Relevance Feedback for 3D Shape Retrieval*

George Leifman, Ron Meir, Ayellet Tal
{gleifman@tx,rmeir@ee,ayellet@ee}.technion.ac.il

Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa, Israel

Abstract

The last few years have witnessed an increasing interest in shape-based retrieval of 3D models for computer graphics applications. Object similarity is a subjective matter, dependent on the human viewer, since objects have semantics and are not mere geometric entities. Relevance feedback aims at addressing the subjectivity of similarity. This paper presents a novel relevance feedback algorithm which is based both on supervised and unsupervised feature extraction techniques. We show that the proposed approach produces good results and outperforms previously proposed techniques.

Keywords: 3D shape retrieval, 3D shape similarity, relevance feedback

Conference topics: Geometric algorithms, Internet related graphics and modeling

1 Introduction

Recent progress in digital data storage, computing power and modeling techniques have made large repositories of digital 3D objects increasingly more accessible. As a result, more applications require the ability to retrieve 3D models from databases, based on their shape similarity [Hilaga et al. 2001; Kazhdan et al. 2003; Osada et al. 2001; Paquet et al. 2000; Vranic et al. 2001].

Different users might have conflicting interpretations regarding similarity of objects, as illustrated in Figure 1. What is more similar to a centaur – a man or a horse? The subjectivity of similarity might make unsupervised retrieval infeasible. *Relevance feedback (RF)* lets the user incorporate his or her perceptual feedback in the search [Ishikawa et al. 1998; Salton and McGill 1983; Tieu and Viola 2000; Tong and Chang 2001]. The user can manually determine relevance between the query and the retrieved objects. The relevant / irrelevant objects are then used to refine the query results.

Relevance feedback retrieval schemes iterate the following three stages. First, the user is presented with a list of similar objects, in descending order of similarity. Next, the user provides feedback regarding the relevance of the current retrieval results. Finally, the system uses these examples to learn and to improve the performance in the next iteration.

Relevance feedback shields the user from the details of query reformulation because all the user has to provide is a relevance judgment on the results. Moreover, it breaks down the search task into a sequence of small steps which are easy to grasp and provides a controlled process designed to emphasize relevant features and deemphasize irrelevant ones.

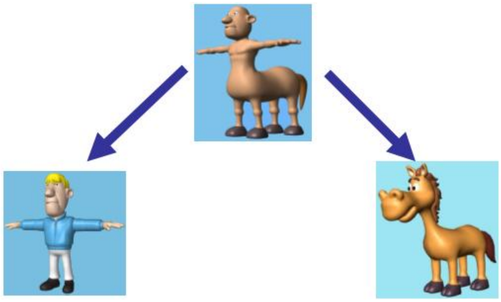


Figure 1: What is more similar to a centaur?

Relevance feedback algorithms should find an appropriate transformation that maps the original feature space into a new space, where the user-desired high-level concepts are better reflected [Aksoy et al. 2000; Ishikawa et al. 1998; Rui et al. 1998]. The major question is what can be learned from the examples the user provides. Some methods propose to learn the importance of various features [Peng et al. 1999; Santini and Jain 2000]. Others use density estimation to find the position of relevant examples in feature space [Chen et al. 2001; Meilhac and Nastar 1999]. Some recent approaches use learning or classification algorithms [Tieu and Viola 2000; Zhou and Hunag 2000].

In this paper we study relevance feedback for 3D models. A learning technique based on *Support Vector Machine (SVM)* was studied in [Elad et al. 2001]. A feature space warping approach was presented in [Bang and Chen 2002]. We propose a scheme which is based both on supervised and unsupervised feature extraction(i.e., [Peng et al. 1999; Santini and Jain 2000]). We show that our approach produces good results and outperforms those presented in [Elad et al. 2001] and [Bang and Chen 2002].

The rest of this paper is organized as follows. Section 2 provides a short overview of our system. Section 3 describes our relevance feedback algorithm. Section 4 presents some results. Finally, Section 5 concludes this paper.

2 System Overview

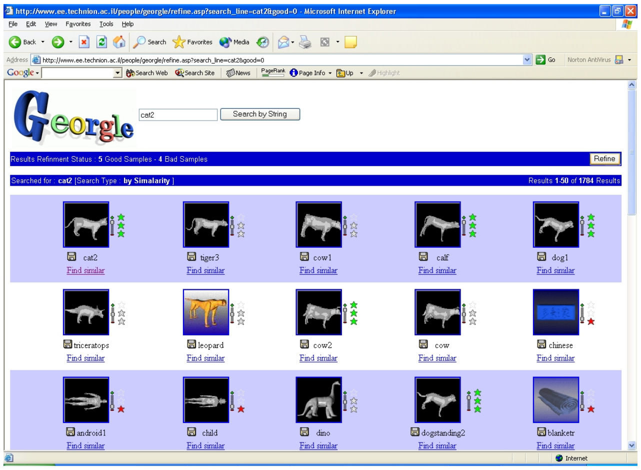
We have developed a search engine for 3D models that supports shape-based queries and lets the user provide perceptual feedback. The user can refine a query by marking objects as relevant or irrelevant.

Figure 2 shows some screen-shots of our 3D Web search engine. By pressing the *Find similar* button under any of the model icons, shape-based query search is invoked and similar shapes are presented to the user. The user can provide his feedback by pressing the *Refine Results* button and then marking relevant and irrelevant results with green and red stars, respectively. After providing the feedback and pressing the *Refine* button the system learns from these training examples, applies the relevance feedback algorithm and presents the improved results. This process can be repeated

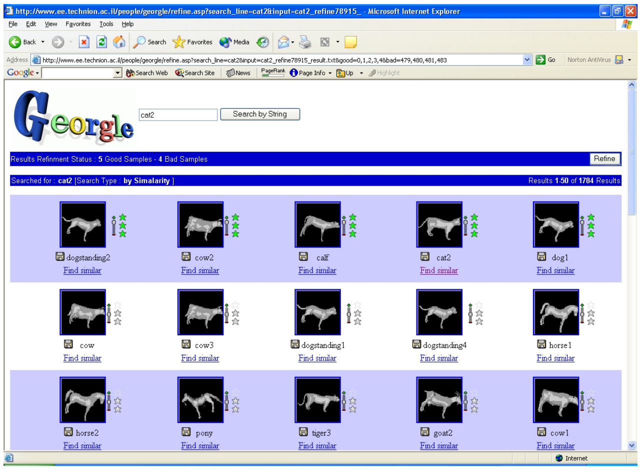
---

[1] This work was partially supported by European FP6 NoE grant 506766 (AIM@SHAPE) and by the Ollendorff foundation.

until the user achieves the desired results.



(a) Providing Feedback (green / red)



(b) Feedback Results

Figure 2: Screen-shots of the 3D Web search engine

In the next section we describe our relevance feedback algorithm. The algorithm is general and can be applied to any signature represented as a vector. In our case, we use the *sphere projection and topology* signatures described in [Leifman et al. 2003]. Briefly, this signature represents the distance from the unit sphere to the object and the distance from the object to the unit sphere, along with radii variance and Betti numbers.

3 Relevance Feedback Algorithm

Our relevance feedback algorithm consists of a pre-processing off-line stage and an online computation. During pre-processing, an *unsupervised feature extraction* technique is applied to the whole database. During a retrieval query, *query refinement* and *supervised feature extraction* are combined at each relevance feedback step.

Pre-processing: Given N observations on d variables, *feature extraction* refers to the reduction of the dimensionality of the data by finding r new variables, where $r \leq d$ [Fukunaga 1990]. The goal of this projection onto a lower dimensional manifold is to obtain a more efficient combination of the original features in the sense of estimation variance.

In *unsupervised feature extraction*, the relevance information is not used. The most widespread linear mapping is *Principal Component Analysis (PCA)* [Jolliffe 1986].

PCA finds a projection matrix W: $\mathbf{y} = W^\top \mathbf{x}$, where $\mathbf{y} \in \mathcal{R}^r$ is a transformed data point, W is a $d \times r$ transformation matrix and $\mathbf{x} \in \mathcal{R}^d$ is an original data point. PCA can be performed by diagonalizing the covariance matrix C of the original data: $C = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top$, where N is the number of observations, $\mathbf{x}_i$ is the i-th observation, and $\mathbf{m}$ is the mean vector of the input data. This is done by solving the eigenvalue equations $C\mathbf{v} = \lambda\mathbf{v}$ for eigenvalues $\lambda \geq 0$ and nonzero eigenvectors $\mathbf{v} \in \mathcal{R}^d \setminus \{0\}$.

Since standard PCA cannot capture nonlinear structures of the input data, we use a more advanced technique, the *Kernel Principal Component Analysis (KPCA)* [Scholkopf et al. 1998]. KPCA is a feature selection method in a high dimensional feature space, with dimension $M \leq \infty$. It is based on the computation of the standard linear PCA in a new feature space, into which input data is mapped using a nonlinear transformation. It allows us to obtain features with higher-order correlations between input data and to extract nonlinear components up to the number of observations N, where $N \leq M$.

To avoid computationally expensive calculations of high-dimensional dot products, *kernels* are used [Scholkopf and Smola 2002]. Kernels are simple functions defined on pairs of input patterns. Specifically, a kernel is a function K, such that for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $K(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$, where Φ is a mapping from $\mathcal{X}$ to a high-dimensional feature space $\mathcal{H}$. Some standard choices of the kernels are:

$$Gaussian : K(\mathbf{x}, \mathbf{y}) = e^{-|\mathbf{x}-\mathbf{y}|^2/2\sigma^2}, \qquad (1)$$

$$Polynomial : K(\mathbf{x}, \mathbf{y}) = (t + \lambda\mathbf{x} \cdot \mathbf{y})^p. \qquad (2)$$

In our experiments we use Gaussian kernel with $\sigma = 1$.

The KPCA is performed off-line on the whole database, in order to considerably decrease the dimensionality of the problem, thus reducing the computation time. Experimentally, decreasing the dimensionality from 219 to 100 has no impact on retrieval results.

In addition to reducing the computational complexity, our experiments show that KPCA improves the performance of the retrieval algorithms by 5%, both in the initial search and in the following RF iterations. This improvement stems from the fact that the KPCA finds correlations between the original features and increases the weight of the more important features.

Relevance Feedback Step: Relevance feedback is performed by a combination of a query refinement technique and a supervised feature extraction technique. Query refinement refers to a family of techniques that attempt to improve the estimate of the ideal query point by moving it towards relevant examples and away from irrelevant ones [Rocchio 1971; Ide 1971]. Supervised feature extraction methods search among all possible transformations for the best one, which preserves class separability as much as possible in the space with the lowest possible dimensionality [Fukunaga 1990].

The simplest form of supervised feature extraction is as follows. We are given N observations on d variables, divided into two subsets $\mathcal{D}_1$ and $\mathcal{D}_2$ with N_1 and N_2 samples in each subset, respectively. We aim at finding a projection $\mathbf{w}$, $y = \mathbf{w}^\top \mathbf{x}$, where $\{y_i\}_{i=1}^N$ are divided into the subsets $\mathcal{Y}_1$ and $\mathcal{Y}_2$, so as to achieve the maximal separation between $\mathcal{Y}_1$ and $\mathcal{Y}_2$. Let $\mathbf{m}_i = \frac{1}{N_i}\sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}$ be the sample means and $\tilde{m}_i = \mathbf{w}^\top \mathbf{m}_i$ be the projected means. We wish to measure the separation of the means. In other words, we wish to maximize the cost function

$$J(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}, \qquad (3)$$

where $\tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{m}_i)^2$. The cost function J is the *Fisher's Linear Discriminant Criterion* [Duda et al. 2000].

The above principles can be extended to a more general theory of *Linear Discriminant Analysis (LDA)* [Fukunaga 1990; Duda et al. 2000]. Instead of projecting the original d-dimensional data onto a single direction, the data is projected onto some r-dimensional subspace, $r \leq d$, by $\mathbf{y} = W^\top \mathbf{x}$, where $\mathbf{y} \in \mathcal{R}^r$ is a transformed data point and W is a $d \times r$ transformation matrix. The goal is similar – finding a matrix W that preserves the class separability as much as possible.

The optimization problem can be formulated as follows. Define two scatter matrices: the between-class scatter matrix S_B and the within-class scatter matrix S_W:

$$S_B = \sum_{i=1}^{2} N_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^\top, \tag{4}$$

$$S_W = \sum_{i=1}^{2} \sum_{\mathbf{x} \in \mathcal{D}_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^\top, \tag{5}$$

where $\mathbf{m}$ is the mean vector of all observations. The optimal transformation matrix W is defined as

$$W_{opt} = \underset{W}{\operatorname{argmax}} \left\{ \frac{W^\top S_B W}{W^\top S_W W} \right\}. \tag{6}$$

In our system, the user provides both relevant and irrelevant examples. The LDA finds an optimal linear transformation that re-weights the signature entries so that the maximal separation is achieved. However, the LDA also aims at clustering the irrelevant examples in the discriminating subspace, which is not only unnecessary but can also be potentially damaging.

The irrelevant examples are often too sparse to represent their true distribution. Moreover, they can be heterogeneous and reside far from each other in feature space. The set of relevant examples, however, is more likely to represent the true distribution since in reality the class of interest has a compact support. It is thus preferred to treat relevant and irrelevant examples differently. *Biased Discriminant Analysis (BDA)* addresses this asymmetry between the relevant and irrelevant examples [Zhou and Hunag 2001].

The difference between BDA and LDA lies in the definition of the scatter matrices. S_B and S_W are replaced by S_z and S_x as follows.

$$S_z = \sum_{i=1}^{N_z} (\mathbf{z}_i - \mathbf{m}_x)(\mathbf{z}_i - \mathbf{m}_x)^\top, \tag{7}$$

$$S_x = \sum_{i=1}^{N_x} (\mathbf{x}_i - \mathbf{m}_x)(\mathbf{x}_i - \mathbf{m}_x)^\top, \tag{8}$$

where $\{\mathbf{x}_i\}_{i=1}^{N_x}$ denote the relevant examples, $\{\mathbf{z}_i\}_{i=1}^{N_z}$ are irrelevant examples and m_x is the mean vector of relevant examples. Note, that this mean vector, $\mathbf{m}_x$, is subtracted from the observations. This is done in order to cluster the relevant examples together, while keeping them away from the irrelevant examples.

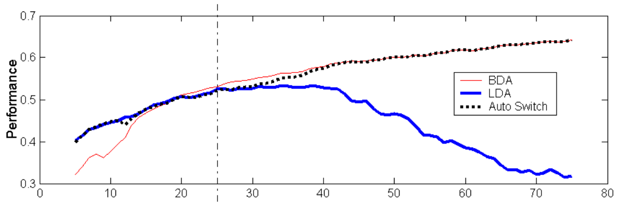
The optimization problem can then be rewritten as:

$$W_{opt} = \underset{W}{\operatorname{argmax}} \left\{ \frac{W^\top S_z W}{W^\top S_x W} \right\}. \tag{9}$$

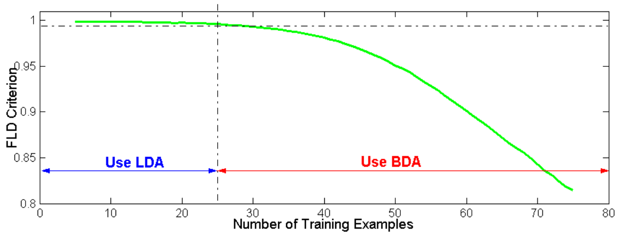
Despite its advantages, our experiments show that the BDA method fails for a small number of training examples. Fortunately, in this case, the LDA achieves relatively good results. On the other hand, the latter fails for a large number of training examples. This is the case because when the number of irrelevant examples is too high, it becomes difficult to cluster them together. Figure 3(a) illustrated the performance of the two methods, which is evaluated using the *Discounted Cumulated Gain (DCG)* criterion, described in the next section.

In a relevance feedback retrieval systems, the number of training examples provided by the user should not be controlled. Naturally, we would like our system to yield the best performance for any sample size. Obviously, a valid solution is to use LDA for a small set of training examples and BDA otherwise. But how should the system automatically determine which is the case?

We propose to use Fisher's Linear Discriminant Criterion (FLD) to decide which method to use. In our scheme, the FLD criterion is computed for each query. The higher its value, the higher the probability that LDA successfully discriminates between the relevant and the irrelevant classes. Figure 3 shows how the FLD criterion is used to switch between the LDA and the BDA methods. The "Auto Switch" line shows the average performance of the algorithm that automatically switches between LDA and BDA.



(a) The performance of LDA, BDA and automatic switch



(b) The FLD criterion

Figure 3: Using FLD to switch between LDA and BDA

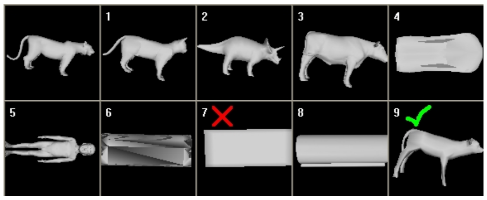
4 Experimental Results

Our experiments were performed on a database containing 1850 3D models, where 725 were classified into 25 classes with an average of 20 objects per class and 1125 free objects. Each model was represented by a 219-feature vector of its *sphere projection* signature [Leifman et al. 2003].

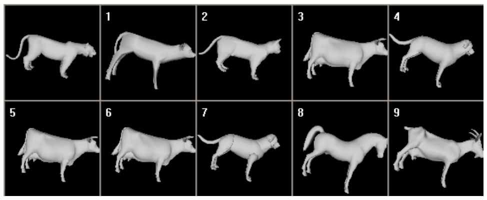
To evaluate the average performance, relevant and irrelevant training examples were chosen automatically, as follows. For each query, after the initial search, the top results that belong to the query object class were marked as relevant training examples and the rest were marked as irrelevant. The performance is evaluated using the *Discounted Cumulated Gain (DCG)* [Jarvelin and Kekalainen 2000; Leifman et al. 2003]. Roughly speaking, this criterion takes into account not only the relevance of the retrieved objects but also their positions among the the relevant results. We chose to use this criterion not only because of this property but also because it has the lowest standard deviation among all the standard measures [Leifman et al. 2003].

Figures 4– 7 demonstrates different uses of relevance feedback. Figure 4 shows how providing only a few judgments (one relevant result and one irrelevant result), the results of a relatively poor initial search are drastically improved. After a single RF iteration, all the top nine results are relevant. Figure 5 is an example of using

RF to narrow down the retrieval results. Using an open-roof car as a query object, both regular and race cars are retrieved. By marking the race cars as irrelevant and some regular cars as relevant, the next iteration retrieves only regular cars. Figure 6 is an example of using RF when the query object has only a remote similarity to the objects searched for. Using a helicopter as a query, airplanes can be retrieved after a single RF iteration. Figure 7 shows how two RF iterations are used to filter out geometrically-similar, but semantically-dissimilar, objects, i.e., only guitars are retrieved in the nine top-ranked results.

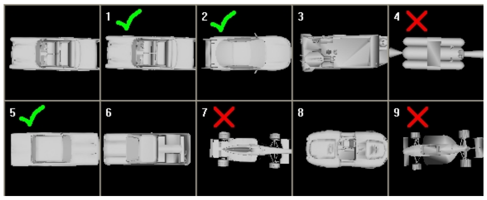


(a) Initial search

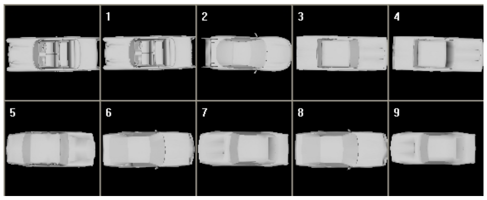


(b) One RF iteration

Figure 4: Retrieving four-legged animals – Improving the results



(a) Initial search

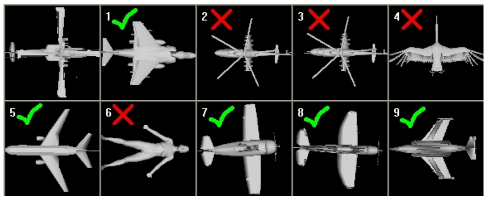


(b) One RF iteration

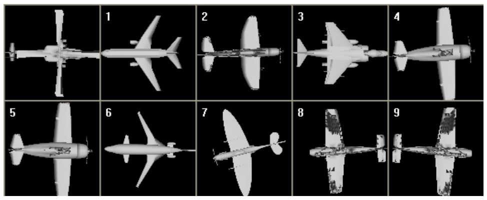
Figure 5: Retrieving cars – Narrowing down the results

Figure 8 shows the performance as a function of the number of training examples. The most drastic improvement is achieved after the first and second RF iterations, while the third and fourth iterations improve the results only slightly. Increasing the number of training examples improves the performance, as expected. Overall, relative to the initial search, the performance quality is almost doubled.

Figure 9 compares the performance of our algorithm to the performance of the SVM-based approach [Elad et al. 2001] and the feature space warping approach [Bang and Chen 2002], as a function of number of training examples. It can be seen that our algorithm outperforms both algorithms.



(a) Initial search



(b) One RF iteration

Figure 6: Retrieving airplanes using a helicopter as a query – query having a remote similarity to desirable objects

5 Conclusions

Relevance feedback provides the user with the added ability of influencing the search as it is being conducted. In this paper we proposed a novel relevance feedback scheme for retrieving 3D models. Our algorithm uses unsupervised feature extraction in a preprocessing stage and an optimal combination of supervised feature extraction techniques, LDA and BDA, and refinement during the online retrieval stage.

We have shown that our technique works well not only for improving the retrieval results, but also for narrowing them down. This is the case even when the query object is only remotely geometrically similar to the objects sought, and even when very few objects are used for training.

The algorithm achieves most of the improvement in the first couple of iterations, which is an important aspect in interactive techniques. Finally, this algorithm is shown to outperform previously proposed techniques.

References

Aksoy, S., Haralick, R., Cheikh, F., and Gabbouj, M. 2000. A weighted distance approach to relevance feedback. In *International Conference on Pattern Recognition*, 4812–4815.

Bang, H., and Chen, T. 2002. Feature space warping: An approach to relevance feedback. In *ICIP*.

Chen, Y., Zhou, X. S., and Huang, T. S. 2001. One-class svm for learning in image retrieval. In *ICIP*.

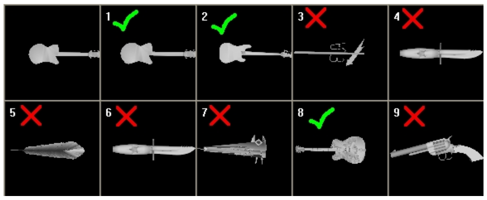
Duda, R., Hart, P., and Stork, D. 2000. *Pattern Classification*. John Wiley & Sons, New York.

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

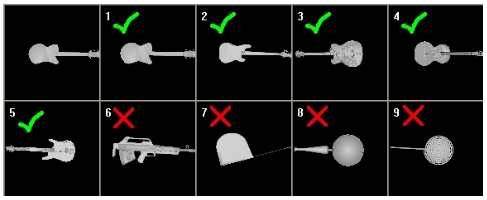
Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition.

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

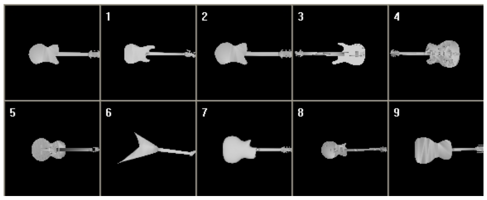
Ide, E. 1971. New experiments in relevance feedback. *The Smart System - Experiments in Automatic Document Processing*, 337–354.



(1) Initial search



(2) One RF iteration



(3) Two RF iterations

Figure 7: Filtering out geometrically but not semantically similar objects (query object at the top-left)

Ishikawa, Y., Subramanya, R., and Faloutsos, C. 1998. MindReader: Querying databases through multiple examples. In *Proceedings of 24th Int. Conference Very Large Data Bases*, 218–227.

Jarvelin, K., and Kekalainen, J. 2000. Ir evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Jolliffe, I. 1986. *Principal Component Analysis*. Springer-Verlag.

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

Leifman, G., Katz, S., Tal, A., and Meir, R. 2003. Signatures of 3d models for retrieval. In *Proceedings of the 4th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, 159–163.

Meilhac, C., and Nastar, C. 1999. One-class svm for learning in image retrieval. In *IEEE Intl Conf. on Multimedia Comp. and Sys.*

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

Paquet, E., Murching, A., Naveen, T., Tabatabai, A., and Rioux, M. 2000. Description of shape information for 2-d and 3-d objects. *Signal Processing: Image Communication*, 103–122.

Peng, J., Bhanu, B., and Qing, S. 1999. Probabilistic feature relevance learning for content-based image retrieval. *Computer Vision and Image Understanding 75*, 150–164.

Rocchio, J. 1971. Relevance feedback in information retrieval. *Prentice Hall*, 313–323. In The SMART Retrieval System.

Rui, Y., Huang, T., and Mehrotra, S. 1998. Relevance feedback techniques in interactive content-based image retrieval. In *Storage and Retrieval for Image and Video Databases (SPIE)*, 25–36.

Salton, G., and McGill, M. 1983. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

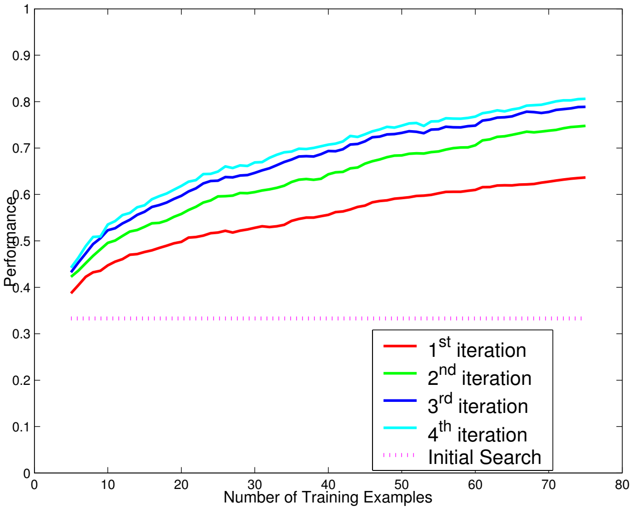


Figure 8: Performance vs. # of training examples

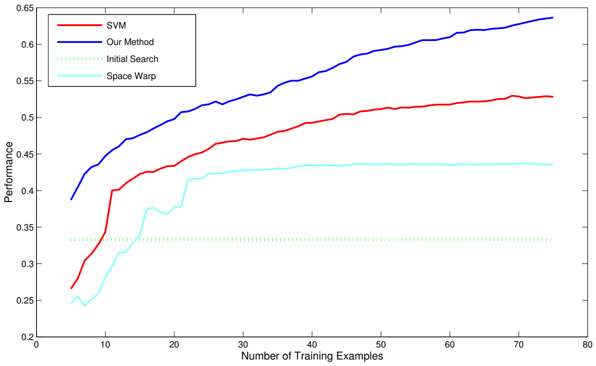


Figure 9: The performance of our algorithm, [Elad et al.] and [Bang and Chen]

Santini, S., and Jain, R. 2000. Integrated browsing and querying for image database. *IEEE Multimedia 7*, 26–39.

Scholkopf, B., and Smola, A. 2002. *Learning with Kernels*. MIT Press, Cambridge, MA.

Scholkopf, B., Smola, A., and Muller, K. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation 10*, 1299–1319.

Tieu, K., and Viola, P. 2000. Boosting image retrieval. In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, 228–235.

Tong, S., and Chang, E. 2001. Support vector machine active learning for image retrieval. In *Proceedings of ACM International Conference on Multimedia*, 107–118.

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

Zhou, X. S., and Hunag, T. S. 2000. A generalized relevance feedback scheme for image retrieval. In *Internet Multimedia Management Systems*.

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