EVALUATION OF KERNEL BASED METHODS FOR RELEVANCE FEEDBACK IN 3D SHAPE RETRIEVAL

Marcin Novotni, Gil-Joo Park, Raoul Wessel, Reinhard Klein

University of Bonn
Institute of Computer Science
{marcin,parkg,wesselr,rk}@cs.uni-bonn.de

ABSTRACT

Relevance feedback in information retrieval is an iterative search technique to bridge the *semantic gap* between the high level user intention and low level data representation. The main motivation for this work is that there is very little research done on applicability of relevance feedback methods to 3D shape retrieval. We experimentally evaluate 4 state-of-the-art techniques stemming from machine learning adopted to relevance feedback: Support Vector Machines (SVM), One Class SVM, Biased Discriminant Analysis (BDA) and Kernel BDA (KBDA). In contrast to previous results for content based image retrieval, SVM was found to perform the best, closely followed by KBDA and BDA. The major advantage of these methods lies in their efficiency and simplicity – (K)BDA is very simple to implement, and there are several publicly available implementations of SVMs. To warrant reproducibility, the experiments were conducted on a pre-classified 3D object data set, the Princeton Shape Benchmark, which is publicly available online, as well. 3D Zernike descriptors are used as object descriptors which were computed using the public domain reference implementation.

1. INTRODUCTION

The recent proliferation of 3D objects over various digital archives was triggered by new advances in geometrical modeling, 3D acquisition and graphics hardware. However, the generation of 3D content still remains a time- and cost intensive process. Therefore, the need for re-use of already available models and hence their automatic retrieval has arisen. Since the first such system has been presented, several methods for 3D object description and retrieval emerged in various domains ranging from CAD through chemistry and biology to digitalized cultural heritage artifacts. The domain of general objects is the target of our investigations in this paper. Such models are used e.g. in virtual reality environments or computer games and thus are of special interest for the computer graphics community. A nice overview of the recent developments in this domain can be found in [15].

According to recent evaluations, e.g. [14], the retrieval performance of 3D shape descriptors is very limited in spite of meanwhile several years of research in this field. In contrast to other fields of information retrieval, defining similarity between 3D objects has turned out to be quite difficult. The problems occuring are similar to those in image retrieval, which include poor generalization results. While it seems to be very difficult to develop new descriptors yielding substantial improvement in performance, it is worthwhile to look at what can be done to get better retrieval given state-of-the-art descriptors. The descriptors utilized in retrieval capture various properties of the objects, i.e. spatial frequencies, etc. It can be well observed that different classes of objects are best characterized by different properties. In terms of frequencies for instance, some object classes may be well characterized by their low frequencies, but not by their high frequencies. In other words, the within class variance of low frequency values is low, whereas the variance of high frequencies is large. For other classes the opposite may be the case. As a consequence, if we invariantly tune our distance measure to yield optimal retrieval results for one class, it may be poor for other classes. This is one of the manifestations of the *semantic gap*, i.e. the gap between the abstract, high level user intention and low level data representation and processing. To cope with this problem relevance feedback was introduced. Relevance feedback is an iterative search scheme where the user is presented with results of a search and is allowed to select relevant and irrelevant objects. Thus, the search process may be tuned to recognize relevant objects and discriminate these from irrelevant ones.

In this paper we evaluate the relevance feedback methods in context of *category search* for 3D shapes which is the most common. In this setting the goal of the user is

Thanks to the German Research Foundation (Deutsche Forschungsgemeinschaft DFG) for funding this work within the scope of the strategic research initiative V3D2 ("Distributed Processing and Delivery of Digital Documents").

to find objects similar to a given example. The question to investigate is, which of the available methods are best suited for relevance feedback in context of a 3D shape retrieval system. Our criteria for the choice of techniques to be evaluated are: (i) good previous performance for other data, e.g. images, and (ii) the ease of application, i.e. the methods should be easy to implement and integrate into the retrieval system. We present evaluation results on four state-of-the-art relevance feedback methods that have been investigated by Zhou and Huang [18] in context of content based image retrieval: utilizing Support Vector Machines (SVM), Biased Discriminant Analysis (BDA), Kernel BDA (KBDA). Due to its close relation to SVM, we include One Class SVM based relevance feedback [3] in our evaluation. To warrant reproducibility, the experiments were conducted on a pre-classified 3D object data set, the Princeton Shape Benchmark [14]. 3D Zernike descriptors are used as object descriptors which were computed using the public domain reference implementation. In spite of the existence of several 3D shape retrieval systems, apart from [5, 8] we are not aware of utilization of relevance feedback schemes in such systems, not to mention an evaluation study. It is well known that the retrieval accuracy of 3D shapes is still extremely low due to the above mentioned reasons. Our results show that the retrieval performance of 3D shape search engines may be significantly improved by the deployment of relevance feedback. Moreover, this improvement is achieved by using very simple off-the-shelf algorithms evaluated in this paper that may easy be plugged into most search engines.

The outline of the paper is as follows. In Section 2 we give an overview over related methods in 3D shape retrieval and relevance feedback. The 3D Zernike descriptors and the methods evaluated in the paper are briefly described in Section 4 and 3, respectively. Our experimental setup is presented in Section 5 which is followed by the evaluation in Section 6. We conclude in Section 7.

2. RELATED WORK

2.1. 3D shape descriptors

Although several sophisticated powerful representations of 3D objects have been elaborated, most of the descriptions represent 3D geometries as points in a vector space $\mathbb{R}^N$ or $\mathbb{C}^N$. The main advantage of this representation lies in its simplicity facilitating the utilization of efficient metrics and indexing schemes. Furthermore, numerous pattern recognition techniques may be applied to it which have been proven to be very important in information retrieval. Some of the techniques rely on shape histograms [1] and shape distributions [10]. These methods are outperformed by some recent ones based on spherical harmonics (SH). In [17] the 3D geometry is sampled by rays shot from the centroid of the object, generating spherical functions for a number of radial intervals which are then transformed into SH representation. [6] uses essentially the same idea but the spherical functions are computed by sampling the previously voxelized object by concentric shells. 3D Zernike descriptors [9] extend this latter method by using a radial basis component capturing coherence in this direction, too. Although there are some slight differences in performance, according to the experiments in [14] and [9], these SH based descriptors belong to the best performing vectorial 3D shape characterizations.

To obtain a more general overview of the 3D shape descriptors, we refer the reader to an excellent recent survey of 3D shape retrieval [15].

2.2. Relevance feedback

The corner stone of a relevance feedback module is the learning algorithm. There is a vast body of research on machine learning, however, there are some constraints that caused some specific methods to prevail: the tuning must succeed using a relatively small training set – the user usually selects only a small number of examples. Furthermore, the target class distribution often has a complex shape in the descriptor space. Very importantly, the learning phase and ranking of documents must allow for interactive retrieval times.

Early approaches include query point movement [12] attempting to find an optimal query point by moving it towards positive and away from negative examples. This scheme was extended by re-weighting of the components of descriptor vectors to reduce the importance of dimensions preventing the retrieval of relevant objects, see e.g. [7].

Since recently, algorithms from machine learning like SVM and other kernel methods have been successfully utilized in relevance feedback, see e.g. [11, 16]. These have numerous advantageous properties with respect to the relevance feedback: they are among others efficient to compute, can correctly classify object classes of complex shape in the descriptor space, etc. In content based retrieval there is usually a number of classes and the user wants to retrieve a particular one. Therefore, the classical SVM often used in context of relevance feedback, may potentially be suboptimal, as it attempts to explicitly model all the irrelevant objects as a single class. Based on this observation, the usage of One-Class SVM [3] and Biased Discriminant Analysis and its kernel version [18] was investigated.

In this paper we experimentally verify the results on the latter methods for 3D data. In context of 3D shape retrieval a method similar to SVM was applied [5] to descriptors based on geometrical moments. This method was shown to be outperformed by the BDA in [8].

3. 3D ZERNIKE DESCRIPTORS

The theory of 3D Zernike moments has been introduced by Canterakis [2], practical aspects along with an evaluation for 3D shape retrieval followed in [9]. In the remainder of this section we provide some basic information on the descriptors, we refer to the original papers for more details.

Like other moments, 3D Zernike moments are computed as projections of the (square integrable) object function $f \in L^2$ onto a set of functions $\Psi = \{\psi_i\},\ i \in \mathbb{N}$ over the domain Ω:

$$\mu_i = \langle f, \psi_i \rangle = \int_{\Omega} f(\mathbf{x}) \cdot \overline{\psi_i(\mathbf{x})}\, d\mathbf{x}. \tag{1}$$

The function set Ψ should be chosen in such a way that the resulting values characterize the objects well, are compact and easily comparable, etc. Furthermore, the descriptors should be invariant under similarity transformations.

Construction: For the 3D Zernike moments, $\Psi = \{Z_{nl}^m\}$. If we define the conversion between Cartesian and spherical coordinates by $\mathbf{x} = |\mathbf{x}|\xi = r\xi = r(\sin\vartheta\sin\varphi, \sin\vartheta\cos\varphi, \cos\varphi)^T$, the functions can be written

$$Z_{nl}^m(\mathbf{x}) = R_{nl}(r) \cdot Y_l^m(\vartheta, \phi).$$

with l restricted so that $l \leq n$ and $(n-l)$ be an even number. Here, $R_{nl}(r)$ are polynomials in radius and Y_l^m denote the spherical harmonics which form a Fourier basis on the sphere:

$$Y_l^m(\vartheta, \varphi) = N_l^m P_l^m(\cos\vartheta) e^{im\varphi},$$

where N_l^m is a normalization factor

$$N_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-m)!}{(l+m)!}},$$

and P_l^m denotes the associated Legendre functions. The radial polynomial part of Eqn. 1 is defined so that the function set forms an orthonormal basis within the unit ball. To summarize, 3D Zernike moments Ω_{nl}^m are computed as

$$\Omega_{nl}^m := \frac{3}{4\pi} \int_{|\mathbf{x}| \leq 1} f(\mathbf{x}) \overline{Z_{nl}^m(\mathbf{x})} d\mathbf{x}.$$

Interpretation: In the above form, 3D Zernike moments are a tensor product of an angular function – the spherical harmonics (SH) – and a radial polynomial. It is well known that the SH can be interpreted as a Fourier basis on the sphere, i.e. this part of the basis functions measures the angular frequencies of the object. The radial polynomial can be interpreted in a similar manner, namely that with different orders of the polynomial, its dot products with the object measure different frequencies of the object in radial direction. Intuitively, we slice the object with infinitely many spheres with radii $r \in (0, 1]$, compute the SH coefficients for each of the spheres and combine the coefficients for different radii using the radial polynomial.

Invariance: The invariance under translation is achieved by shifting the center of gravity of the object into the origin, the scale invariance is achieved e.g. by scaling the object, so that the spread of its points is $1/2$.

Achieving rotation invariance is more of a challenge. Most of the works on 3D shape retrieval achieve rotation invariance by aligning the principal axes of the object with the coordinate system axes. For the 3D Zernike descriptors we use the invariance property of the SH basis to obtain the corresponding invariance. Let $\Omega_{nl} = (\Omega_{nl}^l, \Omega_{nl}^{l-1}, \Omega_{nl}^{l-2}, \ldots, \Omega_{nl}^{-l})^t$ denote $2l+1$ dimensional vectors of according Zernike moments. It can be proven that by defining 3D Zernike descriptors F_{nl} as norms of vectors Ω_{nl}, i.e. $F_{nl} := ||\Omega_{nl}||$, we get rotationally invariant features of the objects, meaning that the F_{nl} computed for a given object and its arbitrarily rotated version are equal. In terms of the frequency interpretation mentioned above – loosely speaking – this implies discarding the phase while retaining the magnitudes of spherical frequencies for a given order of the spherical harmonics.

4. METHODS

The problem we are trying to solve in our setting is basically an instance of supervised learning. Let $X \subseteq \mathbb{R}^n$ denote the input domain, i.e. the domain of the vectorial shape descriptors, and $Y \subseteq \mathbb{R}$ the output domain. Now, given l training examples, i.e. the relevant and irrelevant objects marked by the user $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)) \subseteq (X \times Y)^l$, we try to tune a decision function $f : X \to Y$. In the training phase $y_i = +1$ and $y_i = -1$ indicate relevant and irrelevant objects, respectively. While in our case the real value of f indicates the relevance of a particular input (the larger the value, the more relevant the object), in the usual binary classification setting the sign of the f value is taken as an indicator showing to which class the input belongs to.

In what follows, we provide a brief description of the learning algorithms used. Due to space constraints we strive only to give an intuition of how these methods work, in-depth discussions may be found elsewhere, e.g. [4] for SVMs, and [18] for BDA.

4.1. Kernel trick

Before we actually describe the SVM and KBDA, we briefly introduce an important building block of both of these methods, the *kernel trick*. The main motivation here is that simple, e.g. linear classifiers will not succeed due to complex topology and/or shape of the input classes in the descriptor space. Hence, the idea is that by mapping the input data into a higher dimensional *feature space* using an appropriate non-linear mapping $\phi : x \rightarrow \phi(x)$, the same simple classifiers may be applied in the feature space. The dimensionality of the feature space may be very high, though, or even infinite. Therefore, it would be advantageous not to be forced to explicitly compute the mapping. This is facilitated by the kernel trick, which may be applied to decision functions that can be formulated in terms of inner products of the input data. According to this technique, the computation of inner products in the feature space is carried out implicitly by evaluating a kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. Although there are numerous kernels that may be applied (in fact all those satisfying the Mercer conditions [4]), we use the Gaussian radial basis kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ in our experiments.

Beside the algorithms presented in the remainder of this section, the kernel trick has proven to successfully "non-linearize" classical data analysis methods that can be formulated in terms of inner products of the input like PCA, ICA, discriminant analysis, etc.

4.2. Support Vector Machines and One-Class SVM

The basic idea of SVMs is to divide the input space $X \subseteq \mathbb{R}^n$ into two halfspaces by a hyperplane so that the generated halfspaces contain the classes to be distinguished. The hyperplane is defined using a real valued linear decision function $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$, $f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. The sign of f determines to which class a given input belongs to.

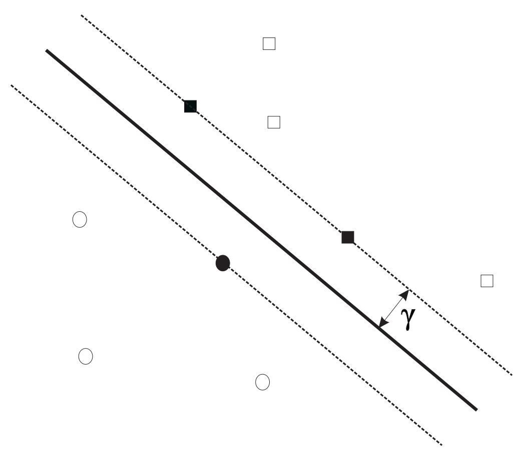

Fig. 1. Squares and circles indicate the training data of two classes, filled shapes indicate the support vectors. γ denotes the margin and the thick line is the optimal hyperplane.

It is plain that there are many possible hyperplanes separating the training data, see Fig. 1. The idea is to find a hyperplane so that the distance (*margin*) of the training data closest to it (*support vectors*) is as large as possible. In spite of its simplicity, it can be proven that such a classifier is optimal with respect to generalization, i.e. the ability of the classifier to correctly classify data not in the training set [4]. The result is the simplest model for an SVM, the Maximal Margin Classifier. This classifier works only for linearly separable data, i.e. where a separating hyperplane exists between the two classes, hence, it is not applicable to many real-world problems. This problem is addressed by the Soft Margin SVMs, where – roughly said – a parameter C controls the tradeoff between the right classification of most of the examples and the size of the margin.

SVMs work best if two classes have to be separated, however, in case of relevance feedback, we want to distinguish one class against all others in the database. This is a big difference, as the classes are usually supposed to exhibit some clustering behavior in the input domain, so that two classes indeed can be separated. In case of one against many classification the separability by a hyperplane may pose a problem. Thus the idea of Chen et al. [3] to use One-Class SVM to estimate the support of the positive examples. Instead of computing an optimally separating hyperplane, this method aims at generating a hypersphere that contains most of the positive examples while being as small as possible. As the name suggests, only the positive examples are used to determine the parameters of this algorithm, moreover similar formulations and optimization techniques are used as in case of SVMs, see [13] for details. The decision function f for One-Class SVM measures in essence the distance of the data points from the center of the computed hypersphere, again, we refer to [13] for more details.

In our evaluation we used the "kernelized" versions of Soft Margin SVM and One-Class SVM, as the decision- and objective functions may be expressed in terms of inner products in both cases. The parameters, i.e. γ for the radial basis function and C for the soft margin are determined by bootstrap estimation, which is described in Section 5.

4.3. (Kernel) Biased Discriminant Analysis

Similar to the SVM, the Biased Discriminant Analysis [18] has a very intuitive interpretation. Here, a transformation of the input space is seeked that brings the positive examples closer together and the negative ones are repelled from the positives. Mathematically, this is formulated in terms of the Rayleigh quotient well known from other Discriminant Analysis solutions:

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^t \mathbf{S}_+ \mathbf{W}|}{|\mathbf{W}^t \mathbf{S}_- \mathbf{W}|} \qquad (2)$$

where $\mathbf{S}_+$ and $\mathbf{S}_-$ are the scatter matrices of the relevant and irrelevant examples

$$\mathbf{S}_* = \sum_{\mathbf{x} \in X_*} (\mathbf{x} - \mathbf{m}_+)(\mathbf{x} - \mathbf{m}_+)^t.$$

Here the subscript $* \in \{+,-\}$ indicates the relevant and irrelevant examples, respectively; $\mathbf{m}_+$ denotes the centroid of the relevant examples. The interpretation is as follows: while seeking a transformation $\mathbf{W}$ maximizing the argument in Eqn. 2, we maximize the ratio of the scatter of the irrelevant examples with respect to the centroid of relevant ones to the scatter of relevant examples to their centroid. Thus, we get the effect described above. According to this formulation the measure of scattering is the determinant of the scatter matrix, which can be written as the product of the eigenvalues in the principal directions. Hence, we measure the ratios of the squared hyperellypsoidal scattering volumes of the relevant and irrelevant point distributions.

The solution matrix $\mathbf{W}$ consists of the generalized eigenvectors $\mathbf{V}$ computed from the generalized eigenvalue problem $\mathbf{S}_-\mathbf{V} = \lambda \mathbf{S}_+\mathbf{V}$. The *discriminating transform* $\mathbf{A}$ is constructed by scaling the eigenvectors by the square root of the corresponding eigenvalues $\mathbf{A} = \Lambda^{1/2}\mathbf{W}$, where Λ is a diagonal matrix of eigenvalues. This way, the effect of "repelling" and "bringing together" described above is achieved. The rank of a data point $\mathbf{x}$ is given by its distance from the centroid $\mathbf{m}_+$ in the transformed coordinates $f(\mathbf{x}) = \|\mathbf{A}(\mathbf{x} - \mathbf{m}_+)\|$.

With small numbers of training examples, the result will be biased towards these, moreover the scatter matrices are likely to be singular. Therefore, some kind of regularization is needed. The simple idea used by [18] is to add some small quantities to the diagonals of the scatter matrices, thereby extending the corresponding hyperellypsoids by small values in directions where no information is available:

$$\mathbf{S}_*^r = (1-\mu_*)\mathbf{S}_* + \frac{\mu_*}{n} tr(\mathbf{S}_*)\mathbf{I}$$

where $\mu_* \in [0,1]$ denotes the regularization factor for the relevant and irrelevant distributions; $\mathbf{I}$ denotes the identity matrix, n is the dimensionality of the input domain and $tr(\cdot)$ the trace operation.

Last but not least, Eqn. 2 can be rewritten using inner products of input data. Hence a kernel version of the algorithm (KBDA) can be formulated, see [18] for more details.

In our evaluation we investigated both the linear and kernel version of the BDA. The optimal parameters, i.e. the regularization factors μ_* and γ controlling the radial basis Gaussian kernel were determined using bootstrap estimation, see Section 5 for details.

5. EXPERIMENTAL SETUP

In this section we give details on the setup of the experiments including the description of the utilized database, ranking the objects and the measures we used to evaluate the relevance feedback methods.

5.1. Dataset

We use the Princeton Shape Benchmark (PSB) [14] in our experiments. Although this database has some drawbacks, among the 3D test datasets used in various retrieval system evaluations, it contains the largest variety of objects, it is fully pre-classified and, very importantly, it is available online. Unfortunately, the PSB is rather unbalanced, meaning that there are some large classes containing up to 50 objects, while others contain only 4 or 5 objects. Hence, the tuning of algorithms and the test results are somewhat biased.

The PSB contains a total of 1814 3D objects of general categories, like cars, human, airplanes, swords, chairs, etc. The dataset is divided into a training and test part, each consisting of 907 objects classified into 90 and 92 classes, respectively. We conducted the bootstrap estimation on the training dataset to obtain the optimal parameters for the learning methods, the evaluation is conducted on the test set using the predefined classification.

5.2. Ranking of objects

After each round of interaction the learning algorithm gets a number of pairs $(\mathbf{x}_i, y_i)$ where $y_i = +1$ or $y_i = -1$ indicates that $\mathbf{x}_i$ was selected as relevant or irrelevant, respectively. Based on these training examples the learning algorithm is tuned (the respective objective function is optimized). Finally, the decision function f is used to rank all the objects in the database, so that the rank *Rank* of the object indicating its position in the relevance ordering obeys: $Rank(\mathbf{x}_i) \leq Rank(\mathbf{x}_j) \Leftrightarrow f(\mathbf{x}_i) \geq f(\mathbf{x}_j)$.

5.3. Evaluation measures

Precision and recall To evaluate the search results we mainly used the precision-recall diagrams commonly used in information retrieval. Let C be the class to be retrieved and $|C|$ the count of elements in the class. For the k top ranked objects from the test set among which $n \leq k$ are relevant the following measures are defined: precision $P_k(C) = \frac{n}{k}$ and recall $R_k(C) = \frac{n}{|C|}$. The interpretation is as follows: precision measures the ratio of relevant objects among the first k matches and recall quantifies the ratio of relevant objects to the total number of objects in the class.

Average precision As we intend to measure and tune the algorithms with respect to several parameters, a single scalar value would be of more use. Let the class C consist of the objects $o_1, \dots, o_n$ and $Rank(o_i)$ denote the rank of o_i in the ordered list of results according to a query. Then $P_{avg}(C) = \frac{1}{|C|}\sum_{i=1}^{|C|} \frac{i}{Rank(o_i)}$ denotes the average precision with respect to the class C. From average precisions w.r.t. all the classes $C_1, \dots, C_l$ we can compute the overall average precision $P_{avg} = \frac{1}{l}\sum_{i=1}^{l} P_{avg}(C_i)$.

Residual measures While the above values are widely used in the search-by-example setting, some further considerations are needed to measure the performance of the relevance feedback correctly. The objects marked relevant or irrelevant are likely to be ranked correctly as the learning machine was given their relevance explicitly. Hence, these objects should not be accounted for in the evaluation measure. To achieve this, we explicitly exclude the marked objects from the precision and recall computations and thus get values we call *residual* precision and recall.

Given a test set, like the PSB, where there are large differences in class sizes, an additional possibility is to weight each class with the ratio of not yet selected objects to $|C|$ while computing the average precision $P_{avg}^{w} = \frac{\sum_{i=1}^{l} w_i \cdot P_{avg}(C_i)}{\sum_{i=1}^{l} w_i}$ with weights $w_i = 1 - \frac{n_i}{|C_i|}$. The idea here is that larger classes should contribute more to the overall result, as the estimation of evaluation measures is more robust for these classes. For instance, if there is a total of 6 objects in the class with 4 objects marked relevant, there are only 2 objects left to estimate the evaluation measure. However, it should be noted that this way the evaluation will be biased towards the larger classes which limits the generality of results.

6. EVALUATION

6.1. Optimal parameters

As already described in Section 4, each learning algorithm has a set of specific parameters that have to be tuned in order to get optimal results. To summarize, these parameters are:

- SVM: kernel width γ, soft margin error penalty C
- BDA: regularization parameters μ_+ and μ_-
- KBDA: kernel width γ, regularization parameters μ_+ and μ_-
- One-Class SVM: kernel width γ, hypersphere radius ν

We use a bootstrap estimation procedure to determine the optimal parameters. The training data consists of a set of positive and negative examples of a given class chosen randomly from the pre-classified training part of the PSB database. Given these and a set of values for the specific parameters, the evaluation measures are computed. This procedure is conducted several times for each single class and the resulting measure values are summarized for the whole training database. A grid search is applied to find the optimal parameter values using a coarse base grid over the maximal parameter intervals, which is then refined in the vicinity of optimal parameter values. We used 4 positive and 4 negative examples as training data, the evaluation measure was the residual average weighted precision. The optimal parameter values found are summarized in Table 1.

SVM	$C = 2$		$\gamma = 660$
One-Class SVM	$\nu = 0.6$		$\gamma = 660$
BDA	$\mu_+ = 0.5$	$\mu_- = 0.9$	
KBDA	$\mu_+ = 0.1$	$\mu_- = 0.9$	$\gamma = 600$

Tab. 1. Optimal parameter values for the learning algorithms.

It should be noted that other evaluation measures and different sizes of training example sets could have been used. We tested the values of specific parameters corresponding to optimal performances with varying training sample size (between 2 to 10) and different evaluation measures (average precision with and without weighting), and found that the variance of parameter values was not significant.

6.2. Results

The evaluation is conducted as follows: first a random pair of 1-1 positive and negative examples is chosen for a class for which the evaluation measures are computed and averaged for the whole test database as described in Section 5. The training set is iteratively extended by a procedure mimicking the user behavior: the highest rank negative and the lowest rank positive example is chosen on the first page of results (i.e. first 24 highest rank objects in our system). If no appropriate object is found on the first page, the highest rank object is chosen. In our evaluation we tested the performance for up to 10 iterations. We conducted the above procedure several times to provide for robustness of evaluation.

An impression of the impact of relevance feedback can be received by looking at Figure 2 where the diagram of weighted precision vs. recall averaged over all classes is shown after the 2nd and 5th iteration. In Figure 3 we show the development of the weighted average precision corresponding to different iteration numbers. As it can be seen, the retrieval performance improves significantly. For instance, having retrieved 50% of the objects in a class, on average we have about 55% precision after the second iteration and about 90% after fifth iteration which is indeed a substantial difference.

Although some differences can already be seen in Figures 2 and 3, further measurements are needed in order to assess the differences in performances of the algorithms. As already discussed in Section 5, the evaluation should not be conducted on marked objects, it is of much more interest, how many objects of the *residual* set are found by the algorithm. In Figure 4 the residual weighted average precision values corresponding to marked object counts can be seen.

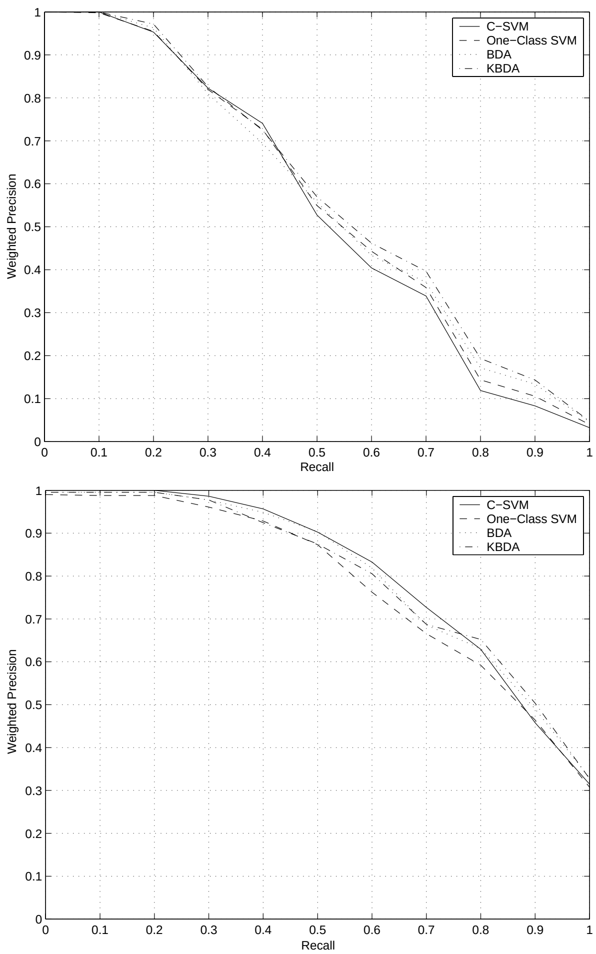

Fig. 2. Weighted precision vs. recall values after the 2nd and 5th iteration of the learning algorithms.

It is interesting to see in Figure 4 that the SVM performs better than KBDA with more than 5 training pairs, as this is inconsistent with findings of [18] (although in a bit different setting) where KBDA was found to clearly outperform SVMs in a face image retrieval experiment. This may be due to the specific topology and/or shape of classes in the descriptor space favoring the SVM induced decision function. It is further interesting that the performance of (linear) BDA is almost as good as that of kernelized algorithms, which implies the simplicity of the shape of class distributions in the descriptor space. The inferiority of One-Class SVM for relevance feedback in our system means that the additional information from negative examples is indeed useful.

It is plain that whatever learning method is used, the performance of relevance feedback is closely related to the performance of the underlying descriptors, the 3D Zernike descriptors in our case. As the latter is comparable with other

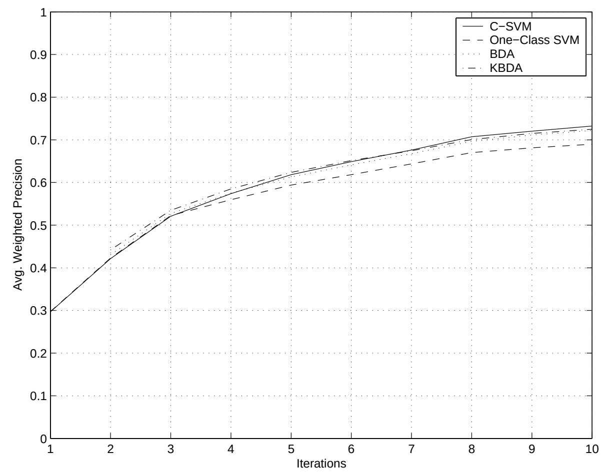

Fig. 3. Average weighted precision with increasing iteration number.

state-of-the-art vectorial descriptors, we believe that the results presented above should generalize to these as well.

7. CONCLUSIONS

Given that current 3D shape retrieval systems perform rather poorly, we believe that the results presented in this paper demonstrate the substantial benefit of relevance feedback. To facilitate the choice of methods we presented an evaluation of 4 relevance feedback algorithms with respect to 3D shape retrieval based on 3D Zernike descriptors. We provided a brief description of utilized algorithms and descriptors along with evaluation measures extended to suit them to quantify relevance feedback performance. The investigated algorithms are very easy to implement, or are readily available, and are very simple to integrate into a search engine using vectorial descriptors.

Our experiments show that the 3D shape retrieval quality is substantially improved by utilizing relevance feedback. In contrast to previous results, SVM was found to perform the best and slightly better then KBDA, the weakest performance was yielded by One-Class SVM.

8. REFERENCES

[1] M. Ankerst, G. Kastenmuller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *Symposium on Large Spatial Databases*, pages 207–226, 1999.

[2] N. Canterakis. 3D Zernike moments and zernike affine invariants for 3D image analysis and recognition. In *11th Scandinavian Conf. on Image Analysis*, 1999.

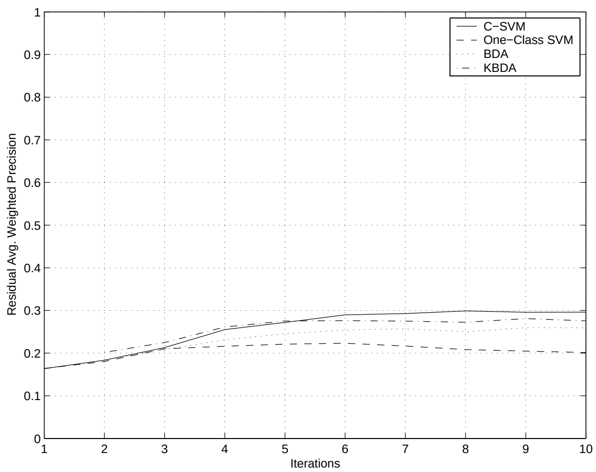

Fig. 4. Residual average weighted precision with increasing iteration number.

[3] Yunqiang Chen, Xiang Zhou, and Thomas S. Huang. One-class svm for learning in image retrieval. In *IEEE Int'l Conf. on Image Processing*, 2001.

[4] N. Cristianini and J. Shawne-Taylor. *An introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, 2000.

[5] M. Elad, A. Tal, , and S. Ar. Content based retrieval of VRML objects - an iterative and interactive approach. In *Eurographics Multimedia Workshop*, pages 97–108, 2001.

[6] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3D models. *ACM Transactions on Graphics*, 22(1), 2003.

[7] Jing Huang, S. Ravi Kumar, and Mandar Mitra. Combining supervised learning with color correlograms for content-based image retrieval. In *MULTIMEDIA '97: Proceedings of the fifth ACM international conference on Multimedia*, pages 325–334. ACM Press, 1997.

[8] George Leifman, Ron Meir, and Ayellet Tal. Relevance feedback for 3d shape retrieval. In *The 5th Israel-Korea Bi-National Conference on Geometric Modeling and Computer Graphics*, pages 15–19, 2004.

[9] Marcin Novotni and Reinhard Klein. Shape retrieval using 3D Zernike descriptors. *Computer Aided Design*, 36(11):1047 – 1062, 2004.

[10] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *International Conference on Shape Modeling and Applications*, 2001.

[11] P.Hong, Q.Tian, and T.S.Huang. Incorporate support vector machines to content-based image retrieval with relevance feedback. In *IEEE International Conference on Image Processing (ICIP 2000)*, 2000.

[12] J. J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, 1971.

[13] Berhard Scholkopf, John C. Platt, John Shawe-Taylor, and Alex J Smola. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443 – 1471, 2001.

[14] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Shape Modeling International*, June 2004.

[15] Johan W.H. Tangelder and Remco C. Veltkamp. A survey of content based 3D shape retrieval methods. In *International Conference on Shape Modeling and Applications 2004 (SMI'04)*, pages 145–156, 2004.

[16] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118. ACM Press, 2001.

[17] D. V. Vranic. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *IEEE International Conference on Image Processing (ICIP 2003)*, pages 757–760, 2003.

[18] Xiang Sean Zhou and Thomas S. Huang. Comparing discriminating transformations and svm for learning during multimedia retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 137–146. ACM Press, 2001.