3D model retrieval using hybrid features and class information

Bo Li · Henry Johan

Abstract To improve the retrieval performance on a classified 3D model database, we propose a 3D model retrieval algorithm based on a hybrid 3D shape descriptor ZFDR and a class-based retrieval approach CBR utilizing the existing class information of the database. The hybrid 3D shape descriptor ZFDR comprises four features, depicting a 3D model from different aspects and it itself is already comparable to or better than several related shape descriptors. To compute the distance between a query model and a target model within a class of a database, we define an integrated distance metric which takes into account the class information. It scales the distance between the query model and the target model according to the distance between the query model and the class. Our class-based retrieval approach CBR is general, it can be used with any shape descriptors to improve their retrieval performance. Extensive generic and partial 3D model retrieval experiments on seven standard databases demonstrate that after we employ CBR, the retrieval performance of our algorithm CBR-ZFDR is evidently improved and the result is better than that achieved by the state-of-the-art method on each database in terms of most of the commonly used performance metrics.

Keywords 3D model retrieval · Hybrid shape descriptor · Class information · Integrated distance

1 Introduction

With the growth in computer graphics and digital media applications, more and more 3D models are created and many professional or generic 3D model databases are available, such as Engineering Shape Benchmark (ESB) [21], Bonn University Architecture Databases Benchmark [45], Princeton Shape Benchmark (PSB) [35],

B. Li (✉) · H. Johan
School of Computer Engineering, Nanyang Technological University,
50 Nanyang Avenue, Singapore 639798, Singapore
e-mail: libo0002@ntu.edu.sg

National Taiwan University Shape Benchmark (NTU) [9], CCCC [43] and Shape Retrieval Contest (SHREC) datasets [1]. Many business and non-profit websites also provide 3D model databases for 3D models shopping or free downloading, such as 3DCafe (http://www.3dcafe.com/), 3DStudio (http://www.the3dstudio.com/) and High Polygon Models Crash3D (http://www.crash3d.net/) (provides free models).

3D model retrieval is a common and important operation on a 3D model database. Many shape descriptors and techniques have been proposed for 3D model retrieval. They can be classified into three categories: geometry-based, view-based and hybrid techniques. In general, hybrid shape descriptors can achieve better retrieval performances. However, it is still difficult to find a shape descriptor which performs well on all types of shape benchmarks.

Usually, existing shape descriptors represent a 3D model as a 3D shape descriptor and compare the shape descriptors of different models directly. However, we believe another promising approach to achieve a better retrieval performance is by exploiting the shape descriptors guided by the database classification information. That is, the available class information is utilized to improve the retrieval performance. In fact, all the 3D databases mentioned above are already classified. As such, in this paper we present a new retrieval algorithm which considers the class information of the target database when retrieving relevant models.

We propose a 3D model retrieval algorithm CBR-ZFDR which is based on a hybrid 3D shape descriptor named ZFDR and a class-based retrieval (CBR) algorithm. Motivated by the fact that different types of features are effective in characterizing different types of models [7], we develop the hybrid feature ZFDR by taking the advantages of both view-based and geometry-based techniques. ZFDR consists of four components, which are **Z**ernike moments, **F**ourier descriptor, **D**epth information and **R**ay-based features, each represents a 3D model from a different angle, either visually or geometrically. It itself has a better performance than the most related view-based shape descriptor Light Field [9] and hybrid shape descriptor DESIRE [44]. Its performance is also close to the state-of-the-art shape descriptors on several databases. To further improve the retrieval performance, we propose a CBR algorithm which incorporates the class information of the target database by defining an integrated distance which scales the model distance using the corresponding class distance. We show an apparent improvement in almost all commonly used performance metrics can be achieved after adopting the integrated distance. Moreover, the CBR approach can be used with any shape descriptors for enhancing their performance. Extensive experiments, for generic and partial retrieval, on seven standard 3D databases demonstrate the best performance of our retrieval algorithm CBR-ZFDR compared to those achieved by previous methods.

The rest of this paper is organized as follows. In Section 2, we review the related work in 3D model retrieval. The hybrid shape descriptor ZFDR is presented in Section 3. In Section 4, we present the details of our class-based 3D model retrieval algorithm CBR-ZFDR. Extensive experiment results are demonstrated in Section 5. Section 6 contains the conclusions and the future work.

2 Related work

Natraj et al. [19] and Tangelder et al. [38] reviewed and classified current typical 3D model retrieval techniques in their respective survey. In this section, we review the related techniques in generic and partial 3D model retrieval.

2.1 Generic 3D model retrieval

Geometry-based techniques   This approach uses the distribution of 3D features to characterize the geometric information of a 3D model. The 3D features can be either global, such as shape distribution [31] and shape histogram [2], or local, such as Extended Gaussian Images (EGI) [17] and conformal factor [4]. Shape distribution is defined as the statistics of the distances between any two points on the surface of a 3D model, while shape histogram measures the global radial distance distribution of a 3D model. Ben-Chen and Gotsman [4] proposed a 3D shape descriptor named conformal factor which depicts the amount of local work involved to transform a model into a sphere. Recently, Shih and Chen [34] proposed an **a**ngular **r**adial **t**ransformation-based **e**levation **d**escriptor (ART-ED) that encodes both external and internal information of a 3D model. Graph-based methods [16, 37] use skeleton or topology graph to represent a 3D model and employ a graph matching method to measure the distance between two graphs. Spherical harmonics [22] can be applied on the extracted rotation-dependent 3D features to make them rotation-invariant.

View-based techniques   Rather than extracting the 3D features directly as the geometry-based techniques, view-based techniques represent a 3D model using a set of views. A typical view-based technique is Light Field [9], which computes the minimum distance between 10 corresponding views of two models. Salient local visual feature-based retrieval method [30] adopts the Bag-Of-Features (BoF) framework to accumulate the Scale Invariant Feature Transform (SIFT) [29] features of multiple depth views into an occurrence histogram to represent a 3D model. Recently, Lian et al. [26] proposed a view-based descriptor which adopts the BoF approach to extract the SIFT features of a view and utilizes an efficient multi-view shape matching approach to find the minimum distance between the corresponding views of two models. They considered the 24 axes permutations of a normalized 3D model. Axenopoulos et al. [3] also adopted a view-based approach but relied on a more accurate 3D model alignment method.

Hybrid techniques   Hybrid approach employs both the visual and geometric information of a 3D model. DESIRE [44] is a hybrid shape descriptor which comprises three shape descriptors: depth buffer-based descriptor, silhouette-based descriptor and ray-based with spherical harmonic representation descriptor. DESIRE achieves superior performance over several famous view-based or geometry-based techniques, such as Light Field [9] and Spherical harmonics [22]. Papadakis et al. [32] proposed another hybrid 3D shape descriptor by combining both depth buffer-based 2D features and spherical harmonics-based 3D features. Papadakis et al. [33] presented another novel hybrid 3D shape descriptor named PANORAMA using a set of panoramic views of a 3D model. The panoramic views not only capture the visual information of the 3D model but also contain the geometric information, such as the 3D location and orientation of the model's surface. The views are generated by projecting the model to three axis-aligned cylinders respectively and then unfolding the projection images into 2D images. They used Fourier and wavelet transforms to extract the features for each panoramic view. Recently, Leng and Xiong [25] proposed a hybrid shape descriptor named TUGE which combines the **t**wo-view

version of the depth buffer-based shape descriptor in [43] and the **GE**DT shape descriptor in [13]. It has a slightly better performance than DESIRE.

According to our knowledge, PANORAMA achieves the best overall performance on several 3D model databases, including PSB [35], ESB [21], CCCC [43] and NIST [12], among the available existing shape descriptors. Our experiments show that our retrieval algorithm CBR-ZFDR can achieve better retrieval performance than that of PANORAMA in terms of most commonly used performance metrics on the above databases.

Using class information   Class-based retrieval scheme has been used in document retrieval or classification [15, 28]. For example, Han et al. [15] first applied centroid-based classifier to automatic text categorization and it achieved good performance. With the growth in 3D model retrieval research, it was introduced to improve its retrieval performance. For instance, Hou et al. [18] proposed a retrieval approach based on semantic labeling: first assign the relevant class for a query model based on the Support Vector Machine (SVM) clustering information of the target 3D model database and then rank all the models belonging to the relevant class based on a feature vector selection technique which is also dependent on the clustering results. Apparently, the accuracy of the first several nearest neighbors is highly dependent on the semantic clustering results. Xu and Li [46] defined the distance between two models by adding the weighted difference between their back propagation neural network (BPNN) 3D model classification output vectors and the Euclidian (L2) distance between their 3D moment feature vectors. Biasotti et al. [6] proposed a 3D model classification approach by comparing a query model with several prototypes selected to represent a class and applied the prototype-based scheme into a 3D model retrieval application. They made a comparison study of the Nearest Neighbor (NN)-based and two prototype-based retrieval methods and the results show that NN achieves the best retrieval accuracy, though may be slower.

Tatsuma and Anon [39] designed a hybrid shape descriptor named multi-Fourier spectra descriptor (MFSD) by applying 2D or 3D Fourier transform to the contour, silhouette and depth images or the voxelization representation of a 3D model. They also utilized a spectral clustering method to cluster the models before retrieval. To measure the distance between the query model and a target model in a clustered database, they used an addition operator to combine the minimum distance between the query model and the models in the most relevant cluster as well as the model distance between the query model and the target model. According to our knowledge, this is the only existing algorithm that directly combines the cluster distance and model distance to form a joint distance for 3D model retrieval. In this paper, we propose an integrated distance which outperforms the above mentioned additive one. Moreover, none of the existing 3D model retrieval algorithms utilize the already available class information of a classified 3D model database. Thus, in this paper we propose a new 3D model retrieval algorithm by taking into account the existing class information.

2.2 Partial 3D model retrieval

Partial retrieval techniques can be mainly classified into two groups: (1) graph-based, such as Tierny et al.'s [40] Reeb Pattern Unfolding (RPU) method, Biasotti

et al.'s [5] Extended Reeb Graph (ERG) approach, and Cornea et al.'s [11] skeleton matching-based approach (CORNEA); (2) local feature-based, such as Toldo et al.'s [41] Bag-of-Words component Feature based approach (BoF). The main idea and performance of the above partial retrieval techniques are as follows.

RPU [40] is a graph-based partial 3D retrieval method based on the reeb graph representation. First, it segments the model based on reeb graph and encodes the relationship of parts into a dual reeb graph. Then, the concept of "reeb pattern" on a reeb graph is introduced to speed up the process of partial matching. It needs 4∼30 sec to process a query on the AIM@Shape Watertight Models Benchmark (**WMB**) database with a 3 GHz P4 PC.

ERG [5] is a graph-based approach based on Extended Reeb Graph (ERG) shape descriptor, which contains not only structural but also geometrical information of a model. A directed attributed graph matching method is adopted to find the maximum common sub-parts between two ERGs. It can be roughly estimated based on the provided performance data [5] that ERG needs at least 3∼11 sec (1.4 sec for feature matching, and 1.5∼10 sec for the preprocessing of feature matching) using a 3.4 GHz PC.

CORNEA [11] is a graph-based approach which extends the skeleton-based matching framework by Sundar et al. [37] with more robust and efficient skeletonization and matching algorithms. The skeletonization is performed by propagating normals to the interior of a 3D model and the matching is based on a distribution-based graph matching method utilizing a distance measure between distributions called Earth Mover's Distance (EMD) [10]. There is no computational time information provided in the paper.

BoF [41] is a local feature-based approach by extending the 2D Bag-of-Words (BoW) features to represent 3D components. First, it segments a 3D model into several subparts and then extracts a local feature for each subpart. Next, the local features are clustered to define a 3D vocabulary. Finally, it uses an occurrence histogram as the shape signature for a subpart or a complete model to do the matching. It needs about 5.5 sec to process a query model on a 1.66 GHz laptop.

In experiment section, we compare our retrieval algorithm with these four partial similarity shape descriptors in terms of accuracy and speed. The results show that our hybrid shape descriptor ZFDR itself already outperforms the above four shape descriptors and after applying our CBR algorithm we achieve an even better performance. In addition, our shape descriptor is also efficient to compute.

3 Hybrid shape descriptor ZFDR

We define a hybrid shape descriptor, which we named ZFDR, to represent a 3D model. ZFDR comprises four components: **Z**ernike moments feature, **F**ourier descriptor feature, **D**epth information feature and **R**ay-based feature. It contains both visual and geometric information of a 3D model. The computation of the shape descriptor consists of two steps: first normalize the 3D model and then compute the descriptor. Figure 1 graphically shows the ZFDR feature extraction process. For the normalization part, we first compute the bounding sphere of the 3D model. Then, we translate the model so that the center of the bounding sphere coincides with the origin of the coordinate system and then uniformly scale the model to make
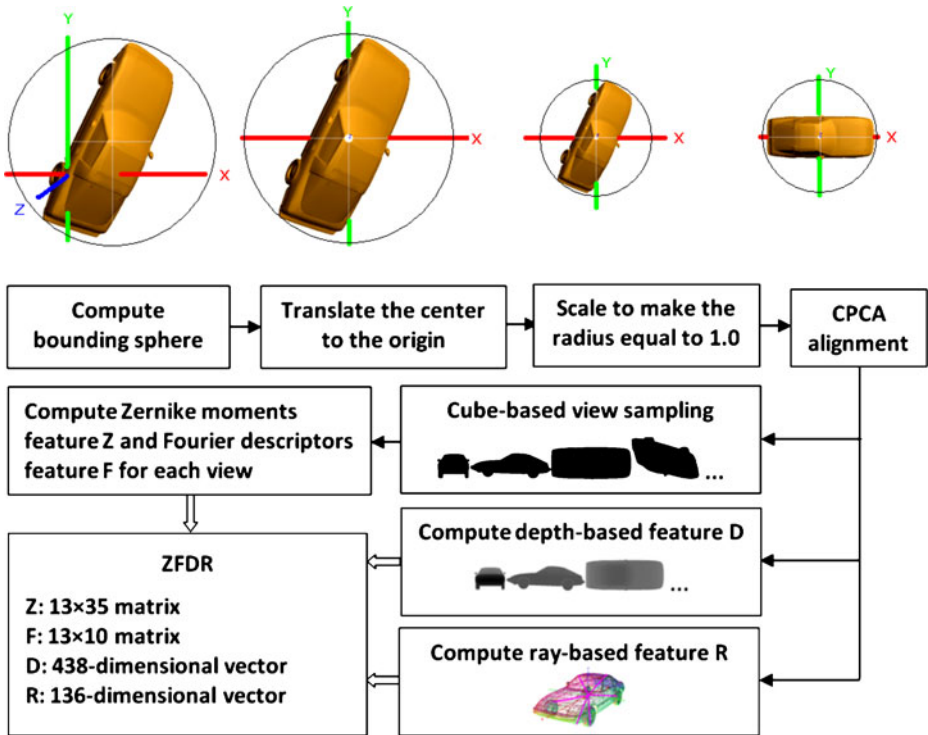
Fig. 1 ZFDR feature extraction process

the radius of its bounding sphere equal to 1.0. Next, we utilize Continuous Principle Component Analysis (CPCA) [43] alignment algorithm to align the 3D model. For shape descriptor computation, we present the details as follows.

3.1 Visual information features

In this section, we first introduce the view sampling method for extracting the two visual information features and then present each feature respectively.

Cube-based view sampling To balance between the computational time for feature extraction and retrieval performance, we sample 13 silhouette views to represent a 3D model. We set cameras at 13 sampled locations on a cube: $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, $(1, 1, 1)$, $(-1, 1, 1)$, $(-1, -1, 1)$, $(1, -1, 1)$, $(1, 0, -1)$, $(0, 1, -1)$, $(1, 1, 0)$, $(0, 1, 1)$, $(1, 0, 1)$, $(1, -1, 0)$. As shown in Fig. 2a, they are composed of three adjacent face center views (magenta squares), four top corner views (red squares) and six middle edge views (blue squares), respectively. Figure 2b shows an example of 13 silhouette views of a chair model.

To characterize the features of a silhouette view, we adopt an image descriptor proposed by Zhang and Lu [48]. It is composed of Zernike moments and Fourier descriptors.
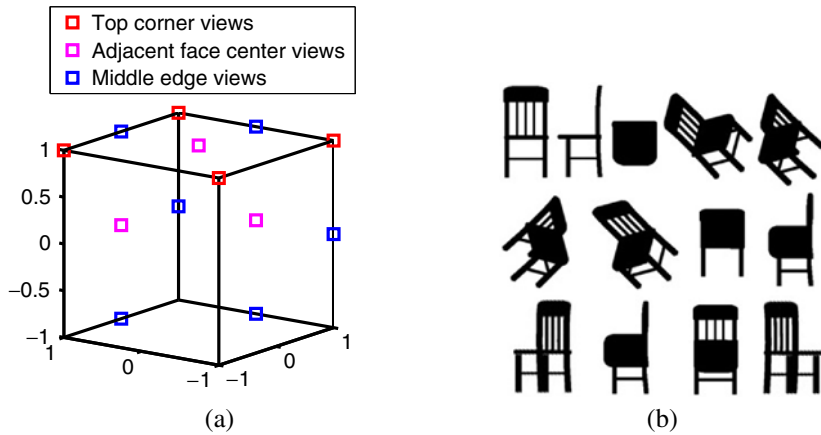
Fig. 2 View sampling. **a** Camera locations; **b** An example of 13 silhouette views of a chair model

Zernike moments feature (Z) Zernike moments depict the region-based features of a silhouette view. We compute the Zernike moments [23] (up to the 10th order, totally 35 moments) of each view image and concatenate them orderly according to the order of the view sequence to form a 13×35 matrix to define the Zernike moments feature of a 3D model.

Fourier descriptor feature (F) Fourier descriptor represents the contour information of a silhouette view using a series of Fourier coefficients (one dimensional vector). Fourier descriptors can be defined on different features of the contour, such as curvature and centroid distance. However, Fourier descriptor defined on centroid distance was proved [47] to have better performance than other types in retrieving 2D shapes and thus we also adopt the centroid distance-based Fourier descriptor [47]. We use the first 10 Fourier coefficients as the Fourier descriptor. By combining the Fourier descriptors of 13 views, we forms a 13×10 matrix as the Fourier descriptor feature of a 3D model.

3.2 Geometric information features

Zernike moments and Fourier descriptor features capture the visual information of a 3D model. These types of features are found to be effective in characterizing some certain types of models like "sea animal" models, but for other certain types of models, such as "car models", depth buffer-based features is more effective [7]. That is, different types of features have advantages in measuring different types of models. Motivated by this, we also extract the geometric information features to form a hybrid shape descriptor that can represent more types of models effectively. Vranic [43] defined a depth buffer-based feature and a ray-based with spherical harmonic representation feature. These two features characterize the geometric information from different perspectives and we integrate them into our hybrid shape descriptor.

Depth information feature (D) This feature is composed of 2D Fourier coefficients of six depth buffer images. We first render the six depth views of a 3D model and

then apply 2D Fourier Transform to them and finally 438 Fourier coefficients are used as the depth features of a 3D model.

Ray-based feature (R)   First, the ray-based feature vector in the spatial domain is extracted based on the outmost intersections between the model and a set of rays emanating from the center of the model. Then, the obtained radial distance feature vector is transformed from the spatial domain to the spectral domain using Spherical Harmonics Transform [22]. We use a 136-dimensional feature vector to depict the ray-based features.

3.3 Combining the visual and geometric features

We define the hybrid shape descriptor of model m_i by combining Zernike moments feature Z_i, Fourier descriptor F_i, Depth information feature D_i and Ray-based descriptor R_i as ZFDR.

To compute the hybrid descriptor distance d_{ZFDR} between two models m_i and m_j, we first assign appropriate distance metrics to measure the component distances d_Z, d_F, d_D and d_R, then we combine the four component distances to determine the hybrid descriptor distance d_{ZFDR}. After comparing the performance of different distance metrics [24], such as city block distance (L1), Euclidean distance (L2), Canberra distance [24], correlation distance, divergence distance and scaled-L1 distance [43], we choose the scaled-L1 distance metric for Z_i, D_i and R_i and the Canberra distance metric for F_i, respectively. Scaled-L1 means scaling or normalizing the feature vector by its L1-norm before applying the L1 distance metric. We find it improves the retrieval performance for our features of Z, D and R. While, Canberra distance is only applied to the Fourier descriptor F is also based on the performance comparison in terms of the overall performances of the complete shape descriptor ZFDR on several 3D model benchmarks. Now, we give the definitions for the four component distances d_Z, d_F, d_D and d_R.

$$d_Z = \frac{1}{13} \sum_{p=1}^{13} \sum_{r=1}^{35} \left| \frac{Z_i(p,r)}{\left\| Z_{i,p} \right\|_1} - \frac{Z_j(p,r)}{\left\| Z_{j,p} \right\|_1} \right|, \tag{1}$$

where Z_i and Z_j are the Zernike moments feature matrices of models m_i and m_j. $Z_{i,p}$ and $Z_{j,p}$ represent the p-th row vector of Z_i and Z_j. $\|.\|_1$ represents the L1-norm of a vector. Here, we apply the scaled-L1 distance metric on the corresponding views

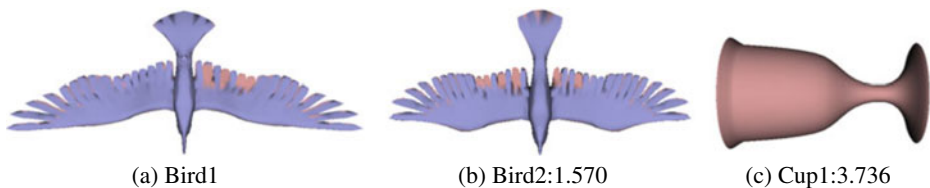(a) Bird1   (b) Bird2:1.570   (c) Cup1:3.736

Fig. 3 An example of ZFDR distances. The number is the ZFDR distance between the model Bird1 and the respective model

Table 1 Z, F, D, R component distances for the example in Fig. 3

Distances	d_Z	d_F	d_D	d_R	d_{ZFDR}
Bird2	0.352	0.450	0.275	0.494	1.570
Cup1	0.912	0.956	0.673	0.912	3.736

of two models and use the average distance of view pairs to represent the Zernike moments distance between the two models, $d_Z \in [0, 1]$.

$$d_F = \frac{1}{13 \times 10} \sum_{p=1}^{13} \sum_{r=1}^{10} \frac{\left| F_i(p, r) - F_j(p, r) \right|}{F_i(p, r) + F_j(p, r)}, \tag{2}$$

where F_i and F_j are the Fourier descriptors of m_i and m_j, $d_F \in [0, 1]$.

$$d_D = \frac{1}{438} \sum_{p=1}^{438} \left| \frac{D_i(p)}{\|D_i\|_1} - \frac{D_j(p)}{\|D_j\|_1} \right|, \tag{3}$$

where D_i and D_j are the Depth information descriptor vectors of m_i and m_j, $d_D \in [0, 1]$.

$$d_R = \frac{1}{136} \sum_{p=1}^{136} \left| \frac{R_i(p)}{\|R_i\|_1} - \frac{R_j(p)}{\|R_j\|_1} \right|, \tag{4}$$

where R_i and R_j are the Ray-based descriptor vectors of m_i and m_j, $d_R \in [0, 1]$.

Then, we define the hybrid descriptor distance d_{ZFDR} between model m_i and model m_j as follows,

$$d_{ZFDR} = d_Z + d_F + d_D + d_R. \tag{5}$$

The four component features Z, F, D, R depict a model from different aspects and they have the same contribution for the hybrid descriptor distance computation. Therefore, we linearly combine them. In addition, the pair feature distances for the four features fall in the same range of [0, 1], as such, we assign the same weight for each component feature. An example showing the hybrid shape distance computation is demonstrated in Fig. 3 and Table 1.

4 3D model retrieval algorithm using class information

In this section, we propose a 3D model retrieval algorithm named CBR-ZFDR which utilizes a new **C**lass-**B**ased **R**etrieval algorithm CBR and the ZFDR hybrid descriptor presented in Section 3. For CBR, we define an integrated distance to fuse the model and class distances. One advantage of our CBR scheme is that it is general, that is, we can use any shape descriptors to represent 3D models when we apply the CBR scheme.

The scenario for our retrieval is that given a query model q and a classified 3D model database $M = \{m_i | i = 1 \cdots n\}$, where m_i are the models in the database, we retrieve similar target models in database M. Both the query and target 3D models

are defined as triangular meshes. Our 3D model retrieval algorithm CBR-ZFDR is composed of the following five steps.

(1) **Shape descriptors extraction.** We extract the ZFDR shape descriptors of query model q (on-line processing) and all the models $\{m_i\}$ in the database M (off-line preprocessing), based on the method in Section 3.

(2) **Model distance computation.** We compute the shape descriptor distance $d(q, m_i)$ between query model q and every model m_i in the database based on (1–5).

(3) **Class distance computation.** To measure the dissimilarity between query model q and a class in the database, we can use minimum, average or centroid distances.

The classified 3D model database M has a number of classes, each of which contains some models. We denote C_j as the j-th class of database M and assume model $m \in C_j$. The minimum distance between query model q and all models in class C_j is defined as the class distance $d_c(q, C_j)$,

$$d_c(q, C_j) = \min_{m \in C_j} \{d(q, m)\} \tag{6}$$

Average distance is computed by averaging all the distances between query model q and the models in C_j. Centroid distance [15] is determined by first computing the shape descriptor centroid of class C_j by averaging the shape descriptors of the models in C_j and then computing the distance between this shape descriptor centroid and the shape descriptor of query model q to define the centroid distance. In our experiments, if the query model is selected from the database, to avoid bias we exclude this model from C_j when computing the class distance. Based on experiments (Section 5.1.1), we found that minimum distance performs the best and thus we adopt this class distance.

(4) **Integrated distance computation.** To measure the distance between query model q and target model m_i (assume $m_i \in C_j$), we scale its model distance $d(q, m_i)$ using the corresponding class distance $d_c(q, C_j)$ to define a class-based distance d_{cbr},

$$d_{cbr}(q, m_i) = d_c(q, C_j)^\alpha \times d(q, m_i), \tag{7}$$

where α ($\alpha > 0$) is a constant to adjust the relative weight of the class distance with respect to the model distance. We set α to be 3 in our retrieval algorithm based on experimental results (Section 5.1.1). This definition of integrated distance is general and can be used with any shape descriptors to improve their retrieval performance.

(5) **Ranking and output.** Sort all the models in the database in ascending order based on their integrated distances and output the retrieval lists accordingly.

5 Experiments

To investigate the generic 3D model retrieval performance as well as the characteristics of our retrieval algorithm CBR-ZFDR, we selected the following six representative standard benchmark databases,

– Princeton Shape Benchmark (**PSB**) [35]. It contains 1,814 models totally, which are classified into two parts: test and train datasets. Both datasets contain 907 models and the test dataset is classified into 131 classes and the train dataset is classified into 129 classes. We use the train dataset only in Section 5.2.2 and for other cases we only use the test dataset.
– Engineer Shape Benchmark (**ESB**) [21]. This is a CAD model database which contains 867 models, classified into 45 classes.
– National Taiwan University database (**NTU**) [9]. This database contains 1,833 3D models and only 549 3D models are grouped into 47 classes and the rest 1,284 models are assigned as the "miscellaneous". Thus, we only use the 549 classified models as the NTU database.
– Konstanze 3D Model Benchmark (**CCCC**) [43]. CCCC comprises 1,838 models and 473 models are grouped into 55 types and other 1,365 models are unclassified. Thus, we only use the 473 classified models as the CCCC database.
– McGill 3D Shape Benchmark (**MSB**) [36]. This database is to test the performance of articulated or non-rigid models, such as humans and ants. It is composed of 19 classes and 457 models.
– NIST Generic Shape Benchmark (**NIST**) [1, 12]. This database is to overcome several shortcomings or biases of previous benchmarks, such as different sizes of each class. It contains 800 models, classified into 40 classes, 20 models each.

To comprehensively evaluate the generic 3D model retrieval results, we employ six metrics including Precision-Recall, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Discounted Cumulative Gain (DCG) [35] and Average Precision (AP). Precision indicates how much percentage of the top K models belongs to the same class as the query model while recall means how much percentage of a class has been retrieved among the top K retrieval list. NN measures the percentage of the closest matches that are relevant models. FT is the recall of the top $C - 1$ list, where C is the cardinality of the relevant class of the query model. Similarly, ST is the recall of the top $2(C - 1)$ list. DCG is defined as the summed weighted value related to the positions of the relevant models. AP is to measure the overall performance and it combines precision, recall as well as ranking positions. A good AP needs both high recall and precision. AP can be computed by counting the total area under the Precision-Recall curve.

To assess our algorithm's ability in partial 3D model retrieval, we choose the 3D model database benchmark used in the SHREC 2007 partial matching track [42],

– AIM@Shape Watertight Models Benchmark (**WMB**) [42]. The target dataset has 400 watertight models, divided into 20 classes, 20 each. The query dataset contains 30 models by combining the parts of two or more models of the target database (two typical examples are the query models in Figs. 10 and 11).

We use the Normalized Discounted Cumulative Gain (NDCG) [20] metric to evaluate the performance of partial retrieval results. This metric is explained in Section 5.3 which is dedicated for partial 3D model retrieval experiments.

## 5.1 Comparative evaluation with respect to algorithm configurations

In this section, we justify our choice of class distance, where ZFDR is used as the shape descriptor and the evaluation of our hybrid shape descriptor ZFDR.

5.1.1 Choices of class distance and parameter α

Three different types of class distance, which are minimum, average and centroid distances, are mentioned in Section 4. To justify our choice of using minimum distance, for each of the seven databases, we perform a comparison of our class-based retrieval algorithm with respect to different class distances. Two representative examples for PSB and NIST databases are shown in Fig. 4. In general, we find that the best performance is achieved by using the minimum class distance for all the databases.

The parameter α controls the relative weight of the class distance. To set an appropriate weight value for α for our CBR algorithm, we perform a comparison experiment for each database by selecting five values (1, 2, 3, 4, 5) for parameter α. We have found that bigger α will evidently improve the metrics of FT, ST, DCG

Fig. 4 Precision-Recall plots comparison in terms of different class distance definitions. "Minimum", "Average" and "Centroid" denote the class-based retrieval approaches using minimum, average and centroid class distances, respectively
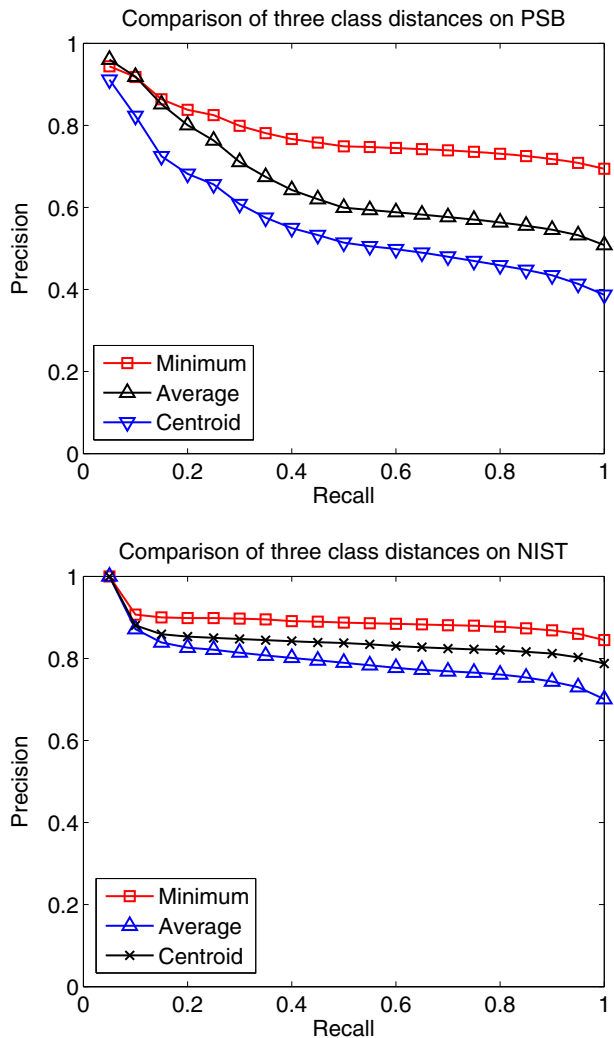
and AP. However, the front part (e.g. recall ≤0.2 for PSB when using CBR-ZFDR) of the Precision-Recall plots with the biggest α does not give the best result in terms of precision. Based on the fact that the front part of the Precision-Recall plot is relatively more important than the rear part in retrieval applications and we also need to consider other performance metrics such as FT, ST, DCG and AP, we set $\alpha = 3$ in our class-based algorithm because it can achieve the best overall performance.

The weight value selection for α is directly related to our formulation of the integrated distance and is insensitive to the descriptors employed. Thus, there is no need to adjust a chosen weight value for parameter α each time we use CBR with a new shape descriptor. For example, we also verify the above property of our CBR algorithm with PANORAMA.

5.1.2 Analysis of our hybrid shape descriptor ZFDR

To justify the feature selection for our hybrid shape descriptor, we analyze the contribution of visual and geometric features by performing experiments on all the seven databases. To find the intrinsic properties of the hybrid shape descriptor ZFDR, in the experiments, we use only the shape descriptor itself and do not employ the class-based retrieval approach. We also compare ZFDR with the two most related shape descriptors: DESIRE [44] and LF [9]. For DESIRE, we generate the results based on their provided execution files [43]. For LF, we refer to the experiment results in PSB [35] and PANORAMA [33]. Some "DCG" results are not provided in these papers and are indicated by "–". Two representative results are shown in Fig. 5 and Table 2 and others are very similar.

As can be seen in both Fig. 5 and Table 2, firstly, ZFDR has a better performance compared to only visual information-based descriptor ZF or only geometric information-based descriptor DR. Therefore, our hybrid shape descriptor containing both the visual and geometric features outperforms the ones using the visual or geometric features alone. Secondly, ZFDR also outperforms DESIRE and LF. ZFDR exceeds DESIRE in NN, FT, ST and AP by 6.1%, 8.7%, 7.0% and 5.8% on PSB, 4.9%, 8.1%, 4.9% and 5.6% on NIST. This is contributed to our carefully selecting and integrating different types of features as well as related distance metrics to make them complement with each other and thus the hybrid shape descriptor can represent more types of models comprehensively and effectively.

5.2 Generic 3D model retrieval

5.2.1 Standard benchmark databases

To compare the performance of our retrieval algorithm CBR-ZFDR, we consider the following three state-of-the-art algorithms,

– **2D-3D** [32]: a 2D/3D hybrid descriptor based on 2D depth images and 3D spherical harmonics.
– **MFSD** [39]: a multi-Fourier spectra descriptor (MFSD) by integrating depth information, contour and silhouette features of rendered views as well as a 3D Fourier features through voxelization. It adopts a cluster-based approach by clustering the target models before retrieval. An addition operator is used to

Fig. 5 ZFDR features
contribution analysis on PSB
and NIST databases. ZF and
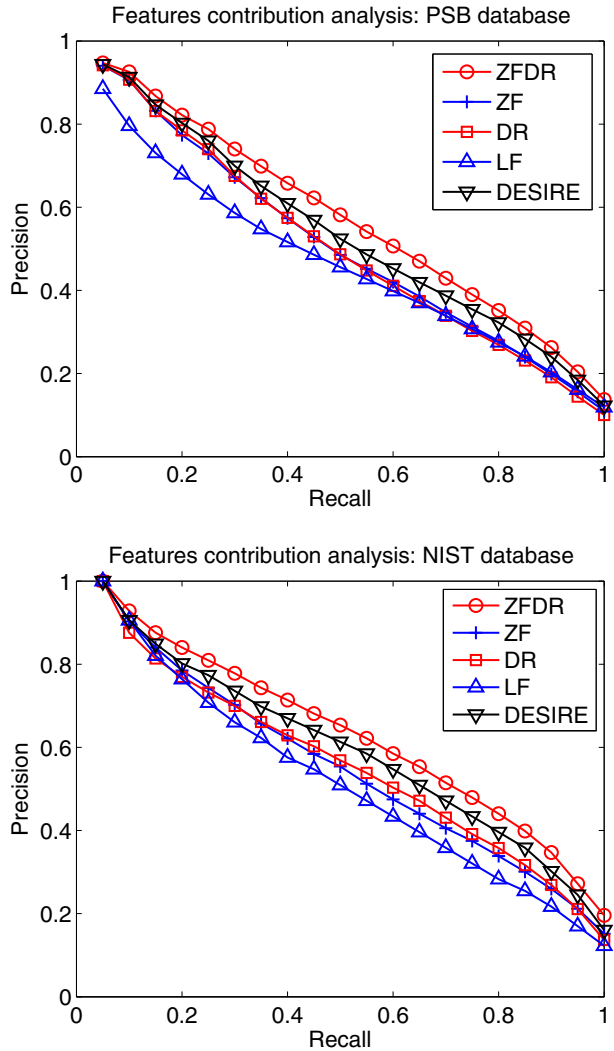DR are the two main features
of the hybrid descriptor ZFDR

Table 2 Other performance
metrics for the ZFDR features
contribution analysis on PSB
and NIST databases

Methods	NN	FT	ST	DCG	AP
PSB					
ZFDR	69.8	43.9	54.9	0.691	60.6
DESIRE	65.8	40.4	51.3	0.663	57.3
LF	65.7	38.0	48.7	–	50.2
DR	64.7	37.2	48.1	0.637	54.0
ZF	62.1	37.3	48.3	0.638	54.3
NIST					
ZFDR	87.8	55.0	68.1	0.821	66.2
DESIRE	83.7	50.9	64.9	–	62.7
LF	84.1	43.9	56.0	–	55.1
DR	80.3	48.0	61.7	0.771	59.2
ZF	83.6	47.5	60.7	0.775	58.6

combine the spectral clustering (SC) distance and model distance to form a MFSD+SC algorithm.

– **PANORAMA** [33]: a hybrid 3D shape descriptor that performs best by utilizing a set of panoramic views. A local relevance feedback (LRF) is developed to further improve the retrieval performance and the method is named PANORAMA+LRF.

PANORAMA and 2D-3D do not utilize class information, but they represent the state-of-the-art performances that have been achieved on the six databases and thus we can know which performance level we can reach if incorporating the already available class information based on our class-based retrieval algorithm CBR-ZFDR. Figure 6 and Table 3 compare the performance of our CBR-ZFDR algorithm and the above mentioned three shape descriptors. To demonstrate the superior performance of our integrated distance, we compare CBR-ZFDR with a modified CBR-ZFDR algorithm which applies the addition operator to fuse the class and model distances and we denote it as CBR-ZFDR-A. To evaluate the effectiveness of our CBR algorithm, for comparison, we also list the performances when using only the ZFDR shape descriptor. For the performances of 2D-3D and PANORAMA, we mainly refer to the experiment results in 2D-3D [32], PANORAMA [33] and PSB [35]. For the performance of PANORAMA on NTU, MSB and NIST, we performed experiments based on their provided executable files [33]. Some "DCG" results that are not provided in these papers are indicated by "–".

As can be seen in Fig. 6 and Table 3, firstly, by comparing the performance of CBR-ZFDR and CBR-ZFDR-A, we can conclude that by using the scaling operation
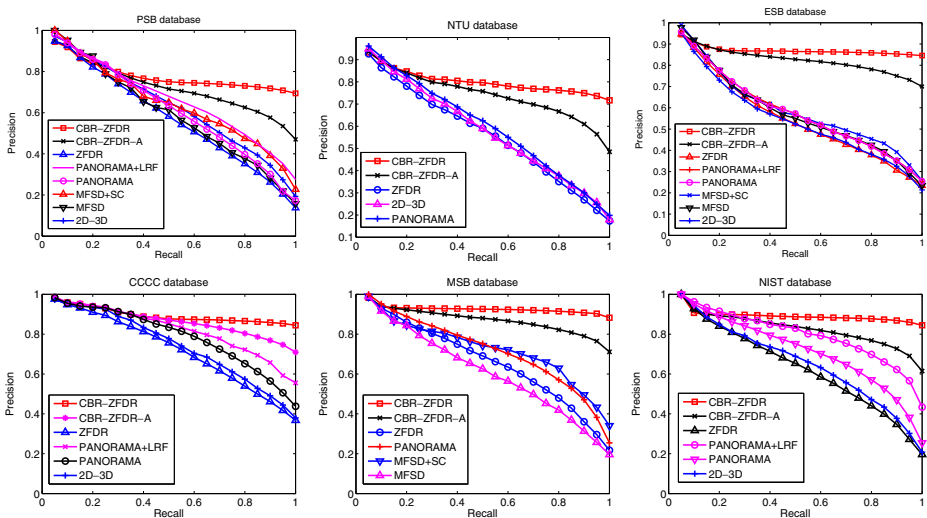

Fig. 6 Performance comparison: Precision-Recall plots of our retrieval algorithm CBR-ZFDR and three state-of-the-art shape descriptors. "CBR-ZFDR" denotes our retrieval algorithm that utilizes the CBR algorithm described in Section 4 and the ZFDR shape descriptor presented in Section 3. "CBR-ZFDR-A" denotes a variation of CBR-ZFDR algorithm which uses addition to fuse the class and model distances. "ZFDR" means using only our hybrid shape descriptor ZFDR and do not use the CBR algorithm

Table 3 Performance metrics of our CBR-ZFDR algorithm and three state-of-the-art shape descriptors

| Methods	NN	FT	ST	DCG	AP
PSB					
CBR-ZFDR	69.8	69.5	77.0	0.801	79.2
CBR-ZFDR-A	69.8	60.8	71.3	0.773	74.5
ZFDR	69.8	43.9	54.9	0.691	60.6
PANORAMA+LRF	75.2	53.1	65.9	–	69.4
PANORAMA	75.3	47.9	60.3	–	64.5
MFSD+SC	71.1	50.9	63.1	0.723	67.4
MFSD	71.6	45.3	59.1	0.704	62.6
2D-3D	74.2	47.3	60.6	–	66.1
NTU					
CBR-ZFDR	74.7	74.3	79.7	0.833	81.4
CBR-ZFDR-A	74.7	65.8	74.4	0.809	76.4
ZFDR	74.7	44.9	57.5	0.725	59.4
PANORAMA	79.7	49.0	61.0	0.755	63.0
2D-3D	76.2	46.6	59.1	–	61.2
ESB					
CBR-ZFDR	84.1	84.2	88.4	0.909	87.8
CBR-ZFDR-A	84.1	77.7	85.5	0.895	83.8
ZFDR	84.1	46.8	60.9	0.769	58.0
PANORAMA+LRF	87.0	49.9	65.8	–	61.1
PANORAMA	86.5	49.4	64.1	–	61.0
MFSD+SC	87.5	51.0	71.2	0.793	62.3
MFSD	87.5	49.4	65.8	0.789	60.7
2D-3D	82.9	46.5	60.5	–	57.5
CCCC					
CBR-ZFDR	84.7	83.8	88.6	0.898	90.2
CBR-ZFDR-A	84.7	78.0	85.9	0.884	87.7
ZFDR	84.7	58.8	72.6	0.814	74.7
PANORAMA+LRF	87.4	70.3	86.6	–	84.1
PANORAMA	87.9	66.3	81.2	–	81.2
2D-3D	87.4	60.2	75.8	–	76.7
MSB					
CBR-ZFDR	92.1	90.2	94.7	0.954	93.0
CBR-ZFDR-A	92.1	82.7	90.3	0.935	88.1
ZFDR	92.1	58.1	70.3	0.852	69.0
PANORAMA	94.5	62.6	75.4	0.880	74.6
MFSD+SC	90.3	65.7	76.7	0.868	74.8
MFSD	92.6	53.2	65.4	0.828	64.2
NIST					
CBR-ZFDR	87.8	85.7	92.9	0.930	89.8
CBR-ZFDR-A	87.8	76.6	87.4	0.907	84.3
ZFDR	87.8	55.0	68.1	0.821	66.2
PANORAMA+LRF	90.4	71.5	84.1	–	81.8
PANORAMA	90.8	63.4	77.6	0.869	74.6
2D-3D	88.1	55.6	72.1	–	68.6

"CBR-ZFDR" denotes our retrieval algorithm that utilizes the CBR algorithm described in Section 4 and the ZFDR shape descriptor presented in Section 3. "CBR-ZFDR-A" denotes a variation of CBR-ZFDR algorithm which uses addition to fuse the class and model distances. "ZFDR" means using only our hybrid shape descriptor ZFDR and do not use the CBR algorithm

proposed in our integrated distance rather than the addition approach used in MFSD to fuse the class and model distances, we can achieve apparently better performance. For example, for PSB our integrated distance outperforms the additive one by 14.3%, 8.0%, 3.6% and 6.3% in FT, ST, DCG and AP respectively and for NIST the

corresponding increments are 11.9%, 6.3%, 2.5% and 6.5%. Secondly, our hybrid descriptor ZFDR itself is comparable to the 2D-3D shape descriptor and it is close to PANORAMA on several datasets, especially on PSB, NTU and ESB. However, after applying our CBR algorithm, CBR-ZFDR achieves better performances than PANORAMA as well as PANORAMA+LRF and its performance is also better than the cluster-based method MFSD+SC. This indicates that after applying our CBR approach, we achive more improvement compared to the LRF and SC techniques. There are apparent improvements in either Precision-Recall plots or other performance metrics including FT, ST, DCG and AP. We also find that usually NN remains unchanged and this is because using the minimum distance as class distance will typically have no impact on NN. Therefore, our integrated distance keeps the nearest model in the beginning of the retrieval list while pushing the relevant models to the front of retrieval lists (FT, ST, DCG and AP are thus higher). One example to demonstrate this is shown in Fig. 7. We can see that the distance gap between the relevant class (horse) and other irrelevant classes (e.g. dog) also becomes bigger after adopting the CBR approach. This indicates that CBR pushes the irrelevant models to the rear part of the retrieval list. Thus, the retrieval errors (e.g. the two dog models m86 and m88) happened when only using the hybrid shape descriptor ZFDR itself are rectified. This is contributed to the utilization of the class information/distance.
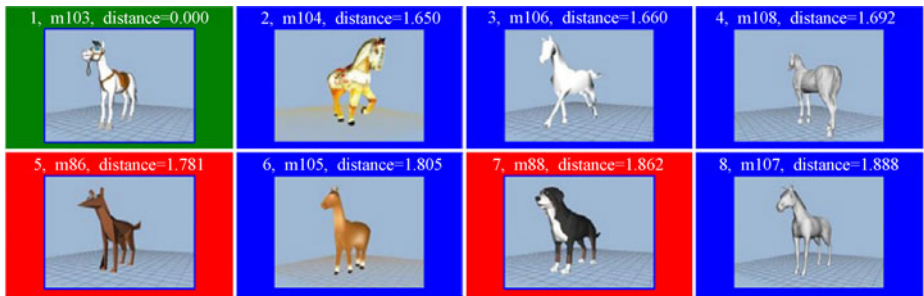
(a) ZFDR
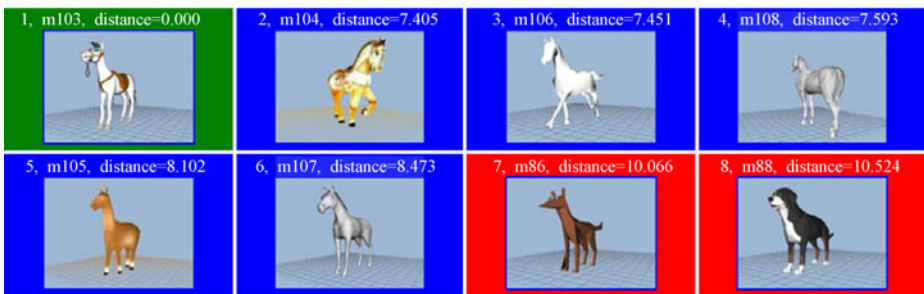
(b) CBR-ZFDR

Fig. 7 A retrieval example in the PSB database using ZFDR and CBR-ZFDR. *Green* Query models; *Blue* Correct class; *Red* Wrong class. The distances are shown above the images. In total, there are six models in the horse class that the query model belongs to

Assume C as the cardinality of the relevant class, our retrieval algorithm CBR-ZFDR has the ability to find most relevant models belonging to the same class as the query model in the front part (e.g., the top $(C-1)$ or at least $2(C-1)$ models) of the retrieval list, thus FT and ST are higher. Usually there are very few relevant models in the rest of the list, hence the recall remains almost unchanged in the rear parts of the Precision-Recall plots.

Our retrieval algorithm mainly comprises two processes: ZFDR feature extraction for a query model and feature matching with all the models in the database. ZFDR feature extraction requires rendering to compute the features Z, F, D and line-triangle intersection computation for the feature R, both of which depend on the number of triangles of the query model. Feature matching is a simple computation based on (1)–(7) and the matching time is proportional to the number of models.

Table 4 lists the timings of CBR-ZFDR on different databases based on a computer with an Intel Xeon CPU E5520 @2.27 GHz and 12.0 GB of RAM. We want to mention that the implementation is not optimized in terms of computational time. Nevertheless, our CBR-ZFDR algorithm already meet the requirements for interactive retrieval applications. Typically, the response time is less than 2 sec. Basically, only some small deviations in the rendering time happen due to different number of triangles in each model. Other processes mainly remain constant or are proportional to the number of models.

5.2.2 SHREC 2009 and PSB test vs train

In these two experiments, the query models are not selected from the target database, that is, the query set and the target set are two completely different datasets. For this purpose, we utilize the following two datasets:

– **SHREC 2009 NIST dataset** [14]: the dataset used in the Shape Retrieval Contest (SHREC) 2009 (generic track). It was constructed based on the NIST Generic Shape Benchmark (described in the beginning of Section 5), from which two models in each class were selected as query models and the rest as the target models. Therefore, there are 80 query models and 720 target models in the dataset.
– **PSB test and train datasets**. We use the test dataset as query dataset and the train dataset as the target dataset. Since the classes in the train and test datasets are not completely the same, we only consider the classes that exist in both datasets when measuring the retrieval performance.

Table 4 Timings information of CBR-ZFDR on different databases

Time	PSB	NTU	ESB	CCCC	MSB	NIST
t_f (s)	1.12	2.07	1.28	1.41	1.53	1.97
t_m (s)	0.11	0.04	0.09	0.05	0.05	0.09
t (s)	1.23	2.11	1.37	1.46	1.58	2.06

t_f, t_m, t denote the feature extraction time for a query model, feature matching time between the query model and all the models in the database, and response time for one query model, respectively

For the SHREC 2009, we compare with the top two methods in SHREC 2009 [14], which are the composite descriptor proposed by Lian et al. [27] ("Composite") and the multi-view depth line approach ("MDLA") proposed by Chaouch and Verroust-Blondet [8]. For PSB, we apply our CBR algorithm to both ZFDR and DESIRE shape descriptors for a comparative evaluation. We denote the CBR algorithm using the DESIRE shape descriptor as CBR-DESIRE. Figure 8 and Table 5 give their performance comparison. Obviously, our CBR-ZFDR approach has a better overall performance. Since the query models is not included in the target datasets, these experiments demonstrate the robustness of our retrieval algorithm. The experiments with DESIRE also demonstrate that our CBR algorithm is general and can be applied to any shape descriptors to evidently elevate their retrieval performance.

Fig. 8 Performance comparison: Precision-Recall plots of our retrieval algorithm CBR-ZFDR and other methods on SHREC 2009 NIST and PSB databases
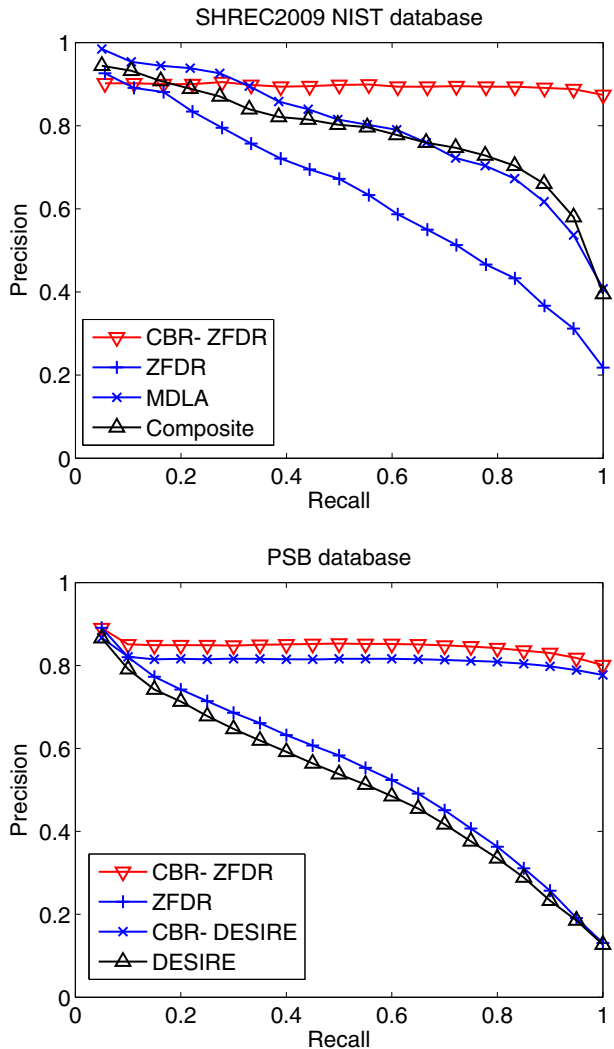
Table 5 Performance metrics for the performance comparison on SHREC 2009 NIST and PSB databases

Methods	NN	FT	ST	DCG	AP
NIST					
CBR-ZFDR	88.7	87.3	94.0	0.937	81.2
ZFDR	88.7	58.1	70.6	0.844	60.2
MDLA	96.3	73.0	84.8	0.917	73.8
Composite	92.5	72.4	84.4	0.904	72.9
PSB					
CBR-ZFDR	81.8	81.6	89.0	0.899	85.6
ZFDR	81.8	50.6	64.6	0.780	58.3
CBR-DESIRE	77.5	78.6	86.4	0.879	82.4
DESIRE	77.5	47.8	60.9	0.757	55.2

5.3 Partial 3D model retrieval

To demonstrate the versatility of our retrieval algorithm CBR-ZFDR, we also test and compare the performance of our algorithm in a partial matching scenario using the previously described WMB benchmark [42].

The goal is to retrieve similar subparts. To evaluate the partial similarity retrieval performance, we adopt the average Normalized Discounted Cumulative Gain (NDCG) [20] over all the query models. NDCG is defined by dividing the DCG of a partial retrieval algorithm by the ideal DCG related to the database. Thus, the range of NDCG will be [0, 1].

Because a query model (e.g. the query models in Figs. 10 and 11) is composed of several parts cut from models of different classes, the ground truth [42] classifies the target models into "relevant", "marginally-relevant" and "non-relevant" classes for every query model and assign relevance scores of 2, 1 and 0 for these three classes respectively. These scores are used to compute NDCG. To determine NDCG, we first compute the gain vector G. For example, the Centaur model in Fig. 10 is relevant to "four legs" and "human" classes and marginally relevant to "armadillo"

Fig. 9 Performance comparison: NDCG plots of our retrieval algorithm and other methods on SHREC 2007 Watertight database
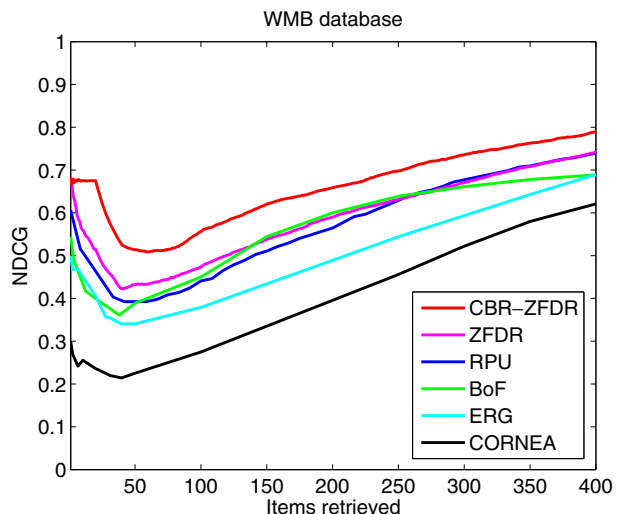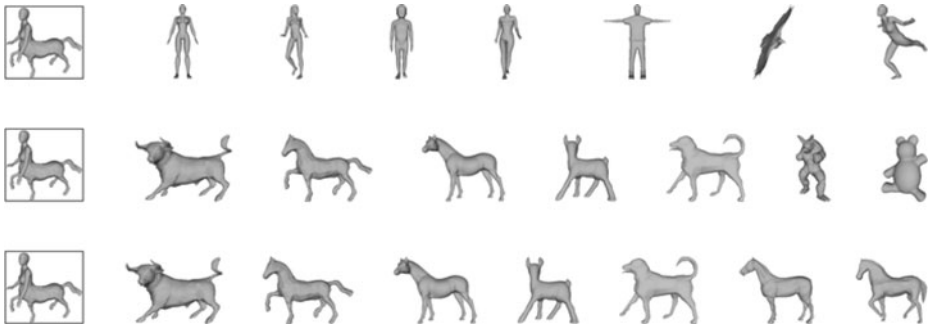
Fig. 10 A partial matching example showing the top-7 retrieval results using RPU (*1st row*), ZFDR (*2nd row*) and CBR-ZFDR (*3rd row*) methods. The first model in each row is the query model

and "teddy" classes. Then, the models in its retrieval list will be replaced by the corresponding scores to compute the gain vector G: (2, 2, 2, 2, 2, 0, 2), (2, 2, 2, 2, 2, 1, 1), (2, 2, 2, 2, 2, 2, 2) for the three rows respectively. The remaining steps of computing NDCG can be referred to WMB [42] and [20].

We compare with four previous partial retrieval algorithms: RPU, BoF, ERG and CORNEA. ERG and CORNEA are the only two participants in SHREC 2007 partial retrieval track [42] while the latest RPU and BoF algorithms outperform ERG and CORNEA. Figure 9 gives the NDCG performance comparison results. As can be seen, using only our ZFDR shape descriptor, we already can achieve an apparently better NDCG performance than RPU, ERG and CORNEA and an overall better performance than BoF. After adopting our CBR algorithm, we achieve an even better performance than any of the five methods.

Figures 10 and 11 show two retrieval examples using our CBR-ZFDR and ZFDR methods as well as the RPU method. Similarly, we can also see that CBR-ZFDR approach pushes the relevant models to the front of the retrieval lists. Additionally, we can also find that better than RPU, our methods can find the geometrically more relevant models first, which is more reasonable and easier for our understanding. The

Fig. 11 Another partial matching example showing the top-7 retrieval results using RPU (*1st row*) method and the top-9 retrieval results using ZFDR (*2nd row*) and CBR-ZFDR (*3rd row*) methods. The first model in each row is the query model

Fig. 12 CBR generality: Precision-Recall plots of our CBR algorithm with different shape descriptors on NIST, NTU, ESB and MSB databases
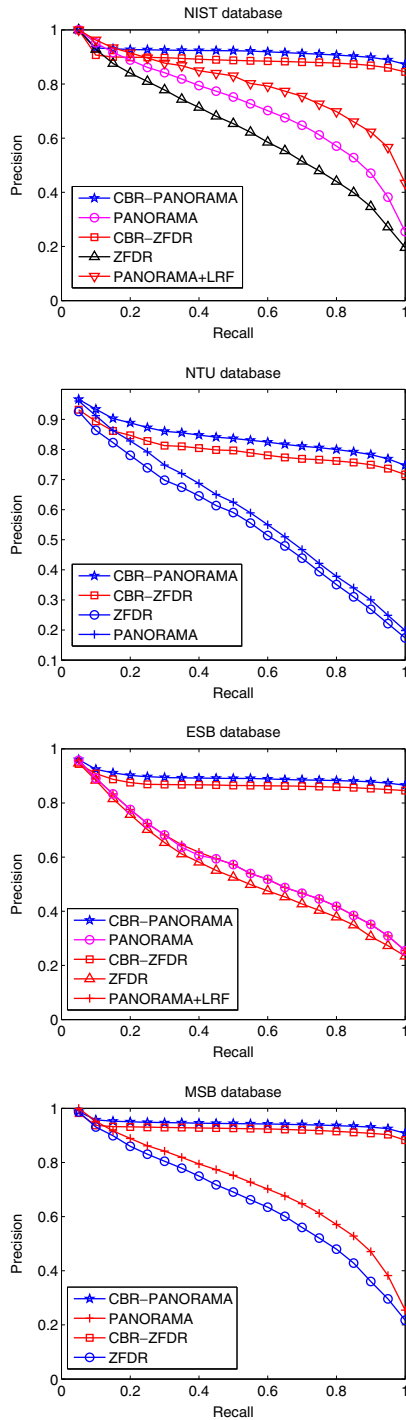
average time to process a query model using our CBR-ZFDR and ZFDR methods is about 2.8 sec (2.79 sec for feature extraction, 0.04 sec for feature matching). To some degree, this experiment demonstrates the superior performance of our retrieval algorithm CBR-ZFDR in terms of partial retrieval.

5.4 Generality of our CBR approach

Our CBR approach is general and can be used with any shape descriptors. By combining it with a better shape descriptor, we can achieve even better performances. To verify this, we replace the ZFDR shape descriptor with PANORAMA, which has better performances than ZFDR and perform experiments using the provided executable files [33] on four representative benchmarks: NIST, NTU, ESB and MSB. Figure 12 shows the performance comparison with CBR-ZFDR, as well as PANORAMA together with the local relevance feedback (LRF) technique, that is PANORAMA+LRF. The results show that CBR-PANORAMA apparently outperforms PANORAMA and it is also superior to PANORAMA+LRF. Thus, the performance improvement of our CBR approach is general and it is not dependent on the shape descriptors themselves. In addition, we can find that CBR achieves more apparent improvements compared to the LRF technique when both applied to PANORAMA, which also demonstrates the advantage of our CBR approach.

5.5 Limitations

Our CBR-ZFDR algorithm has achieved good performance on both generic and partial 3D model retrieval. However, it has some limitations. Firstly, ZFDR is not the best shape descriptor if we compare it with PANORAMA. Nevertheless, by incorporating the CBR algorithm, we can achieve a better performance than PANORAMA. Secondly, we only can directly apply our CBR-ZFDR algorithm to the already classified 3D model databases. If the 3D model database is unclassified, we can still apply our algorithm by first clustering the models in the database.

6 Conclusions and future work

In this paper, to improve the retrieval performance on a classified 3D model database, we have proposed a 3D model retrieval algorithm named CBR-ZFDR which is based on the proposed hybrid shape descriptor ZFDR and class-based retrieval (CBR) algorithm which makes use of the already available class information. ZFDR integrates a 3D model's both visual and geometric information from different aspects. By optimizing the choices of the four component features and carefully choosing the Scaled-L1 and Canberra distance metrics, we achieve better performances than the most related view-based shape descriptor Light Field and hybrid-based shape descriptor DESIRE. In addition, its performance is also close to the state-of-the-art shape descriptors on several databases. To further improve the retrieval performance, we define a new integrated distance to fuse the model distance and class distance in the CBR algorithm. We compute the integrated distance, which incorporates the class information of the database, by scaling the model distance using the class distance. Our CBR scheme is general, it can be applied to any shape

descriptors to evidently improve their retrieval performance. Extensive experiments demonstrated that: (1) with respect to generic retrieval, for most of the performance metrics, our results are better than the state-of-the-art methods on each of the six databases used in the experiments; (2) with respect to partial retrieval, it also shows an appealing performance both in terms of accuracy and speed: not only better than the two participants in SHREC 2007 partial retrieval track [42], but also outperforms the two latest shape descriptors RPU [40] and BoF [41].

Through experiments, we have shown that our retrieval algorithm is promising for retrieving models in a classified database. In order to enable us to apply our retrieval algorithm to unclassified databases, as future work, we would like to develop a method to group the models of unclassified 3D model databases and integrate it into our retrieval algorithm.

References

1. AIM@SHAPE (2010) SHREC contest. Home page http://www.aimatshape.net/event/SHREC/. Accessed 2011
2. Ankerst M, Kastenmüller G, Kriegel HP, Seidl T (1999) 3D shape histograms for similarity search and classification in spatial databases. In: Güting RH, Papadias D, Lochovsky FH (eds) SSD. Lecture notes in computer science, vol 1651. Springer, New York, pp 207–226
3. Axenopoulos A, Litos G, Daras P (2011) 3D model retrieval using accurate pose estimation and view-based similarity. In: De Natale FGB, Del Bimbo A, Hanjalic A, Manjunath BS, Satoh S (eds) Proceedings of the 1st international conference on multimedia retrieval, ICMR 2011. Trento, Italy, 18–20 April 2011
4. Ben-Chen M, Gotsman C (2008) Characterizing shape using conformal factors. In: Eurographics workshop on 3D object retrieval, 3DOR 2008, pp 1–8
5. Biasotti S, Marini S (2006) Sub-part correspondence using structure and geometry. In: Gallo G, Battiato S, Stanco F (eds) Eurographics Italian chapter conference, Eurographics, pp 23–28
6. Biasotti, S, Giorgi D, Marini S, Spagnuolo M, Falcidieno B (2007) 3D classification via structural prototypes. In: Falcidieno B, Spagnuolo M, Avrithis YS, Kompatsiaris I, Buitelaar P (eds) SAMT, Lecture notes in computer science, vol 4816. Springer, New York, pp 140–143
7. Bustos B, Keim DA, Saupe D, Schreck T, Vranic DV (2004) Using entropy impurity for improved 3D object similarity search. In: ICME. IEEE, Piscataway, pp 1303–1306
8. Chaouch M, Verroust-Blondet A (2007) A new descriptor for 2D depth image indexing and 3D model retrieval. In: ICIP (6). IEEE, Piscataway, pp 373–376
9. Chen DY, Tian XP, Shen YT, Ouhyoung M (2003) On visual similarity based 3D model retrieval. Comput Graph Forum 22(3):223–232
10. Cohen SD, Guibas LJ (1999) The Earth mover's distance under transformation sets. In: ICCV, pp 1076–1083
11. Cornea ND, Demirci MF, Silver D, Shokoufandeh A, Dickinson SJ, Kantor PB (2005) 3D object retrieval using many-to-many matching of curve skeletons. In: SMI. IEEE Computer Society, Los Alamitos, pp 368–373
12. Fang R, Godil A, Li X, Wagan A (2008) A new shape benchmark for 3D object retrieval. In: Bebis G et al (eds) ISVC (1). Lecture notes in computer science, vol 5358. Springer, New York, pp 381–392
13. Funkhouser T, Min P, Kazhdan M, Chen J, Halderman A, Dobkin D, Jacobs D (2003) A search engine for 3D models. ACM Trans Graph 22(1):83–105
14. Godil A et al (2009) SHREC '09 track: generic shape retrieval. In: Eurographics workshop on 3D object retrieval, 3DOR 2009, pp 61–68
15. Han EH, Karypis G (2000) Centroid-based document classification: analysis and experimental results. In: Zighed DA et al (eds) PKDD, Lecture notes in computer science, vol 1910, Springer, New York, pp 424–431
16. Hilaga M, Shinagawa Y, Komura T, Kunii TL (2001) Topology matching for fully automatic similarity estimation of 3D shapes. In: SIGGRAPH 2001, pp 203–212

17. Horn B (1984) Extended Gaussian images. Proc IEEE 72(12):1671–1686
18. Hou S, Lou K, Ramani K (2005) SVM-based semantic clustering and retrieval of a 3D model database. CAD Appl 2:155–164
19. Iyer N, Jayanti S, Lou K, Kalyanaraman Y, Ramani K (2005) Three-dimensional shape searching: state-of-the-art review and future trends. CAD 37(5):509–530
20. Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of ir techniques. ACM Trans Inf Sys 20(4):422–446
21. Jayanti S, Kalyanaraman Y, Iyer N, Ramani K (2006) Developing an engineering shape benchmark for CAD models. CAD 38(9):939–953
22. Kazhdan MM, Funkhouser TA, Rusinkiewicz S (2003) Rotation invariant spherical harmonic representation of 3D shape descriptors. In: Symposium on geometry processing, pp 156–164
23. Khotanzad A, Hong Y (1990) Invariant image recognition by Zernike moments. IEEE Trans Pattern Anal Mach Intell 12(5):489–497
24. Laga H, Nakajima, M (2008) Supervised learning of similarity measures for content-based 3D model retrieval. In: Tokunaga T, Ortega A (eds) LKR. Lecture notes in computer science, vol 4938. Springer, New York, pp 210–225
25. Leng B, Xiong Z (2011) Modelseek: an effective 3D model retrieval system. Multimedia Tools Appl 51:935–962
26. Lian Z, Godil A, Sun X (2010) Visual similarity based 3D shape retrieval using bag-of-features. In: Shape modeling international. IEEE Computer Society, Los Alamitos, pp 25–36
27. Lian Z, Rosin P, Sun X (2010) Rectilinearity of 3D meshes. Int J Comput Vis 89(2):130–151
28. Liu X, Croft WB (2004) Cluster-based retrieval using language models. In: 27th annual international ACM SIGIR conference (SIGIR 2004), pp 186–193
29. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
30. Ohbuchi R, Osada K, Furuya T, Banno T (2008) Salient local visual features for shape-based 3D model retrieval. In: Shape modeling international. IEEE, Piscataway, pp 93–102
31. Osada R, Funkhouser TA, Chazelle B, Dobkin DP (2001) Matching 3D models with shape distributions. In: Shape modeling international. IEEE Computer Society, Los Alamitos, pp 154–166
32. Papadakis P, Pratikakis I, Theoharis T, Passalis G, Perantonis SJ (2008) 3D object retrieval using an efficient and compact hybrid shape descriptor. In: Eurographics workshop on 3D object retrieval, 3DOR 2008, pp 9–16
33. Papadakis P, Pratikakis I, Theoharis T, Perantonis S (2010) PANORAMA: a 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. Int J Comput Vis 89(2–3):177–192. http://www.dis.uniroma1.it/~papadakis/
34. Shih JL, Chen HY (2009) A 3D model retrieval approach using the interior and exterior 3D shape information. Multimedia Tools Appl 43(1):45–62
35. Shilane P, Min P, Kazhdan MM, Funkhouser TA (2004) The Princeton shape benchmark. In: Shape Modeling International. IEEE Computer Society, Los Alamitos, pp 167–178
36. Siddiqi K, Zhang J, Macrini D, Shokoufandeh A, Bouix S, Dickinson SJ (2008) Retrieving articulated 3-D models using medial surfaces. Mach Vis Appl 19(4):261–275
37. Sundar H, Silver D, Gagvani N, Dickinson SJ (2003) Skeleton based shape matching and retrieval. In: Shape modeling international. IEEE Computer Society, Los Alamitos, pp 130–139
38. Tangelder JWH, Veltkamp RC (2008) A survey of content based 3D shape retrieval methods. Multimedia Tools Appl 39(3):441–471
39. Tatsuma A, Aono M (2009) Multi-fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. Vis Comput 25(8):785–804
40. Tierny J, Vandeborre JP, Daoudi M (2009) Partial 3D shape retrieval by reeb pattern unfolding. Comput Graph Forum 28(1):41–55
41. Toldo R, Castellani U, Fusiello A (2009) Visual vocabulary signature for 3D object retrieval and partial matching. In: Eurographics workshop on 3D object retrieval, 3DOR 2009, pp 21–28
42. Veltkamp RC, ter Haar FB (2007) SHREC 2007 3D retrieval contest. Technical report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University
43. Vranic D (2004) 3D model retrieval. PhD thesis, University of Leipzig
44. Vranic DV (2005) DESIRE: a composite 3D-shape descriptor. In: ICME. IEEE, Piscataway, pp 962–965
45. Wessel R, Blümel I, Klein R (2009) A 3D shape benchmark for retrieval and automatic classification of architectural data. In: Eurographics workshop on 3D object retrieval, 3DOR 2009, pp 53–56

46. Xu D, Li H (2007) 3D shape retrieval integrated with classification information. In: Fourth international conference on image and graphics, 2007. ICIG 2007, pp 774 –779
47. Zhang D, Luo G (2001) A comparative study on shape retrieval using Fourier descriptors with different shape signatures. In: Proc. of international conference on intelligent multimedia and distance education (ICIMADE01), pp 1–9
48. Zhang D, Luo G (2002) An integrated approach to shape based image retrieval. In: Proc. of the 5th Asian conference on computer vision (ACCV 2002), pp 652–657
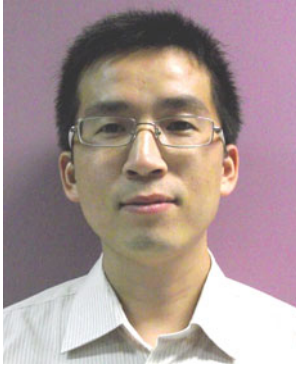
Bo Li received the BS and MS degrees in computer science and Technology from Xi'an Polytechnic University (China) in 2002 and Xi'an Jiaotong University (China) in 2005, respectively. He was a teaching assistant in the School of Science at Xi'an Polytechnic University from 2002 to 2007. From 2005 to 2007, he was a senior software testing engineer in Zhong Xing Telecommunication Equipment Company Limited. Since 2007, he has been a PhD student in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests include 3D model retrieval, shape matching and 3D modeling.

Henry Johan received his BS, MS and PhD degrees in computer science from the University of Tokyo (Japan) in 1999, 2001 and 2004, respectively. From 2004 to 2006, he was a post-doctoral fellow in the Department of Complexity Science and Engineering at the University of Tokyo. Since 2006, he has been an assistant professor in the School of Computer Engineering at Nanyang Technological University (Singapore). His research interests in computer graphics include rendering, animation and shape analysis. He is a member of ACM SIGGRAPH.