3D Shape Classification and Retrieval Using Heterogenous Features and Supervised Learning

Hamid Laga
Tokyo Institute of Technology
Japan

1. Introduction

Content-based 3D model retrieval (CB3DMR) aims at augmenting the text-based search with the ability to search 3D data collections by using examples, sketches, as well as geometric and structural features. In recent years there is an increasing demand on such tools as 3D graphics technology is becoming widely accessible and a large amount of 3D contents is being created and shared.

Usually an algorithm for 3D model classification and retrieval requires: (1) an efficient representation of the 3D data that is suited for search, and (2) a good similarity function in order to measure distances between entities in the feature space. The first step involves feature extraction, feature selection strategy to keep only the most relevant features, and a method for encoding the features as real-valued vectors called *shape descriptors*. Shape descriptors provide a numerical representation of the salient features of the data. They should be an abstraction of the semantics of the shape and shape category. Many descriptors have been proposed for content-based 3D model classification and retrieval but none of them has achieved high-level performance on all shape classes. For instance:

- Global geometric features, which are easy to compute and compare, are poor in terms of discrimination power since they are unable to capture the intra-class shape variability. Alternatively, local features, such as spin images (Johnson, 1997) and shape contexts (M.Kortgen et al., 2003) are more effective for intraclass retrieval. However, their extraction and comparison are expensive in terms of computation and storage requirements. A key observation is that many of these features are redundant and only a small subset of them, called representative feature set, is really discriminative. Thus, there is a need for selecting automatically the optimal set of features. The selected set should be specific to each class of shapes, and adapted to different types of user queries and data classifications.
- Geometry-based features, such as Light Fields (LFD) (Chen et al., 2003) and spherical harmonic (Funkhouser et al., 2003) descriptors, represent shapes with their global geometric characteristics. On the other hand, graph-based descriptors, such as Reeb-graphs and skeleton representations (Hilaga et al., 2001; T.Tung & F.Schmitt, 2005), encode the structural characteristics of the shape, and therefore are more suitable for indexing articulated shapes. Consequently, there is a need for combining *heterogeneous*

features in order to achieve best performance. By heterogeneous we mean features of different types and scales.

From the machine learning point-of-view, efficient selection and combination of heterogeneous features for classification and retrieval poses many challenges. The first issue is how to choose among a large set of features, a subset that allows to achieve high-level performance. The second issue is the feature normalization problem. Heterogenous features are often of different scales. Therefore, incorporating them directly into the similarity function will result in low retrieval performance as higher scale features will influence more the similarity. This issue is related to the feature weighting strategy.

The goal of this chapter is to develop an effective 3D shape classification and retrieval method that uses discriminative shape features automatically selected from a large set of heterogeneous features. The construction of the representative set can be regarded as a machine learning task. Particularly, supervised learning allows to capture the high-level semantic concepts of the data using low-level geometric features. Our key idea is to use a large set of local and global features, eventually not orthogonal, then use a supervised learning algorithm to select only the most efficient ones. We experimented with AdaBoost which provides a mean for feature selection and classifier combination. Boosting, like many machine-learning methods, is entirely data-driven in the sense that the classifier it generates is derived exclusively from the evidence present in the training data itself (Schapire, 2003). Moreover, allowing redundancy and overlapping in the feature set has been proven to be very efficient in recognition and classifications tasks than orthogonal features (Tieu & Viola, 2004). Specifically, we make the following contributions:

- An algorithm for learning the discriminative features of a class of shapes from a training set. The algorithm allows also to quantify the discrimination ability of a shape feature with respect to the underlying classification. Features of high discrimination ability of each class of shapes will be used for processing unseen objects (classification of the query, and retrieving the most similar shapes to the query).
- A method for matching shapes using only the most relevant features to each class of shapes. This approach can be used with either a flat or a hierarchical classification of the data resulting in a multi-scale organization of the feature space.
- The ability to use heterogeneous features for classification is a major deviation from previous work.

The remainder of this paper is organized as follows: the next section reviews the related work. Section 2.3 gives and overview of the proposed framework and outlines the main contributions. In Section 3 we describe the type of 3D shape descriptors we will use in this chapter. Section 4 details the developed algorithm for feature selection and combination in the case of a binary classification (Section 4.1), and its generalization to a multi-class problem (Section 4.2). In Section 5 we detail the query processing method for classification and retrieval. Experimental results and evaluations are given in Section 6. Section 7 concludes the paper and outlines the major issues for future work.

2. Related work

3D shape analysis, classification and retrieval received significant attention in recent years. In the following we review the most relevant techniques to our work. For more details, we refer the reader to the recent surveys of the topic (Lew et al., 2006; Tangelder & Veltkamp, 2004; Iyer et al., 2005).

2.1 Descriptors for 3D model retrieval

For efficient comparison and similarity estimation, 3D models can be represented with a set of meaningful descriptors that encode the salient geometric and topological characteristics of their shapes. The database objects are then ranked according to their distance to the descriptors of the query model. These descriptors are either global, local, or structural. Structural descriptors such as Reeb graphes (Hilaga et al., 2001; T.Tung & F.Schmitt, 2005) aim at encoding the topological structure of the shape. They can be used for global matching as well as partial matching (Biasotti et al., 2006).

Global descriptors describe an entire 3D shape with one single feature vector. In this family, the Light Fields (LFD) (Chen et al., 2003) are reported to be the most effective (Shilane et al., 2004). (Funkhouser et al., 2003) map a 3D shape to unit spheres and use spherical harmonics (SH) to analyze the shape function. Spherical harmonics can achieve rotation invariance by taking only the power spectrum of the harmonic representation, and therefore, discarding the rotation dependent information (Kazhdan et al., 2003). (Novotni & Klein, 2003) use 3D Zernike moments (ZD) as a natural extension of SH. (Laga et al., 2006) introduced flat octahedron parameterization and spherical wavelet descriptors to eliminate the singularities that appear in the two poles when using latitude-longitude parameterization, and therefore, achieve a fully rotation invariant description of the 3D shapes. Recently, (Reuter et al., 2006) introduced the notion of shape DNA as fingerprints for shape matching. The fingerprints are computed from the spectra of the Laplace-Beltrami operators. They are invariant under similarity transformations and are very efficient in matching 2D and 3D manifold shapes. However, it is not clear how they can be extended to polygon soup models.

Global descriptors are very compact, easy to compute, and efficient for broad classification of 3D shapes. However, they cannot capture the variability of the shapes and their subtle details necessary for intra-class retrieval. Local feature-based methods can overcome these limitations by computing a large set of features at different scales and locations on the 3D model. Spin images (Johnson, 1997) and shape contexts (M.Kortgen et al., 2003) have been used for shape retrieval as well as for matching and registering 3D scans. Local features are very efficient to discriminate objects within the same class. However, similarity estimation requires combinatorial comparison, making them not suitable for realtime applications such as retrieval.

2.2 Feature selection and relevance feedback

3D model retrieval by matching low level features does not fully reflect the semantics of the data. For instance, most of the previous techniques cannot distinguish between a flying bird and a commercial airplane. This is commonly known as the *semantic gap*. Recent progress in pattern recognition suggested the use of supervised learning to narrow the semantic gap. This allows the automatic selection of salient features of a single 3D model within a class of shapes, and also the use of the results of classification to improve the performance of retrieval algorithms.

The basic learning approach is the Nearest Neighbor classifier. It has been used for the classification of 3D protein databases (Ankerst et al., 1999), and also 3D engineering parts (Ip et al., 2003).

Hou et al. (2005) introduced a semi-supervised semantic clustering method based on Support Vector Machines (SVM) to organize 3D models semantically. The query model is first labeled with some semantic concepts such that it can be assigned to a single cluster.

Then the search is conducted only in the corresponding cluster. Supervised learning and ground-truth data are used to learn the patterns of each semantic cluster off-line. Later, (Hou & Ramani, 2006) combine both semantic concepts and visual content in a unified framework using a probability-based classifier. They use a linear combination of several classifiers, one per descriptor. The individual classifiers are trained in a supervised manner, and output an estimate of the probability of data being classified to a specific class. The output of the training stage is used to estimate the optimal weights of the combination model. The retrieval is performed in two stages; first they begin by estimating the conditional probability of each class of shapes given the query. Then they perform shape search inside each candidate class. The new similarity measure is a unified distance that integrates the probability estimation from the classifiers, a combination of classifiers learned off-line, and a shape similarity distance. This is the closest work to ours. In this approach features and type of classifiers are set manually. In our case, we aim at selecting automatically the most salient features.

(Shilane & Funkhouser, 2006) investigated on how to select local descriptors from a query shape that are most distinctive and therefore most relevant for retrieval. Their approach uses supervised learning to predict the retrieval performance of each feature, and select only a small set of the most effective ones to be used during the retrieval. (Funkhouser & Shilane, 2006) introduced priority-driven search for partial matching of 3D shapes. The algorithm produces a ranked list of c-best target objects sorted by how well any subset of k features on the query matches features on the target object. As reported by the authors, the timing results is dominated by the number of features for each target object and the number of scales for each feature. The algorithm we propose can deal with large set of features while maintaining the processing time at interactive rates.

The approach most similar to our own is that of (Tieu and Viola, 2004) where they applied the AdaBoost algorithm (Schapire, 2003) to online learning of the similarity of a given query to the target objects in image retrieval. It has been recently extended to learn the intrinsic features for boosting 3D face recognition (Xu et al., 2006). AdaBoost enables the use of a very large set of features while keeping the processing time at the run-time very attractive. We improve over this approach in two important ways. First we investigate the application of AdaBoost to the general problem of 3D model retrieval. Second, we learn, off-line, the optimal salient and discriminative set of features for each class of shapes with respect to objects in the entire database. These improvements allow our 3D model retrieval algorithm to achieve high-level performance in terms of retrieval efficiency and computation time.

2.3 Overview

Figure 1 gives an overview of our approach. At the training stage a strong classifier is learned using AdaBoost. The classifier returns the likelihood that a given 3D model O belongs to a class of shapes C. First a large set of features are extracted from every model in the database. Then a set of binary classifiers are trained using AdaBoost. Each binary classifier learns one class of shapes and its optimal set of salient features. Finally, the binary classifiers are combined into one multi-class classifier. In our implementation we experimented with the Light Field Descriptors (LFD) (Chen et al., 2003) (100 descriptors per-shape), the Gaussian Euclidean Distance transform (GEDT) (Shilane et al., 2004) (32

descriptors per-shape, each descriptor is computed on a concentric sphere of radius r, $0 \leq r \leq 1$), and a combination of the two descriptors which will be referred by LFD-GEDT.

At the run-time, given a query model Q, a ranked list of k–best matches is produced in a two-stage process that involves classification and search. First, a large set of features are computed from the query model Q, in the same manner as for the database models. Then in the classification stage, a set of highly relevant classes to Q is found. Each binary classifier $\mathcal{C}_i$ decides wether the class C_i is relevant to the query Q or not. In the retrieval stage, the similarity between Q and the models in every relevant class C_i is estimated and a ranked list of the best matches is returned.
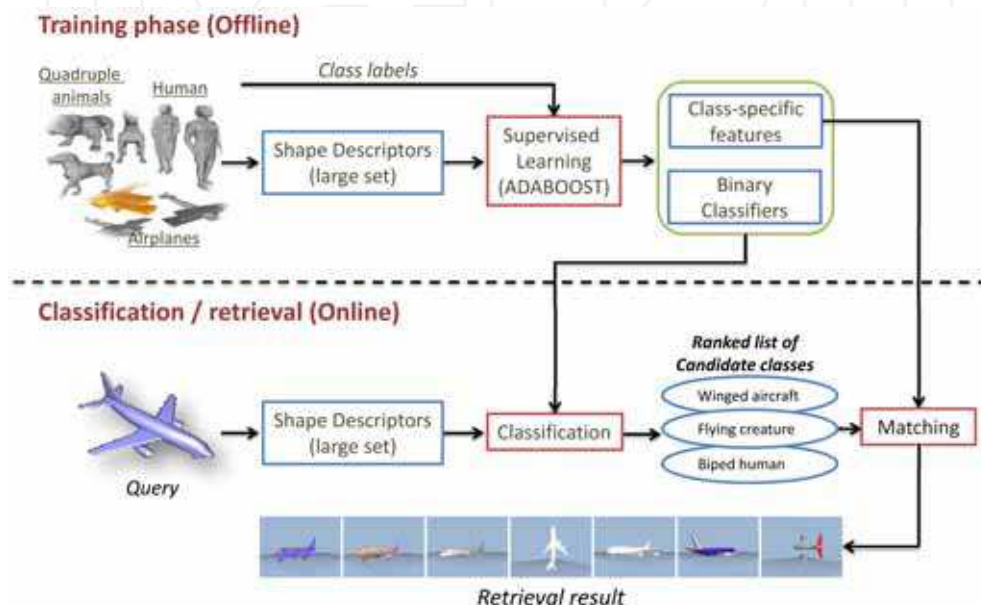

Fig. 1. Overview of AdaBoost-based 3D model classification and retrieval. At the training stage a strong classifier is learned using AdaBoost. The strong classifier is based on a combination of the most discriminative features of the shape. At the run-time, a query is first classified into a set of candidate classes, then the search for the best matches is performed inside the candidate classes.

The key step in this process is the way we predict the saliency of each feature with respect to a class of shapes in the training set. More formally, the saliency of a feature $\vec{v}$ with respect to a class of shapes C is the ability of this feature to discriminate the shapes of class C from the shapes of other classes in the database. Mathematically, given the binary classifier $\mathcal{C}_{\vec{v}}$ trained with the feature $\vec{v}$, the saliency of $\vec{v}$ is directly related to the overall classification error of $\mathcal{C}_{\vec{v}}$ on the data set. However, none of the existing classifiers that are based on a single feature can achieve zero classification error. Therefore none of the features is sufficiently salient. AdaBoost provides a way for combining weak classifiers and shape features, eventually of different types and saliency degrees, into a single strong classifier with high classification performance. There are several advantages of this approach:

Although a large set of features is extracted both at the training and online stages, only a small subset of the features (between 10 to 50) is used during the similarity estimation. This allows retrieval at interactive rates.

- The algorithm selects automatically the representative set of features for each class of shapes, and provides a mean for automatic combination of the selected features. This has potential applications in shape classification and recognition.
- The algorithm provides an automatic way to truncate the list of the $k-$best matches, i.e, it provides a mean for saying wether the database contains models which are similar to a given query or not.
- This approach allows to perform both inter-class and intra-class retrieval. AdaBoostbased classifier allows to find the relevant classes to the query. Then, in a second step, the search can be performed inside each relevant class using, eventually, different types of descriptors.

For feature extraction, we use the Light Field descriptors (LFD) (Chen et al., 2003) and Gaussian Euclidean Distance Transform (GEDT) (Shilane et al., 2004). However, a further investigation is required to test the efficiency of other descriptors when boosted, which is beyond the scope of this paper.

3. 3D shape descriptors

The process starts by computing a large set of features for each model in the training set, which is the content of the database to search. There are many requirements that the features should fulfill: (1) compactness, (2) computation speed, and (3) the ability to discriminate between dissimilar shapes. However, in real applications it is hard to fulfill these requirements when the goal is to achieve high retrieval accuracy. In fact, compact features, which are easy to compute, are not discriminative enough to be used for high accuracy retrieval. We propose to extract a large set of features following the same idea as in (Tieu & Viola, 2004).

There are many shape descriptors that can be computed from a 3D model. A large set of Spherical harmonics (Funkhouser & Shilane, 2006) and spherical waveletbased descriptors (Laga et al., 2006) can be computed by moving the center of the sphere across different locations on the shape's surface or on a 3D grid. However, in the literature, it has been proven that view-based descriptors outperform significantly the spherical descriptors. In our implementation we considered two shape descriptors evaluated in the Princeton Shape Benchmark: the Light Fields Descriptor (LFD), and the Gaussian Euclidean Distance Transform Descriptor (GEDT). For the completeness purpose we give a brief overview of these descriptors but the reader can find further details in the original paper (Shilane et al., 2004):

- **Light Field Descriptor (LFD) (Chen et al., 2003):** a view-based descriptor computed from 100 images rendered from cameras positioned on the vertices of a regular dodecahedron. Each image is encoded with 35 Zernike moments, and 10 Fourier coefficients. In this paper we use our own implementation.
- **Gaussian Euclidean Distance Transform (GEDT) (Shilane et al., 2004):** a 3D function whose values at each point is given by composition of a Gaussian with the Euclidean Distance Transform of the surface. It is computed on 64×64×64 axial grid, translated such as the origin is at the point (32, 32, 32), scaled by a factor of 32, and then

represented by 32 spherical descriptors representing the intersection of the voxel grid with concentric spherical shells. Values within each shell were scaled by the square-root of the corresponding area and represented by their spherical harmonic coefficients up to order 16.

To evaluate the performance of the feature selection algorithm we will consider also a combination of the two descriptors, herein after referred by LFD-GEDT. Notice that these two descriptors are encoding different properties of the shape and may have different scales. Also, the set of features contains many redundancies: in the case of LFD for example, two symmetric view points will have the same 2D projection, and close points in the Euclidean sense will have their associated LFDs very similar. On one hand, this will increase significantly the storage and computation time required for matching and retrieval. However, on the other hand, it will guarantee that the feature set can capture the shape variability. Therefore, we rely on the learning stage to select the salient ones that achieve best classification and retrieval performance.

4. Supervised classification

The first task in our approach is to build a classifier $\mathcal{C}$ that decides wether a given 3D model O belongs to a class of shapes C or not. The challenge is to define a feature space such that 3D shapes belonging to the same class are mapped into points close to each other in the new feature space. Clusters in the new space correspond to classes of 3D models. There are many feature spaces that have been proposed in the literature, but it has been proven that none of them achieved best performance on all classes of shapes. We propose to follow a machine learning approach where each classifier is obtained by the mean of training data.

4.1 Boosting the binary classification

A brute force approach for comparing a large set of features is computationally very expensive, and in the best case, it requires $M \times d \times N$ comparisons, where M is the number of feature vectors used to describe each $3D$ model, d is the dimension of the feature space, and N is the number of models in the database.

Previous works consider this problem from the dimensionality reduction point of view. (Ohbuchi et al., 2007) provide an overview and performance evaluation of six linear and non-linear dimensionality reduction techniques in the context of 3D model retrieval. They demonstrated that non-linear techniques improve significantly the retrieval performance. There have been also a lot of research in classifiers that have a good generalization performance by maximizing the margin. Speed is the main advantage of boosting over other classification algorithms such as Support Vector Machines (SVM) (Hou et al., 2005), and non-linear dimensionality reduction techniques (Ohbuchi et al., 2007; Ohbuchi & Kobayashi, 2006). It can be also used as a feature selection algorithm, and provides a good theoretical quantification of the upper bound of the error rate, therefore a good generalization performance.

We use AdaBoost version of boosting. Every weak classifier is based on a single feature of a 3D shape (recall that we have computed a large set of features for each 3D model). The final strong classifier, a weighted sum of weak classifiers, is based on the most discriminating features weighted by their discriminant power. The algorithm is summarized in Algorithm 1.

Algorithm 1: AdaBoost algorithm for binary classification

Input: Training set $S_C = \{(V_i, y_i), i = 1 \ldots N\}$, where $V_i = \{\vec{v}_1, \ldots, \vec{v}_K\}$ is a
large set of K features computed from the 3D object O_i, and
$y_i \in \{+1, -1\}$ is the targeted classification of O_i.

Output: The decision function f_C, such that, $f_C(O) > 0$ is $O \in C$, and
$f_C(O) < 0$ if $O \notin C$.

1. Initialize the sample weights: $w_0 = \{w_i^0, i = 1, \ldots, N\}$:

$$w_i^0 = \begin{cases} \frac{1}{N^+}, \text{if } O_i \text{ is a positive example} \\ \frac{1}{N^-}, \text{otherwise}. \end{cases}$$

where N^+ and N^- are, respectively, the number of positive and negative
examples.

2. **for** $t=1, \ldots, T$ **do**

    (a) Train one weak classifier $h_k, k = 1 \ldots K$ for each
    feature vector v_k, using the training set S_C
    sampled according to the probability distribution
    $w_t = \{w_i^t, i = 1, \ldots, N\}$.

    (b) Choose the hypothesis h_t with the lowest classification error ϵ_t.

    (c) Update the sample weights: $w_i^{t+1} = \dfrac{1}{Z_t} w_i^t e^{-\alpha_t h_t(O_i) \cdot y_i}$

    where $h_t(O_i) \in \{+1, -1\}$ wether O_i is correctly or incorrectly
    classified by the weak hypothesis h_t,
    $\alpha_t = 0.5 \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$, and Z_t is a normalizing constant
    so that w_{t+1} is a distribution.

end

3. Final classifier: $f_C(O) = \sum_{t=1}^{T} \alpha_t h_t(O)$

The sample weights w_i^t, $i = 1, \ldots, N$, $t = 1, \ldots, T$ are very important; at step t, the weights of the samples with high classification error at step $t - 1$ is increased, while the weights of samples with smaller classification error is decreased (Algorithm 1). This will let the classifier at step t focus on difficult samples which have not been correctly classified in the previous step. The output of the strong classifier can be interpreted as the posterior probability of a class C given the shape O and it is given by:

$$P(C|O) = \frac{e^{f_C(O)}}{e^{f_C(O)} + e^{-f_C(O)}}. \tag{1}$$

The AdaBoost algorithm requires two parameters to tune: the type of weak classifier, and the maximum number of iterations. The weak classifier is required to achieve better classification than random. We experimented with the decision stumps and Least Mean Squares (LMS) classifier for their simplicity. The parameter T can be set such that $E[f_C]$, the upper bound of the classification error on the training data of the strong classifier f_C, is less than a threshold θ. In

our experiments we found that a value of T between 20 and 50 is sufficient to achieve an upper bound of the classification error on the training set less than 1.0%.

Building the training set

We use as positive and negative examples for our training set the relevant and nonrelevant models provided in the Princeton Shape Benchmark (PSB) classification. For example, to build a strong classifier that learns the decision boundary between the *biped human* objects and *non-biped human* objects, the positive examples are set to all models that belong to the class *biped human*, while the negative examples are the remaining models in the database. The PSB is provided with a train and test classifications. We use the train classification to train our classification and the test classification to assess the performance of the classification and retrieval.

4.2 Generalization to multiple classes

Two straightforward extension schemes are the one-vs-all classifier and the pairwise classifier (Hao & Luo, 2006). The pairwise classifier uses $L(L-1)/2$ binary classifiers where L is the number of classes in the training set, to separate every two classes. A voting scheme at the end is used to determine the correct classification. With the one-vs-all classifier, L binary classifiers are trained, each of which is able to distinguish one class from all the others. The pairwise classifier has a smaller area of confusion in the feature space compared to the one-vs-all. However, the number of the required binary classifiers increases quadratically with the number of classes in the database, while the one-vs-all increases linearly.

In our implementation we used a one-vs-all classifier for its simplicity. The output of the training stage is a set of L binary classifiers, where L is the number of classes in the database. Given a query model Q each binary classifier will return a vote for a certain class. We use the positive votes to construct the set of candidate classes to which the query Q may belong. Notice that when a new 3D model or a new class of models are added to the database, only the classifier that corresponds to the model's class that needs training.

It is important to outline that the algorithm is data-driven that is different classifiers are obtained when given a different classification of the data. This allows to capture the semantics of the data. Furthermore, existing 3D model collections are often provided with multiple classifications. We plan in the future to extend the framework to handle hierarchical classifications of the data.

Algorithm 2: One-vs-all extension of binary AdaBoost for multi-class problem.

Input: Training set $S_{C_l} = \{(V_i^l, y_i^l), i = 1 \ldots N\}, l = 1, \ldots, L$, is the class index and $V_i^l = \{\vec{v}_1^l, \ldots, \vec{v}_K^l\}$ is a large set of K features computed from the 3D object O_i, and $y_i^l \in \{+1, -1\}$ is the targeted classification of O_i.

Output: L binary decision functions f_{C_l}, such that, $f_{C_l}(O) > 0$ is C_l is a candidate class for the 3D model O, and $f_{C_l}(O) < 0$ otherwise.

for $l=1, \ldots, L$ **do**

  Train one strong binary classifier $\mathcal{C}_l$, using Algorithm 1.

$$f_{C_l}(O) > 0 \text{ if } O \in C_l, \text{ and } f_{C_l}(O) < 0 \text{ otherwise}$$

end

Collect the binary classifiers $\mathcal{C} = \{\mathcal{C}_l, l = 1, \ldots, L\}$.

4.3 Interpretation of the selected features

Boosting algorithm can be used as a feature selection and combination technique. Each iteration learns a new weak classifier that is based on the most discriminative feature according to the probability distribution of the training data. In the case of LFD, the selected feature is the descriptor of a 2D projection of the 3D model. Therefore, by adopting a Boosting approach we provide a tool for best view selection and view ordering based on their ability to discriminate the shapes of a certain class from the other classes in the database. Here we assume that the quality of a view is quantified as its discrimination ability, i.e, the ability of the 2D view to discriminate the shape from other shapes that belong to different classes.

The interpretation of the weak classifier may differ according to the type of descriptor used for training. In the case of the GEDT, which computes the restriction of the shape to concentric sphere, the selected feature can be seen as the radius of the concentric sphere on which the most important features of the class lie. Furthermore, the weight of each weak classifier can be considered as a measure of the saliency of the selected feature. Recall also that AdaBoost is a stochastic approach. Therefore, different runs of the algorithm on the same data will generate different sets of selected features. This is the case when the problem has many solutions (local optima). At each run it finds a different solution but with similar performance.

Figure 2 shows the top-best views selected with our algorithm. We can see that the important features of each class of shapes are visible from the selected views. This shows first that the selected views are consistent across all models of a same class, and the selected views are visually plausible. Hence, boosting captures some high semantic features of the data set. Best view selection has many applications in Computer Graphics and also online browsing of digital media contents. The framework we proposed provides an easy method to achieve this. We plan in the future to evaluate the quality of the selected views compared to other algorithms (Lee et al., 2005; Yamauchi et al., 2006).

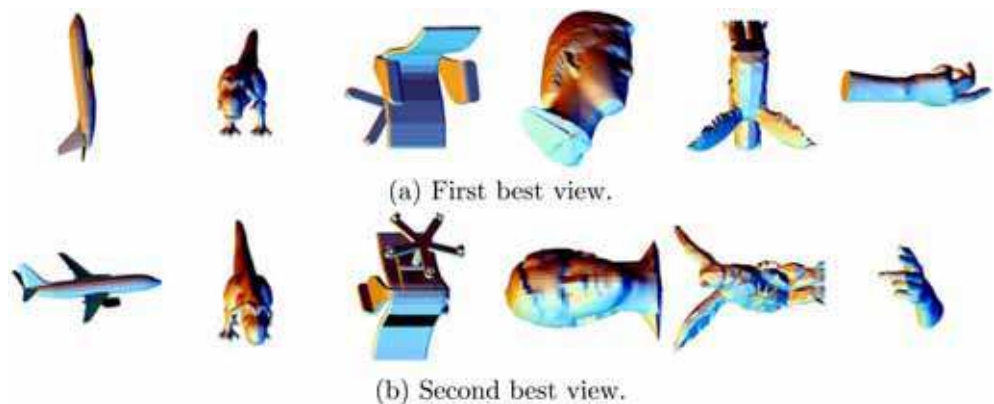(a) First best view.

(b) Second best view.

Fig. 2. Boosted LFD descriptor allows for automatic best-view selection. The first and second rows show respectively the first and second best views of objects belonging to different classes of shapes. Automatic best view selection can be used for visual browsing of large collections of 3D models.

4.4 Combining heterogeneous features

One important property of the developed classification algorithm is its ability to combine heterogeneous features in a straightforward manner; at each step of the training process, a set of weak classifiers are trained on the features, one-per basic feature, and the one with minimum training error is picked. Therefore, the features are considered independently. Although this may ignore possible correlations between basic features, it allows however to handle features of different types. We use this property to combine heterogeneous features for efficient classification.

5. Query processing

At the run time, the user specifies a query Q and seeks either to classify it into one of the shape classes (classification), or retrieve models in the database that are most similar to the query (retrieval).

To classify the query Q, we compute a set of M feature vectors (LFD and GEDT descriptors in our case) in the same manner as in the training stage (Section ??). Then we let each binary classifier $\mathcal{C}_l$ vote for a the class C_l, $l = 1, \ldots, L$. The candidate classes are determined by the classifiers that have positive response to the query Q. We build the candidate classes set by collecting the indices of classes whose classifiers gave positive response, and we order them in descending order of the class posterior probabilities given in Equation 1.

We perform the retrieval in two steps combining classification and search: first we find the candidate classes C_i to which the query Q may belong. Then, we run a search operation inside each candidate class by computing the similarity between the query Q and every model in the candidate class C_i. The 3D models of the candidate classes are sorted according to their similarity to the query model. We return one ranked list per class. The ranked lists are merged to form the k-best matches to the query. Here we use only the salient features of the class C_i, and the matching is performed only on a subset of the entire database. This reduces significantly the computation time.

Search inside classes requires the use of a distance function which measures the distance between the descriptor of the query and the descriptors of the class's shapes. In our implementation we used the Euclidean distance when working with a single descriptor type, i.e., LFD or GEDT. When using heterogeneous features however(ex. LFD-GEDT), the descriptor with larger scale will have higher impact on the Euclidean distance. To overcome this limitation we modify slightly the distance measure as follows; first we compute the Euclidean distance between the query model and the candidate models using each descriptor independently. The final distance is then taken as the minimum over the computed distances.

Examples of retrieval results are shown in Figures 4 and 5 with queries that are not part of the database. In these examples the query models (first column) do not belong to the database and have not been used during the training phase.

6. Results

To evaluate the performance of the proposed framework for 3D model classification and retrieval, we we use the Princeton Shape Benchmark (PSB) Shilane et al. (2004) as a ground truth, and the Shape Retrieval Evaluation Contest (SHREC2006) (Veltkamp et al., 2006) query set and performance evaluation tools. The Princeton Shape Benchmark contains 1814

polygon soup models, divided into the training set (907 models) and the test set (907 models). Every set contains four classification levels; the base train classification contains 129 classes while the coarsest classification (coarse3) contains two classes: man-made vs. natural objects.

6.1 Classification performance

Figure 3 summarizes the classification performance of the developed AdaBoost classifier. In this figure, the average classification performance is the ratio between the number of correctly classified models of a class C to the total number of models in the class. We see that, for the coarse3 classification (Figure 3-(d)), which contains only two classes with very high shape variability within each class, the classification performance is at 65.3% for natural shape and 73% for man-made models. This clearly proves that the training procedure captures efficiently the semantic concepts of the shape classes and generalizes relatively well to unseen samples.

(a) Base classification

(b) Coarse1 classification

(c) Coarse 2 classification.

(d) Coarse 3 classification
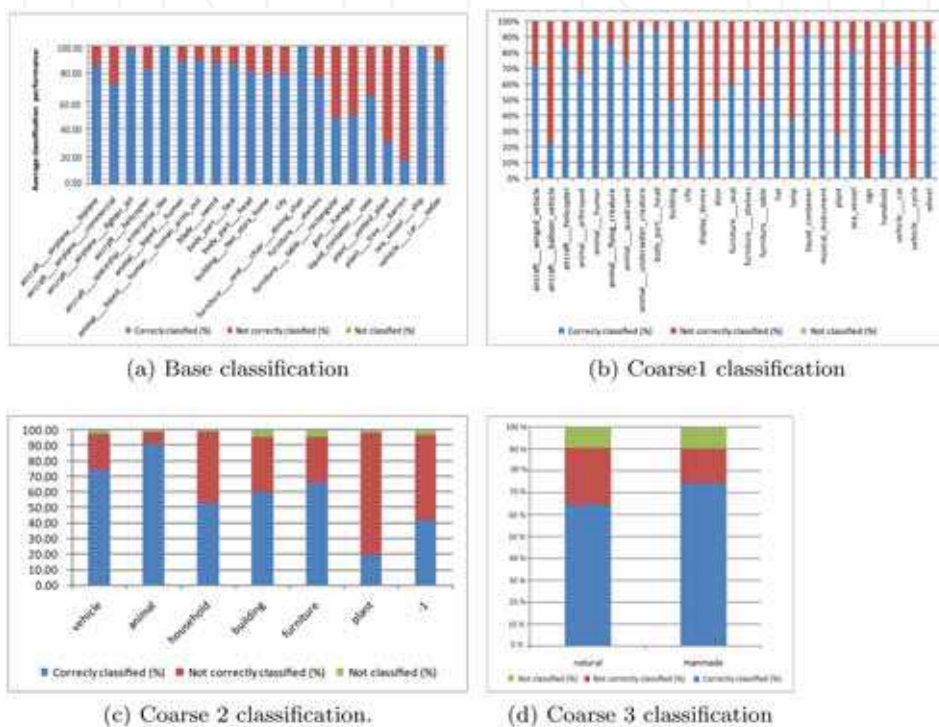
Fig. 3. Average classification performance of the Boosted-LFD for each class of shapes in the test set of the Princeton Shape Benchmark. Class labeled by (-1) contains models that cannot be classified to any of the other classes.

The performance on the other classification levels: base, coarse1 and coarse2 are shown in Figure 3-(a),(b) and (c). In this experiments we show only the classification results on the classes of the test set that exist in the training set. On the base classification (Figure 3-(a)), we can see that the classifiers achieve 100% classification performance on *space ship entreprise*

like, dining chair and *sea vessel*. The worst performance is on the *plant tree* models. This is probably because of the high shape variability within the class , which cannot be captured by the LFD descriptors.

6.2 Retrieval performance

To evaluate the retrieval performance we train our classifier with the entire base classification (train and test sets) of the PSB. This classification contains 160 shape categories with varying number of samples in each class. For testing, we use the 30 queries of the SHREC2006. Each query contains a set of highly relevant and relevant models in the database. Recall that these queries do not belong the database, and therefore, have not been used during the training of the AdaBoost classifiers. We measure the retrieval performance using the SHREC2006 tools and compare to the other descriptors used in contest.

Tables 1, 2, and 3 summarize the performance of our descriptors on the mean average precision, mean first tier, mean second tier, dynamic average recall, mean normalized cumulative gain (MNCG), and the mean normalized discounted cumulative gain (MNDCG) measures. We tested the GEDT and LFD descriptors without boosting, the Boosted-GEDT and Boosted-LFD (i.e., the GEDT and LFD descriptors after boosting), and combination of LFD and GEDT denoted by Boosted-LFD-GEDT.

We can see first that the boosted versions of the LFD and GEDT algorithms perform much better than before boosting. This confirms that learning the salient features of the data by the mean of supervised learning improves the performance of the descriptors as it captures the semantic structure of the database to query. Although we tested only the LFD and GEDT, our approach is more general and it can be applied to other types of descriptors.

The second observation is that the Boosted-LFD-GEDT descriptor outperforms the Boosted-LFD and Boosted-GEDT in most of the measures. This shows that combining different types of features precision as well as the the mean dynamic average recall of the retrieval algorithm. In our implementation we used a simple similarity measure for intra-class search for the combined descriptor. We believe that there is a window for improvement by investigating more elaborated similarity measures.

Finally, Figures 4 and 5 some retrieval results of the Boosted-LFD and Boosted-GEDT descriptors. We use the SHREC2006 queries (first column) and we show the top-10 best matches. Notice that for some queries (the dolphine for the Boosted-LFD and the horse for the Boosted-GEDT), the algorithm returned less then 10 results. This is an important property of our algorithm: it is able to say whether a model is relevant to the query or not and therefore discard irrelevant models from the retrieval list.

7. Conclusion

We proposed in this chapter a new framework for 3D model retrieval based on an off-line learning of the most salient features of the shapes. By using a boosting approach we are able to use a large set of features, which can be heterogeneous, in order to capture the high-level semantic concepts of different shape classes. The retrieval process is a combination of classification and intra-class search. The experimental results showed that (1) the boosted descriptors outperform their non-boosted counter part, and (2) an efficient combination of descriptors of different types improves significantly the retrieval performance.

	Methods	Value
1	Boosted-LFD-GEDT	0.64
2	Boosted-GEDT	0.59
3	Boosted-LFD	0.55
4	Shilane et al. (R3)	0.53
5	Zaharia et al. (R1)	0.50
6	Makadia et al. (R2)	0.48
7	Shilane et al. (R2)	0.48
8	Makadia et al. (R1)	0.47
9	GEDT	0.35
10	LFD	0.28

(a) Mean Average Precision(highly relevant).

	Participant	Value
1	Boosted-LFD-GEDT	0.74
2	Boosted-LFD	0.67
3	Boosted-GEDT	0.60
4	Shilane et al. (R3)	0.52
5	Zaharia et al. (R1)	0.51
6	Shilane et al. (R2)	0.49
7	Makadia et al. (R2)	0.43
8	Makadia et al. (R1)	0.42
9	GEDT	0.28
10	LFD	0.22

(b) Mean Average Precision (Relevant)

	Methods	Value
1	Boosted-LFD-GEDT	61.9%
2	Boosted-GEDT	56.2%
3	Boosted-LFD	51.95%
4	Makadia et al. (R2)	44.77%
5	Makadia et al. (R1)	43.77%
6	Daras et al. (R1)	42.74%
7	Papadakis et al.(R1)	41.85%
8	Shilane et al. (R3)	40.86%
9	GEDT	34.06%
10	LFD	24.51%

(c) Mean First Tier (Highly relevant).

	Participant	Value
1	Boosted-LFD	64.89%
2	Boosted-LFD-GEDT	64.46%
3	Boosted-GEDT	56.02%
4	Makadia et al. (R2)	40.55%
5	Makadia et al. (R1)	38.78%
6	Shilane et al. (R3)	37.40%
7	Papadakis et al.(R1)	37.40%
8	Shilane et al. (R2)	37.30%
9	GEDT	27.68%
10	LFD	21.63%

(d) Mean First Tier (Relevant)

	Participant	Value
1	Boosted-LFD-GEDT	57.84%
2	Boosted-GEDT	54.75%
3	Boosted-LFD	52.39%
4	Makadia et al. (R2)	27.86%
5	Makadia et al. (R1)	26.62%
6	Daras et al. (R1)	25.66%
7	Shilane et al. (R3)	25.63%
8	Papadakis et al.(R1)	25.61%
9	GEDT	20.57%
10	LFD	15.80%

(e) Mean Second Tier (Highly Relevant)

	Participant	Value
1	Boosted-LFD	64.50%
2	Boosted-LFD-GEDT	61.58%
3	Boosted-GEDT	55.70%
4	Shilane et al. (R2)	26.58%
5	Shilane et al. (R3)	26.26%
6	Makadia et al. (R2)	25.22%
7	Zaharia et al. (R1)	24.63%
8	Papadakis et al.(R1)	24.24%
9	GEDT	18.21%
10	LFD	14.32%

(f) Mean Second Tier (Relevant)

Table 1. Mean Average precision, Mean First Tier and Second Tier performance.

As future work, there are many avenues for improvements. First, most of existing 3D model repositories are often provided with a hierarchical classification. We plan to extend our framework to handle such structure of the data as well as fuzzy classification, since in nature a same model may belong to several categories simultaneously. Also we plan to

investigate on the meaning of the selected feature space for each shape class and extend the framework to the problem of building creative prototypes of 3D models. The prototype should capture the high-level semantic features of the class.

8. Acknowledgement

9. References

Ankerst, M., Kastenmoller, G., Kriegel, H.-P. and Seidl, T. (1999), Nearest neighbor classification in 3D protein databases, *in* 'the Seventh International Conference on Intelligent Systems for Molecular Biology', AAAI Press, pp. 34–43.

Biasotti, S., Marini, S., Spagnuolo, M. and Falcidieno, B. (2006), "Sub-part correspondence by structural descriptors of 3D shapes.", *Computer-Aided Design* , Vol. 38, pp. 1002–1019.

Chen, D.-Y., Tian, X.-P., Shen, Y.-T. and Ouhyoung, M. (2003), "On visual similarity based 3D model retrieval.", *Computer Graphics Forum* , Vol. 22, pp. 223–232.

Funkhouser, T. A., Min, P., Kazhdan, M. M., Chen, J., Halderman, J. A., Dobkin, D. P. and Jacobs, D. P. (2003), "A search engine for 3D models.", *ACM Transactions on Graphics* , Vol. 22, pp. 83–105.

Funkhouser, T. and Shilane, P. (2006), Partial matching of 3D shapes with prioritydriven search, *in* 'SGP '06: Proceedings of the fourth Eurographics Symposium on Geometry Processing', Eurographics Association, pp. 131–14.

Hao, W. and Luo, J. (2006), Generalized multiclass adaboost and its applications to multimedia classification, *in* 'CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop', IEEE Computer Society, p. 113.

Hilaga, M., Shinagawa, Y., Kohmura, T. and Kunii, T. L. (2001), Topology matching for fully automatic similarity estimation of 3D shapes, *in* 'Proceedings of the 28th annual conference on Computer graphics and interactive techniques', ACM Press, pp. 203–212.

Hou, S., Lou, K. and Ramani, K. (2005), "SVM-based semantic clustering and retrieval of a 3D model database", *Journal of Computer Aided Design and Application* , Vol. 2, pp. 155–164.

Hou, S. and Ramani, K. (2006), A probability-based unified 3D shape search, *in* 'European Commission International Conference on Semantic and Digital Media Technologies, Lecture notes in computer science', Vol. 4306, pp. 124–137.

Ip, C. Y., Regli, W. C., Sieger, L. and Shokoufandeh, A. (2003), Automated learning of model classifications, *in* 'SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications', ACM Press, pp. 322–327.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y. and Ramani, K. (2005), "Threedimensional shape searching: state-of-the-art review and future trends.", *Computer-Aided Design*, Vol. 37, pp. 509–530.

Johnson, A. (1997), Spin-Images: A Representation for 3-D Surface Matching, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Kazhdan, M., Funkhouser, T. and Rusinkiewicz, S. (2003), Rotation invariant spherical harmonic representation of 3D shape descriptors, *in* 'SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing', pp. 156–164.

Laga, H., Takahashi, H. and Nakajima, M. (2006), Spherical wavelet descriptors for content-based 3D model retrieval, *in* 'SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)', pp. 75–85.

Lee, C. H., Varshney, A. and Jacobs, D.W. (2005), Mesh saliency, *in* 'SIGGRAPH '05: ACM SIGGRAPH 2005 Papers', ACM Press, New York, NY, USA, pp. 659–666.

Lew, M. S., Sebe, N., Djeraba, C. and Jain, R. (2006), "Content-based multimedia information retrieval: State of the art and challenges", *ACM Trans. Multimedia Comput. Commun. Appl.* , Vol. 2, ACM Press, New York, NY, USA, pp. 1–19.

M.Kortgen, G-J.Patrick, M.Novotni and R.Klein (2003), 3D shape matching with 3D shape contexts, *in* 'the 7th Central European Seminar on Computer Graphics'.

Novotni, M. and Klein, R. (2003), 3D Zernike descriptors for content based shape retrieval, *in* 'SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications', ACM Press, New York, NY, USA, pp. 216–225.

Ohbuchi, R. and Kobayashi, J. (2006), Unsupervised learning from a corpus for shapebased 3D model retrieval, *in* 'MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval', ACM Press, pp. 163–172.

Ohbuchi, R., Kobayashi, J., Yamamoto, A. and Shimizu, T. (2007), Comparison of dimension reduction method for database-adaptive 3D model retrieval, *in* 'Fifth International Workshop on Adaptive Multimedia Retrieval (AMR 2007)'.

Reuter, M., Wolter, F.-E. and Peinecke, N. (2006), "Laplace-Beltrami spectra as "shape-DNA" of surfaces and solids", *Computer-Aided Design* , Vol. 38, pp. 342– 366.

Schapire, R. E. (2003), The boosting approach to machine learning: An overview., *in* 'In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, Nonlinear Estimation and Classification', Springer.

Shilane, P. and Funkhouser, T. (2006), "Selecting Distinctive 3D Shape Descriptors for Similarity Retrieval", *IEEE International Conference on Shape Modeling and Applications (SMI2006)* , Vol. 0, IEEE Computer Society, Los Alamitos, CA, USA, p. 18.

Shilane, P., Min, P., Kazhdan, M. and Funkhouser, T. (2004), The princeton shape benchmark, *in* 'SMI'04: Proceedings of the Shape Modeling International 2004 (SMI'04)', pp. 167–178.

Tangelder, J. W. and Veltkamp, R. C. (2004), A survey of content based 3D shape retrieval, *in* 'Shape Modeling International 2004, Genova, Italy', pp. 145–156.

Tieu, K. and Viola, P. (2004), "Boosting image retrieval", *International Journal of Computer Vision* , Vol. 56, Kluwer Academic Publishers, Hingham, MA, USA, pp. 17–36.

T.Tung and F.Schmitt (2005), "The augmented multiresolution reeb graph approach for content-based retrieval of 3D shapes", *International Journal of Shape Modeling (IJSM)* , Vol. 11, pp. 91–120.

Veltkamp, R. C., Ruijsenaars, R., Spagnuolo, M., van Zwol, R. and ter Haar, F. (2006), SHREC2006: 3D Shape Retrieval Contest, Technical Report UU-CS- 2006-030, Department of Information and Computing Sciences, Utrecht University.

Xu, C., Tan, T., Li, S. Z., Wang, Y. and Zhong, C. (2006), Learning effective intrinsic features to boost 3D-based face recognition, *in* 'ECCV 2006, 9th European Conference on Computer Vision', Springer, pp. 416–427.

Yamauchi, H., Saleem, W., Yoshizawa, S., Karni, Z., Belyaev, A. and Seidel, H.- P. (2006), Towards stable and salient multi-view representation of 3D shapes, *in* 'Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)', p. 40.

	Methods	Value
1	Boosted-LFD-GEDT	0.66
2	Boosted-GEDT	0.58
3	Boosted-LFD	0.57
4	Makadia et al. (R2)	0.55
5	Makadia et al. (R1)	0.54
6	Daras et al. (R1)	0.52
7	Chaouch et al. (R1)	0.50
8	Papadakis et al.(R1)	0.49
9	GEDT	0.41
10	LFD	0.32

(a) Mean Dynamic Average Recall

	Methods	Value
1	Daras et al. (R1)	0.68
2	Boosted-LFD-GEDT	0.68
3	Makadia et al. (R1)	0.68
4	Shilane et al. (R3)	0.68
5	Makadia et al. (R2)	0.66
6	Chaouch et al. (R1)	0.65
7	Boosted-LFD	0.62
8	Boosted-GEDT	0.62
9	GEDT	0.60
10	LFD	0.48

(b) MNCG @5

	Methods	Value
1	Boosted-LFD-GEDT	0.67
2	Makadia et al. (R2)	0.61
3	Makadia et al. (R1)	0.60
4	Daras et al. (R1)	0.60
5	Pepadakis et al.(R1)	0.59
6	Chaouch et al. (R1)	0.58
7	Boosted-LFD	0.57
8	Boosted-GEDT	0.56
9	GEDT	0.50
10	LFD	0.40

(c) MNCG @10

	Method	Value
1	Boosted-LFD-GEDT	0.59
2	Makadia et al. (R2)	0.55
3	Makadia et al. (R1)	0.52
4	Daras et al. (R1)	0.52
5	Shilane et al. (R3)	0.52
6	Papadakis et al.(R1)	0.50
7	Boosted-LFD	0.45
8	GEDT	0.39
9	Boosted-GEDT	0.38
10	LFD	0.32

(d) MNCG @25

	Method	Value
1	Makadia et al. (R2)	0.55
2	Shilane et al. (R3)	0.53
3	Makadia et al. (R1)	0.53
4	Daras et al. (R1)	0.52
5	Papadakis et al. (R1)	0.51
6	GEDT	0.40
8	Boosted-LFD-GEDT	0.39
7	Boosted-GEDT	0.34
9	LFD	0.31
10	Boosted-LFD	0.22

(e) MNCG@50

	Method	Value
1	Makadia et al. (R2)	0.59
2	Papadakis et al.(R1)	0.58
3	Makadia et al. (R1)	0.56
4	Daras et al. (R1)	0.56
5	Shilane et al. (R2)	0.55
6	GEDT	0.43
7	Boosted-GEDT	0.41
8	Boosted-LFD-GEDT	0.34
9	LFD	0.33
10	Boosted-LFD	0.24

(f) MNCG@100

Table 2. Dynamic Average Recall and, Mean Normalized Cumulated Gain (MNCG) performance.

	Methods	Value
1	Daras et al. (R1)	0.70
2	Makadia et al. (R1)	0.69
3	Shilane et al. (R3)	0.69
4	Boosted-LFD-GEDT	0.68
5	Chaouch et al. (R1)	0.68
6	Makadia et al. (R2)	0.68
7	Boosted-LFD	0.63
8	Boosted-GEDT	0.63
9	GEDT	0.61
10	LFD	0.52

(a) MNDCG@5

	Methods	Value
1	Boosted-LFD-GEDT	0.68
2	Daras et al. (R1)	0.64
3	Makadia et al. (R1)	0.64
4	Makadia et al. (R2)	0.64
5	Chaouch et al. (R1)	0.62
6	Pepadakis et al.(R1)	0.62
7	Boosted-LFD	0.59
8	Boosted-GEDT	0.58
9	GEDT	0.54
10	LFD	0.45

(b) MNDCG@10

	Method	Value
1	Boosted-LFD-GEDT	0.60
2	Makadia et al. (R2)	0.59
3	Daras et al. (R1)	0.58
4	Makadia et al. (R1)	0.57
5	Shilane et al. (R3)	0.56
6	Papadakis et al.(R1)	0.55
7	Boosted-LFD	0.46
8	GEDT	0.45
9	Boosted-GEDT	0.41
10	LFD	0.38

(c) MNDCG@25

	Method	Value
1	Makadia et al. (R2)	0.58
2	Makadia et al. (R1)	0.57
3	Daras et al. (R1)	0.56
4	Shilane et al. (R3)	0.56
5	Papadakis et al. (R1)	0.54
6	GEDT	0.45
7	Boosted-LFD-GEDT	0.41
8	LFD	0.37
9	Boosted-GEDT	0.33
10	Boosted-LFD	0.21

(d) MNDCG@50

	Method	Value
1	Makadia et al. (R2)	0.59
2	Makadia et al. (R1)	0.57
3	Daras et al. (R1)	0.57
4	Papadakis et al.(R1)	0.56
5	Shilane et al. (R3)	0.55
6	GEDT	0.45
7	LFD	0.37
8	Boosted-GEDT	0.36
9	Boosted-LFD-GEDT	0.36
10	Boosted-LFD	0.22

(e) MNDCG@100

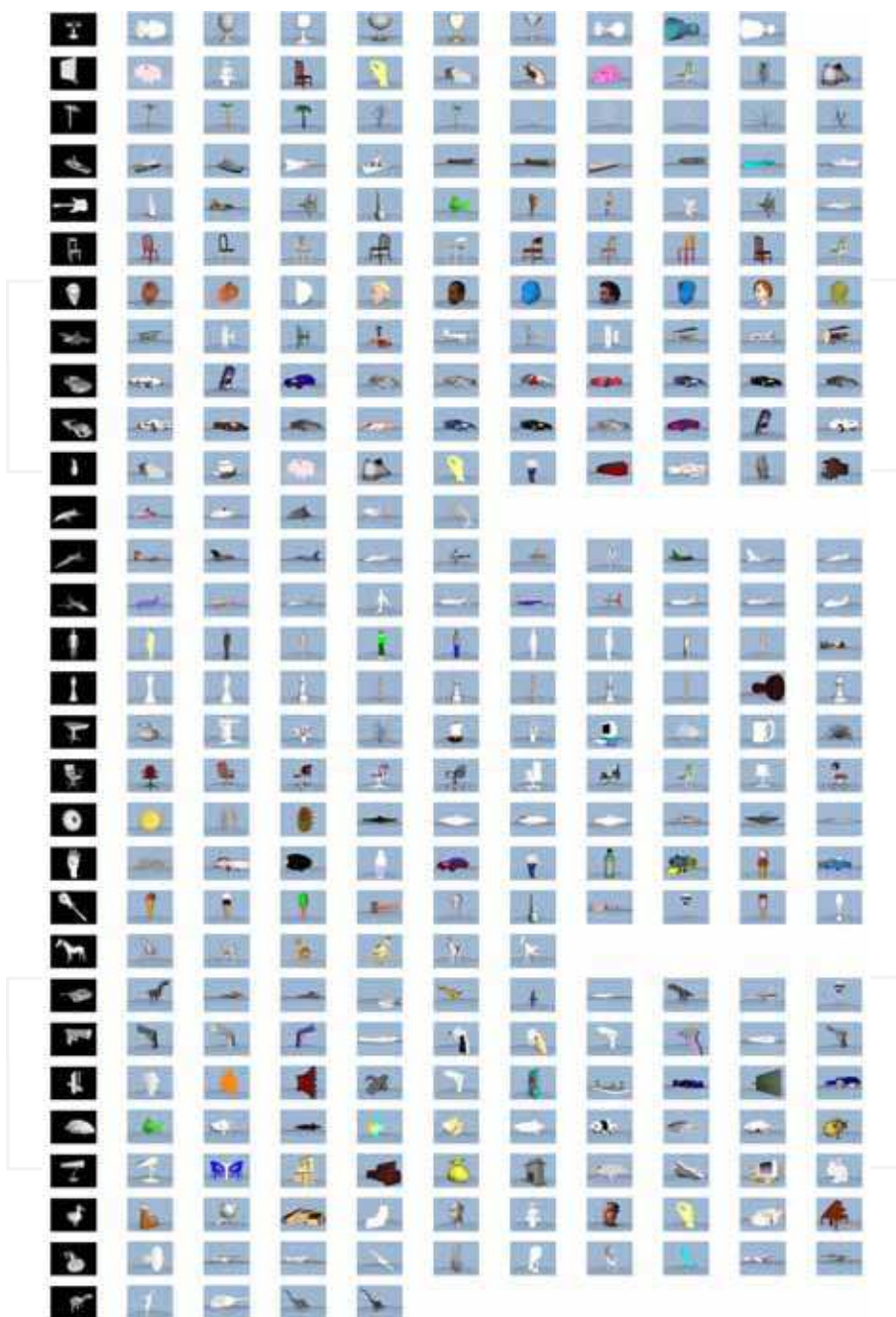Table 3. Mean Normalized Discounted Cumulated Gain (MNDCG) performance.

Fig. 4. Some retrieval results using the Boosted-LFD Descriptors. The models in the first column are used as query models2.2The 10-best matches are displayed.

Fig. 5. Retrieval results using the Boosted-GEDT Descriptors. The models in the first column are used as query models. T2h3e 10-best matches are displayed.

Machine Learning

Edited by Abdelhamid Mellouk and Abdennacer Chebira

ISBN 978-953-7619-56-1

Hard cover, 450 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

Machine Learning can be defined in various ways related to a scientific domain concerned with the design and development of theoretical and implementation tools that allow building systems with some Human Like intelligent behavior. Machine learning addresses more specifically the ability to improve automatically through experience.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hamid Laga (2009). 3D Shape Classification and Retrieval Using Heterogenous Features and Supervised Learning, Machine Learning, Abdelhamid Mellouk and Abdennacer Chebira (Ed.), ISBN: 978-953-7619-56-1, InTech, Available from:

http://www.intechopen.com/books/machine_learning/3d_shape_classification_and_retrieval_using_heterogenous_features_and_supervised_learning

INTECH
open science | open minds