Int J Comput Vis
DOI 10.1007/s11263-009-0294-1

Similarity Learning for 3D Object Retrieval Using Relevance Feedback and Risk Minimization

Ceyhun Burak Akgül · Bülent Sankur · Yücel Yemez · Francis Schmitt

Received: 26 September 2008 / Accepted: 14 September 2009

Abstract We introduce a similarity learning scheme to improve the 3D object retrieval performance in a relevance feedback setting. The proposed algorithm relies on a score fusion approach that linearly combines elementary similarity scores originating from different shape descriptors into a final similarity function. Each elementary score is modeled in terms of the posterior probability of a database item being relevant to the user-provided query. The posterior parameters are learned via off-line discriminative training, while the optimal combination of weights to generate the final similarity function is obtained by on-line empirical ranking risk minimization. This joint use of on-line and off-line learning methods in relevance feedback not only improves the retrieval performance significantly as compared to the totally unsupervised case, but also outperforms the standard support vector machines based approach. Experiments on several 3D databases, including the Princeton Shape Benchmark, show also that the proposed algorithm has a better small sample behavior.

Keywords Relevance feedback · Similarity models and learning · Empirical ranking risk · Support vector machines · 3D object retrieval

1 Introduction

There exist two major research problems concerning the design of content-based multimedia retrieval systems. In the first problem, one is concerned with finding robust representation schemes describing the content of multimedia objects in terms of compact surrogates. In the context of 3D objects, content description is synonymous to 3D shape description. Several effective and efficient description algorithms have been proposed in the last decade (Bustos et al. 2005; Tangelder and Veltkamp 2008) and promising performance results have been obtained on standard benchmarks (Akgül 2007; Akgül et al. 2009; Vranic 2004). In the second problem, one seeks computational similarity measures between descriptors that well approximate the semantic similarity between objects, based on the grounds of user requirements and perceptual judgments. This second issue constitutes the main focus of the present paper. Specifically, we propose novel similarity learning algorithms for 3D object retrieval (3DOR) and test them against existing ones.

The common denominator of the 3DOR algorithms discussed in this paper is their reliance on the relevance feedback mechanism (Datta et al. 2008; Smeulders et al. 2000; Zhou and Huang 2003). In many multimedia retrieval instances, relevance feedback has proven to be effective in decreasing the semantic gap, that is, the discrepancy between

Part of this work has been done during C.B. Akgül's Ph.D. studies at Boğaziçi University and Télécom ParisTech.

C.B. Akgül (✉)
Video Processing and Analysis Group, Philips Research, Eindhoven, Netherlands
e-mail: cb.akgul@gmail.com

B. Sankur
Electrical-Electronics Engineering Department, Boğaziçi University, Istanbul, Turkey

Y. Yemez
Computer Engineering Department, Koç University, Istanbul, Turkey

F. Schmitt
Department of Signal-Images, Télécom ParisTech, Paris, France

Published online: 25 September 2009

Table 1 A condensed taxonomy of relevance feedback algorithms

Category	References
Query modification, heuristic feature re-weighting	Huang et al. (1997), Nastar et al. (1998), Peng et al. (1999), Picard et al. (1999), Porkaew et al. (1999), Rui et al. (1998), Santini and Jain (2000)
Subspace-based feature re-weighting	Ishikawa et al. (1998), Rui and Huang (2000), Schettini et al. (1999), Zhou and Huang (2001)
Density estimation and clustering based	Chen et al. (2001), Laaksonen et al. (1999), Wu et al. (2000)
Probabilistic (Bayesian)	Giacinto and Roli (2004), Vasconcelos and Lippman (1999, 2000)
Discriminative learning based	Chen et al. (2001), Wu et al. (2000), Guo et al. (2002), Tao et al. (2006), Tieu and Viola (2000), Tong and Chang (2001), Zhang et al. (2001)

the computational description of the content and its semantic class (Smeulders et al. 2000). Relevance feedback (RF) is an interactive scheme that makes the user an integral part of the retrieval process. Many different implementations have been proposed since its first appearance in the text retrieval domain (Rocchio 1966). Relevance feedback algorithms require the user to label a few presented database items as *relevant* or *irrelevant* to the query. The positively (relevant) and negatively (irrelevant) marked items together reflect the user's preferences and serve as high-level information that will be used by a learning algorithm to refine the search results. Datta et al. (2008) and Zhou and Huang (2003) provide relatively comprehensive reviews of relevance feedback algorithms in image retrieval. A condensed taxonomy derived from (Datta et al. 2008) and (Zhou and Huang 2003) is given in Table 1 along with sample references. Note that this taxonomy is by no means exhaustive and there are no clear cut boundaries between the branches.

Discriminative learning based relevance feedback methods (Chen et al. 2001; Wu et al. 2000; Guo et al. 2002; Tao et al. 2006; Tieu and Viola 2000; Tong and Chang 2001; Zhang et al. 2001) are of particular importance to our work. These methods have gained prominence in recent years, mainly because of powerful statistical classification algorithms, such as support vector machines (SVM), decision trees and boosting methods (see Hastie et al. 2001 for the technical details of these methods). In this paradigm, the system first learns a classifier between positive and negative items provided as a feedback by the user. The classifier can then rank all the database items with respect to their relevance to the query.

In the present work, we investigate two different approaches that fall in this discriminative learning based category. The first one is the popular SVM-RF approach that has already been successfully employed for general image retrieval (Chen et al. 2001; Guo et al. 2002; Tao et al. 2006; Tong and Chang 2001; Zhang et al. 2001). In the 3DOR context, we are aware of only two articles that employed SVM-RF for comparative performance analysis (Leifman et al. 2005; Novotni et al. 2005). In its basic form, SVM-RF minimizes the classification error using the labeled items to learn a (possibly non-linear) decision boundary between the positive and negative classes. Once the decision boundary is learned, the distance of the remaining database items to the boundary can serve as a similarity measure for the next retrieval round. The underlying assumption here is that the farther an item is from the boundary, the more confident we should be about its predicted label (positive or negative). In Sect. 2, we provide more details on the variants of this scheme that appeared in the literature.

The main contribution of this work is a score fusion (SF) approach to relevance feedback, which we abbreviate as SF-RF. We compare the performance of our proposed scheme against SVM-RF, which is well-established and documented in the literature. Although the two approaches are philosophically similar to each other (they are both based on discriminative learning), SF-RF differs from SVM-RF in several aspects. SF-RF scheme starts with the same kind of feedback inputs from the user as SVM-RF, but then it tries to directly find a similarity function based on the minimization of the empirical ranking risk (Clémençon et al. 2008), which is defined simply as the number of incorrectly ranked database items with respect to their similarity to the query. We formulate the final similarity function as a linear combination of elementary similarity scores. In our work, an elementary similarity score corresponds to the posterior probability of a database item being relevant to the user-provided query, given the distance between their respective descriptors. This score fusion scheme (Akgül et al. 2008) has several favorable properties:

- The proposed score fusion algorithm (SF-RF) minimizes the ranking risk (cf. Sect. 3), which we consider as a more suitable optimization criterion for the retrieval task than the classification error, defined as the total number of the relevant database items that are classified as irrelevant and of the irrelevant database items that are classified as relevant. Note that we derive the relevance relations between query and database items from the available ground truth class information, as explained in Sect. 6.2. To the best of our knowledge, this is the first work in the visual data retrieval domain using ranking risk minimization.

- We employ an explicit similarity model whose parameters are rigorously estimated by optimization (cf. Sects. 3 and 4). In this sense, the algorithm does not need a grid search to tune its parameters, such as the kernel width in non-linear SVM learning. Furthermore, the model is linear in elementary similarities, thus scalable to large databases.
- We convert distances between descriptors into posterior probabilities, which allow us to incorporate prior knowledge about individual discriminative powers of different descriptors and/or components of a descriptor (cf. Sects. 4 and 5.3). Furthermore, the resulting $[0,1]$-probability scale can be conveniently interpreted as a similarity scale, where the unit value corresponds to the highest degree of similarity.
- It is also possible to extend the algorithm to no-feedback situations with off-line learning (cf. Sect. 6.6).
- The algorithm is generic in the sense that it can be used with any type of shape descriptor once a matching score that measures the similarity between shapes is available. Consequently, a vector-based descriptor representation is not necessary and the algorithm can work for example with graph-based descriptors without modification, as long as matching scores between graphs are provided.

We report the retrieval performance of these two schemes (SVM-RF and SF-RF) on four different 3D object databases (cf. Sect. 6.1), including the Princeton Shape Benchmark (PSB) (Shilane et al. 2004). On PSB, Novotni et al. (2005) have shown that SVM-RF with 3D Zernike descriptors is valuable for 3DOR and outperforms other relevance feedback approaches. In the present work, we show that even better results can be obtained with score fusion-based relevance feedback via ranking risk minimization. While our baseline shape description is the density-based framework (Akgül 2007; Akgül et al. 2009) (cf. Sect. 5), any other scheme with state-of-the-art performance could also be used.

The paper is structured as follows. In the next section, we describe the SVM-RF approach, discuss its limitations and point to its variants proposed in the relevance feedback literature. Sections 3 and 4 embody our major contributions in this work. In Sect. 3, we present our ranking risk minimization-based score fusion approach in detail. In Sect. 4, we show the derivation of the relevance posteriors using pairs of descriptors. In Sect. 5, after providing the main lines of our chosen 3D shape description methodology, the density-based framework (Akgül 2007; Akgül et al. 2009), we explain its use in the context of relevance feedback-based retrieval. In Sect. 6, we evaluate comparatively the two relevance feedback algorithms, the previously proposed SVM-RF and our contribution SF-RF, on PSB and other popular 3D object databases. In Sect. 7, we discuss our findings and draw conclusions.

2 SVM Based Relevance Feedback

One of the earliest uses of SVM in relevance feedback is described in (Zhang et al. 2001) and its variants can be found in (Chen et al. 2001; Guo et al. 2002; Tao et al. 2006; Tong and Chang 2001). Let Q and X stand for the query and database items respectively and let the indicator variable $y \in \{-1,+1\}$ encode the relevance relation between Q and X. Learning is carried out using a training set $\{(\mathbf{x}^{(m)}, y^{(m)})\}_{m=1}^{M}$, provided by the user to start the relevance feedback process, where $\mathbf{x}^{(m)} \in \mathbb{R}^p$ denotes the descriptor vector of the mth labeled database item $X^{(m)}$, and $y^{(m)}$ its relevance label. The SVM algorithm aims at finding a decision function $S(\mathbf{x})$ for the test vector $\mathbf{x}$ in the form below, in order to maximally separate the positive and negative classes:

$$S(\mathbf{x}) = \sum_{m'=1}^{M'} \alpha_{m'} y^{(m')} \mathcal{K}(\mathbf{x}, \mathbf{x}^{(m')}) + b \tag{1}$$

where the index $m' = 1, \dots, M' \leq M$ runs through the so-called support vectors, that is, the training vectors $\mathbf{x}^{(m')}$ that lie within a prescribed margin from the decision boundary. The scalar variables $\{\alpha_{m'}\}$ are the non-zero Lagrange multipliers arising as part of the SVM optimization and b is the intercept of the decision function $S(\mathbf{x})$. The symmetric form $\mathcal{K}(\mathbf{x}, \mathbf{x}')$ is a typically nonlinear kernel function, enabling the evaluation of dot products in a higher dimensional space than the original input space where the vectors $\mathbf{x}$ live. Good references on kernel functions to enhance class separation in SVM are (Hastie et al. 2001; Schölkopf and Smola 2002).

In the context of relevance feedback, one can view $S(\mathbf{x})$ as a similarity function. Standard SVM classification is performed based on the sign of $S(\mathbf{x})$, that is, the test vector $\mathbf{x}$ is assigned to the positive class if $S(\mathbf{x}) > 0$; otherwise to the negative class. In the RF context however, we are not just interested in classifying the database items but also in ordering them as a function of their relevance to the query. Arguably, the function $S(\mathbf{x})$ itself can serve for scoring relevance like a similarity function, that is, a high positive $S(\mathbf{x})$ indicates that the database item X is very similar to the query Q, while a negative $S(\mathbf{x})$ with high absolute value shows that X is very distant from Q. Using the SVM output as a similarity measure seems useful, yet lacks any theoretical justification. There are instances where the distance to the SVM boundary might fail to capture the true similarity between semantically relevant patterns (see Fig. 1 in Guo et al. 2002). The work in (Guo et al. 2002) tries to remedy this problem by a constrained similarity measure, where the distance-to-boundary is used only for negatively predicted items, while the similarity between positively predicted items is quantified by the Euclidean distance to the query. Nevertheless, we should note that there are several applications where the distance-to-boundary has been shown to provide good empirical retrieval performance (Tao et al. 2006; Tong and Chang 2001; Zhang et al. 2001).

Another limitation of the basic SVM-RF approach is due to differences between positively and negatively labeled items. In general, positive items are compactly clustered in the feature space, but this is usually not the case for negative ones, which, by nature, can be anything irrelevant to the query. In other words, it is much harder to learn the negative class. The naive solution would be using a much larger set of negative items for SVM learning, but then the decision boundary would be shifted towards the "center" of the negative class, in which case, as pointed out in (Tao et al. 2006), many irrelevant (negative) test items would lie on the wrong side of the boundary, hence they would be misclassified as relevant. The approach proposed in (Chen et al. 2001) deals with this issue by a one-class-SVM learned using only positive items at the expense of valuable information contained in the negative ones. The work in (Tao et al. 2006) follows an asymmetric bagging strategy to remedy the imbalance between positive and negative sets. The final decision function is obtained by averaging several SVM classifiers, each trained with the same positive set $\mathcal{X}^+$ but with a different set $\mathcal{X}^-$ of randomly sampled negative items, such that $|\mathcal{X}^+| = |\mathcal{X}^-|$ in each lot. Random sampling is an essential component of bagging, which is a variance reduction technique (see Hastie et al. 2001 for further details). In addition to enhancing the stability of the final classifier, it also constitutes a computationally more convenient alternative to clustering.

All these enhancements certainly improve the basic SVM-RF but they remain as variations on a theme because:

- No similarity model is explicitly available for optimization.
- They all minimize the classification error in the hope that this will also work well for retrieval where we are rather interested in ranking the database items as a function of their relevance.

In the following two sections, we present an approach that directly tackles both of these fundamental issues.

3 Score Fusion Based Relevance Feedback

Consider the problem of ranking two generic database items X and X' based on their relevance to a query Q. Suppose also that we have access to K different elementary similarity functions $s_k(X, Q)$, each reflecting a distinct geometrical and/or topological commonality between the database items and the query. In our context, elementary similarity functions arise from shape descriptors of different nature and/or from different component sets of the same descriptor. These are discussed in Sect. 4.

In 3D retrieval problems, when two database items X and X' are compared in terms of their similarity to a query Q, the more similar item should be ranked higher than the less similar one. Otherwise, the pair (X, X') is said to be an incorrectly ranked pair of database items. Obviously, an ideal similarity measure should score higher for similar pairs (X, Q) as compared to less similar ones. Putting together the elementary scores in a vector form as $\mathbf{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$, we can define our objective as building a scalar similarity function $S(X, Q) = \langle \mathbf{w}, \mathbf{s} \rangle$, where $\mathbf{w} = [w_1, \ldots, w_K] \in \mathbb{R}^K$ is the weight vector. We expect $S(X, Q)$ to assign higher scores to more relevant items, i.e., it should satisfy the following property:

$$\begin{aligned} &S(X, Q) > S(X', Q) \\ &\quad \text{if } X \text{ is more relevant to } Q \text{ than } X', \\ &S(X, Q) < S(X', Q) \quad \text{otherwise,} \end{aligned} \tag{2}$$

where ties are arbitrarily broken. As usual, we encode the relevance of X and X' to Q by indicator variables y and y' respectively. Recall that $y = +1$ means that X is relevant to Q, while $y = -1$ means that it is not relevant. Thus, the above property reads as:

$$\begin{aligned} &S(X, Q) > S(X', Q) \quad \text{if } y - y' > 0, \\ &S(X, Q) < S(X', Q) \quad \text{if } y - y' < 0. \end{aligned} \tag{3}$$

The function $S(X, Q)$ must subsume the similarity information residing in the individual scores s_k in order to emulate the ideal similarity notion between objects, hence to achieve a better retrieval performance. Given the linear form $S(X, Q) = \langle \mathbf{w}, \mathbf{s} \rangle$, we formulate the score fusion problem as finding a weight vector $\mathbf{w}$, which is optimal according to the empirical ranking risk (*ERR*) criterion. *ERR* is defined as the number of incorrectly ranked pairs of database items with respect to a query Q. Given a set of items $\{X^{(m)}\}_{m=1}^{M}$, we can write this criterion formally as:

$$\begin{aligned} ERR(S; Q) = \frac{2}{M(M-1)} \sum_{m<n} \mathbb{I}\{(S(X^{(m)}, Q) - S(X^{(n)}, Q)) \\ \cdot (y^{(m)} - y^{(n)}) < 0\} \end{aligned} \tag{4}$$

where $\mathbb{I}\{\cdot\}$ is the 0–1 loss, which is one if the predicate inside the braces is true and zero otherwise. *ERR* simply counts the number of wrongly ordered database item pairs. If

Table 2 Algorithm 1: learning the ranking weights (on-line)

Given a query Q, a set of labeled database items $\{X^{(m)}, y^{(m)}\}_{m=1}^{M}$ provided by the user and K different basic similarity functions $s_k(X, Q)$.

(1) Calculate a score vector $\mathbf{s}^{(m)} \in \mathbb{R}^K$ for each $(X^{(m)}, Q)$-pair.
(2) Identify the pairs of labels $(y^{(m)}, y^{(n)})$ such that $y^{(m)} - y^{(n)} \neq 0$.
(3) Construct the score difference vectors $\mathbf{v}^{(m,n)}$ and their rank indicators $z^{(m,n)}$.
(4) Run the SVM algorithm to learn the weight vector $\mathbf{w} \in \mathbb{R}^K$, using the derived training set $\{(\mathbf{v}^{(m,n)}, z^{(m,n)})\}_{m<n} \subset \mathbb{R}^K \times \{-1, +1\}$.

$S(X^{(m)}, Q) < S(X^{(n)}, Q)$ but $y^{(m)} > y^{(n)}$, the scoring function $S(., Q)$ has assigned (wrongly) a higher score to $X^{(n)}$ than to $X^{(m)}$, while $X^{(m)}$ is relevant to Q but $X^{(n)}$ is not. Thus the scoring function has made an error in ranking $X^{(m)}$ and $X^{(n)}$ with respect to the query and *ERR* should be incremented by one. Such errors are naturally undesirable and our task is to find a scoring function (or more appropriately its parameters $\mathbf{w}$) so that the number of incorrectly ranked pairs is as small as possible.

The trick to minimize *ERR* is to identify (4) as the empirical classification risk in a different domain. We should first introduce another indicator variable $z \in \{-1, 0, +1\}$ such that $z = (y - y')/2$. This leads to the following observation:

$$z = \begin{cases} +1 & x \text{ should be ranked higher than } x', \\ -1 & x \text{ should be ranked lower than } x'. \end{cases}$$

Note that when $z = 0$, i.e., if database items X and X' have the same relevance label, we can decide arbitrarily. Corresponding to each non-zero z, we define a *score difference vector* $\mathbf{v} \triangleq \mathbf{s} - \mathbf{s}'$, i.e., the difference between the score vectors $\mathbf{s}$ and $\mathbf{s}'$ of database items X and X' respectively. With this new notation and writing the scoring function $S(., Q)$ explicitly in terms of its parameters $\mathbf{w}$, (2) now reads as

$$ERR(\mathbf{w}; Q) = \frac{2}{M(M-1)} \sum_{m<n} \mathbb{I}\{z^{(m,n)} \langle \mathbf{w}, \mathbf{v}^{(m,n)} \rangle < 0\} \quad (5)$$

where the index pairs (m, n) correspond to database item pairs $(X^{(m)}, X^{(n)})$ with different relevance labels, that is, $z^{(m,n)}$ is either $+1$ or -1. Thus, we have converted *ERR* written in terms of *score* vectors $\mathbf{s}$ and *relevance* indicators y (see (4)) into an empirical classification error written in terms of *score difference* vectors $\mathbf{v}$ and *rank* indicators z (see (5)). The problem of finding the parameter vector $\mathbf{w}$ of the scoring function $S(., Q)$ is now identical to binary classification of the score difference vectors. We can employ the SVM algorithm in a straightforward way, however with the interpretation that *the weight vector learned by SVM in the score difference domain can directly be used to evaluate the scoring function at the next retrieval round*. The training algorithm to learn the parameter $\mathbf{w}$ is summarized in Table 2.

The computational complexity of this ranking algorithm is quadratic in the number of marked items M, in contrast to the standard SVM learning, which has linear complexity (excluding the number of required operations in solving the associated optimization problem). While this might be a disadvantage for large M in general, we did not run into any practical difficulty in the particular relevance feedback context because M should be kept small anyway for user convenience. We illustrate the complexity of Algorithm 1 with a typical relevance feedback case where $M = 16$ and the number of positive and negative instances are equal ($M^+ = M^- = 8$). Note that in this example, SVM-RF should learn the decision function with a training set of size $M = 16$. For ranking on the other hand, the total number of the pairs of relevance indicators $(y^{(m)}, y^{(n)})$ such that $y^{(m)} - y^{(n)} \neq 0$, hence the number of training score difference vectors, is 64. On-line learning with so few training vectors is computationally feasible with standard SVM packages such as LibSVM (Chang and Lin 2001). In fact, we see this quadratic increase in the size of the training set as an advantage of ranking over classification because the relevance information provided by the user is exploited more efficiently.

4 Relevance Posterior as an Elementary Similarity Function

In this section, we elicit our elementary similarity functions modeled in terms of posterior probabilities. Suppose that a query Q and a database item X are each described by K descriptors $\{Q_k\}_{k=1}^K$ and $\{X_k\}_{k=1}^K$ respectively. Suppose also that we are able to measure the dissimilarity between these descriptors via scalar-valued functions $d_k(X_k, Q_k) \in [0, \infty)$. We intentionally avoid the vector notation $\mathbf{q}_k \in \mathbb{R}^p$ and $\mathbf{x}_k \in \mathbb{R}^p$ for the descriptors in order to emphasize that this approach is generic. To clarify, if Q_k and X_k were graphs, $d_k(., .)$ would be a graph matching distance; if they were vectors, we could use any Minkowski metric of the form $d_k(\mathbf{x}_k, \mathbf{q}_k) = \|\mathbf{x}_k - \mathbf{q}_k\|$. They can even be scalars $q_k \in \mathbb{R}$ and $x_k \in \mathbb{R}$, e.g., the kth entry of a high-dimensional descriptor vector, in which case the absolute difference $d_k(x_k, q_k) = |x_k - q_k|$ would serve the purpose.

We can directly use the plain dissimilarity values $d_k(X_k, Q_k)$ to learn a weighted *dissimilarity function* $D(X, Q) = \sum_k w_k d_k(X_k, Q_k)$ via ranking risk minimization. The trick to do this is in fact trivial and we can obtain an algorithm as in Sect. 3 by just changing the polarities of the expressions in (2) and (3). However, we conjecture that using the posterior probability of positive relevance given

Table 3 Algorithm 2: learning the posterior model (off-line)

Given a training set of N descriptors $\{X_k^{(n)}\}_{n=1,k=1}^{N,K}$, the corresponding set of pairwise relevance labels $\mathcal{Y} = \{y^{(m,n)}\} \in \{-1,+1\}^{N \times N}$, dissimilarity functions $d_k(.,.)$, $k = 1, \dots, K$.

For all $k = 1, \dots, K$:

(1) Calculate the set $\mathcal{D}_k = \{d_k^{(m,n)} = d_k(X_k^{(m)}, X_k^{(n)})\} \in [0,\infty)^{N \times N}$

(*) Note that $y^{(m,n)} = \begin{cases} +1 & \text{if } X^{(m)} \text{ is relevant to } X^{(n)} \\ -1 & \text{otherwise} \end{cases}$

(**) To each $d_k^{(m,n)}$, there is an associated relevance label $y^{(m,n)} \in \{-1,+1\}$.

(2) Sample and estimate T times. For $t = 1, \dots, T$:

- Randomly choose two equally sized subsets $\mathcal{Y}^+$ and $\mathcal{Y}^-$ of $\mathcal{Y}$, consisting of only positive and only negative relevance labels respectively. Construct the corresponding dissimilarity sets $\mathcal{D}_k^+$ and $\mathcal{D}_k^-$.
- Input the positive and negative sets $\{(\mathcal{D}_k^+, \mathcal{Y}^+)\}$ and $\{(\mathcal{D}_k^-, \mathcal{Y}^-)\}$ to the algorithm in (Lin et al. 2007) to estimate $A_k^{(t)}$ and $B_k^{(t)}$.

(3) $A_k \leftarrow Average(A_k^{(t)})$ and $B_k \leftarrow Average(B_k^{(t)})$.

a query and a database item as a similarity measure is a better approach than using plain distances as it allows us to incorporate some prior knowledge on pairwise relevance relationships into the retrieval process. Formally, our elementary similarity functions $s_k(X_k, Q_k)$ have the following form:

$$s_k(X_k, Q_k) = \mathbb{P}(y = 1 | X_k, Q_k) \sim \mathbb{P}(y = 1 | d_k(X_k, Q_k)). \tag{6}$$

Equation (6) conveniently models the relevance information carried by the descriptor pair (X_k, Q_k) via a mapping from the scalar-valued dissimilarity $d_k(X_k, Q_k)$ to the [0, 1]-probability scale. Here $d_k(X_k, Q_k)$ can be viewed as a plausible approximation to the given information, since we are solely interested in the affinity between the descriptors X_k and Q_k. In order to concretize this idea, we have to determine the explicit form of the posterior probability. We find the following logistic model flexible:

$$\mathbb{P}(y = 1 | d_k(X_k, Q_k)) \propto \frac{1}{1 + \exp(A_k d_k(X_k, Q_k) + B_k)}, \tag{7}$$

where A_k and B_k are model parameters to be estimated from data. In choosing the posterior model in (7), we are inspired by Platt's work (Platt 1999) on mapping scalar SVM outputs into probabilities. The above logistic model is flexible because it does not make any distributional assumption about the dissimilarity values. The parameters A_k and B_k can be estimated by *off-line* discriminative training as given in Table 3. More details about the objective function and the algorithm to optimize the model in (7) can be found in (Platt 1999) and (Lin et al. 2007) respectively.

The randomization in step (2) of Algorithm 2 is reminiscent of the asymmetric bagging approach in (Tao et al. 2006) and aims at remedying the imbalance between the sizes of positive and negative sets. Though, we must emphasize that the learning of the posterior model is performed only one time on a representative training set of objects in an off-line manner. Note also that we repeat this procedure for each of the available K descriptors. Once the posterior model parameters $\{(A_k, B_k)\}_{k=1}^K$ are available, our full similarity model reads as

$$S(X, Q) = \sum_{k=1}^{K} \frac{w_k}{1 + \exp(A_k d_k(X_k, Q_k) + B_k)}, \tag{8}$$

where the weights $\{w_k\}_{k=1}^K$ are determined using the Algorithm 1 (see previous section) during the on-line relevance feedback stage.

5 3D Shape Description

5.1 The Density-Based Framework

For shape description, we employ the density-based framework (DBF), which has a retrieval performance comparable to other state-of-the-art methods (Akgül et al. 2009). In DBF, the descriptor of a 3D object is derived from the probability density function (pdf) of a multivariate local feature computed on the surface of the object. Specifically, the vector of pdf values obtained by kernel density estimation (KDE) becomes the shape descriptor. In (Akgül 2007), the discriminative power of several multivariate surface features within DBF has been investigated on different 3D object databases. Three of the DBF descriptors are particularly interesting:

- The *radial* descriptor is the pdf-vector of a surface point's normalized coordinates at various distances from the object's center of mass.

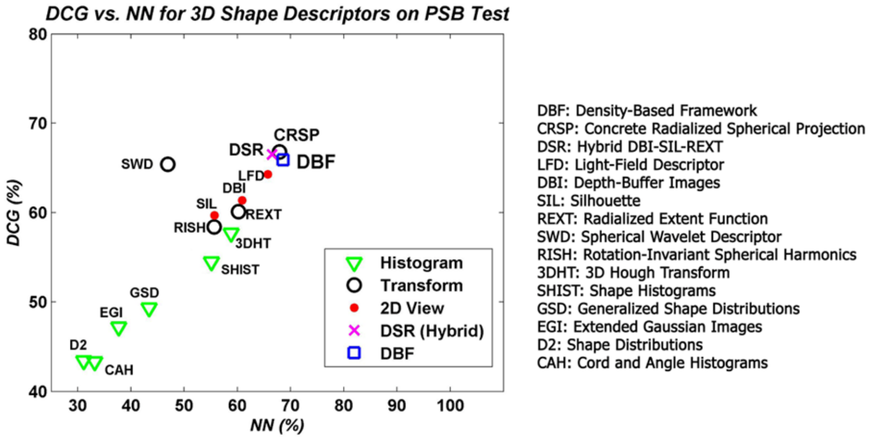


Fig. 1 DCG vs. NN performance plot of 3D shape descriptors on the PSB test set (Shilane et al. 2004) (see Sect. 6.2 for the definitions of DCG and NN performance measures; *markers* indicate methodological categories, see Akgül et al. 2009 for details and the references therein)

- The *t-plane* descriptor is the pdf-vector of the local surface normal at various distances from the object's center of mass.
- The *sec-order* descriptor is the pdf-vector of a multivariate local feature consisting of shape index (a function of principal curvatures) and radial-normal alignment at various distances from the object's center of mass.

As usual, a query object Q and a database object X can be compared by evaluating the distance between their vector-based descriptors $\mathbf{q}$ and $\mathbf{x}$. For instance, the distance values $d(\mathbf{x}_{radial}, \mathbf{q}_{radial}) = \|\mathbf{x}_{radial} - \mathbf{q}_{radial}\|$ can be used to sort the database items X based on their *radial* similarity to the query Q. In order to benefit from different types of shape information carried by density-based descriptors, we can simply sum their corresponding distance values. Note that this elementary score fusion is unsupervised and does not involve any statistical learning. The retrieval performance of DBF with the unsupervised fusion of the above descriptors on PSB test set (907 objects in 92 classes) is illustrated in Fig. 1 with other state-of-the-art descriptors (see Sect. 6.2 for the definitions of DCG and NN performance measures; markers indicate methodological categories, see Akgül et al. 2009 for details and the references therein). We see that DBF not only outperforms a great portion of descriptors but is also on a par with two well-known highly effective approaches: the hybrid DSR descriptor (Vranic 2004) and the concrete radialized spherical projection descriptor (CRSP) (Papadakis et al. 2007).

The good performance of DBF can be explained by the following facts: (i) all the available local surface information up to second order is exploited within the KDE setting; (ii) KDE copes with measurement uncertainties due to small pose normalization errors, small shape variations and/or mesh degeneracies; (iii) invariance against coordinate axis mislabelings and mirror reflections is achieved by taking the minimum distance between two descriptors over the whole set of coordinate axis relabelings and mirror reflections (Akgül et al. 2009).

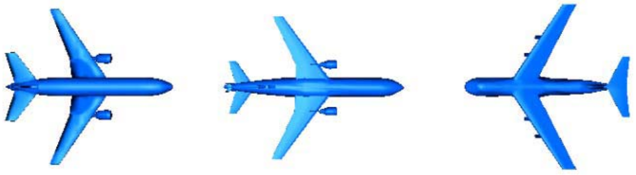


Fig. 2 Three airplane models after PCA-based pose normalization: major axes are correctly found but the front of the fuselage of the rightmost model is in the opposite direction

5.2 Descriptor Alignment

Since pose invariance contributes critically to the success of DBF, we briefly explain it here (see Akgül et al. 2009 for further details). Our *radial* and *t-plane* descriptors depend on the particular coordinate frame in which the 3D object is placed. Consequently, if two 3D objects to be compared are not properly aligned with respect to each other, a spuriously large distance between descriptors might occur. That is, transformations such as rotations, reflections, and labeling of the coordinate axes might eclipse semantic similarities. PCA-based methods (Vranic 2004) partially resolve this pose normalization issue by finding the directions of the three major object axes. Axis labels can be assigned according to the decreasing rank of the eigenvalues found by PCA, while polarities can be estimated by moment-based approaches as in (Vranic 2004). However, there still remain ambiguities about the axis labels and polarities, since this scheme does not always yield consistent results, as illustrated in Fig. 2. We find that minimizing the distance between two descriptors over all possible axis relabelings and reflections constitutes a better alternative, which is also computationally feasible within DBF. The *radial* and *t-plane* descriptors enjoy the convenient property that a given transformation Γ changing the axes of an object (by a relabeling and/or reflection) corresponds to a unique permutation π of the descriptor entries. In other words, if the descriptor of an object O is $\mathbf{x} = [x_k]$, then the descriptor of its transformed version $\Gamma(O)$ becomes $\pi(\mathbf{x}) = [x_{\pi(k)}]$. Thanks to this permutation property or pairings $\Gamma^{(k)} \leftrightarrow \pi^{(k)}$; $k = 1, \dots, 48$, all the 48-transformations can be implemented rapidly without the need to recompute the descriptors. In retrieval, we make use of this advantage of DBF either to derive an invariant distance measure between descriptors or to align each database object with respect to the query.

5.3 DBF in Relevance Feedback

In the SVM-RF context, we simply concatenate shape descriptors from different modalities, the *radial*, *t-plane* and *sec-order* descriptors into a single vector. On the other hand in SF-RF, we need elementary similarity scores to be linearly combined via ranking risk minimization; these elementary scores are in turn generated using segments of the descriptor vectors, which we call *chunks*. Let $\mathbf{x} \in \mathbb{R}^p$ be one of the *radial*, *t-plane* and *sec-order* descriptor vectors. We can write $\mathbf{x}$ in terms of the concatenation of K descriptor chunks of equal size $\mathbf{x}_k \in \mathbb{R}^{p'}$ such that $\mathbf{x} = [\mathbf{x}_1, \dots, \mathbf{x}_k, \dots, \mathbf{x}_K]$ and $p = Kp'$. We call this operation as *descriptor chunking*. Each descriptor chunk provides its dissimilarity value $d_k = \|\mathbf{x}_k - \mathbf{q}_k\|$, which can be mapped to a similarity score s_k via the posterior model $s_k = \mathbb{P}(y = 1|d_k)$ (cf. Sect. 4). For instance, if we have $p = 1024$ as the original descriptor dimension and $p' = 128$ as the chunk dimension, we obtain $K = 8$ scores that will be fed to the subsequent fusion stage where a distinct weight w_k for each of the individual scores is learned by ranking risk minimization. In the limit $p' = 1$, we obtain as many scores as the original dimension p of the descriptor, i.e., $K = p$.

6 Experiments

6.1 3D Object Databases

In this work, we have experimented with four different 3D object datasets. All of the objects are complete 3D models given by triangular meshes. These datasets differ substantially in their semantic content as discussed below.

The Princeton Shape Benchmark (PSB) is a publicly available database containing 1814 models, categorized into general classes such as animals, humans, plants, household objects, tools, vehicles, buildings, etc. (Shilane et al. 2004). Its classification is induced by functionality as well as by form. Accordingly, there are many instances where unsupervised shape description methods might fail to resolve semantic ambiguities due to high within-class variation. It can be conjectured that PSB is one of the 3D datasets where relevance feedback schemes can be particularly effective. The dataset was released by the Princeton group as two equally sized subsets, called *training set* and *test set*. In the 3D retrieval community, it is now a common practice to use the training set (90 classes) for tuning the parameters involved in the computation of a particular shape descriptor, and report the retrieval results on the test set (92 classes) using the tuned parameters. Only 21 classes are common to both sets, and the remaining classes occur in one or the other set, but not in both. The class sizes are somewhat imbalanced: there are small classes consisting of only five items and large ones containing up to 50 items. For the sake of clarity, we would like to point out that the naming convention of the sets in PSB is different from the one adopted in the context of statistical classifiers where the term "training set" is reserved for the set used for learning the classifier and the term "test set" is reserved for actually testing the classifier.

The Sculpteur Database (SCU) is a private database containing over 513 models corresponding to mostly archaeological objects residing in museums (Goodall et al. 2004). SCU consists of 53 categories of comparable set sizes, including utensils of ancient times such as amphorae, vases, bottles, etc.; pavements; and artistic objects such as human statues (part and whole), figurines, and moulds. SCU classes are in general more homogeneous compared to PSB but discriminating the vases from some of the amphorae might still be difficult unless high level information is incorporated in the search process.

The SHREC'07 Watertight Database (SHREC-W) was released for the Watertight track of the Shape Retrieval Contest (SHREC) in 2007 (Giorgi et al. 2007). It consists of 400 watertight meshes of high resolution, classified into 20 equally sized classes such as human, cup, glasses, octopus, ant, four-legged animal, etc. Classification in SHREC-W is largely induced by topological equivalences.

The Purdue Engineering Shape Benchmark (ESB) is another database that was used in the SHREC'07 event and consists of 865 models representing engineering parts (Jayanti et al. 2006). This dataset is organized based on a ground truth classification with two levels of hierarchy. Overall there are three super-classes, namely, flat-thin objects, rectangular-cubic prisms, and solids of revolution, which are further categorized into 45 classes (we consider this base classification in our evaluations). It is particularly interesting to see the performance of relevance feedback on such a database, since CAD offers an important application domain for content-based 3D shape retrieval.

6.2 Evaluation Methods

We test our algorithms in what we call the *two-round protocol*, which can be viewed a particular form of relevance feedback. In the first round, the retrieval machine returns a ranked list of database objects using an unsupervised similarity measure obtained from a set of 3D shape descriptors (cf. Sect. 5). The user is then asked to mark M items starting from the top of the list as either *relevant* $(y = +1)$ or *irrelevant* ($y=-1$). The second round proceeds either by the SVM-RF scheme (cf. Sect. 2) or the score fusion based approach (cf. Sect. 3), both using the set of M items as the training set. Standard relevance feedback simulations proceed in multiple iterations (usually much more than one). New labeled items are added to the training set, which progressively grows and gets semantically more expressive after each round. It is naturally desired that the algorithm return satisfactory results after a small number of iterations. The two-round protocol as considered here is actually standard relevance feedback with a single iteration. In our experiments, we let M to vary from as small as 4 up to 64 with increments of 4. Note that, concerning the size of the training set, a large M is practically equivalent to allowing a large number of iterations in standard relevance feedback; the only difference is that, in the two-round protocol, the training set is formed in one shot but not as a result of a progressive accumulation. While standard relevance feedback can provide a more detailed analysis of the recall capability through iterations; from a benchmarking perspective, we think that the two-round protocol is a more direct way of assessing the behavior of the algorithms under a small training set size.

The user behavior is simulated using the available ground-truth class information. Accordingly, for a given dataset, we consider each 3D model as a query and the remaining models as database items (this is basically a leave-one-out procedure). When it comes to generating the relevance label of a database item (from the set of the first M items returned after the first round), we compare the class names (or *tags*) of the query and the database item. If the class names match, we set the relevance label to $+1$ (*relevant*); otherwise we set it to -1 (*irrelevant*). This simulation model assumes that the user's judgments are in line with common knowledge that generated the ground truth information associated with the considered 3D datasets. To give an example, it is natural to state that a 3D object tagged as *horse* is relevant to another *horse* object. With this user model, it is also possible to consider *tag* hierarchies (e.g., for retrieving *four-legged* animal objects), but we do not pursue this kind of analysis as it lies beyond the purpose of the present work.

In our comparative analyses, we use the *nearest-neighbor* (NN), *precision-recall curve* and *discounted cumulative gain* (DCG) performance measures. NN is simply the percentage of the correct matches among the closest ones. For a fixed number N of retrieved items, *recall* measures the ratio of correct matches with respect to the size of the query class and *precision* measures the ratio of correct matches within the N retrieved items. By varying N, one obtains a series of precision-recall points, each of which corresponds to a fixed value of N. These points are then interpolated and displayed in terms of the so-called *precision-recall* curve. DCG is a statistic that weights correct results near the front of the list higher than those appearing later. It provides a compact summary of the overall retrieval performance. To calculate this measure, the ranked list of retrieved objects is converted to a list, where an element L_n is one if the nth object is in the same class as the query and otherwise it is zero.

Discounted cumulative gain at the nth rank is then defined as

$$\mathrm{DCG}_n = \begin{cases} L_n, & \text{if } n=1,\\ \mathrm{DCG}_{n-1} + \frac{L_n}{\log_2 n}, & \text{otherwise.}\end{cases}$$

The final DCG score for a query in class $\mathcal{C}$ is the ratio of $\mathrm{DCG}_{N_{\max}}$ to the maximum possible DCG that would be achieved if the first $|\mathcal{C}|$ retrieved elements were in the class $\mathcal{C}$, where $N_{\max}$ is the total number of objects in the database. DCG has normalized values in the range [0, 1] and higher values reflect better performance. In order to give the overall performance on a database, the DCG values for each query are averaged to yield a single average performance figure.

6.3 The Effect of Descriptor Chunking in Score Fusion

We investigated the effect of descriptor chunking (cf. Sect. 5.3) with our score fusion scheme on plain distances. We varied the number of chunks K_{rad}, K_{tp}, and K_{sec} from 1 (where $K=3$ in (8)) up to the respective descriptor dimensions p, which were 1024, 1024, and 576 for *radial*, *t-plane* and *sec-order* descriptors (where $K=2624$ in (8)). DCG profiles in the two-round protocol on PSB training set (907 objects in 90 classes) are shown in Fig. 3 for several choices of $(K_{rad}, K_{tp}, K_{sec})$. Also shown with a hor-

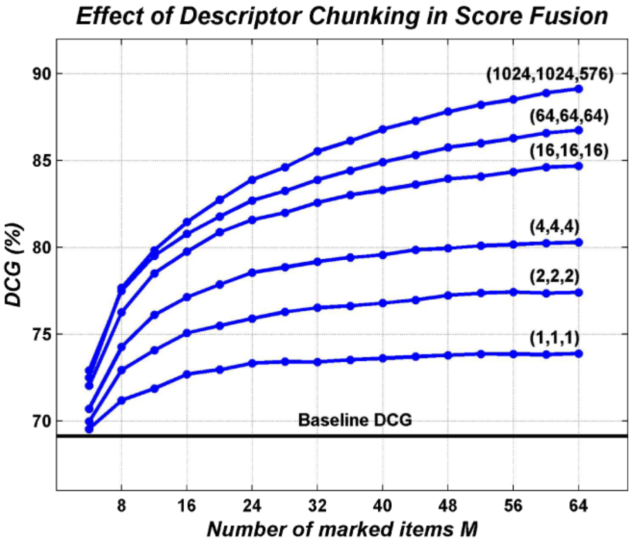


Fig. 3 DCG profiles in the two-round protocol for several choices of $(K_{rad}, K_{tp}, K_{sec})$ using score fusion on plain distances on PSB training set. The triples on top of each *curve* stand for $(K_{rad}, K_{tp}, K_{sec})$

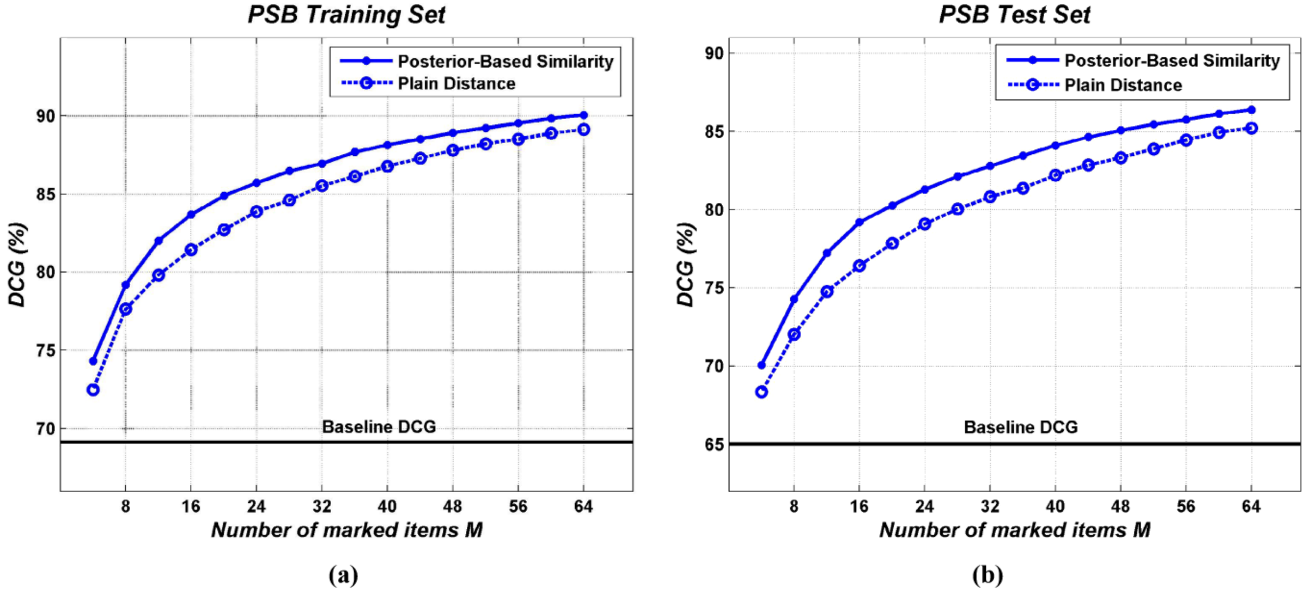


Fig. 4 DCG profiles for score fusion using posterior-based similarities and plain distances on (**a**) PSB Training Set and (**b**) PSB Test Set

izontal line is the baseline DCG, i.e., the performance after the first round. Note that the triples appearing on top of each curve are instances of $(K_{rad}, K_{tp}, K_{sec})$ and the sum $K_{rad} + K_{tp} + K_{sec}$ always gives the total number of scores K involved in the score fusion stage. As can be clearly seen from Fig. 3, when we increase the level of decomposition, i.e., the number of chunks, the DCG after the second round increases for all values of the feedback size M. Top performance is obtained in the limiting case of $p' = 1$, that is, when $(K_{rad}, K_{tp}, K_{sec}) = (1024, 1024, 576)$. The chunking operation adds more degrees of freedom to score fusion, hence induces a more flexible similarity measure adapting itself to the given query. In other words, when $p' = 1$, each component in the descriptor vector becomes equipped with its own adjustable weight that is estimated via ranking risk minimization. In the subsequent experiments, we always report the results corresponding to this limiting case of $p' = 1$.

6.4 Posterior-Based Similarities vs. Plain Distances in Score Fusion

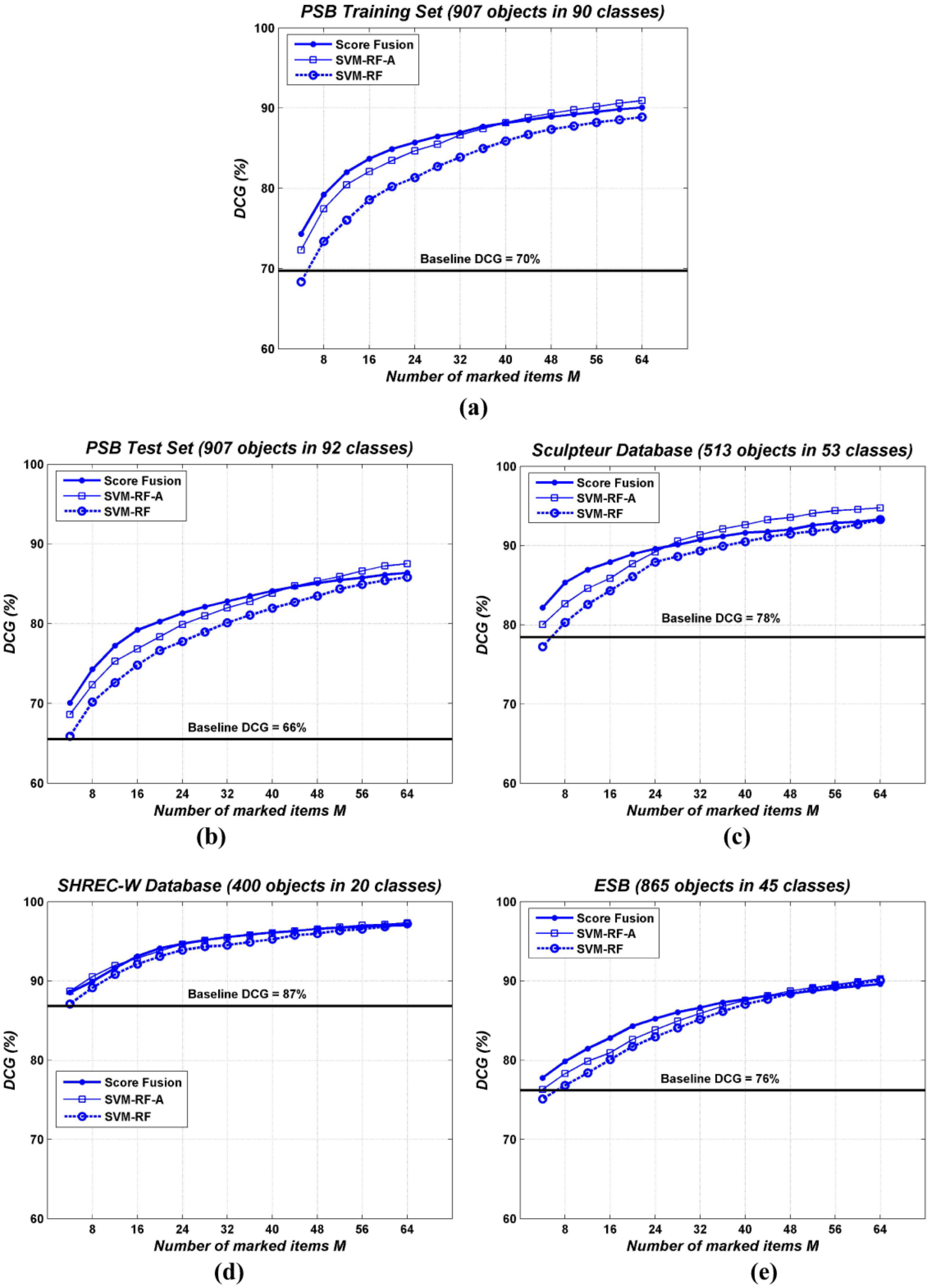
In this section, we validate the conjecture that the posterior-based similarity model is more advantageous for score fusion than using plain distances. The posterior model parameters are learned on PSB Training Set using Algorithm 2 (cf. Sect. 4). Figure 4(a) depicts the DCG convergence profiles corresponding to posterior-based similarities and plain distances on PSB Training Set. For all values of M, the score fusion with posterior-based similarities has better retrieval performance than with plain distances.

It is important to stress that the posterior-based approach generalizes well to the instances that are unseen during the posterior model learning. To confirm this, we have conducted the same comparison on PSB Test Set using the posterior model parameters learned on PSB Training Set. Figure 4(b) shows that score fusion posterior-based similarities is superior on PSB Test Set as well.

6.5 Score Fusion vs. SVM-RF

In Sects. 5.3 and 5.4, we have experimentally shown that the proposed score fusion approach attains its best performance with descriptor chunking down to single-entry level using posterior-based similarities. Now, we carry out a one-to-one comparison against the standard SVM-RF scheme. For SVM-RF, we have used the Gaussian radial basis kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \exp(-\rho\|\mathbf{x} - \mathbf{x}'\|^2)$, where ρ is a parameter that was observed to drastically affect the SVM-RF performance. An inappropriate selection might yield surprisingly poor results. Unfortunately, as in most of the SVM problems, training data give no prior indication about the optimal value of this parameter. In our context, a time-consuming grid search on PSB Training Set over a broad range of values has revealed that $\rho^* = 200$ was the best option. Regarding the choice of the regularization parameter C in SVM optimization, we observed that several settings ($C = 0.1, C = 1, C = 10$ and $C = 100$) provided practically equivalent results. We set the parameter C to 10 in all cases involving SVM optimization.

Figure 5(a) depicts the DCG profiles of the score fusion and SVM-RF approaches on PSB Training Set. We included two context-dependent variants of SVM-RF in the comparison: SVM-RF without descriptor alignment and with descriptor alignment (cf. Sect. 5.2), denoted as SVM-RF-A.



(a)

(b)

(c)

(d)

(e)

Fig. 5 DCG profiles for score fusion, SVM-RF with descriptor alignment (SVM-RF-A) and SVM-RF without descriptor alignment on PSB Training Set on (**a**) PSB Training Set, (**b**) PSB Test Set, (**c**) Sculpteur Database, (**d**) SHREC-W, and (**e**) ESB

From Fig. 5, we observe that SVM-RF without alignment has the worst performance albeit it still enhances the first round results as a function of M. Score fusion exhibits faster improvement and it is markedly better for $M \leq 40$, while SVM-RF-A overtakes it slightly but only after $M > 40$, showing that standard SVM-RF requires a larger training

Table 4 DCG gains of score fusion (SF) and SVM-RF with respect to baseline performance

	$M=4$		$M=8$		$M=16$	
	SF	SVM-RF-A	SF	SVM-RF-A	SF	SVM-RF-A
PSB Train	4.6	3.2	9.5	8.3	14.0	13.0
PSB Test	4.5	3.1	8.8	6.8	13.7	11.3
SCU	3.7	1.6	6.9	4.2	9.5	7.4
SHREC-W	1.7	1.8	3.1	3.7	6.3	6.0
ESB	1.6	0.1	3.7	2.1	6.6	4.8

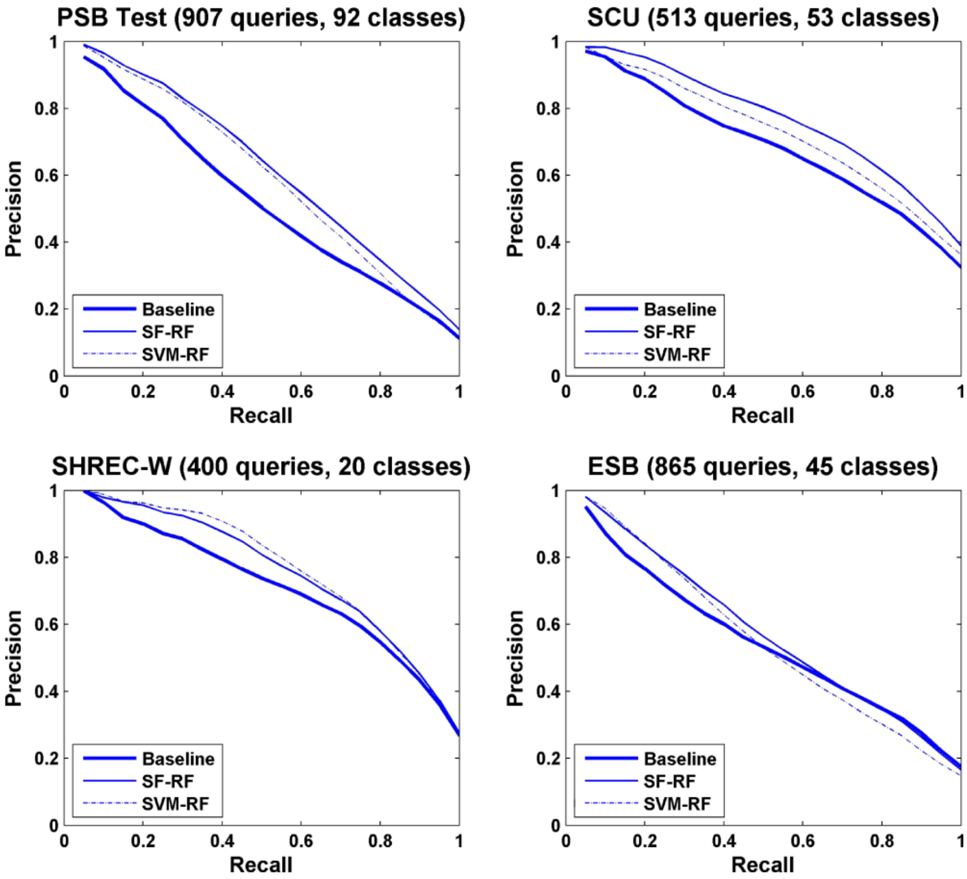


Fig. 6 Precision-recall curves for the case of $M=8$ on all databases

set than score fusion. Similar observations can be made for other databases as can be seen from Figs. 5(b–e). Note that in obtaining these results we always used the posterior model parameters learned from PSB Training Set. The DCG profiles for PSB Test Set (Fig. 5(b)) follow virtually the same pattern as for PSB Training Set, starting from a lower baseline. For Sculpteur (Fig. 5(c)), the break-even point between score fusion and SVM-RF-A occurs at $M=28$, earlier than in PSB. For SHREC-W (Fig. 5(d)), where the baseline is already very high, the profiles for score fusion and SVM-RF-A are virtually identical. For ESB (Fig. 5(e)), SVM-RF-A cannot surpass score fusion even for high values of M.

Table 4 provides a closer view of the performance for the practical small sample cases of $M=4$, 8, and 16. Additive DCG gains, defined as the difference between the DCG obtained with relevance feedback and the baseline DCG, reveals that score fusion is markedly superior to SVM-RF-A, except for SHREC-W where the comparison is rather inconclusive (DCG differences are in the order of decimals). We complete this analysis with precision-recall curves for the case of $M=8$ on all databases. The curves displayed in Fig. 6 corroborate the performance results measured by DCG. On a general basis, we can state that score fusion has better small-sample performance than SVM-RF. This aspect makes the posterior score fusion approach more appealing for relevance feedback because, admittedly, it is always bet-

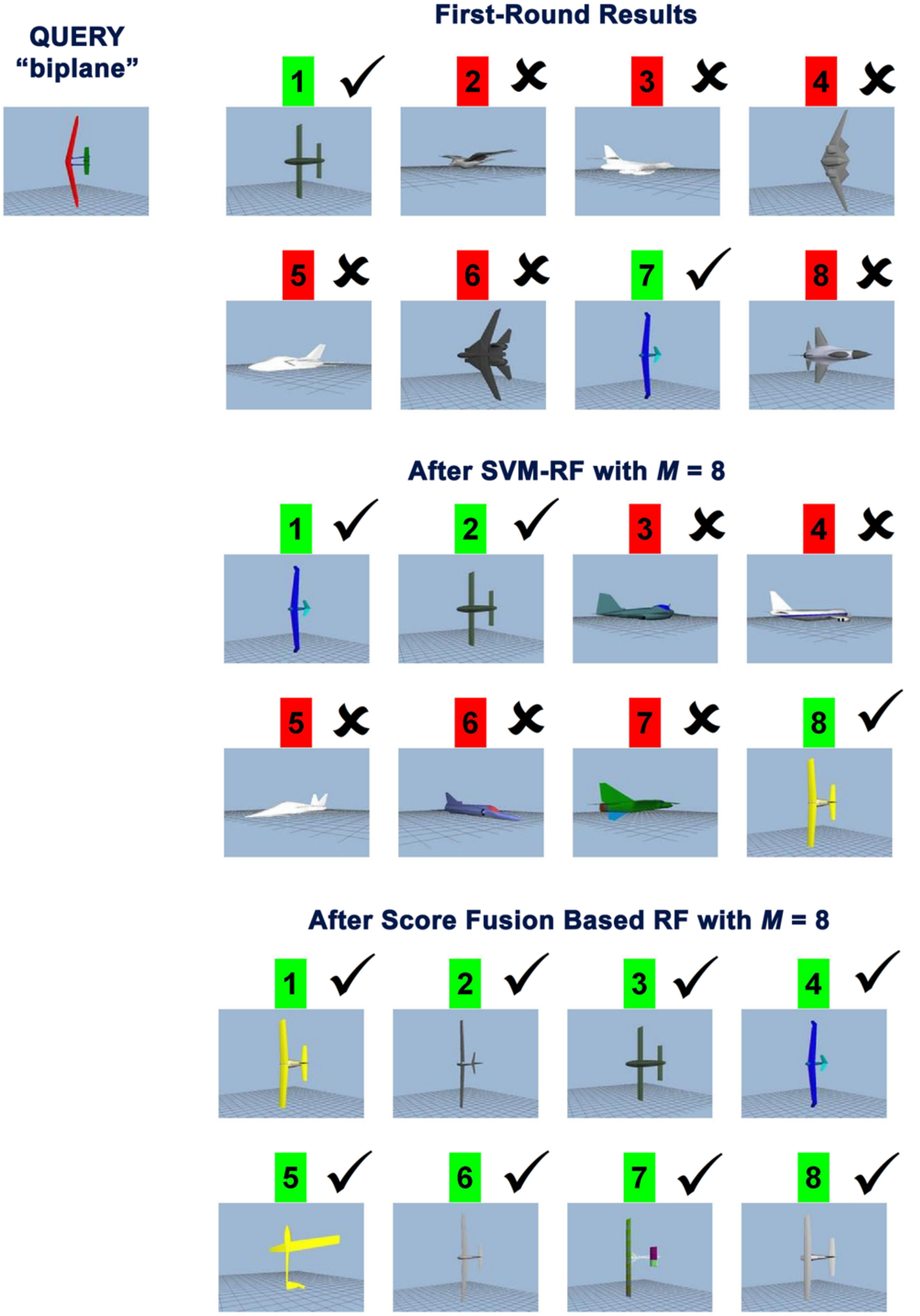


Fig. 7 (Color online) A sample "biplane" query from PSB: first-round results (*top*), after SVMRF (*middle*), and after score fusion based RF (*bottom*). Items marked with a *check sign* are relevant and items marked with a *cross sign* are irrelevant to the query

ter to demand less from the user. Figure 7 illustrates the case in point with a sample "biplane" query from PSB.

6.6 Score Fusion without User Feedback

In this section, we show the applicability of our score fusion scheme in no-feedback situations. Recall first that we have a linear similarity model given by $S(X, Q) = \langle \mathbf{w}, \mathbf{s} \rangle$, where X is a database item and Q is a query. In the RF context, the parameter vector $\mathbf{w}$ is learned *on-line* using a small set of items that are provided by the user in view of their relevance to the query. Thus, $\mathbf{w}$ implicitly depends on Q and one can view the function $S(X, Q)$ as a *locally-adaptive* weighted similarity measure. To apply our score fusion scheme to the no-feedback case, we rely on a continuity argument and make the following assumption: *queries that are similar in the descriptor space should induce similar weight vectors*. Accordingly, we can avoid the on-line estimation of $\mathbf{w}$ by replacing it with an "approximate" version $\hat{\mathbf{w}}$ corresponding to a training query $\hat{Q}$. In this variant of score fusion, given

Table 5 DCGs and additive DCG gains for classes shared by both PSB training and test sets

Class	Baseline	SF-OFF		SF-ON with $M = 4$	
biplane	91.8	99.7	+7.9	92.5	+0.7
commercial airplane	76.3	75.5	−0.8	87.5	+11.2
fighter jet	93.1	92.9	−0.2	93.4	+0.3
helicopter	63.0	74.5	+11.5	72.5	+9.5
enterprise spaceship	70.8	77.6	+6.8	77.4	+6.6
human	92.4	94.4	+2	93.5	+1.1
human (arms out)	74.3	83.6	+9.3	78.3	+4.0
sword	68.4	78.0	+9.6	66.4	−2.0
face	83.1	84.4	+1.3	88.1	+5.0
head	85.8	85.9	+0.1	89.2	+3.4
two-story home	36.8	35.6	−1.2	41.6	+4.8
city	70.0	66.3	−3.7	68.9	−1.1
dining chair	71.7	79.6	+7.9	73.3	+1.6
shelves	66.2	73.2	+7	72.2	+6
rectangular table	67.3	69.3	+2	68.4	+1.1
handgun	89.6	97.0	+7.4	97.4	+7.8
vase	42.7	41.2	−1.5	45.1	+2.4
potted plant	54.3	57.9	+3.6	60.1	+5.8
barren tree	43.1	45.5	+2.4	54.1	+11
ship	69.6	78.9	+9.3	73.3	+3.7
sedan car	95.5	93.4	−2.1	97.5	+2
AVERAGE	75.6	79.0	+3.4	78.9	+3.3

an on-line query Q and a set of training queries $\{Q_{train}^{(n)}\}$ whose weight vectors $\{\mathbf{w}_{train}^{(n)}\}$ have already been learned off-line, we first identify the best matching training query by $\hat{Q} = \arg\min_n dist(Q, Q_{train}^{(n)})$, where the distance is evaluated between the corresponding descriptors. We then fetch the weight vector $\hat{\mathbf{w}}$ corresponding to $\hat{Q}$ and return the retrieval results using the "approximate" similarity function $S(X, Q) = \langle \hat{\mathbf{w}}, \mathbf{s} \rangle$.

Table 5 reports the performance of this off-line scheme, denoted as SF-OFF in order to differentiate from the on-line version SF-ON for 21 shape classes in PSB. In these experiments, 3D models from PSB Training Set are used as training queries, those from PSB Test Set as test queries and the performance is evaluated using PSB Test Set base classification. The set of weights $\{\mathbf{w}_{train}^{(n)}\}$ are learned off-line by a leave-one-out procedure on the PSB Training Set, so that at a given time, a 3D model in the set is considered as the query and the remainder as the set of database items. To construct the associated training set, we opted for robustness and labeled all the database items using the available relevance information (in the on-line version, this corresponds to letting M as large as the number of all database items). We restricted the analysis to the classes that are shared by both PSB Training and Test Sets because of the continuity assumption. In other words, we had to make sure that there is actually a shape among the training queries, which is semantically relevant to the test query. Without such a restriction, the "approximate" similarity function $S(X, Q) = \langle \hat{\mathbf{w}}, \mathbf{s} \rangle$ might be very inaccurate, since there will always be a best-matching training query albeit semantically irrelevant to the test query. Table 5 shows that this off-line variant of score fusion leads to substantial improvements for the majority of the considered shape classes (a few classes suffer from minor degradations). On average, an additive DCG gain of 3.4% is obtained with respect to the baseline, equivalent to the average performance of score fusion based RF (SF-ON) with $M = 4$ on the considered classes.

7 Discussion and Conclusion

There is an alternative perspective to look at our score fusion approach. An admittedly sound conjecture for retrieval is that the full relevance posterior $\mathbb{P}(y = 1 | \{X_k, Q_k\}_{k=1}^{K})$ would be the ideal similarity measure to decrease the semantic gap that arises from semantic uncertainties and descriptor imperfections. However, direct estimation of this full relevance posterior is a difficult task as the joint dimension of the given descriptor information $\{X_k, Q_k\}_{k=1}^{K}$ is very high and, for any practical purpose in relevance feedback, available training instances are scarce. The score fusion approach that we introduced in this work can be viewed as an operational approximation to the full relevance posterior. First, with descriptor chunking down to single-entry level, the difficult problem of estimating the full relevance posterior $\mathbb{P}(y = 1|\{X_k, Q_k\}_{k=1}^K)$ is cast into several much simpler problems of tractable dimension, where elementary (but also much less discriminative) relevance posteriors $\mathbb{P}(y = 1|X_k, Q_k)$ are estimated independently with *off-line* discriminative learning (cf. Sect. 4). Then, at the relevance feedback stage, the importance (measured by the weight w_k) of each such posterior is estimated by *on-line* ranking risk minimization (cf. Sect. 3). On a conceptual level, this fusion scheme resembles boosting methods (Hastie et al. 2001), where several weak classifiers are combined into one strong classifier with much better performance. Our main conclusion is that the joint use of off-line and on-line learning in score fusion makes our algorithm more effective than the prevailing SVM-RF approach, as established by experiments on several 3D object databases. In particular, the markedly better small sample behavior of score fusion is advantageous for relevance feedback-driven retrieval.

The findings of the present work also suggest that near perfect performance on standard 3D benchmarks can be obtained by combining many different shape descriptors in a supervised setting. The current DCG performance on PSB Test Set is already impressive, at around 80% with $M = 16$ marked items ($\sim$14% more than the unsupervised baseline obtained with density-based descriptors alone). We believe that the addition of other powerful descriptors such as CRSP (Papadakis et al. 2007) and DSR (Vranic 2004) would allow even further improvements, we hope above 90% DCG. Our score fusion algorithm is computationally flexible enough to handle such extensions. Furthermore, this flexibility calls also for other challenging multimedia retrieval tasks. Other research directions that we plan to pursue in the future include generative learning of relevance posteriors via Bayesian modeling and extending the ranking algorithm to ordinal relevance relations.

References

Akgül, C. B. (2007). *Density-based shape descriptors and similarity learning for 3D object retrieval*. Ph.D. thesis, Dept. Signals-Images, Télécom ParisTech, Paris.

Akgül, C. B., Sankur, B., Yemez, Y., & Schmitt, F. (2008). Similarity score fusion by ranking risk minimization for 3D object retrieval. In *Proceedings of the Eurographics workshop on 3D object retrieval*.

Akgül, C. B., Sankur, B., Yemez, Y., & Schmitt, F. (2009). 3D model retrieval using probability density-based shape descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(6), 1117–1133.

Bustos, B., Keim, D. A., Saupe, D., Schreck, T., & Vranic, D. V. (2005). Feature-based similarity search in 3D object databases. *ACM Computing Surveys*, *37*(4), 345–387.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chen, Y., Zhou, X., & Huang, T. S. (2001). One-class SVM for learning in image retrieval. In *Proceedings of the IEEE international conference on image processing* (pp. 815–818).

Clémençon, S., Lugosi, G., & Vayatis, N. (2008). Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics*, *36*(2), 844–874.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys*, *40*(2), 1–60.

Giacinto, G., & Roli, F. (2004). Bayesian relevance feedback for content-based image retrieval. *Pattern Recognition*, *37*(7), 1499–1508.

Giorgi, D., Biasotti, S., & Paraboschi, L. (2007). Shape retrieval contest 2007: watertight models track. In R. C. Veltkamp & F. B. T. Haar (Eds.), *SHREC2007: 3D shape retrieval contest*, Technical Report UU-CS-2007-015 (pp. 5–10).

Goodall, S., Lewis, P. H., Martinez, K., Sinclair, P. A. S., Giorgini, F., Addis, M., Boniface, M. J., Lahanier, C., & Stevenson, J. (2004). SCULPTEUR: multimedia retrieval for museums. In *Proceedings of the international conference on image and video retrieval* (pp. 638–646).

Guo, G., Jain, A. K., Ma, W., & Zhang, H. (2002). Learning similarity measure for natural image retrieval with relevance feedback. *IEEE Transactions on Neural Networks*, *12*(4), 811–820.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: data mining, inference, and prediction*. Berlin: Springer.

Huang, J., Kumar, S. R., & Mitra, M. (1997). Combining supervised learning with color correlograms for content-based image retrieval. In *Proceedings of the ACM international conference on multimedia* (pp. 325–334). New York: ACM Press.

Ishikawa, Y., Subramanya, R., & Faloutsos, C. (1998). MindReader: query databases through multiple examples. In *Proceedings of the international conference on very large databases* (pp. 218–227).

Jayanti, S., Kalyanaraman, Y., Iyer, N., & Ramani, K. (2006). Developing an engineering shape benchmark for CAD models. *Computer-Aided Design*, *38*(9), 939–953.

Laaksonen, J., Koskela, M., & Oja, E. (1999). PicSOM: self-organizing maps for content-based image retrieval. In *Proceedings of the INNS-IEEE international joint conference on neural networks* (pp. 1199–1207).

Leifman, G., Meir, R., & Tal, A. (2005). Semantic-oriented 3D shape retrieval using relevance feedback. *The Visual Computer*, *21*(8), 865–875.

Lin, H.-T., Lin, C.-J., & Weng, R. C. (2007). A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, *68*(3), 267–276.

Nastar, C., Mitschke, M., & Meilhac, C. (1998). Efficient query refinement for image retrieval. In *Proceedings of the IEEE international conference on computer vision and pattern recognition* (pp. 547–552).

Novotni, M., Park, G.-J., Wessel, R., & Klein, R. (2005). Evaluation of kernel based methods for relevance feedback in 3D shape retrieval. In *Proceedings of the international workshop on content-based multimedia indexing*.

Papadakis, P., Pratitakis, I., & Theoharis, T. (2007). Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognition*, *40*(9), 2437–2452.

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

Picard, R. W., Minka, T. P., & Szummer, M. (1999). Modeling user subjectivity in image libraries. In *Proceedings of the IEEE international conference on image processing* (pp. 777–780).

Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In A. Smola et al. (Eds.), *Advances in large margin classifiers* (pp. 61–74). Cambridge: MIT Press.

Porkaew, K., Mehrotra, S., & Ortega, M. (1999). Query reformulation for content-based image retrieval in MARS. In *Proceedings of the IEEE international conference on multimedia computing and systems* (pp. 747–751).

Rocchio, J. J. (1966). *Document retrieval system—optimization and evaluation*. Ph.D. thesis, Harvard Computational Lab, Harvard University, Cambridge, MA.

Rui, Y., & Huang, T. S. (2000). Optimizing learning in image retrieval. In *Proceedings of the IEEE international conference on computer vision and pattern recognition* (pp. 236–243).

Rui, Y., Huang, T. S., Ortega, M., & Mehrotra, S. (1998). Relevance feedback: a power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, *8*(5), 644–655.

Santini, S., & Jain, R. (2000). Integrated browsing and querying for image database. *IEEE Transactions on Multimedia*, *7*(3), 26–39.

Schettini, R., Ciocca, G., & Gagliardi, I. (1999). Content-based color image retrieval with relevance feedback. In *Proceedings of the IEEE international conference on image processing* (pp. 75–79).

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge: MIT Press.

Shilane, P., Min, P., Kazhdan, M., & Funkhouser, T. (2004). The Princeton shape benchmark. In *Proceedings of the shape modeling international conference* (pp. 167–178).

Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1349–1380.

Tangelder, J.-W. H., & Veltkamp, R.-C. (2008). A survey of content-based 3D shape retrieval methods. *Multimedia Tools and Applications*, *39*, 441–471.

Tao, D., Tang, X., Li, X., & Wu, X. (2006). Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(7), 1088–1099.

Tieu, K., & Viola, P. (2000). Boosting image retrieval. In *Proceedings of the IEEE international conference on computer vision and pattern recognition* (pp. 228–235).

Tong, S., & Chang, E. (2001). Support vector machine active learning for image retrieval. In *Proceedings of the ACM international conference on multimedia* (pp. 107–118). New York: ACM Press.

Vasconcelos, N., & Lippman, A. (1999). Learning from user feedback in image retrieval systems. In *Proceedings of the neural information processing systems* (Vol. 12).

Vasconcelos, N., & Lippman, A. (2000) A probabilistic architecture for content-based image retrieval. In *Proceedings of the IEEE international conference on computer vision and pattern recognition* (pp. 216–221).

Vranic, D. V. (2004). *3D model retrieval*. Ph.D. thesis, University of Leipzig.

Wu, Y., Tian, Q., & Huang, T. S. (2000). Discriminant EM algorithm with application to image retrieval. In *Proceedings of the IEEE international conference on computer vision and pattern recognition* (pp. 222–227).

Zhang, L., Lin, F., & Zhang, B. (2001). Support vector machine learning for image retrieval. In *Proceedings of the IEEE international conference on image processing* (pp. 721–724).

Zhou, X. S., & Huang, T. S. (2001). Small sample learning during multimedia retrieval using BiasMap. In *Proceedings of the IEEE international conference on computer vision and pattern recognition* (pp. 11–17).

Zhou, X., & Huang, T. S. (2003). Relevance feedback for image retrieval: a comprehensive review. *Multimedia Systems*, *8*(6), 536–544.