Dipartimento di Informatica e
Scienze dell'Informazione

Improving 3D Shape retrieval with SVM

Elisabetta Delponte, Curzio Basso, Francesca Odone and Enrico Puppo

Technical Report **DISI-TR-2008-02**

DISI, Università di Genova

v. Dodecaneso 35, 16146 Genova, Italy http://www.disi.unige.it/

Improving 3D Shape retrieval with SVM

Elisabetta Delponte, Curzio Basso, Francesca Odone and Enrico Puppo

Abstract: In this paper we propose a technique that combines a classification method from the statistical learning literature with a conventional approach to shape retrieval. The idea that we pursue is to reduce the shapes gallery complexity by filtering it before retrieval with a classifier for shapes, that allows us to keep only the shapes belonging to the classes similar to the query shape. The experimental analysis that we report shows how our approach improves the computational cost in the average case, as well as leading to better results.

Contents

1 Introduction

Recent advances in 3D digitalization techniques lead to an increase of available 3D shapes datasets — see, for instance the Princeton dataset [32], the National Taiwan University dataset [28], or the data collected under the EU funded project AIM@SHAPE [1]. Among the application domains where 3D shapes are applied with profit we recall cultural heritage and archaeology (e.g., the digital Michelangelo project [24]), and biology (e.g., the well known Protein DataBank [29, 12]). In order to be able to access these 3D shapes repositories it may be convenient to set up retrieval systems. Such systems support the user in extracting from a potentially big repository only the shapes that match a given specificiation. For a recent survey on 3D shape retrieval see, e.g., [7].

In this paper we focus on the query by example (QBE) approach, very common on multimedia retrieval systems. In such a context, retrieval is based on applying appropriate similarity measures to the query example (or test example) shape descriptor and all or some descriptors from the dataset (or gallery, or repository). Therefore, most research in this direction focused on finding robust and discriminative shape descriptors, on top of which applying rather conventional similarity measures [27, 23, 31, 15, 33]. From the pattern recognition point of view, similarity matching is usually coupled with a nearest neighbour approach, that orders the available dataset according to the degree of similarity with respect to the shape used as a query. Efficient indexing techniques, derived from information retrieval literature, may be adopted [33]. It is easy to understand that this approach can suffer from the increase of the number of shapes available in the repository and, in the case the chosen descriptor is not discriminant enough, retrieval can easily fail.

We propose to adopt a combined strategy that couples conventional shape retrieval with classification methods coming from the statistical learning literature, in order to increase the retrieval efficiency and effectiveness as the shape repository size grows. For what concerns the classification algorithms to adopt, we focus here on the well known Support Vector Machines (SVMs) [11]. Different regularized approaches (e.g. RLS [30] or iterative methods [17]) could be applied as well, at the price of some loss in performance.

Throughout the paper we will point out how the choice of the shape distribution and of the distance measure is not crucial for the method that we propose. In fact, we start off from a simple and well known approach to shape retrieval [27] and show that filtering the repository based on classification can improve performance and results.

More in details, let us assume that we have a dataset of labeled shapes divided in N classes. We describe each shape by means of the D2 descriptor [27], which is a distribution of the distances between two random points of the shape. Then, on the basis of D2 descriptions, we exploit a classification tool to select a reduced number of classes more similar to a given test or query shape. Finally we perform a retrieval on the basis of a similarity measure (the L_1 norm) computed for the shapes belonging to the reduced number of classes obtained with previous classification. This initial classification step improves performance and effectiveness of retrieval.

The paper is organized as follows.

2 Background on statistical learning

In this section we recall some basics of statistical learning that will be used thorughout the paper. Specifically we focus on penalized empirical risk minimization approaches to supervised learning and, implicitly, we refer to classification problems.

We first assume we are given two random variables $\mathbf{x} \in X \subseteq \mathbb{R}^d$ (input) and $y \in Y \subseteq \mathbb{R}$ (output) — in the classification case $y \in \{-1, 1\}$. We then consider a set of data

$$S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$$

that we call a *training set* obtained by sampling the set $X \times Y$ according to a fixed but unknown probability distribution $p(\mathbf{x}, y)$.

Supervised learning approaches use the training set to *learn* a function $f : X \to Y$ that can be applied to previously unseen data. Indeed, we say that an algorithm is *predictive* or that it has good generalization properties if it applies successfully to data other than training examples. This is important to obtain functions that may be used in practical applications. Predictivity is a trade-off between information provided by training data and the complexity of the solution we are looking for. The latter is related to the choice of an appropriate hypothesis space $\mathcal{H}$, i.e., the space of functions where we look for our solution f. The ideal solution is defined as the minimizer of the *expected risk*

$$I[f] = \int_{X \times Y} V(y, f(\mathbf{x}))p(\mathbf{x}, y)d\mathbf{x}dy$$

where V is some loss function measuring the goodness of our solution f. Unfortunately, the minimizer of $I[f]$ cannot be found in practice since $p(\mathbf{x}, y)$ is unknown. One could build an approximation of it, based on the training set, called the *empirical risk*

$$I_{emp}[f, S] = \frac{1}{n} \sum V(y, f(\mathbf{x}_i)).$$

However, the minimizer of the empirical risk is not unique (and thus the minimization problem is ill posed) and, if the space $\mathcal{H}$ is too large it may lead to complex solutions, and compromise prediction properties. To restore well-posedness a possible way is to look for a regularized solution by constraining the hypothesis space via a regularization or penalty term. We thus re-write the minimization problem as

$$\min_{f} \quad I_{emp}[f, S] + \lambda PEN(f)$$

where λ is a *regularization paramterer*, trading off between the two terms. Within this framework we will refer in particular to the so-called Tikhonov regularization, using a L^2-norm in $\mathcal{H}$ as a penalization term:

$$\min_{f} \frac{1}{n} \sum V(y, f(\mathbf{x}_i)) + \lambda ||f||^2_{\mathcal{H}}.$$

As for the choice of $\mathcal{H}$, a very useful class of spaces to enclose some notion of smoothness in their norm are the Reproducing Kernel Hilbert Spaces (RKHS). It can be shown [38] that to every RKHS correspond a unique positive-definite function K that we call a *kernel function*. Conversely, for each positive-definite function K on $X \times X$ there is a unique RKHS $\mathcal{H}$ that has K as a reproducing kernel. Roughly speaking, the kernel function K can be expressed as a dot product in a higher dimensional space. Choosing a kernel or, equivalently, choosing a hypothesis space, allows us to formalize a non-linear problem by mapping the original observations (feature vectors in the input space) in a different space where a linear algorithm may be applied. Thanks to the kernel the mapping may be performed implicitly with the so-called kernel trick:

$$K(\mathbf{x}_1, \mathbf{x}_2) = \phi(\mathbf{x}_1) \cdot \phi(\mathbf{x}_2)$$

This makes a linear classification in the new space equivalent to non-linear classification in the original space. In practical terms, the choice of an appropriate kernel is often associated to choosing a similarity measure between feature vectors. If no prior knowledge on how to compare feature vectors is available, a number of off-the-shelf kernels may be used. Besides from the linear kernel that corresponds to a linear learning algorithm we mention *polynomial kernels* $K(\mathbf{x}_i, \mathbf{x}_j) = (< \mathbf{x}_i \cdot \mathbf{x}_j > +1)^d$ that, in the classification case, lead to polynomial separating functions in the input space and *Gaussian kernels*, $K(\mathbf{x}_i, \mathbf{x}_j) = exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/2\sigma^2)$. The parameter σ represents standard deviation and needs to be tuned appropriately. Its choice is somehow alternative to the choice of λ: a small σ may lead to overfitting, a big σ to oversmoothing. A very important theorem in statistical learning is the *representer theorem* stating that, under general conditions on the loss function V, the minimizer of a Tikhonov regularization problem is of the form $f(\mathbf{x}) = \sum_{i=1}^{n} c_i K(\mathbf{x}_i, \mathbf{x})$ for some $(c_1, \ldots, c_n) \in \mathbb{R}^n$.

The choice of the loss function leads to different learning algorithms. The L^2-norm leads to Regularized Least Squares (RLS) algorithms [35, 8]. Instead the so-called Hinge loss $(1-yf(x))_+$ leads to Support Vector Machines (SVM) [36]. SVMs have been used with success in a number of different application domains. They are characterized by many nice properties, some of them are not apparent from the regularized formulation followed in this section. We mention here, as it will be useful later in the paper, the fact that they produce a sparse solution on the set of input data. This means that the solution they produce $(c_1, \ldots, c_n)$ tends to be sparse (will contain few non-zero entries). The training data associated to non-zero weights are referred to as *support vector*. For a geometric intuition of such a property the reader is referred to [36].

3 Related work

3.1 3D shape retrieval

The problem of retrieval and matching of shapes has been extensively studied in numerous fields such as computer vision, computer graphics and molecular biology. See [7, 19, 33] for general surveys on this subject.

Most approaches to shape retrieval are based on shape descriptors and exhaustive search. A shape descriptor is computed for each object in a database, as well as for the query object.

Then the descriptor of the query object is compared with *all* descriptors of objects in the database through some measure for pattern matching, and a ranked list of most similar objects is retrieved. Most efforts in the literature have been devoted to find shape descriptors that are robust, can give good results in term of retrieval, and can be computed and compared efficiently. A large class of works deal with global descriptors in the form of *feature vectors*, i.e., arrays of scalar values computed by some analysis of the 3D shape (e.g., [4, 10, 14, 16, 27, 37]). Some such feture vectors are in fact histograms of either scalar fields computed on the shape, or some form of shape distriution. The D2 shape distribution histogrm [27] that we adopt here, is one of the simplest descriptors to compute. More recent descriptors exhibits better performances though. As already remarked, the scope of this work is somehow orthogonal to the descriptor adopted, as long as this falls in the class of feature vectors, so our approach could (and will) be adopted also with other descriptors.

Other works use descriptor based on local features and more complex matching procedures [9, 15, 22, 31].

Spin images [22] are often used to represent a 3D surface by means of a 2D image and thus they can be considered as 3D shape descriptors. Spin images have been used for many different applications: object recognition, studying the protein docking, 3D shapes retrieval.

More recently Gal and Cohen-Or [15] introduced a new local surface descriptor that is called *salient geometric feature* and consists of a cluster of descriptors that locally describe a nontrivial region of the surface. The salient features of a surface typically characterize the surface well and form a basis for a nonglobal similarity measure among subparts of shapes.

Shilane and Funkhouser [31] start computing a descriptor based on harmonic shape for a random set of points and a given set of scales of the object. Then they focus on the computation of how distinctive every shape descriptor is with respect to a database containing multiple classes of objects. Instead of finding an explicit definition of important features, they select them on the basis of shape classification performance.

Castellani et al. [9] propose a local descriptor based on hidden Markov models.

3.2 Supervised learning for 3D shape classification

A few approaches to 3D object retrieval based on statistical learning have been proposed in the literature. In their seminal work, Elad, Tal and Ar [13] proposed a semi-interactive retrieval system based on a feature vector descriptor and on the use of SVM together with relevance feedback. Different method based on relevance feedback were compared by Novotni et al. [25] and those based on SVM were found to give better performances. Relevance feedback on SVM was also adopted in a more recent work by Leng, Qin and Li [5], where the authors adopt an algorithm for SVM active learning, proposed in 2001 for image retrieval.

Hou, Lou and Ramani [18] use SVM for organizing a database of shapes through clustering and then to perform the classification and retrieval. A test shape is first classified to belong to one of the clusters that form the database, then only such a cluster is searched for similar shapes. In [39] the authors propose a simple mechanism to exploit classification methods for retrieval problems. Given a training set for N classes, they train N one-vs-all classifiers (implemented by back-propagation neural networks). At evaluation time, the values returned by each classifier can be ranked, resulting in an N-dimensional vector of rankings. The retrieval is then performed as a usual matching, but with a metric which mixes the euclidean distance between the descriptors of the two shapes and a distance between the vectors of class rankings.

Shape classification has also been addressed by Barutcouglu and De Coro [3] by using a bayesian approach to exploit the dependence between classes, assuming that they are organized in a hierarchical fashion. They show how, taking explicitly into account hierarchies and discarding inconsistent results, classification results improve. However, their work is focused only on the classification problem, and retrieval is not addressed.

A peculiarity of kernel methods is that they allow for the design of ad hoc kernels able to capture the expressiveness of feature vectors. A number of kernels for different application domains have been proposed in the literature [34]. For instance, in the case of histogram-like descriptions, a possible way to deal with them is to treat them as probability distributions and to resort to kernels defined on probability measures [20]. Alternatively, kernel functions derived from signal or image processing may be adopted — see, for instance, [26].

All methods above feed the classification algorithms with shape descriptors of a unique type.

However, in [2] the authors propose a fusion algorithm, based on the so-called empirical ranking risk minimization, which combines different descriptor. The algorithm can also be implemented via SVMs. The learning algorithm returns the weights to associate to the various descriptors. After this, conventional retrieval may be performed, for instance by means of relevance feedback.

4 Our approach

We propose a technique that is a combination of classical retrieval methods with a classification approach typical of the learning from examples framework.

We assume that our repository of shapes is labeled and that, for simplicity, a unique label is assigned to each shape. We can thus organize our repository in classes using the labels to build a dataset per each class. The idea that we pursue is to reduce complexity of search by filtering the available dataset before retrieval, i.e., by using only shapes belonging to the k most relevant classes to the query or test shape.

As a benchmark method, we consider a popular work proposed by Osada *et al.*, based on statistical shape descriptors [27]. In that work, shape distributions of two different 3D models (the D2 descriptors represented as histograms of the distance between pairs of vertices randomly selected on the shape) are compared with the L1 distance. Thus, for a given test shape, its description is compared with all the descriptions available on the database, and the output returned by the retrieval system is a ranked list of the n most similar shapes. In this paper, we adopt the same shape description and follow a similar pipeline. Our variation of the original pipeline consists of performing shape classification prior to shape retrieval, in order to reduce the size of the dataset of shapes to be analyzed.

The reminder of the section describes the details of the classification (filtering) procedure and discusses the computational advantages of our choice. Then we evaluate the appropriateness of our choice in terms of nearest-neighbour performance and retrieval indicators (the so-called first tier and second tier).

4.1 Classification with SVM

Let us assume we start from a shape repository containing M labeled shapes, belonging to N different classes. Each class is composed by a number of shapes that could be arbitrarily different.

We aim at designing a shape classifier that returns the most likely shape classes $\hat{S} = \{S_1, \ldots, S_k\}$, to a shape example. This will allow us to restrict retrieval to the shapes belonging to set $\hat{S}$.

The shape classification problem that we consider is a *multi-class classification* problem. We adopt a one-vs-all procedure that requires we train N binary classifiers of the type class $\mathcal{C}_i$ $i = 1, \ldots, N$ versus all the other classes (see, e.g., [6]).

Thus, for each class we train a classifier: each training set is made of an equal number of positive and negative examples of the shapes, where the negative examples are extracted from all the other classes of the dataset.

In this work we adopt Support Vector Machines (SVM) [36] with Gaussian kernels as a classification algorithm. Our choice is mainly motivated by performance reasons, in particular the fact that the solution of a SVM classifier is sparse on the training data, but other binary classifiers could be used within the same pipeline. In practice, we use the SVM implementation of SVM^{light} [21].

The optimal values for the two free parameters, the width σ of the Gaussian kernel and the regularization parameter $C = (\lambda n)^{-1}$, are set with a standard leave-one-out (LOO) procedure (see for instance [6]).

4.2 Classes selection and partial retrieval

At run time, we apply a test or query example to each classifier available, then we rank the output of the classification results, thus obtaining a list of classes sorted from the most likely to the least likely to a given test.

We exploit this result to filter shapes available in the repository before we apply a standard nearest neighbor (NN) retrieval. In other words, given a query example, we test it against the N classifiers, rank the results and then keep the shapes belonging to the first k classes for the following retrieval process.

Notice how the choice of an appropriate k is crucial both for computational and performance reasons. A small k will make the retrieval very fast but it may impoverish the results. A big k ($k \to N$) would increase retrieval time but not necessarily improve performance - see the discussion in Section 5.

4.3 Computational advantages

The computational cost of shape filtering followed by a retrieval restricted on the filtered classes is equal to the maximum cost between the two operations.

If the filtering procedure considers all shapes from the training set (that is, if it consists in comparing the query shape against all training shapes) the filtering phase is more costy than actual retrieval, no matter the choice of k, and it is equivalent to conventional retrieval in terms of number of comparisons between shapes performed.

For instance, if we consider a Regularized Least Squares (RLS) [35, 8] approach, the test datum x is compared with *each* training datum x_i via a kernel function K, then for each class C we have

$$f(x) = \sum_{i=1}^{N_C} c_i K(x, x_i).$$

In the case of SVM, instead, the sparsity on the training data in the obtained solution means that filtering requires fewer comparisons - one per each support vector - since in this case the summation runs on the support vectors only. There is no way to evaluate a priori the number of obtained support vectors, as they depend on the training set (both their cardinality and the data representation chosen) and on the choice of the kernel function. In the average case, and assuming an appropriate choice of the representation and the kernel function, we observed a saving in terms of number of comparisons between the test datum and training data. This effect becomes more relevant as the training set size grows (see Section 5).

5 Experimental results

In this section we show the results of retrieval made on a subset of the Princeton Shape Benchmark (PSB) dataset [32] and we compare our results with the ones reported in [27].

We first describe the dataset, then specialize the filtering and retrieval procedure that we adopted on such data, reporting the experimental results obtained.
To evaluate the performances of the retrieval we use the following evaluation methods:

Nearest Neighbour : the percentage of closest matches that belong to the same class as the query.

First Tier : the percentage of models in the query's class that appear within the top k matches where k is the size of the query's class.

Second Tier : the percentage of models in the query's class that appear within the top $2k$ matches where k is the size of the query's class.

5.1 The dataset

The Princeton Shape Benchmark (PSB) is a publicly-available database of 3D models, widely adopted by the shape retrieval community. The dataset contains 1814 polygonal models collected from the World Wide Web and classified by humans with respect to function and form in 27 classes. For our experiments, we have used the PSB and the classes organization provided in it: each shape class is evenly split in two parts, one for the training (907 shapes) and another one for the test (907 shapes).
According to the classification provided with the PSB, the dataset is split in 42 categories for training and 38 for test, therefore some of the training classes are not present in the test. Moreover there are some of the categories which contain very few examples (less than 10) and there are some classes containing objects with very different shapes.
Indeed, one of the main problems of the classification provided with the PSB is that some of the classes are clustered on a functional basis: this means they are grouped according to their semantics, rather thn to their 3D shape. For instance there is a class named "musical instruments" which contains guitars as well as pianos (see Figure 1). Another class named "handheld" contains shapes of icecreams, skate-boards and microscopes (see Figure 2). This does not favour retrieval based on shape similarity.
On the basis of previous considerations we choose a subset of 18 PSB classes, that fulfill requirements of intra-class shape homogeneity and a threshold for cardinality (we keep only classes

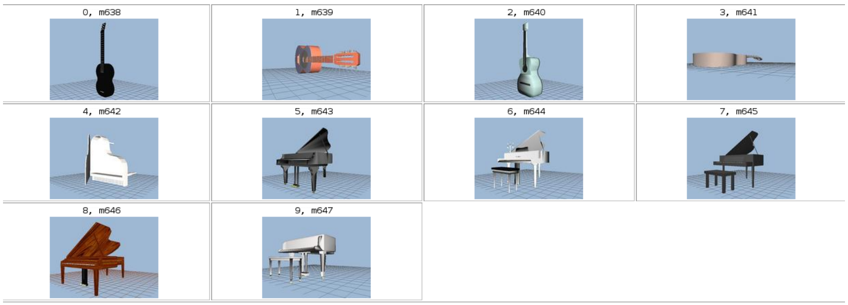

Figure 1: The 3D models contained in the class of "musical instrument". It consists of only 10 shapes.

with at least 10 elements on the training and on the test set).

It is worth noticing that in shape retrieval it is common practice to use subsets of available datasets: for instance in [16] the authors focus on flexible objects, thus they select few classes of those presented in PSB. This may cause problems if trying to compare different methods by referring to previously published results. Therefore in our comparisons with [27] we recompute the performances obtained with the method on our dataset. Table 1 shows the set of 18 classes that we have chosen for our experiments with relative cardinality for the training and test sets that are used for classification procedure. Each training set contains the same number of positive and negative examples: the negative are randomly selected from all the other classes of shapes. Each test set contains the positive examples given by the PSB dataset and as negative examples we have chosen to use all the other 3D shapes belonging to the test set. Therefore the number of negative test examples is usually very high.

5.2 Filtering and retrieval results

We train 18 SVM classifiers according to the previously described procedure. After a preliminary analysis on the performance of different standard kernels, we adopted a Gaussian kernel. Because of the presence of very small classes, we adopted a LOO procedure to select the regularization parameter C and the parameter σ of the Gaussian kernel.

First of all, we analyse the solutions obtained by the 18 classifiers, noticing how the number of support vectors becomes significantly smaller than the training set size when the latter grows

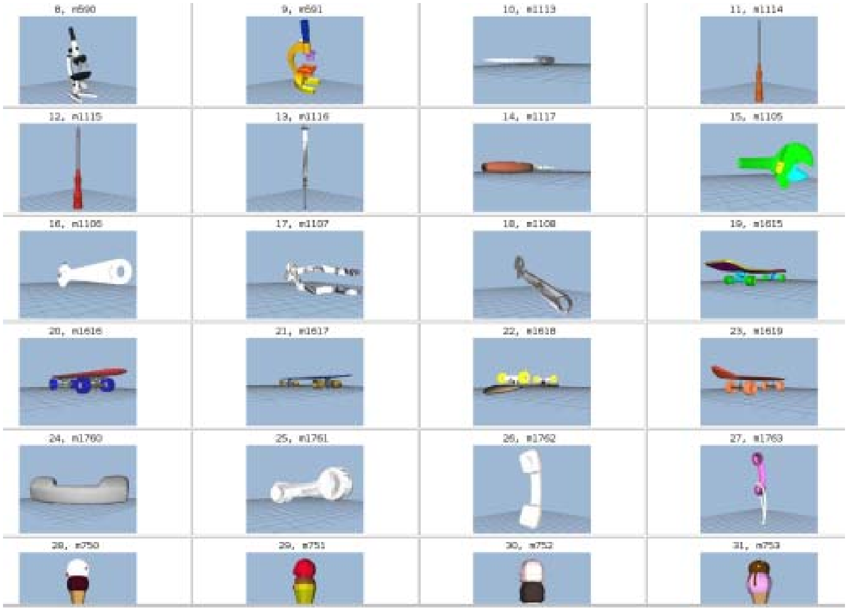

Figure 2: A subset of the 40 shapes contained in the class of "handheld". It is apparent that these objects are clustered according to their function and not their shape similarity.

(see Figure 3). This is an advantage, suggesting that if we adopt SVMs the sparsity of the solution with respect to the training data may reflect on an overall computational saving. The dataset considered in our experiments is too small to allow for an exaustive analysis. At run time we test all test shapes against the 18 classifiers. Table 2 shows the ranking of the percentage of correctly retrieved shapes for each SVM classifier. Each row refers to a given test class, in decreasing size order. For each test shape S_{tj} we consider its class ranking $C_1^j, \ldots, C_{18}^j$. Columns $Ri, i = 1, \ldots, 18$ show the percentage of test shapes of a given class including the correct class in the first i classes of their ranking. An alternative view of the same information, averaged with respect to the classes, is described in Fig. 4 where median values and standard deviations are highlighted. The small crosses (+) indicate the presence of an outlier class the does not behave in line with other classes (a further inspection shows that it is class "hat"). Standard deviations decrease as k grows, meaning that the average behaviour of the various classes becomes more homogeneous - this is apparently the only advantage in keeping high values for k.

As we pointed our previously in the paper, the choice of the number of classes to keep for shape retrieval is crucial and there is no obvious common sense rule to apply. Ideally, one should choose i in order to guarantee that the correct class is always kept. Observe how, in

Class name	Training set		Test Set	
	Positive	Negative	Positive	Negative
Aircraft winged vehicle	107	107	135	772
Plant	78	78	60	847
Animal biped human	71	71	78	829
Vehicle car	62	62	51	856
Furniture table	43	43	35	872
Furniture seat	40	40	37	870
Body part head	40	40	38	869
Liquid container	35	35	24	883
Sea vessel	19	19	26	881
Aircrafthelicopter	17	17	18	889
Displaydevice	16	16	24	883
Lamp	14	14	8	899
Animal quadruped	14	14	17	890
Furnitureshelves	13	13	13	894
Animal underwater creature	12	12	23	884
Hat	10	10	6	901
City	10	10	10	897
Door	10	10	18	889

Table 1: The set of classes that we have chosen for our experiments: each class is split in a training set and a test set.

the example under consideration, this would mean keeping *all* classes. Thus we weaken the requirement choosing k so that *enough* correct classes are kept. Notice, for instance, that if we consider the first 4 classes we keep the correct class on our dataset for about 66%.

Obviously the presence of the right class in the reduced set of shapes does not guarantee a successful retrieval, but its absence means that the retrieved elements will all be wrong. At the same time we notice how, as the size of the dataset grows, the average performance of the retrieval indicators degrade. Fig. 5 shows how they vary as the number of retained classes grow. From this analysis it is the case to conclude that a small number of classes ([2-5]) should

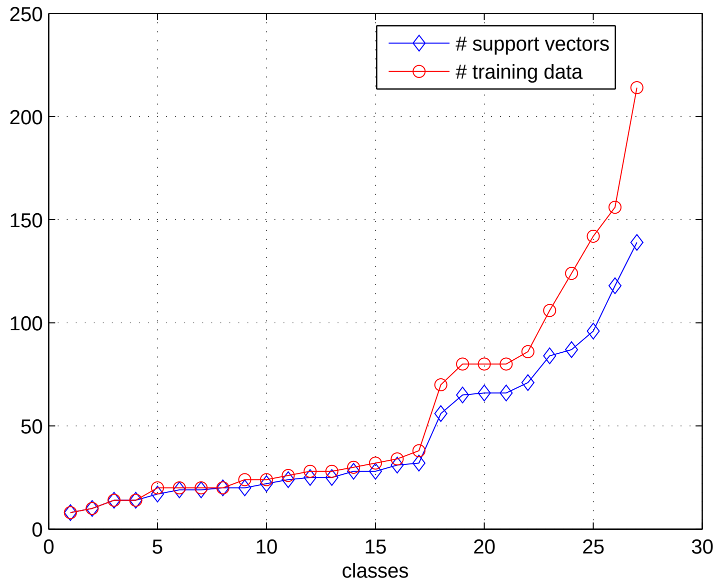

Figure 3: Comparison between the original size of the training sets and the number support vectors for the 18 classes considered; the sparsity of the solution is more noticeable for datasets with more than 20 entries.

be kept both for efficiency and performance reasons. The remaining experiments are performed with $k = 4$.

	Cardinality	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18
aircraft winged vehicle	135	0.53	0.65	0.69	0.73	0.76	0.76	0.78	0.79	0.81	0.84	0.84	0.85	0.86	0.88	0.89	0.9	0.97	1
animal biped human	78	0.45	0.64	0.77	0.86	0.91	0.92	0.92	0.92	0.92	0.95	0.95	0.97	0.97	0.97	0.97	0.97	0.99	1
plant	60	0.38	0.43	0.53	0.6	0.68	0.7	0.77	0.8	0.85	0.88	0.9	0.9	0.93	0.95	0.97	0.98	1	1
vehicle car	51	0.2	0.57	0.67	0.73	0.75	0.78	0.78	0.86	0.86	0.88	0.88	0.9	0.9	0.9	0.9	0.9	0.94	1
body part head	38	0.45	0.53	0.79	0.84	0.84	0.84	0.87	0.92	0.95	0.95	0.97	1	1	1	1	1	1	1
furniture seat	37	0.7	0.78	0.78	0.84	0.86	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.97	0.97	0.97	1	1	1
furniture table	35	0.46	0.57	0.6	0.71	0.77	0.86	0.91	0.91	0.94	0.94	0.94	0.94	0.94	0.97	0.97	0.97	0.97	1
sea vessel	26	0.62	0.65	0.73	0.77	0.77	0.77	0.77	0.81	0.85	0.88	0.88	0.88	0.88	0.92	1	1	1	1
liquid container	24	0.38	0.58	0.79	0.96	1	1	1	1	1	1	1	1	1	1	1	1	1	1
display device	24	0	0.46	0.46	0.63	0.67	0.67	0.79	0.88	0.88	0.92	0.92	0.92	0.92	0.96	0.96	1	1	1
animal underwater creature	23	0	0.13	0.3	0.57	0.61	0.61	0.61	0.65	0.7	0.74	0.74	0.74	0.74	0.83	0.96	1	1	1
aircraft helicopter	18	0.56	0.67	0.83	0.89	0.89	0.94	0.94	0.94	0.94	0.94	0.94	0.94	1	1	1	1	1	1
door	18	0	0.06	0.11	0.44	0.72	0.83	0.89	0.89	0.94	1	1	1	1	1	1	1	1	1
animal quadruped	17	0.47	0.47	0.59	0.65	0.71	0.82	0.82	0.82	0.82	0.82	0.94	0.94	1	1	1	1	1	1
furniture shelves	13	0.23	0.54	0.69	0.92	0.92	0.92	1	1	1	1	1	1	1	1	1	1	1	1
city	10	0.1	0.3	0.5	0.6	0.6	0.8	0.8	0.9	1	1	1	1	1	1	1	1	1	1
lamp	8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
hat	6	0	0	0	0.17	0.17	0.33	0.33	0.33	0.5	0.5	0.67	0.83	1	1	1	1	1	1

Table 2: For each class, we report the percentage of test shapes correctly classified by the ranking of the SVM output. The cardinality of the class of the test shapes is reported in the second column. The R1 column shows for each class the percentage of test shapes whose first nearest neighbour belongs to the correct class.

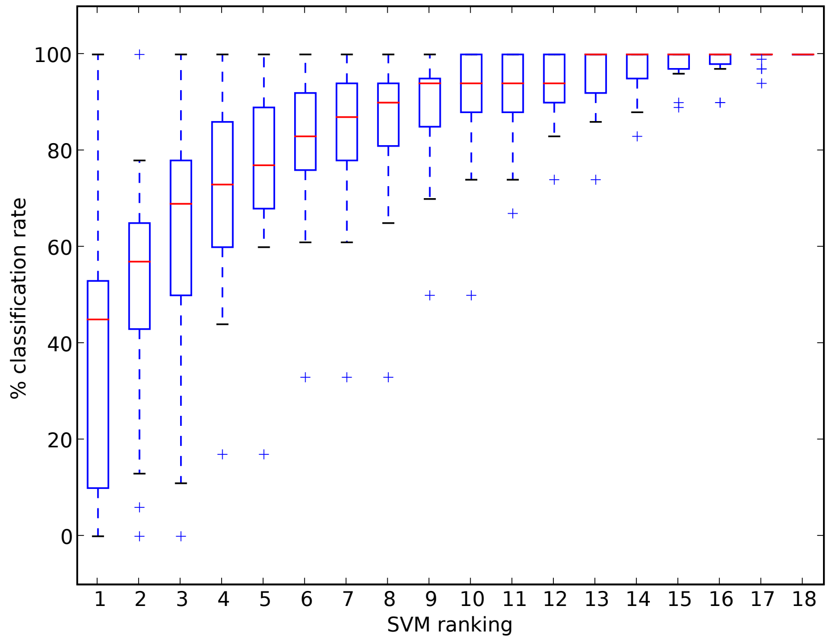

Figure 4: Performance of the SVM-based filter as the number of retained classes varies. The performance is evaluated as the percentage of 3D shapes whose true class was among the k classes ranked first by the SVM. The boxplots show the median values and distributions across the various classes (see text).

By analysing the different classes it is possible to notice that the performances of direct retrieval are comparable or above our filtering method for those classes which have very small training sets (less than 10 elements), while SVM filtering is a clear advantage when the training set has more than 40 elements.

Table 4 reports the average retrieval results over all the classes, in the case of direct retrieval and SVM filtering with $k = 4$. Notice that direct retrieval represents the results obtained with the original work by Osada *et al.* [27] on our datasets. The advantage of our approach is evident. The results presented in the original work [27] are relative to a different (and smaller) dataset and the performances are described with different indicators, therefore comparison is more complex. We conclude reporting that in [16] the following results, obtained with Osada approach on a different subset of the PSB, are reported: $FT = 33\%$, $ST = 47\%$, $NN = 59\%$. The results seems to be superior to the ones that we obtain, the reasons may be due to the

		SVM filtering			Retrieval		
	Training class cardinality	FT	ST	NN	FT	ST	NN
aircraft winged vehicle	107	0.52	0.73	0.63	0.36	0.59	0.64
plant	78	0.26	0.48	0.45	0.17	0.28	0.47
animal biped human	71	0.49	0.76	0.74	0.46	0.69	0.73
vehicle car	62	0.36	0.65	0.47	0.27	0.43	0.39
furniture table	43	0.29	0.55	0.31	0.22	0.38	0.26
furniture seat	40	0.25	0.47	0.43	0.14	0.24	0.40
body part head	40	0.37	0.71	0.5	0.29	0.48	0.58
liquid container	35	0.17	0.42	0.25	0.12	0.24	0.17
sea vessel	19	0.28	0.45	0.46	0.21	0.26	0.46
aircraft helicopter	17	0.15	0.32	0.56	0.11	0.17	0.44
display device	16	0.11	0.26	0.17	0.08	0.19	0.04
lamp	14	0.5	0.76	0.75	0.21	0.30	0.13
animal quadruped	14	0.16	0.2	0.35	0.15	0.18	0.41
furniture shelves	13	0.11	0.18	0.15	0.04	0.12	0.15
animal underwater creature	12	0.11	0.18	0.04	0.07	0.15	0.04
hat	10	0	0.02	0	0.02	0.02	0.17
city	10	0.1	0.15	0.3	0.10	0.12	0.30
door	10	0.06	0.12	0	0.03	0.10	0.00

Table 3: Performance of retrieval for the different classes, with and without SVM-filtering (left and right, respectively). We show the results of first tier (FT), second tier (ST) and nearest neighbors (NN).

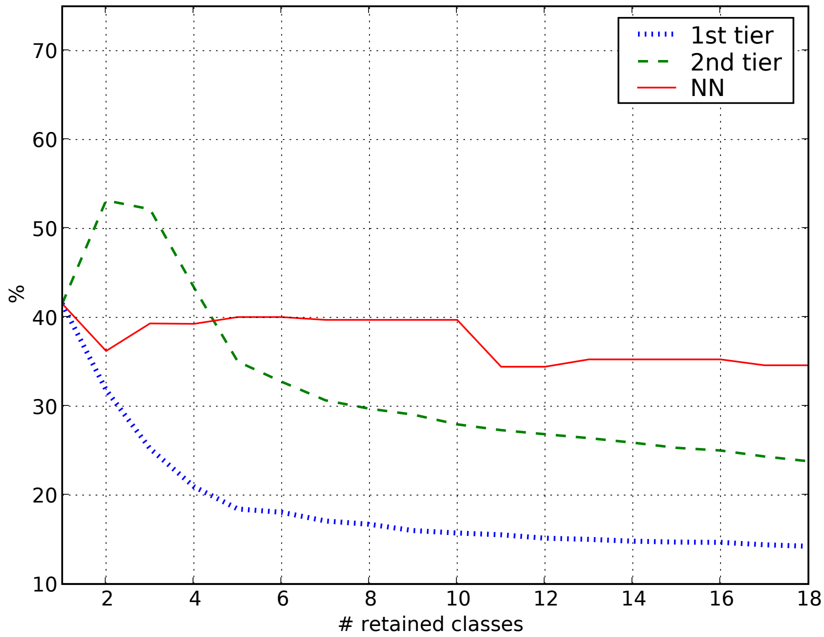

Figure 5: The behaviour of our performance evaluation measures as the number k of retained classes before retrieval grows.

	First Tier	Second Tier	Nearest Neighbour
SVM filtering (4 classes)	33%	55%	47%
our implementation of Osada et al [27]	24%	39%	44%

Table 4: Results with our approach and Osada et al [27] computed by us.

different characteristics of the selected classes and to the fact that apparently only stable subsets of shapes per each class are kept.

6 Conclusion

We have proposed a method based on SVM for filtering the relevant classes in a 3D object database prior to shape retrieval. SVM classifiers are built for all classes of object in a database, and just the k most relevant classes or a query object are searched to answer a query by similarity. We have shown that not only our method can improve performance by pruning the dataset to be searched, but also results are better with respect to those obtained with exhaustive search, using the same shape descriptor, according to the usual evaluation methods (nearest neighbor, first tier and second tier).

The overall performance of the shape descriptor used in this initial work is beated buy others at the state-of-the-art. Therefore, we plan to test our approach also with other, more performant, descriptors. As already mentioned, our filtering is somehow orthogonal with respect to the descriptor used. However, the quality of a descriptor may highly influence the performance of classification through SVM. If some other descriptor could give us a better performance in classification, we could restrict search to an even smaller number k of classes, thus improving performance further.

In our future work, we also plan to test the use of other kernels to build the classifiers. Also, a more careful tuning of SVM offset to contrast false negatives would allow us to minimize the number of misses.

Aknowledgement

The work is supported by the PRIN project 3SHIRT (3-dimensional SHape Indexing and Retrieval Techniques) funded by the Italian Ministry of Research and Education. The authors thank A. Repetto for the development of the tool for computing the statistical descriptors, and N. Rebagliati for the normalized multiway cut algorithm.

References

[1] Aim@shape shape repository. http://shapes.aim-at-shape.net.

[2] C. B. Akgul, B. Sankur, Y. Yemez, and F. Schmitt. Similarity score fusion by ranking risk minimization for 3d object retrieval. In *Eurographics workshop on 3D object retrieval*, 2008.

[3] Zafer Barutcuoglu and Christopher DeCoro. Hierarchical shape classification using bayesian aggregation. In *SMI '06: Proceedings of the IEEE International Conference on Shape*

Modeling and Applications 2006, page 44, Washington, DC, USA, 2006. IEEE Computer Society.

[4] Mirela Ben-Chen and Craig Gotsman. Characterizing shape using conformal factors. In *Proceedings of the Eurographics 2008 Workshop on 3D Object Retrieval*, pages 1–8, Aire-la-Ville, Switzerland, 2008. Eurographics association.

[5] LENG Biao, QIN Zheng, and LI Li-qun. Support vector machine active learning for 3d model retrieval. *Journal of Zhejiang University SCIENCE A*, 8(12):1953–1961, 2007.

[6] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] Benjamin Bustos, Daniel A. Keim, Dietmar Saupe, Tobias Schreck, and Dejan V. Vranić. Feature-based similarity search in 3d object databases. *ACM Comput. Surv.*, 37(4):345–387, 2005.

[8] A. Caponnetto and E. De Vito. Optimal rates for regularized least-squares algorithm. *Found. Comput. Math.*, 2006. In Press, DOI 10.1007/s10208-006-0196-8, Online August 2006.

[9] Umberto Castellani, Marco Cristani, Simone Fantoni, and Vittorio Murino. Sparse points matching by combining 3d mesh saliency with statistical descriptors. *Computer Graphics Forum*, 27(2):??–??, 2008.

[10] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.

[11] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[12] P. Daras, D. Zarpalas, A. Axenopoulos, D. Tzovaras, and M.G. Strintzis. Three-dimensional shape-structure comparison method for protein classification. *IEEE/ACM transactions on Computational Biology and Bioinformatics*, 3(3):193–207, July 2006.

[13] Michael Elad, Ayellet Tal, and Sigal Ar. Content based retrieval of vrml objects: an iterative and interactive approach. In *Proceedings of the sixth Eurographics workshop on*

Multimedia 2001, pages 107–118, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[14] Thomas Funkhouser, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[15] Ran Gal and Daniel Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, 2006.

[16] Ran Gal, Ariel Shamir, and Daniel Cohen-Or. Pose-oblivious shape signature. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):261–271, 2007.

[17] L. Lo Gerfo, L. Rosasco, F. Odone, E. De Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20:1873–1897, 2008.

[18] S. Hou, K. Lou, and K. Ramani. SVM-based semantic clustering and retrieval of a 3d model database. *Computer Aided Design and Applications*, 2(1-4):155–164, 2005.

[19] Natraj Iyer, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37:509–530, 2005.

[20] Tony Jebara, Risi Imre Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.

[21] T. Joachims. Making large-scale svm learning practical, 1999.

[22] Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(5):433–449, 1999.

[23] Chang Ha Lee, Amitabh Varshney, and David W. Jacobs. Mesh saliency. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 659–666, New York, NY, USA, 2005. ACM.

[24] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The digital michelangelo project: 3d scanning of large statues. In *SIGGRAPH '00:*

Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pages 131–144, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[25] M. Novotni, G.J. Park, R. Wessel, and R. Klein. Evaluation of kernel based methods for relevance feedback in 3d shape retrieval. In *Proceedings 4th Int. Workshop on Content-based Multimedia Indexing*, Riga, Latvia, 2005.

[26] Francesca Odone, Annalisa Barla, and Alessandro Verri. Building kernels from binary strings for image matching. *IEEE Transactions on Image Processing*, 14(2):169–180, 2005.

[27] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *ACM Trans. Graph.*, 21(4):807–832, 2002.

[28] M. Ouhyoung, D. Y. Chen, H. Yeh, X.P.Tian, Y.T.Shen, and W. C. Luo. 3d model retrieval system. http://3d.csie.ntu.edu.tw/ dynamic/.

[29] Rcsb protein databank. http://www.rcsb.org/pdb/home/home.do.

[30] Ryan Rifkin, Gene Yeo, and Tomaso Poggio. Regularized least-squares classification.

[31] Philip Shilane and Thomas Funkhouser. Distinctive regions of 3d surfaces. *ACM Trans. Graph.*, 26(2):7, 2007.

[32] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *SMI '04: Proceedings of the Shape Modeling International 2004*, pages 167–178, Washington, DC, USA, 2004. IEEE Computer Society.

[33] J.W.H. Tangelder and R.C.Veltkamp. A survey of content based 3d shape retrieval methods. In *Proceedings of Shape Modeling International*, 2004.

[34] J. Shawe Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[35] A.N. Tikhonov and V.Y. Arsenin. *Solutions of Ill Posed Problems.* W. H. Winston, Washington, D.C., 1977.

[36] V.N. Vapnik. *Statistical learning theory. Adaptive and learning system for signal processing, communication and control.* John Wiley and Sons Inc., 1998.

[37] D. V. Vranic. Desire: a composite 3d-shape descriptor. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Amsterdam, NL, July 2005.

[38] G. Wahba. *Spline models for observational data*, volume 59. SIAM, Philadelphia, PA, 1990.

[39] Dong Xu and Hua Li. 3d shape retrieval integrated with classification information. In *Proceedings of the Fourth International Conference on Image and Graphics*, pages 774–779, Washington, DC, USA, 2007. IEEE Computer Society.