Computer Aided Geometric Design 35–36 (2015) 192–205

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Iterative 3D shape classification by online metric learning

Mofei Song, Zhengxing Sun *, Kai Liu, Xufeng Lang

State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, PR China

ARTICLE INFO

Article history:
Available online 24 March 2015

Keywords:
3D shape categorization
Iterative classification
Online metric learning
Scalable collection

ABSTRACT

To provide a scalable and flexible tool for 3D shape classification, this paper proposes an iterative 3D shape classification method by integrating incrementally updating, online learning and user intervention. It classifies the collection of 3D shapes iteratively by combing unsupervised clustering with online metric learning, and puts the user intervention into the loop of each iteration. The features of our method lie in three aspects. Firstly, it discovers the potential groups in the collection group by group without any pre-labeled samples or any pre-trained classifiers. Secondly, the users can get the desirable classes by directly confirming the required members of each group and annotate them with any free label to suit for different applications. Finally, the scalable collection can be handled dynamically and efficiently by our incrementally updating mechanism based on the online metric learning. The experimental results prove the effectiveness of the proposed method.

1. Introduction

3D shape classification is a fundamental problem in digital geometry processing, which is important for browsing, managing and organizing large 3D shape collections. Most existing works (Csakany and Wallace, 2003; Biasotti et al., 2006; Huber et al., 2004; Tabia et al., 2011, 2013; Marini et al., 2011; Barra and Biasotti, 2014; Qin et al., 2014) have focused on learning a classifier by a large number of labeled samples to classify a given shape. As the training set utterly determines the scope of categories and the trained classifier is hardly updated, almost all of them cannot generalize well to unknown object category and multiple classification criterions to fit the needs of the diversity of taxonomies and the different application requirements. Moreover, the practical classification task often faces the dynamic collection as an explosive amount of 3D shapes in these days. Thus, it remains an open problem to provide a scalable and flexible tool for 3D shape classification.

From our knowledge, there should be at least three aspects of requirements for a scalable and flexible 3D shape classification tool. *Firstly, no pre-labeled samples and pre-trained classifiers are required*. The unsupervised clustering method (Everitt et al., 2011) is intuitively one of the best choices as used in the 3D shape classification (Donamukkala et al., 2005) and co-segmentation (Wang et al., 2012). Unfortunately, the class number of the data sets, which affects the clustering result heavily, must be the default known or specified by the user for most of the clustering methods, and the data sets are often classified in a one-pass batch process without any other user intervention. Obviously, 3D shape classification done only by the unsupervised clustering makes it inflexible. *Secondly, the classification must be done online and interactively.* In general, the user is accustomed to put some relatively similar data together into a group from the collection according to different applications and assign it with a label to distinguish it from other groups, rather than classify and label the data one by

* Corresponding author.
E-mail address: szx@nju.edu.cn (Z. Sun).

http://dx.doi.org/10.1016/j.cagd.2015.03.009
0167-8396/© 2015 Elsevier B.V. All rights reserved.

one. Therefore, the task of classification is preferably performed group by group, instead of one by one as the traditional pre-trained classifier does. Meanwhile, a useful classification tool may work in an online circular manner to discover recurrently the potential groups in the collection, and put the user in the loop of classification to interactively identify the member of each group and label each of groups. Though such a framework has recently been proposed for annotation of a large set of images (Galleguillos et al., 2014), how to classify 3D shape effectively in such a manner remains unopened. *Thirdly, the scalable collections of 3D shape must be processed dynamically.* The incremental learning and updating strategies have been widely used in the traditional classifier research, such as image classification (Ristin et al., 2014), object tracking (Collins et al., 2005) and shape segmentation (Zhang et al., 2015). However, the way of classifying the collection of 3D shapes brings three aspects of new dynamic requirements: the size of the shape set might be too big to be handled at one time, the number of the remaining shapes after several recurrent grouping is too small to be categorized in the current context until enough shapes are added, and the new 3D shapes may be added dynamically to the collection. All of three requirements demand to design the novel incremental and updating mechanism of classification to avoid regrouping all of new and old shapes. In brief, to make the 3D shape classification of the collection be scalable and flexible, online learning, incremental updating and user intervention may be integrated into a framework without any pre-labeling and pre-training. Though there are many solutions for each of them individually within this framework (Galleguillos et al., 2014; Ristin et al., 2014), how they can work as a cohesive whole is still expected to be solved.

In this paper, we propose an iterative 3D shape classification method to fit the needs of the diversity of classifications. It classifies the collection of 3D shapes iteratively by combing unsupervised clustering with online metric learning, and puts the user intervention into the loop of each iteration. The spectral clustering, one of the conventional unsupervised clustering, is used to generate the clusters of the unclassified shapes according to the optimized similarity metric, which measures the similarity between the shapes, and select the most prominent cluster as the candidate set of user judgement in each iteration. Online metric learning is adapted to optimize the similarity metric through minimizing ranking losses, which are defined by the newly classified shapes from user intervention during each iteration. The user is allowed to perform two types of interactions successively for the current cluster: (1) judge whether the shapes in the group are the positives or the negatives and remove the negatives back to the remainders of the collection, and (2) assign or create the group of shapes with a label of an existing or a novel category. After each iteration, all the remaining shapes will be clustered according to the optimized metric in the next iteration. In summary, the features of our method lies in three aspects.

1. The collection of 3D shapes can be classified group by group and iteratively. The unsupervised clustering, online metric learning and user intervention are integrated into a framework and work as a cohesive whole. It discovers the potential groups in the collection repetitively without any pre-labeled samples or pre-trained classifiers, and let user intervention to ensure the composition of the group and create its label, instead of classifying all the shapes at once without any user intervention as the one-pass batch process does.
2. The users can classify the 3D shape collection flexibly and freely. They can get the desirable shape classes by directly confirming the required members of each group and annotate them with any free label to suit for different applications. They can also create the categories of the collection online without any prior determinations about the scope and number of categories. These overcome the inflexibility and limitation of the traditional pre-trained classifier.
3. The scalable collection can be handled dynamically and efficiently. Firstly, the large scaled data set can be classified as a block processing, rather than processing at one time. Secondly, when several shapes are too small to construct a certain category, they are allowed to be kept unclassified until enough similar shapes are added. Finally, the newly added dataset can be processed incrementally without performing the classification on the whole set repeatedly.

2. Related work

Through the state-of-the-art, we can find mainly several families of related work: 3D shape comparison, 3D shape classification, clustering and metric learning. While our algorithm uses some of the same underlying techniques, none of these methods alone are sufficient to meet the goals of this paper.

2.1. 3D shape comparison

To measure the similarity between a pair of 3D models, many shape distance measure methods have been proposed, such as feature-based descriptors (Bronstein et al., 2011; Li et al., 2014) and graph-based descriptors (Barra and Biasotti, 2013). And these methods are widely used for shape retrieval (Tangelder and Veltkamp, 2008; Biasotti et al., 2014). However, any single distance measure among the shapes will not be sufficiently accurate to induce a reliable metric. So in our work, we adopt some feature-based distances to construct the initial similarity metric, and optimize the metric by an online metric learning method.

2.2. 3D shape classification

Most existing shape classification methods usually use the labeled model set as the training data, and train a classifier based on the supervised learning methods, such as nearest neighbor classifier (Csakany and Wallace, 2003; Donamukkala et al., 2005; Biasotti et al., 2006), Bayesian classifier (Huber et al., 2004), SVM (Marini et al., 2011; Barra and Biasotti, 2014), belief function (Tabia et al., 2013) and deep neural network classifier (Qin et al., 2014). A recent work presented by Huang et al. (2013b) uses a semi-supervised method for fine-grained 3D shape classification with several pre-labeled samples. However, the drawback of these methods is that they require a large amount of labeled examples. A few works have addressed this issue without any assistance of the pre-labeled models. Donamukkala et al. (2005) use an agglomerative clustering method to classify a set of vehicles into several disjoint groups. But this method is limited to specific datasets. For a heterogeneous collection of shapes, Huang et al. (2013a) use a quartet analysis method to generate a categorization tree by an optimized metric. But their categorization tree is hierarchical for interactive shapes exploration, which is different from the goal of our method. Overall, all existing unsupervised methods have generally focused on grouping all the models as a one-pass "batch" process, and their categorization result is difficult to control. Besides, when the shape set is expanded, they need performing the analysis method over a whole pipeline. In contrast, our iterative method classifies the models group by group more flexibly, and handles the scalable model set more efficiently.

2.3. Clustering

One of the best practiced means of classifying a set of elements is clustering analysis (Everitt et al., 2011). Unsupervised clustering methods have been used in some geometry processing tasks, such as 3D shape classification (Donamukkala et al., 2005) and co-analysis of shape sets (Wang et al., 2012). Most clustering methods generate all the disjoint groups from the data set at once. Unfortunately, the clustering result heavily depends on the similarity metric and the given number of clusters, which are difficult to be determined prior. In general, the set of potential categories is effectively unbounded, and may grow over time. Thus, several image annotation works (Galleguillos et al., 2014; Lee and Grauman, 2011) interleave unsupervised clustering and metric learning iteratively to generate the sets of similar images group by group, and put the user into the loop of the online processing. Our framework is inspired by these works, but our method incorporates the online metric learning method to optimize the metric more efficiently without re-training all the labeled samples from scratch.

2.4. Metric learning

The quality of any clustering method will ultimately depend upon the similarity metric between a pair of elements. Because the shape distance measure is often unreliable, it is less effective to directly cluster the collection of shapes using the native distance measure. Metric learning (Kulis, 2013) is one important technique that learns a distance function tuned to a particular task, which has been used in image retrieval (Kulis et al., 2009), category discovery (Galleguillos et al., 2014), face recognition (Guillaumin et al., 2009) and 3D model segmentation (Wang et al., 2012). Most metric learning methods (Bar-Hillel et al., 2005; Xing et al., 2003) often adopt batch machine learning approaches. However, the metric has to be re-trained from scratch whenever there is new training data. In recent years, some metric learning studies have explored online learning techniques to tackle the tasks (Shalev-Shwartz et al., 2004; Chechik et al., 2010). From existing methods, we choose the online algorithm for scalable image similarity learning method (OASIS) (Chechik et al., 2010), which is an online dual approach based on the passive–aggressive algorithm. The OASIS method is used to improve the efficiency of image retrieval in their work, and we use it to handle the scalable shape collection effectively. To the best of our knowledge, this is the first time to introduce online metric learning into 3D shape classification.

3. Overview

Depending on the size and increasing of 3D shape collection, there are four typical scenarios of classification. Scenario 1 is the basic scenario, i.e., the initial collection has the suitable size for classification at one time. Scenario 2 indicates the size of the initial collection, which is too large to be handled at one time. Scenario 3 introduces the incremental problem of the collection, which means an incremental collection is provided for classification after several collections have already been classified. Scenario 4 is the version that the user cannot determine the categories of the remaining shapes after several recurrent grouping, thus some shapes are allowed to be kept unclassified in the current context. To achieve the consistent classification result in all the above scenarios, our framework incorporates three main modules (as shown in Fig. 1): incrementally updating, online learning and user intervention. The incrementally updating module handles the loading of the shape collection and the updating record from various scenarios. The online learning module takes the loaded collection as input, and interleaves the candidate cluster creation with the similarity function optimization iteratively. The user intervention module takes the candidate set as input in each iteration, and asks the user to perform member selection and group labeling successively to create a required group of shapes.

Incrementally updating. The main goal is to prepare the shape collection and the similarity function for the online learning module, and this information should be extracted appropriately from various scenarios. According to the four scenarios, the input collection can be divided into two classes: the initial collection and the incremental collection. If no classification task has been done, the collection is the initial collection, otherwise, it is the incremental collection. The incremental collection, which is loaded by the incremental loader, is often composed of two parts: the new collection and the suspended shapes. Both the initial collection and the new collection are provided by the user. And the suspended shapes are the

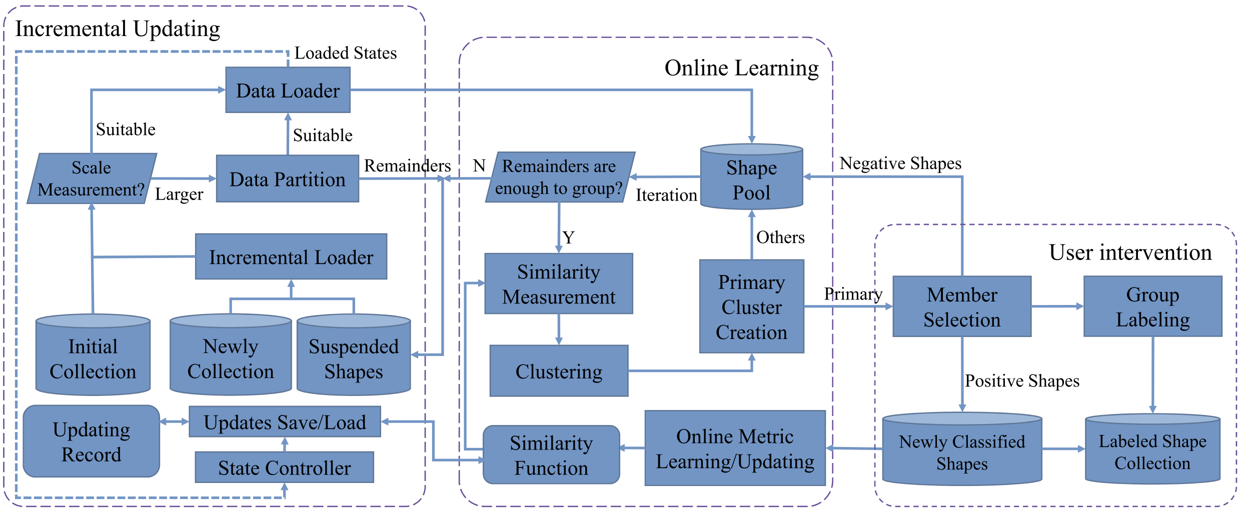


Fig. 1. The framework of our method.

remainder shapes that return from the incrementally updating module or the online learning module. After obtaining the input collection, whether initial or incremental, the size of the collection should be judged. If the size is smaller than a certain threshold $\mathcal{S}$ (the default is 2000), the collection of shapes can be imported into the shape pool by the data loader for online learning. Otherwise, the collection is firstly partitioned into several blocks, and the size of each block is no more than $\mathcal{S}$. Then one of the blocks is selected randomly to be imported into the shape pool by the data loader, while the remaining blocks are returned as the suspended shapes. The processing of the initial and incremental collections has only one difference: when the incremental collection is imported into the shape pool, the state controller will receive a message of a loaded state, which triggers to load the updating record into the online learning module. Specially, when some shapes are kept unclassified in scenario 4, these shapes are return as the suspended shapes, meanwhile, the similarity function will be saved into the updating record.

Online learning. The module includes the core process of the whole pipeline, which interleaves unsupervised clustering and metric optimization iteratively. At each iteration, it proceeds as the following four steps: first, the similarity between each shape of the collection is measured by the similarity function. Second, the collection is grouped by spectral clustering according to the similarity measurement. Third, the primary cluster with the smallest average intra-cluster distance is selected for user intervention from the clustering results, while other clusters are returned into the shape pool. In general, smaller intra-cluster distance implies higher purity. Thus, compared with other clusters, the user can determine the majority category of the primary cluster easily and remove the negatives from this cluster with fewer number of operations. Finally, the similarity function is trained/updated by an online metric learning method according to some constraints, which are defined by the newly labeled shapes. The process repeats until there are not enough shapes for grouping. The algorithm details of this module will be introduced in Section 4.

User intervention. The user intervention is incorporated into each iteration of online learning. The user is asked to perform two types of interactions successively for the primary cluster: first, the user can judge each shape within the cluster either (a) mark the positive instances that belong to the major category of the cluster; or (b) mark the negative instances that do not belong to the majority category. In a word, the user always chooses to perform the fewest assignments in order to create the largest group of instances corresponding to a single category. Second, the user can label the group by a novel or an existing category tag. The newly classified shapes can be saved into the labeled shape collection.

4. Algorithm

In this section, we will describe the whole online learning module in details.

4.1. Model representation

Note that we cannot know a priori which descriptors will be discriminative for 3D shape classification. We therefore opt to include many different descriptors, and represent each shape as a collection of n feature descriptors $\{\chi_i\}_{i=1}^{n}$. In our setting, we employ four descriptors: the Light Field Descriptor (LFD) (Chen et al., 2003), Cord and Angle Histograms (CAH) (Paquet and Rioux, 1997), Shape Distribution (SD) (Osada et al., 2002) and 3D complex function fast Fourier transform descriptor (cFFFTD) (Vranic and Saupe, 2002). In order to integrate heterogeneous descriptors effectively, we first apply multi-dimensional scaling (MDS) to reduce the dimension of each descriptor, and then combine these lower-dimensional descriptors linearly to compute the similarity between a pair of shapes with an optimized metric.

Each descriptor χ_i defines a high-dimensional feature space. To improve the efficiency of shape classification, multi-dimensional scaling is first used to reduce the initial feature spaces into to a lower-dimensional Euclidean space. Suppose the input shape collection in the shape pool is denoted by $\mathcal{M} = \{M_i, i = 1, \ldots, m\}$. Given the distance matrix $\mathcal{D}$ of the input collection $\mathcal{M}$ by the descriptor χ, MDS generates m data points $\phi^1, \phi^2, \ldots, \phi^m$ in a lower-dimensional space, where these data points are found as minimizer of the following cost function

$$\min_{\phi^1,\ldots,\phi^m} \sum_{j<k} (\|\phi^j - \phi^k\| - \mathcal{D}_{jk})^2 \tag{1}$$

where, $\mathcal{D}_{jk} = d(\chi^j, \chi^k)$ and $d(\cdot,\cdot)$ is the distance between the two feature descriptors. Different descriptors are compared by different distance functions, for example, LFD is compared by inner-distance (Ling and Jacobs, 2007), CAH and SD are compared by chi-square distance. Accordingly, we can obtain n lower-dimensional spaces corresponding to the n descriptors, which are denoted by $\{\phi_1, \ldots, \phi_n\}$.

Then the similarity between the two shapes p and q can be measured by a linear combination of all the lower-dimensional features $\{\phi_1, \ldots, \phi_n\}$

$$f(p,q) = \sum_{i=1}^{n} \frac{1}{\sigma_i} S_{W_i}(\phi_i(p), \phi_i(q)) \tag{2}$$

where, σ_i is set as the half of the maximum pair-wise similarity between the shapes by using the ith low-dimensional feature ϕ_i, automatically estimated from the input collection $\mathcal{M}$. And W_i is a positive semi-definite matrix, which defines the similarity between $\phi_i(p)$ and $\phi_i(q)$ by a parametric similarity function that has a bi-linear form

$$S_{W_i}(\phi_i(p), \phi_i(q)) = \phi_i(p)^T W_i \phi_i(q) \tag{3}$$

The matrix W_i will be optimized to make the similar shapes more closer during the online learning process. Initially, all W_i are identity matrix.

4.2. Clustering and primary cluster creation

To create the primary cluster for user judgement, we perform spectral clustering (Ng et al., 2001) to generate several groups from the collection in the shape pool. Our decision to use spectral clustering is motivated mainly by the non-linear feature of the descriptors. The spectral clustering method is performed with the aid of spectral embedding, which take the non-linear and anisotropic structures in the data and unfold them into a new space, so that the similarities between data points are translated into geometric proximity. Accordingly, we first embed the shapes into a new space using the eigenvectors of the affinity matrix. Then the normal K-means clustering can succeed in the embedded space by simply considering Euclidean distances. Finally, the primary cluster is selected as the candidate set of user judgement.

Embedding. We first use the similarity function $f(\cdot,\cdot)$ to compute the affinity matrix $\mathcal{W}$ of the input collection $\mathcal{M}$. By defining a diagonal matrix $D_{i,i} = \sum_j \mathcal{W}_{i,j}$, we can construct a matrix $L = D^{-1/2}\mathcal{W}D^{1/2}$. We then compute the eigendecomposition of the matrix L, and extract the largest k eigenvectors $x_1, x_2, \ldots, x_k$ to form a matrix $X = [x_1 x_2 \ldots x_k]$. Finally, each row of the matrix X is normalized to be unit length, and the shape M_i can be represented by the ith row of matrix X.

Clustering. After obtaining the embedded space, the pairwise distance $d(p,q)$ between two shapes p and q can be computed as the Euclidean distances between the corresponding vectors of matrix X. A simple K-means clustering algorithm (Kanungo et al., 2002) is used to cluster the collection $\mathcal{M}$ into several groups. The number of clusters k is equal to the column number of the matrix X.

Primary cluster creation. After clustering, we select the primary cluster according to the average intra-cluster distance and the size of the cluster. On one hand, the size of the cluster should be larger than a certain threshold δ (the default value is 20). On the other hand, the average intra-cluster distance should be small. Accordingly, the selection criterion of the primary cluster can be express by:

$$\begin{aligned} d_{intra}(c) &= \frac{1}{|c| \cdot (|c|-1)} \sum_{\substack{p \neq q \\ p,q \in c}} d(p,q) \\ c^* &\leftarrow \mathop{\arg\min}_{c,\ \text{s.t.}|c|>\delta} d_{intra}(c) \end{aligned} \tag{4}$$

where, $d(p,q)$ is the distance between the two shapes in the embedded space, which is used in the previous clustering process. We expect that the larger and tighter cluster can be selected by the above criterion. If the selected cluster is large, many shapes may be labeled in each iteration. When it is tight, it may have the high purity, thus the majority of the shapes within the cluster will be labeled. The selected cluster is provided for user judgement, and each instance within the cluster is explicitly confirmed by the user. According to user intervention, a group of newly classified shapes can be generated from the primary cluster.

4.3. Online metric learning

After creating the group of newly classified shapes, we can use these shapes to optimize the similarity metric by an online metric learning method (Chechik et al., 2010). As the metric is optimized with the accumulation of the classified shapes progressively, the shapes belonging to the same category can be drawn closer. Consequently,compared with using the native metric, we expect to generate larger, tighter and purer clusters with the optimized metric. In this section, we first create a constraint set from user intervention, then OASIS (Chechik et al., 2010) is used to optimize the similarity function for satisfying these constraints.

To use OASIS to optimize the metric for our task, we first create a set of training triplets (p, p^+, p^-). Given the primary cluster, the user identifies two shape sets: the positive set $\mathcal{P}_f$ and the negative set $\mathcal{P}_u$. In the classification task, two samples are relative if two samples belong to the same category. Otherwise, the two samples are irrelative. Thus given any two shapes p and p^+ in $\mathcal{P}_f$ and any shape p^- in $\mathcal{P}_u$, the similarity between p and p^+ should be larger than that between p and p^-. Accordingly, a set $\mathcal{T}$ of training triplets can be generated from two sets $\mathcal{P}_f$ and $\mathcal{P}_u$:

$$\mathcal{T} = \{(p, p^+, p^-), \forall p, p^+ \in \mathcal{P}_f, p^- \in \mathcal{P}_u\} \tag{5}$$

Then OASIS receives the triplet from the $\mathcal{T}$ one by one, and updates the matrices W_i for each descriptor incrementally. Specially, the goal of this method is to learn the similarity function $S_W(\cdot,\cdot)$ (Eq. (3)) that assigns higher similarity scores to pairs of more relevant samples, i.e.

$$\begin{aligned} &S_W(p, p^+) > S_W(p, p^-), \forall p, p^+, p^- \in P \\ &\text{such that } r(p, p^+) > r(p, p^-) \end{aligned} \tag{6}$$

where $r(\cdot,\cdot)$ reflects the relevance between two samples.

Given a set of triplets $\{p_t, p_t^+, p_t^-\}$, OASIS finds a parametric similarity function S_W such that all triplets satisfy the following constraints:

$$S_W(p_t, p_t^+) > S_W(p_t, p_t^-) + 1 \tag{7}$$

The hinge loss function for each triplet is defined by

$$l_W(p_t, p_t^+, p_t^-) = \max\{0, 1 - S_W(p_t, p_t^+) + S_W(p_t, p_t^-)\} \tag{8}$$

In order to minimize this loss, the Passive–Aggressive algorithm (Crammer et al., 2006) is used iteratively over triplets to optimize W. The online optimization problem is formulated as

$$W^t = \arg\min_W \frac{1}{2}\|W - W^{t-1}\|^2_{Fro} + C l_W(p_t, p_t^+, p_t^-) \tag{9}$$

where $\|\cdot\|$ is the Frobenius norm. At each iteration, W^t is selected to optimize a trade-off between remaining close to the previous W^{t-1} and minimizing the loss on the current triplet $\{p_t, p_t^+, p_t^-\}$. The parameter C controls this trade-off. Accordingly, the online solution of updating W is

$$W^t = W^{t-1} + \tau_t V^t \tag{10}$$

where $\tau_t = \min\{C, \frac{l_{W^{t-1}}(p_t, p_t^+, p_t^-)}{\|V^t\|^2}\}$ and $V^t = p_t(p_t^+ - p_t^-)$. For more details of the optimization we refer the reader to Chechik et al. (2010).

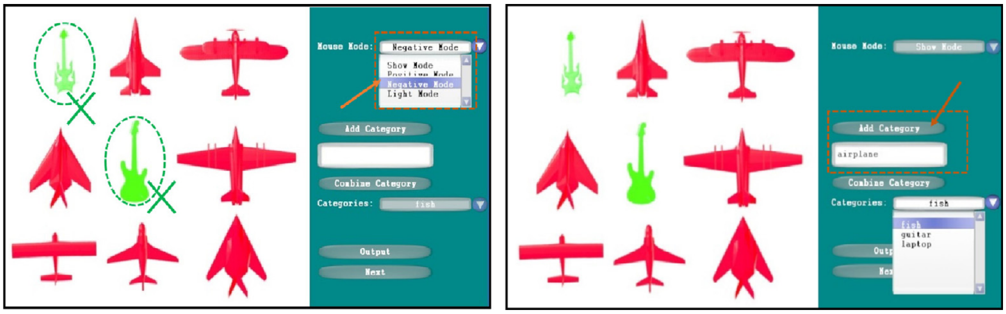
After updating, the similarity between the remaining shapes are computed according to Eq. (2) with the set of optimized matrices W_i.

4.4. Stop criterion

The iterative algorithm terminates when the distribution of the remaining shapes is sparse. It means that any two shapes are difficult to form a group. To measure the sparse degree of the remaining shape collection in the shape pool, we compute the uniform degree of the distances (Beyer et al., 1999) by

$$\zeta = \frac{MaxDis - MinDis}{MinDis} \tag{11}$$

where *MaxDis* and *MinDis* are the distance between the farthest and closest shapes respectively. The whole iteration stops, when ζ is lower than a certain threshold, or the size of the remaining shapes is lower than the cluster number k. The remaining shapes are returned as the suspended shapes.



(a) (b)

Fig. 2. User intervention. (a): Member selection; (b): Group labeling.

Table 1
The selected data set of SHREC2014.

Class	Size	Class	Size	Class	Size	Class	Size
train	175	armchair	333	bed	169	bookshelf	202
chair	632	keyboard	142	table	601	helicopter	162
guitar	185	airplane	314	fish	204	cell_phone	170
knife	152	laptop	188	tree	224	potted_plant	169
sword	245	cabinet	208	head	189	motorbike	172

Table 2
The four shape collections.

Dataset	PSB	SHREC2009	SHREC2011	SHREC2012
Size	1814	800	600	1200
Number of category	53	40	30	60

5. User intervention

During each iteration of online learning, the primary cluster is provided for the user. The user can perform two types of interactions successively to create a required group of newly labeled shapes: member selection and group labeling (Fig. 2).

Through the member selection interface, the users can judge each shape within the cluster in turn. They can either (a) mark the positive instances that belong to the major category of the cluster; or (b) mark the negative instances that do not belong to the majority category. The interaction is shown as Fig. 2(a). Since the number of positive instances is larger than half of the size of cluster, the user selects the 'negative mode', and removes the two negative instances from the cluster.

By the group labeling interface, the user can label the group with a novel or an existing category tag. The interaction is shown as Fig. 2(b). Since there is no 'airplane' label in the existing categories, the user creates an 'airplane' tag and labels the group of shapes with it. Meanwhile, the tag 'airplane' becomes an existing category. Our interface can achieve the shape classification and category discovery simultaneously. Besides, the label for every shape is explicitly confirmed by the user in at least one group, thus the classification result is very consistent with the user intention.

6. Results

In this section, we describe the experimental results and demonstrate the effectiveness and efficiency of our classification approach. The experiment environment is Intel(R) Core(TM) i5-2400 3.10 GHz with 8 GB of memory.

Data set. We evaluate our iterative classification method on the large scale comprehensive 3D shape set from the SHape Retrieval Contest 2014 (Li et al., 2014, 2015). The SHREC2014 collection is a complex database, which collects the relevant models from eight 3D object retrieval benchmarks. The SHREC2014 Large Scale Comprehensive Retrieval Track Benchmark has 8987 models, categorized into 171 classes. From the 171 classes, we choose 20 largest classes from the data set, and the size of each class is larger than 140. The selected classes and the size of each class are shown in Table 1. The benchmark gives the standard classification of these shapes. Besides, we also evaluate our method on four easier shape collection: Princeton Shape Benchmark (PSB) (Anon, 2004), SHREC2009 New Generic Shape Benchmark (SHREC2009) (Godil et al., 2009), SHREC2011 Non-rigid Track Benchmark (SHREC2011) (Lian et al., 2011), and SHREC2012 Generic Track Benchmark (SHREC2012) (Li et al., 2012). The size and the number of category of these four collection are shown in Table 2.

6.1. Evaluation of online metric learning

In order to evaluate the quality of the learned similarity space, we assess the purity of clusters generated from the data set. From the 20 classes of shapes, we choose 100 samples from each class to create the evaluation set, which includes

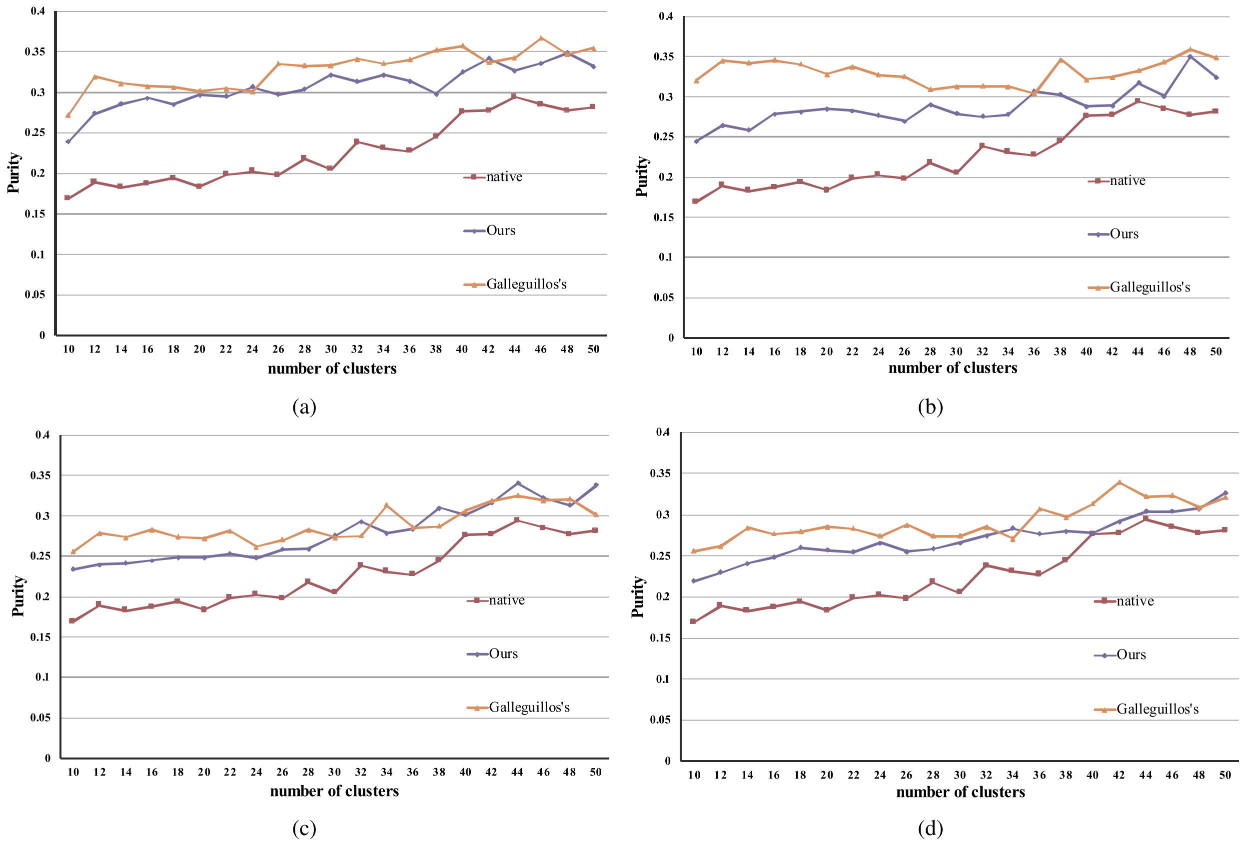


Fig. 3. The curve of the average cluster purity. (a): Training with one whole group; (b): Training with 4 classes; (c): Training with 8 classes; (12): Training with 12 classes.

2000 shapes. At each time, we divide the evaluation set into two sets: training set and testing set. Galleguillos's method (Galleguillos et al., 2014) and our method are used to train a distance function respectively. Then the learned distance and the native distance are used to cluster the testing set respectively. We vary the number of clusters from 10 to 50 (the step size is 2), and for each of them, compute the average purity of the clustering with the ground-truth labels, where the purity of a cluster B is computed by

$$purity(B) = \max_{l \in \mathcal{L}} \frac{|\{x' \in B \wedge l(x') = l\}|}{|B|} \tag{12}$$

where, $\mathcal{L}$ is the label set of all the categories, and $l(x')$ is the ground-truth label of sample x'.

We use 5-fold cross validation to measure whether the learned distance can improve the average purity of clusters. The evaluation set is divided into five groups, and each group has 20 classes of shapes while the size of each size is the same. Each time, we extract the training data from one group while the other four groups are used as the testing data. One group can be selected as the training set. Beside, we randomly select 4, 8 and 12 classes of shapes from one group to construct the training set. The goal is to measure whether the learned distance can improve the clustering result when the testing set has some unknown categories.

The resulting curves are shown in Fig. 3. From the figure, we can see that the average purity achieved by our method lies above that of the native space, and gets a close performance to the offline metric learning method (Galleguillos et al., 2014). When the testing data includes some unknown categories, the average purity also lies significantly above the native distance, especially when the number of clusters is smaller than 30 (Figs. 3(b)–3(d)). However, when the category number of the training set is small, the performance of our method is lower than that of the batch metric learning method. With the increase of the category number, the performance can converge to that of the batch metric learning method.

An important feature of our method is that it can update the metric online without training the metric from scratch. We also use 5-fold cross validation to measure the average purity of clustering result. In this experiment, we use one group as the test training set, and add the other four groups incrementally as the training set. We then measure the average purity and the training time once one group has been trained. Table 3 shows the evaluation results. From the table, we can see that the learned metric can be more and more accurate with the accumulation of shapes, while the incrementally training time is stable with respect to the same amount of the incremental training data.

Table 3
The average purity with the increase of the training set.

Number of training set	1	2	3	4
Average purity	23.1%	32.4%	42.5%	44.8%
Training time (s)	5.48	5.63	5.56	5.43

Table 4
The user effort for different classes of collections from SHREC2014. *NC* denotes the number of the clusters used in the clustering algorithm.

Classes	5			10			15			20		
Distance type	Native	Batch	Online	Native	Batch	Online	Native	Batch	Online	Native	Batch	Online
$NC = 10$	0.29	0.24	**0.23**	0.48	**0.42**	0.43	0.60	**0.52**	0.54	0.91	**0.79**	0.82
$NC = 20$	0.25	0.17	**0.16**	0.30	**0.27**	**0.27**	0.30	0.29	**0.27**	0.50	**0.47**	**0.47**
$NC = 30$	0.30	**0.23**	0.27	0.29	**0.23**	0.26	0.26	**0.22**	0.24	0.37	**0.30**	0.34

6.2. The iterative classification of initial collection

In this section, we evaluate the efficiency of the iterative classification method. We only consider the classification of the initial collection, i.e., the initial metric matrix is identity matrix, which has not been optimized. We assume the size of the collection is 2000, which can be loaded into the shape pool at once. The setting of the evaluation set is the same as the set in Section 6.1.

To evaluate whether our method can improve the annotation efficiency, we compute the average number of operation required for classifying one shape. In our method, the user effort includes two aspects: member selection and group labeling. The interaction amount is computed as follows: if the user performs to select a shape within the primary cluster, whether positive or negative, the interaction amount is added by one; if the user performs to label the group of shapes, whether novel or existing, the interaction amount is added by one. We denote the size of the collection is $\mathcal{M}$, and the class number of the collection is $\mathcal{C}$. The best case possible would to require only $\mathcal{C}$ group labeling operations, when the clustering result is perfect, i.e. every cluster is pure and includes all the shapes belong to the same category. The worst case possible would to be require $2\mathcal{M}$ operation, when there is only one positive instance in the given cluster at each iteration. Accordingly, the user is required to perform one member selection operation and one group labeling operation for each shape. We assume the interaction amount is α when $\mathcal{M}$ shapes have been labeled. Then we compute the number of operations μ required per shape by

$$\mu = \alpha/\mathcal{M} \tag{13}$$

Our experiment uses μ to represent the user effort for comparing the efficiency of the classification method. We compute the values of μ when the collection has different number of classes and the clustering is performed with various number of clusters. We compare three types of distances: the native distance, the distance learned by our online method, and the distance learned by the batch processing method (Galleguillos et al., 2014). Table 4 shows the statistical result. From the table, we can see that the learned distance outperforms the native distance across all number of cases (12.4% improvement of the mean value), especially when the number of clusters is small. Besides, the performance of our method gets a close performance to the batch processing method.

In practice, due to the unreliable quantitative distance measure, we cannot achieve the perfect clustering result (Huang et al., 2013a), so the value μ is difficult to approach the theoretical minimum $\mathcal{C}/\mathcal{M}$. From Table 4, we can see that hundreds of operations are still required when there are thousands of models. Consider the state-of-the-art classification algorithms based on supervised learning (Huber et al., 2004; Tabia et al., 2013; Marini et al., 2011; Barra and Biasotti, 2014), the accuracies of classification are from 54% to 96% depending on the number of categories and their generality. To correct the error of the classification result, the user is required to perform two operations per shape: (1) select the incorrect classified shape, and (2) correct the label of the shape. So the value μ of the supervised classification algorithm equals twice of the classification error (from 0.08 to 0.92). Thus, our method is comparable with the existing supervised algorithms in the aspect of interactive complexity (the mean of all the values μ is 0.358, which is equal to the user effort when the classification accuracy of the supervised algorithm is 82.1%). The above discussion shows that our method provides a potentially useful solution of 3D shape classification.

To show the scalability of our method, we also evaluate the efficiency by the following four 3D model collections: PSB, SHREC2009, SHREC2011, and SHREC2012. We compute all the values of μ by using the four collections with different numbers of the clusters. The result is shown in Table 5. From the table, we can see that the learned distance outperforms the native distance across almost of all the four collections (13.8% improvement of the mean value), but a bit lower than that of the batch process method. The statistical result demonstrates the generality of our method.

6.3. The combination of multiple features

One of the strengths of our online metric learning method is the combination of several features. In our method, we choose four well-known shape features but others could be used as well. To illustrate the strength of combining multiple

Table 5
The user effort for different 3D model collections. *NC* denotes the number of the clusters used in the clustering algorithm.

Collections	PSB			SHREC2009			SHREC2011			SHREC2012		
Distance type	Native	Batch	Online	Native	Batch	Online	Native	Batch	Online	Native	Batch	Online
$NC = 10$	0.60	**0.54**	0.56	0.79	**0.64**	0.66	0.72	**0.57**	0.64	0.52	0.42	**0.41**
$NC = 20$	0.35	**0.30**	0.33	0.56	**0.36**	0.42	0.30	**0.25**	**0.25**	0.47	**0.35**	0.37
$NC = 30$	0.32	**0.26**	0.28	0.51	**0.39**	0.41	0.24	**0.21**	0.24	0.47	**0.28**	0.36

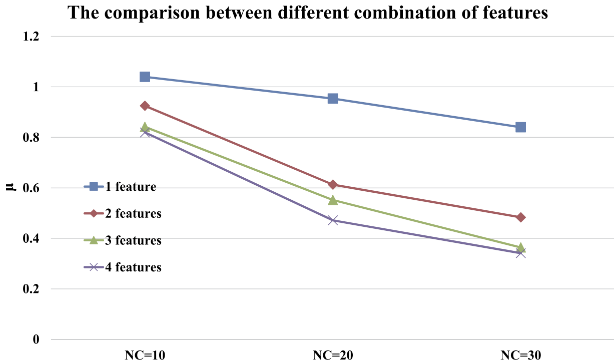


Fig. 4. The comparison between the combination of different number of features.

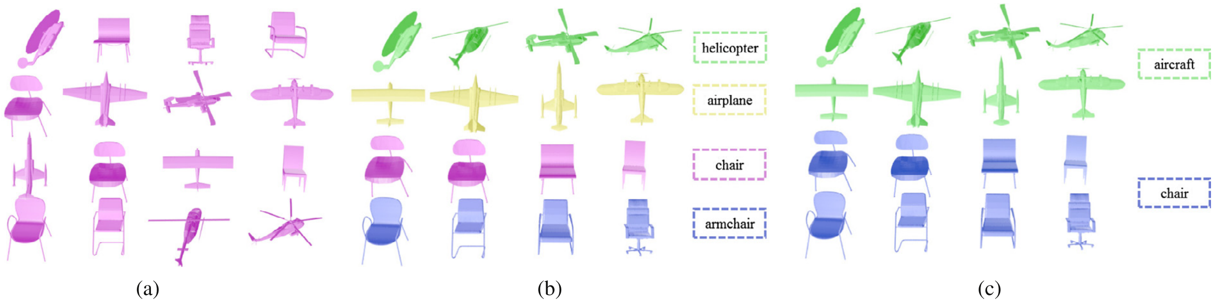


Fig. 5. Different criterions of classification. (a): The unclassified shapes; (b): The result of the semantic classification; (c): The result of the functional classification.

distances, we compare the user effort μ when only a subset of the features is used. The setting of the evaluation set is the same as the set in Section 6.1.

The four features used are listed in Section 4.1, which are denoted by χ_i, $i = 1, \ldots, 4$. Let Q_j be the set of all the possible combined features, where j is the number of the combined features. For example, $Q_1 = \{\chi_1, \chi_2, \chi_3, \chi_4\}$, and $Q_2 = \{\chi_1 + \chi_2, \chi_1 + \chi_3, \ldots, \chi_3 + \chi_4\}$, etc. For each Q_j, we compute the set of the value μ corresponding to the combined features in Q_j (e.g., there are six μ values for Q_2), and calculate the average value as the required user effort. The results of these average values are shown in Fig. 4. The figure shows that the average value μ using two kinds of features is significantly smaller than using a single feature. Similarly, using a combination of three or four distances also increase the efficiency but less significantly. Hence, we choose to combine all four features in all our experiments.

6.4. The expression of user intention

Since shape classification is generally a basic procedure of many other applications, it is necessary to classify shapes in different ways according to the specific application requirements. By our interface, the users can control the number and categories of the collection, which are not restricted by the standard benchmark. To show the expression of user intention, we ask some users to define some different criterions. Some users prefer the semantic criterion, which is consistent with the benchmark. And other users may prefer the function-level criterion. Fig. 5 shows the multiple classification results. Given the shape collection in Fig. 5(a), the user can classify them into four groups by their semantic information or two groups according to their functional information.

According to the function of the shapes, the whole shape collection can be reclassified into several groups, as shown in Table 6. We then evaluate the human effort required for the two classification results by the value μ. Table 7 shows the statistical result. From the table, we can see that the functional classification requires more human effort than the semantic one. The reason is that our descriptors are purely geometric, which does not include any functional information.

Table 6
The data set classified by function.

Class	Composition	Class	Composition
traffic	airplane, helicopter, motorbike, train	animals	fish, head
furniture	armchair, bed, bookshelf, cabinet, chair, table	plant	potted_plant, tree
electronics	cell_phone, keyboard, laptop	instrument	guitar, knife, sword

Table 7
The user effort for different criterions. *NC* denotes the number of the clusters used in the clustering algorithm.

Classification	$NC = 10$	$NC = 20$	$NC = 30$
semantic	0.61	0.29	0.14
functional	0.60	0.31	0.21

Table 8
The user effort for the incremental collections. *NC* denotes the number of the clusters used in the clustering algorithm.

Group id	Distance type	$NC = 10$	$NC = 20$	$NC = 30$
2	*Native*	0.55	0.51	0.44
	Learned	0.42	0.37	0.30
3	*Native*	0.59	0.52	0.50
	Learned	0.50	0.47	0.37

This implies that the success of the online metric learning method still depends on the quality of the features and the similarity distances employed.

6.5. The classification of incremental and large-scaled collection

We then evaluate the classification of the incremental and large-scaled collection. Since the large-scaled collection can be turned into the streaming data, which arrives block by block, the process of the large-scaled collection is as same as that of the incremental collection. Accordingly, we measure whether our incrementally updating module can improve the efficiency of classifying the incremental collection. Since an optimized distance function is saved into our updating record, we can use the learned metric to classify the incremental collection directly. We then compare the user effort by μ with the native and optimized distance function.

In our experiment, we partition the whole collection into three groups. After classifying the first group, we will classify the second group with the optimized distance, meanwhile the distance is optimized during the classification process. Then the third group is handled as the same way. Table 8 shows the statistical result. From the table, we can see that the incrementally learned distance outperforms the native distance significantly. The average improvement of the user effort is 20.8%.

6.6. User study

We perform a user study, where we ask 15 participants to classify 600 models selected from the PSB collection. The users in our study have not used our system before, and start to classify the model collection directly after we briefly describe the function and usage of our system. We do not tell the users the constitution of the model collection, so they do not know the scope and number of categories of the model collection. During the classification process, they can only browse the classified models and the models in the primary cluster.

After all the models have been classified, we find that the classification results are not very consistent. The category number of the classification results is from 6 to 17, which shows the significant diversity of taxonomies. Fig. 6 shows three of the classification results. Fig. 6(a) shows a fine-grained classification with 17 categories, where the ‘airplane’ models are classified into three types according to the shape of the wing. Fig. 6(c) shows the most coarse level of classification, and the collection is classified into six categories. Fig. 6(b) shows one of the intermediate level of classification. Most participants prefer the intermediate level, and have different requirements for different coarse categories of models. For example, some participants classify the ‘traffic’ models into the fine-grained level, but classify the ‘furniture’ models into the coarse level. These examples just prove that our method can fit the needs of the diversity of classifications, and the users can classify the collection flexibly and freely.

Fig. 7 shows the average times required for classifying the collection into different numbers of categories. In general, the users can complete the classification process in under 7 minutes, and the time cost depends on the number of category of the classification results. The fasted participant use only about 260 seconds to classify all the 600 models. From the figure, we can see that the time cost increases with the increase of the number of category slightly, and achieves the convergence when the number of category is more than 14. All the participants say the usage is intuitive. Besides, they

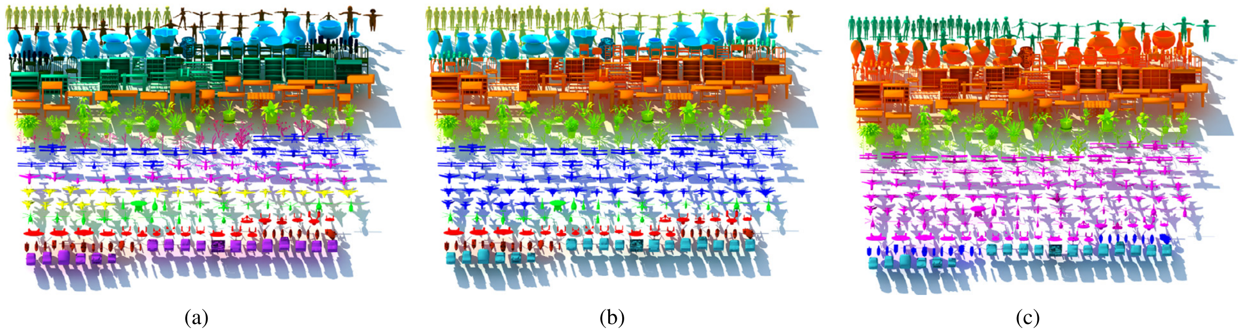


Fig. 6. Three different classification results. (a): Result 1; (b): Result 2; (c): Result 3.

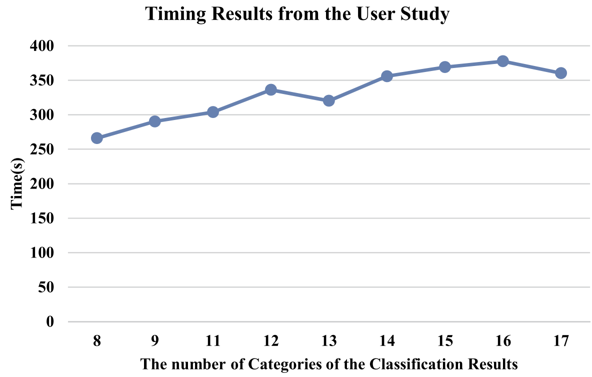


Fig. 7. Timing results from the user study.

can create the required groups from the primary cluster during each iteration fast, because the member selection interface only requires the judgement of the positives and negatives. One suggestion for improvement is to add the ability to provide some candidate labels of the majority category of the primary cluster, because the label selection of the grouping labeling interface is a little tedious. When the number of existing labels becomes larger, they may require more time to determine the correct label. This may explain the result in Fig. 7. When the number of category is small, the label selection is more faster and the time cost is much lower.

6.7. Limitation and discussion

Though online metric learning can improve the similarity function, the learned metric is still limited on the quality of the initial descriptors. When the initial shape distances are not consistent with users' requirement, the user effort is heavy. Especially at the early stage when the metric is not optimized, the purity of the given cluster is often very low, and the user is difficult to determine the required members within the cluster. Active learning (Settles, 2010) is an effective method to find which examples are more informative to be annotated, which may be our future work. Besides, OASIS and MDS are in general the linear learning methods, however, most shape descriptors are often nonlinear and high-dimensional. Introducing the online multiple kernel-based learning techniques (Xia et al., 2014) will be another important improvement. Moreover, the efficiency and friendliness of our interface can be improved by providing more useful hints of classification, such as the candidate labels of the primary cluster.

7. Conclusion

In this paper, we propose an iterative 3D shape classification method using online metric learning. The features of our method lie in three aspects. Firstly, the collection of 3D shapes can be classified group by group and iteratively. The unsupervised clustering, online metric learning and user intervention are integrated into a framework and work as a cohesive whole. Secondly, the users can classify the 3D shape collection flexibly and freely. They can get the desirable shape classification result without the prior determination about the scope and number of the categories. Finally, the scalable collection can be handled dynamically and efficiently. By our updating mechanism, our method can process the large scaled data set, and the dynamic increasing data set. Experimental results show that the proposed method improves both the effectiveness and efficiency of 3D shape classification.

Acknowledgements

We are grateful to the anonymous reviewers for their comments, suggestions, and additional references. This work is supported by the National High Technology Research and Development Program of China (2007AA01Z334), National Natural Science Foundation of China (61321491, 61272219, 61021062 and 61100110), Program for New Century Excellent Talents in University of China (NCET-04-04605), Project (No. ZZKT2013A12) supported by Key Projects Innovation Fund of State Key Laboratory, Natural Science Foundation of Jiangsu Province (BK2010375), Key Technology R&D Program of Jiangsu Province (BY2012190, BE2010072, BE2011058, and BY2013072-04), Project (No. CXLX12_0054) supported by the Graduate Training Innovative Projects Foundation of Jiangsu Province, the Program B for Outstanding PhD Candidate of Nanjing University (201302B06).

References

Anon, 2004. The Princeton shape benchmark. In: Proceedings of the Shape Modeling International 2004. SMI '04. IEEE Computer Society, Washington, DC, USA, pp. 167–178.
Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D., 2005. Learning a Mahalanobis metric from equivalence constraints. J. Mach. Learn. Res. 6, 937–965.
Barra, V., Biasotti, S., 2013. 3D shape retrieval using kernels on extended Reeb graphs. Pattern Recognit. 46 (11), 2985–2999.
Barra, V., Biasotti, S., 2014. 3D shape retrieval and classification using multiple kernel learning on extended Reeb graphs. Vis. Comput., 1–13.
Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U., 1999. When is "nearest neighbor" meaningful? In: Proceedings of the 7th International Conference on Database Theory. ICDT '99. Springer-Verlag, London, UK, pp. 217–235.
Biasotti, S., Cerri, A., Abdelrahman, M., Aono, M., Hamza, A.B., El-Melegy, M., Farag, A., Garro, V., Giachetti, A., Giorgi, D., Godil, A., Li, C., Liu, Y.-J., Martono, H.Y., Sanada, C., Tatsuma, A., Velasco-Forero, S., Xu, C.-X., 2014. Retrieval and classification on textured 3D models. In: Eurographics Workshop on 3D Object Retrieval 2014. 3DOR 2014, pp. 111–120.
Biasotti, S., Giorgi, D., Marini, S., Spagnuolo, M., Falcidieno, B., 2006. A comparison framework for 3D object classification methods. In: Gunsel, B., Jain, A., Tekalp, A., Sankur, B. (Eds.), Multimedia Content Representation, Classification and Security. In: Lecture Notes in Computer Science, vol. 4105. Springer, Berlin, Heidelberg, pp. 314–321.
Bronstein, A.M., Bronstein, M.M., Guibas, L.J., Ovsjanikov, M., 2011. Shape google: geometric words and expressions for invariant shape retrieval. ACM Trans. Graph. 30 (1), 1:1–1:20.
Chechik, G., Sharma, V., Shalit, U., Bengio, S., 2010. Large scale online learning of image similarity through ranking. J. Mach. Learn. Res. 11, 1109–1135.
Chen, D.-Y., Tian, X.-P., Shen, Y.-T., Ouhyoung, M., 2003. On visual similarity based 3D model retrieval. Comput. Graph. Forum 22 (3), 223–232.
Collins, R., Liu, Y., Leordeanu, M., 2005. Online selection of discriminative tracking features. IEEE Trans. Pattern Anal. Mach. Intell. 27 (10), 1631–1643.
Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y., 2006. Online passive–aggressive algorithms. J. Mach. Learn. Res. 7, 551–585.
Csakany, P., Wallace, A., 2003. Representation and classification of 3-D objects. IEEE Trans. Syst. Man Cybern., Part B, Cybern. 33 (4), 638–647.
Donamukkala, R., Huber, D., Kapuria, A., Hebert, M., 2005. Automatic class selection and prototyping for 3-D object classification. In: Fifth International Conference on 3-D Digital Imaging and Modeling. 3DIM 2005, pp. 64–71.
Everitt, B.S., Landau, S., Leese, M., Stahl, D., 2011. Cluster Analysis, 5th edition. John Wiley & Sons, Ltd.
Galleguillos, C., Mcfee, B., Lanckriet, G.R., 2014. Iterative category discovery via multiple kernel metric learning. Int. J. Comput. Vis. 108 (1–2), 115–132.
Godil, A., Dutagaci, H., Akgül, C., Axenopoulos, A., Bustos, B., Chaouch, M., Daras, P., Furuya, T., Kreft, S., Lian, Z., Napoléon, T., Mademlis, A., Ohbuchi, R., Rosin, P.L., Sankur, B., Schreck, T., Sun, X., Tezuka, M., Verroust-Blondet, A., Walter, M., Yemez, Y., 2009. Shrec'09 track: generic shape retrieval. In: Proceedings of the 2nd Eurographics Conference on 3D Object Retrieval. 3DOR '09. Eurographics Association, Aire-la-Ville, Switzerland, pp. 61–68.
Guillaumin, M., Verbeek, J., Schmid, C., 2009. Is that you? Metric learning approaches for face identification. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 498–505.
Huang, S.-S., Shamir, A., Shen, C.-H., Zhang, H., Sheffer, A., Hu, S.-M., Cohen-Or, D., 2013a. Qualitative organization of collections of shapes via quartet analysis. ACM Trans. Graph. 32 (4), 71:1–71:10.
Huang, Q.-X., Su, H., Guibas, L., 2013b. Fine-grained semi-supervised labeling of large shape collections. ACM Trans. Graph. 32 (6), 190:1–190:10.
Huber, D., Kapuria, A., Donamukkala, R., Hebert, M., 2004. Parts-based 3D object classification. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2004, vol. 2, pp. II-82–II-89.
Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., Wu, A., 2002. An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell. 24 (7), 881–892.
Kulis, B., 2013. Metric learning: a survey. Found. Trends Mach. Learn. 5 (4), 287–364.
Kulis, B., Jain, P., Grauman, K., 2009. Fast similarity search for learned metrics. IEEE Trans. Pattern Anal. Mach. Intell. 31 (12), 2143–2157.
Lee, Y.J., Grauman, K., 2011. Learning the easy things first: self-paced visual category discovery. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 1721–1728.
Li, B., Godil, A., Aono, M., Bai, X., Furuya, T., Li, L., López-Sastre, R., Johan, H., Ohbuchi, R., Redondo-Cabrera, C., Tatsuma, A., Yanagimachi, T., Zhang, S., 2012. Shrec'12 track: generic 3D shape retrieval. In: Proceedings of the 5th Eurographics Conference on 3D Object Retrieval. EG 3DOR'12. Eurographics Association, Aire-la-Ville, Switzerland, pp. 119–126.
Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Burtscher, M., Chen, Q., Chowdhury, N.K., Fang, B., Fu, H., Furuya, T., Li, H., Liu, J., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A., Wan, Y., Zhang, C., Zou, C., 2015. A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Comput. Vis. Image Underst. 131 (0), 1–27. Special section: Large Scale Data-Driven Evaluation in Computer Vision.
Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Chen, Q., Chowdhury, N.K., Fang, B., Furuya, T., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A., 2014. Large scale comprehensive 3D shape retrieval. In: Eurographics Workshop on 3D Object Retrieval 2014. 3DOR 2014, pp. 131–140.
Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Chen, Q., Chowdhury, N.K., Fang, B., Furuya, T., Johan, H., Kosaka, R., Koyanagi, H., Ohbuchi, R., Tatsuma, A., 2014. Large scale comprehensive 3D shape retrieval. In: Bustos, B., Tabia, H., Vandeborre, J.-P., Veltkamp, R. (Eds.), Eurographics Workshop on 3D Object Retrieval. Eurographics Association.
Lian, Z., Godil, A., Bustos, B., Daoudi, M., Hermans, J., Kawamura, S., Kurita, Y., Lavoué, G., Nguyen, H.V., Ohbuchi, R., Ohkita, Y., Ohishi, Y., Porikli, F., Reuter, M., Sipiran, I., Smeets, D., Suetens, P., Tabia, H., Vandermeulen, D., 2011. Shrec'11 track: shape retrieval on non-rigid 3D watertight meshes. In: Proceedings of the 4th Eurographics Conference on 3D Object Retrieval. 3DOR '11. Eurographics Association, Aire-la-Ville, Switzerland, pp. 79–88.
Ling, H., Jacobs, D., 2007. Shape classification using the inner-distance. IEEE Trans. Pattern Anal. Mach. Intell. 29 (2), 286–299.
Marini, S., Patane, G., Spagnuolo, M., Falcidieno, B., 2011. Spectral feature selection for shape characterization and classification. Vis. Comput. 27 (11), 1005–1019.
Ng, A.Y., Jordan, M.I., Weiss, Y., 2001. On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems. MIT Press, pp. 849–856.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., 2002. Shape distributions. ACM Trans. Graph. 21 (4), 807–832.

Paquet, E., Rioux, M., 1997. Nefertiti: a query by content software for three-dimensional models databases management. In: Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling, pp. 345–352.

Qin, F.-w., Li, L.-y., Gao, S.-m., Yang, X.-l., Chen, X., 2014. A deep learning approach to the classification of 3D CAD models. J. Zhejiang Univ. Sci. C 15 (2), 91–106.

Ristin, M., Guillaumin, M., Gall, J., Van Gool, L., 2014. Incremental learning of NCM forests for large-scale image classification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. CVPR, pp. 3654–3661.

Settles, B., 2010. Active learning literature survey. Tech. rep.

Shalev-Shwartz, S., Singer, Y., Ng, A.Y., 2004. Online and batch learning of pseudo-metrics. In: Proceedings of the Twenty-First International Conference on Machine Learning. ICML '04. ACM, New York, NY, USA, pp. 94–101.

Tabia, H., Colot, O., Daoudi, M., Vandeborre, J.-P., 2011. Non-rigid 3D shape classification using bag-of-feature techniques. In: 2011 IEEE International Conference on Multimedia and Expo. ICME, pp. 1–6.

Tabia, H., Daoudi, M., Vandeborre, J.-P., Colot, O., 2013. A parts-based approach for automatic 3D shape categorization using belief functions. ACM Trans. Intell. Syst. Technol. 4 (2), 33:1–33:16.

Tangelder, J.W., Veltkamp, R.C., 2008. A survey of content based 3D shape retrieval methods. Multimed. Tools Appl. 39 (3), 441–471.

Vranic, D., Saupe, D., 2002. Description of 3D-shape using a complex function on the sphere. In: Proceedings of the 2002 IEEE International Conference on Multimedia and Expo. ICME '02, vol. 1, pp. 177–180.

Wang, Y., Asafi, S., van Kaick, O., Zhang, H., Cohen-Or, D., Chen, B., 2012. Active co-analysis of a set of shapes. ACM Trans. Graph. 31 (6), 165:1–165:10.

Xia, H., Hoi, S., Jin, R., Zhao, P., 2014. Online multiple kernel similarity learning for visual search. IEEE Trans. Pattern Anal. Mach. Intell. 36 (3), 536–549.

Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S., 2003. Distance metric learning, with application to clustering with side-information. In: Advances in Neural Information Processing Systems, vol. 15. MIT Press, pp. 505–512.

Zhang, F., Sun, Z., Song, M., Lang, X., 2015. Progressive 3D shape segmentation using online learning. In: Solid and Physical Modeling 2014. Comput. Aided Des. 58 (0), 2–12.