Eurographics Workshop on 3D Object Retrieval (2008), pp. 1–8
I. Pratikakis and T. Theoharis (Editors)

Similarity Score Fusion by Ranking Risk Minimization for 3D Object Retrieval

Submission ID: 1005

Abstract

In this work, we introduce a score fusion scheme to improve the 3D object retrieval performance. The state of the art in 3D object retrieval shows that no single descriptor is capable of providing fine grain discrimination required by prospective 3D search engines. The proposed fusion algorithm linearly combines similarity information originating from multiple shape descriptors and learns their optimal combination of weights by minimizing the empirical ranking risk criterion. The algorithm is based on the statistical ranking framework [CLV07], for which consistency and fast rate of convergence of empirical ranking risk minimizers have been established. We report the results of ontology-driven and relevance feedback searches on a large 3D object database, the Princeton Shape Benchmark. Experiments show that, under query formulations with user intervention, the proposed score fusion scheme boosts the performance of the 3D retrieval machine significantly.

Categories and Subject Descriptors (according to ACM CCS): H.3.3 [Information Search and Retrieval]: Retrieval Models I.5.1 [Models]: Statistical

This work was supported by Boğaziçi University Project No. 05HA203.

1. Introduction

Next generation search engines will enable query formulations, other than text, relying on visual information encoded in terms of images and shapes. The 3D search technology, in particular, targets specialized application domains ranging from computer aided design to molecular data analysis. In this search modality, the user picks a query from a catalogue of 3D objects and requests from the retrieval machine to return a set of "similar" database objects in decreasing relevance. 3D object retrieval hinges on shape matching, that is, determining the extent to which two shapes resemble each other. Shape matching is commonly done by reducing the characteristics of the shapes to vectors or graph-like data structures, called *shape descriptors* [BKS*05, TV04, IJL*05], and then, by evaluating the similarity degrees between the descriptor pairs. We call the similarity degree between two descriptors as *the matching score* between two shapes. In the retrieval mode, the matching scores between a query and each of the database objects are sorted. The retrieval machine then displays database objects in descending order of the scores. Effective retrieval means that the objects displayed in the upper part of the list better match to the query object than the rest of the list.

Ongoing research in 3D object retrieval shows that no single shape descriptor is capable of providing satisfactory retrieval performance for a broad class of shapes and independently of the associated semantics [TV04, SMKF04]. Figure 1 displays the response of two different descriptors from the density-based framework [ASYS07a], A and B, to two different queries from the Princeton Shape Benchmark (PSB) [SMKF04]. The first query is a biplane model and the second one is a chair model. In response to the biplane, descriptor A returns correctly four biplanes in the first three and in the sixth matches, while the fourth and the fifth retrieved models are not biplanes, but still flying objects that can be considered as relevant. Descriptor B, on the other hand, returns models that are completely irrelevant to the biplane query (three shelf models, two coarse human models and a microscope!). For the chair query, Descriptor B is more successful since it has retrieved six chair models; while descriptor A, after first three correct matches, returns two tree models and a monument! Thus, the adequacy of the descriptors A or B depends on the nature of the query. Furthermore, these examples can be multiplied; for instance, there are cases where sets of relevant matches for different descriptors are even disjoint. Much like in the case of classifier construction, we conjecture that improved retrieval algo-

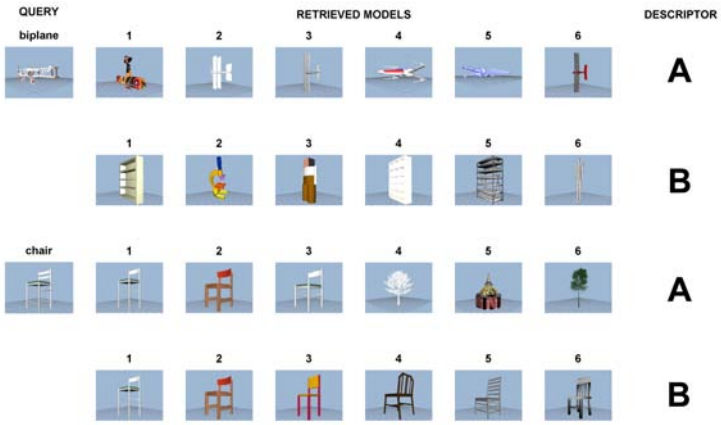

Figure 1: *Response of two different descriptors A and B to two different queries* biplane *and* chair

rithms can be built by using diverse set of descriptors/scores provided there is a practical way to "fuse" them.

As the example in Figure 1 illustrates, experimental evidence motivates us to consider a score fusion scheme that uses a certain amount of supervision. The question here can be formulated as follows. How can one combine a set of similarity scores $\{s_k\}$ into a final scoring function $\varphi = \sum_k w_k s_k$ to achieve better retrieval result than with anyone of them? In the present work, we tackle this score fusion problem by minimizing a convex regularized version of the empirical risk associated with ranking instances. We follow the statistical learning framework developed in [CLV07] and identify that learning a linear scoring function can be cast into a binary classification problem in the score difference domain. Given a query, the set of weights $\{w_k\}$ found as the solution of the binary classification problem can be considered as optimal also with respect to the empirical risk associated with ranking instances. Statistical ranking has found many applications such as text-based information retrieval and collaborative filtering [HGO00, Joa02, FISS03]. Our score fusion approach can be employed with different types of supervisory information provided by the user, as in ontology-driven search and relevance feedback.

The contribution of the present work is three-fold. First, to the best of our knowledge, there is no prior work in the 3D domain using statistical ranking techniques to combine shape similarity information coming from different descriptors. Second, the present work is a direct application of a recently introduced rigorous statistical framework [CLV07], where consistency and fast rate of convergence of empirical ranking risk minimizers have been established. Third, our algorithm operates on scores, and not on descriptors themselves, unlike other risk minimization-based approaches [HGO00]. This adds greater generality and flexibility to our approach for a broad spectrum of retrieval applications.

The paper is organized as follows. In Section 2, we introduce the score fusion problem and give a solution based on support vector machines (SVM) [HTF01]. We also explain the use of our score fusion algorithm in two different protocols, *bimodal* and *two-round* searches, which can be viewed as particular instances of ontology-driven search and relevance feedback respectively. In Section 3, we give an overview of the density-based framework [ASYS07a] that we use for shape description. In Section 4, we experiment on PSB [SMKF04] and show the degree by which we can boost the retrieval performance of density-based shape descriptors using the proposed score fusion algorithm. In the final Section 5, we conclude and discuss further research directions.

2. Score Fusion by Ranking Risk Minimization

2.1. The Score Fusion Problem

Consider the problem of ranking two generic database shapes x and x' based on their relevance to a query shape q. Suppose that we have access to K similarity values s_k and s'_k for each of the pairs (x,q) and (x',q) respectively. These K similarity measures can be obtained from different descriptor sets and/or by using different metrics operating on the same set of descriptors. In our context, a similarity value $s_k \triangleq sim_k(x,q)$ arises from a certain shape descriptor and reflects some, possibly different, geometrical and/or topological commonality between the database shape x and the query shape q. An ideal similarity measure should score higher for similar shape pairs as compared to less similar ones. In retrieval problems, a shape x in the database that is more similar to the query q is expected to be ranked higher than any other intrinsically less similar shape x'. These similarity values/scores can be written more compactly in the vector form as $\mathbf{s} = [s_1, \ldots, s_K] \in \mathbb{R}^K$. Our objective is to build a scalar-valued final scoring function φ of the form $\varphi(x,q) = \langle \mathbf{w}, \mathbf{s} \rangle$, where $\mathbf{w} = [w_1, \ldots, w_K] \in \mathbb{R}^K$ is a vector, whose components

form the weights of the corresponding scores s_k. The scoring function φ should assign a higher score to the more relevant shape, i.e., it should satisfy the following property:

$$\begin{array}{ll} \varphi(x,q) > \varphi(x',q) & \text{if } x \text{ is more relevant to } q \text{ than } x', \\ \varphi(x,q) < \varphi(x',q) & \text{otherwise,} \end{array}$$

where ties are arbitrarily broken. The relevance of the shapes x and x' to the query q can be encoded by indicator variables y and y' respectively. In this work, we assume crisp relevances $y = 1$ (*relevant*) and $y = -1$ (*not relevant*), in which case, the above property reads as:

$$\begin{array}{ll} \varphi(x,q) > \varphi(x',q) & \text{if } y - y' > 0, \\ \varphi(x,q) < \varphi(x',q) & \text{if } y - y' < 0. \end{array}$$

The function φ must subsume the similarity information residing in the individual scores s_k in order to emulate the ideal similarity notion between shapes, hence to achieve a better retrieval performance. Given the linear form $\varphi(x,q) = \langle \mathbf{w}, \mathbf{s} \rangle$, score fusion can be formulated as the problem of finding a weight vector $\mathbf{w}$, which is optimal according to some criterion, as we explain in the following section.

2.2. Ranking Risk Minimization

The criterion of interest is the so-called *empirical ranking risk* (*ERR*) defined as the number of misranked pair of database shapes (x_m, x_n) with respect to a query q. Formally, we can write this criterion as:

$$ERR(\varphi; q) = \frac{1}{T} \sum_{m<n} \mathbb{I}\{(\varphi(x_m,q) - \varphi(x_n,q)) \cdot (y_m - y_n) < 0\}. \tag{1}$$

where T is the number of shape pairs (x_m, x_n) and $\mathbb{I}\{\cdot\}$ 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 misranked shape pairs in the database with respect to the query. Basically, if $\varphi(x_m,q) < \varphi(x_n,q)$ but $y_m > y_n$, the scoring function $\varphi(\cdot,q)$ (wrongly) assigns a higher score to x_n than to x_m while x_m is relevant to the query q but x_n is not. Thus the scoring function has made an error in ranking x_n and x_m with respect to the query q and *ERR* should be incremented by one. Such misrankings are naturally undesirable and our task is to find a scoring function (or more appropriately its parameters $\mathbf{w}$) so that the number of misranked pairs is as small as possible.

We can identify *ERR* as an empirical classification error. To see this, first let $z \triangleq (y - y')/2$, taking values within $\{-1, 0, 1\}$. We observe then the following:

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

When $z = 0$, i.e., if shapes x and x' are both relevant ($y = y' = 1$) or both *not* relevant ($y = y' = -1$), we have no particular preference in ranking them with respect to each other (we can decide arbitrarily). Corresponding to each non-zero z, we can define *a score difference vector* $\mathbf{v}$, which is given simply by $\mathbf{v} \triangleq \mathbf{s} - \mathbf{s}'$, the difference between the score vectors $\mathbf{s}$ and $\mathbf{s}'$ of the shapes x and x' respectively. With this new notation and writing the scoring function φ explicitly in terms of its parameters $\mathbf{w}$, Eq. 1 now reads as

$$ERR(\mathbf{w}; q) = \frac{1}{T} \sum_{m<n} \mathbb{I}\{z_{m,n} \langle \mathbf{w}, \mathbf{v}_{m,n} \rangle < 0\}, \tag{2}$$

where the index pairs (m,n) correspond to pairs of shapes x_m and x_n whose respective relevance labels y_m and y_n are different ($z_{m,n}$ is either 1 or -1). Thus, we have converted *ERR* written in terms of *score* vectors $\mathbf{s}$ and *relevance* indicators y (Eq. 1) into an empirical classification error in terms of *score difference* vectors $\mathbf{v}$ and *rank* indicators z (Eq. 2). In both cases, the sought after parameter vector $\mathbf{w}$ is the same. As is the common practice in statistical learning [HTF01], we replace the 0-1 loss in Eq. 2 with a convex loss function and we add a regularization term on some norm of $\mathbf{w}$ to obtain a tractable convex optimization problem in $\mathbf{w}$. In particular, using the *hinge* loss as the convex loss and the L^2-norm as the regularization term leads to the well-known SVM problem, for which we can find a global solution.

In summary, the problem of finding the parameter vector $\mathbf{w}$ of the linear scoring function φ is the same as the SVM problem in the domain of score difference vectors. The key point here is that *the weight vector learned by SVM in the score difference domain can directly be used to evaluate the scoring function at the matching stage.* We can now summarize the training algorithm to learn the parameter $\mathbf{w}$ of the scoring function φ:

Given

Database shapes x_n
Query q
Relevance labels y_n
K different shape description schemes

(1) Calculate a score vector $\mathbf{s}_n \in \mathbb{R}^K$ for each (x_n, 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 set $\{(\mathbf{v}_{m,n}, z_{m,n})\}_{m<n} \subset \mathbb{R}^K \times \{-1, 1\}$.

2.3. Applications

In this section, we illustrate our score fusion scheme in two different retrieval protocols: (i) *bimodal search* and (ii) *two-round search*.

In *the bimodal protocol*, the user provides a textual description associated with the query shape (see Figure 2). The keyword can be selected from one of the predefined shape concepts. We call this protocol as *bimodal* since the query is formulated in terms of two information modalities, a 3D shape and a concept keyword. This protocol can be

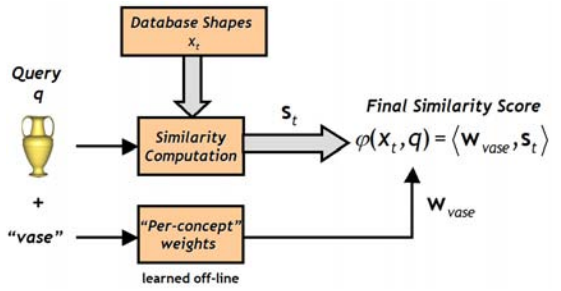

Figure 2: *Bimodal protocol*

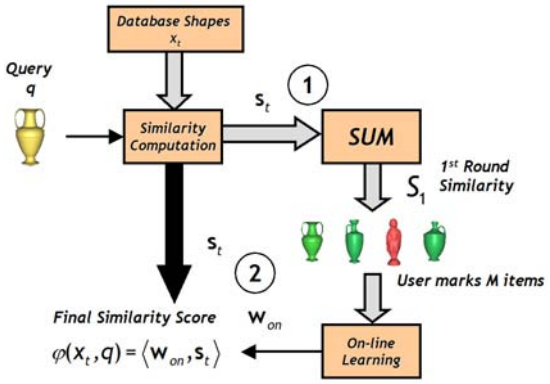

Figure 3: *Two-round protocol* on-line *version*

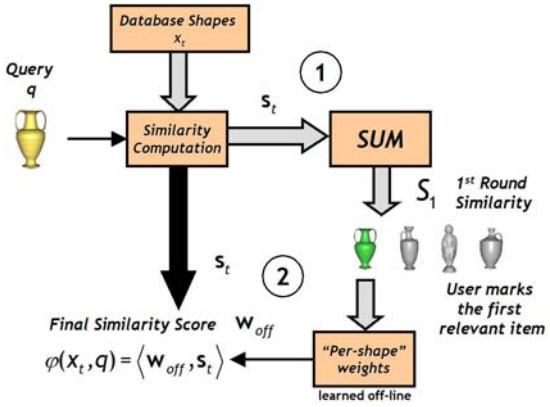

Figure 4: *Two-round protocol* off-line *version*

viewed as an ontology-driven search and necessitates an off-line stage during which the weight vectors associated with each shape concept are learned. Note that the criterion of Section 2.2 is *per-query* and should be extended to a *per-concept* risk $ERR(\mathbf{w},\mathcal{C})$, where $\mathcal{C}$ stands for the working concept. This can be done straight-forwardly by averaging per-query risks associated with a given concept, that is,

$$ERR(\mathbf{w};\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{q\in\mathcal{C}} ERR(\mathbf{w};q),$$

where $|\mathcal{C}|$ is the number of training shapes belonging to $\mathcal{C}$. However, since the minimization should be performed in the score difference domain, the problem turns out to be a very large-scale one even for moderately sized classes. Given a training database $\mathcal{D}$ of size $|\mathcal{D}|$, the number of score difference instances per concept is $|\mathcal{C}| \times (|\mathcal{C}|-1) \times (|\mathcal{D}|-|\mathcal{C}|)$, e.g., for $|\mathcal{D}| = 1000$ and for $|\mathcal{C}| = 10$, the number of training instances becomes ~ 90000, in which case we incur to memory problems using standard SVM packages [CL01]. In order to maintain the generality and practical usability of our approach in this protocol, we develop two heuristics:

- **Average per-query weight vector.** The weight vector $\hat{\mathbf{w}}_{\mathcal{C}}$ for a given shape concept is computed as the average of the per-query weight vectors $\hat{\mathbf{w}}_q$ corresponding to the training shapes within that class, that is,

  $$\hat{\mathbf{w}}_{\mathcal{C}} = \frac{1}{|\mathcal{C}|}\sum_{q\in\mathcal{C}} \hat{\mathbf{w}}_q.$$

  The per-query weight vector $\hat{\mathbf{w}}_q$ is obtained by the algorithm given in Section 2.2. We denote this heuristic by *AVE-W*.
- **Per-class risk minimization using per-query support vectors.** In this second heuristic, we exploit the sparsity of the SVM solution, which means that the per-query weight vector found by the algorithm in Section 2.2 is the weighted sum of usually a small number of training score difference instances, called as *support vectors* (svs) in general SVM terminology. It is a well known fact that, for a given binary classification problem, the SVM solution remains identical when only the svs are provided for training [HTF01]. The svs form a parsimonious surrogate for the set of training instances with exactly the same separability properties in the original data. Accordingly, the learning of a per-concept weight vector can be carried in two stages. First, we identify the svs of per-query problems by per-query minimization. Then, we pool all the svs corresponding to a given concept and perform the minimization using this newly formed set to learn the per-concept weight vector. We repeat this procedure as many times as the number of predefined shape concepts. We denote this heuristic by *PCMIN-W*.

The two-round protocol is a particular form of relevance feedback and requires user intervention during the querying process. In the first round, the retrieval machine returns a ranked list of shapes using a simple scoring scheme, e.g., the sum of the available raw scores $\varphi_1 = \sum_k s_k$. After the first found, we can invoke the score fusion scheme in two different ways:

- **On-line.** The user marks M shapes among the returned ones, as either *relevant* ($y = 1$) or *non-relevant* ($y = -1$) with respect to his/her query (see Figure 3). In the second round, the retrieval machine returns a refined ranked list

using the scoring function $\varphi_{\mathbf{w}} = \langle \mathbf{w}, \mathbf{s} \rangle$. The weight vector $\mathbf{w}$ is learned *on-line* using the M marked shapes as training instances. In order not to demand too much from the user, M should not be large and is typically limited to a few first shapes. For example, when $M = 8$ and the number of positive M^+ and negative instances M^- are equal ($M^+ = M^- = 4$), the total number of training score difference vectors is just 16. Consequently, on-line learning is computationally feasible.

- **Off-line.** In this variant of the two-round search, all the shapes in the database have their individual weight vectors stored, which have already been learned *off-line* (see Figure 4). The individual per-query weight vectors can be obtained as in the bimodal protocol. At querying time, the user is asked to mark just the first relevant item in the displayed page of the results. The second round evaluates the scoring function $\varphi_{\mathbf{w}} = \langle \mathbf{w}, \mathbf{s} \rangle$ using the weight vector corresponding to the marked shape. Clearly, this protocol does not perform any on-line learning and constitutes a less demanding option than the former in terms of user interaction needed, as the user is asked to mark just one item.

3. Density-Based Shape Description Framework

Although the research on 3D shape descriptors for retrieval started just a decade ago or so, there is a considerable amount of work reported so far. The most up-to-date and complete reviews in this rapidly evolving field are given in [BKS*05, TV04, IJL*05]. The score fusion algorithm described in the previous section can be used with any type of shape descriptors. In the present work, we employ a relatively recent 3D shape description methodology, the density-based framework (DBF) [ASYS07a]. As we expose in this section, DBF produces a rich set of descriptors, which has a good retrieval performance compared to other state-of-the-art methods.

In DBF, a shape descriptor consists of the sampled probability density function (pdf) of a local surface feature evaluated on the surface of the 3D object. The sampling locations of the pdf are called *targets* and the pdf value at each target is estimated using kernel density estimation. The vector of estimated feature pdf values is a density-based shape descriptor. In [ASYS07b], the discriminative power of several multivariate surface features within DBF has been investigated on different databases. Three of these features are particularly interesting as they capture local surface information up to second-order:

- At zero-order, the **R**-feature $(R, \hat{\mathbf{R}})$ parametrizes the coordinates of a surface point at a distance R from the object's center of mass. The unit direction of the ray traced from the center of mass to the point is denoted by $\hat{\mathbf{R}}$.
- At first-order, the **T**-feature $(D, \hat{\mathbf{N}})$ parametrizes the tangent plane at a surface point. Here, D stands for the absolute normal distance of the tangent plane to the center of mass and $\hat{\mathbf{N}}$ is the unit normal at the point.
- At second-order, the **S**-feature (R, A, SI) carries categorical local surface information through the shape index SI [KvD92], which is enriched by the radial distance R and the alignment $A \triangleq \left| \langle \hat{\mathbf{R}}, \hat{\mathbf{N}} \rangle \right|$.

Table 1: *Retrieval Performance of State-of-the-Art Descriptors on PSB Test Set*

	NN (%)	DCG (%)
DBI	66.5	66.3
REXT	60.2	60.1
R⊕**T**⊕**S**	67.4	65.0

Using DBF, these local features are summarized into global shape descriptors that we denote as **R**-, **T**- and **S**-descriptors. A simple way to benefit from different types of shape information carried by these descriptors is to sum their corresponding similarity values s_k, that is, $\varphi_{\mathbf{1}} = \sum_k s_k$. The retrieval performance of this basic score fusion, denoted as **R**⊕**T**⊕**S**, on PSB test set (907 objects in 92 classes) is shown in the third row of Table 1 in terms of discounted cumulative gain (DCG) and nearest neighbor (NN) measures. Note that this fusion is unsupervised and does not involve any statistical learning.

In Table 1, we also display performance figures corresponding to two state-of-the-art descriptors: depth buffer images (DBI) [BKS*05] and radialized extent function (REXT) [Vra03]. DBI is a 2D image-based method, which describes a shape by the low-frequency Fourier coefficients of six depth buffer images captured from orthogonal parallel projections. REXT, on the other hand, relies on a purely 3D idea by describing the shape as a collection of spherical functions giving the maximal distance from center of mass as a function of spherical angle and radius. DBI among 2D methods and REXT among 3D methods are the best performing descriptors on PSB test set in their own methodological categories based on the results reported in [BKS*05] and [FS06] respectively. From Table 1, we see that (i) **R**⊕**T**⊕**S** is significantly better than REXT in terms of both DCG and NN; (ii) its performance is comparable to that of DBI.

Learning-based fusion of two or three scores does not have enough degrees of freedom to boost the retrieval performance significantly. We conjecture that we can reap the benefits of statistical ranking upon employing a larger set of descriptors produced by DBF. To prove our conjecture, we decided to decompose the pdf of a feature into cross-sections. Observe first that all of the **R**-,**T**- and **S**- descriptors are radialized in the sense that they capture the distribution of some *subfeature* at concentric shells with radius r_k (or d_k for the **T**-descriptor). The subfeatures are the radial direction $\hat{\mathbf{R}}$ for the **R**-descriptor, the normal $\hat{\mathbf{N}}$ for the **T**-descriptor and

the (A, SI)-pair for the **S**-descriptor. We refer to these distributions as *cross-sectional* descriptors. For instance, let us take the $N_R \times N_{\hat{\mathbf{R}}} = 8 \times 128 = 1024$-target pdf of the **R**-feature, where $N_R = 8$ is the number of points sampled within the R-domain and $N_{\hat{\mathbf{R}}} = 128$ is the number of points sampled on the unit-sphere. The 1024-point **R**-descriptor is then considered as $N_R = 8$ chunks of $N_{\hat{\mathbf{R}}} = 128$-point cross-sectional descriptors, each of which can be used to evaluate a similarity score s_k between two objects at a given concentric shell, say at a distance r_k from the origin. Of course, these individual scores do not capture the shape similarity to the full extent. However, this decoupling adds more degrees of freedom to the subsequent score fusion stage, where we learn a distinct weight w_k for each of the individual scores s_k by ranking risk minimization. Accordingly, for each of the **R**-, **T**- and **S**-descriptors, we obtain 8 per-chunk similarity scores and work with 24 scores in total.

4. Experiments

We have tested our score fusion algorithm on a modified version of PSB. Originally, PSB training and test sets do not share the same shape classes. Accordingly, we have merged these two sets into a single one, consisting of 1814 models in 161 classes. The number of classes shared by original training and test sets is 21, hence the merged PSB contains $90 + 92 - 21 = 161$ classes. We have then split them into two subsets A and B of sizes 946 and 868, drawing them randomly from the same 161 classes. This reorganization of the PSB database offers us an even more challenging problem since the number of classes is increased from 92 to 161.

4.1. Performance in the Bimodal Search

Recall that the bimodal search protocol assumes the existence of a training set categorized into a fixed set of concepts. Learning is done off-line. In the bimodal experiments, we have taken the PSB Set A as the training set, which we have used to learn per-concept weight vectors. PSB Set B has been reserved for testing purposes. In Table 2, we provide the results of fusing 8 **R**-scores, 8 **T** and 8 **S**-scores, making 24 scores in total. We also display the results of the basic *SUM* rule for reference (i.e., that of **R**⊕**T**⊕**S**). Although, the learning-based score fusion does improve the average DCG performance significantly on the training set, it does not lead to a significant gain in the test set (only 2% using the *AVE-W* heuristic). That is, learning-based score fusion does not work well for certain concepts. This might be due to heuristics-based learning of per-concept weight vectors, but, we think that the following arguments better explain the situation:

- For some concepts, the linear similarity model might not be flexible enough to maintain good classification accuracy in the score difference domain. When instances from queries belonging to a certain concept are pooled together, the discrimination problem in the score difference domain might become more complex than what can be solved using a simple linear decision boundary. However, if the linear similarity model were totally unacceptable, we would not expect a good performance on the training set either. In fact, in only 4 out of 161 concepts in PSB Set A, the *AVE-W* fusion has worsened the performance by not more than 2.3% DCG points with respect to the baseline *SUM* rule. In PSB Set B, on the other hand, 61 concepts (again out of 161) have suffered from an average performance loss of 8.5% DCG points.
- In Table 3, we provide the DCG scores when we use the basic *SUM* rule instead of learning-based score fusion (*AVE-W* or *PCMIN-W*) for negatively affected concepts (i.e., those concepts for which learning-based score fusion has worsened the DCG performance). The right most columns give the number of positively affected concepts. We deduce that the linear similarity model is adequate for the training set and generalizes well on the previously unseen instances of ∼100 concepts in the test set.

Table 2: *DCG (%) Performance of Score Fusion in the Bimodal Protocol*

Rule	PSB Set A	PSB Set B
SUM	61.6±28.1	60.6±28.1
AVE-W	71.8±26.5	62.6±28.4
PCMIN-W	74.9±25.2	62.5±27.7

Table 3: *DCG (%) Performance of Score Fusion in the Bimodal Protocol when the basic* SUM *rule instead of learning-based score fusion has been used for negatively affected concepts*

Rule	PSB Set B	# P.A. Concepts
SUM	60.6±28.1	-
AVE-W	64.0±24.1	106
PCMIN-W	64.4±23.9	100

P.A. Concepts: Positively Affected Concepts

4.2. Performance in the Two-round Search

In the two-round query formulation, the benefits of the proposed score fusion scheme become much more evident. To evaluate the performance in this search protocol, we have reserved the PSB Set A as the database shapes and PSB Set B as the query shapes. The first round results have been obtained by the basic *SUM* rule (i.e., **R**⊕**T**⊕**S**).

In Figure 5, we display the DCG performance of the *online* sub-protocol as a function of the number of *marked* items M from 4 to 32. In this figure, the line at the bottom stands for the DCG of the first round (i.e., the performance of the *SUM* rule, DCG = ∼62%). The line at the top stands for the DCG when all database models are marked as either relevant or non-relevant, serving as an empirical ideal,

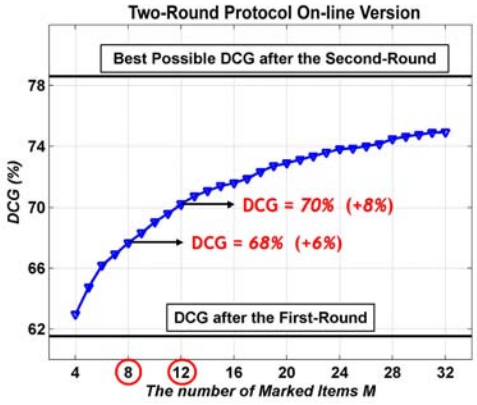

Figure 5: *DCG performance of the two-round search with* on-line *learning as a function of the number of* marked *items M in the first round*

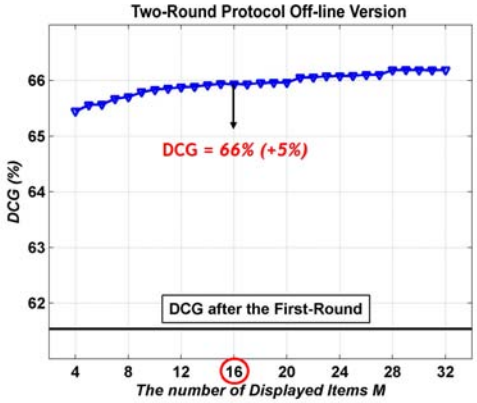

Figure 6: *DCG performance of the two-round search with* off-line *learning as a function of the number of* displayed *items M in the first round*

i.e., the maximum achievable DCG on this data set using the presented score fusion algorithm and the running set of description schemes (DCG = ∼79%). Based on these results, we make the following comments:

- As the number of marked items M increases, we observe a steep increase in the DCG performance, compatible with theoretical fast rates of convergence proven in [CLV07]. The DCG profile converges smoothly to the empirical ideal as the user marks more and more items in the first round.
- To give certain performance figures, for $M = 8$, DCG obtained after fusing the scores becomes ∼68%, giving a 6% improvement compared to the baseline. The 70% DCG barrier is reached after $M = 12$ marked items.

In Figure 6, we display the DCG performance of the *off-line* sub-protocol as a function of the number of *displayed* items M again 4 to 32. We emphasize that, in this mode, M refers to the number of *displayed* items and the user interaction needed is limited to mark just one shape, the first relevant one after the first round. Accordingly, here, M is not related to the convergence of the algorithm. Increasing M does not cost anything in terms of user interaction. After this clarification, we have the following comments:

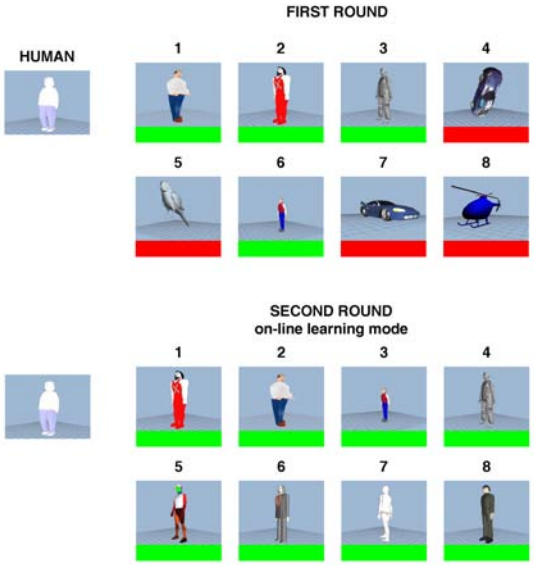

Figure 7: *Two-round search with* on-line *learning on a* human *query*

- At $M = 1$, score fusion boosts the retrieval performance by ∼4% and the DCG profile keeps a slow but constant increase as the number of displayed items M in the first round is increased.
- In a typical retrieval scenario, displaying $M = 32$ items has no cost. These results tell us that we can obtain DCG improvements by ∼5% with respect to the baseline. Noting that the performances of top 3D shape descriptors differ only by a couple of percentage points, this 5% gain can be considered as significant and comes virtually at no cost at the querying process. The only bottleneck is the off-line processing of the database shapes to learn the weight vectors, which may eventually be used in the second round.

With on-line score fusion, we can obtain significant improvements as the user is asked to mark more and more items. In special applications where the user voluntarily marks the demanded number of items, the on-line scheme is preferable. The off-line scheme, on the other hand, comes at no cost at query time and still yields satisfactory improvements. Sample two-round searches using these two variants are shown in Figures 7 and 8.

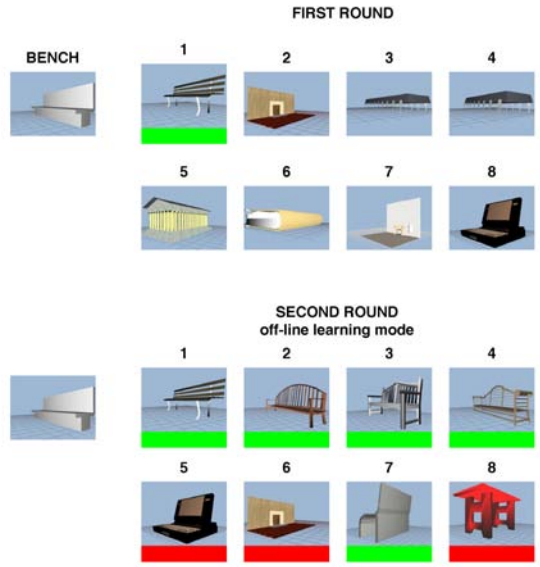

Figure 8: *Two-round search with* off-line *learning on a* bench *query*

5. Conclusion

Several studies in the 3D domain pointed out the non-existence of a single "miracle" 3D shape descriptor to provide adequate discrimination for retrieval [TV04, SMKF04]. In this work, motivated by the fact that different descriptors may work well on different sets of shape classes, we have addressed a relatively less studied problem in 3D object retrieval: combining multiple shape similarities to boost the retrieval performance. Our linear score fusion algorithm based on ranking risk minimization proved to be effective on ontology-driven bimodal query formulations and much more on the two-round protocol which is a particular instance of relevance feedback.

An immediate perspective for further research is to extend this general score fusion scheme to other type of shape descriptors, notably to 2D image-based ones [BKS*05]. Furthermore, we may obtain performance improvements using kernel methods [HTF01] to learn a non-linear scoring function. Direct minimization of per-concept risks, optimization of DCG-based criteria and kernelization of the score fusion algorithm will constitute our future research directions in this field.

References

[ASYS07a] AKGÜL C. B., SANKUR B., YEMEZ Y., SCHMITT F.: Density-based 3D shape descriptors. *EURASIP Journal on Advances in Signal Processing 2007* (2007), Article ID 32503, 16 pages. doi:10.1155/2007/32503.

[ASYS07b] AKGÜL C. B., SANKUR B., YEMEZ Y., SCHMITT F.: Multivariate density-based 3D shape descriptors. In *Proc. of the Shape Modeling International (SMI'07)* (Lyon, France, June 2007).

[BKS*05] BUSTOS B., KEIM D. A., SAUPE D., SCHRECK T., VRANIC D. V.: Feature-based similarity search in 3D object databases. *ACM Comput. Surv. 37*, 4 (2005), 345–387.

[CL01] CHANG C.-C., LIN C.-J.: *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[CLV07] CLÉMENÇON S., LUGOSI G., VAYATIS N.: Ranking and empirical risk minimization of U-statistics. *The Annals of Statistics to appear* (2007).

[FISS03] FREUND Y., IYER R., SCHAPIRE R. E., SINGER Y.: An efficient boosting algorithm for combining preferences. *J. Mach. Learn. Res. 4* (2003), 933–969.

[FS06] FUNKHOUSER T., SHILANE P.: Partial matching of 3D shapes with priority-driven search. In *Symposium on Geometry Processing* (June 2006).

[HGO00] HERBRICH R., GRAEPEL T., OBERMAYER K.: *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.

[HTF01] HASTIE T., TIBSHIRANI R., FRIEDMAN J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, NY, 2001.

[IJL*05] IYER N., JAYANTI S., LOU K., KALYANARAMAN Y., RAMANI K.: Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design 37*, 5 (April 2005), 509–530.

[Joa02] JOACHIMS T.: Optimizing search engines using clickthrough data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)* (2002), pp. 133–142.

[KvD92] KOENDERINK J. J., VAN DOORN A. J.: Surface shape and curvature scales. *Image Vision Comput. 10*, 8 (1992), 557–565.

[SMKF04] SHILANE P., MIN P., KAZHDAN M., FUNKHOUSER T.: The Princeton shape benchmark. In *Proc. of the Shape Modeling International 2004 (SMI'04)* (Genoa, Italy, 2004), pp. 167–178.

[TV04] TANGELDER J. W. H., VELTKAMP R. C.: A survey of content based 3D shape retrieval methods. In *Proc. of the Shape Modeling International 2004 (SMI'04)* (Genoa, Italy, 2004), pp. 145–156.

[Vra03] VRANIĆ D. V.: An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)* (Barcelona, Spain, Spetember 2003), pp. 757–760.