View-based techniques for 3D model retrieval

Li, Bo

2012

Li, B. (2012). View-based techniques for 3D model retrieval. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/50686

https://doi.org/10.32657/10356/50686

NANYANG TECHNOLOGICAL UNIVERSITY

View-Based Techniques for 3D Model Retrieval

Li Bo

A Thesis submitted to The Nanyang Technological University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in

School of Computer Engineering

2012

Abstract

With the increase in the number of available 3D models, the ability to accurately and efficiently search for 3D models is crucial in many applications such as Computer-Aided Design (CAD), on-line 3D model shopping and 3D game, movie and animation production. As a result, 3D model retrieval has become an important research area. In recent years, several typical algorithms that extract different types of 3D model features have been proposed. However, 3D model feature supporting multi-modal queries such as 3D models and 2D sketches is an important research direction which has little related work. In addition, 3D normalization is an important process in 3D model retrieval to extract rotation-dependent features and currently there still exists much room in terms of alignment accuracy and consistency. In this thesis work, we propose several algorithms to contribute solutions for the above issues. Motivated by the mechanism of human perception and multi-view vision, that is 3D shape information of a 3D object can be obtained based on multiple views, together with the retrieval performance comparison of previous retrieval work as well as the verifications of our proposed algorithms, we adopt a view-based approach which extracts features based on the rendered views of a 3D model.

The first part of our work is dealing with 3D pose normalization. A novel Minimum Projection Area-based (MPA) alignment method is proposed for pose normalization based on the idea of successively finding two perpendicular principal axes with minimum projection area. Experimental results demonstrate that MPA has a good performance in finding accurate axes; can robustly find a consistent pose for similar models and outperforms PCA, CPCA, and NPCA in terms of 3D model retrieval performance.

Next, we propose a view-based 3D model feature named view context to support both Query-by-Model and Query-by-Sketch retrieval. The view context of a particular view captures the distribution of visual information differences between this view and a set of arranged views. Experimental results demonstrate that our Query-by-Model retrieval algorithm outperforms the related view-based approach of Light Field. Comparative and evaluative experiments also demonstrate the effectiveness and robustness of our Query-by-Sketch retrieval algorithm which incorporates a 2D sketch-3D model alignment step based on view context and it significantly outperforms several latest sketch-based retrieval algorithms.

Finally, 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 and a class-based approach utilizing the existing class information of the database. We define an integrated distance metric which takes into account the class information. Extensive experiments demonstrate that our class-based retrieval approach apparently improves the retrieval performance and it is also general and can be used with any shape descriptors to improve their retrieval performance.

In conclusion, we have conducted substantial research in several aspects of 3D model retrieval techniques and proposed our solutions by mainly adopting a view-based framework. Specifically,

- We make substantial contribution to the research of 3D model alignment, an important aspect of 3D model retrieval techniques, by proposing the MPA algorithm which achieves better performance than existing alignment algorithms.
- We develop a 3D shape descriptor supporting multi-modal queries by defining the view context shape descriptor; and also foster the research of sketch-based 3D model retrieval by the first proposal of incorporating a 2D sketch-3D model alignment step for more accurate 2D-3D matching for the retrieval, as well as by the apparent improvement in the sketch-based retrieval performance.
- We perform the research of retrieval on a classified 3D model database. We propose a hybrid shape descriptor which is already comparable to or better than several related shape descriptors; we are also the first to utilize the class information available in the already classified 3D model database and develop a general class-based retrieval approach to obviously improve the retrieval performance of any shape descriptors.

Acknowledgments

This thesis could not be presented in this form without the help and support from so many people who are gratefully acknowledged here.

Firstly, I would like to take this opportunity to appreciate Dr. Henry Johan, my supervisor, for his so much precious time and continuous effort dedicated for training me of doing good research. His invaluable suggestions, honest criticism and kind encouragement are constant sources of inspiration throughout the research. His personal enthusiasm, highly professional academic standards and serious attitude to research work will guide and benefit me in the future. I am also grateful to Prof. Seah Hock Soon for providing me a good research environment, and Dr. He Ying, Dr. Qian Kemao and Dr. Zheng Jianmin for their invaluable lectures and advices. Deep thanks also go to Mr. Budianto Tandianus and Mr. Nicholas Mario Wardhana for their help and the useful discussions during our weekly research meetings with our supervisor. Here, I also want to acknowledge Mr. Wei Yuanmin and Mr. Iskandarsyah in relation to the collaboration in 3D model alignment research.

I am also sincerely thanks to Mr. Wang Dayong and Mr. Zhao Ming, Mr. Xia Jiazhi and Mr. Han Shuchu, Miss Xu Xiang and Miss Sun Qian for the rich and colorful days with all of you.

I should appreciate the financial support to my Ph.D. study provided by Nanyang Technological University. I am also thankful to our gameLab technician Mr. Yee Lee Poh for his kind help.

Finally, I would like to express my sincere gratitude to my parents and my wife Wang Juan for their constant support, encouragement and blessings.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

3D models consist of data stored in computer files to represent 3D objects and they can be used to support a wide variety of applications. At present, we use 3D models in a lot of fields. In industry, digitalization is conducted in many courses of the production system and one of them is to use computer to design products. In the design process, the shape information of the products is usually expressed as 3D models, such as 3D meshes. Moreover, in the fields of computer graphics such as visualization and entertainment, 3D models are indispensable. A 3D object is modeled as a 3D model in computer graphics and the 3D model functions as the most important data structure and the basis for the three 3D computer graphics-related research fields: 3D modeling (e.g. deformation, simplification, denoising), 3D rendering and 3D animation. That is, there are a lot of needs for 3D models.

With the progress in 3D scanning techniques and the availability of affordable 3D scanners, the number of 3D models has grown in an exponential way. Then, there exist a lot of 3D model retrieval applications, such as 3D prototyping based on automatic Computer-Aided Design (CAD) and existing CAD models, on-line 3D model shopping, and 3D game, film and animation production, where we need to search and select 3D models from existing

3D model databases. Therefore, it is very important to have an effective, efficient and easy way to retrieve relevant 3D models from 3D model databases.

To find relevant 3D models, we usually first extract a 3D shape descriptor to depict a 3D model. During the retrieval, shape descriptors of different models are then compared with each other in the feature space for ranking. In this stage, different distance metrics from the simplest $L1$ distance to the complicated ones like Earth Mover's distance (EMD) [24, 106] and Wasserstein [149] distance metrics may be employed to compare two 3D shape descriptors. In the case of extracting a rotation-dependent shape descriptor, if not utilizing certain pair-wise matching techniques such as the spherical correlation method in [114] and the iterative comparison method used in the Light Field descriptor [23], we often need performing pose normalization first before the feature extraction. This is because 3D models are created in arbitrary scale, orientation and position in 3D space. Therefore, pose normalization of 3D models is important in many computer graphics applications such as 3D model retrieval, 3D model recognition and 3D visualization. The goal of 3D model pose normalization is to transform a model into a canonical coordinate frame, where the representation of the model is independent of its scale, orientation and position. An ideal canonical coordinate frame of a 3D model is defined as a coordinate frame whose axes are parallel to the front-back, left-right and top-bottom directions of the model. The normalization process includes three steps: translation, scaling and alignment. The important and difficult step is 3D model alignment and the traditional method to deal with this is Principle Component Analysis (PCA) [71]. To improve the accuracy, various alignment algorithms based on the idea of PCA have been proposed, such as Continuous PCA (consider the area of each face) [190] and Normal PCA (consider the normal of each face) [146]. Other approaches utilize symmetry information [22], virtual contact area (VCA) [145], or projection area [101, 125].

Currently, 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. Geometry-based techniques extract features based on the distribution of a model's geometric elements or topological structures while view-based techniques characterize a 3D model based on its rendered view images. Hybrid techniques utilize both techniques. Recently, several typical retrieval algorithms extracting different kinds of 3D model features have been proposed. For example, shape histogram [4], 3D shape context [80], shape distribution [135], moment [35] and 3D harmonics [78] are some examples for geometry-based techniques. Multiple view descriptor [68], Light Field [23] and salient local visual feature-based retrieval method [131] are instances for view-based techniques. Two representatives of hybrid approaches are DESIRE [191] and PANORAMA [142].

However, till now it is still difficult to find a shape descriptor which performs well on all types of 3D model benchmarks. Some main challenges include: (1) the great diversity of models within a class; (2) the different initial positions for even one model; (3) the different poses and deformations for some special types of models like non-rigid and deformable ones; and (4) the intrinsic gap between the geometrical properties of a 3D model and its semantic class. On the other hand, as mentioned before, alignment of 3D models is necessary to utilize rotation-dependent 3D model features for rotation-invariant retrievals. But the currently available 3D alignment algorithms still have much room for further improvement in terms of finding accurate alignment axes, aligning similar models in similar poses under different conditions such as model variations, noise and initial poses, as well as the retrieval performance improvement for a rotation-dependent shape descriptor. Last but not least, 3D model retrieval supporting multi-modal queries such as text, 3D models, 2D/3D sketches is another important research direction because of its versatility and wider applications by considering end users' diverse input preferences and modalities to meet their requirements.

Some challenges of 3D model retrieval techniques are listed here according to our knowledge.

- Extract 3D model features that outperform others for specific applications.
- Develop a 3D shape descriptor that supports different types of queries such as 3D models and 2D sketches.
- Devise better 3D model alignment algorithms that improve the performance of 3D normalization for diverse types of models. The alignment accuracy and consistency within a class of 3D models usually have nontrivial influence on the retrieval performance of a rotation-dependent shape descriptor.
- Develop a universal search engine or platform to facilitate the development of various applications dependent on a 3D model retrieval module. The engine has versatile capabilities and it can select appropriate features, retrieval rules and algorithms for different applications.
- Build professional 3D benchmark databases for different application fields, such as biology, architecture and mechanics. This is important for comparing different kinds of retrieval algorithms to help people to select appropriate ones. This is because different algorithms/shape descriptors often perform better in certain fields or for certain types of 3D models.
- Use interdisciplinary techniques such as machine learning and computer vision to develop 3D model retrieval algorithms.

Our project concentrates on providing solutions for the first three challenges: (1) new and better performing shape descriptors; (2) multi-modal 3D model retrieval; and (3) better 3D model alignment methods. We will present the details in the next section.

To test a retrieval algorithm, we selected the following seven representative standard benchmark databases,

- Princeton Shape Benchmark (**PSB**) [167]. It contains 1814 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 6.4.2.2 and for other cases we only use the test dataset.
- Engineer Shape Benchmark (**ESB**) [67]. This is a CAD model database which contains 867 models, classified into 45 classes.
- National Taiwan University database (**NTU**) [23]. This database contains 1833 3D models and only 549 3D models are grouped into 47 classes and the rest 1284 models are assigned as the "miscellaneous".
- Konstanze 3D Model Benchmark (**CCCC**) [190]. CCCC comprises 1838 models and 473 models are grouped into 55 types and other 1365 models are unclassified.
- McGill 3D Shape Benchmark (**MSB**)[168]. 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**) [3, 36]. 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.
- AIM@Shape Watertight Models Benchmark (**WMB**) [187]. The dataset has 400 watertight models, divided into 20 classes, 20 each.

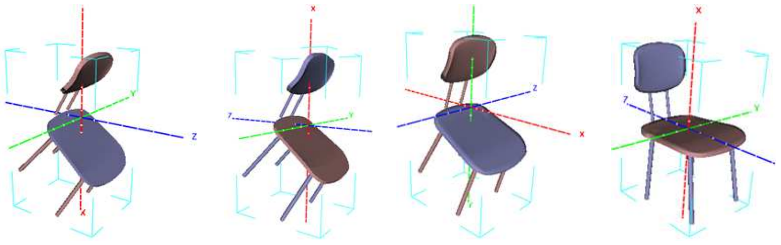
To evaluate the 3D model retrieval performance, we employ seven metrics including Precision-Recall, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), E-measure (E), Discounted Cumulative Gain (DCG) [167] and Average Precision (AP). Precision indicates how much percentage of the top K models belongs 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. E is used to measure the performance of the retrieval results with a fixed length, e.g. the first 32 models. It combines both the precision P and recall performance R: $E = 2/(\frac{1}{P} + \frac{1}{R})$. DCG is another accuracy measure of the retrieval list based on the idea that relevant models with different positions will have different weights. The nearer the relevant models are, the higher the weights will be assigned. 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.

1.2 Overview of Our Research

In this research, we have proposed algorithms to tackle the first three aforementioned challenges by adopting a view-based approach. Usually, human perception and understanding of a 3D object are based on several views of the object. According to the human vision theory proposed by David Marr [117], vision proceeds from a 2D visual array as input to a 3D description of the world as output. It includes three stages: (1) a 2D or primal sketch of a scene comprising of fundamental features like edges and regions; (2) a 2.5D sketch of the scene, where textures, colors and shadings provide depth information; (3) a 3D model, where the scene is presented in our mind in the form of a continuous 3D map. Since the first stage is dealing with the most important and basic information of a scene, the 3D shape information existing in the 2D primal sketch provides us the most crucial and fundamental information during the process of human perception of a 3D object while the depth information functions as a complement for better analysis and understanding. Therefore, we mainly concentrate on the 3D shape information extraction based on multiple 2D views of a 3D object for our 3D model retrieval research. According to the field of multiple view geometry [53], we can estimate the depth information of an object based on the pixel disparity information existed in its multiple views. Thus, based on several views captured around an object, we can estimate the 3D locations of the sample points on the surface of the 3D object, that is we can percept its shape information. Therefore, we can use appropriate number of sample views to approximately represent a 3D object.

The reason of our selection of view-based approach is also based on the fact that this type of technique often achieves better performance compared to many geometric-based methods. It is also proved by our proposed algorithms. For example, based on the observation that for many objects, one of their canonical views (that is, either front-back view or left-right view or top-bottom view) has a minimum projection area compared to the other arbitrary views of the objects, we propose a view-based 3D model alignment approach which outperforms commonly used approaches such as Principal Component Analysis (PCA) [49, 71], Continuous Principal Component Analysis (CPCA) [190] and Normal Principal Component Analysis (NPCA) [146]. To support multi-modal queries including both 2D and 3D queries, we propose a view-based 3D shape descriptor named view context. In addition, hybrid approach is also used in proposing a 3D shape descriptor in our project: motivated by the fact that different types of features are effective in characterizing different types of models, we develop hybrid feature ZFDR by taking the advantages of both view-based and geometry-based techniques. By adopting a view-based framework, we have proposed four algorithms for the first three challenges:

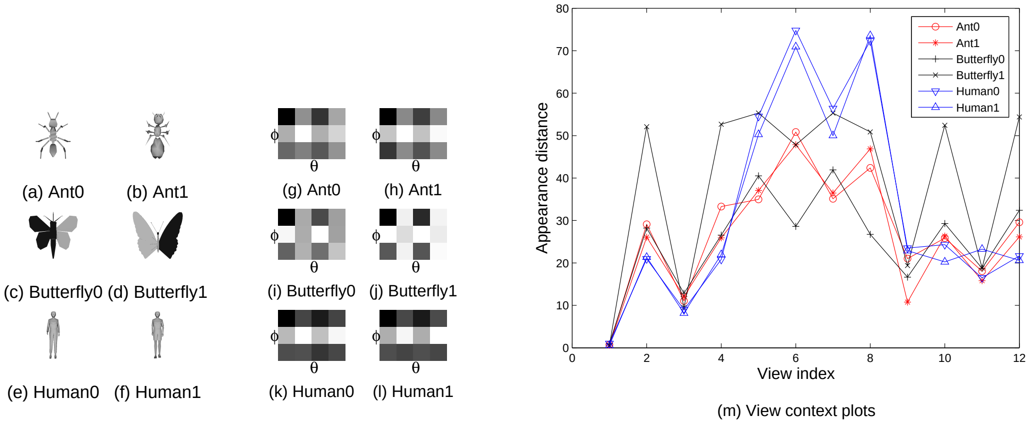
- **Minimum Projection Area (MPA) Based 3D Model Alignment Algorithm**. This is for the third challenge of developing better 3D model alignment algorithms to improve retrieval performance of rotation-dependent shape descriptors. The basic idea of our alignment algorithm MPA is successively finding three perpendicular principle axes with minimum projection areas to align a model: the first principle axis gives the minimum projection area when we perform an orthographic projection of the model in the direction parallel to this axis, the second axis is perpendicular to the first axis and gives the minimum projection area, and the third axis is the cross product of the first two axes. We devise an optimization method based on Particle Swarm Optimization (PSO) [34] to efficiently find the axis with minimum projection area. For application in retrieval, we further perform axis ordering and orientation in order to align similar models in similar poses. We have tested MPA on several standard databases which include rigid/non-rigid and open/watertight models. Experimental results demonstrate that MPA has a good performance in finding alignment axes which are parallel to the ideal canonical coordinate frame of models and aligning similar models in similar poses under different conditions such as model variations, noise and initial poses. In addition, it achieves a better 3D model retrieval performance than several commonly used approaches such as CPCA, NPCA and PCA. One example to demonstrate this is shown in Fig. 1.1.



(a) PCA (b) CPCA (c) NPCA (d) MPA

Figure 1.1: An example demonstrating that our MPA can find more accurate alignment axes than PCA, CPCA and NPCA.

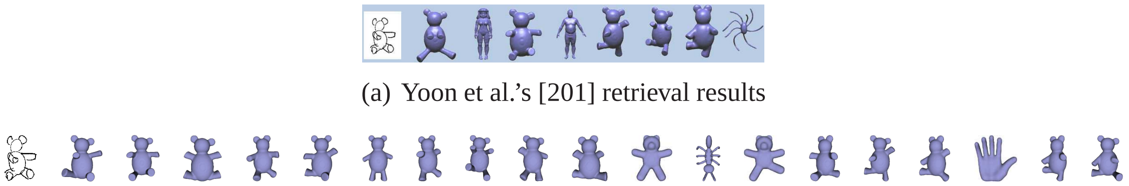
- **View Context 3D Shape Descriptor and Related Retrieval Algorithm**. It is a new view-based shape descriptor dedicated to the first challenge. View context captures the shape appearance deviation of a 3D model by measuring the difference between the current view and a set of arranged views. It can differentiate models based on the fact that similar models have similar view contexts and the view contexts of models from different classes are usually distinctively different. Fig. 1.2 shows the view context features of several example models. We develop a combined shape descriptor based on view context. Experimental results demonstrate that this combined shape descriptor outperforms the related view-based Light Field descriptor.



(a) Ant0 (b) Ant1 (c) Butterfly0 (d) Butterfly1 (e) Human0 (f) Human1 (g) Ant0 (h) Ant1 (i) Butterfly0 (j) Butterfly1 (k) Human0 (l) Human1 (m) View context plots

Figure 1.2: View context [95] of six models. (a)∼(f): six models; (g)∼(l): matrix representation of the view context of the initial poses of the models in (a)∼(f). (m): view context plots. 12 arranged views are selected.

- **Sketch-Based Retrieval (SBR) Algorithm by Incorporating 2D-3D Alignment Step.** It is dedicated to the second challenge of multi-modal retrieval to support other queries such as 2D sketches/images besides 3D models. Besides the differentiation property of view context for different models, we found a new property of it: view contexts of different views of the same model are also often different. This property can be utilized to distinguish different views of the same model, which is employed



(a) Yoon et al.'s [201] retrieval results

(b) Our retrieval results

Figure 1.3: A sketch-based retrieval example indicating our better performance than the state-of-the-art sketch-based retrieval algorithm proposed by Yoon et al. [201]. The left most of each row is the query sketch.

for integrating an efficient 2D sketch-3D model alignment step in our sketch-based retrieval algorithm.

Our sketch-based retrieval algorithm is based on the view context 3D model feature and 2D relative shape context matching. To enhance the accuracy of 2D sketch-3D model correspondence as well as the retrieval performance, we propose to align a 3D model with a query 2D sketch before measuring their distance. Based on view context, we can efficiently select some candidate views from a set of densely sampled views of the 3D model to align the sketch and the model based on their view context similarities. Our sketch-based retrieval algorithm is composed of two stages which are precomputation and retrieval. The retrieval stage comprises two steps which are 2D-3D alignment and 2D-3D matching. Comparative and evaluative experiments based on hand-drawn and standard line drawing sketches demonstrate the effectiveness and robustness of our approach and it significantly outperforms several latest sketch-based retrieval algorithms. One example indicating our better performance than the state-of-the-art sketch-based retrieval algorithm proposed by Yoon et al. [201] is demonstrated in Fig. 1.3.

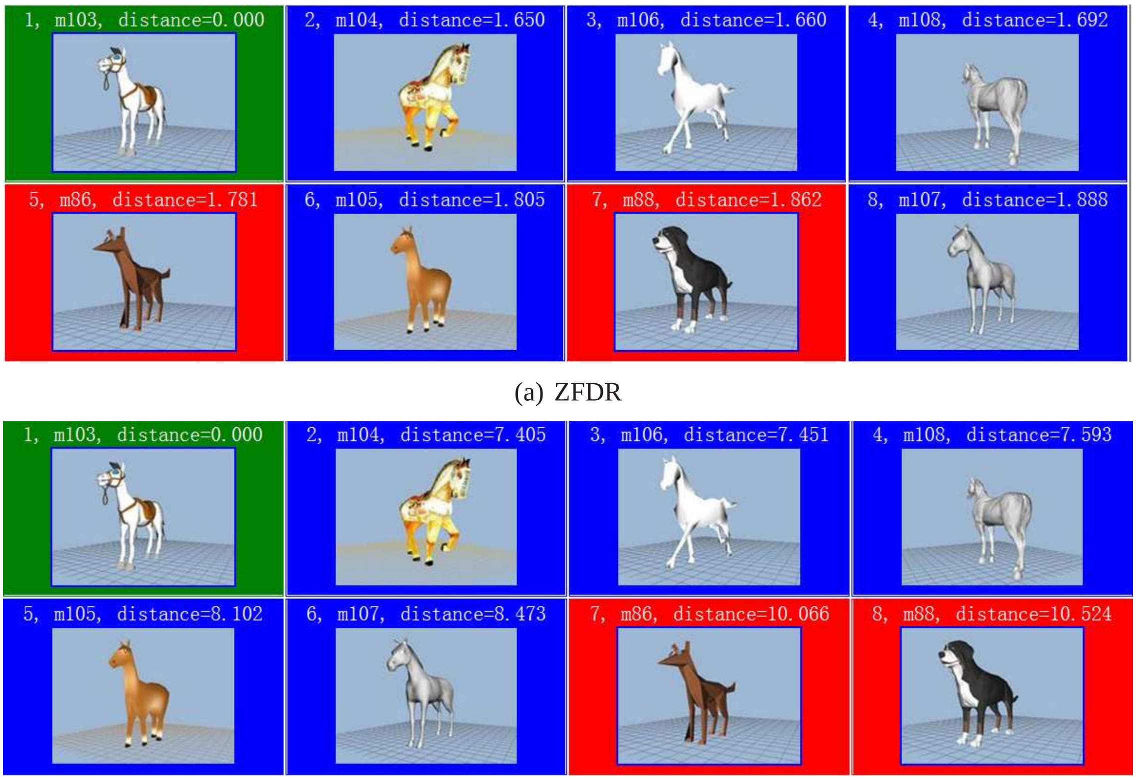
- **Class-Based Retrieval (CBR) Algorithm Utilizing Hybrid Features.** It is designed for the second challenge of new retrieval framework. It is dedicated for the retrieval on a classified 3D model database by adopting a new retrieval framework by taking into account the class information and a proposed hybrid shape descriptor named ZFDR. An integrated distance metric is defined to combine the model distance between the query model and a target model in a class of a database as well as the class distance between the query model and the class. The hybrid feature ZFDR is composed of four components which characterize a 3D model from different aspects and it itself is already comparable to or better than several related shape descriptors. Fig. 1.4 shows a retrieval example using pure ZFDR descriptor and our complete class-based retrieval algorithm CBR-ZFDR.

Our CBR approach is general and 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.

1.3 Contributions

Motivated by the existing challenges and previous work in 3D model retrieval research, we have done substantial research in several aspects and proposed our solutions by mainly adopting a view-based framework. Specifically, we contribute in the following three aspects:

- We propose a novel Minimum Projection Area-based (MPA) alignment method for pose normalization. It outperforms current available 3D alignment algorithms including PCA, CPCA and NPCA and we also found it improves the retrieval performance of rotation-dependent shape descriptors such as modified Light Field descriptor. In a word, we have made substantial contribution to the research of 3D model alignment,



(a) ZFDR

(b) CBR-ZFDR

Figure 1.4: A retrieval example in the Princeton Shape Benchmark (PSB) [167] database using ZFDR and CBR-ZFDR to indicate that CBR pushes the irrelevant models to the rear part of the retrieval list. 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.

an important aspect of 3D model retrieval techniques, by proposing an alignment algorithm with better performance.

- We develop a new 3D model feature named view context which supports multi-modal queries. We propose a view context descriptor for retrieval using 3D model queries and a 2D sketch-3D model alignment algorithm for retrieval using sketch queries. Our proposed view context-based shape descriptor achieves a better performance than the related view-based Light Field descriptor. Our sketch-based retrieval algorithm based on the view context 3D model feature and 2D relative shape context matching also achieves significantly better performance than the state-of-the-art sketch-based retrieval algorithm. In a word, we contribute in the following two related aspects: (1) developing a 3D shape descriptor supporting multi-modal queries by defining the view context shape descriptor; (2) fostering the research of sketch-based 3D model retrieval by the first proposal of incorporating a 2D sketch-3D model alignment step for more accurate 2D-3D matching for retrieval, as well as by the apparent improvement in the sketch-based retrieval performance.

- We advise a new retrieval algorithm utilizing Class-Based Retrieval (CBR) framework and hybrid features. Our proposed hybrid shape descriptor ZFDR achieves comparable or better performance compared to several related shape descriptors; we are also the first to utilize the class information "already" available in the classified 3D model database to develop a general class-based retrieval approach which can be applied to any shape descriptors to improve their retrieval performance. Our extensive experiments also demonstrate that the retrieval performance is evidently improved after adopting our CBR framework.

1.4 Thesis Organization

The thesis is organized as follows.

- Chapter 2 reviews the related work in 3D model retrieval and 3D model alignment techniques.
- Chapter 3 first presents the Minimum Projection Area-based (MPA) 3D model alignment algorithm and then shows two evaluation experiments with respect to axes accuracy and retrieval performance.

- Chapter 4 first presents the idea of view context 3D model feature and based on it we propose a view context shape descriptor to depict the visual appearance deviation feature of a 3D model. Then, a 3D model retrieval algorithm using the view context shape descriptor is explained. Finally, the retrieval experiments results are demonstrated.
- Chapter 5 proposes a sketch-based retrieval algorithm by incorporating an efficient 2D-3D alignment. We first introduce the 2D and 3D feature extraction. After that, our sketch-based 3D model retrieval algorithm which utilizes the proposed view context feature for an efficient 2D-3D alignment is proposed and finally evaluative and comparative experiments are conducted.
- Chapter 6 presents a new retrieval algorithm utilizing hybrid features and class information for query on classified 3D database. The hybrid shape descriptor ZFDR is first presented followed by the details of our class-based 3D model retrieval algorithm CBR-ZFDR which uses ZFDR. Then, extensive experiments, for generic and partial retrieval, on seven standard 3D databases are demonstrated.
- Chapter 7 contains the conclusions and future work. We first draw a conclusion on the thesis work and then propose two new directions for the research topics of sketch-based 3D model retrieval and partial similarity 3D model retrieval as the future work.

Chapter 2

Related Work

In this chapter, we present a literature review on research areas related to our project. The main objective of our project is to propose solutions to deal with the challenges in 3D model retrieval. Therefore, we first review 3D retrieval techniques in Sections 2.1$\sim$2.3: generic, partial and sketch-based 3D model retrieval. For sketch-based 3D model retrieval, we propose to estimate the pose information of a 3D model to correspond to a 2D sketch for an accurate 2D sketch-3D model correspondence. Thus, for a comparison, in Section 2.3 both the 2D sketch-3D model alignment and generic 2D image-3D model alignment techniques are also reviewed. Since 3D model alignment is often needed in 3D model retrieval techniques, we review existing 3D model alignment methods in Section 2.4. We compare the algorithms we have proposed with the related work in Section 2.5.

2.1 Generic 3D Model Retrieval

Natraj et al. [63] and Tangelder et al. [176] reviewed and classified current typical 3D model retrieval techniques in their respective survey. The existing generic 3D model retrieval techniques can be classified into three categories: geometry-based, view-based and hybrid techniques. Geometry-based techniques use the distribution of geometric elements, such as vertices and faces, or some intrinsic topological structures to characterize the features of 3D models while view-based techniques extract features based on the rendered view images. Hybrid techniques employ both geometry-based and view-based techniques.

2.1.1 Geometry-Based Techniques

Most of the previous work in 3D model retrieval belongs to the geometry-based techniques. This type of 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 [135] and shape histogram [4], or local, such as, 3D shape context [41, 60, 80], Extended Gaussian Images (EGI) [55] and conformal factor [11].

Shape distribution focuses on geometric shape function that measures the distance between two random points on the surface of a model. Shape histogram [4] is an extension of the 2D shape matching techniques to 3D. For each surface point, it computes the distance from the center of mass and spherical angle. The distance distribution is encoded into a histogram, whose bins are formed according to three types of 3D space partitioning methods: Shells (only use distance), Sectors (only use spherical angle) and Spider Web (use both). 3D shape context is based on the idea of 2D shape context [10], which is a log-polar histogram and defines the relative distribution of other points with respect to a point. Based on the aforementioned different 3D space partitioning methods, three forms of 3D shape context have been proposed accordingly: the support of volume 3D shape context [41] based on the Shell model, the 3D cylindrical shape context [60] using the Sector model and the 3D point shape context [80] utilizing the Spider web model.

Ben-Chen and Gotsman [11] proposed a 3D shape descriptor named conformal factor which depicts the amount of local work involved to transform a model into a sphere. Graph-based methods [54, 173] use skeleton or topology graph to represent a 3D model and employ a graph matching method to measure the distance between two graphs. Kazhdan et al. [78] employed spherical harmonics to decompose a spherical function (e.g. shape histogram) into orthogonal component while preserving the norms, thus there is no need to align 3D models' orientations when computing their similarities. It can make a rotation-dependent shape descriptor based on spherical function to be rotation-invariant, but it also decreases the discrimination power of retrieval.

Recently, Shih and Chen [166] proposed an **a**ngular **r**adial **t**ransformation-based **e**levation **d**escriptor (ART-ED) and a shell grid descriptor to encode the external and internal shape information of a 3D model, respectively. By employing Poisson equation to depict a 3D object, Pan et al. [139] proposed a Poisson histogram descriptor to depict the structural feature of a 3D model.

During the past few years, geodesic distance and spectrum analysis approaches have received much attention, especially for dealing with the retrieval of non-rigid 3D models. We give a brief review for them as follows.

2.1.1.1 Geodesic Distance-Based Descriptors

Geodesic distance is an inelastic deformation invariant distance metric, thus popular for the analysis and recognition of non-rigid objects. Typically, the extracted geodesic distance-based feature is a geodesic distance matrix (GDM) measuring the distances among a set of points uniformly sampled on the surface of an object.

To deal with deformable 3D model retrieval, Smeets et al. [169] proposed a modal representation method based on the singular value decomposition (SVD) of the geodesic distance matrix of a 3D model. They utilized several largest eigenvalues of a GDM as the shape descriptor and this modal approach outperforms the direct GDM histogram comparison method. To improve the retrieval performance further, in [170], they advised to combine

GDM with diffusion distance tensors (DDT) to utilize their respective advantages. They found that GDM has advantages in differentiating small inter-class variations while DDT performs better with respect to noise and topology robustness.

Rabin et al. [149] devised a geodesic distance-based 2D and 3D shape retrieval algorithm. They employed several global or local geodesic distance-based features (e.g. geodesic distance distribution and geodesic quantile measures) to form a hybrid feature set comprising several distributions and utilized Wasserstein metric [188] to measure the distance between two joint distributions.

Different from the above algorithms which use a 2D geodesic distance matrix to represent a 3D model, Hamza and Krim [50] proposed to use a geodesic shape distribution. The idea is similar to shape distribution [135] but they adopted the kernel density estimation (KDE) to associate with the geodesic distance shape distribution of the model to approximate its probability density function and utilized Jensen-Shannon divergence distance to measure the dissimilarity of two probability distributions.

2.1.1.2 Laplace-Beltrami Spectrum Analysis Method

Spectrum analysis on 3D models has been steadily become an important research field in the community of geometry processing and analysis. Two good surveys about spectral geometry processing methods are presented by Zhang et al. [207] and Lévy [93], respectively.

The pioneer work of applying Laplace-Beltrami spectrum for shape analysis is proposed by Reuter et al. [155]. They defined a 3D shape descriptor which they called “Shape-DNA”. It is composed of the eigenvalues of the Laplace-Beltrami operator of a 3D model. Another pioneering work which applies spectrum analysis on articulated or non-rigid models is proposed by Jain et al. [64, 65]. They inspire following work in several related fields: (1)

shape matching and retrieval [59, 154, 169]; (2) shape analysis: [46]; (3) shape segmentation [30, 103, 152, 153] ; (4) shape correspondence [65, 183]; and (5) shape registration [152].

Besides the standard definitions of Laplace-Beltrami operators, Wu et al. [196] proposed a symmetric mean-value Laplace-Beltrami representation based on manifold harmonic analysis. They extended the Laplace-Beltrami operator representations to a new representation which has a better reconstruction quality. Based on this, they further performed spectral analysis on a local region and combined it with the global version to form a hybrid one for both global and partial similarity matching. The basic framework for feature extraction is still as the same as its precedents, but utilizes a pyramid matching method for the feature matching process based on the histogram-based representation. Unfortunately, they did not perform the algorithm on a database level and only showed one retrieval example. In addition, several methods also have been proposed for efficient Laplace-Beltrami spectrum computation for a 3D mesh, such as [9, 120, 182].

2.1.1.3 Heat Kernel Descriptors

Heat kernel $k_t(x,y)$ is defined as the probability/amount of the heat that has been transferred from a unit heat source point x to point y. It is the fundamental solution to the heat equation, an important function to study heat conduction and diffusion. We can also use heat kernel for Laplacian spectrum analysis based on the relationship between their eigenvalues and eigenfunctions: ρ_i=$e^{-t\lambda_i}$, where ρ_i and λ_i are the eigenvalues of heat kernel and Laplace-Beltrami operators respectively, and they also have the same eigenfunctions ϕ_i for the corresponding eigenvalues ρ_i and λ_i.

Sun et al. [172] first proposed a novel shape descriptor named Heat Kernel Signature (HKS) in 2009. HKS measures how much percentage of the heat will transfer from a point on the surface of a model to other points at time t. It has many good properties, such as isometry-invariant, multi-scale, robust and informative. The Heat Kernel Signatures at all the points of a model can uniquely define the model up to isometry. HKS can be utilized for many applications, such as shape correspondence [138], shape registration, shape retrieval and partial matching [31].

Dey et al. [31] further applied persistent homology to detect a stable set of feature points with maximum HKS and a feature vector of the combination of its HKS values at different time scales. Shape distance is defined as the minimum L1-distance between two sets of feature vectors. One advantage of this shape descriptor is that it is applicable for either partial, incomplete or complete models.

Ovsjanikov et al. [137] developed a HKS-based scale-invariant shape descriptor in 3D space by an analogy to the scale invariant feature transform (SIFT) [110] in 2D space. They applied the framework of Bag-of-Features (BoF) to the HKS feature space and further integrated the spatial relationship constraints into it to develop a spatially-sensitive Bag-of-Features for non-rigid shape retrieval.

Recently, Bronstein and Kokkinos [16] developed a scale-invariant heat kernel signature (SI-HKS) for non-rigid shape recognition. It apparently outperforms HKS and "Shape-DNA" on a database named ShapeGoogle [137], which comprises both non-rigid and rigid models. Bronstein et al. [15] explored the applicability of diffusion distances within the Gromov-Hausdorff framework. Raviv et al. [150] proposed a volumetric heat kernel by extending HKS to an isometry-invariant volumetric descriptor.

To accelerate the computation of heat kernels on a mesh, a multi-resolution approach utilizing the heat kernel in a 2D space is proposed by Vaxman et al [184]. It shortens the computation time needed by traditional methods while maintaining a good approximation. This approach is specially good for heat kernel computation for models with many details.

2.1.2 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 and the visual similarities between the view images of different models are compared with each other to measure model differences. Multiple view descriptor [68], Light Field descriptor [23] and our proposed view context shape descriptor (Chapter 4) belong to this category. Multiple view descriptor classifies models by comparing the views rendered from the primary, secondary and tertiary viewing directions of principle axes after an alignment with PCA [71]. Chen et al. [23] proposed the Light Field descriptor to define the distance of two models as the minimum distance between their 10 corresponding silhouette views, rendered from the vertices of a dodecahedron using the orthographic projection. An alignment process is also proposed to find this minimum distance and it is simplified by rotating a camera system which consists of 20 cameras set on the vertices of a regular dodecahedron. Therefore, essentially, the Light Field descriptor defines a 3D model alignment method to compare models. The features in each image are encoded using the Zernike moments and Fourier descriptor.

Chaouch and Verroust-Blondet [21] proposed a multi-view depth line approach (MDLA) to represent a 3D model. 20 depth images, rendered from the vertices of a regular dodecahedron, are coded into sequences of symbols and then the dynamic programming method is employed to measure their differences. It can achieve a better performance than the famous Light Field descriptor. Salient local visual feature-based retrieval method [131] adopts the Bag-Of-Features (BoF) framework to accumulate the Scale Invariant Feature Transform (SIFT) [110] features of multiple depth views into an occurrence histogram to represent a 3D model. It first renders a set of depth view images for a 3D model and then extracts the multi-scale local features of these views using SIFT, which is invariant to translation, scaling and rotation. Finally, it fuses all these local features into a histogram using the Bag-Of-Features (BoF) approach, which accumulates the visual words (extended from the bag-of-words in text retrieval) of multiple views into a single occurrence histogram to represent the feature of a 3D model.

Recently, Daras and Axenopoulos [29] developed a multi-view descriptor that supports multi-modal queries including 3D models, 2D images and 2D silhouette sketches. To extract 3D model features, they rendered multiple silhouette or depth sample views for a 3D model and utilized Zernike moments and Krawtchouk moments [200] to represent their features. Experiments on the PSB database demonstrate that if using silhouette views the performance is very similar to Light Field and using depth views can achieve a slightly better performance than Light Field. To demonstrate the performance of retrieval using sketch queries, they directly used rendered silhouette views of the 3D models in the target database, which apparently has some bias because users often draw a sketch as the query without knowing the information about the 3D models in the target database.

Lian et al. [100] proposed a view-based descriptor which also 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. [8] also adopted a view-based approach but relied on a more accurate 3D model alignment method.

2.1.3 Hybrid Techniques

Hybrid approach employs both the visual and geometric information of a 3D model. Several hybrid shape descriptors with superior retrieval performance and promising results have been proposed in recent years. DESIRE [191] 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. Depth-buffer based descriptor is composed of the 2D Fourier coefficients of six depth buffer images. Silhouette-based descriptor applies one-dimensional Fourier transform to three silhouette views to extract the features. Ray-based with spherical harmonic representation descriptor first extracts the ray-based feature vector in the spatial domain based on the outmost intersections between the model and a set of rays emanating from the center of the model and then transforms the obtained features from spatial domain to the spectral domain by Spherical Harmonics Transform [78]. DESIRE achieves superior performances than several famous view-based and geometry-based techniques, such as Light Field [23] and Spherical harmonics [78].

Papadakis et al. [141] proposed another hybrid 3D shape descriptor by combining both depth buffer-based 2D features and spherical harmonics-based 3D features. Papadakis et al. [142] 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 the Fourier and wavelet transforms to extract the features for each panoramic view. Recently, Leng and Xiong [89] proposed a hybrid shape descriptor named TUGE which combines the **t**wo-view version of the depth b**u**ffer-based shape descriptor in [190] and the **GE**DT shape descriptor in [43]. 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 [167], ESB [67], CCCC [190] and NIST [36], among the available existing shape descriptors.

2.1.4 Using Class Information

Class-based retrieval scheme has been used in document retrieval or classification [51, 105]. For example, Han et al. [51] first applied centroid-based classifier to automatic text categorization and achieved good performance. With the growth in 3D model retrieval research, class-based retrieval scheme was introduced to improve retrieval performance. For instance, Hou et al. [56] 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. Some other learning-based example algorithms include those based on supervised [83, 84, 85, 195] or semi-supervised [132, 198] learning algorithms. Xu and Li [197] 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 Euclidean (L2) distance between their 3D moment feature vectors. Biasotti et al. [13] proposed a 3D model classification approach by comparing a query model with several prototypes selected to represent a class and applied this prototype-based scheme into a 3D model retrieval application. They made a comparison study between the prototype-based retrieval methods and the commonly used Nearest Neighbor (NN)-based and the results show that NN achieves the best retrieval accuracy, though may be slower.

Tatsuma and Anon [177] 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.

2.2 Partial 3D Model Retrieval

2.2.1 Partial Retrieval Techniques

Partial retrieval can be mainly classified into two groups: (1) graph-based, such as Tierny et al.'s [179] Reeb Pattern Unfolding (RPU) method, Biasotti et al.'s [14] Extended Reeb Graph (ERG) approach, and Cornea et al.'s [26] skeleton matching-based approach (CORNEA); (2) local feature-based, such as Toldo et al.'s [180] Bag-of-Words component Feature based approach (BoF), Liu et al.'s Shape Topics [107] and Gal and Cohen-Or's salient local features [45]. The main idea and performance of the above partial retrieval approaches are as follows.

RPU [179] 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 model with a 3 GHz P4 PC.

ERG [14] 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 subparts between two ERGs. It can be roughly estimated based on the provided performance data [14] that ERG needs at least 3$\sim$11 sec (1.4 sec for feature matching, and 1.5$\sim$10 sec for the preprocessing of feature matching) using a 3.4 GHz PC.

CORNEA [26] is a graph-based approach which extends the skeleton-based matching framework by Sundar et al. [173] 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) [24]. There is no computational time information provided in the paper.

BoF [180] 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.

Shape Topics [107] uses spin image [70] as local feature and also adopted a similar Bag-of-Feature framework for partial matching. Salient geometric features [45] are defined based on a local region characterized by curvature and area. They are employed to extract local shape descriptors to represent the salient parts of a 3D model and thus used to match similar parts of different models. Ferreira et al. [38] presented a shape decomposition method for parts-based retrieval by considering the contextual information of the parts from the same collection.

Marini et al. [115] studied the feature selection issue for spectral matching based on three approaches, such as Hill climbing and Adaboost. They found that not all eigenvalues are necessary for shape matching. Wessel and Klein [195] proposed to decompose a man-made object into primitives and then learn its compositional relationship for 3D model retrieval.

In fact, this primitive-level decomposition method may be also useful for partial similarity retrieval.

Liu et al. [106] proposed to learn a ground distance to adapt the Earth Mover's Distance (EMD) framework for partial similarity matching. Attene et al. [7] extended the coarse-to-fine strategy to part-in-whole 3D shape matching scenario to shorten the matching time. They utilized layered or onion 3D shape descriptors and in an iterative manner, they used increasing portions of the features for the search each time till to the whole descriptor.

3D mesh segmentation are often employed in partial 3D retrieval algorithms and different approaches have been proposed to partition a mesh into semantic parts. Typical methods include graph cut [76], fuzzy clustering [77], spectral clustering [103], fitting primitives [6], shape diameter function (SDF) [164], random cut [48] and learning method [74]. For a point on the surface of a model, shape diameter function (SDF) is to define the local diameter information in terms of its neighboring volume within the region of a cone, centered around the opposite normal of the point and with a default opening angle of 120°. Recently, Kalogerakis et al. [74] proposed a data-driven approach to learn the segmentation of a 3D model. They utilized several available features including shape context, SDF, geodesic distances and spin image [70]. They also found that shape context is the most important feature. They formulated the objective function based on Conditional Random Field (CRF) [82] model and adopted JointBoost [181] classifier for the segmentation.

2.2.2 3D Parts Analysis

3D parts analysis is important and often employed in partial 3D model retrieval. Shapira et al. [162, 163] proposed a method for 3D model contextual parts analysis by using a partition based approach. They adopted the shape diameter function (SDF) defined in [164] for 3D model segmentation. For the segmented parts, they defined a part shape descriptor integrating several local features such as shape diameter distribution, shape distribution [136] and conformal factor [11]. Based on the local features, they defined a contextual-aware distance metric and applied it into parts retrieval and parts annotation. The main shortcoming is that the approach is highly restricted to the initial segmentation of the 3D objects in the target database.

Liu et al. [104] defined a part-aware feature named Volumetric Shape Image (VSI) to encode the visibility information of a surface point based on SDF [164]. A combination of VSI, geodesic and normal features is utilized to define a part-aware surface metric. The main issue is the efficiency of computing VSI (e.g., 15 seconds for a dragon model with 50K faces). Similarly, the parts selection is dependent on the initial segmentation method for target models.

2.3 Sketch-Based 3D Model Retrieval

Sketch-based 3D model retrieval is to retrieve 3D models using a 2D sketch as input. This scheme is intuitive and convenient for users to search for relevant 3D models and also important for several applications including sketch-based modeling [134] and sketch-based recognition [202]. One example of integrating a sketch-based retrieval algorithm into a sketch-based modeling process is proposed by Fonseca et al. [40]. Dependent on the 3D model view sampling strategy adopted in the retrieval algorithm, we classify current sketch-based 3D model retrieval techniques into two categories: using predefined views and using clustered views.

2.3.1 Using Predefined Views

Funkhouser et al. [43] developed a search engine that supports 2D/3D sketch queries. To measure the distance between a 2D sketch and a 3D model, they applied the 3D spherical harmonics [78] method to the 2D sketch in an analogous way to extract a rotation-invariant amplitude-related feature vector and then compared it with those of a *13* sample views. Similarly, Pu and Ramani [147, 148] extended 3D spherical harmonics [78] and shape distribution [135] from 3D models to 2D drawings and proposed a 2.5D spherical harmonics and 2D shape histogram respectively for the retrieval of CAD drawings. Lee et al. [88] matched a sketch with *24* possible orthogonal contour views, based on 6 standard view directions and 4 axis-aligned up-vectors. Squared distance transform is then applied and a sum of squared distances-based similarity metric is adopted to measure the sketch-model distance. Hou and Ramani [57, 58] used a multi-classifier to estimate the probability of the sketch belonging to each class and adopted a classifier combination scheme to find relevant classes. Cao et al. [18] proposed a different retrieval framework by reconstruction of a 3D query model using Bezier surface representation based on user drawn sketches. It constructs an accurate enough 3D query model where users need to draw enough curves to specify the features, which means it may take more time for users to perform retrieval. They mainly compared the performances of different relevance feedback methods using several models and did not compare with other retrieval algorithms using a whole benchmark database.

Kanai [75] proposed a sketch-based retrieval interface by employing two rotation-invariant features, which are generic Fourier descriptor (GFD) [204] and a variation of local binary pattern (LBP) initially proposed by Ojala et al. [133], to measure the distance between a 2D sketch and a rendered view of a 3D model. Wang et al. [192] proposed a sketch-based CAD model retrieval interface using three sketches and a skeleton image as input. To measure the similarity of a 2D outline sketch and the outlines of a 3D model, they adopted angular radial partitioning (ARP) [19]. It decomposes an outline sketch into a set of angular radial sectors, then applies Fourier transform to the statistics of the feature points' distribution, and finally uses the rotation-invariant magnitude vector to represent the 2D sketch. However, they compared the sketch with only the *3* standard outline views of a normalized 3D model. This is feasible for CAD model retrieval but not appropriate for general 3D model retrieval, for which the pose of the query sketch is often not one of the principle views.

Eitz et al. [119] sampled 50 views and utilized apparent ridges [73] to depict the features of a 3D model to correspond with a 2D sketch. They tested on the PSB benchmark [167] using several sketches but did not provide the overall performance. While, Mahmoudi and Daoudi [112, 113] chose only 7 (3 principle and 4 secondary) characteristic views to represent a 3D model. Eitz et al. [119] adopted a Bag-of-Features (BoF) framework to develop an efficient sketch-based 3D model retrieval algorithm. They extracted the local feature of Histogram of Gradient (HOG) for the subdivided patches of both sketch and image. The HOG feature stores for each image cell the sum of squared gradient magnitudes falling into one of six discrete orientation bins. Because of the local feature and the BoF framework, the algorithm supports part-based retrieval. Takeda [175] proposed to utilize a scale, shift, and rotation invariant shape descriptor named IMage Euclidean Distance (IMED) [193] to measure the minimum distance between 20 uniformly sampled silhouette views of a 3D model and 8 rotation images of a sketch.

Recently, Napoléon and Sahbi [125, 126] proposed another sketch-based retrieval algorithm. They utilized a multi-scale convexity/concavity (MCC) shape representation [2] to represent the contours of a set of (*3*$\sim$*9*) sampled views. To speed up the retrieval, a pruning strategy and a dynamic programming approach are adopted to match the MCC features of the sketch and the contours. Yoon et al. [201] proposed a sketch-based retrieval algorithm by matching the sketch with *14* rendered suggestive contours [32] feature views of a model based on the diffusion tensor fields feature representation for the sketch and sampled views. Using the same view sampling method and feature views as Yoon et al. [201], Saavedra et al. [158] proposed a sketch-based 3D model retrieval algorithm using a structure-based local approach (STELA) and achieved a better performance than an improved histogram of edge local orientations-based global approach (HELO) proposed by Saavedra and Bustos [157].

To summarize, to avoid comparing the sketch with a large number of sample views, many previous sketch-based 3D model retrieval algorithms adopted a 2D sketch-3D model matching framework based on several predefined sample views. However, this framework has a shortcoming with respect to how representative the sample views are and the accuracy of the 2D-3D correspondence.

2.3.2 Using Clustered Views

Compared to the approaches based on predefined views, much less research work has been done based on the strategy of view clustering.

Mokhtarian and Abbasi [1, 121, 122] proposed a view clustering method by matching the rendered views and discarding the similar views whose matching costs fall in a predefined threshold. They first created a list of similar views for each sample view and then sorted all the sample views based on the number of similar views they have and finally only kept several top ones as the final characteristic views set.

Ansary et al. [5] proposed an image-based 3D model retrieval algorithm by clustering 320 sample views into a set of characteristic views based on the Bayesian probabilistic approach. They also developed a method to optimize the number (varying from 1 to 40) of characteristic views based on the X-means [143] clustering method. Zernike moments [79] are adopted to represent the views or 2D image queries. Unfortunately, only one demo result for sketch queries was given and no overall performance were evaluated.

2.3.3 2D Sketch-3D Model Alignment

In our proposed sketch-based 3D model retrieval algorithm, we propose aligning the sketch with the 3D model, that is 2D sketch-3D model alignment, before 2D-3D matching. 2D sketch-3D model alignment is aligning a 2D sketch of an object with a similar 3D model. It is important for related applications, such as model-based vision or recognition [109] and sketch-based 3D modeling. For example, a concept sketch-based 3D modeling system [91] also involves estimating the camera parameters of 2D sketches in order to establish correspondences between the 2D features in the sketches and those on the surface of a 3D model.

Generic 2D-3D alignment is to estimate the pose of a 3D model for a 2D image. Most previous 2D-3D alignment methods [37, 61] are based on optimizing the rotation angles (ϕ, θ, ψ) and the translation parameters (T_x, T_y, T_z). For example, Lavallee et al. [86] extended the famous 3D-3D optimization alignment algorithm Iterative Closest Point (ICP) [12] to solve the 2D-3D alignment problem.

However, our research of 2D sketch-3D model alignment has several apparent differences when compared with previous generic 2D-3D alignment algorithms. For a comparative reference, we review generic 2D-3D alignment techniques in this section.

2.3.3.1 Generic 2D-3D Alignment

2D-3D alignment usually comprises two steps: feature extraction and model transformation. The accuracy of the alignment result is highly dependent on the quality of feature extraction and matching. According to the different features adopted, we classify 2D-3D alignment into three types: mutual information-based, silhouette or contour-based and other features-based methods. For a general 2D-3D alignment, registration and alignment have the same meaning, so we use these two words interchangeably.

Mutual Information-Based Methods. Viola et al. [189] introduced the concept of mutual information (MI) as a similarity measure based on information theory. They demonstrated its usage for aligning untextured 3D objects to images using the interpolated surface normals. Leventon et al. [92] used multiple views rather than only a single view to improve the accuracy of the 2D-3D registration result based on the same framework of maximization of mutual information. Liebelt et al. [102] combined the contour matching and mutual information to measure the similarity of a synthetic aperture radar (SAR) image and a view image of a 3D model. They also used an evolutionary strategy named particle swarm optimization (PSO) [34] to optimize the similarity measure. Maes et al. [111] proposed a multimodality image registration algorithm by maximizing a more general notion of mutual information. It can achieve subvoxel accuracy. Corsini et al. [27] designed a method by measuring the mutual information difference between the image to be registered and a model's renderings incorporating certain illumination-related geometric properties, such as normals, reflection directions, and ambient occlusions. It can cover different conditions of lighting or materials. In several publications, extensions to the classical MI formulation are proposed, notably by introducing normalization terms [62].

Mutual information-based registration techniques are robust with respect to variations of illumination [189]. However, they are based on the grayscale information of the views of an object, rather than the geometric features, such as lines, curves and regions. Therefore, it cannot be applied to solve our 2D sketch-3D model alignment problem.

Silhouette or Contour-Based Methods. These approaches usually optimize the spatial distances between the silhouettes or contours of a 2D image and the projected contours of a 3D model. Iwashita et al. [62] proposed a fast alignment algorithm utilizing a 2D distance map constructed using the level set method (LSM). They adopted fast marching method (FMM) [160] to rapidly construct a distance map on the 2D image plane. Finally, they compared the distances between the contours in the 2D distance map and the projections of the 3D model to align the model. Lavallee et al. [86] utilized a precomputed 3D distance map of a freeform object for 3D pose estimation. The error metric is defined as the minimum distance between the surface of the 3D model and a projection ray. The sum of the error is minimized using the Levenberg-Marquardt [90, 116] method. To make the registration process more efficient, 3D distance from the surface is precomputed and stored in an octree structure. Cyr et al. [28] proposed a global hierarchical shape matching approach based on aspect graph segmentation of the object view space. They adopted a medial axis metric to measure the topological difference between two views. First, a set of sampled 2D views are generated and matched against the given view. Next, additional views are generated near the best view and the process is repeated until convergence.

Silhouette or contour-based alignment approaches perform the alignment process through an optimization process by defining some error metrics. Thus, most related algorithms compare the silhouette or contour features of each projection view of a 3D model with a 2D image directly. While, for our case of 2D sketch-3D model alignment, since there is no one exact pose to perfectly align the 2D sketch with 3D models, the optimization process is inapplicable.

Other Feature-Based Methods. Hara et al. [52] proposed to solve the problem of 2D-3D alignment based on four defined geometrical consistencies: linear consistency, planar consistency, parallel consistency and orthogonal consistency. Firstly, a geometrical consistency based registration method is applied to determine a rough relative pose and then an edge-based registration algorithm is used to estimate a precise pose. Chang-Chang et al. [20] proposed a method by matching scale-invariant feature transform (SIFT) features of a single image to viewpoint invariant patches (VIP) of a 3D model by warping the SIFT features approximately into the orthographic frame of the VIP features. This approach significantly increases the number of feature correspondences which results in a reliable and robust pose estimation. Kurazume et al. [81] proposed the simultaneous registration algorithm using 2D texture images and reflectance images. Epipolar constraints are utilized to estimate relative poses of multiple texture images simultaneously. Based on the mutual information alignment framework, Cprsomo et al. [27] utilized illumination-related 3D model geometric properties such as normals, ambient occlusion and reflection directions to solve the 2D image and 3D model alignment.

Shape context matching-based human body pose estimation was proposed by Mori and Malik [124]. They adopted deformable template matching to localize human body joint positions. First, store a number of exemplar 2D views of a human body in a variety of different configurations and viewpoints with respect to the camera. The joints of the body in each exemplar view are manually labeled. Then, match input images and exemplar views based on shape context matching [10] and finally transfer the results to 3D body configurations. The goal of this method is to find a pose for articulated objects which is different from ours.

2.3.4 2D Shape Descriptors

Developing or selecting an appropriate 2D shape descriptor is an important part for a sketch-based retrieval algorithm. In this section, we present several typical 2D shape descriptors that are promising for sketch-based retrieval.

Fourier descriptor (FD) is an important shape descriptor and has been successfully applied in many pattern recognition related applications such as shape analysis, classification and retrieval as well as character recognition [203]. However, it assumes that we can get the boundary information of a shape beforehand and it does not consider the internal information of the shapes. Considering the above limitations of Fourier descriptor, Zhang and Lu [159, 204] extended the Fourier descriptor and proposed a more robust and accurate shape descriptor called generic Fourier descriptor (GFD) which applies Fourier transform on a polar-raster sampled shape image.

Zernike moments feature [79] is one typical moment descriptor that outperforms other moments in terms of performance in different applications. For example, 3D Zernike moments [129] has been developed to deal with 3D model retrieval. Revaud et al. [156] proposed an improved Zernike moments [79] comparator which considers not only the magnitude of the moments (classic Zernike moments comparator) but also their phase information. They demonstrated its better performance than the classic one.

Local binary pattern [75, 133] divides the surrounding regions of any pixel in a binary image into eight directions, computes the percentages of the pixels falling in each bin and regards this distribution information as a local binary pattern (LBP) encoded using an 8-bit binary number, and finally represents the whole image based on the statistical distribution of all the local binary patterns. It can be used to measure the similarity between the 2D sketch after a preprocessing and the rendered feature images of a 3D model.

Scale-invariant feature transform (SIFT) feature together with the Bag-of-Features (BoF) framework has many applications in various computer vision research fields. To optimize the search accuracy, efficiency and memory usage in a large scale image retrieval scenario which utilizes SIFT features and BoF framework, Jégou et al. [69] proposed a new compact image representation to aggregate SIFT local descriptors. It achieves a significantly better performance than BoF on condition that the feature vectors used have the same size.

Shape context [10] is a log-polar histogram and defines the relative distribution of other points with respect to a point. It has been successfully applied into diverse applications. The default shape context definition partitions the surrounding area of a sample point of a 2D shape into 5 distance bins and 12 orientation bins, as shown in Fig. 2.1 (c). Thus, the shape context is represented by a 5×12 matrix. In Fig. 2.1, we show the shape context features of three points in two shapes. As can be seen, different points have different shape context features in one shape, such Fig. 2.1 (d) and Fig. 2.1 (e) and similar points in two similar shapes usually have similar shape context features, like Fig. 2.1 (d) and Fig. 2.1 (f).

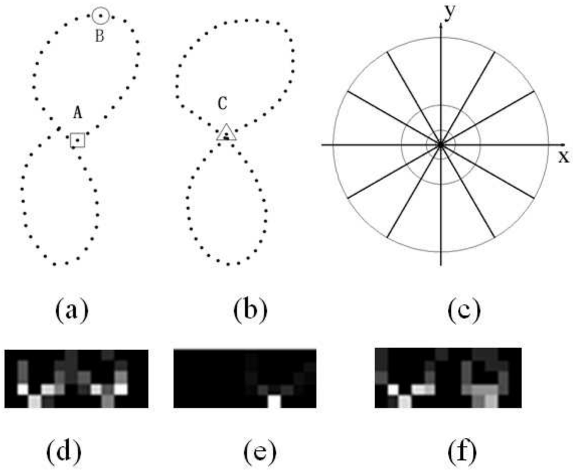


Figure 2.1: Shape context examples. (d), (e), (f) are the shape context features of points A and B in (a) and point C in (b) respectively. The grayscale value of each element represents the percentage of other points in each bin. Darker means smaller.

Shape context is scale and transformation-invariant but not rotation-invariant. To achieve the property of rotation invariance, in [10] a relative frame is defined by adopting the local tangent vector at each point as the reference *x* axis for angle computation and we named it relative shape context. Shape context has been successfully applied into diverse applications, such as 2D shape matching [123] and human body pose estimation [124]. 3D shape context [41, 60, 80] extends the idea of shape context [10] from 2D to 3D. There are three forms of 3D shape context according to the aforementioned (Section 2.1.1) three types of 3D space partitioning methods. Shell model divides the 3D space into a set of concentric spheres, resulting in the support of volume 3D shape context [41]. Sector model only divides the spherical angle space, and we will get 3D cylindrical shape context [60]. Spider web model combines both and we get the 3D point shape context [80].

2.4 3D Model Alignment

In this section, we review the related techniques in 3D model alignment. Our proposed MPA 3D model alignment algorithm (Chapter 3) is based on minimum projection area, so it can be considered as a view-based approach. As such, we also review some viewpoint selection techniques in Section 2.4.5. Please note that though the spherical correlation method proposed by Makadia et al. [114] (mentioned in Section 1.1) is also a view-based approach, it is a pair-wise 3D model matching technique used for distance computation and cannot be classified as a 3D model alignment approach which should independently transform a 3D model into a canonical frame.

Nowadays, there are several approaches to align a 3D model. Here, we review four different approaches.

2.4.1 PCA-Based Approach

Principle Component Analysis (PCA) [71] and Continuous Principle Component Analysis (CPCA) [190] are two commonly used alignment algorithms. They utilize the statistical information of vertex coordinates and extract the three orthogonal components with largest extent to depict the principle axes of a 3D model. Axis ordering and axis orientation are further applied to solve the axis reflection (flipping) invariance, which usually utilize variance or moment analysis [191]. An extension of the idea of CPCA is Normal Principle Component Analysis (NPCA) [140], which applies CPCA to the normals of the surface points of a 3D model.

The shortcoming of PCA-based approach is that the directions of the largest extent are not necessary parallel to the axes of the ideal canonical coordinate frame of 3D models. CPCA is generally regarded as a more stable PCA-based method. However, Papadakis et al. [140]

found that for some models (e.g. car, shovel, hammer and plotted plant) CPCA outperforms NPCA, but for some other models (like plane, chair, gun and desktop computer) NPCA has a better alignment performance.

2.4.2 Symmetry-Based Approach

Chaouch and Verroust-Blondet [22] proposed an approach based on the analysis of the reflection symmetry property of a 3D model, such as cyclic, dihedral and rotation symmetries. Podolak et al. [144] developed a symmetry transform to measure the degree of symmetry of a 3D model with respect to any candidate symmetry plane. Tedjokusumo and Kheng Leow [178] developed an alignment algorithm using bilateral symmetry planes (BSPs) by considering the 3D aspect ratio of a model. They defined three BSP axes in an analogous way as PCA [71]: the first BSP axis has the largest extend in the BSP, the second is perpendicular to the first, and the third is the normal of the BSP. However, symmetry-based approach has a limitation in dealing with models without apparent symmetry property or non-symmetrical models.

2.4.3 Optimization-Based Approach

Fu et al. [42] proposed an upright alignment algorithm for man-made models. The algorithm first computes the convex hull of a model, then finds a set of candidate bases, and finally selects the base with the largest assessment function value as the bottom of the model. The assessment function is composed of four geometrical properties: static stability, symmetry, parallelism and visibility. Random Forest classifier and Support Vector Machine (SVM) classifier are adopted to train the function. The upright orientation algorithm achieves around 90% prediction accuracy in terms of the vertical extent of models. Sfikas et al. [161] proposed an alignment algorithm named ROSy by minimization of the bounding box of a 3D model based on PCA and Reflective Object Symmetry (ROSy). They achieved this by minimizing a combined spatial and angular distance metric. Martinek and Grosso [118] proposed an optimization and GPU-based approach to align two 3D models. They constructed a model function with respect to the intersection and union of the projection results of two models.

2.4.4 Projection Area-Based Approach

Recently, we notice that there are other two papers which use projection area for alignment. Lian et al. [101] proposed a method that first determines two sets of candidate axes using PCA and the rectilinearity metric. Then, the final alignment axes are decided by selecting the set of candidate axes which minimizes the sum of the projected area of silhouettes. Napoléon and Sahbi [125] presented an alignment method which selects one of three alignment results (original pose, PCA and NPCA) that gives the minimum visual hull, that is minimizes the sum of the projected areas on the three projection planes.

2.4.5 Viewpoint Selection

The goal of viewpoint selection is to find a set of representative views to depict a 3D model. Usually, it is used to select the best views of a 3D model.

Lee et al. [87] defined the idea of mesh saliency for 3D models in terms of Gaussian-weighted mean curvatures. Viewpoint selection, one of the applications of mesh saliency, was demonstrated based on a gradient-descent search to find the candidate views with local maximums and a random search algorithm to find the global maximum. Yamauchi et al. [199] proposed a method to find a set of representative views for a 3D model by clustering the views and using mesh saliency [87] to characterize the quality of a view. Laga and Nakajima [83, 84] proposed a supervised learning approach to select a set of best views of a 3D object that discriminate itself from other objects and maximize the similarity between itself and the models belonging to the same class. Vázquez et al. [185, 186] proposed an information theory-related measurement named viewpoint entropy to depict the amount of information a view contains and based on this they developed a method to automatically find a set of best views with top view entropy values. All these viewpoint selection techniques can select the best views of 3D models with respect to their view quality metrics, however, in general the direction of the best views are not parallel to the axes of the ideal canonical coordinate frame of 3D models.

2.5 Comparison Between Our Proposed Techniques with Previous Methods

2.5.1 View-Based 3D Shape Descriptor: View Context Versus Other Approaches

Firstly, our view context descriptor supports multi-modal queries, such as 3D models and 2D sketch/images while many view-based approaches reviewed in Section 2.1.2 only support 3D model queries. Secondly, previous view-based shape descriptors like the Light Field descriptor compare the views of different models directly, while our view context shape descriptor encodes the differences of views of the same model first and then we compare the view context features of different models to measure their difference. That is, our view context shape descriptor characterizes the shape deviation difference of a 3D model while previous view-based shape descriptors including Light Field do not encode such information. As such, we utilize this property of the view context descriptor and devise a 2D sketch-3D model alignment algorithm and incorporate it in our Query-by-Sketch retrieval algorithm.

2.5.2 Class-Based 3D Model Retrieval: Integrated Distance Versus Additive Distance

Firstly, in Chapter 6, we propose an integrated distance which outperforms the aforementioned and commonly used additive one (Section 2.1.4). Moreover, none of the existing 3D model retrieval algorithms utilize the "already" available class information of a classified 3D model database. For example, existing class information-based approaches reviewed in Section 2.1.4 usually learn the class information, such as [56, 83, 84, 85, 132, 195, 197, 198], or employ a spectral clustering method like [177], or adopt a prototype-based retrieval framework like [13]. None of them consider and directly use the already available class information existing in the classified databases. Thus, in Chapter 6, we propose a new 3D model retrieval algorithm by taking into account the existing class information. Thirdly, our experiments in Chapter 6 show that our retrieval algorithm CBR-ZFDR can achieve better retrieval performance for most of the commonly used performance metrics than PANORAMA on six standard databases.

2.5.3 3D Model Alignment: MPA Versus Other Approaches

Firstly, different from the two projection area-based 3D model alignment methods introduced in Section 2.4.4, our proposed method performs alignment by successively selecting two axes with minimum projection areas. Moreover, we perform a global optimization search for finding the minimum projection area and our algorithm does not rely on the PCA-based approach.

Secondly, MPA has several differences compared to Fu et al.'s upright orientation method [42] (Section 2.4.3). They target on computing the upright orientation (not complete alignment) of man-made models with a flat supporting base while MPA is to align generic 3D models; their approach is based on geometry alone while our method is mainly view-based;

they utilize a supervised learning approach and need constructing a manually labeled training dataset while our MPA algorithm is unsupervised and independent of training.

Thirdly, diverse experiments also demonstrate that MPA achieves a better performance than several commonly used approaches such as CPCA, NPCA and PCA, in terms of alignment accuracy and retrieval performance.

2.5.4 Sketch-Based 3D Model Retrieval: 2D Sketch-3D Model Alignment

Firstly, different from previous sketch-based 3D model retrieval algorithms which directly compare the sketch with a set of predefined views of a 3D model, our sketch-based retrieval algorithm first aligns the sketch with the 3D model based on the view context similarities between the sketch and the sample views of the model. We utilize the precomputed view context features of a set of densely sampled views to shortlist a set of good candidate sample views to align with the sketch. In the mean time, since the 2D sketch-3D model alignment process is very efficient, thus the computational cost for alignment is low.

Secondly, our 2D sketch-3D model alignment is also different from the common 2D image-3D model registration techniques which optimize the rotation angles and the translation and scaling parameters to register a 3D model with a 2D image. On one hand, their 2D image and 3D model depict the same object. However, for our case they are not and some differences are often existent. On the other hand, previous 2D image-3D model registration techniques used 2D real image which has brightness (shading) information and developed an as accurate as possible 2D-3D alignment. For our case, we use 2D sketch which only has line information and since the 2D sketch and the 3D model are not completely the same, an approximate alignment is sufficient for our retrieval applications.

Thirdly, comparative and evaluative experiments based on hand-drawn and standard line drawing sketches demonstrate the effectiveness and robustness of our approach and it significantly outperforms several latest sketch-based retrieval algorithms. Incorporating our 2D sketch-3D model alignment step to shortlist better candidate views apparently improves the retrieval performance and our alignment is also general and can be incorporated in other sketch-based 3D model retrieval algorithms to improve their performance.

Chapter 3

3D Model Alignment Based on Minimum Projection Area

3.1 Overview

As described in Section 1.1, several typical alignment algorithms (PCA, CPCA, NPCA and VCA) have been proposed to deal with 3D model alignment. However, the existing alignment algorithms still have room for improvement in terms of the performance in finding alignment axes which are parallel to the ideal canonical coordinate frame and 3D model retrieval. This motivates us to propose a novel 3D alignment algorithm which finds the alignment axes based on minimum projection area (MPA). Our proposed algorithm is based on the observation that many objects have a minimum projection area when we orthogonally project them in the direction parallel to one of the axes of the ideal canonical coordinate frame. To efficiently find the axis with minimum projection area, a search algorithm based on Particle Swarm Optimization (PSO) [34] is developed. To apply MPA in 3D model retrieval application, two additional steps, axis ordering and axis orientation, are devised to align similar models in similar poses. Based on experimental results, we find our MPA algorithm can align most 3D models in terms of axes accuracy (the axes are parallel to the ideal canonical coordinate frame). Our alignment algorithm can also align similar models in similar poses which is important for 3D model retrieval. It is also robust with respect to model variations, which include different shapes of the same class, non-rigid deformation and articulation transformations, noise in terms of vertex displacement and initial poses.

The rest of this chapter is organized as follows. In Section 3.2, we present the details of our MPA alignment algorithm. Extensive experiment results are shown in Section 3.3. Section 3.4 contains the conclusions and the future work.

3.2 Minimum Projection Area-based (MPA) Alignment

3.2.1 Basic Idea

Based on its ideal canonical coordinate frame, every 3D model has six canonical orthographic projection views, which are front, back, left, right, top and bottom views as shown in Fig. 3.1. If we only consider the projection area (the area of the region occupied by the object in the view images in Fig. 3.1), then there are only three different canonical views because under orthographic projection, the front view has the same projection area as the back view, the left view has the same area as the right view, and the top view has the same area as the bottom view. We observe that for many objects, one of their canonical views (that is, either front-back view or left-right view or top-bottom view) has a minimum projection area compared to the other arbitrary views of the objects. Fig. 3.2 shows two such examples. In fact, we conduct experiments on several 3D model databases and verify that the above-mentioned observation is true for a large number of 3D models.

Motivated by the above findings, we develop a Minimum Projection Area-based alignment algorithm (MPA). Our algorithm finds three principle axes of a 3D model which satisfy the followings. The first principle axis gives the minimum projection area when we perform an orthographic projection of the model along (parallel to) this axis, the second axis is

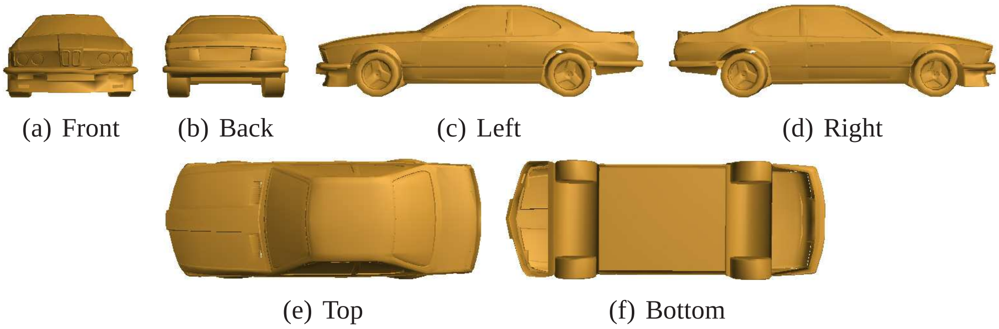


Figure 3.1: Six canonical orthographic projection views of a car model based on its ideal canonical coordinate frame.

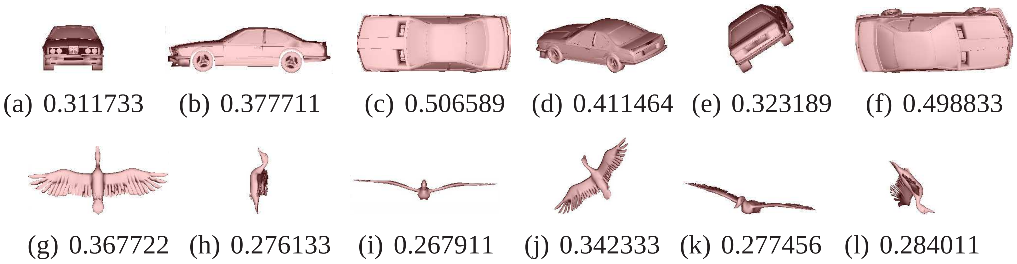


Figure 3.2: Examples showing that one canonical view of a 3D model usually has the minimum projection area. In each row, the first three images are the front, left and top views of a 3D model and the remaining three images are three arbitrary views of the same model. The number underneath each view is its normalized projection area.

perpendicular to the first axis and gives a minimum projection area, and the third axis is the cross product of the first two axes.

3.2.2 MPA Alignment Algorithm

Given a 3D model, the set of candidate axes is generated by using a sphere. A *candidate axis* is defined as a line which connects a point on the surface of the sphere and the center of the sphere. To compute the projection area of this axis, we perform an orthographic projection of the model in the direction parallel to the axis and determine the projection area by counting the number of pixels occupied by the model in the projection image.

The framework of our MPA algorithm is shown in Fig. 3.3 and the detailed steps are as

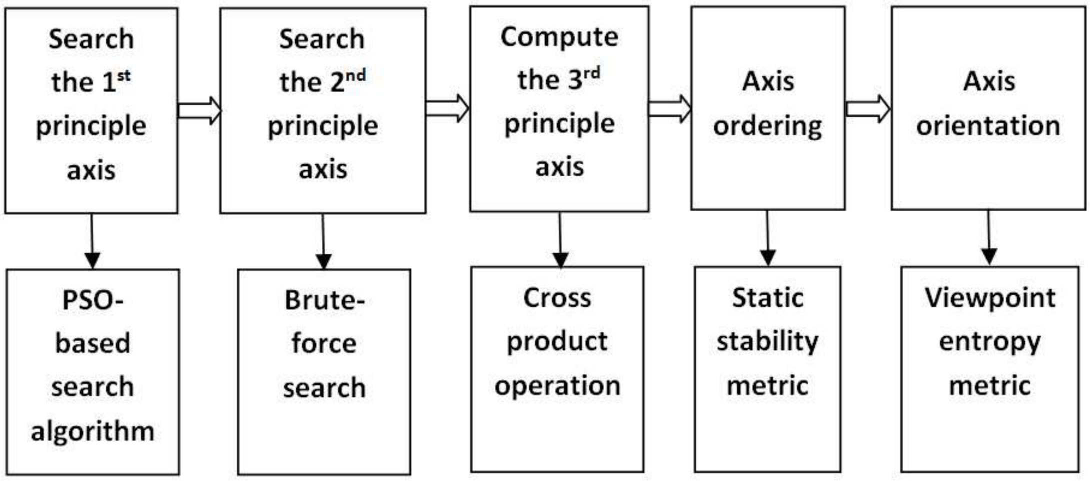


Figure 3.3: The framework of our MPA algorithm.

follows.

Step 1: Find the 1st principle axis. We sample a set of points on the surface of the sphere, compute the candidate axes based on these points and find the axis with minimum projection area. To find this axis, we devise an efficient search algorithm based on the Particle Swarm Optimization (PSO) [34] method (see Section 2.3).

Step 2: Find the 2nd principle axis. We find the axis with minimum projection area by sampling on the perimeter of a circle which is perpendicular to the 1st principle axis. Since this is only a 1D search, we perform a brute-force search to find the 2nd principle axis by sampling the perimeter in the range of $[0^\circ, 180^\circ)$ and choosing a step of 1°.

Step 3: Compute the 3rd principle axis. We compute the third axis as the cross product of the first two principle axes.

For 3D model retrieval application, the following two steps are performed to align similar models in similar poses.

Step 4: Axis ordering. First, we determine the top and bottom orientations of the model by adopting the static stability metric in upright orientation [42]. We compute six static stability values of the principle axes (two for each axis, in the positive and negative directions). The direction with the largest static stability value is set as the bottom of the model (the y negative axis of the model) and the corresponding principle axis is set as the y axis. Then, we determine the x and z axes based on the variance of the remaining two principle axes. The axis with a larger variance is set as the x axis and the other as the z axis. In order to compute the variance, we employ a similar method as [190] by considering the area of each face of the model.

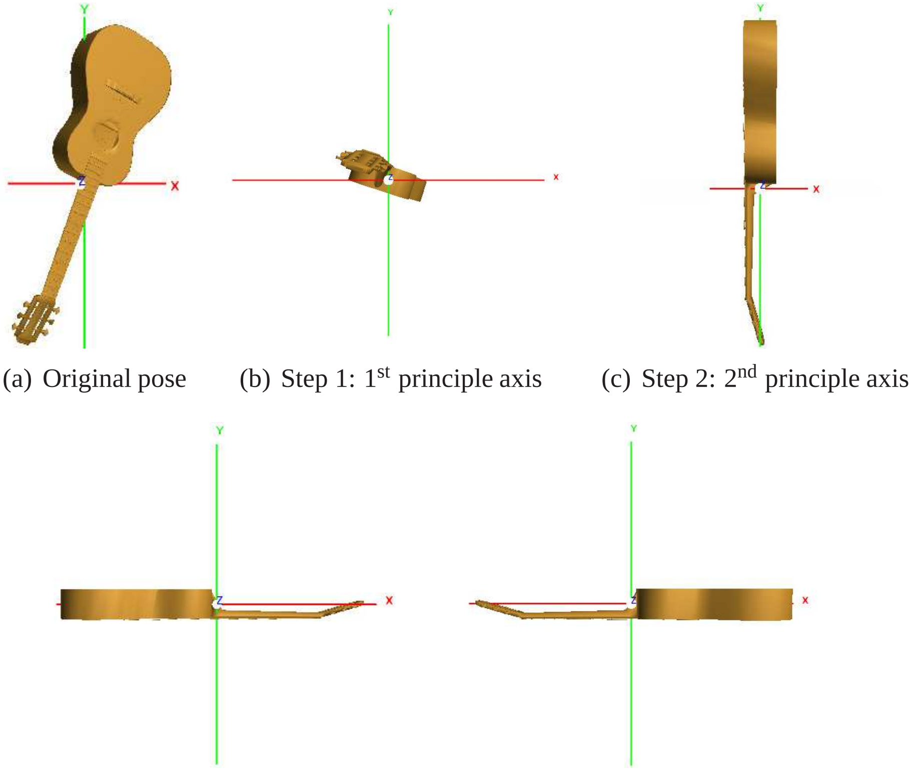
Step 5: Axis orientation. We employ the viewpoint entropy metric [186] to decide the orientations of the x and z axes. We render two views of the model from the positive and negative sides of the x axes (z axes) and select the one with a larger entropy value as the left side (front part) of the model.

Fig. 3.4 shows the result at each step of MPA alignment for a guitar model.

3.2.3 PSO-Based Search for Minimum Projection Area

The simplest method to find the axis with minimum projection area is by performing a brute-force search. We can uniformly sample a set of points on the surface of the sphere based on the subdivision of a regular icosahedron which is denoted as the zero level icosahedron L_0. Fig. 3.5 shows the resulting icosahedrons at different levels of subdivision by applying the Loop subdivision rule [108] once (L_1), twice (L_2), thrice (L_3) and four times (L_4).

Fig. 3.6 shows the distribution of projection area of two models in NIST database [36] using the 3$^{\text{rd}}$ level icosahedron L_3 for sampling the axes and mapping their projection areas as colors on the surface of the spheres. The drawback of the brute-force search is the high computational cost. Based on experimental results, we find that in order to get a result with good accuracy, we have to use at least L_6 icosahedron (40002 sampling points). As such,

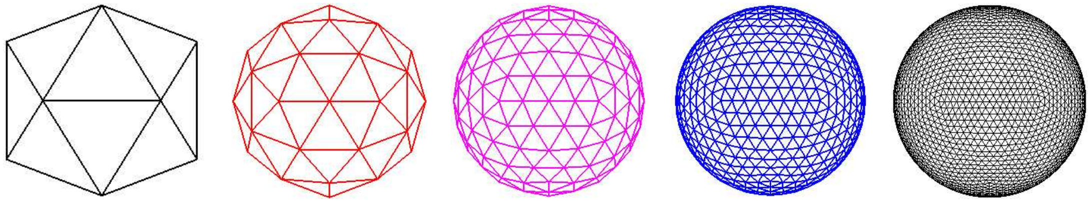


(a) Original pose (b) Step 1: 1st principle axis (c) Step 2: 2nd principle axis

(d) Step 3∼4: axis ordering (e) Step 5: axis orientation (final result)

Figure 3.4: An example of the alignment process using our MPA algorithm. (b)∼(e) show the intermediate alignment results of the five steps in the algorithm, respectively.

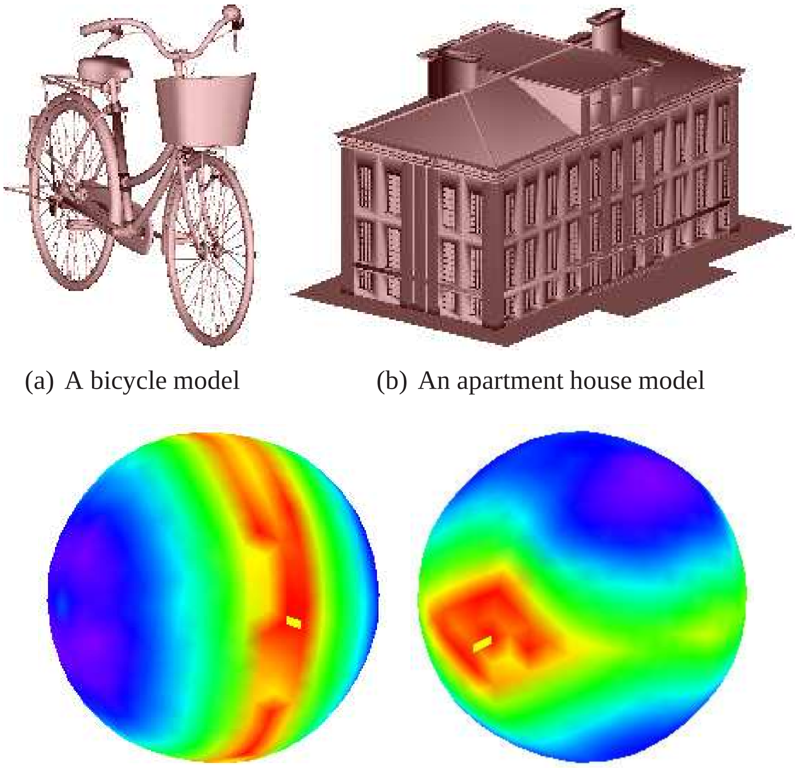
the brute-force search is not the ideal method for finding the axis with minimum projection area.



(a) L_0 (12) (b) L_1 (42) (c) L_2 (162) (d) L_3 (642) (e) L_4 (2562)

Figure 3.5: Subdivision of an icosahedron. The number in each bracket is the number of sample points of the corresponding subdivided icosahedron.

To find the axis efficiently, we develop a search method based on Particle Swarm Optimization (PSO) [34] which is a global search optimization algorithm. PSO belongs to swarm



(a) A bicycle model

(b) An apartment house model

(c) Area distribution of (a)

(d) Area distribution of (b)

Figure 3.6: Distribution of the projection area of two models. Area is coded using HSV color model and smooth shading. Red: small area; green: mid-size area; blue: large area. The yellow bar depicts the sample point with minimum area.

intelligence optimization techniques and it imitates the random search actions of a flock of birds seeking for a piece of food in a region. Each bird adopts the same strategy of searching the surrounding area of the bird that is nearest to the food and they learn with each other and update themselves based on the obtained information. PSO has been found to be robust and fast in solving non-linear and non-differentiable problems [165].

The steps of our PSO-based search are as follows.

Step 1: Initialization. We initialize the number N_P and positions of a set of search particles and then compute the private best for each particle and current global best based on all the private bests. In practice, we use the 42 sample points in L_1 to distribute the search particles. To compute the private best of a search particle, we consider its $\lfloor N_P/3 \rfloor$ nearest neighboring particles in terms of geodesic distance. Then, we set the value for the maximum number of search iteration N_t.

Step 2: Update. We compute the velocity update step s inversely proportional to the current iteration number i,

$$s = \frac{N_t - i}{N_t} + c, \tag{Eq. 3.1}$$

where c is a constant variable. We choose c to be 0.5 in our experiment according to the trade-off between the accuracy of the experimental results and speed. Based on the following two equations [165], we update the new position for each particle as follows.

$$\vec{x}(i+1) = \vec{x}(i) + s \cdot \vec{v}(i), \tag{Eq. 3.2}$$

$$\vec{v}(i+1) = \omega * \vec{v}(i) + c_1 \cdot r_1 \cdot (\vec{x}_p(i) - \vec{x}(i)) + c_2 \cdot r_2 \cdot (\vec{x}_g(i) - \vec{x}(i)). \tag{Eq. 3.3}$$

$\vec{x}(i)$ and $\vec{v}(i)$ are the position and velocity of a particle; $\vec{x}_p$ and $\vec{x}_g$ are the positions of private and global bests. c_1 and c_2 are non-negative constant number, typically c_1=c_2=2 [34]; r_1 and r_2 are random variables between 0 and 1. ω is an inertia-weight to balance the abilities of global search and local search. Bigger ω means more global search power and less dependency on the initial positions of the search particles. Smaller ω corresponds to finer search in a local region. Similar as [165], we dynamically decrease ω from 1.4 to 0 based on an inversely proportional function with respect to the iteration number i:

$$\omega = \frac{\omega_{min} - \omega_{max}}{N_t} \cdot i + \omega_{max}, \tag{Eq. 3.4}$$

where ω_{max} (1.4) and ω_{min} (0) is the maximum and minimum inertia-weight values. The new position $\vec{x}(i+1)$ may not be located on the surface of the sphere, as such we project it to the surface of the sphere in the direction from the center to the computed $\vec{x}(i+1)$.

Step 3: Evaluation. Based on its new position, for each particle, we compute the corresponding axis, render the 3D model, compute the projection area and update its private best. Based on all the private bests, we update the global best.

Step 4: Verification. If the current iteration number has exceeded N_t, we stop and output the axis which corresponds to the position of the current global best as the 1st principle axis; otherwise, go to **Step 2: Update** to continue the search.

3.3 Experiments and Discussion

To intensively investigate the performance of our MPA alignment algorithm, we test the MPA algorithm on four representative standard databases described in Section 1.1: PSB (test dataset), NIST, WMB and ESB.

3.3.1 Evaluation with Respect to Axes Accuracy

Experiments on different types of models, such as general models in PSB, CAD models in ESB, and non-rigid models in WMB, demonstrate that our MPA can align most of them accurately, robustly and consistently. Some examples are shown in Fig. 3.7.

Finding three alignment axes which are parallel to the ideal canonical coordinate frame is important. Therefore, we perform axes accuracy experiments on the above-mentioned four databases and compare MPA with CPCA in terms of the percentages of the alignment results that have three axes parallel to the ideal canonical coordinate frame (allow a very small rotational difference). For a database, we calculate the average percentage over all the models as well as the percentage for each class. Table 3.1 compares their performances and Table 3.2 lists the classes in which MPA achieves a much better performance than CPCA on the PSB database.

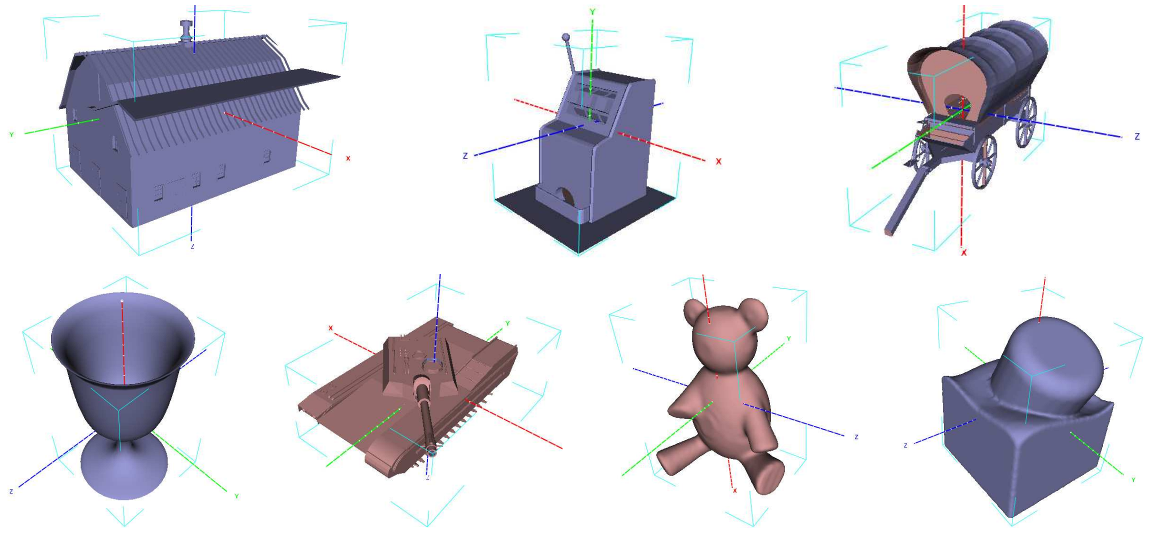


Figure 3.7: Example alignment results for different types of models using our MPA alignment algorithm.

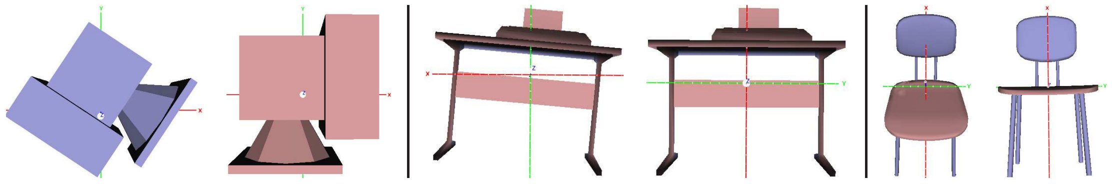
As shown in Table 3.1, our MPA approach achieves apparently better overall performance than CPCA. MPA is better than CPCA in aligning 53.3% classes for PSB, and 52.5%, 70.0%, 55.5% for NIST, WMB and ESB, respectively. Conversely, the percentages for the cases in which CPCA outperforms MPA are much smaller (15.2%, 17.5%, 15% and 11.1%, respectively). MPA has a much better performance (the surpassing percentage difference is more than 20) in aligning the listed 41 classes of PSB models in Table 3.2, especially for box-like shapes, such as desktop computer, computer monitor, school desk and church. Fig. 3.8 shows some examples which demonstrate that MPA can find more accurate axes than CPCA.

For certain models, MPA cannot find their accurate axes and usually there exists some small rotational differences. The reason for these differences is that a small rotation from the accurate axes will make the projection area even smaller. These types of classes include dog, desk chair, potted plant, barren tree, conical tree, handgun and fireplace. Some alignment results for these classes are shown in Fig. 3.9. Nevertheless, we can see even if the axes

Table 3.1: Comparison of the axes accuracy performances in terms of models and classes using MPA and CPCA on the PSB, NIST, WMB and ESB databases. R^{CPCA} and R^{MPA} is the average performance over all the models in a database. Δ is the performance difference value of subtracting MPA's axes accuracy percentage by CPCA's for one class. $\Delta \geq 20$: MPA is much better than CPCA; $20 > \Delta > 0$: MPA is better than CPCA; $\Delta = 0$: MPA is the same as CPCA; and vice versa.

Databases	PSB	NIST	WMB	ESB
#(models)	907	800	400	867
R^{CPCA}	632 (69.7%)	652 (81.5%)	270 (67.5%)	657 (76.1%)
R^{MPA}	804 (88.6%)	695 (86.9%)	327 (81.8%)	744 (86.2%)
#(classes)	92	40	20	45
$\Delta \geq 20$	41 (44.6%)	5 (12.5%)	6 (30.0%)	15 (33.3%)
$20 > \Delta > 0$	8 (8.7%)	16 (40.0%)	8 (40.0%)	10 (22.2%)
$\Delta = 0$	29 (31.5%)	12 (30.0%)	3 (15.0%)	15 (33.3%)
$0 > \Delta > -20$	6 (6.5%)	6 (15.0%)	1 (5.0%)	4 (8.9%)
$\Delta \leq -20$	8 (8.7%)	1 (2.5%)	2 (10%)	1 (2.2%)

found are not the perfect ones, their alignment results are still consistent among the models in the same class, which is important for applications, such as 3D model retrieval.



(a) CPCA (b) MPA (c) CPCA (d) MPA (e) CPCA (f) MPA

Figure 3.8: Examples indicating that our MPA algorithm achieves better alignment results than CPCA.

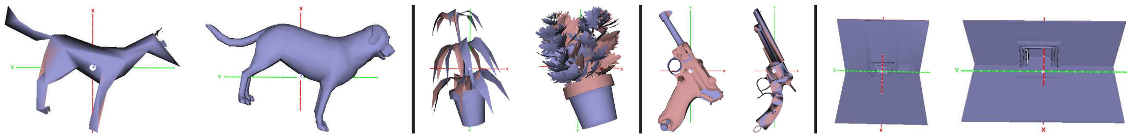


Figure 3.9: Examples showing that the alignment results of our MPA algorithm are still consistent within classes even if the result axes are not the perfect ones.

3.3.2 Evaluation with Respect to Robustness

In this section, we test the robustness properties of MPA with respect to model variations, noise and initial poses as well as the convergence of PSO with respect to iteration number.

(1) Robustness to model variations. The basic requirement for alignment in applications such as 3D model retrieval and recognition is to align similar models in a similar way under different conditions such as variations and deformations. For this purpose, we investigate the alignment performance on non-rigid models with different variations, for example, hand, teddy and head models in the previously mentioned four databases. Some example alignment results for these types of models in the WMB database are shown in Fig. 3.10. The first nine models are examples of deformable models. We can also see that the head models with different variations are aligned consistently, such as the three similar head models looking to the front and the other three similar head models looking to the left.

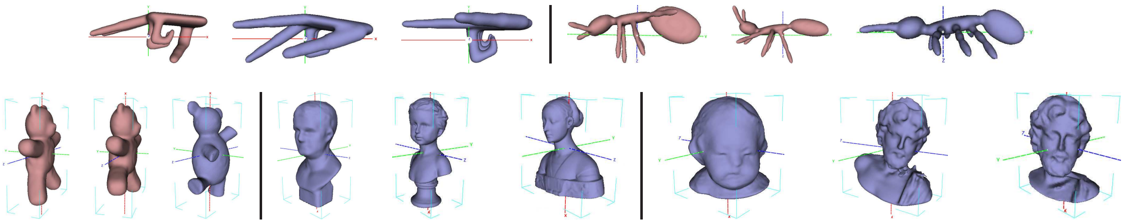
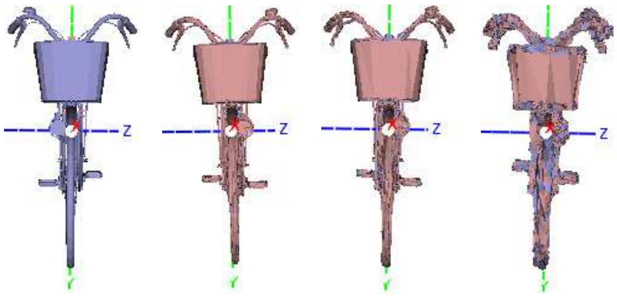


Figure 3.10: Examples indicating MPA can align similar models in similar poses.

(2) Robustness to noise. 3D models may have noise due to storage, transmission and modification. A 3D model alignment algorithm should be insensitive to small amount of noise. We test the robustness of our MPA algorithm against noise by randomly adding a small amount of displacement to the vertices of a 3D model. Fig. 3.11 shows that MPA has a good robustness property against a small amount of noise. This is contributed to our utilization of projection area for aligning a 3D model since in general projection area is stable under small changes of the vertices' coordinates.

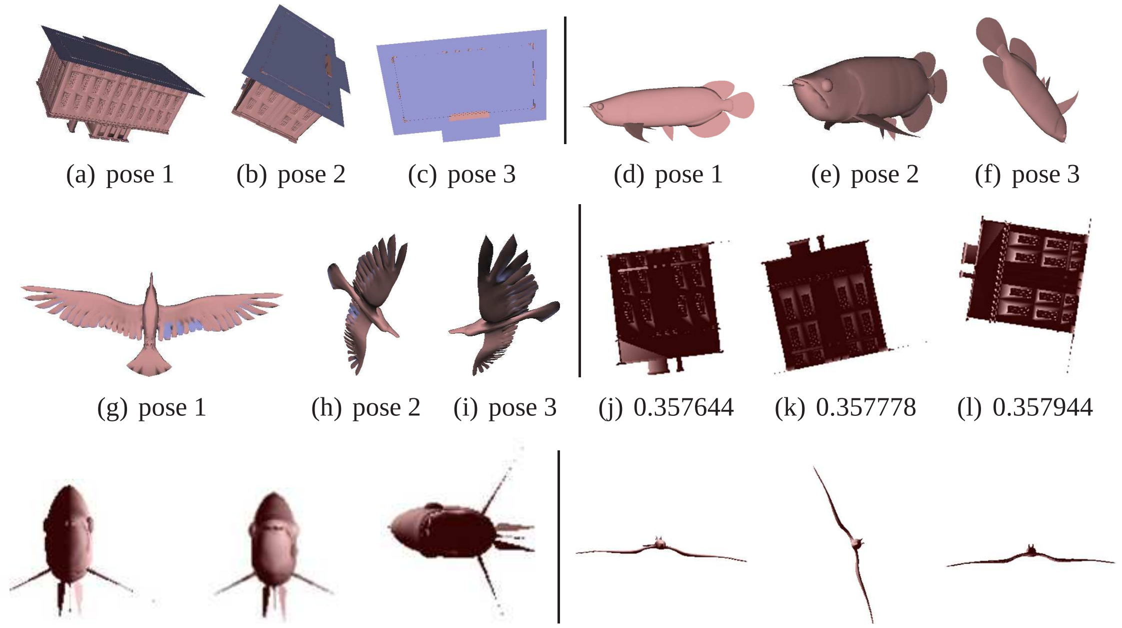


(a) 0.291556 (b) 0.292822 (c) 0.295589 (d) 0.308511

Figure 3.11: Examples indicating MPA's robustness to noise. (a) The view from the 1st principle axis with minimum projection area for the original bicycle model, (b)∼(d): the views from the 1st principle axis of the bicycle model when we added noise by randomly moving each vertex with a small displacement vector whose norm is bounded by 0.12%, 0.25% and 1% of the diameter of the model's bounding box, respectively. The number underneath each view is its normalized projection area.

(3) Robustness to initial poses. 3D models may have arbitrary initial poses. It is important for our alignment algorithm to align a model with different initial poses to the same pose. Fig. 3.12 illustrates three sets of examples indicating MPA's robustness to initial poses. As can be seen, MPA is not dependent on the initial poses of a 3D model and only a very small difference exists among the minimum area found. MPA's independence of initial poses is because we adopt the global optimization approach PSO to find the 1st principle axis with minimum projection area. In the initial stage of the search, it uses a global search to avoid local minimums and then enhances the local search ability to find an as accurate as possible global minimum projection area.

(4) Evaluation with respect to PSO's iteration number. In PSO, the number of iteration is an important factor which influences the accuracy and search time. To test the influence of iteration number on the alignment results, we apply MPA using different iteration numbers to find the 1st principle axis. Fig. 3.13 shows the results. We can find that after 11 iterations the area converges to about 0.291 and we achieve the best results which are below 0.2913 at 30∼40 iterations. We also find that the convergence speed is fast. Usually after 10 iterations, MPA already finds an area which is close to the optimal one. For the same



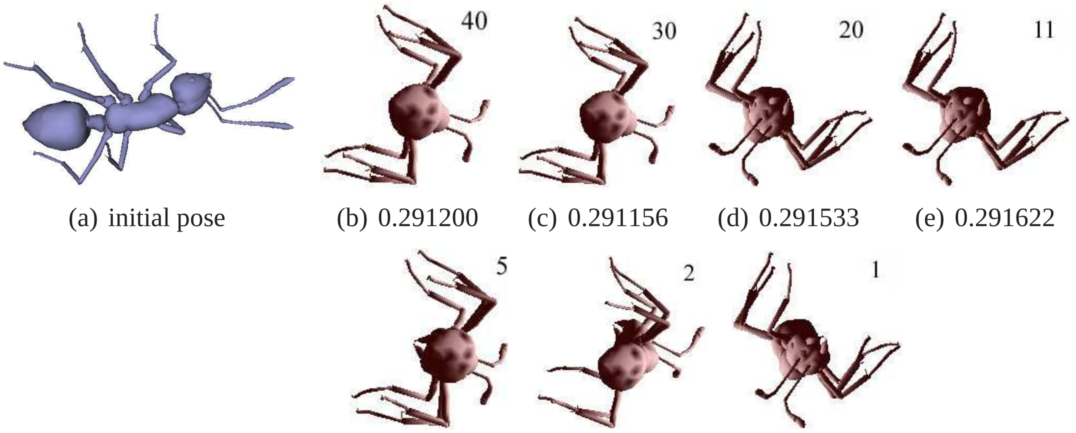
(a) pose 1 (b) pose 2 (c) pose 3 (d) pose 1 (e) pose 2 (f) pose 3

(g) pose 1 (h) pose 2 (i) pose 3 (j) 0.357644 (k) 0.357778 (l) 0.357944

(m) 0.267322 (n) 0.267300 (o) 0.267144 (p) 0.260689 (q) 0.261144 (r) 0.260900

Figure 3.12: Three sets of examples indicating MPA's robustness to initial poses. PSO is used to search for the 1st principle axis. The second row shows the corresponding views from the 1st principle axis for the models in the first row. The rotated views are only due to different up-vectors of the cameras during rendering. Note that we get the same final alignment results for each set of models. The number underneath each view is its normalized projection area.

accuracy as PSO, brute-force method needs a much longer time. For example, MPA based on 10 iterations finds smaller area than brute-force method using L_4 icosahedron (2562 vertices) for axis sampling; MPA needs about 8 seconds while brute-force method takes about 43 seconds. If based on 40 iterations, MPA finds smaller area than brute-force method based on L_6 icosahedron (40002 vertices); MPA averagely needs 46 seconds and the brute-force needs 530 seconds for the PSB models. Table 3.3 compares the average alignment time using Brute-force and PSO-based search algorithms on the NIST benchmark. On each row of the table, MPA has better alignment accuracy. We can see that the time difference between the Brute-force and PSO-based algorithms becomes larger and larger (2.0, 3.7 and 9.7) in order to achieve a more and more accurate alignment performance. However, for our PSO-based algorithm there is still a trade-off between the time and accuracy according to the requirements and available resources of the applications. Our suggestion is that selecting the value of the iteration number N_t by performing an initial test on several example models by setting different values (e.g. 10, 20, 30, 40). Based on the different results of alignment accuracy and time, we compare and decide an appropriate value for N_t. Sometimes, we also can choose different values for N_t. For example, we set a bigger value to achieve better accuracy during pre-processing target models in a database and select a smaller value when aligning a query model to speed up the retrieval.



(a) initial pose (b) 0.291200 (c) 0.291156 (d) 0.291533 (e) 0.291622

(f) 0.293244 (g) 0.293456 (h) 0.295944

Figure 3.13: Examples showing MPA's 1st principle axis results based on PSO's iteration number. (b)$\sim$(h): the axes with minimum projection area based on the iteration numbers displayed on the upper-right corner. The rotated views are due to different up-vectors of cameras during rendering. The number underneath is the normalized projection area.

3.3.3 Evaluation with Respect to Retrieval Performance

In this section, we evaluate MPA in terms of retrieval performance improvement on a rotation-dependent shape descriptor by comparing the retrieval performances when using different alignment methods such as PCA, NPCA, CPCA and our MPA. For the selection of a rotation-dependent shape descriptor, we choose to modify the Light Field descriptor [23], which is a famous and typical shape descriptor. The distance of two models is defined as the minimum distance between 10 corresponding views of the two models. The Light Field descriptor adopts an integrated image shape descriptor which contains 35 Zernike moments and 10 Fourier descriptors and use L1 distance metric to measure the differences. To find the minimum distance between two models, the original Light Field approach performs an alignment process by rotating a camera system of 20 cameras set on the vertices of a regular dodecahedron. We modify this original Light Field descriptor by replacing its internal alignment process with an explicit alignment step using PCA, NPCA, CPCA or our MPA, and we name it modified-LF.

To perform a comprehensive evaluation for 3D model retrieval performance, we employ five metrics including Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Discounted Cumulative Gain (DCG) [167] and Average Precision (AP), as described in Section 1.1.We test the modified-LF retrieval algorithm on the PSB, NIST and WMB databases using the above-mentioned different alignment algorithms. Table 3.4 compares their performances. Compared with PCA, NPCA and CPCA, our MPA achieves better performances in all the five performance metrics. The main reason for the improvement is our achieving higher percentage of consistent alignment results for models belonging to the same class. In addition, according to the experimental results in [161], the ROSy (Reflective Object Symmetry) alignment algorithm proposed by Sfikas et al. [161] has a similar performance as CPCA and NPCA in terms of retrieval performance though they use a different shape descriptor proposed in [141]. In terms of alignment accuracy, ROSy is comparable to the symmetry-based approach proposed by Chaouch and Verroust-Blondet [22]. Compared to ROSy, based on the experimental results in Table 3.4 and the accuracy results in Section 3.3.1 and [22], we anticipate that MPA will achieve better or at least comparable results in terms of either retrieval performance or axes accuracy.

Due to the performance limitation of the shape descriptor adopted, the differences of certain performance metric on some benchmarks may be small, such as the AP difference between

MPA and CPCA on the PSB dataset is 0.7. This indicates that it possibly reaches its accuracy limitation. In addition, as mentioned in Section 3.3.2, we can also choose different values for the iteration number N_t to process target and query models in order to perform an efficient retrieval.

3.3.4 Limitations of MPA

As shown in the previous experiments, MPA has a good performance in 3D model alignment. Nevertheless, it has some limitations. Firstly, it does not work well for certain types of models which do not have normalized poses with minimum projection areas. Some examples are shown in Fig. 3.9. Secondly, though in general the axes found are accurate, we cannot guarantee a perfect alignment for all models, that is the $z+$, $x+$ and $y+$ axes correspond to the front, left and top parts of a model, respectively. This is because we do not consider the semantics information of models during the alignment. Although we already utilize the static stability and view entropy, our approach is still lack of semantics information for deciding the perfect axes orientations for all 3D models.

3.4 Summary

A novel Minimum Projection Area-based alignment approach MPA for 3D model pose normalization was proposed in this chapter. It is based on the idea of finding two perpendicular principle axes with minimum projection area. PSO was employed to efficiently find the axis with minimum projection area. Three evaluation experiments were conducted: (1) Accuracy in terms of finding three axes which are parallel to the axes of the ideal canonical coordinate frame of a 3D model; (2) Robustness of results with respect to model variations, noise, initial poses and PSO iteration number; and (3) 3D model retrieval performance using a rotation-dependent shape descriptor. All the three experiments demonstrated the ability of our MPA approach to find a consistent pose for similar models. Experimental results showed that our MPA algorithm achieves a better performance compared to PCA, CPCA and NPCA in terms of axes accuracy and 3D model retrieval.

Considering the limitations of MPA, we think it can be improved by combining other type of features, such as symmetry, with projection area when searching for the principle axes. We would like to investigate this further. Another possible future work is to perform semantics analysis for axis ordering with the ultimate goal of achieving perfect alignment.

Table 3.2: List of the 41 classes of PSB in which MPA achieves a much better performance than CPCA in terms of axes accuracy percentage.

Class	#	CPCA	MPA
Helicopter	18	77.8	100.0
Enterprise spaceship	11	36.4	100.0
Dog	7	00.0	57.1
Horse	6	16.7	50.0
Rabbit	4	00.0	75.0
Snake	4	25.0	75.0
Head	16	62.5	93.8
Skull	6	00.0	66.7
Barn	5	40.0	100.0
Church	4	00.0	100.0
Gazebo	4	80.0	100.0
One story building	14	35.7	100.0
Skyscraper	5	80.0	100.0
Two story building	10	10.0	100.0
Chess set	9	66.7	100.0
City	8	37.5	75.0
Desktop computer	11	00.0	100.0
Computer monitor	13	00.0	100.0
Eyeglasses	7	71.4	100.0
Fireplace	6	00.0	33.3
Cabinet	9	66.7	100.0
School desk	4	00.0	100.0
Bench seat	11	00.0	45.5
Dining chair	11	00.0	90.9
Desk chair	15	00.0	20.0
Shelves	13	76.9	100.0
Rectangular table	25	72.0	100.0
Single leg table	6	66.7	100.0
Handgun gun	10	00.0	40.0
Ladder	4	50.0	100.0
Streetlight lamp	8	75.0	100.0
Mailbox	7	14.3	85.7
Potted plant	26	53.8	88.5
Satellite	4	25.0	50.0
Large sail boat	6	00.0	50.0
Sink	4	25.0	100.0
Slot machine	4	25.0	100.0
Hammer	4	75.0	100.0
Covered wagon	5	00.0	100.0
Semi vehicle	7	14.3	100.0
Train car	5	40.0	100.0

Table 3.3: Average alignment time (in seconds) comparison between Brute-force and PSO-based search algorithms on the NIST benchmark. T_{Brute} and T_{PSO} denote the average alignment time for the Brute-force and PSO-based alignment algorithms. "L_4/N_t=10" means the Brute-force method using L_4 icosahedron and the PSO-based method sets the iteration number N_t=10. Others are similar. In each row, the PSO-based method achieves more accurate alignment results than the Brute-force algorithm.

	Brute-force	PSO	Time difference ($\frac{T_{Brute}}{T_{PSO}}$)
L_4/N_t=10	37.7	18.7	2.0
L_5/N_t=30	144.0	38.6	3.7
L_6/N_t=40	516.5	53.3	9.7

Table 3.4: Comparison of retrieval performance among our MPA and other three alignment algorithms based on the modified-LF shape descriptor.

Methods	NN	FT	ST	DCG	AP
PSB					
MPA	**60.4**	**33.5**	**43.2**	**0.603**	**50.5**
CPCA	58.7	32.8	42.6	0.597	49.8
NPCA	57.8	32.3	41.6	0.596	49.3
PCA	58.4	31.1	40.7	0.586	48.3
NIST					
MPA	**83.5**	**42.2**	**55.2**	**0.745**	**53.8**
CPCA	81.3	41.5	53.7	0.734	52.7
NPCA	81.1	38.2	49.9	0.724	49.9
PCA	77.3	39.2	50.4	0.710	49.7
WMB					
MPA	**89.5**	**46.7**	**59.8**	**0.783**	**59.7**
CPCA	84.8	44.6	58.8	0.765	57.6
NPCA	86.3	44.2	57.5	0.765	56.5
PCA	85.5	44.2	59.0	0.764	57.1

Chapter 4

Query-by-Model: View Context-Based Retrieval

4.1 Overview

3D model feature supporting multi-modal retrieval is a challenging and significant issue in 3D model retrieval. In this chapter, we propose a novel 3D model feature named "view context" that also supports multi-modal queries. When we look at a 3D model from a view V (i.e. the viewer is located at V), the visible features in the view image form the visual information of the model from this view V. We assume a 3D model is represented as a triangle mesh. We encode the visual information of a 3D model from a view V as an integrated image shape descriptor of the silhouette view rendered from the view V. The view context of a view V is then defined as the differences of the visual information between V and a set of arranged views.

In this chapter, we apply the view context for 3D model retrieval using 3D model queries and in the following Chapter 5 we will present its application in 3D model retrieval using 2D sketch queries. In order to apply view context in Query-by-Model retrieval, we propose a view context shape descriptor for a 3D model. The proposed shape descriptor consists of the view contexts computed at several sampling views. We first align the 3D model using

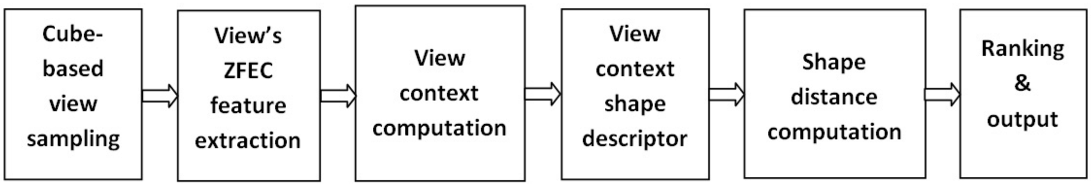

Cube-based view sampling
View's ZFEC feature extraction
View context computation
View context shape descriptor
Shape distance computation
Ranking & output

Figure 4.1: The framework of our view context-based 3D model retrieval algorithm.

our proposed minimum projection area-based method MPA presented in Chapter 3 before computing the shape descriptor for a 3D model. To further improve the retrieval accuracy, we also add the difference between the feature views sets of two models. The framework of our view context-based 3D model retrieval algorithm is shown in Fig. 4.1.

Our main contributions are as follows:

(1) We devise a new 3D model feature named view context for identifying models and it supports multi-modal queries including 3D model and 2D sketch or image. Similar models in general have similar view contexts and the view contexts of models from different classes are often distinctively different. On the other hand, view contexts of different views of the same model are often distinctively different, as well. Thus, the view context feature also demonstrates promising performance in effectively and efficiently aligning a 2D sketch with a 3D model for sketch-based 3D model retrieval applications (Chapter 5).

(2) We propose a retrieval algorithm based on view context and through experiments verify that it itself achieves a comparable retrieval performance as the related descriptors, such as Light Field and a combined shape descriptor based on view context outperforms Light Field.

The remaining of this chapter is organized as follows. Section 4.2 describes in detail the idea of view context. In Section 4.3, we propose the view context shape descriptor. In Section 4.4, an algorithm for 3D model retrieval using the view context shape descriptor is explained. The results of retrieval experiments are demonstrated in Section 4.5. Section 4.6 contains the conclusions and lists some topics for future work.

4.2 View Context

4.2.1 Definition

The view context of a particular view of a 3D model encodes the visual information differences between this view and a set of arranged views. It captures the shape appearance deviation of a 3D model.

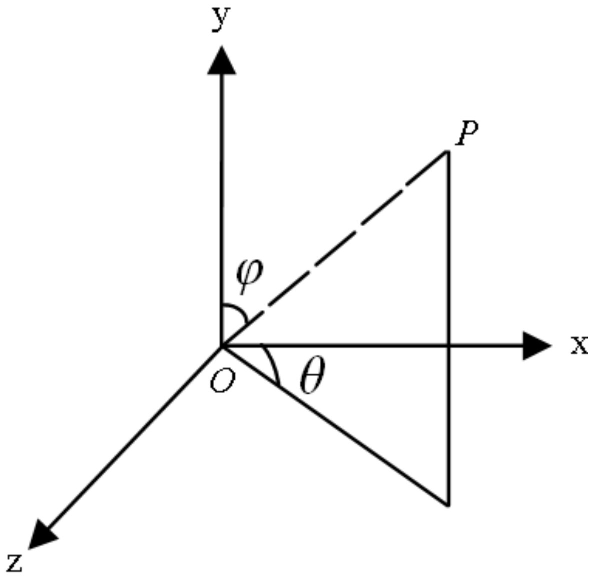
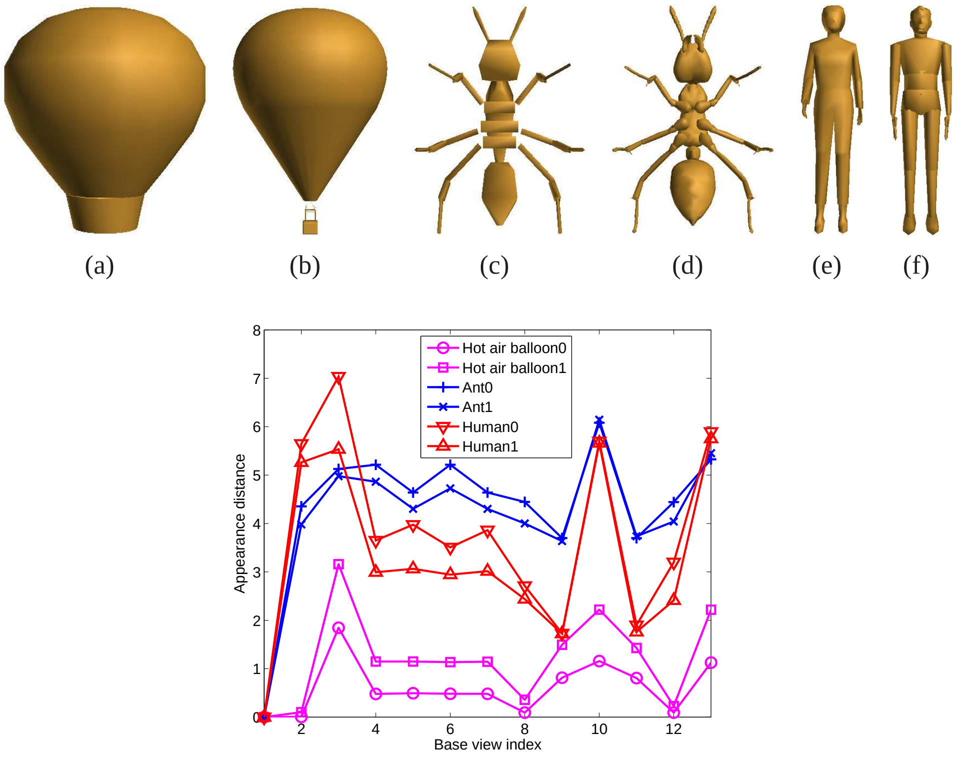


Figure 4.2: View space.

Assume that the 3D model is centered in the origin of a 3D coordinate system. Its view context from a view V_0 is defined as follows. First, we rotate the 3D model such that view V_0 coincides with the z axis of the coordinate system. Then, we orderly sample a set of views $\{(\varphi, \theta)\}$ based on the current pose. As shown in Fig. 4.2, we assume that the origin of the coordinate system is O and the camera locates at a point P, then φ is the angle



(a) (b) (c) (d) (e) (f)

(g) View context plots

Figure 4.3: View contexts of six models: (a) Hot air balloon0; (b) Hot air balloon1; (c) Ant0; (d) Ant1; (e) Human0; (f) Human1.

between the vector OP and the y axis and θ is the angle between the projection of OP on the xoz plane and the x axis. $\varphi \in [0,180]$ and $\theta \in [0,360)$. For example, view (φ,θ) can be generated by first rotating the model φ degrees about the x axis and then θ degrees about the y axis.

The view context of a view V_0 is composed of a set of feature vectors:

$$\{(\varphi,\theta,d)|(\varphi,\theta) \in \mathbf{V}\} \ , \qquad \text{(Eq. 4.1)}$$

where d is the view appearance distance between view (φ,θ) and view V_0. $\mathbf{V}$ is a sequence of base views. The methods of base view sequence sampling and view distance computation will be presented in Sections 4.2.2 and 4.2.3, respectively.

View context represents the relative appearance deviations with other views. Fig. 4.3 gives an example of the view contexts of the initial poses of six models (Fig. 4.3 (a)$\sim$(f)) in Princeton Shape Benchmark Database [167]. In these examples, the base view sequence **V** consists of 13 views based on the cube-based view sampling method described in Section 4.2.2 and the distance between two views is computed according to the method in Section 4.2.3. Fig. 4.3 (g) shows the plot of the six view context features. We can see that similar models have similar view contexts and the view contexts of different models are often distinctively different. This is useful for Query-by-Model retrieval, for which we can use view context features to differentiate different models.

4.2.2 Cube-Based View Sampling

To decide the base view sequence **V**, we need to determine the values of φ and θ in Equation (4.1). Considering the symmetrical property during our feature views generation (Section 4.2.3) as well as the balance between feature extraction time and retrieval accuracy, we sample 13 views by setting the cameras on a cube. The camera locations are (0,0,1), (1,0,0), (0,1,0), (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. 4.4, they comprise 3 adjacent face center views, 4 top corner views and 6 middle edge views. Based on these camera locations, we compute the base view sequence $\{(\varphi,\theta)\}$.

4.2.3 View Appearance Distance

To compute the view appearance distance d between two view images V_0 and V_i, we first define the feature views and then introduce an integrated image descriptor to measure their difference.

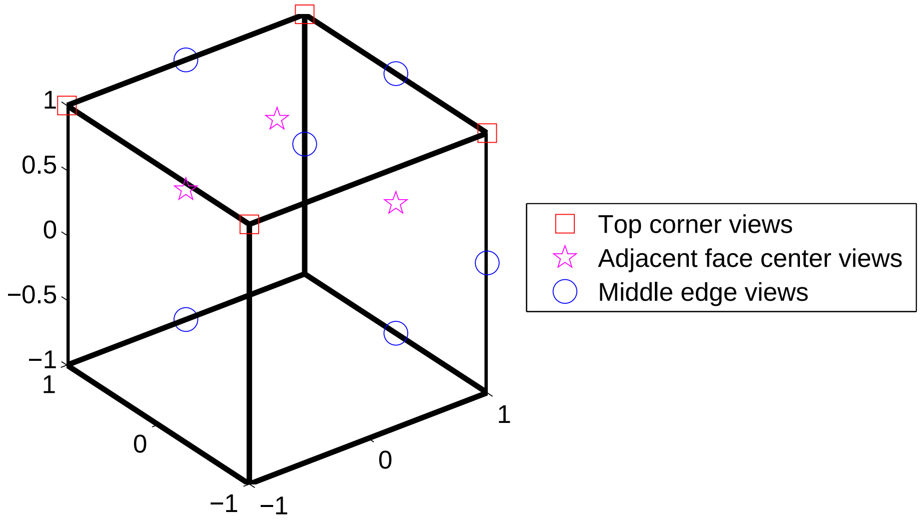


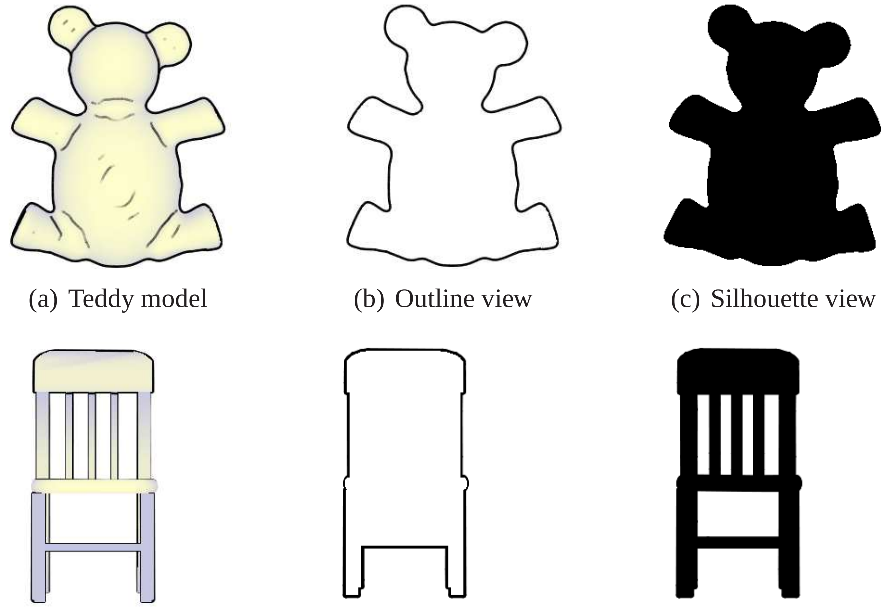
Figure 4.4: Cube-based sampling.

4.2.3.1 Feature views

To represent different features of a view, we adopt two types of feature views, which are outline and silhouette views based on orthogonal projection. Due to the orthogonal projection, both feature views have a symmetrical rendering property: two views rendered from two opposite camera locations are identical. Two examples are shown in Fig. 4.5. Outline view represents the contour information of the view while silhouette view depicts the region information of the view.

4.2.3.2 Integrated image descriptor

We compare two sets of feature views based on an integrated image descriptor, motivated by the Light Field descriptor proposed by Chen et al. [23] and their source code. To represent the region and contour features of the feature views, we adopt the shape descriptor proposed by Zhang and Luo [206] and use Zernike moments Z (up to the 10th order, totally 35 moments) to represent the silhouette view and Fourier descriptors F (the first 10 coef-



(a) Teddy model (b) Outline view (c) Silhouette view

(d) Duck model (e) Outline view (f) Silhouette view

Figure 4.5: The feature views of a 3D teddy model and a 3D chair model, respectively.

ficients of centroid distance-based Fourier descriptor [205]) to represent the outline view. In addition, to depict the geometric information of the outline view, we extract its eccentricity and circularity features. Eccentricity is to measure how much a shape deviates from a circle. For a 2D shape defined by n points $\{(x_i, y_i)|i = 1, \cdots, n\}$, we adopt the following definition to compute its eccentricity,

$$E = [\sum_{i=1}^{n}(x_i - c_x)^2 - \sum_{i=1}^{n}(y_i - c_y)^2]^2 + 4 \cdot \frac{[\sum_{i=1}^{n}(x_i - c_x)(y_i - c_y)]^2}{[\sum_{i=1}^{n}(x_i - c_x)^2 + \sum_{i=1}^{n}(y_i - c_y)^2]^2}, \quad \text{(Eq. 4.2)}$$

where (c_x, c_y) is the center of the bounding box of the 2D shape. For our case, the 2D shape is a closed curve and the range of its eccentricity of is [0,1). For example, the eccentricity of a circle is 0 and the eccentricity of an ellipse falls in the range of (0,1). Circularity is to measure the compactness of the shape. It is defined as the quotient of the area of the shape and the area of a circle with the same perimeter,

$$C = 4 * \pi * A/P^2, \quad \text{(Eq. 4.3)}$$

where C is the circularity, A and P are the area and perimeter of the shape respectively.

We use the city block (L1) distance metric to measure the distances for Zernike moments (d_Z), Fourier descriptors (d_F), Eccentricity descriptor (d_E) and Circularity descriptor (d_C):

$$d_Z = \sum_{p=1}^{35} |Z_1(p) - Z_2(p)|, \qquad \text{(Eq. 4.4)}$$

where Z_1 and Z_2 are the Zernike moments features of two silhouette views.

$$d_F = \sum_{q=1}^{10} |F_1(q) - F_2(q)|, \qquad \text{(Eq. 4.5)}$$

where F_1 and F_2 are the Fourier descriptors features of two outline views.

$$d_E = |E_1 - E_2|, \qquad \text{(Eq. 4.6)}$$

where E_1 and E_2 are the eccentricity features of two outline views, $d_E \in [0,1)$.

$$d_C = |C_1 - C_2|, \qquad \text{(Eq. 4.7)}$$

where C_1 and C_2 are the circularity features of two outline views, $d_C \in [0,1)$.

The integrated image distance d between two sets of feature views is the combination of the above four component distances,

$$d = d_Z + d_F + d_E + d_C. \qquad \text{(Eq. 4.8)}$$

The four features Z, F, E, C depict a feature view from different aspects and we follow the source code of Light Field descriptor and regard that they have the same contribution in the computation of the integrated image distance. Therefore, we linearly combine them and assign the same weight for each feature. However, a weighted sum approach should be promising in achieving even better performance.

4.3 View Context Shape Descriptor

4.3.1 Definition

The process to construct a view context shape descriptor for a 3D model is as follows: first, to achieve translation and scale invariance, we translate the center of the bounding sphere of the model to the origin of the coordinate system and then scale it to normalize the radius of its bounding sphere. To achieve rotation-invariance, we align the 3D model using our proposed minimum projection area-based alignment method MPA proposed in Chapter 3. Then we select the 13 cube-based sampling views (Section 4.2.2) as our feature views. Finally, we compute the view context of the sample views with respect to a set of selected base views (Section 4.2.2) and compute the view contexts of these views, and finally concatenate them into a 13×13 matrix as the view context shape descriptor. We want to mention that in the experiment section, for comparison we also test using the original models, that is without alignment, directly for the view sampling.

4.3.2 Invariance Properties of the View Context Shape Descriptor

The invariance properties of our view context shape descriptor are as follows.

Translation invariance. From the definition of view context, we can infer that our view context shape descriptor is translation-invariant because we have centered the 3D model at the origin before extracting its view context shape descriptors.

Scale invariance. We achieve scale invariance by scaling all the 3D models such that their bounding spheres have the same radius.

Rotation invariance. We perform a pose normalization by MPA to make it rotation-invariant. MPA maps the same model with different initial poses into the same canonical coordinate frame. Thus, our view context shape descriptor extraction is independent of the initial poses of the model.

4.3.3 Comparison with Related 3D Shape Descriptors

In this section, we compare our view context shape descriptor with three related (in terms of the structure of shape descriptor definition or view features adopted) 3D shape descriptors, which are shape context, shape distribution and Light Field descriptor.

Comparison with shape context. View context is different from shape context. View context depicts the distribution of other views' appearance deviation from the current view. Shape context of one point depicts the relative location distribution of other sampling points. In addition, a feature view can adopt different types of features, while the feature of a point can only use its relative position.

Comparison with shape distribution. The most obvious and important difference is in the features they depict. View context shape descriptor is about sample views and shape distribution is about sample points. Shape distribution is to measure the distance distribution of any two sample points of a model, while view context shape descriptor encodes the appearance deviation of a set of selected sample views with respect to a predefined base view sequence of the model.

Comparison with Light Field descriptor. The fundamental assumptions of Light Field descriptor and view context shape descriptor are different. The former is based on the idea that if two 3D models are similar, they will look similar from all similar viewing angles. Therefore, the distance between two models is defined as the minimum distance between their 10 corresponding silhouette views. The latter, however, is based on the idea that if two models are similar, they will have similar appearance variations from similar viewing angles. Therefore, we define the view context distance between two models as the summed view context dissimilarities between their sample views.

4.4 3D Model Retrieval with View Context

4.4.1 Retrieval Algorithm

We focus on retrieval using 3D models as queries. Given a query model, we propose a 3D model retrieval algorithm as follows.

(1) **Compute view context shape descriptor.** We compute the view context shape descriptor for each target model (off-line processing) and the query model (on-line processing) as described in Section 4.3.1.

(2) **Compute the shape distance matrix and ranking.** We design two shape distance metrics to measure the difference between two view context shape descriptors extracted with or without 3D model alignment. We also propose a combined shape distance by combining the dissimilarity between two view context shape descriptors and the difference between two models' feature views sets. We describe these three distances in Section 4.4.2.

4.4.2 Shape Distance Metrics

Two candidate metrics that can be used to measure the distance between two view contexts are correlation and χ^2 distance. After comparing their differentiation capabilities through experiments, we found that correlation performs better. Therefore, we use correlation to measure the difference of two view contexts. As depicted in Section 4.3.1, we select 13 cube-based sampling views as feature views set after performing our MPA alignment or using the original 3D models directly. Accordingly, we design one shape distance metric for each, described as follows.

4.4.2.1 Ordered Correlation Distance

We assume that the view contexts of two models are $\mathbf{VC}$=$\{VC_0, \cdots, VC_{12}\}$ and $\widetilde{\mathbf{VC}}$=$\{\widetilde{VC}_0, \cdots, \widetilde{VC}_{12}\}$ according to the order of the base view sequence. Since we have aligned the 3D models using MPA, the view context dissimilarity d_v between two models' view context shape descriptors is computed by summing all the 13 view context correlation differences of their corresponding pair views.

$$d_v = \sum_{i=0}^{12} (1 - corr(VC_i, \widetilde{VC}_i)). \qquad \text{(Eq. 4.9)}$$

where $corr(VC_i, \widetilde{VC}_i)$ is the correlation between the view contexts of two sample views VC_i and $\widetilde{VC}_i$,

$$corr(VC_i, \widetilde{VC}_i)) = \frac{\sum_{j=0}^{12} (VC_{ij} - \overline{VC_i})(\widetilde{VC}_{ij} - \overline{\widetilde{VC}_i})}{\sqrt{\sum_{j=0}^{12} (VC_{ij} - \overline{VC_i})^2 \sum_{j=0}^{12} (\widetilde{VC}_{ij} - \overline{\widetilde{VC}_i})^2}}. \qquad \text{(Eq. 4.10)}$$

where $\overline{VC_i}$ and $\overline{\widetilde{VC}_i}$ are the mean value of view contexts VC_i and $\widetilde{VC}_i$, respectively.

4.4.2.2 LAP Correlation Distance

For a view context shape descriptor directly extracted based on the initial 3D models, we use the Jonker's Linear Assignment Problem (LAP) algorithm [72] to correspond these two sets of feature views and use the minimal matching cost as the distance between them.

4.4.2.3 Combined Shape Distance

To further improve the retrieval performance, we also consider the difference between two sparse sample views sets to form a combined shape descriptor. We propose a combined shape distance by combining the dissimilarity between two view context shape descriptors, depicted by d_v, and the integrated shape descriptor feature (Section 4.2.3.2) difference between two models' sample views sets, depicted by d_m. We combine the two distances based on an automatic weighting method according to the differentiation ability of each type of distance. First, we normalize d_v and d_m into $\widetilde{d_v}$ and $\widetilde{d_m}$ by their respective maximum distances based on all the models in the database. Then, we compute the weights ω_v and ω_m for the view context feature distance and sample views' difference,

$$\omega_v = \frac{s_1}{s_1+s_2},\ \ \omega_m = \frac{s_2}{s_1+s_2}\ . \qquad \text{(Eq. 4.11)}$$

s_1 and s_2 are the standard deviations of d_v and d_m over all the models in the database. Finally, we combine these two normalized features by their corresponding weights,

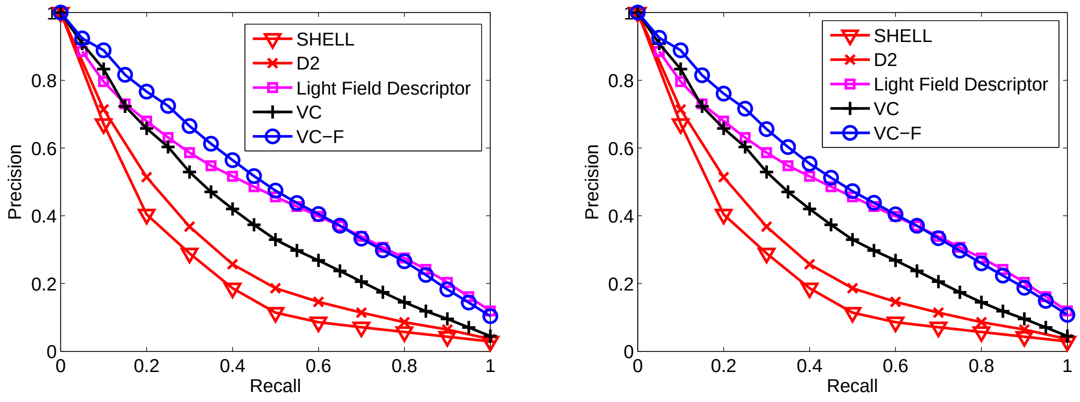
$$d = \omega_v * \widetilde{d_v} + \omega_m * \widetilde{d_m}\ . \qquad \text{(Eq. 4.12)}$$

4.5 Experiments and Discussion

To investigate the retrieval performance and the characteristics of our view context shape descriptor, we tested our view context descriptor on the PSB (test dataset) and NTU benchmarks (Section 1.1) and compared it with other related descriptors. To compare the retrieval performance, we use the six metrics described in Section 1.1: Precision-Recall, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Discounted Cumulative Gain (DCG) and Average Precision (AP).

4.5.1 PSB Database

Fig. 4.6 and Table 4.1 compare the performances of our two view context-based shape descriptors, denoted by VC and VC-F as well as other three shape descriptors. For the performances of D2 [135] and SHELL [4], we referred to the experiment results in [167]. For Light Field descriptor [23], we generated the results using their provided execution file. Fig. 4.6 (a) shows the results on the MPA aligned PSB database, while Fig. 4.6 (b) shows the results on the original PSB database.



(a) (b)

Figure 4.6: Precision-Recall plots of our view context and other three descriptors: (a) MPA aligned PSB database: Ordered correlation distance; (b) Original PSB database: LAP correlation distance. “VC”: our view context descriptor. “VC-F”: combined shape descriptor by integrating the sample views’ difference and VC.

Compared to the shape distribution (D2) and SHELL descriptor (one type of shape histogram descriptor), view context descriptor performs apparently better in terms of all the six metrics. From Fig. 4.6, we can see that compared with the Light Field descriptor, view context shape descriptor can achieve a better or comparable Precision-Recall performance when retrieving a certain percentage of models (for example around 15 percent). While the

Table 4.1: Other performance metrics for the performance comparison on the PSB database.

Methods	NN	FT	ST	DCG	AP
Ordered correlation distance					
VC-F	60.9	35.9	46.3	0.627	**50.8**
VC	49.7	25.9	34.9	0.532	39.9
LAP correlation distance					
VC-F	61.6	35.8	46.2	0.627	**50.8**
VC	49.1	25.9	34.9	0.535	39.9
Light Field Descriptor	**65.7**	**38.0**	**48.7**	**0.643**	48.0
D2	31.1	15.8	23.5	0.434	29.7
SHELL	22.7	11.1	17.3	0.386	24.4

combined shape descriptor VC-F integrating the difference of sample views sets can improve the performance further and we can achieve better Precision-Recall results than the Light Field descriptor. As shown in Table 4.1, in terms of other performance metrics including NN, FT, ST and DCG, Light Field outperforms VC-F on this benchmark; while in terms of the AP metric, VC-F performs better, which indicates a better overall performance.

4.5.2 NTU Database

For the full MPA aligned or original database (1833 models), we compared our view context shape descriptors with the three descriptors mentioned in [23]: Light Field [23], 3D harmonics [78], and multiple view descriptor [68]. Fig. 4.7 and Table 4.2 show the comparisons of their performances. For the Light Field descriptor, we ran their program on the NTU database based on their provided execution file, while for 3D harmonics and multiple view descriptor, we referred to the experiment results in [23]. Some performance metrics that are not provided in these papers are indicated by "-". We can see that our view context shape descriptor apparently outperforms the 3D harmonics and the multiple view descriptor on all the six metrics. As can be seen from Fig. 4.7, it itself can achieve a similar and comparable Precision-Recall performance as the Light Field descriptor while the combined shape descriptor VC-F evidently outperforms the Light Field descriptor. As shown in Table 4.2, in terms of ST and DCG, VC-F is comparable to the Light Field descriptor; in terms of NN and FT, Light Field descriptor performs better; while in terms of AP, VC-F outperforms the Light Field descriptor, similarly demonstrating a better overall performance of our VC-F.

Fig. 4.8 and Table 4.3 compare the performances between our view context based methods and Light Field descriptor over the 549 classified models. Similarly, we can see that our view context itself is comparable to the Light Field descriptor in terms of the Precision-Recall metric and integrating the difference of feature views sets can apparently improve

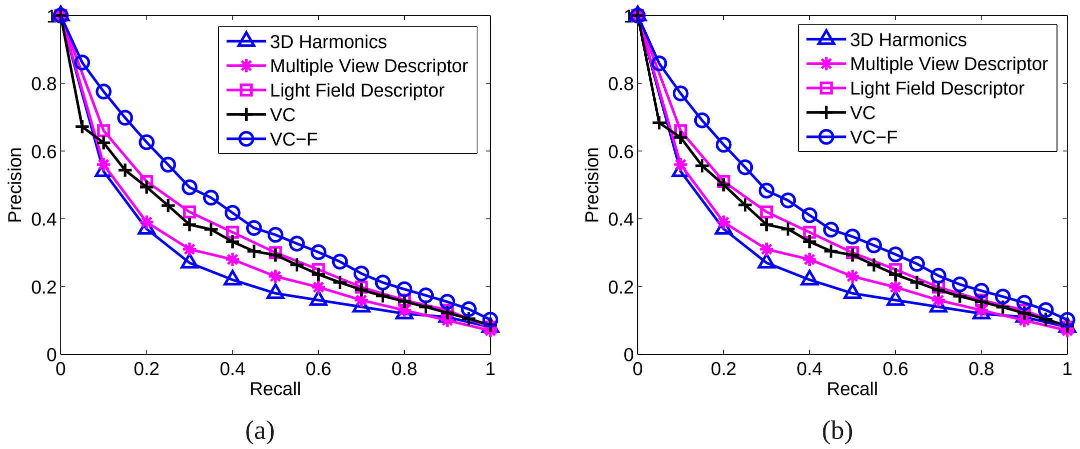
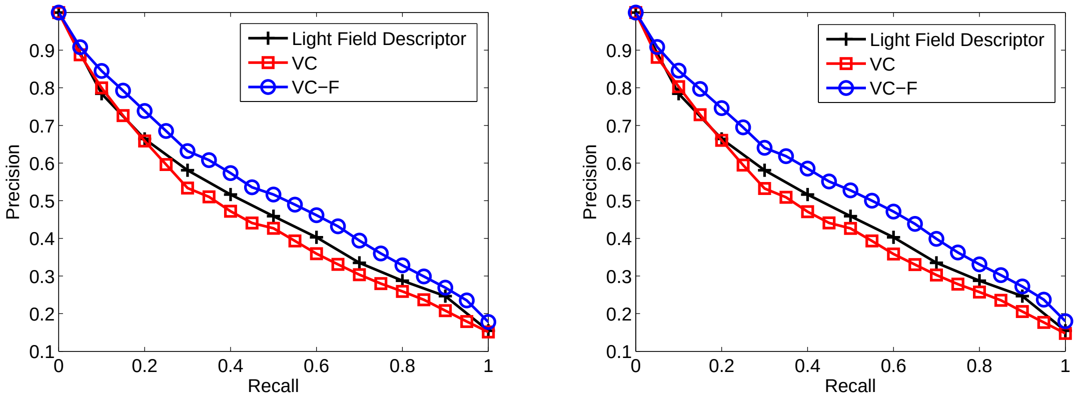


Figure 4.7: Precision-Recall plots of our view context and other three descriptors: (a) MPA aligned complete NTU database (1833 models): Ordered correlation distance (Section 4.4.2.1); (b) Original complete NTU database (1833 models): LAP correlation distance (Section 4.4.2.2). "VC": our view context descriptor. "VC-F": combined shape descriptor by integrating the sample views' difference and VC.

the Precision-Recall performance. Overall, we can achieve better results than the Light Field descriptor, in terms of almost all the performance metrics.

Table 4.2: Other performance metrics for the performance comparison on the complete NTU database (1833 models).

Methods	NN	FT	ST	DCG	AP
Ordered correlation distance					
VC-F	59.2	29.6	**39.2**	0.592	**40.9**
VC	54.2	25.3	33.5	0.520	33.0
LAP correlation distance					
VC-F	58.8	29.3	38.5	0.588	40.3
VC	52.1	25.2	33.6	0.519	33.2
Light Field Descriptor	**63.5**	**31.2**	39.1	**0.604**	35.3
Multiple View Descriptor	-	-	-	-	28.9
3D Harmonics	-	-	-	-	26.5



(a) (b)

Figure 4.8: Precision-Recall plots of our view context and other three descriptors: (a) MPA aligned classified NTU database (549 models): Ordered correlation distance (Section 4.4.2.1); (b) Original classified NTU database (549 models): LAP correlation distance (Section 4.4.2.2). "VC": our view context descriptor. "VC-F": combined shape descriptor by integrating the sample views' difference and VC.

4.5.3 Timing Information

On average, it takes 4.6 seconds and 6.1 seconds to extract the view context feature for a model in PSB and NTU respectively using a computer with an Intel Xeon CPU E5520@2.27 GHz and 12.0 GB of RAM. It takes only 0.27 millisecond for a pair VC-F feature comparison, that is, 0.25 and 0.49 second of matching time for a query on PSB and NTU,

Table 4.3: Other performance metrics for the performance comparison on the classified NTU database (549 models).

Methods	NN	FT	ST	DCG	AP
Ordered correlation distance					
VC-F	67.2	41.5	53.3	0.689	53.5
VC	61.5	33.5	45.0	0.629	45.9
LAP correlation distance					
VC-F	67.8	**42.2**	**53.8**	**0.696**	**54.1**
VC	60.1	33.4	44.9	0.628	45.8
Light Field Descriptor	**70.0**	39.0	50.1	-	48.4

respectively. We also need to mention that our implementation is not optimized. Comparatively, Light Field needs 2.6 seconds for feature extraction for a model in PSB or NTU database. 2 milliseconds are needed to perform a pair feature comparison, thus, 1.9 and 1.2 seconds are required to perform a query on PSB and NTU respectively. Overall, our view context-based retrieval algorithm has a comparable efficiency performance as the Light Field descriptor and also meets the requirements for a real-time retrieval application.

4.5.4 Limitations

We have demonstrated the comparable performance of VC as well as the superior performance of VC-F when compared with the related shape descriptors. Since view context mainly measures the deviation in the visual information features of a 3D model based on a set of base views and sample views, combining it with the visual information itself in the sample views can achieve better differentiation ability and the resulting combined shape descriptor improves the retrieval performance further and it outperforms the related view-based shape descriptor Light Field.

Nonetheless, there are some limitations. Firstly, the performance of VC itself is not among the state-of-the-art compared to other top view-based retrieval techniques, such as BF-DSIFT [44] and CM-BOF [100], let alone other view-based retrieval algorithms which further utilize distance metric learning techniques to further improve the retrieval performances, such as the DG1SIFT shape descriptor [130] which performs the best in the SHREC 2012 generic 3D model retrieval track [127]. The performance of VC can be boosted further. For example, utilizing a more powerful 2D feature for view context computation can improve VC's performance further. Secondly, rather than combining another visual information feature to form the combined one VC-F, a hybrid shape descriptor that contains the visual information feature of VC and some certain geometrical features to depict a 3D model, has not been tested and compared. It should have chance to outperform the existing one. Thirdly, the influence of different parameter settings related to the view context definition on the retrieval performances can be further explored.

4.6 Summary

We have presented a new 3D model feature view context which captures the shape deviation distribution feature of a 3D model. It can differentiate models effectively because similar models have similar view contexts and different models in general have apparently different view contexts. To improve the retrieval accuracy, we propose a combined shape descriptor which also integrates the difference of feature views. Experiment results show that the view context shape descriptor is comparable with the related view-based descriptors in retrieval performance and the combined shape descriptor can achieve a superior performance than the related ones.

There are still many facets about the view context to be explored. For example, we can adopt a different view sampling method, such as setting the cameras on the 20 vertices of a regular dodecahedron. In addition, since the result of shape analysis is often dependent on the scale selected, a multi-scale view context can be utilized. We can organize a view context scale-space by uniformly dividing the view space for the base view sampling at a series of scales, arranged from coarse to fine, and then compute view context at each scale. The view context features at several scales are utilized jointly for the shape analysis task.

Chapter 5

Query-by-Sketch: Incorporating 2D-3D Alignment

5.1 Overview

Currently, there exist many sketch-based 3D model retrieval algorithms such as [43, 75, 88, 126, 158, 192, 201]. However, to the best of our knowledge, all the available approaches compare the query 2D sketch with a very limited number of sample views of the 3D model. For example, Funkhouser et al. [43] only sampled 13 views rendered from 4 top corners, 6 edge midpoints and 3 adjacent face centers of a cube; Kanai [75], Yoon et al. [201] and Saavedra et al. [158] sampled only 14 views comprising 6 orthographic and 8 isometric views by sampling viewpoints on a cube or a sphere. In fact, this sparse view sampling approach is subject to inaccurate 2D sketch-3D model correspondence because the pose of the query sketch, that is, the viewpoint of the viewer when drawing the sketch, may have big difference with any of the sample views. Thus, the 2D-3D correspondence is not robust based on only several sample views generated using predefined fixed sample locations.

When retrieving 3D models using a 2D query sketch, we need to compute the distance between the 2D sketch and the 3D model. Ideally, it is good if we compare the 2D sketch with the most similar view or the optimal corresponding view of the 3D model. However, if we sparsely sample a limited number (e.g. 3∼24 in previous work) of views, the chance that the optimal view is among the selected sample views will be low. However, due to the high computational cost, we also cannot exhaustively compare with a large amount of sample views of a 3D model. Experimental results in Section 5.5 show that considering more views of a target 3D model to correspond with a query sketch for a more accurate 2D-3D correspondence improves the retrieval performance.

Motivated by the above findings and in order to improve the retrieval performance, we propose a novel sketch-based 3D model retrieval algorithm which first performs a 2D sketch-3D model alignment before 2D-3D matching. The framework of our sketch-based 3D model retrieval algorithm is shown in Fig. 5.1. Our proposed 2D sketch-3D model alignment utilizes the 3D model feature view context proposed in Chapter 4 to rapidly select some candidate views from a set of densely sampled views. View context is utilized because we have found a new property of it: view contexts of different views of the same model are often distinctively different. This property facilitates us to distinguish different views during the candidate views selection for 2D-3D correspondence. Our sketch-based retrieval algorithm is composed of two stages which are precomputation and retrieval. The retrieval stage comprises two steps which are 2D-3D alignment and 2D-3D matching. The effectiveness as well as the robustness of our approach are demonstrated by comparative and evaluative experiments, using both hand-drawn sketches and standard line drawings as queries and a standard 3D model dataset as target database. Moreover, we have achieved a better performance than several latest sketch-based retrieval algorithms.

The rest of this chapter is organized as follows. Feature extraction and feature distance computation methods for 3D model and 2D sketch are presented in Section 5.2 and Section 5.3, respectively. In Section 5.4, we present our sketch-based 3D model retrieval algorithm. Experiments are conducted in Section 5.5. Section 5.6 concludes the chapter and lists several future research directions.

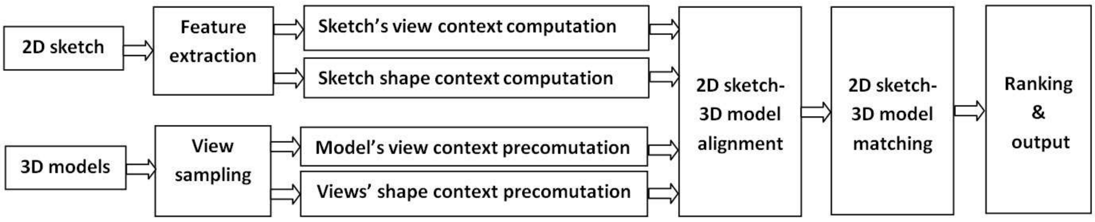

2D sketch
Feature extraction
Sketch's view context computation
Sketch shape context computation
3D models
View sampling
Model's view context precomutation
Views' shape context precomutation
2D sketch-3D model alignment
2D sketch-3D model matching
Ranking & output

Figure 5.1: The framework of our sketch-based 3D model retrieval algorithm.

5.2 Feature Extraction

To measure the difference between a 2D sketch and the views of a 3D model effectively and efficiently, we need to extract similar yet simple features. In our algorithm, to represent different features of a view, we extract **silhouette** and **outline** feature views for both 2D sketches and 3D models. Silhouette feature view is selected because of its robustness for the 2D sketch-3D model alignment, while outline view is chosen because of its better accuracy in selecting the relevant models during the 2D-3D matching in the retrieval stage. Silhouette and outline feature views are simple in essence and often coexist in both the sketches and the sample views of an object and thus form a simple and similar feature set. Compared to the features in the related work section, such as 3D spherical harmonics, generic Fourier descriptor (GFD), local binary pattern (LBP), multi-scale convexity/concavity (MCC) as well as diffusion tensor fields feature representations of suggestive contours, the features we selected have the virtues of simpleness and low computational complexity. In Section 4.2.3.1, we have present extracting silhouette and outline feature views for 3D models. Now we mainly introduce the feature view extraction method for 2D sketches.

5.2.1 2D Sketch Feature Extraction

A sketch is composed of a set of curves. It can be: (1) a hand-drawn sketch drawn by non-artist people, such as the sketches built by Yoon et al. [201]; (2) a sketch drawn by artists, for example, the sketch dataset created by Cole et al. [25]; or (3) a standard line drawing such as the 260 Snograss and Vanderwart's standardized 2D object images [171].

We need to extract the silhouette and outline feature views for a 2D sketch to correspond with a 3D model. We generate a silhouette view based on the following steps: binarization, Canny edge detection, morphological closing operations (repeat until the result no longer changes), gap connection and region filling. After that, we apply the Canny edge detector on the binary silhouette image to extract the outline of the sketch. Fig. 5.2 gives an example indicating the silhouette feature view generation process. Fig. 5.3 illustrates two groups of examples of hand-drawn sketches in Yoon et al. [201] and standard line drawings in Snograss and Vanderwart [171].

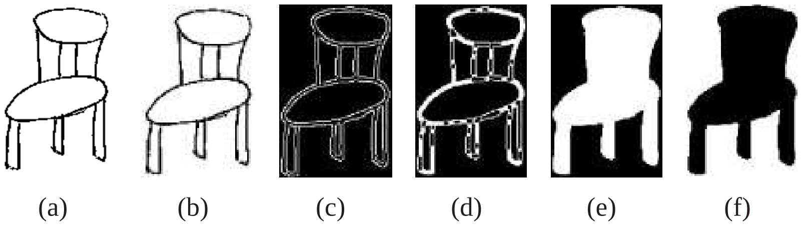
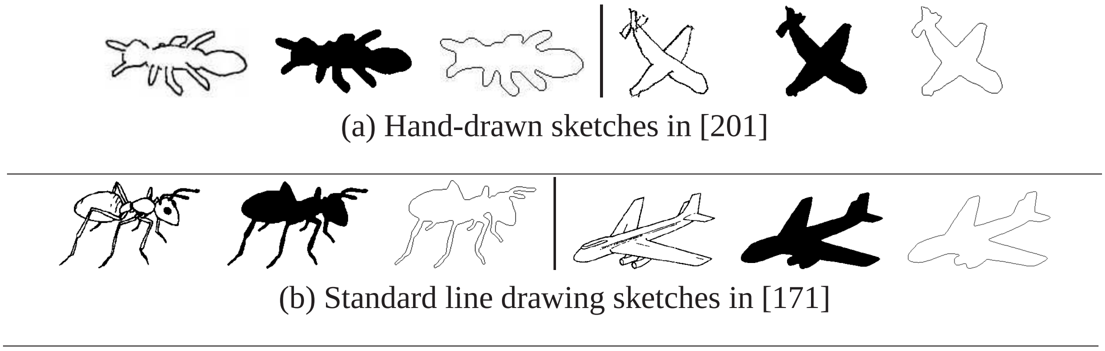


Figure 5.2: Silhouette feature view generation example: (a) original sketch; (b) binarization; (c) Canny edge detection; (d) morphological closing; (e) gap connection and region filling; (f) inversion.



(a) Hand-drawn sketches in [201]

(b) Standard line drawing sketches in [171]

Figure 5.3: Four sets of examples of sketch feature extraction for both hand-drawn sketches in [201] and standard line drawings in [171]. For every set, from left to right: sketch, silhouette view and outline view.

5.3 Feature Distance

To compute the distance between two feature views, we need to extract appropriate shape descriptors to balance the efficiency and accuracy in different stages of our retrieval algorithm. For the view context feature extraction used in the precomputation stage and the 2D-3D alignment step in the retrieval stage, we adopt a computationally efficient integrated image descriptor presented in Section 4.2.3.2. For the 2D-3D matching during the retrieval stage, we utilize the more accurate relative shape context descriptor as follows.

5.3.1 Relative Shape Context Matching Distance

We use the relative shape context matching [10] to compute a more accurate distance to measure the difference between the sketch and each candidate view resulting from the alignment step during the retrieval stage. Relative shape context (Section 2.3.4) is defined to achieve rotation invariance property and it is necessary for our sketch-based retrieval scenario, for which sample views should be independent of camera up-vectors during rendering and the orientation of the sketch.

To compute the difference between two outline feature views, we first sample a set of feature points for each image and then use the relative shape context matching algorithm described in [10] to measure their distance.

(1) Feature points sampling. We sample 100 points for every outline feature view based on the following steps: contour extraction, cubic B-Spline interpolation and uniform sampling.

(2) Relative shape context matching. We first extract the relative shape context feature [10] for every feature point in an outline view and then adopt Jonker's LAP algorithm [72] to correspond the feature points of two outline views and finally use the minimum matching cost to measure their distance. To compute the relative shape context, we compute the

tangent vector to define the local relative *x* axis for each sample point. This can be easily achieved considering that we use a cubic B-spline to interpolate the contour during the above feature points sampling process and the derivative curve of a cubic B-spline curve is a quadric B-spline curve [151].

5.4 Our Sketch-Based 3D Model Retrieval Algorithm

As described in Section 5.1 and Section 2.3, many previous sketch-based 3D model retrieval algorithms (e.g. [88, 126, 192]) sample only a limited number (e.g. 3∼24) of views to match a 3D model with a query 2D sketch. Apparently, as mentioned in Section 5.1, this sparse view sampling approach will limit the accuracy of the 2D-3D correspondence. This is because if the pose of the query sketch is apparently different from those of the limited number of predefined sampling views, the 2D-3D correspondence is not accurate. Thus, the 2D-3D matching distance cannot represent the real difference between the 2D sketch and the 3D model. Motivated by the above findings, we propose to first perform a 2D sketch-3D model alignment step to find a set of candidate views for the 2D-3D correspondence and then compute the 2D-3D matching distance based on the candidate views. It should be noted that as compared in Section 2.3.3 and Section 2.5.4, our 2D sketch-3D model alignment is different from the common 2D image-3D model registration techniques.

In this section, we present a sketch-based 3D model retrieval algorithm utilizing the 3D model feature view context presented in Chapter 4 and relative shape context matching [10]. It includes two stages: precomputation and retrieval. During the retrieval stage, we first select a set of candidate views to align a 2D sketch with a 3D model based on the precomputed view context features of the 3D model before measuring their more accurate distances, in terms of relative shape context matching cost. The 2D-3D alignment step avoids brute-force direct matching between the sketch and many sample views, that is reducing the search space to only a set of candidate views, by utilizing the features of all the densely sampled views to efficiently shortlist several good candidate views for a more accurate 2D-3D correspondence.

The idea of view context was originally proposed in Chapter 4 for 3D model retrieval using 3D model queries. In [96] we found that view context can be utilized to align a 2D sketch with a similar 3D model and performed preliminary tests on some models. Based on Chapter 4 and [96], we found a new property of view context: for different views of the same model, their view contexts are often different. Therefore, view context can be utilized to distinguish different sample views of the model, thus useful for candidate views selection for the 2D-3D alignment. Based on this, we develop our main idea to align a 3D model with a 2D sketch as follows: we replace each sample view of the 3D model with the sketch and compute its view context and if the obtained new view context is very similar to its original one, then this sample view is considered as a candidate view for the 2D-3D alignment.

5.4.1 View Context

To meet the requirements of the 2D-3D alignment step in our sketch-based retrieval algorithm, we modify the view context proposed in Chapter 4 and [96] by choosing a fixed set of base views described as follows and an integrated image descriptor presented in Section 4.2.3.2 for feature distance computation. For a 3D model centered at the origin, we select a series of views as base view set $\mathbf{V}^b$,

$$\mathbf{V}^b = < V_1^b, V_2^b, \ldots, V_m^b >, \qquad \text{(Eq. 5.1)}$$

where m is the number of base views. For a view V, its view context is defined as the visual information differences between V and each view in the base view set $\mathbf{V}^b$,

$$\{d(V, V_j^b) | V_j^b \in \mathbf{V}^b, 1 \leq j \leq m\} \ , \qquad \text{(Eq. 5.2)}$$

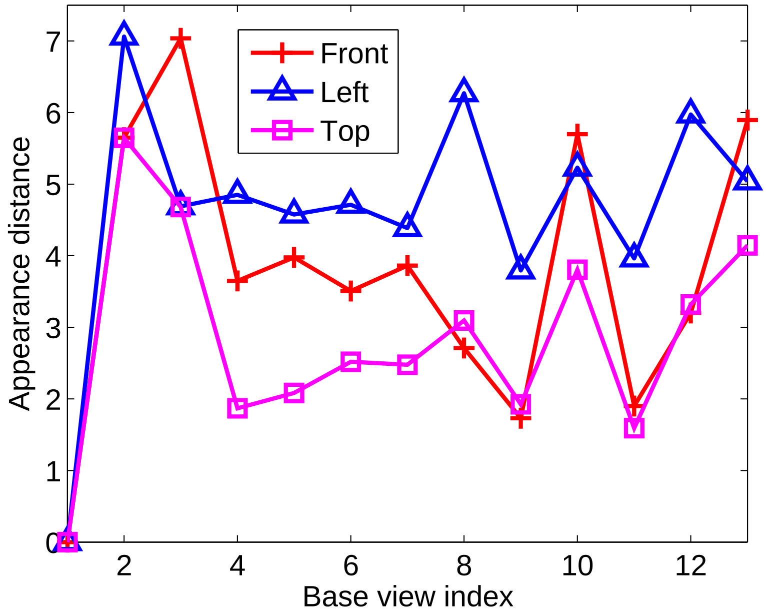


Figure 5.4: An example indicating that view contexts of different views of the same model are often different. The view contexts of the front, left and top views of the model Human0 in Fig. 4.3(e) are shown.

where $d(V, V_j^b)$ is the integrated image distance (Section 4.2.3.2) between V and V_j^b. Thus, view context measures the shape appearance deviation feature of a 3D model with respect to a set of base views.

In Fig. 4.3, we show the view contexts of several models. Moreover, we found that view contexts of different views of the same model are also often different, as shown in Fig. 5.4. This newly found property is important for sketch-based retrieval framework to distinguish different sample views of a 3D model for the 2D-3D alignment.

5.4.2 Precomputation Stage

To speed up retrieval, we precompute the view context and relative shape context features for a set of sample views of each target 3D model in database.

5.4.2.1 View Context Precomputation

The view context feature computation for every 3D model is detailed as follows.

(1) Base and sample views definitions. We define the viewpoints for the base and sample views by subdividing an icosahedron based on the Loop subdivision rule [108]. Fig. 5.5 shows the view sampling by subdividing the icosahedron (L_0) once (L_1) and twice (L_2) and we set the cameras at the vertices of the subdivided icosahedron for the base and sample view sequence generation. Considering the symmetrical property in rendering the feature views (Section 4.2.3.1), we select half-L_1 (select one from pair symmetric vertices, 21 views) for the base views and half-L_2 (81 views) for the sample views. We denote the sample view set $\mathbf{V}^s$ as follows,

$$\mathbf{V}^s =< V_1^s, V_2^s, \ldots, V_n^s >, \qquad \text{(Eq. 5.3)}$$

where n is the number of sample views. Thus, n=81, m=21.

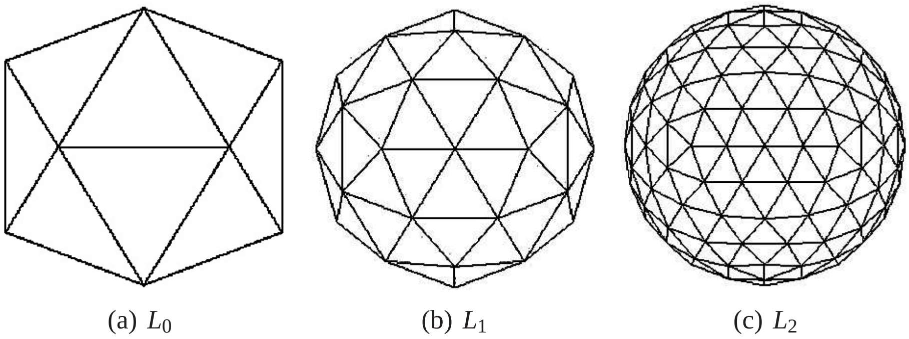

(a) L_0 (b) L_1 (c) L_2

Figure 5.5: Viewpoints sampling method. L_0: icosahedron; L_1, L_2: subdivide the icosahedron once and twice.

(2) View context feature extraction. We compute the integrated image distance (Section 4.2.3.2) between each sample view in $\mathbf{V}^s$ and each base view in $\mathbf{V}^b$. Assume that d_{ij} $(i = 1, \ldots, n; j = 1, \ldots, m)$ is the distance between the sample view V_i^s and the base view V_j^b, then for each model we form an $n \times m$ view distance matrix $D^s = \{d_{ij}\}_{n \times m}$. The i^{th} row represents the view context feature of the sample view V_i^s, that is, $D_i^s =< d_{i1}, d_{i2}, \ldots, d_{im} >$.

5.4.2.2 Relative Shape Context Precomputation

We also precompute the relative shape context features (Section 5.3.1) for the sample views of each 3D model. They will be used in the retrieval stage. To improve the storage efficiency, we adopt a sparse matrix representation to denote the relative shape context features and only keep the feature values that are non-zeros (e.g. $>$1e-5) and save them into a series of three-dimensional vector $<\theta, r, value>$, where (θ, r) denotes one bin (θ: orientation, r: distance) of the relative shape context partition, for which we use the default 5×12 partition. During the retrieval stage, we thus only need to extract the relative shape context features for the query sketch.

5.4.3 Retrieval Stage

Based on the precomputed view context and relative shape context features of the 81 sample views for each target 3D model, we develop a retrieval algorithm comprising two steps: 2D-3D alignment and 2D-3D matching. The details are as follows.

5.4.3.1 Step 1. 2D-3D Alignment

(1) 2D sketch feature extraction. First, we extract the silhouette and outline views of the query 2D sketch based on the method in Section 5.2.1 and then compute its Zernike moments, Fourier descriptors, eccentricity and circularity features (Section 4.2.3.2).

(2) Sketch's view context feature extraction. Similar to the view context precomputation (Section 5.4.2.1) for a 3D model, we compute the integrated image distances (Section 4.2.3.2) between the sketch and all the base views of the target model and name the resulting distance vector D^k=$<d_1,d_2,\ldots,d_m>$ sketch's view context.

(3) 2D-3D alignment. To align the 2D sketch and a 3D model before 2D-3D matching, we choose some candidate views by keeping a certain percentage T (e.g. 20%, 10% or 5%, that is, 16, 8 or 4 sample views in our experiments) of the sample views with top view context similarities as the sketch, in terms of correlation similarity S_i,

$$S_i = \frac{D_i^s \cdot D^k}{\left\|D_i^s\right\| \left\|D^k\right\|}. \qquad \text{(Eq. 5.4)}$$

where, D_i^s (defined in Section 5.4.2.1) and D^k are the view contexts of the sample view V_i^s of the 3D model and the 2D sketch, respectively.

5.4.3.2 Step 2. 2D-3D Matching

(1) Sketch-model distance computation. To more accurately measure the similarity between the sketch and the model as well as to encompass the orientation differences between the sketch and the sample views, we compare the sketch with every candidate outline view using the relative shape context matching (Section 5.3.1) and regard the minimum relative shape context distance obtained as the sketch-model distance.

(2) Ranking and output. We sort all the sketch-model distances between the sketch and the models in an ascending order and list the retrieved models accordingly.

5.5 Experiments and Discussion

To evaluate our sketch-based retrieval algorithm using a 2D-3D alignment, we perform comparative and evaluative experiments based on both hand-drawn and standard line drawing query sketches, as well as a standard 3D model database. We would like to mention that our 2D sketch-3D model alignment is different from the previous 2D image-3D model registration techniques, where the 2D image contains the view of the same object as the

3D model. Thus, they can use objective metrics to measure the alignment accuracy. However, for us the object in the 2D sketch is not completely the same as the 3D model and thus it is not one of its complete views. Therefore, there is no one exact pose to perfectly align the 2D sketch with 3D models. As a result, we mainly evaluate the alignment accuracy by comparing the robustness (change in performance) of our retrieval algorithm while reducing the number of candidate views during the alignment.

5.5.1 Hand-Drawn Sketches

We first test and evaluate our sketch-based retrieval algorithm by performing a similar experiment as the one described in a 2010 paper by Yoon et al. [201]. They built a benchmark database by using the first 260 models (13 classes, 20 each) of WMB dataset (Section 1.1) as target 3D model dataset and 250 hand-drawn sketches as the query set. For each class, one typical 3D model and sketch are shown in Fig. 5.6. We need to mention that: (1) to compare with the available retrieval results obtained by Yoon et al. in this section, we select the same sketches as those in their paper; (2) the hand-drawn sketches were drawn by non-artists and some of them are very simple sketches, e.g., using only 4 line segments to represent an ant. We will give the overall performance of our approach later.

For the precomputation (Section 5.4.2), on average it takes 97 seconds to process a model using a computer with an Intel Xeon CPU E5520@2.27 GHz and 12.0 GB of RAM: 8.8 seconds for the view context precomputation and 88.2 seconds for the relative shape context precomputation, for all the 81 sample views of the 3D model. During the retrieval stage (Section 5.4.3), we set the default value for the percentage T for candidate views selection (Section 5.4.3) to be 20%, that is, keeping top 16 candidate views. We use the sketches in Fig. 5.6 as queries and the top-20 retrieved models are listed respectively in Fig. 5.7. Compared to the retrieval results obtained by Yoon et al. [201], as shown in Fig. 5.8, our

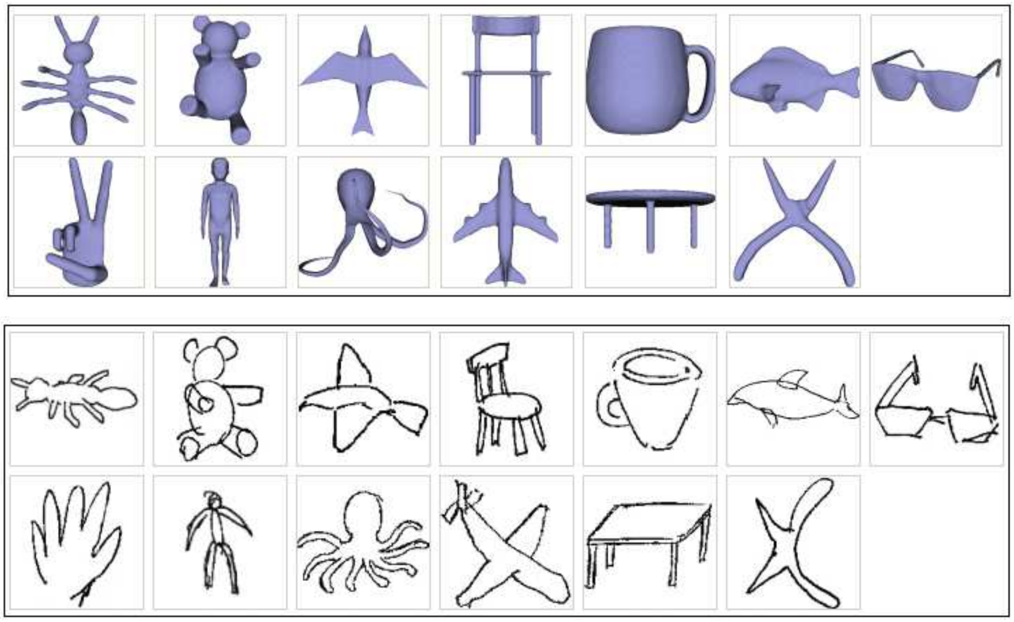


Figure 5.6: Typical 3D model and 2D sketch for each class of Yoon et al.'s [201] benchmark.

retrieval lists are better for the bear, ant and hand queries and comparable for the chair and cup queries. For the human and glasses sketches, Yoon et al. achieved better results (in Section 5.5.3, we will show that our approach achieves better performance on class level). For the seven queries, the average accuracy (the percentage of the relevant models) in the top-8 retrieval results of our algorithm and Yoon et al.'s are 80.4% and 76.8%, respectively. Thus, we have achieved a better performance.

To measure the retrieval accuracy of our algorithm, we adopt the performance metric of First Tier (FT), referred to Section 1.1. We test the same queries as in Fig. 5.7 with different percentages (T=20%, 10% and 5%) for candidate views selection. Table 5.1 compares their FT scores.

We can see that when we reduce the number of candidate views to be half of the default value (T=20%, 16 views), that is, 8 views, the average FT score decreases only 3.6%. Even after reducing it further to be only a quarter of the default value, that is, only 4 candidate

Table 5.1: First Tier performance comparison using different percentage T values and the thirteen query sketches in Fig. 5.7.

$T(\%)$	20	10	5
chair	70%	55%	50%
cup	85%	85%	80%
teddy	85%	80%	80%
ant	90%	85%	85%
hand	75%	70%	70%
human	45%	50%	45%
glasses	10%	10%	10%
plane	85%	75%	70%
table	75%	70%	55%
plier	80%	80%	75%
fish	65%	65%	55%
bird	45%	45%	35%
octopus	35%	35%	40%

views, the FT score drops only 9.3% averagely. This indicates the robustness of our sketch-based retrieval algorithm with respect to the number of candidate views. The relatively high FT scores also demonstrate the accuracy of our retrieval algorithm. We note that for some classes, such as human and octopus, when T becomes higher, FT may decrease somehow. Our explanation is as follows. When T is increased, more candidate views are considered to compute the sketch-model distance, that is to say, a longer sequence (e.g. 8 views when $T = 10\%$ and 16 views when $T = 20\%$) of sketch-view distances will be computed for each model. The sketch-model distance computed based on more candidate views may be smaller than that computed based on less candidate views. Therefore, when more candidate views are considered, the sketch-model distances between the sketch and some irrelevant models may become smaller and thus these irrelevant models will be pushed forward in the retrieval lists and this decreases the First Tier performance.

5.5.2 Standard Line Drawings

We perform a similar experiment as described in Section 5.5.1 using line drawing queries. We still use the same WMB database but utilize Snograss and Vanderwart's standard line drawings dataset [171] as queries. Fig. 5.9 shows several line drawings examples that have relevant classes in WMB.

Similarly, we set the percentage T for candidate views selection to be 20%. We use the sketches in Fig. 5.9 as queries and their top-20 retrieval results are shown in Fig. 5.10. Table 5.2 shows the changes of the FT performance when using different percentage T values for candidate views selection. The robustness of our sketch-based retrieval algorithm is verified again by the standard line drawing sketch queries. The decreases in the FT performance by changing T from 20% to 10% and from 20% to 5% are 3.1% and 7.1% on average.

Table 5.2: First Tier performance comparison using different percentage T values and the seven query sketches in Fig. 5.10.

T(%)	cup	bear	ant	plane	hand	table	chair
20	90%	70%	55%	70%	80%	60%	75%
10	85%	65%	40%	70%	75%	55%	75%
5	90%	55%	30%	70%	70%	55%	80%

5.5.3 Overall Performance Comparison

To assess the overall performance of our algorithm on a database level and perform a comparative evaluation with other approaches, we test our retrieval algorithm on the complete query set (250 sketches) of Yoon et al.'s [201] benchmark and compare the performance with a 2011 paper by Saavedra et al. [158]. They tested their proposed STELA approach on the same benchmark database and compared with the global shape descriptor-based approach HELO [157]. Table 5.3 and Fig. 5.11 compare the First Tier performances of our approach (T=20%) and these two methods (STELA and HELO) on each class. For the performances of STELA and HELO, we refer to [158]. The average First Tier performances over all the classes are as follows: HELO: 13.9%, STELA: 16.5%, Ours: 41.5%. Apparently, we have achieved much better results in terms of respective classes and overall performance.

Table 5.3: First Tier performance comparison between our method and STELA [158], as well as HELO [157].

Methods	HELO	STELA	Ours
chair	8.8%	12.1%	31.8%
cup	13.8%	14.2%	57.4%
teddy	21.0%	33.8%	62.9%
ant	14.7%	12.6%	64.5%
hand	33.3%	31.9%	37.4%
human	25.5%	32.1%	29.1%
glasses	2.9%	7.9%	25.6%
plane	2.1%	11.7%	40.5%
table	13.5%	12.0%	44.2%
plier	7.9%	4.5%	63.8%
fish	16.2%	15.2%	38.4%
bird	10.7%	11.0%	20.8%
octopus	10.8%	15.0%	22.9%

In addition, we want to compare our approach with the algorithm in Yoon et al. [201], in terms of the overall performance. Though we cannot find the complete overall performance data in the paper, according to our knowledge (personal communication with one of the author of the paper Yoon et al. [201]: Dr. Sang Min Yoon), the performance of Yoon et al. [201] is comparable to STELA, in terms of the overall First Tier performance as well as the First Tier performance for each class. Thus, our alignment-based retrieval approach also outperforms Yoon et al. [201].

To have a comprehensive evaluation of our algorithm, we further provide the results for other performance metrics (Section 1.1) including Precision-Recall plot, Nearest-Neighbor (NN), Second Tier (ST), E-measure (E), Discounted Cumulative Gain (DCG) and Average Precision (AP), as shown in Fig. 5.12 and Table 5.4 respectively.

Table 5.4: Other performance metrics of our algorithm on the Yoon et al.'s [201] benchmark.

NN	ST	E	DCG	AP
0.688	0.581	0.411	0.731	0.556

Similarly, we also perform the robustness experiment by changing the values of T and compare the results in Fig. 5.13. Their average First Tier performances over all the classes are as follows: T=20%: 41.5%; T=10%: 40.8%; T=5%: 38.9%. The conclusion is consistent with the previous ones, thus our retrieval algorithm is robust with respect to either respective models or classes. Though the 2D hand-drawn sketch dataset in the Yoon et al.'s [201] benchmark is composed of 2D sketches corresponding to non-rigid or articulated models in the WMB dataset (Section 1.1), we have achieved much better performances and this also demonstrates the robustness of our algorithm with respect to non-rigid or articulated sketches/models.

5.5.4 Extensibility to Larger or Other Database

To test the extensibility of our SBR algorithm to a larger database, we use the same query set as that of Yoon et al.'s benchmark and the complete 400 models in the WMB database. That is, we add 140 more models, classified into 7 classes, each 20 and regard them as noise. Example models for those 7 classes are shown in Fig. 5.14. We set T=20% and perform a similar experiment as the one in Section 5.5.3. Fig. 5.15 compares the performance with that of the experiment done in Section 5.5.3 which uses Yoon et al.'s benchmark (260 models of the WMB database). We can see even when we added more models to the 3D dataset used in Yoon et al.'s benchmark, the performance is still stable and for most classes there are only trivial decrease. The average FT performance is 38.3% and it only drops 3.2% compared to the performance achieved using Yoon et al.'s benchmark. We need to mention that the accuracy disparity in the "cup" class is due to the newly added "vase" class and some "vases" are quite similar to "cup", either in terms of the overall shape or their outlines. On the other hand, the outlines of some cup sketches are also similar to vases, which also shows a limitation of the outline feature representation. The retrieval time for the Yoon et al.'s benchmark and the above mentioned extended version is averagely 72.3 seconds and 121.8 seconds respectively when keeping 16 candidate views, which indicates a rough linear relationship between the retrieval time and the size of the benchmark. This is a limitation of the extensibility of our algorithm and we will analyze it in detail in Section 5.5.6.

We further tested our algorithm on the NIST database [36] from which we select 260 models that have relevant classes in the Yoon et al.'s sketch dataset. 13 relevant classes were selected from NIST but we combined related classes according to the available sketch categories in Yoon et al.'s sketch database, in the end we got 8 classes. We set T=20% and the First Tier performances are: human: 23.6%, cup: 78.2%, glasses: 31.8%, plane: 60.8%, chair: 57.9%, table: 46.6%, fish: 43.7%, bird: 13.3%. The average First Tier performance is 44.5%, which is comparable to the performance on Yoon et al.'s database.

5.5.5 Discussions

We found that a good pose to align a 3D model with a sketch often ranks high and for many of them it is among the top four. For example, Fig. 5.16 shows the top four candidate views for the cup, teddy and plane hand-drawn query sketches and cup, bear and plane line drawing query sketches. As seen in Fig. 5.16, in the top four candidate views for these sketches, usually we already can find certain views of the 3D models that are accurate enough, in terms of retrieval, to correspond with the sketches. We also want to mention that the orientation differences between the sketch and candidate views in the examples, such as those existing in the four candidate views for the plane line drawing query (alignment results in the third row of Fig. 5.16), are not an issue for the retrieval since we utilize the relative shape context to encompass the variations in camera up-vectors during rendering.

To find out the contribution of 2D-3D alignment, we compared the performances of using the fixed sampling method and our alignment approach based on the same number of sample/candidate views. For the fixed one, we tested with Yoon et al.'s sampling method [201]: 6 orthographic and 8 isometric views. Because of the symmetrical property in rendering our feature views as described in Section 4.2.3.1, only half of the 14 sample views, that is 3 orthographic and 4 isometric views, are selected after aligning 3D models with Continuous Principal Component Analysis (CPCA) [190] method. For our algorithm, we keep the top 7 candidate views. We test them on the Yoon et al.'s database. Table 5.5 compares their First Tier scores with respect to each class and their overall First Tier performances are as follows: Fixed: 32.6%, Ours: 39.8%, which demonstrates an apparent improvement of using the 2D-3D alignment step to shortlist several candidate views. As can be seen from Table 5.5, compared to the fixed sampling approach, our method also achieves a more balanced performance especially on the chair, hand, human, glasses, table and plier classes.

Here, we also want to mention that the relative shape context matching part is also important to achieve a better performance. The clue can be also found from the fact that without alignment, that is using the fixed sampling approach, the relative shape context matching-based retrieval algorithm already achieves a First Tier performance of 32.6%, which already surpasses HELO (around 13.9%) and STELA (around 16.5%), referred to Section 5.5.3. However, if incorporating our 2D sketch-3D model alignment step to shortlist better candidate views, we further improve the retrieval performance to 39.8%. Therefore, both the view context-based 2D sketch-3D model alignment and the relative shape context matching on the outline feature views, have important contributions to our apparently better performance than HELO and STELA.

Table 5.5: First Tier performance comparison between fixed sampling and our method.

Methods	Fixed	Ours
chair	16.3%	28.1%
cup	51.6%	55.3%
teddy	57.1%	59.5%
ant	52.4%	64.2%
hand	16.6%	34.2%
human	16.2%	28.1%
glasses	10.6%	25.6%
plane	40.5%	38.3%
table	32.1%	41.3%
plier	38.5%	60.3%
fish	46.6%	38.4%
bird	20.5%	20.0%
octopus	24.7%	24.7%

In addition, we also successfully participated in the SHREC 2012 Track: Sketch-Based 3D Shape Retrieval [128], held in conjunction with the fifth Eurographics Workshop on 3D Object Retrieval. According to the track report [97], our sketch-based 3D model retrieval algorithm, denoted as “Li(SBR-2D-3D)”, achieves the best accuracy performance and it apparently outperforms other participating approaches. Some results are shown in Fig. 5.17. While, it also shows that our algorithm is not the most efficient, as shown in Fig. 5.18.

To conclude, our 2D-3D matching considers a large number of sample views compared to previously sparse view sampling strategies, thus it is more robust to different poses of the sketches. It can efficiently find several good candidate poses of a 3D model to align the model with a sketch. The above two types of experiments on the hand-drawn sketch queries and standard line drawing queries have demonstrated the effectiveness of our retrieval algorithm, which shows better performance than Yoon et al. [201] and Saavedra et al. [157, 158]. The robustness of our retrieval algorithm is also verified in our experiments.

5.5.6 Limitations

As shown in the above experiments, our approach has a good accuracy in terms of sketch-based retrieval. Nevertheless, it has some limitations. Firstly, the performances for some sketches (e.g. glasses, octopus and bird) are not as good as others and still have room for further improvements. Secondly, relative shape context matching part dominates the most part of the retrieval time: on average, it takes 0.86 second to extract the features (Zernike moments, Fourier descriptors, eccentricity, circularity and relative shape contexts) for a sketch; only 0.37 milliseconds for the 2D-3D alignment for a model; 17.5 milliseconds for the 2D-3D matching based on relative shape context for a pair of sketch-candidate view. The average time for a complete retrieval on the Yoon et al.'s database is 19.5 seconds, 37.3 seconds and 72.3 seconds when keeping 4, 8 and 16 candidate views respectively. The retrieval time t (sec) is proportional to both the number of candidate views M and the number of the 3D models in the database, denoted by N. We denote τ as the matching time for one candidate view, then the retrieval time t (sec) can be approximately formulated as follows: t=M*N*τ. In our experiments, τ=0.0175 sec. Thirdly, our precomputation (Section 5.4.2) also takes time and needs space. For example, as stated in Section 5.5.1 on average the precomputation time of processing a model is 97 seconds: 8.8 seconds for the view context precomputation and 88.2 seconds for the relative shape context precomputation of 81 sample views for a 3D model. The spatial spaces for the precomputed view context and relative shape context data is averagely 16.6K and 2542.6K bytes, respectively. Totally, about 665M bytes are needed to save the precomputated data of the Yoon et al.'s benchmark.

According to the robustness analysis of our algorithm in Section 5.5.3 (Fig. 5.13), there is no much performance decrease when we keep fewer candidate views. Thus, we further tested our algorithm by keeping only 2 and 1 candidate view, and still got the average First Tier performances of 37.4% and 35.9% respectively, compared to the 41.5% when keeping 16 candidate views. The retrieval time is 10 seconds and 5.4 seconds respectively. Thus, our suggestion is that users can make decision for the tradeoff between the accuracy and efficiency based on the requirements of their respective applications and available resources.

5.6 Summary

In this chapter, we have presented a sketch-based 3D model retrieval algorithm based on the idea of first aligning a 3D model with a query 2D sketch before computing their matching distance. The algorithm comprises precomputation and retrieval stages. During the precomputation stage, we compute the view context and relative shape context features of a set of densely captured sample views for each target model. Based on the precomputed view context features of a target model, in the retrieval stage we can efficiently and effectively align the model with the 2D sketch. Experiments based on hand-drawn and standard line drawings sketches demonstrate the superior performance and robustness of our approach. Thus, it has a potential to be used in applications, such as sketch-based 3D model recognition and modeling, as well as 3D scene reconstruction based on 2D sketches.

Several facets of the algorithm can be further explored. First, during the retrieval stage, we can use representative relative shape context [123] to speed up the matching process between the sketch and the candidate views since we can reject the candidate views that are obviously different from the sketches earlier. In addition, if using other faster correspondence algorithms to replace our adopted LAP algorithm or adopting some 2D image descriptors which are comparable in terms of effectiveness but more computationally efficient, we may improve the retrieval performance further. Second, we want to further test our sketch-based retrieval on other types of 3D model databases and sketches. Third, extending our algorithms to other types of queries is another interesting direction. For example, query by a 2D image or even a sketch of a 3D scene comprising several objects.

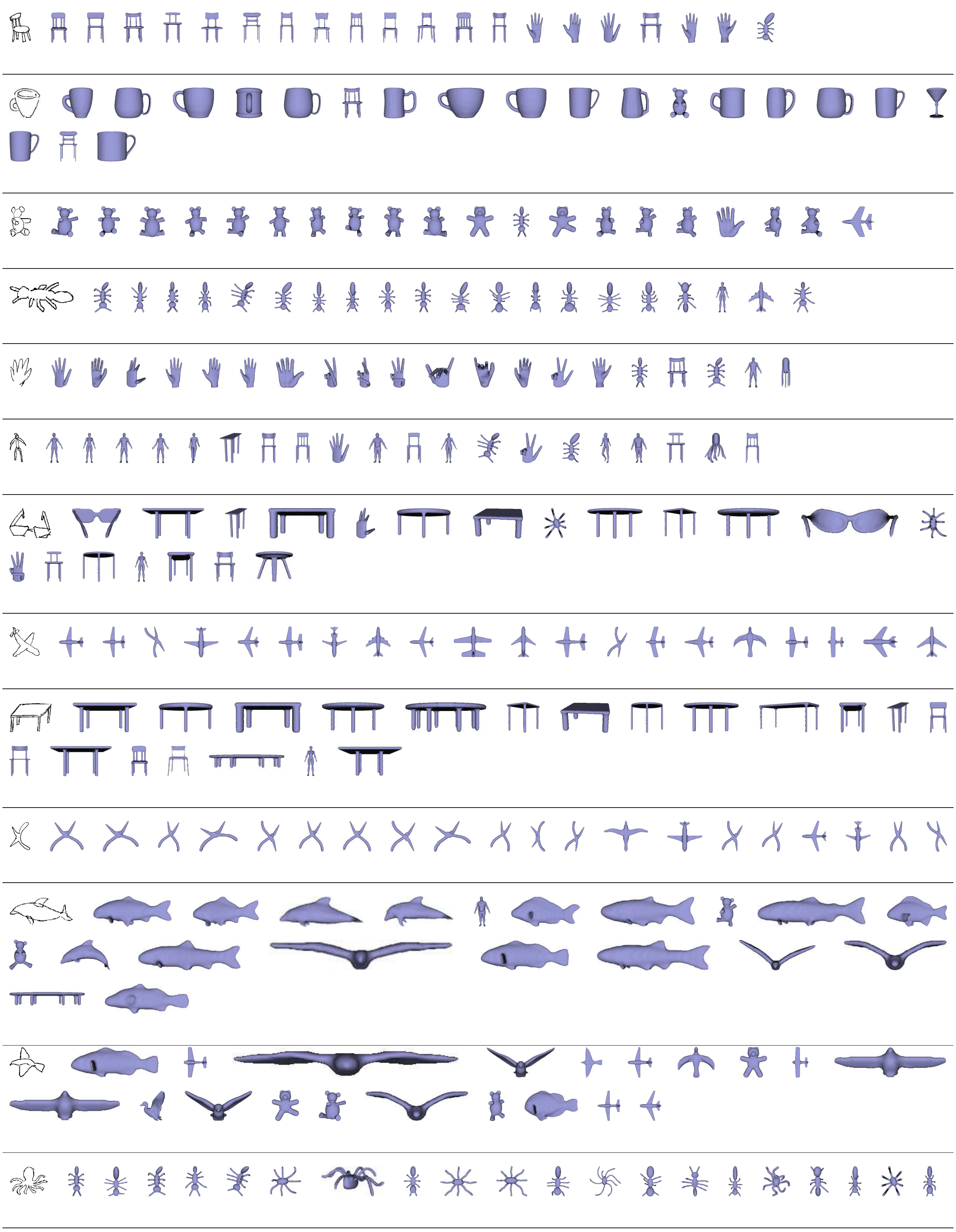


Figure 5.7: Hand-drawn sketch-based retrieval examples on WMB database using hand-drawn sketches in [201]. The first 20 models are listed.

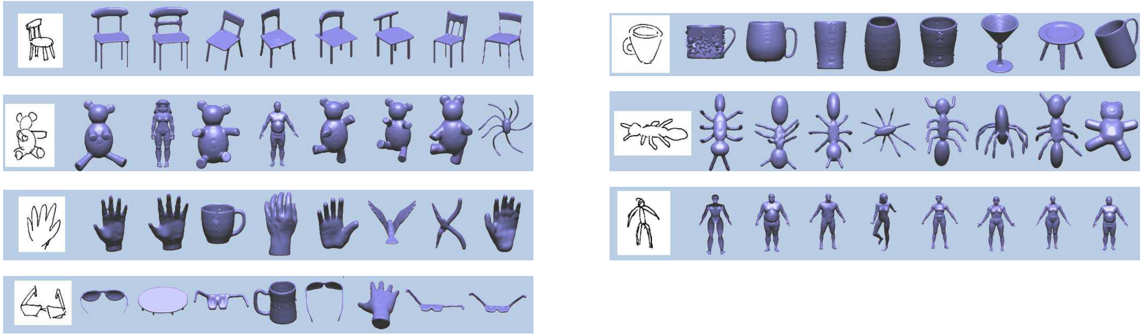


Figure 5.8: Hand-drawn sketch-based retrieval results in [201].

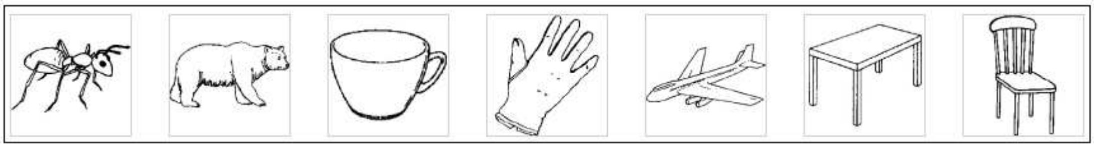


Figure 5.9: Selected typical sketches in Snograss and Vanderwart's dataset [171].

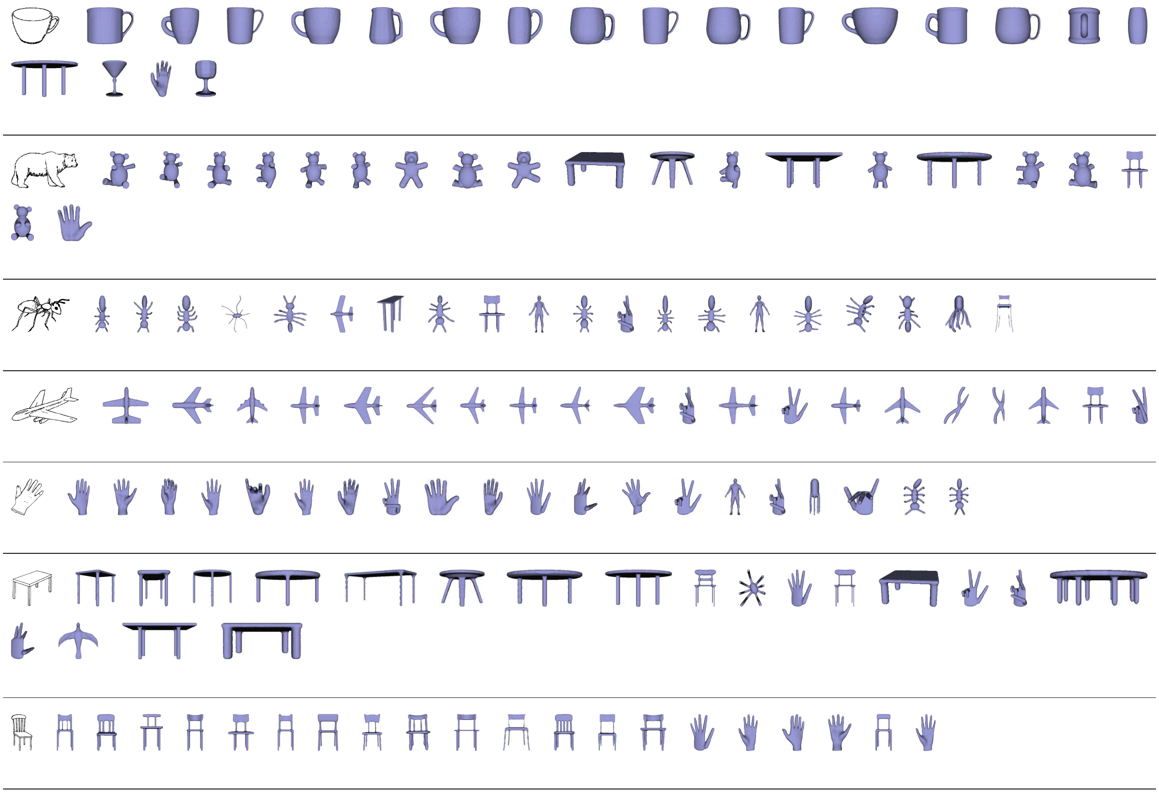


Figure 5.10: Standard sketch-based retrieval examples on WMB database using line drawings in [171]. The first 20 models are listed.

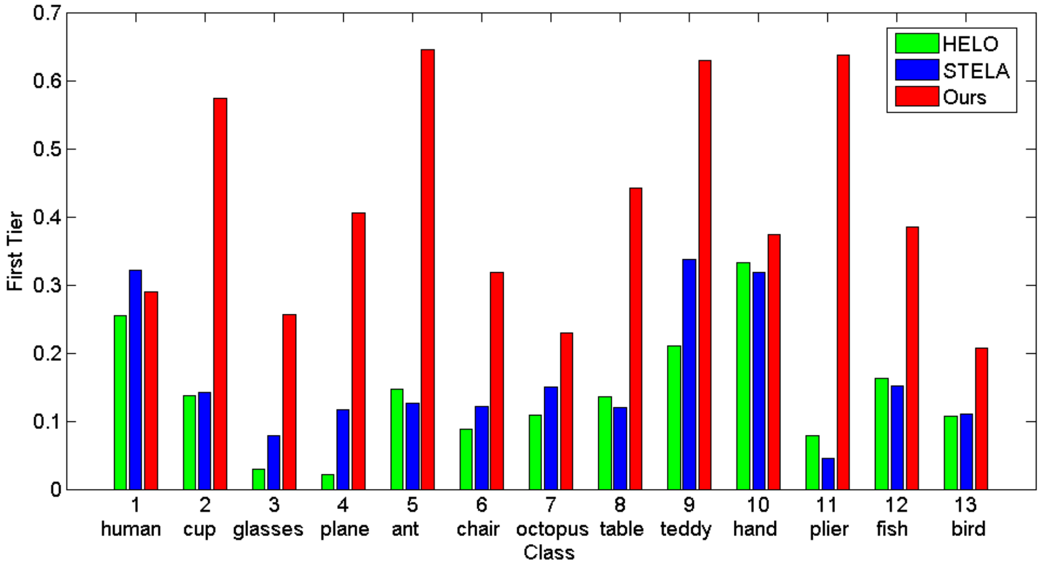


Figure 5.11: First Tier performance comparison between our method and STELA [158], as well as HELO [157].

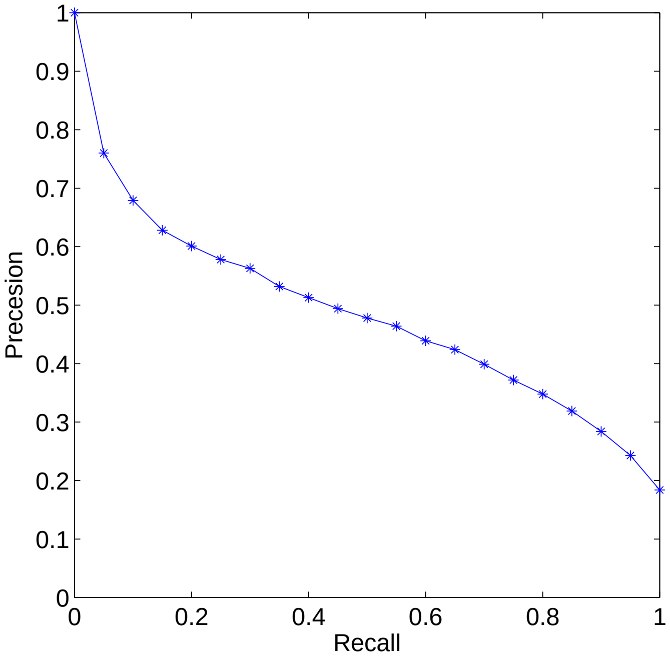


Figure 5.12: Precision-Recall performance of our algorithm on the Yoon et al.'s [201] benchmark.

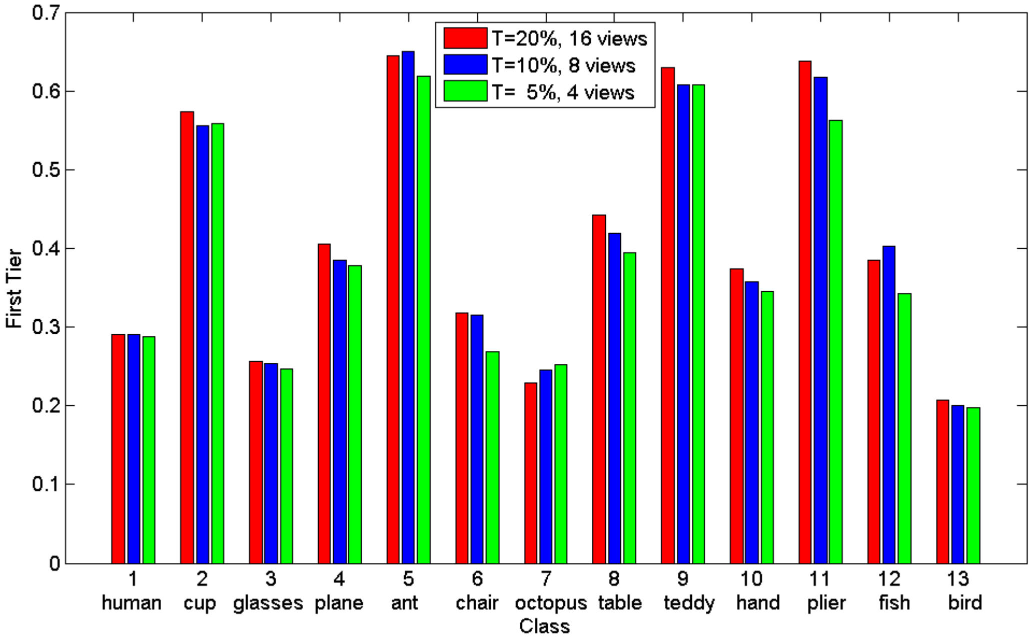


Figure 5.13: First Tier performance comparison using different percentage T values on the Yoon et al.'s [201] benchmark.

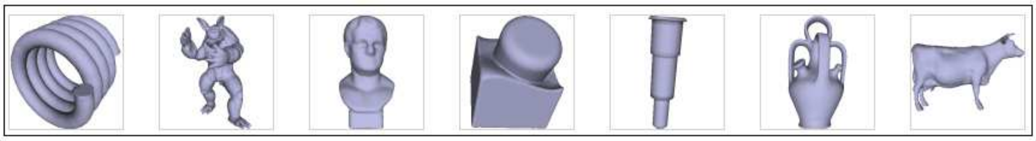


Figure 5.14: Typical 3D model for each of the added 7 classes.

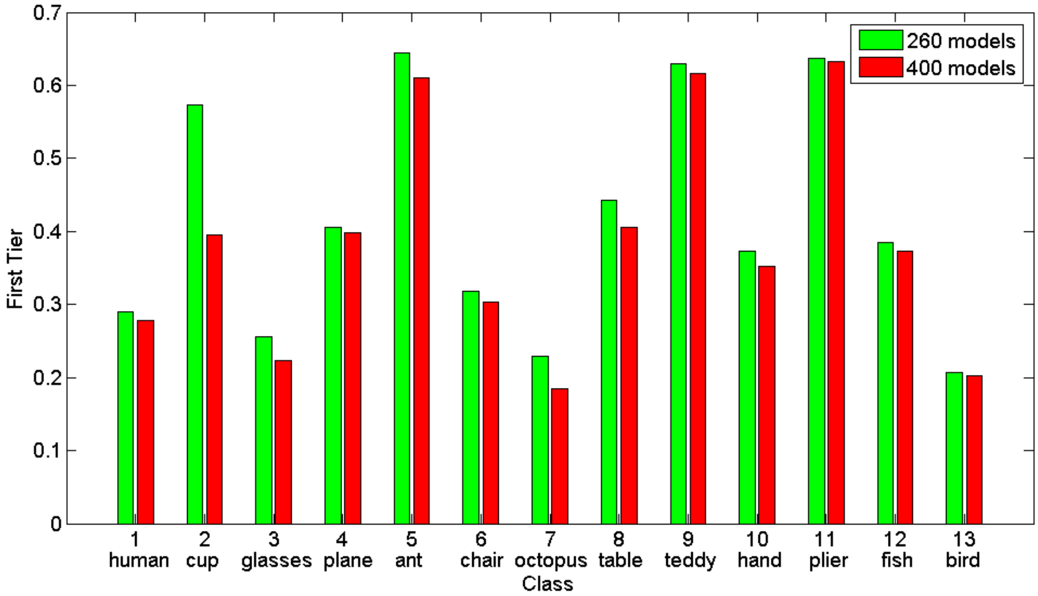


Figure 5.15: First Tier performance comparison using the whole WMB database (400 models) and only the relevant 260 models as the target 3D model database.

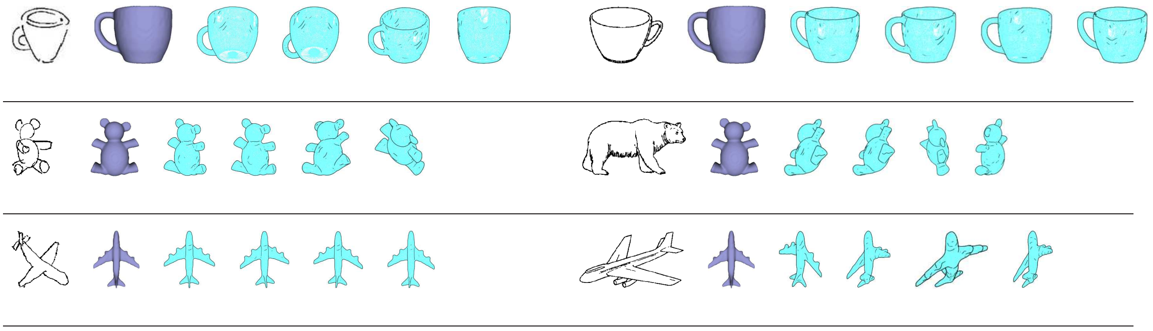
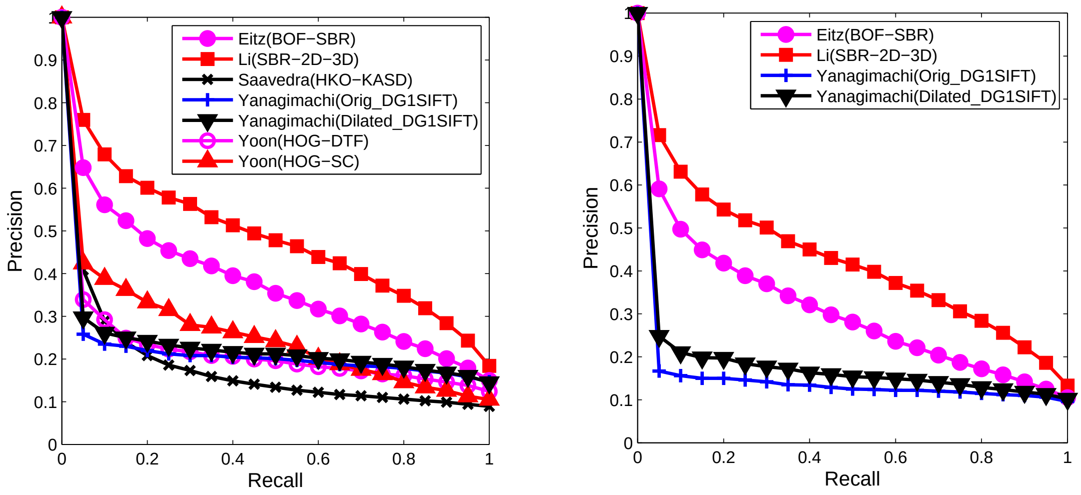


Figure 5.16: 2D-3D alignment examples. Each row shows two sets of alignment results for a hand-drawn sketch and a line drawing sketch. For each result, from left to right: a 2D sketch, a 3D model (in initial pose) and the top 4 candidate views to align the 2D sketch and the 3D model.



(a) (b)

Figure 5.17: Precision-Recall plot performance comparison results in [97]: (a) Yoon et al.'s benchmark; (b) Extended version (see Section 5.5.4) of Yoon et al.'s benchmark.

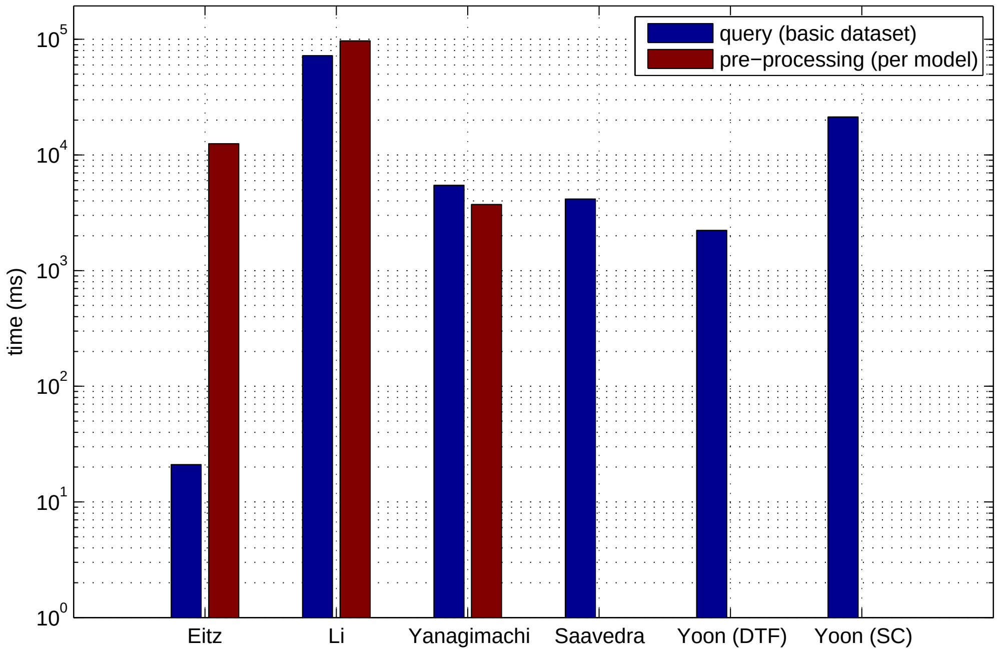


Figure 5.18: Timing information comparison results in [97] on Yoon et al.'s benchmark.

Chapter 6

Query by Utilizing Class Information and Hybrid Features

6.1 Overview

In spite of the fact that many shape descriptors and techniques have been proposed for 3D model retrieval, it is still difficult to find a shape descriptor which performs well on all types of shape benchmarks. 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, many existing professional or generic 3D model databases are already classified. Some examples include Engineering Shape Benchmark (ESB) [67], Bonn University Architecture Databases Benchmark [194], Princeton Shape Benchmark (PSB) [167], National Taiwan University Shape Benchmark (NTU) [23], CCCC [190] and Shape Retrieval Contest (SHREC) datasets [3].

In this chapter, 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 [17], 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 [23] and hybrid shape descriptor DESIRE [191]. 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 that 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 chapter is organized as follows. The hybrid shape descriptor ZFDR is presented in Section 6.2. In Section 6.3, we present the details of our class-based 3D model retrieval algorithm CBR-ZFDR. Extensive experiment results are demonstrated in Section 6.4. Section 6.5 contains the conclusions and the future work.

6.2 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. It has a property that this formation can make its components complement with each other to represent more types of models comprehensively and effectively, thus achieving a comparable retrieval performance as the currently top shape descriptors. The computation of the shape descriptor consists of two steps: first normalize the 3D model and then compute the descriptor. Fig. 6.1 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 the radius of its bounding sphere equal to 1.0. Next, we utilize Continuous Principle Component Analysis (CPCA) [190] alignment algorithm to align the 3D model. For shape descriptor computation, we present the details as follows.

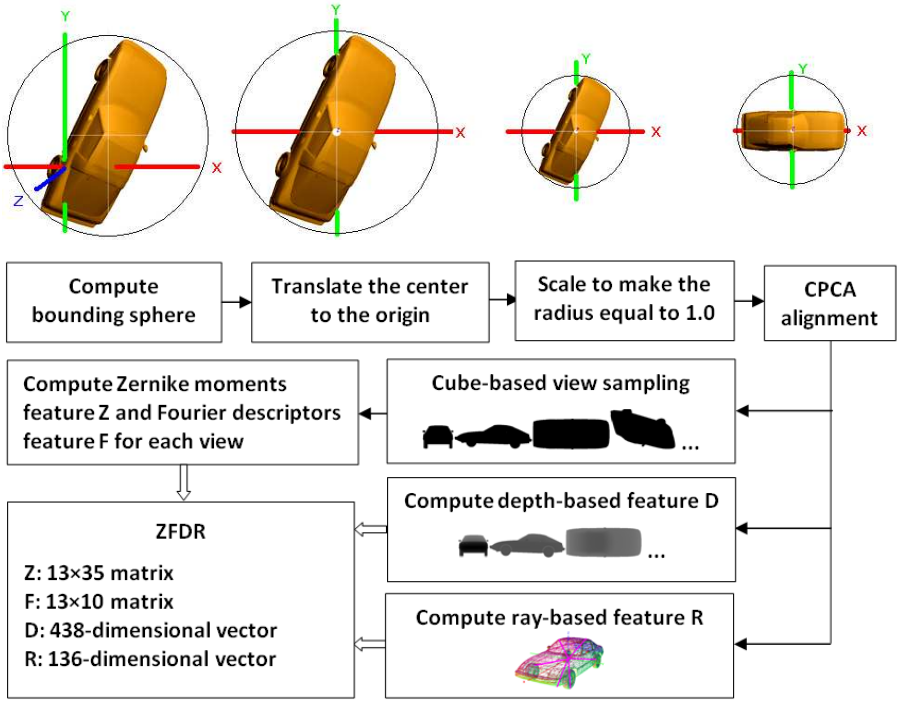
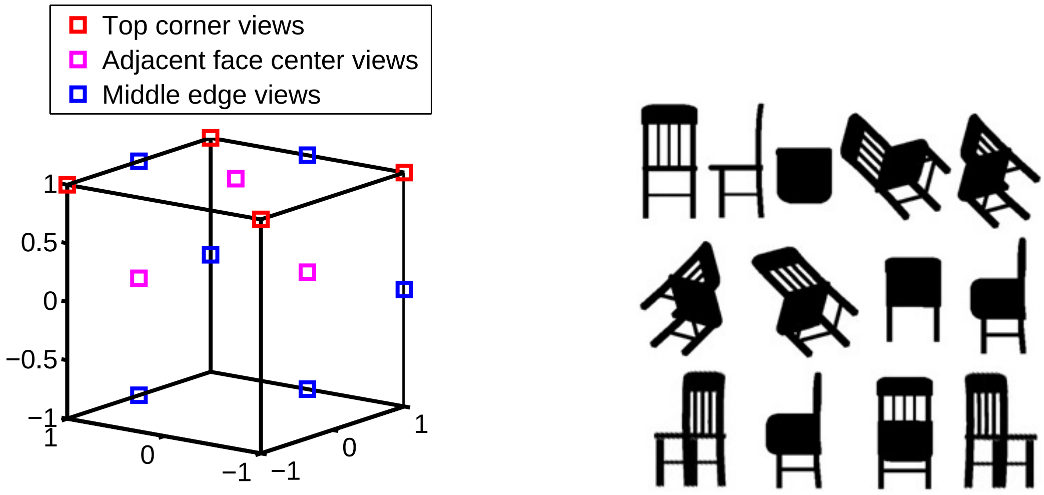

Y
X
Z
Compute bounding sphere
Translate the center to the origin
Scale to make the radius equal to 1.0
CPCA alignment
Compute Zernike moments feature Z and Fourier descriptors feature F for each view
Cube-based view sampling
...
ZFDR
Z: 13×35 matrix
F: 13×10 matrix
D: 438-dimensional vector
R: 136-dimensional vector
Compute depth-based feature D
Compute ray-based feature R

Figure 6.1: ZFDR feature extraction process.

6.2.1 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. 6.2 (a), 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. Fig. 6.2 (b) shows an example of 13 silhouette views of a chair model.



(a)

(b)

Figure 6.2: View sampling. (a) camera locations; (b) 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 [206]. It is composed of Zernike moments and Fourier descriptors.

Zernike moments feature (Z) Zernike moments depict the region-based features of a silhouette view. We compute the Zernike moments [79] (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 [205] to have better performance than other types in retrieving 2D shapes and thus we also adopt the centroid distance-based Fourier descriptor [205]. 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.

6.2.2 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 [17]. 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 [190] 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. As recommended in [190], setting the size of the view image to be 256×256, that is N=256, and the length of the Fourier coefficients will be

$6\times((\log_2 N)^2+\log_2 N+1)$. We utilize the executable file [190] to compute the depth buffer-based descriptor and use the obtained 438 Fourier coefficients 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 [78]. For this one, we also use the provided executable file [190] directly and it is also very fast. As recommended and also used in DESIRE [191], using the first k=16 rows of obtained spherical harmonic coefficients, and the dimension of the feature vector is $(k^2+k)/2 = 136$. Thus, we use the obtained 136-dimensional feature vector to depict the ray-based features.

6.2.3 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 [85], such as city block distance (L1), Euclidean distance (L2), Canberra distance [85], correlation distance, divergence distance and scaled-L1 distance [190], 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.

$$d_Z = \frac{1}{13}\sum_{p=1}^{13}\sum_{r=1}^{35}\left|\frac{Z_i(p,r)}{\left\|Z_{i,p}\right\|_1} - \frac{Z_j(p,r)}{\left\|Z_{j,p}\right\|_1}\right|, \qquad \text{(Eq. 6.1)}$$

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

$$d_F = \frac{1}{13\times 10}\sum_{p=1}^{13}\sum_{r=1}^{10}\frac{\left|F_i(p,r) - F_j(p,r)\right|}{F_i(p,r) + F_j(p,r)}, \qquad \text{(Eq. 6.2)}$$

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

$$d_D = \frac{1}{438}\sum_{p=1}^{438}\left|\frac{D_i(p)}{\|D_i\|_1} - \frac{D_j(p)}{\left\|D_j\right\|_1}\right|, \qquad \text{(Eq. 6.3)}$$

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

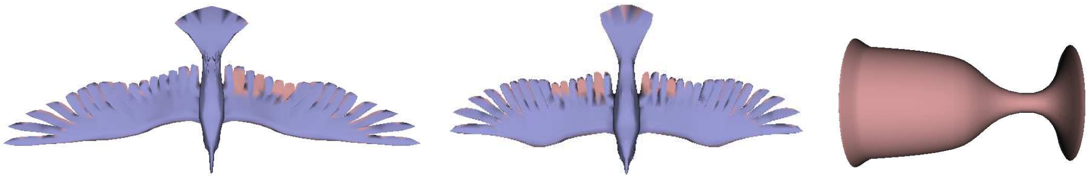
$$d_R = \frac{1}{136}\sum_{p=1}^{136}\left|\frac{R_i(p)}{\|R_i\|_1} - \frac{R_j(p)}{\left\|R_j\right\|_1}\right|, \qquad \text{(Eq. 6.4)}$$

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,

$$d_{ZFDR} = d_Z + d_F + d_D + d_R. \qquad \text{(Eq. 6.5)}$$

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. 6.3 and Table 6.1.



(a) Bird1

(b) Bird2:1.570

(c) Cup1: 3.736

Figure 6.3: An example of ZFDR distances. The number is the ZFDR distance between the model Bird1 and the respective model.

Table 6.1: Z, F, D, R component distances for the example in Fig. 6.3.

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

6.3 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 6.2. For CBR, we define an integrated distance to fuse the model and class distances. One good merit 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 in retrieval.

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 6.2.

(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 Equations (6.1)$\sim$(6.5).

(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 $d_c(q,C_j)$,

$$d_c(q,C_j) = \min_{m \in C_j} \{d(q,m)\} \qquad \text{(Eq. 6.6)}$$

Average distance is computed by averaging all the distances between query model q and the models in C_j. Centroid distance [51] 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 6.4.1), 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},

$$d_{cbr}(q, m_i) = d_c(q, C_j)^{\alpha} \times d(q, m_i), \qquad \text{(Eq. 6.7)}$$

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 6.4.1.1). 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.

Finally, please note that our class information-based retrieval algorithm CBR-ZFDR is different from the distance metric learning approach. Distance metric learning is to generate meaningful distance metrics automatically with machine learning algorithms. It learns a distance metric for the input space of data from the training data with pairwise constraints about whether they belong to the same/different classes. Thus, it employs a machine learning algorithm to generate the distance metric. However, our algorithm is not based on any learning algorithms. It incorporates the available class information of the database directly into the integrated distance and it should be categorized into class information-based retrieval approach, as reviewed in Section 2.1.4.

6.4 Experiments and Discussion

To investigate the generic 3D model retrieval performance as well as the characteristics of our retrieval algorithm CBR-ZFDR, we selected seven representative standard benchmark databases described in Section 1.1: PSB (test dataset), ESB, NTU (only use the 549 classified models), CCCC (only use the 473 classified models), MSB, NIST and WMB.

To comprehensively evaluate the generic 3D model retrieval results, we employ six metrics (Section 1.1) including Precision-Recall, Nearest Neighbor (NN), First Tier (FT), Second Tier (ST), Discounted Cumulative Gain (DCG) [167] and Average Precision (AP).

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 [187]: the target dataset is WMB (Section 1.1) and 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 Fig. 6.12 and Fig. 6.13). We use the Normalized Discounted Cumulative Gain (NDCG) [66] metric to evaluate the performance of partial retrieval results. This metric is explained in Section 6.4.3 which is dedicated for partial 3D model retrieval experiments.

6.4.1 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.

6.4.1.1 Choices of class distance and parameter α

Three different types of class distance, which are minimum, average and centroid distances, are mentioned in Section 6.3. 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. 6.4. 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

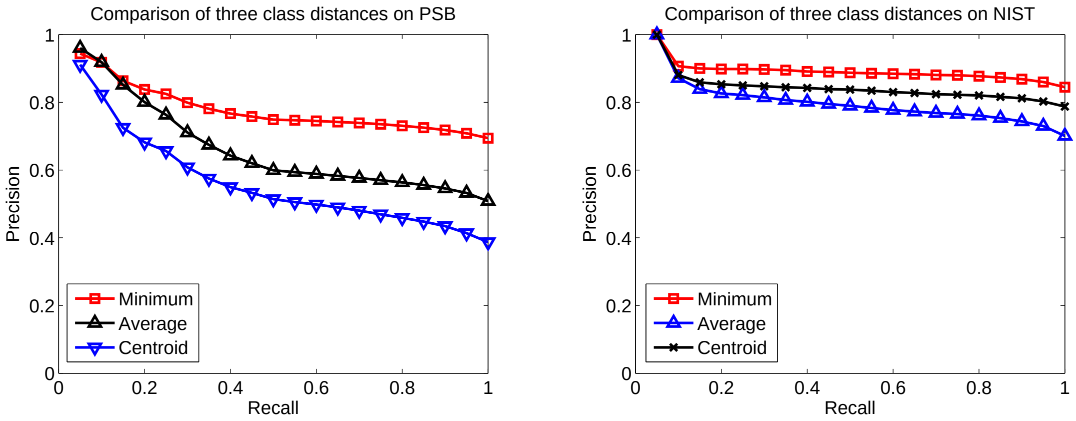


Figure 6.4: Precision-Recall plots comparison in terms of different class distance definitions. “Minimum”, “Average” and “Centroid” denote the class-based retrieval approaches using minimum, average and centroid class distances, respectively.

database by selecting five values (1,2,3,4,5) for parameter α. Similarly, we demonstrate two representative examples on the PSB and NIST databases in Fig. 6.5. We have found that bigger α will evidently improve the metrics of FT, ST, DCG and AP. However, the front part (e.g. recall$\leq$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 α =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 and DESIRE on the above mentioned seven databases.

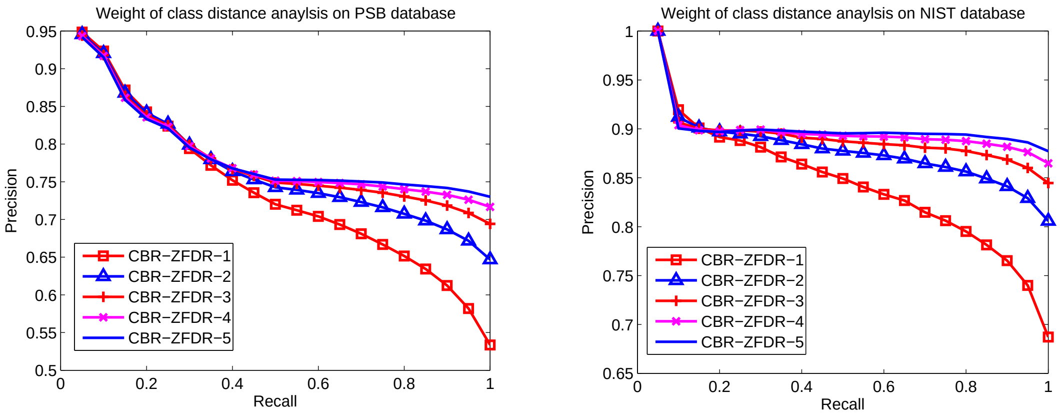


Figure 6.5: Comparative analysis of the weight of class distance on PSB and NIST databases. “CBR-ZFDR-1” means setting α to be “1” in our class-based retrieval algorithm CBR-ZFDR. Others are similar. Note that the “precision” axes do not start from “0”.

6.4.1.2 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 [191] and LF [23]. For DESIRE, we generate the results based on their provided execution files [190]. For LF, we refer to the experiment results in PSB [167] and PANORAMA [142]. Some “DCG” results are not provided in these papers and are indicated by “-”. Two representative results are shown in Fig. 6.6 and Table 6.2 and others are very similar.

As can be seen in both Fig. 6.6 and Table 6.2, 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,

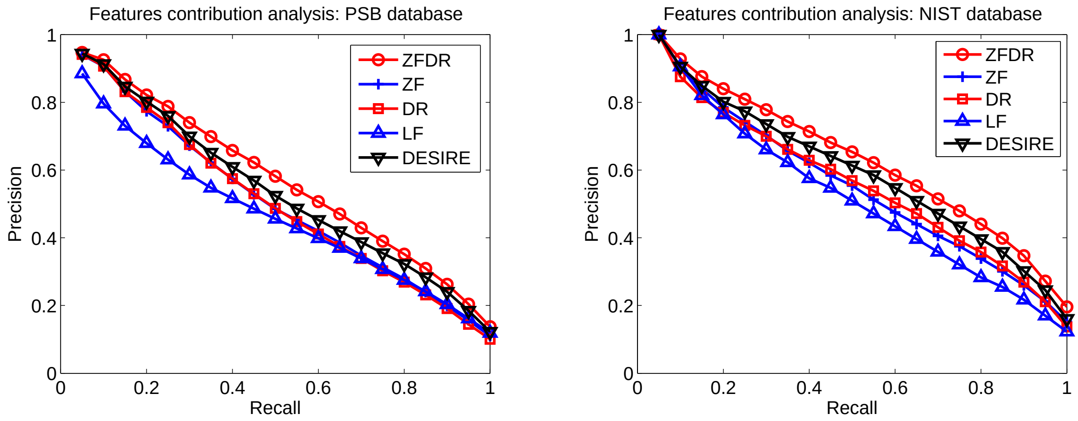


Figure 6.6: ZFDR features contribution analysis on PSB and NIST databases. ZF and DR are the two main features of the hybrid descriptor ZFDR.

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 attributed 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.

Table 6.2: 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	0.643	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

6.4.2 Generic 3D Model Retrieval

6.4.2.1 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** [141]: a 2D/3D hybrid descriptor based on 2D depth images and 3D spherical harmonics.
- **MFSD** [177]: 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 combine the spectral clustering (SC) distance and model distance to form a MFSD+SC algorithm.
- **PANORAMA** [142]: 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 seven 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. Fig. 6.7 and Table 6.3$\sim$6.4 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 refer to the experiment results in 2D-3D [141], PANORAMA [142] and PSB [167]. Some “DCG” results that are not provided in these papers are indicated by “-”.

As can be seen in Fig. 6.7 and Table 6.3$\sim$6.4, firstly, by comparing the performance of CBR-ZFDR and CBR-ZFDR-A, we can conclude that by using the scaling operation 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 achieve 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. 6.8. 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.

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.

In addition, though we do not explicitly consider the issue of robustness to non-rigid deformation and articulation when designing our ZFDR shape descriptor and CBR-ZFDR retrieval algorithm, we have achieved good performances on non-rigid or articulated 3D model benchmarks, such as MSB and WMB. As mentioned in Section 6.2.2, we target on proposing a hybrid shape descriptor which contains both visual and geometric information to effectively represent and deal with different types of models, which contributes the good performances in retrieving non-rigid or articulated models. It also shows the robustness of our algorithm with respect to the retrieval of non-rigid or articulated models.

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 Equations (6.1)$\sim$(6.7) and the matching time is proportional to the number of models. Table 6.5 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 seconds for aligning the query model using CPCA, rendering its 13 views, extracting the Zernike moments and Fourier descriptors features for all the 13 views, extracting the depth information and ray-based features, and finally computing and ranking the differences with all the shape descriptors in the database. Basically, only some small deviations in the rendering time may happen due to different number of vertices in each model. Other processes mainly remain constant or are proportional to the number of models.

6.4.2.2 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** [47]: the dataset used in the Shape Retrieval Contest (SHREC) 2009 (generic track). It was constructed based on the NIST Generic Shape Benchmark described in Section 1.1, 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.

For the SHREC 2009, we compare with the top two methods in SHREC 2009 [47], which are the composite descriptor proposed by Lian et al. [101] ("Composite") and the multi-view depth line approach ("MDLA") proposed by Chaouch and Verroust-Blondet [21]. 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. Fig. 6.9 and Table 6.6 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.

6.4.2.3 SHREC 2012 Generic 3D Benchmark

Based on the ZFDR shape descriptor, we also successfully participated in the SHREC 2012 Track: Generic 3D Shape Retrieval [127], held in conjunction with the fifth Eurographics Workshop on 3D Object Retrieval. It is based on the SHREC 2012 Generic 3D Benchmark which contains 1200 models, evenly divided into 60 classes. According to the track report [94], ZFDR ranks in the third place. Here, we further test our CBR-ZFDR algorithm on the SHREC 2012 Generic 3D Benchmark and the performance comparisons with the participating methods of the SHREC 2012 Generic Track are shown in Fig. 6.10 and Table 6.7. Similarly, we find that our CBR approach has evidently improved the performance of ZFDR, and make CBR-ZFDR achieve the best overall performance.

6.4.3 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 [187] (Section 6.4).

The goal is to retrieve similar subparts. To evaluate the partial similarity retrieval performance, we adopt the average Normalized Discounted Cumulative Gain (NDCG) [66] 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 Fig. 6.12 and Fig. 6.13) is composed of several parts cut from models of different classes, the ground truth [187] 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. 6.12 is relevant to “four legs” and “human” classes and marginally relevant to “armadillo” 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 [187] and [66].

We compare with four previous partial retrieval algorithms: RPU [179], BoF [180], ERG [14] and CORNEA [26]. ERG and CORNEA are the only two participants in SHREC 2007 partial retrieval track [187] while the latest RPU and BoF algorithms outperform ERG and CORNEA. Fig. 6.11 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.

Fig. 6.12 and Fig. 6.13 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 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.

6.4.4 Generality of Our CBR Approach

Our CBR approach is general and can be used with any shape descriptors. In Section 6.4.2.2, we have demonstrated this using the DESIRE shape descriptor on the PSB Test vs Train datasets (Fig. 6.9 (b)). We then further test CBR with DESIRE on other datasets including NIST, NTU and MSB, and the results are shown in Fig. 6.14, from which we can draw a similar conclusion as that in Section 6.4.2.2. That is, our CBR approach is general and an evident improvement can be achieved after employing CBR.

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 [142] on four representative benchmarks: NIST, NTU, ESB and MSB. Fig. 6.15 shows the performance comparison with CBR-ZFDR, as well as PANORAMA together with the local relevance feedback (LRF) technique, that is PANORAMA+LRF. Apparently, the results show that CBR-PANORAMA apparently outperforms PANORAMA and it is also superior to PANORAMA+LRF. With a better shape descriptor PANORAMA applied to our CBR algorithm, we achieve even better retrieval performance. 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 algorithm.

6.4.5 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.

6.5 Summary

In this chapter, 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 seven 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 [187], but also outperforms the two latest shape descriptors RPU [179] and BoF [180].

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.

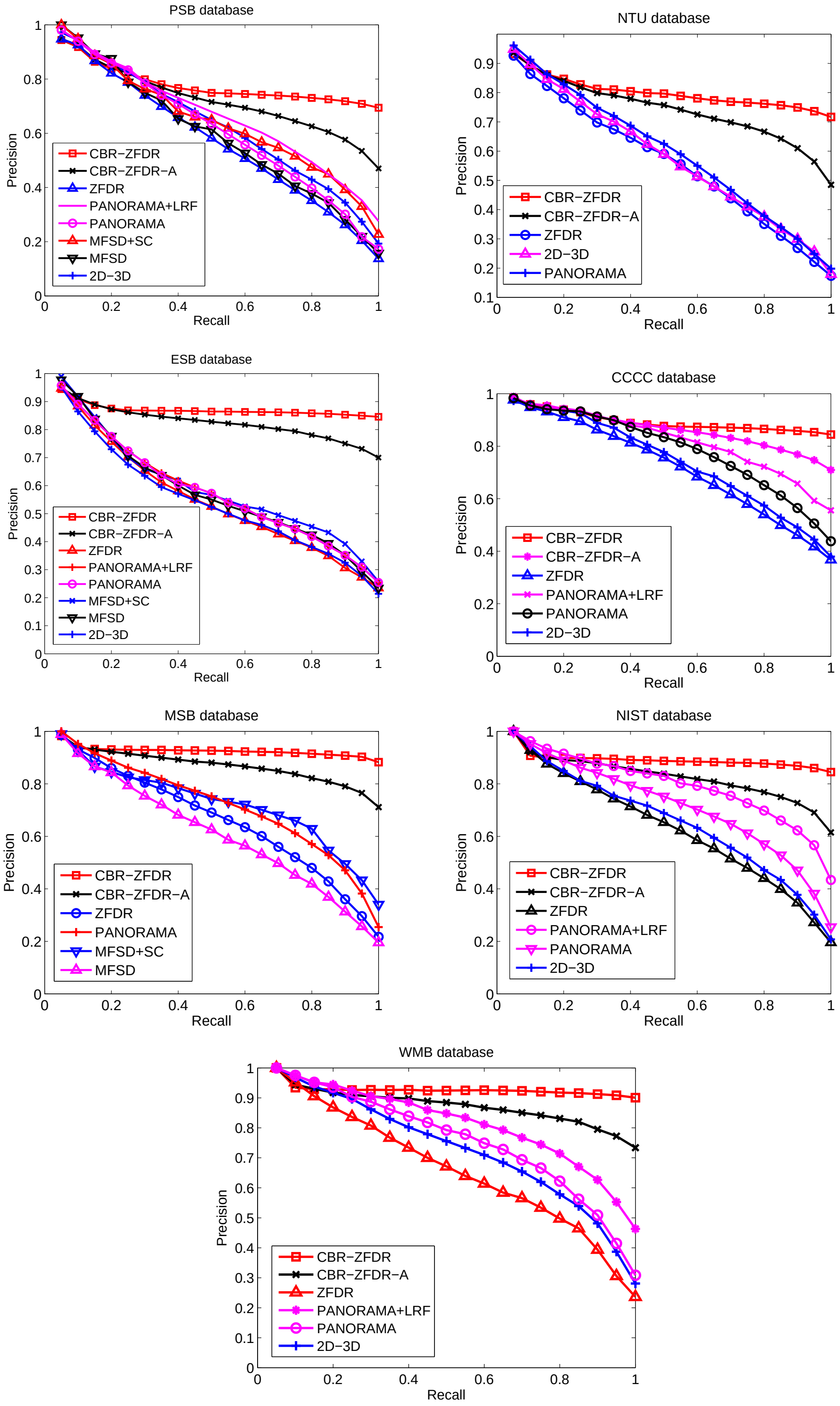


Figure 6.7: Performance comparison: Precision-Recall plots of our retrieval algorithm CBR-ZFDR and three state-of-the-art shape descriptors. “CBR-ZFDR” denotes our retrieval algorithm that utilizes the CBR algorithm described in Section 6.3 and the ZFDR shape descriptor presented in Section 6.2. “CBR-ZFDR-A” 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 6.3: Other performance metrics of our CBR-ZFDR algorithm and three state-of-the-art shape descriptors: PSB, NTU, ESB and CCCC databases. “CBR-ZFDR” denotes our retrieval algorithm that utilizes the CBR algorithm described in Section 6.3 and the ZFDR shape descriptor presented in Section 6.2. “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.

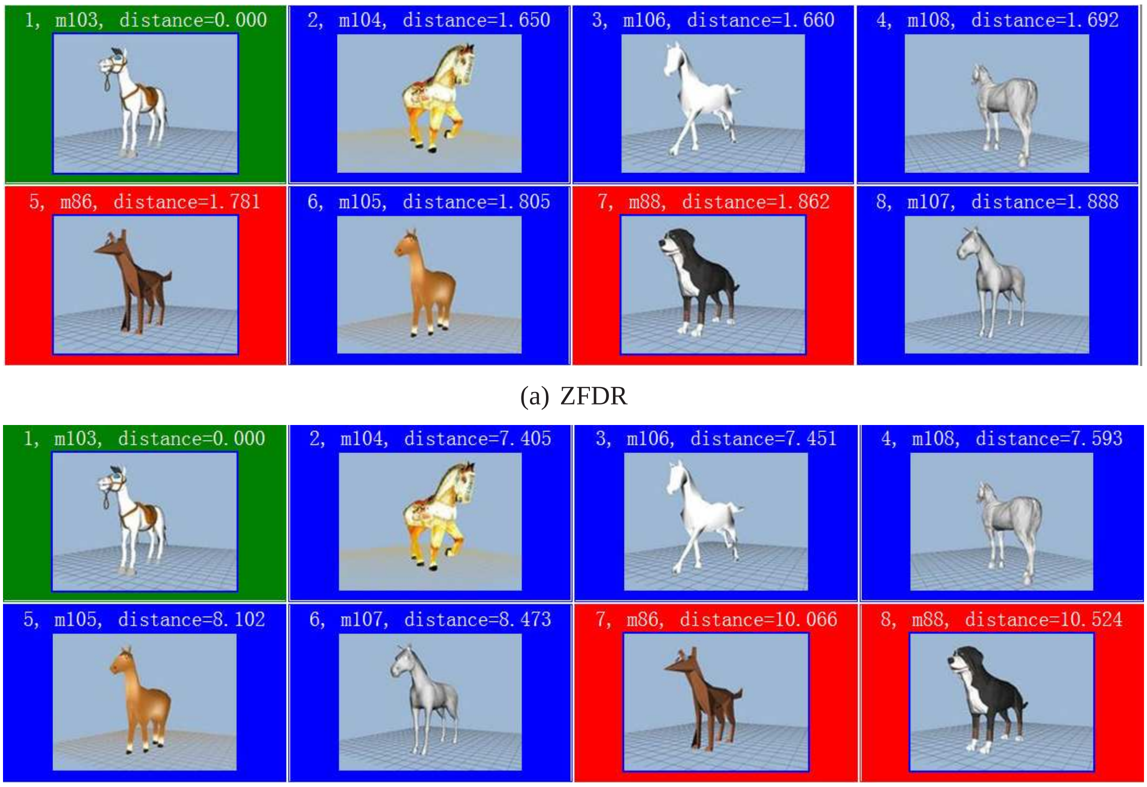
Methods	NN	FT	ST	DCG	AP
PSB					
CBR-ZFDR	69.8	**69.5**	**77.0**	**0.801**	**79.2**
CBR-ZFDR-A	69.8	60.8	71.3	0.773	74.5
ZFDR	69.8	43.9	54.9	0.691	60.6
PANORAMA+LRF	**75.2**	53.1	65.9	-	69.4
PANORAMA	**75.3**	47.9	60.3	-	64.5
MFSD+SC	71.1	50.9	63.1	0.723	67.4
MFSD	71.6	45.3	59.1	0.704	62.6
2D-3D	74.2	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

Table 6.4: Other performance metrics of our CBR-ZFDR algorithm and three state-of-the-art shape descriptors: MSB, NIST and WMB databases. “CBR-ZFDR” denotes our retrieval algorithm that utilizes the CBR algorithm described in Section 6.3 and the ZFDR shape descriptor presented in Section 6.2. “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.

Methods	NN	FT	ST	DCG	AP
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
WMB					
CBR-ZFDR	92.3	**90.2**	**95.1**	**0.951**	**93.1**
CBR-ZFDR-A	92.3	82.5	90.6	0.932	88.5
ZFDR	92.3	57.4	69.5	0.842	69.2
PANORAMA+LRF	**95.7**	74.3	83.9	-	83.5
PANORAMA	**95.7**	67.3	78.4	-	78.4
2D-3D	95.5	64.2	77.3	-	75.7

Table 6.5: Timings information of CBR-ZFDR on different databases. 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.

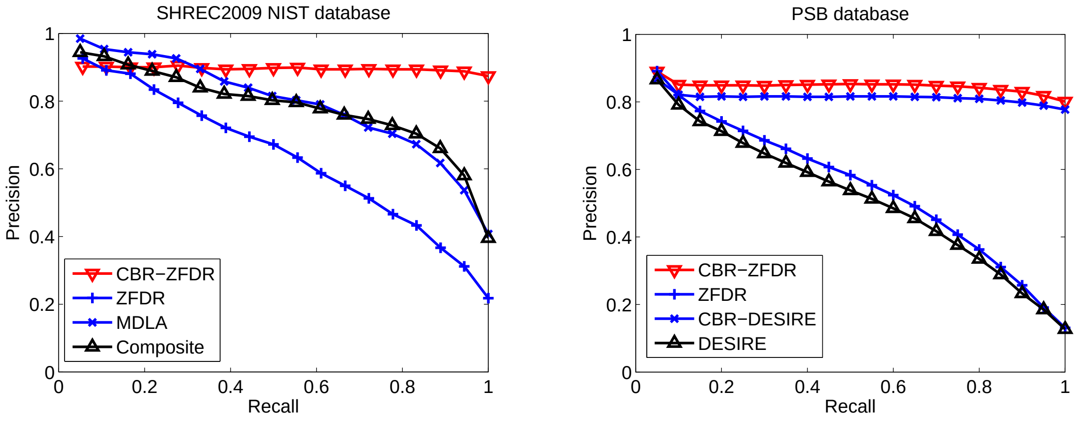
Time	PSB	NTU	ESB	CCCC	MSB	NIST
t_f (s)	1.12	2.07	1.28	1.41	1.53	1.97
t_m (s)	0.11	0.04	0.09	0.05	0.05	0.09
t (s)	1.23	2.11	1.37	1.46	1.58	2.06



(a) ZFDR

(b) CBR-ZFDR

Figure 6.8: 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.



(a) SHREC 2009

(b) PSB Test vs Train

Figure 6.9: Performance comparison: Precision-Recall plots of our retrieval algorithm CBR-ZFDR and other methods on SHREC 2009 NIST and PSB databases.

Table 6.6: Other 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

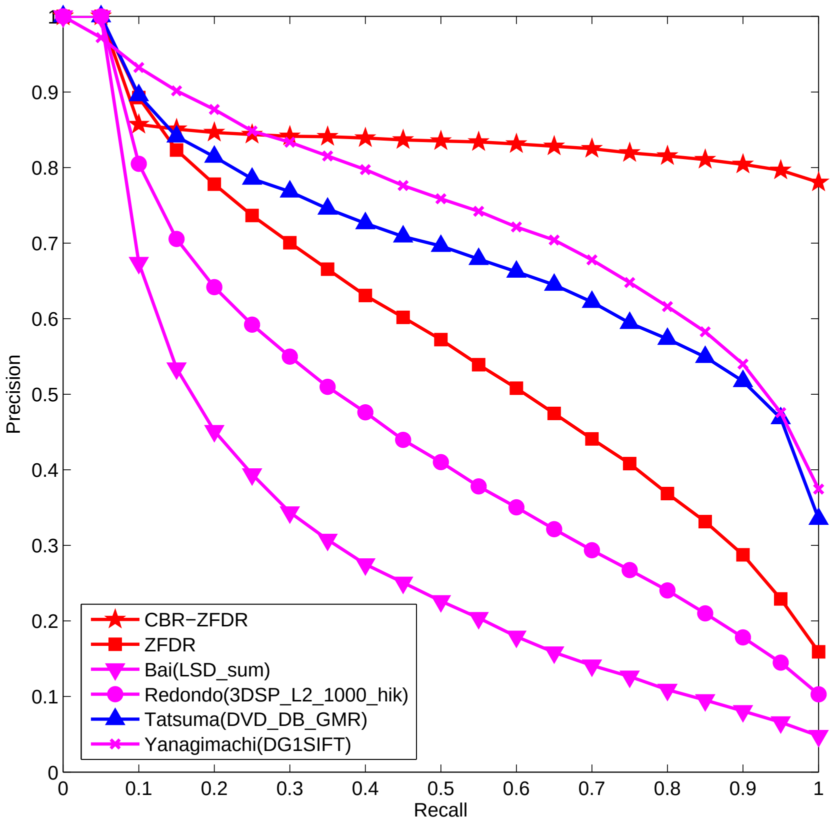


Figure 6.10: Performance comparison on the SHREC'12 Generic 3D Benchmark: Precision-Recall plots of our retrieval algorithm CBR-ZFDR and the participating methods of the SHREC 2012 Generic Track.

Table 6.7: Other performance metrics for the performance comparison on the SHREC'12 Generic 3D Benchmark.

Participant	Methods	NN	FT	ST	DCG	AP
Li	CBR-ZFDR	81.8	**79.2**	**88.1**	**0.894**	**89.8**
Li	ZFDR	81.8	49.1	62.1	0.776	65.0
Bai	LSD-sum	51.7	23.2	32.7	0.565	38.1
Redondo	3DSP_L2_1000_hik	68.5	37.6	50.2	0.685	52.6
Tatsuma	DVD+DB+GMR	82.8	61.3	73.9	0.833	76.5
Yanagimachi	DG1SIFT	**87.9**	66.1	79.9	0.871	81.1

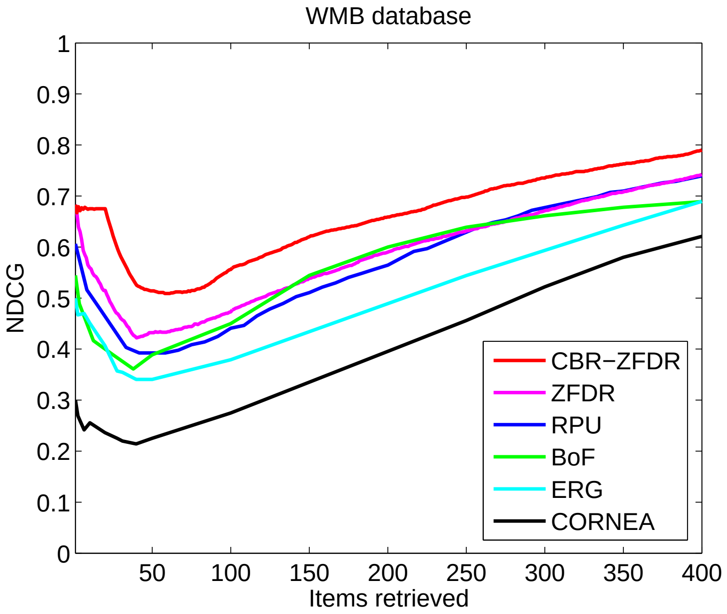


Figure 6.11: Performance comparison: NDCG plots of our retrieval algorithm and other methods on SHREC 2007 Watertight database.

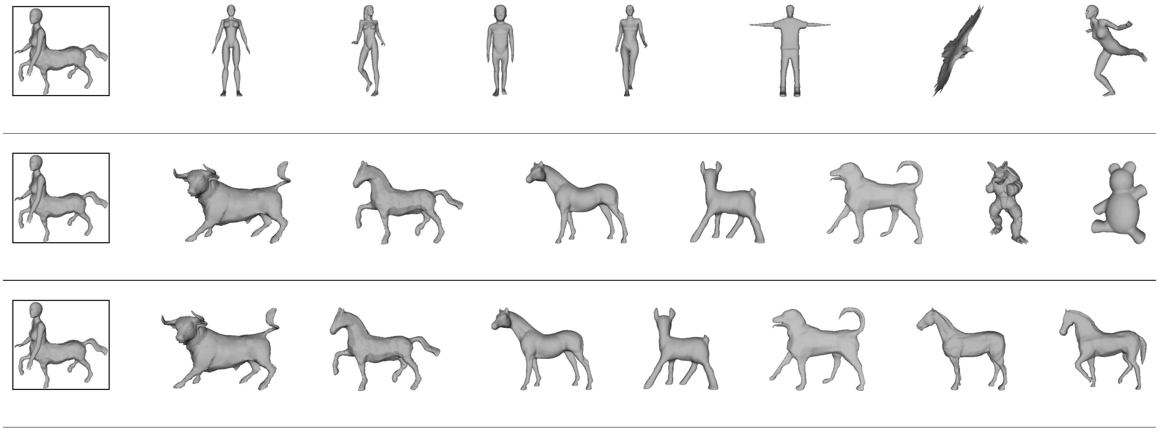


Figure 6.12: A partial matching example showing the top-7 retrieval results using RPU (1st row), ZFDR (2nd row) and CBR-ZFDR (3rd row) methods. The first model in each row is the query model.

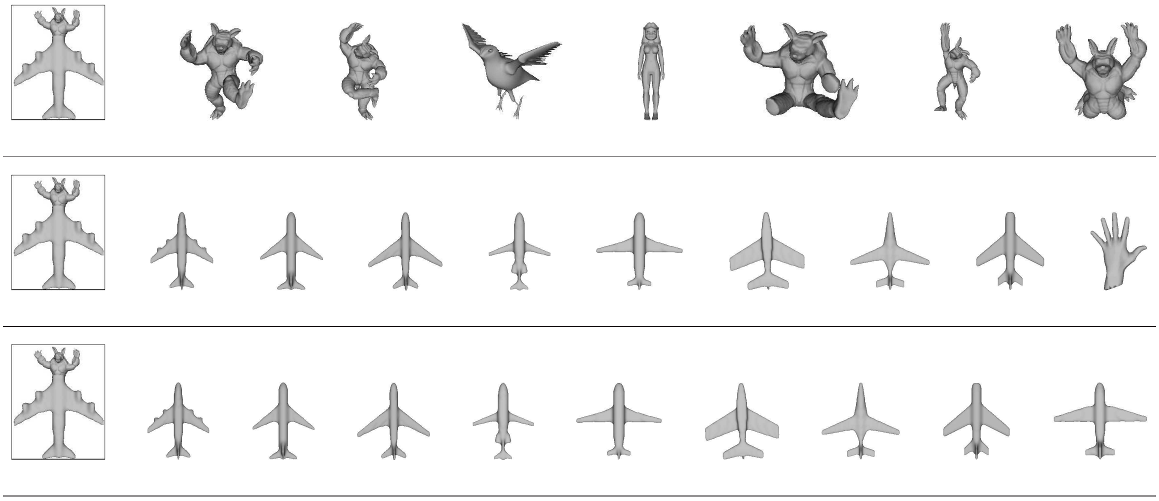


Figure 6.13: Another partial matching example showing the top-7 retrieval results using RPU (1st row) method and the top-9 retrieval results using ZFDR (2nd row) and CBR-ZFDR (3rd row) methods. The first model in each row is the query model.

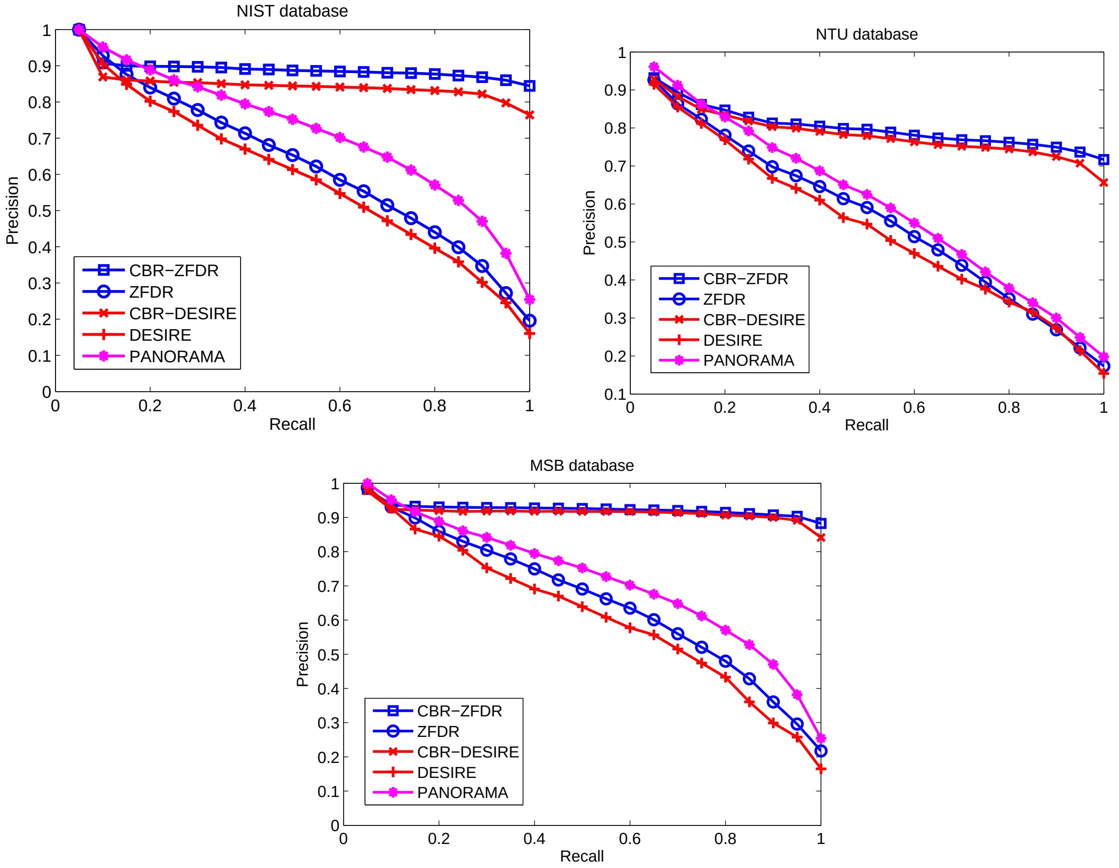


Figure 6.14: CBR generality based on DESIRE: Precision-Recall plots of our CBR algorithm with different shape descriptors on NIST, NTU and MSB databases.

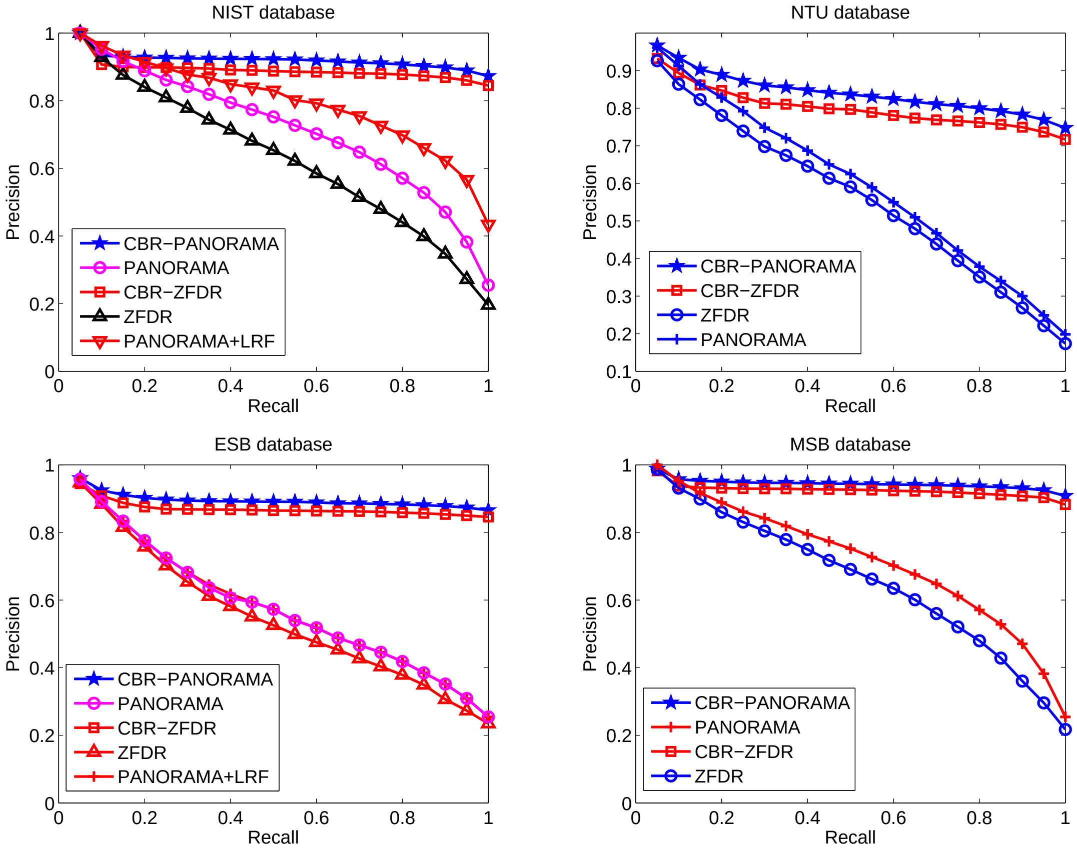


Figure 6.15: CBR generality: Precision-Recall plots of our CBR algorithm with different shape descriptors on NIST, NTU, ESB and MSB databases.

Chapter 7

Conclusions and Future Work

In this chapter, we draw a conclusion and propose several future work. For the conclusions, we briefly present the main idea, results and contributions of our proposed algorithms. For the future work, we propose two new research directions for sketch-based 3D model retrieval and partial similarity 3D model retrieval.

7.1 Conclusions

The objective of our project is to provide solutions for the challenges existing in 3D model retrieval techniques, thus to advance the research field itself as well as the related industrial, academic and entertainment applications. Provided with either a sketch, a model or an image, our goal is to effectively and efficiently retrieve a set of relevant models from a 3D model database. The retrieval process is generally comprised of several stages which are 3D or 2D-3D alignment, 2D/3D feature extraction, feature distance computation and ranking. Now, we conclude the work that has been done for each component as follows.

(1) 2D/3D feature extraction. Firstly, we defined a new 3D model feature view context which supports multi-modal retrieval framework. Based on the idea of view context, we proposed a view context shape descriptor in Chapter 4 for Query-by-Model retrieval and extended it further in Chapter 5 for Query-by-Sketch applications. In detail,

- *A new view-based 3D feature view context and a view context-based 3D model retrieval algorithm were proposed.* Experiment results show that the view context shape descriptor is comparable with the related view-based descriptor Light Field in retrieval performance and a combined shape descriptor based on view context outperforms Light Field.
- *We proposed a sketch-based retrieval algorithm by incorporating a 2D-3D alignment step based on our view context feature.* The basic idea is to perform 2D sketch-3D model alignment before computing their distances. View context has a good performance in distinguishing different views of the same model, thus based on this property we can efficiently select several candidate views to align a 3D model with a 2D sketch. Comparative and evaluative experiments based on hand-drawn and standard line drawing sketches demonstrate the effectiveness and robustness of our approach and it significantly outperforms several latest sketch-based retrieval algorithms.

Secondly, considering different characteristics and differentiation abilities of view-based and geometry-based features, in Chapter 6 we proposed a hybrid feature ZFDR. It integrates both visual and geometric features to make them complement with each other, thus achieves a better performance than its components and also outperforms two related shape descriptors Light Field and DESIRE. Its performance is also close to the state-of-the-art shape descriptors on several databases.

(2) 3D or 2D-3D alignment. We proposed a Minimum Projection Area-based (MPA) approach in Chapter 3 for 3D model alignment, as well as a 2D sketch-3D model alignment algorithm in Chapter 5 based on the proposed feature view context. MPA is used for retrieval using 3D model queries (Chapter 4). As described above, the 2D-3D alignment algorithm is specially designed for retrieval using sketch queries (Chapter 5). It utilizes the differentiation power of view context in distinguishing different views of the same model. The details of our MPA algorithm are as follows.

- *We proposed a novel Minimum Projection Area-based (MPA) method for 3D model pose normalization.* It finds three orthogonal axes by searching three view directions with minimum projection area based on an efficient and effective modified PSO algorithm. To deal with retrieval scenario, we further performed axis ordering and orientation alignment. Experimental results demonstrate that MPA has a good performance in finding alignment axes which are parallel to the ideal canonical coordinate frame of models and aligning similar models in similar poses under different conditions such as model variations, noise and initial poses. In addition, it achieves a better 3D model retrieval performance than several commonly used approaches such as CPCA, NPCA and PCA.

(3) Feature distance computation. To utilize the class information available in the target database, a new retrieval algorithm based on the class-based retrieval (CBR) approach and the proposed ZFDR hybrid shape descriptor was advised for query on classified databases. We proposed an integrated 3D model distance which scales the model distance using the corresponding class distance. Extensive generic and partial 3D model retrieval experiments on seven standard databases demonstrate that CBR apparently improves the retrieval performance of ZFDR and our class-based retrieval algorithm CBR-ZFDR outperforms the top shape descriptor PANORAMA on each database in terms of most of the commonly used performance metrics. Our CBR approach is general and can be with any shape descriptors to apparently improve their performances. We also demonstrated its apparently better performance than the usually adopted additive distance.

In a word, we have done substantial research in several aspects of 3D model retrieval techniques, proposed our solutions by mainly adopting a view-based framework, and particularly contributed in the following four important aspects: 3D model alignment, shape descriptor supporting multi-modal queries, sketch-based 3D model retrieval and query by taking into account the class information of a 3D model database. In detail,

- We make substantial contribution to the research of 3D model alignment, an important aspect of 3D model retrieval techniques, by proposing the MPA algorithm which outperforms existing alignment approaches in several facets such as axis accuracy, robustness, and retrieval performance improvement. The algorithm is also easy to understand and implement as well.

- We develop a 3D shape descriptor supporting multi-modal queries, a challenging research direction of 3D model retrieval research, by defining the view context shape descriptor. We also advance the research of sketch-based 3D model retrieval by first proposing incorporating a 2D sketch-3D model alignment step into sketch-based retrieval algorithm to increase the accuracy of 2D-3D matching. The effectiveness, robustness and significantly superior performance of our approach have been comprehensively demonstrated via several diverse evaluation experiments. Our view context-based 2D sketch-3D model improves the retrieval performance in a nontrivial manner and it is also general and can be with other sketch-based retrieval algorithms to improve their performance.

- We also conduct solid research in the research direction of retrieval on a classified 3D model database. Firstly, our proposed hybrid shape descriptor performs comparably or better than several related shape descriptors. Secondly, we are the first to perform the research of utilizing the class information available in the already classified 3D model database, by developing a general class-based retrieval (CBR) approach to obviously improve the retrieval performance for any shape descriptors.

In addition, there is also possibility of combining two or more of our proposed algorithms. For instance, we can apply the general class-based retrieval (CBR) approach proposed in Chapter 6 to the sketch-based retrieval (SBR) algorithm presented in Chapter 5. On the other hand, in order to deal with the fourth challenge mentioned in Section 1.1, we may also integrate all our proposed four algorithms (MPA, View Context, SBR, CBR-ZFDR) into a search platform providing universal functions, for example, automatically selecting alignment algorithms, 3D model features, retrieval methods and rules.

7.2 Future Work

In this section, we propose two promising directions for sketch-based retrieval and partial similarity retrieval.

7.2.1 Scene Sketch-Based 3D Model Retrieval

Sketch-based 3D model retrieval in the context of a 2D sketch image of a scene, such as a 2D storyboard, is very important for the 3D scene reconstruction from a 2D sketch, which is usually a fundamental step for 3D animation production guided by 2D storyboards [174]. This is mainly because sketches are more human-friendly and people are more accustomed to "sketch" their ideas using a set of sketches. In 3D animation production, 2D storyboards are often first drawn before reconstruction of the corresponding 3D scene. Proper 3D models are retrieved from available 3D databases to build a 3D scene [174] while keeping the context information in the original 2D scene consistent.

The reconstruction process often comprises three main steps [156, 174]: 3D object detection in the 2D scene sketch, sketch-based 3D model retrieval based on the extracted objects and 3D scene building using the retrieved 3D models guided by the original 2D scene sketch. Rather than having a 3D scene available as the input, we only draw a 2D sketch image to represent the scene, thus a new sketch-based 3D model retrieval framework which differs from previous sketch-based 3D model retrieval systems is needed. It involves multiple objects in an input scene sketch.

Currently, there is a lot of research in sketch-based 3D model retrieval. However, they usually target the problem of retrieving a list of candidate models using a single sketch as input. Therefore, the retrieval is ideally assumed as single sketch for single object, rather than in the context of a 2D scene sketch which contains several objects, which may overlap with each other and thus be occluded and also have relative location configurations. Therefore, sketch-based 3D model retrieval in the context of a 2D scene sketch deserves our further exploration.

Compared to sketch-based retrieval in the context of a single sketch, sketch-based retrieval using a 2D scene sketch query is much less studied. Fisher and Hanrahan [39] proposed a novel 3D model retrieval scheme named context-based 3D model retrieval, which means retrieving models according to its spatial context in a 3D scene. They adopted a new pipeline of model retrieval by first locating the position of the model by drawing a 3D box and then searching relevant 3D models based on the dimensionality and context information. The models in the scenes are extracted beforehand and both geometry and tags are utilized to find similar models based on the assumption that similar models appear in similar contexts. However, our research topic differs from theirs in several facets. First, their input is already a 3D scene consisting of several models. Second, it is mainly for scene completion rather than scene reconstruction. Finally, they do not draw sketches to represent a 3D model.

According to our knowledge, few algorithms have been proposed to specially deal with sketch-based 3D model retrieval in a 2D scene sketch. For example, Zernike moments descriptor [79] is a typical and important 2D feature and Ta et al. [174] developed a patch-based Zernike moments descriptor motivated by and based on the improved Zernike moments comparator [156]. It integrates not only the difference between the extracted features themselves but also their local relationship, such as relative distance and rotation angle. Similar experiments as [156] demonstrate that this local shape descriptor is more robust to occlusion than its initial global version [156] and thus achieves better performance in 3D model retrieval as well as viewpoint selection applications.

We plan to modify our sketch-based retrieval algorithm proposed in Chapter 5 to deal with scene sketch-based retrieval and reconstruction. Basically, we incorporate an object detection module to facilitate retrieving the candidate models for reconstruction. The contextual information existing in the scene sketch is utilized to select appropriate models and align the features of the models to the sketch features in the scene.

7.2.2 Partial Similarity Retrieval of Deformable Models

Compared to generic 3D model retrieval, partial similarity 3D model retrieval is more difficult and much less studied. Our target is to propose a 3D shape descriptor that can be used for both global and partial similarity retrieval, especially for non-rigid 3D models, like animals (e.g. ant, bird, cat, fish, octopus and so on). Some candidate 3D features we have considered are shape context, geodesic distance, shape diameter and heat kernel. However, we plan to develop a part-aware hybrid shape descriptor which may utilize the above features. Shape context is considered based on the findings in [74]: among the aforementioned features adopted for 3D mesh segmentation, which is highly related to partial retrieval, shape context is the most important one. We think geodesic distance features are promising because of its superiority in characterizing and recognizing deformable models, which has been demonstrated by Smeets et al. [98, 169, 170]. Motivated by the volumetric heat kernel proposed by [104], we plan to define a part-based heat kernel shape descriptor to further improve the partial retrieval performance. To evaluate our partial retrieval algorithm, we plan to use the following three benchmarks: (1) the benchmark used in the SHREC 2007 partial matching track [187] (Section 6.4); (2) the database for the SHREC 2010 non-rigid 3D shape retrieval track [99]; and (3) the benchmark for the SHREC 2009 query with partial models track [33], for which parts of models are already available.

The key techniques possibly involved in the partial similarity retrieval algorithm include: feature sampling or salient feature points extraction on the surface of a model; efficient feature matching methods, such as the Earth Mover's Distance (EMD) [24]; part and contextual information definitions. The main potential advantages of the retrieval algorithm are as follows: pose invariance for general/deformable models; rotation invariance, compactness and efficiency for dealing with deformable model retrieval.

References

[1] S. Abbasi and F. Mokhtarian. Automatic view selection in multi-view object recognition. In *ICPR*, pages 1013–1016, 2000.

[2] T. Adamek and N. E. O'Connor. A multiscale representation method for non-rigid shapes with a single closed contour. *IEEE Trans. Circuits Syst. Video Techn.*, 14(5):742–753, 2004.

[3] AIM@SHAPE. SHREC contest. Home Page, 2010. http://www.aimatshape.net/event/SHREC.

[4] M. Ankerst, G. Kastenmuller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification. In *Proc. of the 6th International Symposium on Advances in Spatial Databases*, pages 207–226, 1999.

[5] T. F. Ansary, J.-P. Vandeborre, and M. Daoudi. 3D-model search engine from photos. In *CIVR*, pages 89–92, 2007.

[6] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.

[7] M. Attene, S. Marini, M. Spagnuolo, and B. Falcidieno. The fast reject schema for part-in-whole 3D shape matching. In *Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 23–30, 2010.

[8] A. Axenopoulos and P. G. Litos. 3D model retrieval using accurate pose estimation and view-based similarity. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, 2011.

[9] M. Belkin, J. Sun, and Y. Wang. Discrete Laplace operator on meshed surfaces. In *Symposium on Computational Geometry*, pages 278–287, 2008.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[11] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *Eurographics Workshop on 3D Object Retrieval (3DOR 2008)*, pages 1–8, 2008.

[12] P. Besl and N. McKay. A method for registration of 3D shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[13] S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, and B. Falcidieno. 3D classification via structural prototypes. In B. Falcidieno, M. Spagnuolo, Y. S. Avrithis, I. Kompatsiaris, and P. Buitelaar, editors, *SAMT*, volume 4816 of *Lecture Notes in Computer Science*, pages 140–143. Springer, 2007.

[14] S. Biasotti and S. Marini. Sub-part correspondence using structure and geometry. In *Eurographics Italian Chapter Conference*, pages 23–28, 2006.

[15] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *International Journal of Computer Vision*, 89(2-3):266–286, 2010.

[16] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR10*, pages 1704–1711, 2010.

[17] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranic. Using entropy impurity for improved 3D object similarity search. In *ICME*, pages 1303–1306, 2004.

[18] L. Cao, J. Liu, and X. Tang. 3D object retrieval using 2D line drawing and graph based relevance reedback. In *Proceedings of the 14th annual ACM international conference on Multimedia*, MULTIMEDIA'06, pages 105–108, New York, NY, USA, 2006. ACM.

[19] A. Chalechale, A. Mertins, and G. Naghdy. Edge image description using angular radial partitioning. *IEE Proceedings-Vision Image and Signal Processing*, 151(2):28–41, 2004.

[20] W. Chang-Chang, F. Fraundorfer, J. Frahm, and M. Pollefeys. 3D model search and pose estimation from single images using VIP features. In *Proceedings of S3D workshop in conjunction with CVPR'08*, pages 1–8, 2008.

[21] M. Chaouch and A. Verroust-Blondet. A new descriptor for 2D depth image indexing and 3D model retrieval. In *ICIP (6)*, pages 373–376, 2007.

[22] M. Chaouch and A. Verroust-Blondet. Alignment of 3D models. *Graphical Models*, 71(2):63–76, 2009.

[23] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung. On visual similarity based 3D model retrieval. *Comput. Graph. Forum*, 22(3):223–232, 2003.

[24] S. D. Cohen and L. J. Guibas. The Earth Mover's Distance under transformation sets. In *ICCV*, pages 1076–1083, 1999.

[25] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. A. Funkhouser, and S. Rusinkiewicz. Where do people draw lines? *ACM Trans. Graph.*, 27(3), 2008.

[26] N. D. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. J. Dickinson, and P. B. Kantor. 3D object retrieval using many-to-many matching of curve skeletons. In *Shape Modeling International*, pages 368–373, 2005.

[27] M. Corsini, M. Dellepiane, F. Ponchio, and R. Scopigno. Image-to-geometry registration: a mutual information method exploiting illumination-related geometric properties. *Comput. Graph. Forum*, 28(7):1755–1764, 2009.

[28] C. Cyr, A. Kamal, T. Sebastian, and B. Kimia. 2D-3D registration based on shape matching. In *Proceedings of IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, pages 198–203, 2000.

[29] P. Daras and A. Axenopoulos. A 3D shape retrieval framework supporting multimodal queries. *International Journal of Computer Vision*, 89(2-3):229–247, 2010.

[30] F. de Goes, S. Goldenstein, and L. Velho. A hierarchical segmentation of articulated bodies. *Comput. Graph. Forum*, 27(5):1349–1356, 2008.

[31] T. K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, and Y. Wang. Persistent heat signature for pose-oblivious matching of incomplete models. In *Eurographics Symposium on Geometry Processing*, pages 1545–1554, 2010.

[32] D. Doug, F. Adam, R. Szymon, and S. Anthony. Suggestive contours for conveying shape. *ACM Transactions on Graphics*, 22(3):848–855, 2003.

[33] H. Dutagaci, A. Godil, A. Axenopoulos, P. Daras, T. Furuya, and R. Ohbuchi. SHREC'09 track: Querying with partial models. In M. Spagnuolo, I. Pratikakis, R. C. Veltkamp, and T. Theoharis, editors, *3DOR*, pages 69–76. Eurographics Association, 2009.

[34] R. C. Eberhart and X. Hu. Human tremor analysis using particle swarm optimization 1999. In *Proc. of the Congress on Evolutionary Computation*, pages 1927–1930, 1999.

[35] M. Elad, A. Tal, and S. Ar. Content based retrieval of VRML objects - an iterative and interactive approach. In *Proc. of the 6th Eurographics Workshop on Multimedia*, pages 97–108, 2001.

[36] R. Fang, A. Godil, X. Li, and A. Wagan. A new shape benchmark for 3D object retrieval. In *Proc. of the 4th International Symposium on Advances in Visual Computing (ISVC)*, pages 381–392, 2008.

[37] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surfaces. *Computer Vision and Image Understanding*, 65(3):403–424, 1997.

[38] A. Ferreira, S. Marini, M. Attene, M. J. Fonseca, M. Spagnuolo, J. A. Jorge, and B. Falcidieno. Thesaurus-based 3D object retrieval with part-in-whole matching. *International Journal of Computer Vision*, 89(2-3):327–347, 2010.

[39] M. Fisher and P. Hanrahan. Context-based search for 3D models. *ACM Trans. Graph.*, 29:182:1–182:10, 2011.

[40] M. J. Fonseca, A. Ferreira, and J. A. Jorge. Towards 3D modeling using sketches and retrieval . In *Sketch-Based Interfaces and Modeling*, pages 127–136, 2004.

[41] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2004.

[42] H.-B. Fu, D. Cohen-Or, G. Dror, and A. Sheffer. Upright orientation of man-made objects. *ACM Trans. Graph.*, 27(3), 2008.

[43] T. A. Funkhouser, P. Min, M. M. Kazhdan, J. Chen, J. A. Halderman, D. P. Dobkin, and D. P. Jacobs. A search engine for 3D models. *ACM Trans. Graph.*, 22(1):83–105, 2003.

[44] T. Furuya and R. Ohbuchi. Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features. In *CIVR*, pages 1–8, 2009.

[45] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph.*, 25(1):130–150, 2006.

[46] K. Gebal, J. A. Bærentzen, H. Aanæs, and R. Larsen. Shape analysis using the auto diffusion function. *Comput. Graph. Forum*, 28(5):1405–1413, 2009.

[47] A. Godil, H. Dutagaci, C. B. Akgül, A. Axenopoulos, B. Bustos, M. Chaouch, P. Daras, T. Furuya, S. Kreft, Z. Lian, T. Napoleon, A. Mademlis, R. Ohbuchi, P. L. Rosin, B. Sankur, T. Schreck, X. Sun, M. Tezuka, A. Verroust-Blondet, M. Walter, and Y. Yemez. SHREC'09 track: Generic shape retrieval. In *Proc. of Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 61–68, 2009.

[48] A. Golovinskiy and T. Funkhouser. Randomized cuts for 3D mesh analysis. *ACM Transactions on Graphics (Proc. SIGGRAPH ASIA)*, 27(5), Dec. 2008.

[49] S. Gottschalk, M. C. Lin, and D. Manocha. Obbtree: A hierarchical structure for rapid interference detection. In *SIGGRAPH*, pages 171–180, 1996.

[50] A. B. Hamza and H. Krim. Geodesic object representation and recognition. In I. Nyström, G. S. di Baja, and S. Svensson, editors, *DGCI*, volume 2886 of *Lecture Notes in Computer Science*, pages 378–387. Springer, 2003.

[51] E.-H. Han and G. Karypis. Centroid-based document classification: analysis and experimental results. In *PKDD*, pages 424–431, 2000.

[52] K. Hara, Y. Kabashima, Y. Iwashita, R. Kurazume, and T. Hasegawa. Robust 2D-3D alignment based on geometrical consistency. In *Proceedings of the Sixth International Conference on 3-D Digital Imaging and Modeling*, pages 273–280, 2007.

[53] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[54] M. Hilaga, Y. Shinagawa, T. Komura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH 2001*, pages 203–212, 2001.

[55] B. Horn. Extended gaussian images. *Proc. of the IEEE*, 72(12):1671–1686, 1984.

[56] S. Hou, K. Lou, and K. Ramani. SVM-based semantic clustering and retrieval of a 3D model database. *Computer-Aided Design & Applications*, 2:155–164, 2005.

[57] S. Hou and K. Ramani. Sketch-based 3D engineering part class browsing and retrieval. In *Proc. of EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, pages 131–138, 2006.

[58] S. Hou and K. Ramani. Classifier combination for sketch-based 3D part retrieval. *Computers & Graphics*, 31(4):598–609, 2007.

[59] J. Hu and J. Hua. Salient spectral geometric features for shape matching and retrieval. *The Visual Computer*, 25(5-7):667–675, 2009.

[60] K. S. Huang and M. M. Trivedi. 3D shape context based gesture analysis integrated with tracking using omni video array. In *Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[61] D. Huttenlocher and S. Ullman. Object recognition using alignment. In *ICCV*, pages 102–111, 1987.

[62] Y. Iwashita, R. Kurazume, K. Hara, and T. Hasegawa. Fast alignment of 3D geometrical models and 2D color images using 2D distance maps. In *Proceedings of the*

5th International Conference on 3-D Digital Imaging and Modeling (3-DIM), pages 164–171, 2005.

[63] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, 2005.

[64] V. Jain and H. Zhang. Shape-based retrieval of articulated 3D models using spectral embedding. In *Geometric Modeling and Processing 2006*, pages 299–312, 2006.

[65] V. Jain, H. Zhang, and O. van Kaick. Non-rigid spectral correspondence of triangle meshes. *International Journal of Shape Modeling*, 13(1):101–124, 2007.

[66] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, 2002.

[67] S. Jayanti, Y. Kalyanaraman, N. Iyer, and K. Ramani. Developing an engineering shape benchmark for CAD models. *Computer-Aided Design*, 38(9):939–953, 2006.

[68] S. Jeannin, L. Cieplinski, J. R. Ohm, and M. Kim. *MPEG-7 visual part of eXperimentation model version 7.0*. ISO/IEC JTC1/SC29/WG11/N3521, Beijing, 2000.

[69] H. Jegou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

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

[71] I. Jolliffe. *Principal Component Analysis (2nd edition)*. Springer, Heidelberg, 2002.

[72] R. Jonkerand and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[73] T. Judd, F. Durand, and E. H. Adelson. Apparent ridges for line drawing. *ACM Transactions on Graphics*, 26(3):19, 2007.

[74] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *ACM Trans. Graph.*, 29(4), 2010.

[75] S. Kanai. Content-based 3D mesh model retrieval from hand-written sketch. *International Journal on Interactive Design and Manufacturing*, 2(2):87–98, 2008.

[76] D. R. Karger and C. Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996.

[77] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph.*, 22(3):954–961, 2003.

[78] M. M. Kazhdan, T. A. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *Symposium on Geometry Processing*, pages 156–164, 2003.

[79] A. Khotanzad and Y. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[80] M. Körtgen, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *Proc. of the 7th Central European Seminar on Computer Graphics(CESCG 2003)*, pages 1–12, 2003.

[81] R. Kurazume, K. Nishino, Z. Zhang, and K. Ikeuchi. Simultaneous 2D images and 3D geometric model registration for texture mapping utilizing reflectance attribute. In *Proceedings of the Fifth Asian Conference on Computer Vision*, pages 99–106, 2002.

[82] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In C. E. Brodley and A. P. Danyluk, editors, *ICML*, pages 282–289. Morgan Kaufmann, 2001.

[83] H. Laga. Semantics-driven approach for automatic selection of best views of 3D shapes. In *3DOR*, pages 15–22, 2010.

[84] H. Laga and M. Nakajima. Supervised learning of salient 2D views of 3D models. *Journal of Society for Art and Sciences*, 7(4):124–131, 2008.

[85] H. Laga and M. Nakajima. Supervised learning of similarity measures for content-based 3D model retrieval. In *LKR*, pages 210–225, 2008.

[86] S. Lavallee and R. Szeliski. Recovering the position and orientation of free-form objects from image contours using 3D distance maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):378–390, 1995.

[87] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Tran. Graph.*, 24(3):659–666, 2005.

[88] J. Lee and T. Funkhouser. Sketch-based search and composition of 3D models. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*, June 2008.

[89] B. Leng and Z. Xiong. Modelseek: an effective 3D model retrieval system. *Multimedia Tools and Applications*, 51:935–962, 2011.

[90] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, (2):164–168, 1944.

[91] B. Levent, M. Chris, and S. Kenji. Pen-based styling design of 3D geometry using concept sketches and template models. In *Proceedings of the ACM Symposium on Solid and Physical Modeling 2006*, pages 149–160, 2006.

[92] M. Leventon, M. Wells, and W. Grimson. Multiple view 2D-3D mutual information registration. In *Proceedings of Image Understanding Workshop, DARPA97*, pages 625–630, 1997.

[93] B. Lévy. Laplace-Beltrami eigenfunctions towards an algorithm that “understands” geometry. In *SMI*, page 13, 2006.

[94] B. Li, A. Godil, M. Aono, X. Bai, T. Furuya, L. Li, R. Lpez-Sastre, H. Johan, R. Ohbuchi, C. Redondo-Cabrera, A. Tatsuma, T. Yanagimachi, and S. Zhang. SHREC’12 Track: Generic 3D Shape Retrieval. In *M. Spagnuolo, M. Bronstein, A. Bronstein and A. Ferreira, editors, 3DOR*, pages 119–126, 2012.

[95] B. Li and H. Johan. View context: A 3D model feature for retrieval. *In: S. Boll et al. (eds.): MMM 2010, LNCS, Springer, Heidelberg*, 5916:185–195, 2010.

[96] B. Li and H. Johan. View context based 2D sketch-3D model alignment. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 45 –50, 2011.

[97] B. Li, T. Schreck, A. Godil, M. Alexa, T. Boubekeur, B. Bustos, J. Chen, M. Eitz, T. Furuya, K. Hildebrand, S. Huang, H. Johan, A. Kuijper, R. Ohbuchi, R. Richter, J. M. Saavedra, M. Scherer, T. Yanagimachi, G. J. Yoon, and S. M. Yoon. SHREC'12 Track: Sketch-Based 3D Shape Retrieval. In *M. Spagnuolo, M. Bronstein, A. Bronstein and A. Ferreira, editors, 3DOR*, pages 109–118, 2012.

[98] Z. Lian, A. Godil, B. Bustos, M. Daoudi, J. Hermans, S. Kawamura, Y. Kurita, G. Lavoué, H. V. Nguyen, R. Ohbuchi, Y. Ohkita, Y. Ohishi, F. P. M. Reuter, I. Sipiran, D. Smeets, P. Suetens, H. Tabia, and D. Vandermeulen. SHREC '11 track: Shape retrieval on non-rigid 3D watertight meshes. In *3DOR*, pages 79–88, 2011.

[99] Z. Lian, A. Godil, T. Fabry, T. Furuya, J. Hermans, R. Ohbuchi, C. Shu, D. Smeets, P. Suetens, D. Vandermeulen, and S. Wuhrer. SHREC'10 track: Non-rigid 3D shape retrieval. In M. Daoudi, T. Schreck, M. Spagnuolo, I. Pratikakis, R. C. Veltkamp, and T. Theoharis, editors, *3DOR*, pages 101–108. Eurographics Association, 2010.

[100] Z. Lian, A. Godil, and X. Sun. Visual similarity based 3D shape retrieval using Bag-of-Features. In *Shape Modeling International*, pages 25–36, 2010.

[101] Z. Lian, P. Rosin, and X. Sun. Rectilinearity of 3D meshes. *International Journal of Computer Vision*, 89(2):130–151, 2010.

[102] J. Liebelt and K. Schertler. Precise registration of 3D models to images by swarming particles. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07*, 2007.

[103] R. Liu and H. Zhang. Segmentation of 3D meshes through spectral clustering. In *Proc. of Pacific Graphics*, pages 298–305, 2004.

[104] R. Liu, H. Zhang, A. Shamir, and D. Cohen-Or. A part-aware surface metric for shape analysis. *Comput. Graph. Forum*, 28(2):397–406, 2009.

[105] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *27th Annual International ACM SIGIR Conference (SIGIR 2004)*, pages 186–193, 2004.

[106] Y. Liu, X. Wang, H.-Y. Wang, H. Zha, and H. Qin. Learning robust similarity measures for 3D partial shape retrieval. *International Journal of Computer Vision*, 89(2-3):408–431, 2010.

[107] Y. Liu, H. Zha, and H. Qin. Shape topics: A compact representation and new algorithms for 3D partial shape retrieval. In *CVPR (2)*, pages 2025–2032. IEEE Computer Society, 2006.

[108] C. T. Loop. *Smooth subdivision surfaces based on triangles*. Masters Thesis, The University of Utah, 1987.

[109] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[110] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[111] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, 16(2):187–198, 1997.

[112] S. Mahmoudi and M. Daoudi. 3D models retrieval by using characteristic views. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, pages 457 – 460 vol.2, 2002.

[113] S. Mahmoudi and M. Daoudi. A probabilistic approach for 3D shape retrieval by characteristic views. *Pattern Recognition Letters*, 28(13):1705–1718, 2007.

[114] A. Makadia and K. Daniilidis. Spherical correlation of visual representations for 3D model retrieval. *International Journal of Computer Vision*, 89(2-3):193–210, 2010.

[115] S. Marini, G. Patanè, M. Spagnuolo, and B. Falcidieno. Feature selection for enhanced spectral shape comparison. In *Eurographics Workshop on 3D Object Retrieval (3DOR)*, pages 31–38, 2010.

[116] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963.

[117] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, 1982.

[118] M. Martinek and R. Grosso. Optimal rotation alignment of 3D objects using a GPU-based similarity function. *Computers & Graphics*, 33(3):291–298, 2009.

[119] T. B. Mathias Eitz, Kristian Hildebrand and M. Alexa. Sketch-based 3D shape retrieval. In *ACM SIGGRAPH 2010 Talk Program*, 2010.

[120] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. pages 35–57. Springer-Verlag, 2002.

[121] F. Mokhtarian and S. Abbasi. Automatic selection of optimal views in multi-view object recognition. In M. Mirmehdi and B. T. Thomas, editors, *BMVC*. British Machine Vision Association, 2000.

[122] F. Mokhtarian and S. Abbasi. Robust automatic selection of optimal views in multi-view free-form object recognition. *Pattern Recognition*, 38(7):1021–1031, 2005.

[123] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[124] G. Mori and J. Malik. Recovering 3D human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1052–1062, 2006.

[125] T. Napoléon and H. Sahbi. From 2D silhouettes to 3D object retrieval: Contributions and benchmarking. *EURASIP Journal on Image and Video Processing*, 2010(1):1–17, 2010.

[126] T. Napoléon and H. Sahbi. Sketch-driven mental 3D object retrieval. volume 7526, page 75260L. SPIE, 2010.

[127] NIST. SHREC 2012 track: Generic 3D shape retrieval. Home Page, 2012. http://www.itl.nist.gov/iad/vug/sharp/contest/2012/Generic3D/index.html.

[128] NIST. SHREC 2012 Track: Sketch-Based 3D Shape Retrieval. Home Page, 2012. http://www.itl.nist.gov/iad/vug/sharp/contest/2012/SBR/index.html.

[129] M. Novotni and R. Klein. 3D Zernike descriptors for content based shape retrieval. In *Proceedings of the 8th ACM Symposium on Solid Modeling and Applications*, pages 216–225. ACM Press, 2003.

[130] R. Ohbuchi and T. Furuya. Distance metric learning and feature combination for shape-based 3D model retrieval. In *Proceedings of the ACM workshop on 3D object retrieval*, 3DOR '10, pages 63–68. ACM, 2010.

[131] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3D model retrieval. In *Shape Modeling International*, pages 93–102, 2008.

[132] R. Ohbuchi, A. Yamamoto, and J. Kobayashi. Learning semantic categories for 3D model retrieval. In J. Z. Wang, N. Boujemaa, A. D. Bimbo, and J. Li, editors, *Multimedia Information Retrieval*, pages 31–40. ACM, 2007.

[133] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[134] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1):85–103, 2009.

[135] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *Proc. of Shape Modeling and Applications*, pages 154–166, 2001.

[136] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, 2002.

[137] M. Ovsjanikov, A. Bronstein, M. Bronstein, and L. Guibas. Shape google: a computer vision approach to isometry invariant shape retrieval. pages 320 –327, sep. 2009.

[138] M. Ovsjanikov, Q. Mrigot, F. Mmoli, and L. Guibas. One point isometric matching with the heat kernel. In *Eurographics Symposium on Geometry Processing*, pages 1555–1564, 2010.

[139] X. Pan, Q. You, Z. Liu, and Q. H. Chen. 3D shape retrieval by poisson histogram. *Pattern Recognition Letters*, 32(6):787–794, 2011.

[140] P. Papadakis, I. Pratikakis, S. J. Perantonis, and T. Theoharis. Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation. *Pattern Recognition*, 40(9):2437–2452, 2007.

[141] P. Papadakis, I. Pratikakis, T. Theoharis, G. Passalis, and S. J. Perantonis. 3D object retrieval using an efficient and compact hybrid shape descriptor. In *3DOR*, pages 9–16, 2008.

[142] P. Papadakis, I. Pratikakis, T. Theoharis, and S. Perantonis. PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. *International Journal of Computer Vision*, 89(2-3):177–192, 2010.

[143] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, pages 727–734, 2000.

[144] J. Podolak, P. Shilane, A. Golovinskiy, S. Rusinkiewicz, and T. Funkhouser. A planar-reflective symmetry transform for 3D shapes. *ACM Trans. Graph.*, 25(3), 2006.

[145] J. Pu and R. Karthic. An automatic drawing-like view generation method from 3D models. In *Proc. of ASME IDETC/CIE 2005, 25th Computers and Information in Engineering (CIE) Conference*, pages 301–320, 2005.

[146] J. Pu, K. Lou, and K. Ramani. A 2D sketch-based user interface for 3D CAD model retrieval. *Computer-Aided Design & Applications*, 2(6):717–725, 2005.

[147] J. Pu and K. Ramani. On visual similarity based 2D drawing retrieval. *Computer-Aided Design*, 38(3):249–259, 2006.

[148] J. Pu and K. Ramani. Shapelab: A unified framework for 2D & 3D shape retrieval. In *3DPVT*, pages 1072–1079, 2006.

[149] J. Rabin, G. Peyré, and L. D. Cohen. Geodesic shape retrieval via optimal mass transport. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *ECCV (5)*, volume 6315 of *Lecture Notes in Computer Science*, pages 771–784. Springer, 2010.

[150] D. Raviv, M. M. Bronstein, A. M. Bronstein, and R. Kimmel. Volumetric heat kernel signatures. In *Proceedings of the ACM workshop on 3D object retrieval*, 3DOR '10, pages 39–44, New York, NY, USA, 2010. ACM.

[151] S. R.Buss. *3D Computer Graphics: A Mathematical Introduction with OpenGL*. Cambridge University Press, 2003.

[152] M. Reuter. Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions. *Int. J. Comput. Vision*, 89(2-3):287–308, 2010.

[153] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete Laplace-Beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009.

[154] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-spectra as fingerprints for shape matching. In L. Kobbelt and V. Shapiro, editors, *Symposium on Solid and Physical Modeling*, pages 101–106, 2005.

[155] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace-Beltrami spectra as "Shape-DNA" of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[156] J. Revaud, G. Lavoué, and A. Baskurt. Improving Zernike moments comparison for optimal similarity and rotation angle retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(4):627–636, 2009.

[157] J. M. Saavedra and B. Bustos. An improved histogram of edge local orientations for sketch-based image retrieval. In M. Goesele, S. Roth, A. Kuijper, B. Schiele, and K. Schindler, editors, *DAGM-Symposium*, volume 6376 of *Lecture Notes in Computer Science*, pages 432–441. Springer, 2010.

[158] J. M. Saavedra, B. Bustos, M. Scherer, and T. Schreck. STELA: sketch-based 3D model retrieval using a structure-based local approach. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 26:1–26:8, 2011.

[159] A. Sajjanhar, G. Lu, and D. Zhang. Image retrieval using modified generic Fourier descriptors. In *Computers and Their Applications*, pages 32–35, 2004.

[160] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–1595, 1996.

[161] K. Sfikas, T. Theoharis, and I. Pratikakis. ROSy+: 3D Object Pose Normalization Based on PCA andReflective Object Symmetry with Application in 3D Object Retrieval. *International Journal of Computer Vision*, 91(3):262–279, 2011.

[162] S. Shalom, L. Shapira, A. Shamir, and D. Cohen-Or. Part analogies in sets of objects. In *3DOR*, pages 33–40, 2008.

[163] L. Shapira, S. Shalom, A. Shamir, D. Cohen-Or, and H. Zhang. Contextual part analogies in 3D objects. *International Journal of Computer Vision*, 89:309–326, 2010.

[164] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.

[165] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *Proc. of IEEE International Conference on Evolutionary Computation (ICEC)*, pages 69–73, 1998.

[166] J.-L. Shih and H.-Y. Chen. A 3D model retrieval approach using the interior and exterior 3D shape information. *Multimedia Tools and Applications*, 43(1):45–62, 2009.

[167] P. Shilane, P. Min, M. M. Kazhdan, and T. A. Funkhouser. The Princeton shape benchmark. In *Proc. of Shape Modeling International (SMI)*, pages 167–178, 2004.

[168] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. J. Dickinson. Retrieving articulated 3-d models using medial surfaces. *Machine Vision and Applications*, 19(4):261–275, 2008.

[169] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, and P. Suetens. Isometric deformation modelling for object recognition. In *CAIP*, pages 757–765, 2009.

[170] D. Smeets, T. Fabry, J. Hermans, D. Vandermeulen, and P. Suetens. Inelastic deformation invariant modal representation for non-rigid 3D object recognition. In F. J. P.

López and R. B. Fisher, editors, *AMDO*, volume 6169 of *Lecture Notes in Computer Science*, pages 162–171. Springer, 2010.

[171] J. G. Snodgrass and M. Vanderwart. A standardized set of 260 pictures: norms for name agreement, image agreement, familiarity, and visual complexity. *Journal of Experimental Pyschology: Human Learning and Memory*, 6(2):174–215, 1980.

[172] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In M. Alexa, M. Kazhdan, and K. Polthier, editors, *Eurographics Symposium on Geometry Processing*, pages 1383–1392, 2009.

[173] H. Sundar, D. Silver, N. Gagvani, and S. J. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modeling International*, pages 130–139, 2003.

[174] A.-P. Ta, C. Wolf, G. Lavoue, and A. Baskurt. 3D object detection and viewpoint selection in sketch images using local patch-based Zernike moments. In *Proceedings of the 2009 Seventh International Workshop on Content-Based Multimedia Indexing*, pages 189–194, Washington, DC, USA, 2009. IEEE Computer Society.

[175] Y. Takeda. Sketch based 3D model retrieval using a 2D image. In *ACM SIGGRAPH 2010 Poster*, 2010.

[176] J. W. H. Tangelder and R. C. Veltkamp. A survey of content based 3D shape retrieval methods. *Multimedia Tools Appl.*, 39(3):441–471, 2008.

[177] A. Tatsuma and M. Aono. Multi-fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. *The Visual Computer*, 25(8):785–804, 2009.

[178] J. Tedjokusumo and W. K. Leow. Normalization and alignment of 3D objects based on bilateral symmetry planes. In T.-J. Cham and et al., editors, *MMM (1)*, volume 4351 of *Lecture Notes in Computer Science*, pages 74–85. Springer, 2007.

[179] J. Tierny, J.-P. Vandeborre, and M. Daoudi. Partial 3D shape retrieval by reeb pattern unfolding. *Computer Graphics Forum*, 28(1):41–55, 2009.

[180] R. Toldo, U. Castellani, and A. Fusiello. Visual vocabulary signature for 3D object retrieval and partial matching. In *Eurographics Workshop on 3D Object Retrieval (3DOR 2009)*, pages 21–28, 2009.

[181] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(5):854–869, 2007.

[182] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. *Comput. Graph. Forum*, 27(2):251–260, 2008.

[183] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. In *Proc. of Eurographics State-of-the-art Report*, 2010.

[184] A. Vaxman, M. Ben-Chen, and C. Gotsman. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Trans. Graph.*, 29(4), 2010.

[185] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In T. Ertl, B. Girod, H. Niemann, and H.-P. Seidel, editors, *VMV*, pages 273–280. Aka GmbH, 2001.

[186] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic view selection using viewpoint entropy and its applications to image-based modelling. *Comput. Graph. Forum*, 22(4):689–700, 2003.

[187] R. C. Veltkamp and F. B. ter Haar. *SHREC 2007 3D Retrieval Contest*. Technical Report UU-CS-2007-015, Department of Information and Computing Sciences, Utrecht University, 2007.

[188] C. Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2003.

[189] P. Viola and M. Wells. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.

[190] D. V. Vranic. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.

[191] D. V. Vranic. DESIRE: a composite 3D-shape descriptor. In *ICME*, pages 962–965, 2005.

[192] J. Wang, Y. He, H. Tian, and H. Cai. Retrieving 3D CAD model by freehand sketches for design reuse. *Advanced Engineering Informatics*, 22(3):385–392, 2008.

[193] L. Wang, Y. Zhang, and J. Feng. On the Euclidean distance of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1334–1339, 2005.

[194] R. Wessel, I. Blümel, and R. Klein. A 3D shape benchmark for retrieval and automatic classification of architectural data. In *3DOR*, pages 53–56, 2009.

[195] R. Wessel and R. Klein. Learning the compositional structure of man-made objects for 3D shape retrieval. In *3DOR*, pages 39–46, 2010.

[196] H.-Y. Wu, H. Zha, T. Luo, X. Wang, and S. Ma. Global and local isometry-invariant descriptor for 3D shape comparison and partial matching. In *CVPR*, pages 438–445. IEEE, 2010.

[197] D. Xu and H. Li. 3D shape retrieval integrated with classification information. In *Image and Graphics, 2007. ICIG 2007. Fourth International Conference on*, pages 774 –779, 2007.

[198] A. Yamamoto, M. Tezuka, T. Shimizu, and R. Ohbuchi. SHREC'08 entry: Semi-supervised learning for semantic 3D model retrieval. In *Shape Modeling International*, pages 241–243. IEEE, 2008.

[199] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. G. Belyaev, and H.-P. Seidel. Towards stable and salient multi-view representation of 3D shapes. In *Proc. of Shape Modeling International (SMI)*, page 40, 2006.

[200] P.-T. Yap, P. Raveendran, and S.-H. Ong. Image analysis by krawtchouk moments. *IEEE Transactions on Image Processing*, 12(11):1367–1377, 2003.

[201] S. M. Yoon, M. Scherer, T. Schreck, and A. Kuijper. Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours. In *ACM Multimedia*, pages 193–200, 2010.

[202] S. Zamora and T. Sherwood. Sketch-Based Recognition System for General Articulated Skeletal Figures. In M. Alexa and E. Y.-L. Do, editors, *Sketch-Based Interfaces and Modeling*, pages 119–126, Annecy, France, 2010. Eurographics Association.

[203] D. Zhang and G. Lu. A comparative study of Fourier descriptors for shape representation and retrieval. In *Proc. of the Fifth Asian Conference on Computer Vision (ACCV02)*, pages 646–651, 2002.

[204] D. Zhang and G. Lu. Shape-based image retrieval using generic Fourier descriptor. *Signal Processing: Image Communication*, (17):825–848, 2002.

[205] D. Zhang and G. Luo. 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)*, pages 1–9, 2001.

[206] D. Zhang and G. Luo. An integrated approach to shape based image retrieval. In *Proc. of the 5th Asian Conference on Computer Vision (ACCV 2002)*, pages 652–657, 2002.

[207] H. Zhang, O. van Kaick, and R. Dyer. Spectral methods for mesh processing and analysis. In *Proc. of Eurographics State-of-the-art Report*, pages 1–22, 2007.

Publications

1. **Bo Li**, Henry Johan: *Sketch-Based 3D Model Retrieval by Incorporating 2D-3D Alignment*, Multimedia Tools and Applications, Accepted, Online First version available (12 February 2012), pp. 1-23. DOI:10.1007/s11042-012-1009-0, 2012.

2. **Bo Li**, Henry Johan: *3D Model Retrieval using Hybrid Features and Class Information*, Multimedia Tools and Applications, Accepted, Online First version available (22 September 2011), pp. 1-26. DOI:10.1007/s11042-011-0873-3, 2011.

3. Henry Johan, **Bo Li**, Yuanmin Wei, Iskandarsyah. *3D Model Alignment based on Minimum Projection Area*. Computer Graphics International 2011 (CGI 2011), The Visual Computer 27(6-8): 565-574, 2011.

4. **Bo Li**, Henry Johan: *View Context Based 2D Sketch-3D Model Alignment*. In IEEE Computer Society's Workshop on Applications of Computer Vision (WACV 2011), Hawaii, USA, January 5-6, 2011, pages: 45-50.

5. **Bo Li**, Henry Johan: *View Context: A 3D Model Feature for Retrieval*. Lecture Notes in Computer Science, Advances in Multimedia Modeling, The 16th International Conference on MultiMedia Modeling (MMM 2010), LNCS 5916: 185-195, 2010.

6. **Bo Li**, Henry Johan: *3D Model Retrieval Using Local and Global Radial Distances*. The International Workshop on Advanced Image Technology (IWAIT 2010), 2010.

7. **Bo Li**, Henry Johan: *A Comparison Study on Two Multi-scale Shape Matching Schemes*. Lecture Notes in Computer Science, Advances in Visual Computing, The 4th International Symposium on Visual Computing (ISVC 2008), LNCS 5359: 440-449, 2008.

8. **Bo Li**, Afzal Godil, Henry Johan and et al.: *SHREC'12 Track: Sketch-Based 3D Shape Retrieval*, Eurographics Workshop on 3D Object Retrieval 2012 (3DOR 2012), 109-118, 2012.

9. **Bo Li**, Afzal Godil, Henry Johan and et al.: *SHREC'12 Track: Generic 3D Shape Retrieval*, Eurographics Workshop on 3D Object Retrieval 2012 (3DOR 2012), 119-126, 2012.

10. **Bo Li**, Afzal Godil and Henry Johan. *Non-rigid and Partial 3D Model Retrieval Using Hybrid Shape Descriptor and Meta Similarity*. ISVC 2012, Part I, LNCS 7431: 199-209, 2012.