Relevance Feedback in Content-based 3D Object Retrieval
A Comparative Study

Panagiotis Papadakis[1,2], Ioannis Pratikakis[1], Theodore Trafalis[3], Theoharis Theoharis[2] and Stavros Perantonis[1]

[1]National Center for Scientific Research "Demokritos", {ppapadak, ipratika, sper}@iit.demokritos.gr
[2]National and Capodistrean University of Athens, theotheo@di.uoa.gr
[3]University of Oklahoma, ttrafalis@ou.edu

ABSTRACT

In this paper, we present a comparative study that concerns relevance feedback (RF) algorithms in the context of content-based 3D object retrieval. In this study, we employ RF algorithms which range from query modification and multiple queries to one-class support vector machines (SVM). Furthermore, we employ pseudo relevance feedback (PRF) and show that it can considerably improve the performance of content-based retrieval. Our comparative study is based upon extensive experiments that take into account datasets containing generic as well as CAD models.

1. INTRODUCTION

When developing a content-based information retrieval system, the classical approach is to focus, primarily, on the selection of features that are best suited for characterizing the information to be retrieved. If the selected features are suited for information retrieval in agreement to the user's criteria, then the user is satisfied and the purpose of the retrieval system is fulfilled. This is primarily feasible in specialized applications where the semantics of the information to be retrieved can be captured unambiguously by a set of low-level features, i.e. the notion of information similarity is rigid rather than subjective. However, in the case where the semantics of information are more complex, the dependence of similarity estimation from human subjectivity increases which poses significant limitations in the performance of a static retrieval system. This is due to its inability to adapt to the varying information interpretations of different users or changes in the preferences of a single user as well as the semantic gap problem which implies the lack of correspondence between the low-level features used by the machine and the high-level, semantic information used by the human.

To bridge the semantic gap and enable the machine to retrieve information through adapting to individual categorization criteria, relevance feedback was introduced as a mean to involve the user in the retrieval process and guide the retrieval system towards the target. Relevance feedback was first used to improve text retrieval [19], later on successfully employed in image retrieval systems [6],[9],[10],[15],[18],[20] and lately in 3D object retrieval systems [1],[3],[8],[13]. It is the information of relevance with respect to a subset of the retrieved results, acquired from the user's interaction with the retrieval system.

Generally, the RF procedure can be described as follows. The system uses a set of low-level features to present a list of results according to the similarity with respect to an initial query. Then, the user browses through the retrieval list and provides information to the system with respect to the relevance of a subset of the results. Using the relevance feedback, the retrieval system adjusts its parameters in order to match optimally the user's classification criteria. Then, based on the adjusted parameter setting, a new retrieval session is initiated, new results are presented to the user and the above procedure is repeated until the user is satisfied.

Pseudo relevance feedback, also known as local or blind relevance feedback, is different from the conventional approach in that the user does not provide any relevance feedback at all. Instead, the required information is obtained based only on the unsupervised retrieval result. The procedure comprises two steps. First, the user submits a query to the system which uses the low-level features to produce a ranked list of results which is not

displayed to the user. Second, the system reconfigures itself by only using the top m closest matches of the list, based on the assumption that they are relevant to the user's query.

In this paper, we focus on the usage of relevance feedback for the case of 3D object retrieval and experiment on the performance of RF algorithms which range from query modification and multiple queries to one-class support vector machines (SVM). Furthermore, we employ pseudo relevance feedback and show that it can considerably enhance the retrieval performance. We present our results through an extensive evaluation in datasets containing generic as well as CAD models.

2. RELATED WORK

Most of the techniques that have been proposed to exploit the relevance feedback information can be classified into the following categories:

(a) *Query modification - refinement – reformulation*; in this RF technique, a new query is constructed using the feature vectors of the objects that the user considers as relevant. The new query will be closer in feature space to the human target thus, in the next query iteration, more relevant objects will be included in the list of results.

(b) *Feature space transformation*; the system learns the importance of the feature space dimensions(assuming uncorrelated features), by measuring the contribution of each dimension in distinguishing the objects that the user considers relevant from the rest of the objects. Then, the feature space is transformed into a new space where the relevant objects are more closely clustered and the successive query is performed in the transformed space.

(c) *Clustering of labeled results*; the objects that are labeled as relevant by the user are considered to form a cluster in the feature space. Additional clusters may be formed in the case where the user's relevance feedback provides various degrees of relevance for the labeled objects or in the case were objects marked as irrelevant belong to multiple classes. Subsequently, the similarity between the query and the objects of the database is re-evaluated by considering the relative position of each feature vector with respect to the clusters.

(d) *Statistical approach*; the ranking of the objects in the retrieval list is determined using their probability to be relevant to the query as estimated from the user's relevance feedback.

Currently, there has been very limited research in the area of relevance feedback for 3D object retrieval and most efforts were based on combining the aforementioned categories. Elad et al. [8], were the first to incorporate relevance feedback with 3D object retrieval. Their approach was to learn the distance function so as to optimally separate the relevant from the irrelevant results by an appropriate margin which was computed by solving a quadratic optimization problem. In the work of Bang et al. [3], the descriptors of the unlabeled visual objects move toward or away from the query, according to their relative position in the feature space, with respect to the relevant and irrelevant examples. The authors call the proposed approach feature space warping. Leifman et al. [12], used Linear Discriminant Analysis (LDA) [7] combined with Biased Discriminant Analysis (BDA) [24]. In the hybrid scheme, LDA was used when the user's relevance feedback was limited since it was found to perform better than BDA in that case, while BDA performed better for larger numbers of labeled results. In the work of Lou et al. [13], the system moves the query point to the centroid of the relevant results and re-weights the feature dimensions based on their contribution in discriminating the relevant from the irrelevant results, by measuring the features variance. In the approach of Atmosukarto et al. [2], the system learned the importance of different feature representations and pushed possible irrelevant models to the bottom of the retrieval list. The importance of each feature representation was determined by estimating the ratio of the number of the relevant models over the total number of models within the hyper-sphere spanned by the relevant models. The unlabeled models that lied close to labeled irrelevant models where considered irrelevant as well and were pushed to the bottom of the list, thus leaving the possible relevant models close to the top of the list. In the work of Akbar et al. [1], the system learns the importance of the multiple features of the visual descriptor using the variance of the distances of the relevant objects with respect to each other as well as the respective rank positions for each distinctive feature. Then, the distances of the unlabeled models with respect to the query, that are close to the irrelevant models are set to the maximum value while in the opposite case the distance is set to the minimum distance from the set of the relevant models. Novotni et al. [16] evaluate the performance of kernel-based methods for relevance feedback in 3D model retrieval including SVM, one-class SVM [5], BDA [24] and KBDA. In their experiments, using 3D Zernike descriptors as features, they conclude that the SVM performed the best just above KBDA.

Pseudo relevance feedback was first employed for text retrieval [14][4] in order to automatically expand the query with additional keywords taken from a set of top ranked documents which were assumed to be relevant to the query. To our knowledge, this is the first time that pseudo relevance feedback is used for 3D object retrieval.

3. DETAILED DESCRIPTION OF METHODS
This Section is dedicated to the detailed description of the RF methods that we examine (Section 3.1-3.3), namely One-class SVM, Query Modification and Multiple Queries, as well as the description of a pseudo-relevance scheme (Section 3.4) that is further encountered in the evaluation stage.

It is worth noting that the selection of the RF methods used for our comparative study do not require the user *to select irrelevant to the target retrieved objects but only relevant to them*. This choice may avoid certain requirements which are set when irrelevant objects are selected. These requirements comprise (i) selection of large training datasets for irrelevant objects since selection of small training datasets is not representative of the whole population; (ii) weighing of the objects that are labeled as irrelevant since they belong to more than one classes and therefore should not be equally treated.

Furthermore, we should note that requiring the user to select only positive examples becomes a less heavy task since selection of irrelevant objects can become complicated due to a greater amount of user interaction and mental effort that is required.

3.1 One-Class SVM
One-class SVM (OC-SVM) is a novelty detection algorithm [21] that was first used within a relevance feedback framework by Chen et al. [5] for image retrieval. The algorithm is given a set of observations (feature vectors) that all belong to the target class and uses them to find a hyper-sphere in feature space that contains most of the observations while having the minimum possible radius. The surface of the hyper-sphere plays the role of the decision margin, that is, patterns that lie within the hyper-sphere are considered to belong to the target's distribution and those that lie outside the hyper-sphere are considered irrelevant. In our case, we require all objects of the database to be ranked based on their similarity to the target class therefore each object is ranked according to its distance from the hyper-sphere's center. In the following we give the detailed description of the OC-SVM classification algorithm.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ denote a set of observations that belong to the target class, where m is the number of observations and $\mathbf{x}_i \in T$ where $T \subseteq \mathbb{R}^N$. Let Φ be a feature mapping from T to a dot product space H and $k(\mathbf{x}, \mathbf{y})$ a kernel function such that $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$, where $\langle .,. \rangle$ denotes the dot product in H.

We want to compute a hyper-sphere with center $\mathbf{c}$ and radius R that contains the maximum number of observations $\mathbf{x}_i$ while having the minimum possible radius. This is formulated as an optimization problem described as follows:

$$\underset{\mathbf{c} \in H, R \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^m}{\text{minimize}} R^2 + \frac{1}{vm} \sum_{i=1}^{m} \xi_i, \text{ subject to } \left\| \Phi(\mathbf{x}_i) - \mathbf{c} \right\|^2 \leq R^2 + \xi_i \text{ and } \xi_i \geq 0, i = 1, 2, ..., m. \quad (3.1)$$

where ξ_i denote the "slack" variables which are introduced to permit the exclusion from the hyper-sphere's interior of outlier observations. If all observations were to be explicitly bounded by the hyper-sphere's surface then possible outliers would increase the radius of the hyper-sphere and therefore limit the generalization ability of the algorithm. The parameter $v \in (0,1]$ is used to control the amount of slack. By decreasing the value of v, we increase the tolerance of the algorithm to outliers resembling a hard margin classifier, while increasing the value of v we achieve the opposite effect, resembling a soft margin classifier.

To solve the problem, we convert the objective function to its dual form and derive the Lagrangian, from which we get the dual problem:

$$\underset{\boldsymbol{\alpha}}{\text{minimize}} \sum_{i,j=1}^{m} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m} \alpha_i k(\mathbf{x}_i, \mathbf{x}_i), \text{ subject to } 0 \leq \alpha_i \leq \frac{1}{vm} \text{ and } \sum_{i=1}^{m} \alpha_i = 1. \quad (3.2)$$

Eqn. (3.2) is solved through quadratic program optimization and gives the coefficients α_i from which we derive the center $\mathbf{c}$ of the hyper-sphere.

The final decision function is given by:

$$f(\mathbf{x}) = \mathrm{sgn}\left(R^2 - (\mathbf{c} - \Phi(\mathbf{x}))^2\right) = \mathrm{sgn}\left(R^2 - \mathbf{c}^2 + 2\mathbf{c}\Phi(\mathbf{x}) - [\Phi(\mathbf{x})]^2\right) =$$

$$= \mathrm{sgn}\left(R^2 - \left[\sum_{i=1}^{m}\alpha_i\Phi(\mathbf{x}_i)\right]^2 + 2\sum_{i=1}^{m}\alpha_i\Phi(\mathbf{x}_i)\Phi(\mathbf{x}) - [\Phi(\mathbf{x})]^2\right)$$

$$= \mathrm{sgn}\left(R^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j) + 2\sum_{i=1}^{m}\alpha_i\Phi(\mathbf{x}_i)\Phi(\mathbf{x}) - \Phi(\mathbf{x})\Phi(\mathbf{x})\right) \quad (3.3)$$

$$= \mathrm{sgn}\left(R^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) + 2\sum_{i=1}^{m}\alpha_i k(\mathbf{x}_i,\mathbf{x}) - k(\mathbf{x},\mathbf{x})\right)$$

If $f(\mathbf{x}) = +1$ then $\mathbf{x}$ lies inside the hyper-sphere and therefore is considered to belong to the target class, while if $f(\mathbf{x}) = -1$, $\mathbf{x}$ lies outside the hyper-sphere and is not considered to belong to the target class.

To compute the radius of the hyper-sphere we solve the equation $f(\mathbf{x}_i) = 0$ where $\mathbf{x}_i$, is a support vector, i.e. $\alpha_i \neq 0$. For a support vector $\mathbf{x}_i$, $(\mathbf{c} - \Phi(\mathbf{x}_i))$ will be equal to the radius R of the hyper-sphere since all support vectors lie on the decision surface.

In our case, we rank all objects of a database according to their distance from the hyper-sphere's center $\mathbf{c}$, thus an unlabeled object of the database in ranked using a modified version of $f(\mathbf{x})$ which is given by:

$$f'(\mathbf{x}) = -\sum_{i,j=1}^{m}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) + 2\sum_{i=1}^{m}\alpha_i k(\mathbf{x}_i,\mathbf{x}) - k(\mathbf{x},\mathbf{x}) \quad (3.4)$$

If $f'(\mathbf{x}) = 0$, then $\mathbf{x}$ coincides with the center of the hyper-sphere, thus exhibiting the highest possible similarity with the target class. As $f'(\mathbf{x})$ increases, objects are considered less similar to the target class and are ranked further from the top of the retrieval list.

3.2 Query Modification
The query modification technique is used to move the query closer to the objects that belong to the target class, thus improving a nearest neighbor search by retrieving more relevant objects on the top of the list. This is based on the assumption that the initial query is probably a rough representation of what the user is searching for and therefore when relevance feedback is supplied to the system then a new query can be found that is closer in feature space to the user's target. In the following, we give the detailed description of the query modification algorithm.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ denote a set of observations belonging to the target class where feature vectors $\mathbf{x}_i$ correspond to the objects that are labeled as relevant from the user, based on the retrieval results that were acquired from an initial query $\mathbf{x}_q$. We compute a modified query $\mathbf{x}_{q'}$ as the average of $\mathbf{x}_q$ and all $\mathbf{x}_i$, given by:

$$\mathbf{x}_{q'} = \frac{\mathbf{x}_q + \sum_{i=1}^{m}\mathbf{x}_i}{1 + m} \quad (3.5)$$

The new query $\mathbf{x}_{q'}$ is the centroid of the initial query and all relevant objects and thus, it is a better representation of the user's target. The unlabeled objects of the database are then ranked according to their similarity against the modified query $\mathbf{x}_{q'}$.

3.3 Multiple Queries
An alternative approach to perform RF in content-based retrieval is to perform several queries and combine the individual results to a single output. This technique is based on the premise that a single query may be insufficient to express the user's target and suggests using multiple queries to search for the same target. In the following, we give the detailed description of the multiple queries algorithm.

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_m\}$ denote a set of observations belonging to the target class where feature vectors $\mathbf{x}_i$ correspond to the objects that are labeled as relevant from the user, based on the retrieval results that were acquired from an initial query $\mathbf{x}_q$. Since the objects with feature vectors $\mathbf{x}_i$ match the user's target we perform m additional queries using the objects marked as relevant to the initial query. After the completion of this step we obtain m ranked lists of objects.

We combine the results of each individual query according to the following scheme. Let $dist(\mathbf{x}_j, \mathbf{x}_k)$ denote the distance between two objects O_j, O_k. An unlabeled object O_j of the database is ranked according to its overall similarity from the set of m queries, denoted as $sim(O_j)$ which is computed as:

$$sim(O_j) = \frac{1}{m} \sum_{k=1}^{m} dist(\mathbf{x}_j, \mathbf{x}_k) \tag{3.6}$$

3.4 Pseudo Relevance Feedback

In pseudo relevance feedback (PRF), the top m closest matches are considered as relevant to the query and are consequently used as training data. The relevance assumption for the top m closest matches is not always valid and irrelevant retrieved objects that may appear in the top of the list will be erroneously considered as relevant. This case is known as "query drift" phenomenon and implies the scenario where the retrieval system is misled by the irrelevant data and drawn away from the user's target. However, if the low level features are discriminative enough to bring a large portion of the relevant objects to the top of the list, then the relevance assumption will be valid in the majority of queries thus the average performance will improve. This means that PRF is mostly useful at increasing the performance in the case of features that already exhibit good discrimination ability. Otherwise, in the case where the features used are not discriminative enough, PRF may actually decrease the overall performance of retrieval.

The pseudo relevance feedback technique that we use comprises an off-line and an on-line stage. Both stages are based on the idea of moving the feature vector of an object closer to its cluster centroid in feature space, thus improving the performance of a nearest neighbor search, which is similar in spirit to the query modification technique. In the following, we give the detailed description of each stage:

Off-line stage
Let n be the number of objects in the database and $\mathbf{x}_i$ denote the feature vector of an object O_i where i = 1, 2, ..., n. We use each object as a query in the remaining set of n−1 objects of the database and take n ranked lists of results denoted as *list*$_i$. For each O_i, we take the descriptors of the top m closest matches from the corresponding *list*$_i$, which are assumed to be relevant to O_i. Then we update the descriptor $\mathbf{x}_i$ according to the following formula:

$$\mathbf{x}_i' = \frac{\mathbf{x}_i + \mathbf{x}_i^{rel}}{2} \tag{3.7}$$

which denotes the average feature vector between the original descriptor $\mathbf{x}_i$ of the object and a new feature vector $\mathbf{x}_i^{rel}$ which is computed using the corresponding top m closest matches and is given by:

$$\mathbf{x}_i^{rel} = \frac{\sum_{j=1}^{m} \frac{1}{j+1} * \mathbf{x}_{i,j}}{\sum_{j=1}^{m} \frac{1}{j+1}} \tag{3.8}$$

where $\mathbf{x}_{i,j}$ is the jth closest match to object O_i. The $\mathbf{x}_i^{rel}$ feature vector is computed by averaging the top m descriptors $\mathbf{x}_{i,j}$, where the discounting factor $\frac{1}{j+1}$ is used to weigh the closest matches inversely proportional to their rank in the retrieval list. This weighing together with the fact that the updated descriptor is the average of the original $\mathbf{x}_i$ and $\mathbf{x}_i^{rel}$, reduces the effect of the "query drift" phenomenon, i.e. when the top m closest matches include irrelevant objects as well.

On-line stage

Upon submission of a query, the system computes the descriptor of the query $\mathbf{x}_q$ and compares it to the updated descriptors $\mathbf{x}_i'$ of the objects of the database. This produces a sorted list of results $list_q$ from which the top m closest matches are used to update the descriptor of the query as in Eqn. (3.7) and Eqn. (3.8). In the sequel, a new retrieval session is initiated using the updated descriptor of the query $\mathbf{x}_q'$, which is closer in feature space to its cluster-class centroid and the results are shown to the user in decreasing order of similarity. The number m of the closest matches that are considered as relevant to the query is determined by the expected recall of the employed low level features, near the top of the retrieval list. The higher the recall the more closest matches we expect to be relevant to the query.

Employing the above pseudo relevance feedback technique, introduces an additional cost in the time complexity of the overall retrieval procedure. In particular, the time required for the completion of the retrieval process is doubled. This is due to the execution of two retrieval sessions, the first using the original descriptor of the query and the second using the updated descriptor of the query after employing the pseudo relevance feedback.

4. EXPERIMENTAL RESULTS
In this section, we present the performance evaluation of the relevance feedback algorithms taken into consideration, namely, one-class SVM (OC-SVM), query modification (Q-Mod) and multiple queries (Mul-Q) along with the employed pseudo relevance feedback scheme (Pseudo-RF).

As a 3D object signature, we use the CRSP descriptor [17] which attains top performance along with low time requirements. CRSP captures the global shape of a 3D object using a set of concentric spherical shells where the object's surface is projected and the resulting spherical functions are expressed through their spherical harmonic representation. It is a shape signature that is discriminative enough to employ PRF and its low time requirements enable the application of RF which poses significant time constraints since it is an on-line procedure.

To evaluate the performance of the RF algorithms we have used the following datasets:

i.   The dataset of the Princeton Shape Benchmark (PSB) that contains generic 3D models [22].
ii.   The dataset of the Watertight Models Track of SHREC 2007 (WM-SHREC) [23].
iii. The dataset of the Engineering Shape Benchmark (ESB) that contains CAD models [11].

To evaluate the performance in the PSB dataset, we have used the "coarse2" classification, which contains 7 classes, namely, "vehicle", "animal", "household", "building", "furniture", "plant" and "-1" where the "-1" class contains those models that do not belong to any of the remaining 6 classes. We removed the models belonging to the "-1" class and evaluated the performance in the remaining dataset consisting of 1593 models where each class contains 266 models on average. The ESB dataset contains 866 models which are classified into 45 classes with 19 models per class on average and the WM-SHREC dataset contains 400 models which are equally distributed in 20 classes. The performance in each dataset was measured using the leave-one-out method, i.e. using each object of a dataset as a query in turn and evaluating the performance in the remaining set of objects.

The objects that are labeled as relevant and are consequently used as training data are those that appear within the top k results and belong to the query's class, according to the classification file of the corresponding dataset. For the PSB dataset, we set k=40 while for the ESB and WM-SHREC datasets we set k=20. The maximum number of objects that are labeled as relevant within the first k results is set to m=10 objects per feedback iteration for the PSB dataset and m=4 objects per feedback iteration for the ESB and WM-SHREC datasets. For the OC-SVM method we used the Gaussian kernel with $\gamma = 30$ and set the amount of slack to $v = 0.5$. To depict the performance we use the precision and recall evaluation measures. Recall is the ratio of relevant retrieved models to the total number of relevant models while precision is the ratio of relevant retrieved models to the number of retrieved models. In Fig. 1, we compare the performance of all relevance feedback techniques including the pseudo relevance scheme (Pseudo RF) for all three datasets. For the methods that require the user's interaction (OC-SVM, Q-Mod, Mul-Q) the performance is depicted at the 3rd (#3) iteration of relevance feedback.
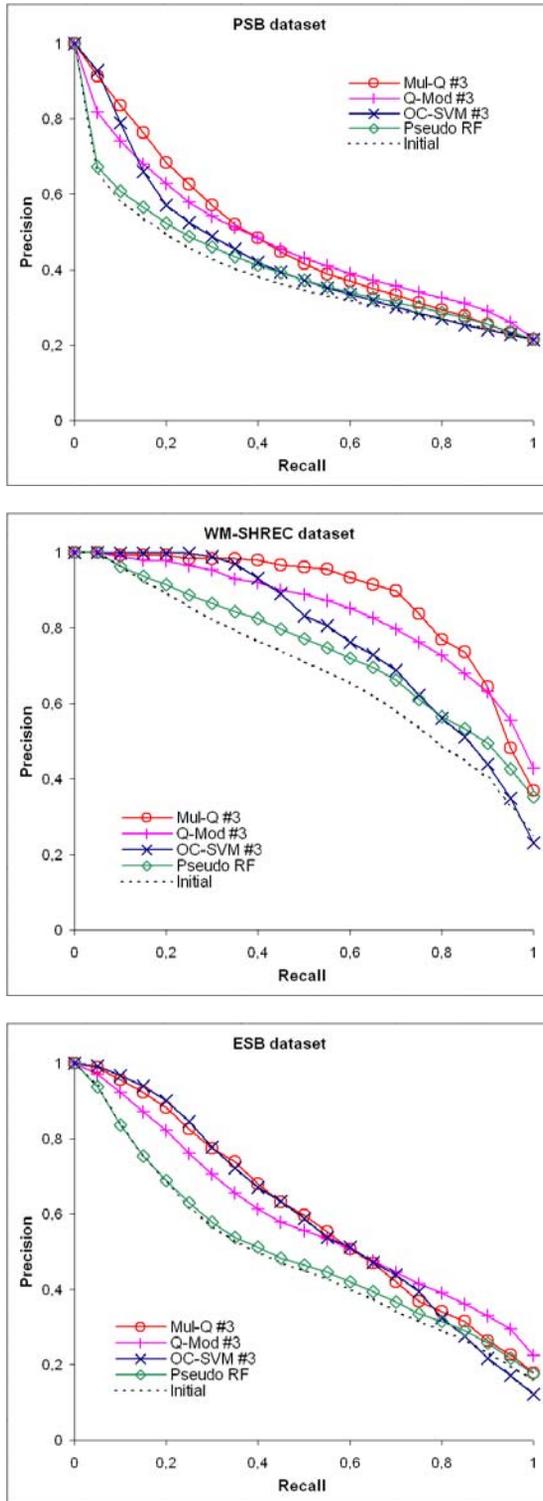
Fig. 1: Evaluation of OC-SVM, Q-Mod, Mul-Q and Pseudo RF for the PSB, WM-SHREC and ESB dataset. For the relevance feedback techniques that require the user's interaction, the performance at the 3rd iteration (#3) is used.

The results shown in Fig. 1 indicate that the Mul-Q method has the best overall performance. Moreover the OC-SVM method seems to achieve the highest precision values for small values of recall but this is not consistent for higher values of recall. Not surprisingly, the Pseudo RF method is ranked last which is due to the fact that the training data are not provided by the user but are taken automatically from the top of the retrieval list based on the assumption that they are relevant. However, the results show that Pseudo RF is an approach that can give a considerable gain in the overall retrieval performance, which is more obvious in the WM-SHREC dataset where the average precision has increased by 5.4%. In the remaining datasets the overall increase in the average precision is approximately 1.7%.

In Fig. 2, we evaluate the performance of each method for different numbers of top k results from which the relevant objects are chosen, namely k=20, k=30 and k=40, using the PSB dataset. The horizontal axis shows the relevance feedback iteration and the vertical axis shows the average precision for recall in the range of (0%-50%). For the ith feedback iteration (i>1), we use not only the new labeled objects but those objects that were labeled as relevant from previous iterations as well.
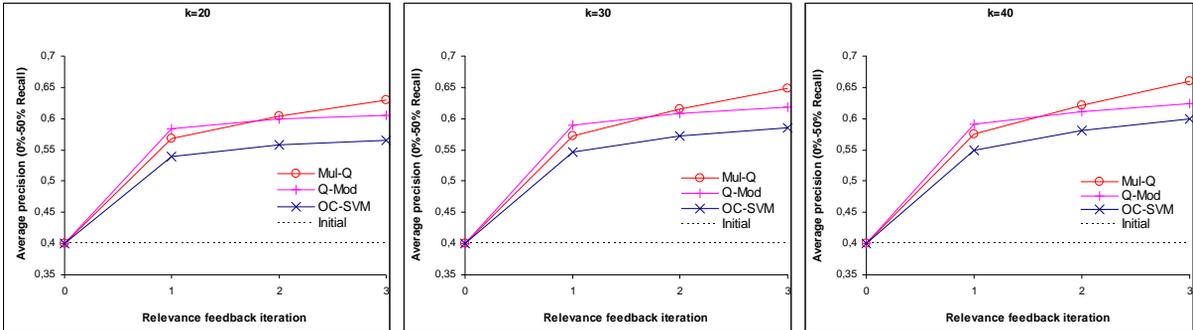
Fig. 2: Evaluation of OC-SVM, Q-Mod and Mul-Q within the PSB dataset, for different numbers of top k results (k=20, 30, 40) from which the relevant objects are chosen, across consecutive relevance feedback iterations using the average precision for recall in the range of (0%-50%).

From Fig. 2, we can conclude that the Mul-Q and Q-Mod methods have the best performance and that OC-SVM is ranked last. Interestingly, the Q-Mod method performs better than Mul-Q in the first relevance feedback iteration, but it does not show significant increase in performance in the following iterations. On the contrary, the performance of the Mul-Q method exhibits a greater increase in performance and surpasses the Q-Mod method after the 2nd iteration.

In Fig. 3, we give some example queries within the ESB dataset and the corresponding results first by using the low-level features and then by employing relevance feedback using the Mul-Q method. We show example queries from the ESB dataset due to the fact that objects have been primarily classified according to the object's functionality while low-level features are used to measure the geometric similarity between objects. This means that objects with very similar shapes may belong to different categories. This is a characteristic of the ESB dataset and in many cases it is impossible to discriminate such objects by only using a set of low-level features and the use of relevance feedback becomes imperative in order to improve the retrieval results.

The example shown in Fig. 3(a) illustrates an ordinary query where retrieval precision gradually increases across consecutive relevance feedback iterations, while the examples shown in Fig. 3(b)-(c) illustrate queries where after the 2nd relevance feedback iteration the retrieval accuracy is excellent.
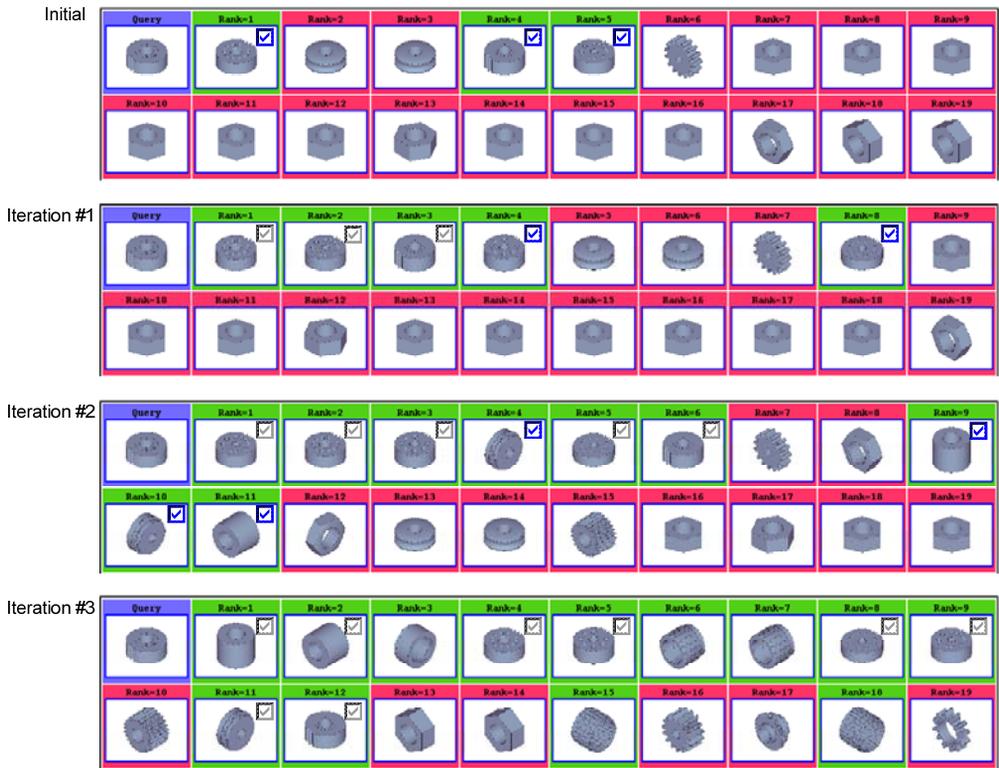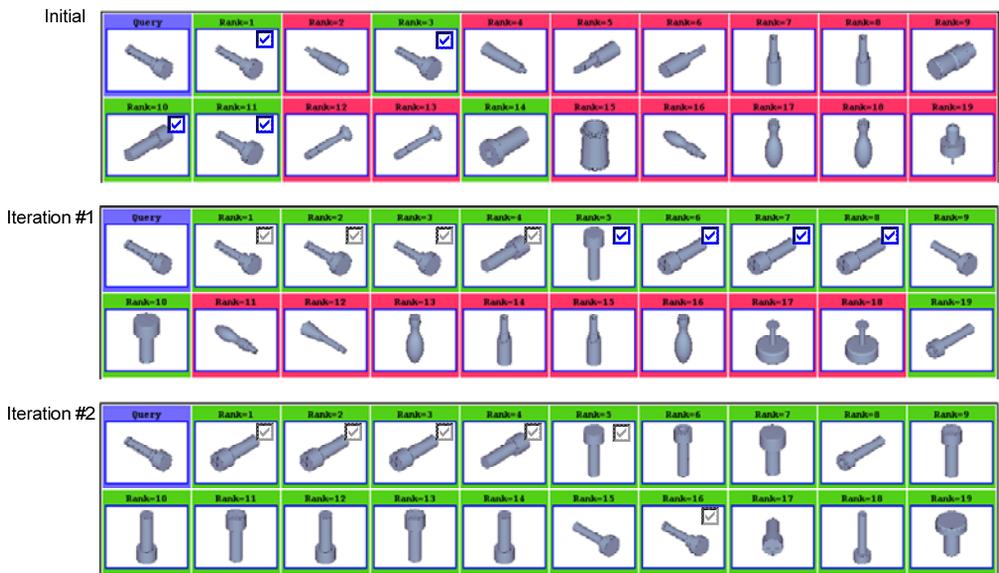
Fig. 3: Example queries within the ESB dataset with the corresponding retrieval results using the low-level features (Initial) and the Mul-Q method after successive relevance feedback iterations. The query is shown in blue color, the models that belong to the query's class are shown in green and the remaining models are shown in red. The models that are labeled as relevant by the user in the current iteration are indicated with a blue check box, while those that were labeled in previous iterations are indicated with a gray checkbox.
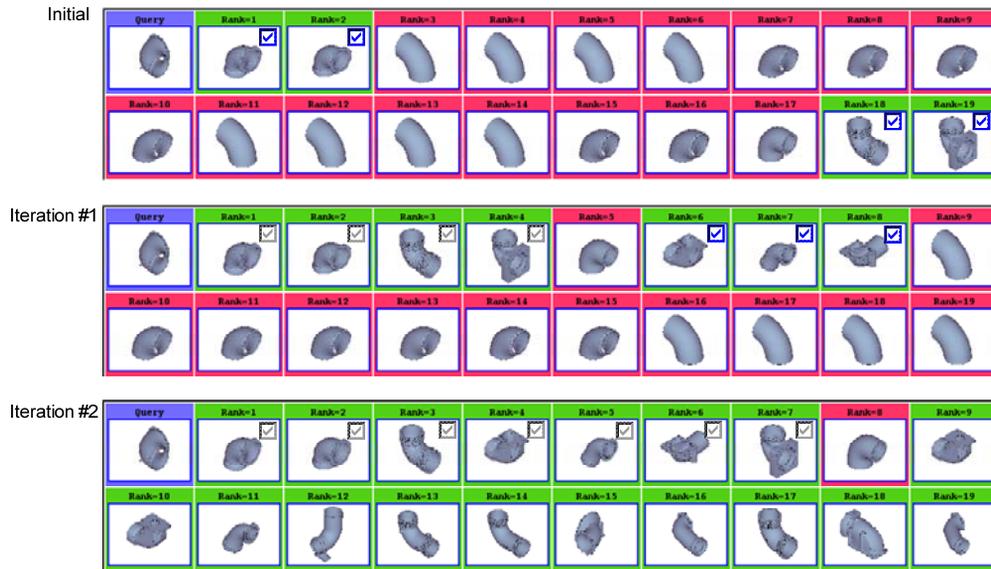
(c)

Fig. 3 (continued).

5. CONCLUSIONS

In this paper we evaluate the performance of a set of standard relevance feedback techniques, namely, one-class SVM, query modification and multiple queries for content-based 3D object retrieval. The data required to train each method are obtained from positive feedback, i.e. the user only labels a set of objects that are relevant to the target, in order to minimize the user's interaction with the system and maintain his/her attention to the target class. The multiple queries method exhibits the best overall performance showing a faster rate of improvement across consecutive relevance feedback iterations. In addition, we employ a pseudo relevance feedback scheme that shifts the feature vector of an object towards the centroid of its neighborhood in feature space and show that it can add a considerable gain in the overall retrieval performance.

6. ACKNOWLEDGEMENT

7. REFERENCES

[1]   Akbar, S.; Kung, J.; Wagner, R.; Prihatmanto, A. S.: Multi-Feature Integration with Relevance Feedback on 3D Model Similarity Retrieval, Int. Conf. on Information Integration and Web-based Applications Services, 2006, 77-86.

[2]   Atmosukarto, I.; Kheng, W. L.; Huang, Z.: Feature Combination and Relevance Feedback for 3D Model Retrieval, 11th Int. Conf. on Multimedia Modeling, 2005, 334-339.

[3]   Bang, H.; Chen, T.: Feature Space Warping: An Approach to Relevance Feedback, Int. Conf. on Image Processing, 2002.

[4]   Buckley, C.; Salton, G.; Allan, J.: Singhal, A.: Automatic Query Expansion using SMART: TREC 3, TREC, 1994.

[5]   Chen, Y; Zhou, X.S.; Huand, T.: One-class SVM for Learning in Image Retrieval, Int. Conf. on Image Processing, 2001, 34-37.

[6]   Cox, I.; Miller, M.; Minka, T.; Yianilos, P.: An Optimized Interaction Strategy for Bayesian Relevance Feedback, Int. Conf. on Computer Vision and Pattern Recognition, 1998.

[7]   Duda, R.; Hart, P.; Stork, D.: Pattern Classification, John Wiley & Sons, New York.

[8]   Elad, M.; Tal, A.; Ar, S.: Content based retrieval of VRML objects: an iterative and interactive approach, Proceedings of the 6th Eurographics workshop on Multimedia, 2001.

[9]   Giacinto, G.; Roli, F.: Bayesian Relevance Feedback for Content-Based Image Retrieval, J. Pattern Recognition, 37(7), 2004.

[10] Ishikawa, Y.; Subramanya, R.; Faloutsos, C.: MindReader Querying Databases through Multiple Examples, Int. Conf. on Very Large Databases, 1998.

[11] Jayanti, S.; Kalyanaraman, Y.; Iyer, N.; Ramani, K.: Developping an Engineering Shape Benchmark for CAD Models, Computer-Aided Design, 38(9), 2006, 939-953.

[12] Leifman, G.; Meir, R.; Tal, A.: Semantic-oriented 3d shape retrieval using relevance feedback, J. Visual Computer, (21) 8-10, 2005, 865-875.

[13] Lou, K.; Jayanti. S.; Iyer, N.; Kalyanaraman, Y.; Prabhakar, S.; Ramani K.: A Reconfigurable 3D Engineering Shape Search System Part II: Database Indexing, Retrieval and Clustering, 23rd Computers and Information in Engineering (CIE) Conference, 2003.

[14] Mitra, M.; Singhal, A.; Buckley, C.: Improving Automatic Query Expansion, 21st Int.l ACM SIGIR Conf. on Research and development in information retrieval, 1998, 206-214.

[15] Nastar, C.; Mitschke, M.; Meilhac, C.: Efficient Query Refinement for Image Retrieval, Int. Conf. on Computer Vision and Pattern Recognition, 1998.

[16] Novotni, M.; Park, G.-J.; Wessel, R.; Klein, R.: Evaluation of Kernel Based Methods for Relevance Feedback in 3D Shape Retrieval, 4th Int. Workshop on Content-Based Multimedia Indexing, 2005.

[17] Papadakis, P.; Pratikakis, I; Theoharis, T.; Perantonis, S.: Efficient 3D Shape Matching and Retrieval Using a Concrete Radialized Spherical Projection Representation, J. Pattern Recognition, 40(9), 2007, 2437-2452.

[18] Porkaew, K.; Chakrabarti, K.: Query Refinement for Multimedia Similarity Retrieval in MARS, ACM Int. Conf. on Multimedia (Part 1), 1999.

[19] Rocchio, J.: Relevance Feedback in Information Retrieval The SMART Retrieval System - Experiments in Automatic Document Processing, G. Salton ed., Chapter 14, 313-323, 1971.

[20] Rui, Y.; Huang, S.; Mehrotra, S.: Content-based image retrieval with relevance feedback in MARS, Int. Conf. on Image Processing, 1997.

[21] Scholkopf, B.; Smola, A.: Learning with Kernels, MIT Press, 2002.

[22] Shilane, P.; Min, P.; Kazhdan, M.; Funkhouser, T.: The Princeton Shape Benchmark, Shape Modeling International, 2004, 167-178.

[23] Veltkamp, R.; Haar F.: SHREC 2007 3D Shape Retrieval Contest, Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, 2007.

[24] Zhou, X.; Huang, T.: Small Sample Learning During Multimedia Retrieval using BiasMap, Int. Conf. on Computer Vision and Pattern Recognition, 2001.