects on the top list and display them in limited-size of display window; thereby the user can mark prospectively relevant objects just after getting the result from an initial query for the next iteration.
2. Proposing a use of both Reduced FV Integration (RFI) for processing initial query and weighted Distance Integration (DI) using RF. RFI enables user to query with unknown objects, improves retrieval effectiveness comparing to the individual FVs, and optimizes the storage by reducing the dimensions. DI enables user to query with known objects, and improves the retrieval effectiveness by employing the weighting factor, which is predicted based on positive feedbacks given by the user.
3. Estimating the feature relevance (the weighting factor) by the variance of the distances and by the ranks, and proposing an the extended exclusion set incorporating with the exclusion set for filtering objects, which are prospectively irrelevant.

3. 3D Shape Descriptors

In this section, we describe several 3D shape descriptors used in our experiments. Our selection is based on the abstraction represented by the FVs. We define FV as an n-dimensional vector of numerical features that represent an object. Our hypothesis is integrating some individual features will enhance the capability to describe 3D shape more comprehensively, and at the end will enhance the performance of similarity-retrieval. We consider some criteria to select individual FVs and obtain a successful integration, i.e.:

1. Discriminating power of individual FV. Some previous researches highlight the superiority of some individual FVs. For example, depth buffer FV is shown to be superior to cord based FV [18], and Light Field Descriptor which is using Zernike moment FV is shown to be superior to geometric moment [34].
2. The possibility to be complementing to each other. For the case of 2D shape (image) based FV, contour and region should be complementing to each other. In addition, our hypothesis is that

image based FV, volume based, and surface based could be complementing to each other since they provide different abstractions of 3D objects.

3. The uniformity in FV representation and similarity measurement. If FVs could be represented in a uniform manner, and have the compatibility of similarity measurement, integration approach could be employed in the same manner. For example, contour based FV and skeleton descriptor are considered not compatible to each other. The former could be represented as a histogram or a sequence of Fourier coefficients and L1 distance could be employed as metric. On the other hand, the latter needs to be represented in a graph and therefore needs graph matching to measure the similarity.
4. The same intermediate representation. The depth buffer image could be used as intermediate representation before extracting contour, Zernike moments, and depth buffer FV, while 3D object voxel could be used for intermediate representation of both volume and surface based FV.

We extend the abstraction depicted in Fig. 2 from 2D image to 3D model case, i.e. by extending contour into surface, and region into volume, and thereby we may expect to have different similarity abstractions of 3D models. We select three different abstractions: 2D-image (C2D, ZNK), Surface (C3D, DB) and Volume (FO).

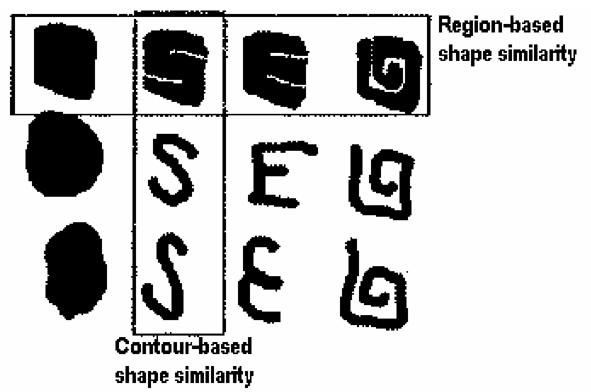
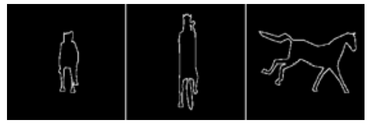


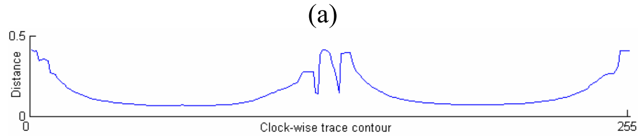
Fig. 2. Region-based versus contour-based shape similarity [31]

3.1 2D Contour (C2D)

A 3D model viewed from the major axes, *z*, *y*, and *x*, forms three 2D projection-images, lying on *xy* plane, *xz* plane, and *yz* plane, respectively. A 2-dimensional contour is defined as a collection of border points, i.e. the outermost points of an image, which intersects with image background as depicted in Fig. 3 (a). Fig. 3 (b) shows that by tracing the contour in a clock-wise direction, we record the distance between contour point and the object's origin. The n-dimensional 2D contour FV is defined as the magnitudes of the first *n* Fourier coefficients obtained by employing 1D Fast Fourier Transform (FFT) to the sequence of distances between contour points and the origin.



(a)



(b)

Fig. 3. (a) 2D Contour of a 3D model, obtained by projecting it to yz-plane, xz-plane, and xy-plane, and (b) the centroid distances of the sample points of the third contour image, recorded in a clock-wise direction.

3.2 Zernike Moments (ZNK) [16]

Different from C2D descriptor, the computation of the Zernike moments descriptor does not need the knowledge of boundary information. Thus, it is more suitable for more complex shape representation. The low order moments represent the global pattern of the shape while the higher order the detail. For a discrete image, if P_{xy} is the current pixel value of an image, two-dimensional Zernike moments order *m* with repetition *n* is defined as:

$$A_{mn} = \frac{m+1}{\pi} \sum_x \sum_y P_{xy} R_{mn}(r) e^{jn\theta}$$
$$R_{mn}(r) = \sum_{s=0}^{(m-|n|)/2} (-1)^s \frac{(m-s)!}{s!((m+|n|)/2-s)!((m-|n|)/2-s)!} r^{m-2s}$$
$$x^2 + y^2 \le 1,\ j = \sqrt{-1}$$
$$m = 0..\infty,\ m-|n| = even,\ |n| \le m,$$
$$r = \sqrt{x^2+y^2},\ \theta = \tan^{-1}\left(\frac{y}{x}\right)$$
$$A^*_{mn} = A_{m,-n} \qquad \text{Eq. 5}$$

The Zernike moments based FV is defined as set of Zernike moments coefficient $A_{n,m}$ where $n \geq 0$, taken from the 2D projection images of the 3D model.

3.3 Depth Buffer (DB) [18]

Depth buffer stores the depth of image projection of 3D model from a certain direction. As there are three major axes in 3D space, we obtain six depth buffers containing the depth information of the model viewed from *x+*, *x-*, *y+*, *y-*, *z+*, and *z-*, as depicted in Fig. 4. Depth buffer based FV is the magnitudes of the first *n* Fourier coefficients obtained by employing 2D FFT to the image.

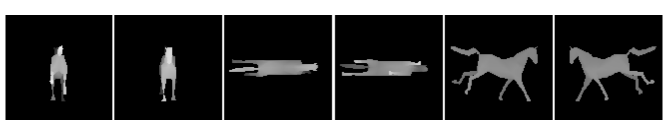


Fig. 4. Depth images of 3D model, viewed from x+, x-, y+, y-, z+, and z-

3.4 Fractional Occupancy (FO)

Adapting from [20], we define fractional occupancy (density) as the ratio of the actual number of voxel in a partition (Vx_i) to the maximum possible number of voxel in the partition (Vp_i).

$$FO_i = \frac{Vx_i}{Vp_i} \quad \text{Eq. 6}$$

In this case, we partition 3D voxel to a grid partitions. The n-dimensional FO based FV is defined as the magnitudes of the first *n* Fourier coefficients of the collections of FO.

3.5 Cords and Spherical Harmonic (C3D) [18][20]

A cord is a vector that goes from the centre of mass of an object to a representing surface point. One issue to be concerned with is which point should be a representing point. We observe two candidate points, which are possible to select: the most distance point, which intersects with the cords [18], or the most distant point in the partition, no matter it intersects with the cord or not. In our implementation, a 3D model is sampled into a collection of voxel. Hence, we decided to use the second, as it is relatively more invariant to the number of sampling than the first. We extract spherical harmonic coefficients by employing a Spherical Harmonic Transformation (SHT) to the collections of cords' length using S2kit [24]. Cords based FV is defined as a collection of the n-first coefficients.

4. Multi Feature Integration with RF

The idea of our approach is as the following. First, before processing user's feedback, it is important to display objects, which are prospectively relevant to the query on the limited-size of display window. In other words, the FVs for processing the first query must provide high discriminating power. We propose to make use of RFI and DI for fulfilling such a requirement. Second, in order to enhance the precision, we enable user to select relevant/irrelevant objects as feedback. Thus, using the relevant objects the query processor calculates the weighting factor of distance integration. Then, the query processor pushes the relevant objects to the top list, and the irrelevant objects to the bottom. In addition, relevant objects affect the object surrounding them to move to the top list, while irrelevant ones affect them to push away from the top list. The technique is based on the multipoint queries approach [35], [36] and the RF algorithm proposed by [21] with some extension.

4.1 Reduced FV Integration (RFI)

Let $f_1, f_2, \ldots f_n$ be a set of normalized FVs and $f'_1, f'_2, \ldots f'_n$ be a new set of FVs obtained by reducing the dimension of the original FVs, a new reduced integrated FV $f^{(1)}$ is defined as

$$f^{(1)} = \left\{ w_1 f_1^{'}, w_2 f_2^{'}, \ldots, w_n f_n^{'} \right\} \quad \text{Eq. 7}$$

Since the FVs are generated from different approaches, they range in different values. In order to combine them proportionally, each FVs should be normalized. We employ Gaussian normalization [22] for normalizing both features and distances before integration. The normalization maps the values of FV components and distances into normalized values within the interval [0,1].

The goal of reducing dimension before the integration is to preserve the dimension of integrated FV equal or almost equal to those of the original FVs; thereby the cost for distance computation is almost the same as those of single original feature. The dimension reduction might also reduce the discriminating power, which is not expected. However, as shown in Fig. 5, in general for a certain range of dimension, i.e. about 50-250, the difference of dimensionality does not yield much difference of discriminating power. Therefore, the integration model could be employed if the reduction should be not too much. Note that, the trend that we can see in Fig. 5 is applied not only to the particular dataset, i.e. CCCC dataset, but also to other datasets.

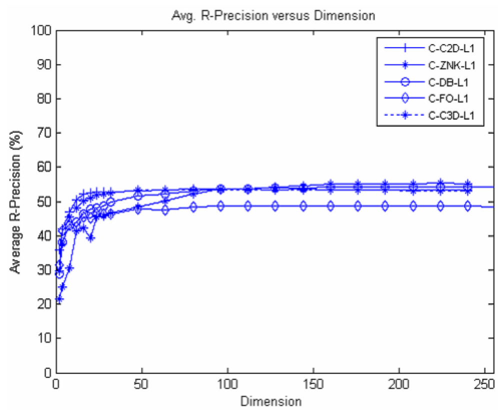


Fig. 5. The effect of dimensionality reduction on average R-Precision for CCCC dataset

4.2 Distance Integration (DI)

Let d_i be a normalized distance between object O_1 and O_2 under specified FV f_i, a new integrated distance $d^{(1)}$ is defined as:

$$d^{(1)} = \sum_{i=0}^{N} w_i d_i \qquad \text{Eq. 8}$$

In this case, we store pair wise distances between two objects, instead of their FV. Hence, there is no way to process unknown query objects where calculating distance from pair wise FV is needed.

4.3 Relevance Feedback Algorithm

The RF algorithm presented in this paper is composed of three ideas: (1) estimating the weighting factor by the variance of the distance of positive feedbacks and maximum rank of them, (2) making use of positive feedbacks for multipoint queries, and (3) building exclusion set (ES) and extended exclusion set (EES) using negative feedbacks. We explain the detail of the technique as follow.

4.3.1 Weighting Factor Estimation

The definition of W comes from the assumption that under a given FV, the more convergent relevant-objects, the more important a FV and the more weighting factor is assigned for it. The convergence indicates how close the relevant objects are to each other. We deduce the convergence from how dense the distances of relevant objects are and derive it from the variance of the distances and from the rank positions. Let R_i be a positive feedback object, σ_i^k be the variance of the distances of relevant objects under query of relevant object R_i using the k-th FV, r_{ij}^k be the rank of relevant object j under query of R_i using the k-th FV, N be the number of objects in the database, and ω be the significance of the weighting vector derived from the variance, the weighting factor for the k-th FV, W_k, is defined as:

$$W_k = \omega \frac{\sum_i 1/\sigma_i^k}{\sum_m \sum_i 1/\sigma_i^m} + (1-\omega) \frac{\sum_i N / \max_j (r_{ij}^k)}{\sum_m \sum_i N / \max_j (r_{ij}^m)} \quad \text{Eq. 9}$$

Therefore, the distance function, where M individual FVs are employed, is changed from Eq. 10 to Eq. 11.

$$D_{old}(P,Q) = \sum_{k=1}^{M} D_k(P,Q)/M \quad \text{Eq. 10}$$

$$D_{new}(P,Q) = \sum_{k=1}^{M} W_k D_k(P,Q) \quad \text{Eq. 11}$$

Note that when performing retrieval our RF technique is dealing with pair-wise distances, which are calculated in advance. Therefore, during the retrieval no calculation for $D_k\,(P,Q)$ is needed.

4.3.2 Positive Feedbacks as Multipoint Queries

As mentioned in the previous section that most CBIR technology, no matter what feedback strategy, assumes that similar images/shapes lay in a single cluster. Therefore, retrieval can be performed by formulating a single query centred in the cluster. However, in the case of 3D model retrieval, where feature vector mostly does not sufficiently capture the whole shape characteristics of 3D model, similar models are often scattered in several cluster. In order to solve the problem, we adapt an approach using multipoint queries [35],[36] by regarding each object in the set of positive feedback as queries. Therefore, the distance of object P w.r.t. multiple queries Q is the minimum distance between P and all query point R_i.

$$D_{mpq}(P,Q) = \min_i D(P,Q_i) \quad \text{Eq. 12}$$

4.3.3 Building EES and ES using negative feedbacks

We assume that negative feedbacks given by the user are considered not desired. Hence, it is expected that they will be ranked in the bottom. In order for that, we set the distance between objects in the negative feedback and the queries to maximum distance value. In addition, a negative feedback influences objects closely surrounding it by moving them farther away from the query. More specifically, we define exclusion set [21] and extended exclusion set for categorizing such objects and formulating different strength of negative feedback's effect for them.

Definition 1 Exclusion Set (ES):
Given a negative/irrelevant object as feedback I, the exclusion set of I is a set containing objects, which are considered very close to I. The distance between objects in the exclusion set with a query is maximum distance 1.0.

Definition 2 Extended Exclusion Set (EES):
Given a negative/irrelevant object as feedback I, the extended exclusion set of I is a set containing objects, which are considered not very close to I but still close enough for getting the effect of the feedback I, i.e. by moving farther away from the query.

Suppose that *P* is an object in the EES of negative feedback object *I*. The closer *P* to *I*, the more it has to be moved farther away from the query. Let D_{es} and D_{ees} be the distance between *I* point to the ES boundary and EES boundary, respectively, D_{ip} be the distance between *I* to *P*, D_{min_rp} and D_{max_rp} be the minimum and maximum distance between *P* and all positive feedback points, respectively. A new distance between *P* and *Q*, *D(P,Q)*, is defined as:

$$D(P,Q) = D_{min\_rp} + \frac{D_{ees} - D_{ip}}{D_{ees} - D_{es}}.(D_{max\_rp} - D_{min\_rp}) \qquad \text{Eq. 13}$$

It is clear that the formula in Eq. 13 implies that for each object in EES the closer an object *P* to a negative object, the more *P* is pushed away from the query. Fig. 6 (a) provides an illustration example how to deal with the multipoint queries, exclusion set, and extended exclusion set. Given positive feedbacks R_1, R_2, and R_3, a negative feedback *I*, and unknown objects $P,...,P_6$. The concept of multipoint queries implies that $D(P_1,Q)=D(P_1,R_1)$, $D(P_2,Q)=D(P_2,R_2)$, $D(P_3,Q)=D(P_3,R_3)$, $D(P_4,Q)=D(P_4,R_3)$. Since P_5 is a member of exclusion set of *I*, $D(P_5,Q)=1.0$. On the other hand, in the case of P_6, which is a member of extended exclusion set of *I*, $D(P_6,Q)$ is set to the value formulated in Eq. 13, which ranges between the minimum distance, $D(P_6,R_1)$, and the maximum distance, $D(P_6,R_2)$, as shown in Fig. 6 (b). Note that the movement vector depicted in Fig. 6 (b) is only an illustration to show that the algorithm moves P_6 farther away from the query object, therefore the direction means nothing since we deal with the distance, instead of the FV.

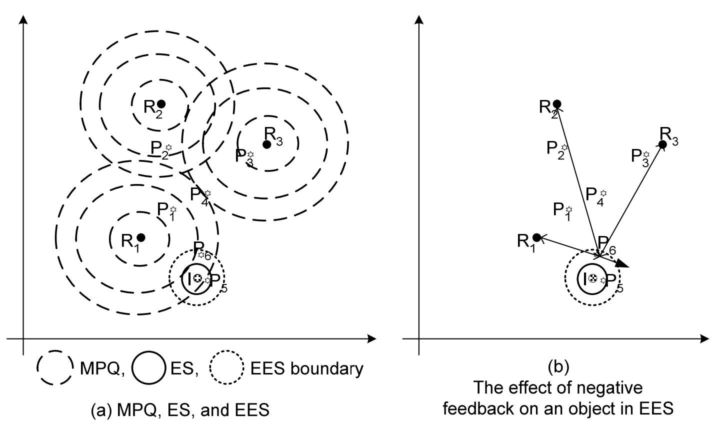


Fig. 6. An illustration of MPQ,, ES, and EES

The overall algorithm of multi-feature integration using RF is depicted in Fig. 7. Note that RFI_FV and RFI_MTX represent a table of FV generated by RFI and a matrix of distance obtained from DI, while MULTI_MTX is an array of matrices of distance generated from the original single FVs.

By using RFI_FV, the system enables users to query with unknown objects with the less possible required storage, and in the same time preserves the retrieval precision as obtained by combining multi FVs. Recall that RFI_FV is obtained by reducing the dimension of single FVs and combine them. On the other hand, RFI_MTX is used for processing known user queries without distance computation. In general, RFI_FV and RFI_MTX are considered the best in providing initial query result, comparing to other single FVs.

---

1. Initialize table entry: RFI_FV, RFI_MTX, MULTI_MTX[1..N]
2. User starts with a query object Q
   2.1 If Q is not available in the database:
      (a) extract the FVs from Q, and build an reduced FV integration of Q,
      (b) calculate the distance between Q and all objects whose FV is stored in RFI_FV,
      (c) present the result with ascending ordered distances.
   2.2 If Q is available in the database, present the result from RFI_MTX with ordered ascending distances
3. While user is not satisfied with the result do
   3.1 User marks objects as relevant (R) and irrelevant (I); put R and I in the RF history,
   3.2 Using R, for each FV observe the data in MULTI_MTX to calculate the weighting factor, *W*, as formulated in Eq. 9
   3.3 Calculate *DMin*, the minimum distance between R and I.
   3.4 Define the radius of EES outer boundary, i.e. $DMin*\lambda$,
   3.5 Define the radius of ES boundary, i.e. μ*radius(EES)*, where $\mu = [0,1]$
   3.6 For objects in the exclusion set, set the distance to the maximum distance 1.0,
   3.7 For objects in the extended exclusion set, set the distance to the value obtained from Eq. 13
   3.8 For objects in the set of relevant query, set with the minimum distance 0.0,
   3.9 For other objects, set the distance to its minimum distance from relevant objects, i.e. $D(O_j, Q) = \min_k D(O_j, R_k)$
   3.10 Use the weighting factor to calculate the new weighting sum of distance as in Eq. 11

Fig. 7. The algorithm of multi-feature integration using RF

The algorithm pushes the relevant objects to the top list and the irrelevant objects to the bottom list. By setting the distance of relevant objects with zero, we ensure that relevant objects are always at the top list, while the distance of irrelevant object with 1.0, we ensure that irrelevant object are always at the bottom list during a query session. A constant λ is used for specifying the outer boundary of EES, while a constant μ is for specifying the proportion of the boundary of ES, which is also the inner boundary of EES. The greater λ and μ are defined, the larger a hyperplane to filter objects, and the faster relevant objects are pushed to the top. By defining EES incorporating ES, the algorithm provides a more flexible way to deal with objects surrounding irrelevant feedback objects. The effect of irrelevant objects on objects in EES is intended to be less than those in ES. Thereby, the algorithm takes more advantage by using EES incorporating with ES than using only ES.

5. Implementation

We implemented a prototype of 3D retrieval system employing the multi-feature integration approaches with RF as described in Section 3 and Section 4. For our experiments, we use Princeton Datasets [27], i.e. Training Dataset (PrincTrain) and Testing Dataset (PrincTest), the subset of CCCC Dataset [28], and Utrecht Dataset [29]. Table 1 shows the number of models and classes in the datasets, while Table 2 shows the partial description of model classification in PrincTrain Dataset.

Table 1 Datasets for the experiments

Dataset	No. Of Models	No. of Classes
Princeton Training (P)	907	90
Princeton Testing (Pt)	907	92
The subset of CCCC (C)	473	55
Utrecht (U)	376	5

Table 2 Partial description of the classified set of models in PrincTrain Dataset

Description	No. Of Models	Description	No. of Models
Commercial airplanes	10	Swords	15
Fighter jet planes	50	Heads	16
Bees	4	Bridges	10
Spiders	11	Light houses	5
Humans	50	Chest	7
Humans arms out	21	Beds	8
Trex	6	Shelves	13
Pigs	4	Round tables	12
Dolphins	5	Desk lamps	14
Sharks	7	Flowers	15

Fig. 8. An example of retrieval result using various FVs: C2D, ZNK, DB, FO, C3D, and RFI. Note that the result using RFI contains more match objects than others.

After pose normalization using weighted PCA, a modification of [25], we extract five features from the models in the dataset, as described in Section 3. For each individual FV, we choose the dimension of about 256, while for RFI, we reduce the dimension of every FV to 51 such that the reduced integrated FV will have the dimension of 255. In order to calculate the distance between two objects, we make use of L1 distance as our preliminary experiments show that it is the best among the others, such as Euclidean distance and Quadratic distance. An example of retrieval result with query object *498* using each FV is shown in Fig. 8, while the retrieval result with the same query object after the third iteration of relevance feedback is depicted Fig. 9. The star (*) in the image caption indicates that the model is classified in the same class in the ground truth and considered similar to the query.

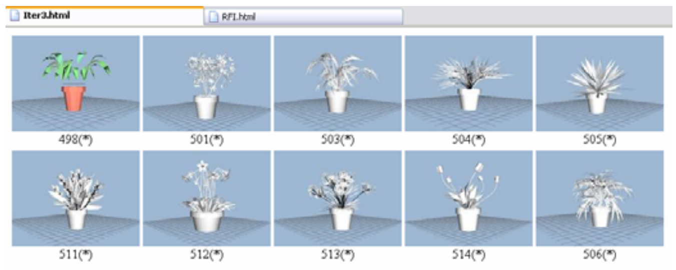


Fig. 9. An example of retrieval result after the third iteration of relevance feedback.

6. Experimental Results and Discussion

We performed several experiments to evaluate our approach. Experimental results show that for each class in the data set, employing RFI does not always improve the retrieval precision. For a particular class, RFI improves the precision, but for another class it decreases the precision. However, it is interesting to note that on average RFI outperforms all original individual FVs, as shown in Table 3. Moreover, another observation of RFI shows that the frequency that one or more similar objects, excluding the query object itself, are put on the display window with the size 25 is high enough, i.e. 90.2% (P Dataset), 89.5% (Pt Dataset), 91.5% (U Dataset), and 98.9% (C Dataset). Therefore, RFI is considered to be the most powerful representation comparing to the individual FVs.

Table 3. Average Precision Recall for Each Individual FVs Compared With RFI

Dataset	Average Precision Recall					
	C2D	ZNK	DB	FO	C3D	RFI
Princeton Training (P)	0.381	**0.416**	0.387	0.346	0.390	0.429
Princeton Testing (Pt)	0.386	**0.417**	0.395	0.351	0.388	0.436
Utrecht (U)	0.621	0.619	0.592	0.622	**0.634**	0.636
The subset of CCCC (C)	0.534	**0.552**	0.543	0.483	0.530	0.589

The values in the bold face are the highest performance among the individual FVs for the same dataset.

The next following experiments aims to measure the effectiveness of the RF algorithm as described in Section 4. Since we have limited space in the application for displaying the retrieval results, it is important to note that of the *N* models in the database, the most important measure is the *M*-first rank result that can be seen in the window by the user. Therefore, we make use of retrieval accuracy metric, which measures only the *M* first rank no matter how the ranks in range *M*+1..*N*. That is the proportion of the number of relevant object retrieved and displayed in the window to the class size. If the class size is more than the window size, we consider the window size as the class size. In fact, this cannot compare the effectiveness among different class of objects since the proportions have different denominators, i.e. the class size or the window size. Nevertheless, it is obvious that this measure can indicate how the successive iteration produces more accuracy.

The second experiment aims to observe the effect of the weighting factor. We compare the precision improvement from the zero iteration to the first iteration for two cases, i.e. without the weighting factor and with the weighting factor. The first case is obtained by setting equal weighting factor for the distances under specified FV. Table 4 gives evidence that the precision is improved not only by pushing the relevant objects to the top list and the irrelevant objects to the bottom, but also by the weighting factor. The experiment also highlights that using both variance of distance of positive feedbacks and the maximum rank of them for feature relevance estimation, i.e. ω=0.5, tends to be better than using only one of them. Although this is not always the case, we use the same weighting factor for the next experiments.

Table 4 Comparison between unweighted (U) and weighted (W) integration

Dataset	Precision improvement (%) from 0-1 iteration (M=25, λ=0.0, μ =0.0)			
	U	W (ω=0.5)	W (ω=1.0)	W (ω=0.0)
Princeton Training (P)	33.95	35.04	34.13	34.58
Princeton Testing (Pt)	31.38	32.97	32.84	32.34
The subset of CCCC (C)	26.04	27.26	27.63	26.87
Utrecht (U)	18.85	19.49	19.51	19.39

In the third experiment, we explore the sensitivity of ES and EES. Various values of λ and μ are employed throughout the experiment. In order to do that, we define various value of λ, i.e. 0.5, 1.0, 1.5, 2.0, and for each λ, we define μ = 0.0, 0.1,…,1.0. The experiment shows that for λ =0.5, 1.0, 1.5, 2.0 the maximum average precision is obtained by setting μ=1.0, 0.7, 0.4, 0.2, respectively. As shown in Fig. 10, among the four parameter setting, we found that the parameter setting with λ=1.5 and μ=0.4 provides the highest average precision.

We also explore if the EES is useful for increasing the average precision. In order for that, under the same size of ES, i.e. 0.6, we compare two cases, where EES is enabled (λ=1.5 and μ=0.4), and EES is disabled (λ=0.6 and μ=1.0). As depicted in Fig. 10, the former is better than the latter.

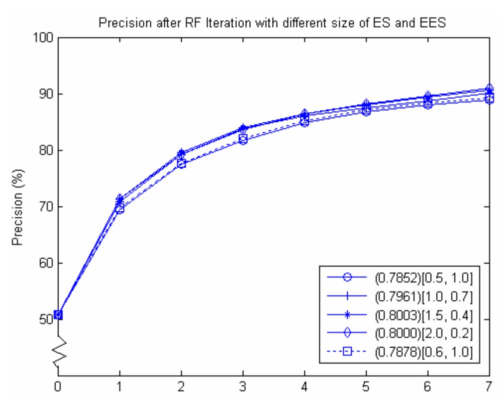


Fig. 10. Precision versus number of iteration on PrincTrain dataset, with windows size M=25 and different values of λ and μ

In the last experiment, we want to see how the algorithm enhances the retrieval performance when it runs for all datasets. We use the optimum setting, i.e. ω=0.5 for feature relevance estimation (weighting), the radius of EES is 1.5 (λ=1.5), and the radius of ES is 0,6 (μ=0.4 with λ=1.5). Fig. 11 shows that with the small number of RF iterations, the precision significantly increases.

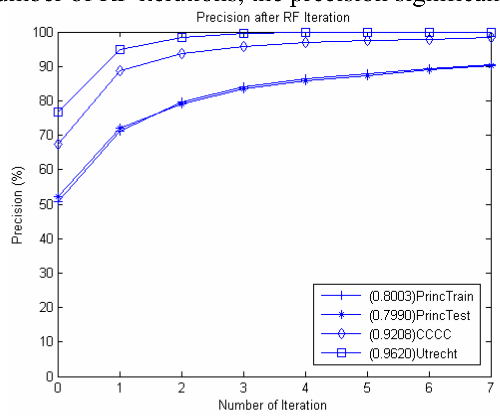


Fig. 11. Precision improvement after RF iterations (M=25, λ=1.5, and μ = 0.4)

7. Conclusion

In this paper, we propose to combine the use of RFI and DI with RF. Using RFI, we obtain the highest possible precision during the initial query, with known or unknown object query, and in the same time speed up the query processing. The experimental results show that RFI generally outperforms the individual FVs, i.e. contour (C2D), Zernike Moments (ZNK), Depth Buffer (DB), Fractional Occupancy (FO), and Cords (C3D). The experiments also highlight that FVs extracted from different abstractions of 3D model tend to be complementing to each other. The use of RFI also provides the similar object on the display window with the probability of about 90% for all datasets; thereby gives more possibility for the user to have and mark the relevant objects as feedback after the first query.

We estimate the weighting factor for each FV using the variance of relevant objects' distance and the maximum rank of the relevant objects, which are given by the user during RF iteration. The RF algorithm makes use of relevant objects as multipoint queries such that make it possible to find similar object in different clusters. The algorithm also employs extended exclusion set incorporating with exclusion set to push prospectively irrelevant objects away from the query. The experimental results show that the RF technique using distance integration (DI) significantly improves the precision of retrieval result only with the small number of RF iterations.

Acknowledgement

During this work, Saiful Akbar was supported in part by ASEA-UNINET Technology Grant of OeAD, which he greatly acknowledges.

References

[1] G. Cybenko, A. Bhasin, K. D. Cohen: "Pattern Recognition of 3D CAD Objects: Towards Electronic Yellow Pages of Mechanical Part". Smart Eng. Design, vol. 1, pp. 1-13, 1997.

[2] H. -P. Kriegel, P. Kröger, Z. Mashael, M. Pfeifle, M. Potke, T. Seidl: "Effective Similarity Search on Voxelized CAD Objects". Proc. 8th Conf. On Database Systems for Advanced Applications, Japan, 2003.

[3] Y. Liu, F. Dellaert: "A Classification Based Similarity Metric for 3D Image Retrieval". Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, 1998.

[4] D. A. Keim: "Efficient Geometry-based Similarity Search of 3D Spatial Databases". Proceedings of the 199 ACM SIGMOD International Conf. on Management of Data. ACM Press, NY, pp 419-430, 1999.

[5] J.-S. Yeh, D.-Y. Chen, B.-Yu. Chen, M. Ouhyoung: "A Web-based Three-dimensional Protein Retrieval System by Matching Visual Similarity". Bioinformatics 2005 21: 3056-3057, 2005.

[6] Z. Geradts, H. Hardy, A. Poortman, J. Bijhold: "Evaluation of contents based image retrieval methods for a database of logos on drug tablets". Proceedings of SPIE Enabling Technologies for Law Enforcement and Security. Vol. 4232, pp. 553-562, 2001.

[7] M. Novotni, R. Klein: "Geometric 3D Comparison – an Application". ECDL WS Generalized Documents, 2001.

[8] O. Ronneberger, H. Burkhardt, E. Schultz: "General-purpose Object Recognition in 3D Volume Data Sets using Gray-Scale Invariants – Classification of Airborne Pollen-Grains Recorded with a Confocal Laser Scanning Microscope". Proceedings of the 16th International Conference on Pattern Recognition, Quebec, Canada, September 2002.

[9] M. Ankerst, G. Kastenmueller, H.-P. Kriegel, T. Seid: "3D Shape Histograms for Similarity Search and Classification in Spatial Databases". SSD, 1999.

[10] M. Hilaga, Y. Shinagawa, T. Kohmura, T. L. Kunii: "Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes". ACM SIGGRAPH, 2001.

[11] H. Laga, H. Takashi, M. Nakajima: "Geometry Image Matching for Similarity Estimation of 3D Shapes". CGI'04 pp. 490-496, 2004.

[12] H. Sundar, D. Sliver, N. Gagvani, S. Dickinson: "Skeleton Based Shape Matching and Retrieval". International Conf. on Shape Modeling and Applications 2003, May 12 - 15, 2003.

[13] M. Elad, A. Tal, S. Ar.: "Directed Search in a 3D Objects Database Using SVM". HPL-2000-20(R.1), 2000.

[14] C. -S. Wang, J. -F. Chen, L. -P. Hung, C. -H. Huang.: "Efficient Indexing and Retrieval Scheme for VRML Database". IC CSCW, 2004.

[15] C. M. Cyr, B. B. Kimia: "3D Object Recognition Using Shape Similarity-Based Aspect Graph". ICCV, 2001.

[16] D. -Y. Chen, X. -P. Tian, Y. -T. Shen, M. Ouhyoung: "On Visual Similarity Based 3D Model Retrieval". Eurographics, 2003.

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

[18] D. V. Vranić: "3D Model Retrieval". Ph.D. Dissertation, Universität Leipzig, 2004.

[19] S. Akbar, J. Küng, R. Wagner: "Multi-feature based 3D Model Similarity Retrieval". International Conference on Computing and Informatics, Malaysia, 2006.

[20] E. Paquet, M. Rioux, A. Murching, T. Naveen, A. Tabatai: "Description of Shape Information for 2-D and 3-D Objects". Signal Processing Image Comm., Elsevier Science B.V., 2000.

[21] I. Atmosukarto, W. K. Leow, Z. Huang: "Feature Combination and Relevance Feedback for 3D Model Retrieval". MMM, pp. 334-339, 11th International Multimedia Modeling Conference, Australia, 2005.

[22] Q. Iqbal, J.K. Aggarwal: "Feature Integration, Multi-image Queries and RF in Image Retrieval". 6th Int'l Conf. on Visual Information System 2003, pp. 467-474, Florida 2003.

[23] B. Bustos, D. Keim, D. Saupe, T. Schrech, D. Vranić: "Automatic Selection and Combination of Descriptors for Effective 3D Similarity Search". IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04) pp. 514-521, 2004.

[24] P. J. Kostelec, D. R. Rockmore: "S2Kit: A Lite Version of SpharmonicKit". Department of Mathematics, Dartmouth College, 2004.

[25] F. Murtagh: The source code of Principal Component Analysis, The Departement of Statistics, Carnegie Mellon University. *http://lib.stat.cmu.edu/multi/pca.c*. Last Access: December 2005.

[26] S. Akbar, J. Küng, R. Wagner: "Multi-feature Integration on 3D Model Similarity Retrieval". The 1st International Conference on Digital Information Management (ICDIM), India 2006.

[27] Princeton Shape Retrieval and Analysis Group: "3D Model Search Engine". *http://shape.cs.princeton.edu /search.html*. Last access: September 2005.

[28] D. V. Vranić: "Content-based Classification of 3D-models by Capturing spatial Characteristics". *http://merkur01.inf. uni-konstanz.de/CCCC/*. Last access: September 2005.

[29] Utrecht University Object Database, *http://www.cs.uu.nl/ centers/give/imaging/3drecog/ 3dmatching.html*. Last Access: September 2005.

[30] J.J. Rocchio Jr.: "Relevance Feedback in Information Retrieval", The SMART System, G. Salton, ed., pp. 313-323, Prentice Hall, 1971.

[31] Georgy Gimel'farb. CBIR: Indexing and Retrieval. http://www.cs.auckland.ac.nz/ compsci708s1c/lectures/Glect-html/topic5c708FSC.htm. 2005.

[32] X. S. Zhou, T. S. Huang: "Small Sample Learning during Multimedia Retrieval using BiasMap". CVPR (1). pp 11-17. 2001.

[33] H. Y. Bang, T. Chen: "Feature Space Warping: An Approach to Relevance Feedback". In ICIP 2002.

[34] A. D. Bimbo, P. Pala: "Content-Based Retrieval of 3D Models". ACM Transactions on Multimedia Computing, Comm. and Applications, Vol. 2, No. 1, pp. 20-43, February 2006.

[35] K. Porkaew, K. Chakrabarti: "Query refinement for multimedia similarity retrieval in MARS", Proceedings of the 7th ACM int'l conference on Multimedia, pp: 235 – 238. 1999.

[36] X. Jin, J. C. French. "Improving Image Retrieval Effectiveness via Multiple Queries". Proc. of the 1st ACM international workshop on Multimedia databases, pp. 86-93, 2003.