# 3D Object Retrieval and Recognition

GONG, Boqing

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Philosophy
in
Information Engineering

The Chinese University of Hong Kong
May 2010

# Abstract

As the development of techniques for modeling, digitizing and visualizing, 3D data are now becoming explosion in number and are widely recognized as the upcoming wave of digital media. In particular, 3D data acquisition techniques make it possible to acquire one 3D object with detailed shape information in a short time. Accordingly, this led to the development of techniques including processing, recognition, categorization, and retrieval for such 3D objects. In this thesis, we studied 3D object retrieval and 3D facial expression recognition within this field.

3D object retrieval is to search for 3D object(s) meeting some specific requirements within a database or the World Wide Web. In this thesis, a novel feature which is called object flexibility, is proposed at a point of a 3D object to describe how the neighborhood of this point is massively connected to the object. This feature is stable to the deformation of objects' articulations, in addition to commonly concerned linear transforms, i.e., translation, scale, and rotation. A shape descriptor is obtained based on this feature using the bag-of-words model. As an application, the descriptor is used to perform 3D object retrieval. Extensive experiments demonstrate its superiority over a variety of existing 3D shape descriptors in the retrieval of articulated objects, as well as its enhancement of other shape descriptors to retrieve generic 3D objects.

Facial expression recognition has many applications in multimedia processing, and the development of 3D data acquisition techniques make it possible to identify expressions using 3D shape information. We propose an automatic

facial expression recognition approach based on a single 3D face. The shape of an expressional 3D face is approximated as the sum of two parts, a basic facial shape component (BFSC) and an expressional shape component (ESC). The BFSC represents the basic face structure and neutral-style shape and the ESC contains shape changes caused by facial expressions. To separate the BFSC and ESC, our method firstly builds a reference face for each input 3D non-neutral face by a learning method, which well represents the basic facial shape. Then, based on the BFSC and the original expressional face, a facial expression descriptor is designed. The surface depth changes are considered in the descriptor. Finally, the descriptor is input into an Support Vector Machine (SVM) to recognize the type of expression. Unlike previous methods which recognize a facial expression with the help of manually labeled key points and/or a neutral face, our method works on a single 3D face without any manual assistance. Extensive experiments are carried out on the BU-3DFE database and comparisons with existing methods are conducted. Experimental results show the effectiveness of our method.

# Acknowledgement

First of all, I would like to thank to my supervisor, Prof. Xiaoou Tang, who opens the door of my postgraduate study and leads me into a rich and colorful life in The Chinese University of Hong Kong. His deep insight and suggestive guidance play a decisive role in selecting goals of my future life. I also want to show my appreciation and respect to his friendly aptitude to us, and his understanding of students' frequently changing minds. He is always ready to help us.

I would like to thank to my advisor, Prof. Jianzhuang Liu. He is also the direct supervisor on my research. Thanks to many times of enlightening and effective discussions with Prof. Liu, I published my first paper after amount of versions of revisions by him. I have learned a lot from his serious and industrious working spirits during this process. He is not only an enthusiastic, kind hearted, and experienced advisor in research, but also a good teacher in life. I learned swimming and playing tennis from him.

My labmates Yueming Wang and Chunjing Xu offered great assistance to my research progress. I benefit from their ways of thinking and styles of working. Especially Yueming introduced a new field to me—the 3D facial expression recognition, which opens my mind and vision about a panoramic view of my research. My research progress must be slower without their help.

Besides, I can never forget the happy life with all my labmates. We share opinions on the future. We talk about all kinds of issues when we go back to the dormitory. We play tennis, badminton, basketball, and a lot of other kinds

of sports. We also encourage and help each other. I would like to appreciate all of them: Yueming, Deli, A Feng, Zhenguo, Chunjing, Chen Mo, Weige, Liu Ming, Yiwen, Chenyu, Xiao Weige, Yingze, Kaiming, Zhimin, Du Hao, Tianfan, Liu Ke, Xiaotian, Jia Kui, and Yichen.

Last but not the least, I want to thank to my parents and my girlfriend, Zhaojun. They mean everything to me and they support me the most. I am refreshed whenever I think of them. I love you guys!

# Contents

# Chapter 1

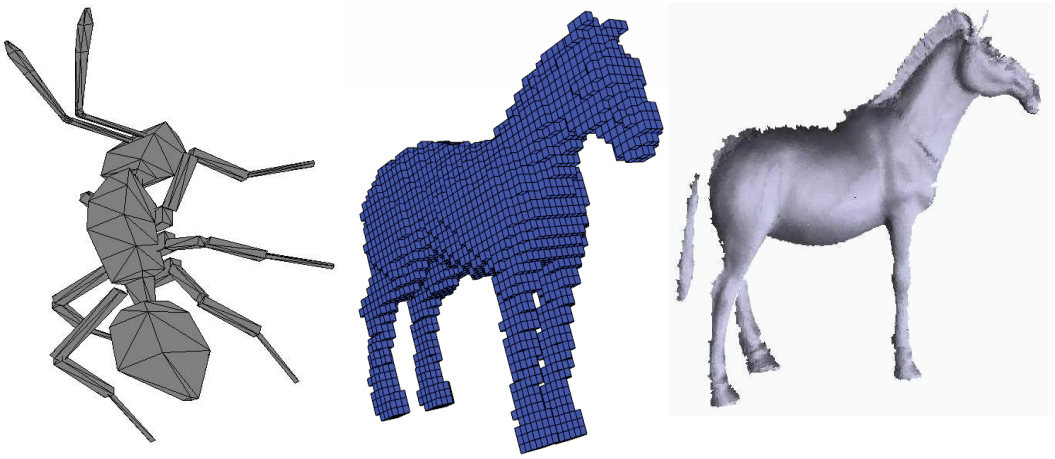
# Introduction

Digital multimedia is gaining prominence on both the Internet and corporate data warehouses. One type of information becoming popular recently is the three-dimensional models. The advancement of modeling, digitizing, and visualizing techniques for three-dimensional (3D) shapes has led to a plenty number of 3D models, from computer-aided design (CAD) to complex protein and molecule representations. As a result, many research efforts have been attracted to deal with problems raised during this upcoming wave. This thesis shows our initial attempt on 3D object retrieval and 3D facial expression recognition.

In this chapter, I will introduce three kinds of representations of 3D objects which are the target and inputs of our algorithms, and then introduce the structure of this thesis about 3D object retrieval and 3D facial expression recognition.

## 1.1 3D Object Representation

The underlying representation of a 3D object can be with various forms. Although currently some 3D scanner is able to capture both shape and appearance of objects or scenes, we focus on only the geometric information here. Polygon mesh, voxel, and range image are the most widely used ones for 3D



(a) Polygon Mesh Representation (b) Voxel Representation (c) Range Image Reopresentation

Figure 1.1: A 3D object can be represented in different forms, like (a)polygon mesh representation, (b)voxel representation, and (c)range image representation.

object representations (see Figure 1.1).

### 1.1.1 Polygon Mesh

Figure 1.1(a) shows one polygon mesh modeled object (an ant). A polygon mesh is a collection of vertices, edges, and faces that defines the shape of a polyhedral object. It explicitly represent the surface of an object, but leaves the volume implicitly modeled [8]. Polygons are able to model an object with various tesselation, and are particularly efficient in representing simple 3D structures with lots of empty or homogeneously-filled space. The study of polygon meshes is a large sub-field of computer graphics and geometric modeling which is out of the scope of this thesis. We refer readers to [32, 39, 35, 9] for recent research progress about polygon mesh.

### 1.1.2 Voxel

The key for computer programs to look for 3D objects is the voxel, which is a volume element representing a set of (graphical) values on a regular grid, like color, density, etc., in 3D space. Since voxels are good at representing regularly sampled spaces that are non-homogeneously filled (comparing to polygon mesh representation in section 1.1.1), voxels are frenquently used in the visualization and analysis of medical and scientific data [8]. A voxel model of a horse with low resolution is shown in Figure 1.1(b).

### 1.1.3 Range Image

The third type of 3D object representation is to use the range image (see Figure 1.1 for example). Rang image can be seen as the raw data of the output of a range camera (e.g. a 3D scanner). It is a 2D image showing the distance to points in a scene from a specific point, normally associated with some type of sensor device. The resulting image, the range image, has pixel values which correspond to the distance, e.g., brighter values mean shorter distance, or vice versa. If the sensor which is used to produce the range image is properly calibrated, the pixel values can be given directly in physical units such as meters [8]. Note that due to different scanning techniques adopted by range cameras, range image may be a polygon mesh or voxel based volume.

The above three models are the main representations of a 3D object currently. The different characteristics lead to various challenges. Our algorithms are able to deal with the first two categories, and are easy to be modified to the third one.

## 1.2 Content-Based 3D Object Retrieval

Since an increasing amount of 3D information is making its way onto the Internet and into corporate databases, users need ways to store, index, and search this information efficiently. Typical text-based searching approaches are not suitable for this, because most 3D objects are not labeled with keywords. As a result, the demand is to retrieve based on the object's own content. We will simply put it as "3D object retrieval" in the left part of the thesis.

Traditionally, a 3D object retrieval algorithm consists of canonical coordinate normalization and preprocessing, feature extraction, similarity match, query formulation and user interface, and performance evaluation, most of which can be found in the next chapter for detailed description.

Chapter 3 is about our algorithm on this topic. We find that most existing methods are only designed for generic objects but not suitable for articulated ones. Therefore, we propose to extract shape descriptor based object flexibility, which is particularly useful for retrieving objects with articulated parts. Besides its stability to articulated parts, flexibility is also invariant to object's position, orientation and size. We find that it is also a good supplement for generic 3D object retrieval. We will describe detailed motivation and definition of flexibility, and present extensive experiments in this chapter.

We make some trial in chapter 4 in improving the performance of 3D object retrieval by incorporating extra information from a set of referent objects. A very weak prior, whether an object is man-made or natural , is found a great enhancement to the retrieval results. Therefore, we introduce a 3D object classifier to learn the prior from the referent set. Some post-processing of the classifier is also proposed in this chapter.

## 1.3 3D Facial Expression Recognition

Anther work in this thesis is about 3D object recognition, in particular, 3D facial expression recognition. Facial expression recognition has a wide application in human-computer interface (HCI) community [24], face recognition [41], and many other multimedia systems. Common concerns like illumination and pose on appearance based methods, have no affection to 3D face models. Therefore, 3D facial expression recognition is supposed to have better results than appearance based methods.

In chapter 5, an automatic 3D facial expression recognition algorithm is developed. To the best of our knowledge, it is the first automatic system without any manually labeling process on this topic. The system consists of a posture alignment step, a learning method to synthesize the basic facial shape component, a feature extraction step, and an expression classifier to indicate the facial expression.

Our main contributions are summarized as below:

- A comprehensive survey on 3D object retrieval is made in chapter 2. We emphasize not only existing algorithms but also the publicly available research resources, e.g., databases and experimental systems.
- We propose a novel shape descriptor for retrieving objects with articulated parts, as well as its combination method to boost generic 3D object retrieval [21].
- We demonstrate a favorable effect of referent objects on 3D object retrieval in chapter 4. Some work will follow this promising experimental result.
- An automatic facial expression recognition system is developed [20]. To

the best our knowledge, this is the first one which do not relies on manually labeling process. We also proposed a learning method to synthesize the basic facial shape component of an expressional face.

# Chapter 2

# 3D Object Retrieval

In this chapter, a conceptual 3D object retrieval framework and some important modules are introduced. During the introduction, some related work will be covered. After that, we will review some publicly available research resources (e.g., experimental 3D search engine and public databases). The closure of this chapter will summarize some challenges for 3D object retrieval.

## 2.1 A Conceptual Framework for 3D Object Retrieval

3D object retrieval is to retrieve object(s) from a database or the Internet meeting some requirements, which often bases on the similarities between objects and the query. Figure 2.1 shows a conceptual framework for a integrated 3D object retrieval system. A user is allowed to submit a query in different ways. After some preprocessing like canonical coordinate normalization, smooth or simplification, shape descriptors are extracted from the query. This descriptor is compared to those stored in the database, which have been extracted off line, based on some matching measures. To speed up this retrieval process, an indexing scheme is often needed for large databases. Then objects are returned as the retrieval result ordered in their similarities to the query. Due to the divergence of the 2D device screen and the 3D object, some special visualization

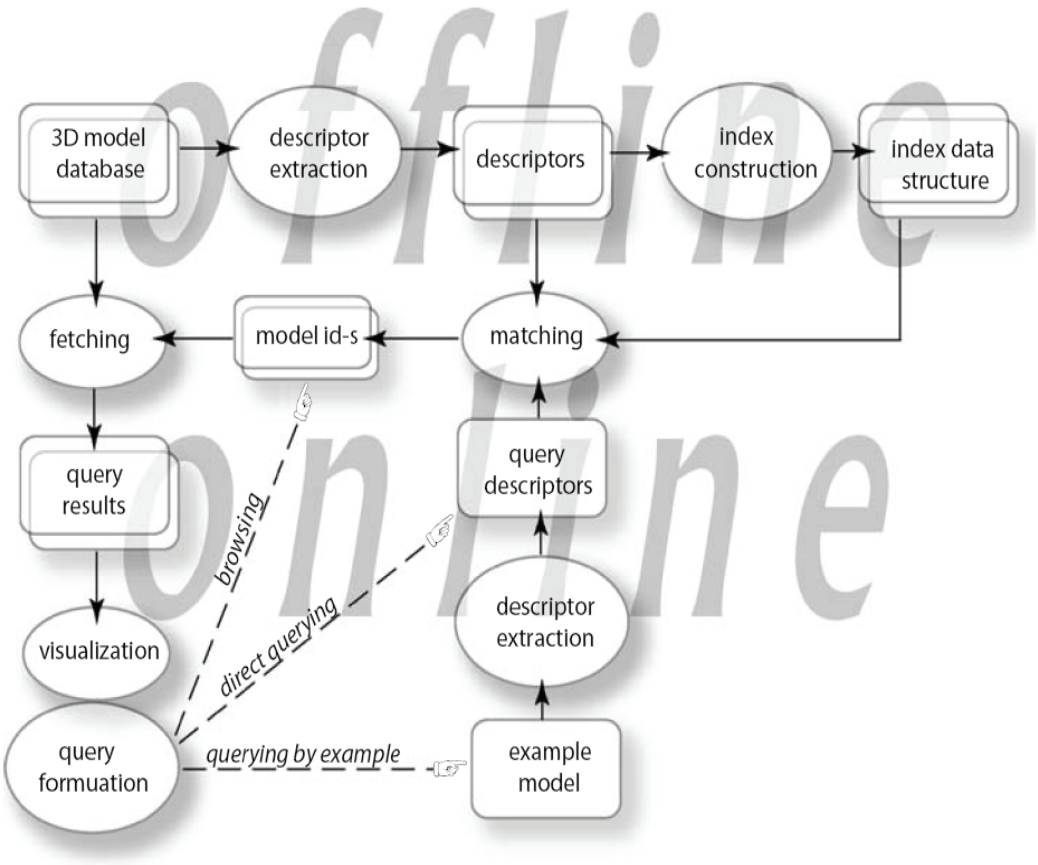



Figure 2.1: A conceptual framework for 3D object retrieval [46].

processes are needed which are quite different from the traditional multimedia (e.g., text or image) retrieval.

Important modules in this framework will be covered in following, including query formulation and user interface, canonical coordinate normalization, some feature extraction methods, similarity match, and performance evaluation.

### 2.1.1 Query Formulation and User Interface

A true 3D search system ought to offer convenient query formulation method and friendly user interface. A convenient way of query formulation becomes much more important than other multimedia retrieval tasks (e.g., text or image retrieval), because currently there are no proper tools for a common user

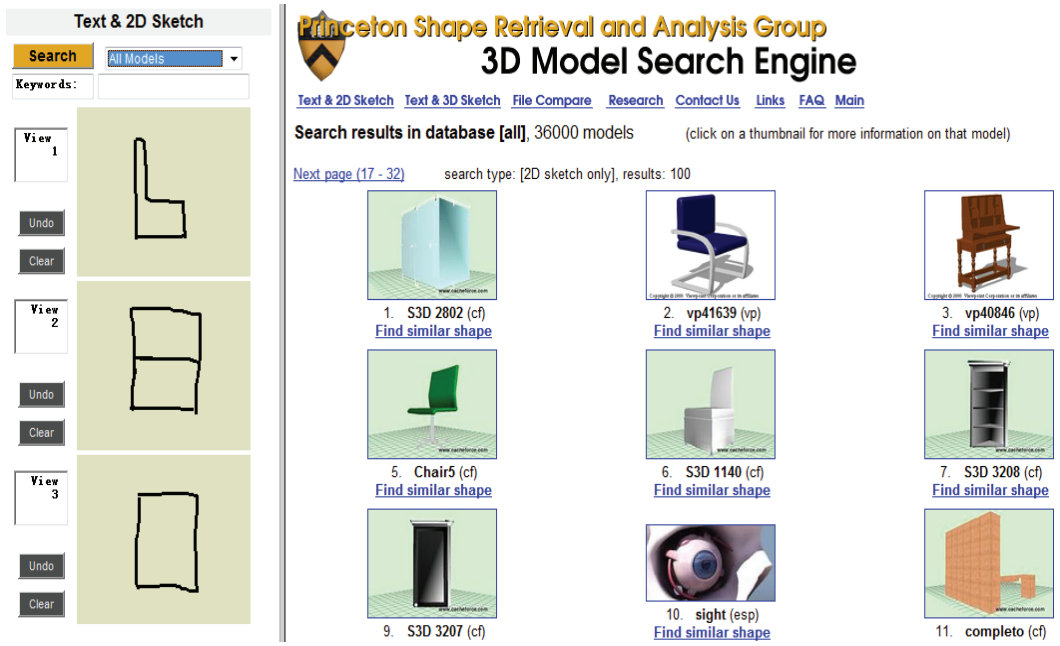
to design a 3D object to be used as the query. Querying by sketching and Querying by example are the two main ways for query formulation.

Figure 2.2 shows two experimental 3D object retrieval systems. The 3D Model Search Engine at Princeton University [6, 28] allows users to formulate a query by sketching three 2D views of a 3D object, sketching a 3D frame of an object, or uploading a file containing 3D object(s), while the 3D Search Engine at the Informatics and Telematics Institute Greece [3] is a typical querying-by-example system. Objects are pre-cataloged and a user can select one from them as the query. In this thesis, our work is currently based on the querying-by-example method.

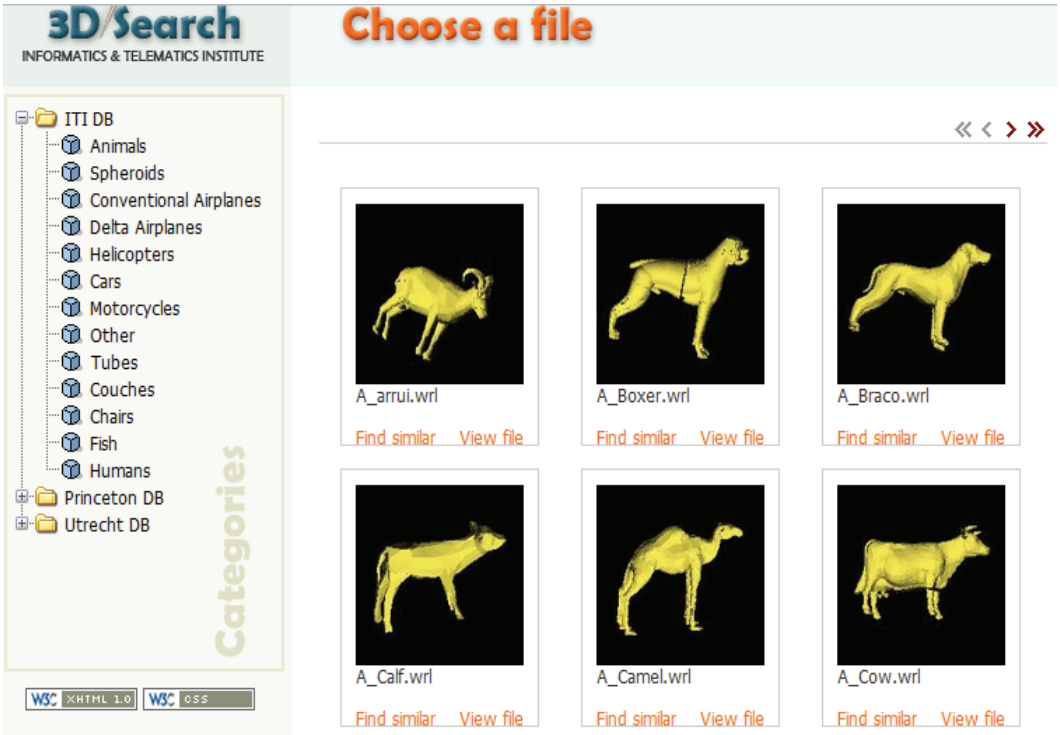
### 2.1.2 Canonical Coordinate Normalization

A 3D object can be with various orientation, scale, and position. Some shape descriptors are invariant to translation, scaling, or rotation, while others are covariant with such linear transformations. A normalization step is thus necessary for the latter case, to transform objects to be compared into the same canonical coordinate frame. Otherwise, the similarity between two analogical objects may be very small if they are with different orientations.

By translating the mass center to the origin and scaling the maximal radius to unit length, objects can be easily normalized to similar position and scale. Principal Component Analysis (PCA) is the most widely used way to rotate an object into canonical orientations. Vranić modified PCA to analyze infinite number of points represented by a union of triangles [49]. However, misalignment is common when two similar shapes but with different point distributions. Tedjokusumo and Leow proposed to use bilateral symmetry planes to normalize and align 3D shapes in [47] and showed its superiority over PCA.




(a)




(b)

Figure 2.2: Experimental 3D object retrieval systems: (a) 3D Model Search Engine at Princeton University [6, 28], which allows user to formulate a query by sketching or uploading a 3D object; (b) 3D Search Engine at the Informatics and Telematics Institute, Greece [3], a typical querying-by-example system that allows users to select queries from cataloged objects.

### 2.1.3 Representations of 3D Objects

A key issue is the type of shape representation(s) that a shape retrieval system accepts [46]. An ideal 3D object representation is not only stable to linear transformation (i.e., translation, scale, and rotation) and non-linear deformation (e.g., articulation and degeneration), but also supposed to be a fair balance between expressiveness of a shape and robustness across different shapes of the same class. In other words, the representations of different 3D objects of the same class ought to be similar, and those of objects from different classes should be diversified.

Amount of work can be found on the representation of 3D objects, and authors of survey papers categorize them into different types [46, 12, 52]. We refer readers to these survey papers for a thorough understanding of existing methods. In this thesis, we would like to list some sort of them and will compare their effectiveness in the following chapter. They are,

- point distribution based D2 [31] and AAD (Absolute-Angle Distance histogram) [29];
- image based LFD (LightField Descriptor) [15], DBD (Depth Buffer Descriptor) [49], and SIL (Silhouette-based descriptor) [49];
- volume or surface based EDT (negatively exponentiated Euclidean Distance Transform) [27] and RSH (Ray-based with Spherical Harmonic representation) [49].

With these representations, the 3D object retrieval process is carried out by calculating the distances or similarities between the query and objects in the database, and then ranking portions of the database in terms of the similarities to the query. Since most shape representations can be regarded as a feature vector or histogram, the similarity measures are often $L_p$ norms, $\chi^2$ distance,

or symmetric KL divergence. Graph match is also applied for some topological representation of an object.

### 2.1.4 Performance Evaluation

It is necessary to define some performance evaluations to compare the effectiveness of different 3D object retrieval algorithms. Suppose that we have a test set $\mathbf{S}_t$ and each object $Q_i \in \mathbf{S}_t$ $(i = 1, 2, \cdots, N)$ is used as the query. The retrieved objects $(O_{i1}, O_{i2}, \cdots)$ are ranked by their similarities to $Q_i$. Objects in the database that belong to the same class as $Q_i$ constitute a set of $\mathbf{C}_i$. The performance of a retrieval algorithm is then evaluated by the average results of all the queries.

In this thesis, a Precision-Recall plot and some overall scores [40] are taken to assess different retrieval algorithms.

**Precision-Recall Plot** describes the relationship between precision $P_i$ (vertical axis) and recall $R_i$ (horizontal axis) for query $Q_i$ in the ranked list $(O_{i1}, O_{i2}, \cdots)$, where

$$P_i(k) = \frac{|\{O_{ij}|O_{ij} \in \mathbf{C}_i, j = 1, 2, \cdots, k\}|}{k}, \tag{2.1}$$

$$R_i(k) = \frac{|\{O_{ij}|O_{ij} \in \mathbf{C}_i, j = 1, 2, \cdots, k\}|}{|\mathbf{C}_i|}. \tag{2.2}$$

Based on the above definition, we can obtain a sequence of precisions and recalls by increasing $k$, and hence we can draw a Precision-Recall plot for each query $Q_i$. Usually the average plot of all queries in $\mathbf{S}_t$ is reported as the performance measure of an algorithm. An ideal result would generate a horizontal line at $P = 1$.

**Nearest Neighbor** is the percentage of the first objects in the ranking lists that belong to the same class as the queries:

$$NN = \frac{|\{O_{i1}|O_{i1} \in \mathbf{C}_i, i = 1, 2, \cdots N\}|}{|\mathbf{S}_t|}. \tag{2.3}$$

Obviously, the ideal score is 100%, the range of $NN$ is $[0, 1]$, and the higher $NN$ is the better the results are.

**First Tier** of a query $Q_i$ is defined as the percentage of objects in $Q_i$'s class $\mathbf{C}_i$ that appear within the top $k$ matches:

$$FT_i = \frac{|\{O_{ij}|O_{ij} \in \mathbf{C}_i, j = 1, 2, \cdots, k\}|}{k}, \tag{2.4}$$

$$FT = \frac{1}{N} \cdot \sum_{i=1}^{N} FT_i. \tag{2.5}$$

where $k$ depends on the size of the class. If $\mathbf{C}_i$ contains $Q_i$, $k = |\mathbf{C}_i| - 1$; otherwise $k = |\mathbf{C}_i|$. We can see that $FT$ is also within the range of $[0, 1]$ and the higher the better.

**Second Tier** of a query $Q_i$ is defined in the same way as Equation 2.4. The only difference is that $k = 2 * (|\mathbf{C}_i| - 1)$ when $Q_i$ is in $\mathbf{C}_i$, and $k = 2 * \mathbf{C}_i$ when $Q_i$ is not included by $\mathbf{C}_i$.

**E-Measure** is a composite measure of precision and recall for a fixed number $k$ of retrieved results ($k = 32$ in this thesis):

$$E = \frac{1}{1/P + 1/R}. \tag{2.6}$$

It assumes that a user is more interested in the first page of query results than in later pages. The maximal value of $E$ is 1 and the higher the better.

**Discounted Cumulative Gain** is proposed in [26] and widely used in information retrieval as an evaluation criterion. Its intuition is that the retrieved relevant results are more important near the front of the list than those at the later. The specific definition is by firstly convert the retrieved object list $(O_{i1}, O_{i2}, \cdots)$ to a boolean sequence $G_i = (G_{i1}, G_{i2}, \cdots)$, where $G_{ij} = 1$ if $O_{ij} \in \mathbf{C}_i$ and $G_{ij} = 0$ otherwise. The Discounted

Cumulative Gain (DCG) series is then defined based on the boolean list $G$,

$$DCG_{ij} = \begin{cases} G_{i1} & j = 1, \\ DCG_{ij-1} + G_{ij}/\log_2(i) & j \geq 2. \end{cases} \tag{2.7}$$

The DCG is then defined as below,

$$DCG_i = \frac{DCG_{iM}}{1 + \sum_{j=2}^{|\mathbf{C}_i|} \frac{1}{\log_2(j)}}, \tag{2.8}$$

where $M$ is the length of the list $G$. The denominator denotes the perfect $DCG_i$ value when all objects that are within the same class $\mathbf{C}_i$ as $Q_i$ are ranked at the top $|\mathbf{C}_i|$ positions. Note that we also use the average value of all queries' $DCG_i$ ($i = 1, 2, \cdots, N$) as the final evaluation score. Higher $DCG$ indicates a better retrieval algorithm.

The Precision-Recall plot provides a visualized comparison of different algorithms. Near Neighbor predicts how well a retrieval algorithm in retrieving a most similar object to the query. First Tier, Second Tier, and E-Measure evaluate the correct results near the front of the retrieved and ranked list, while DCG assesses the whole list with some discounted scheme. These performance measures provide a complete assess of a retrieval algorithm with respect to different emphasis.

## 2.2 Public Databases

Some 3D shape benchmarks have been released to promote standard comparison of different competitive methods. Based on the characteristics of objects they contain, we categorize them into databases of generic, articulated, and domain-specific shape models.

### 2.2.1 Databases of Generic 3D Objects

Among all the available databases, those containing generic objects are the most influent ones. By generic objects we mean that in these databases there are both natural and man-made objects, both rigid and articulated objects, both whole and partial objects, etc. Besides, these objects are often with different "defined levels"—some objects are strict closed meshes (watertight) while others may be not; some objects are represented very accurately while others may be represented by very coarse tessellation.

Princeton Shape Benchmark (PSB) [40] is a representative one of such type of databases. It contains a database of 3D polygonal models collected from the World Wide Web. For each 3D model, there is an Object File Format (.off) file with the polygonal geometry of the model, a model information file (e.g., the URL from where it came), and a JPEG image file with a thumbnail view of the model. It contains 1,814 models in total and is declared that more models would be added in the future. These models are divided into training and test sets equally, each with 907 models. The training set is classified into 90 classes in the base classification, while the test set consists of 92 classes. An interesting issue is that PSB also provides some coarser classification criteria. For example, in the third level of categorization, the objects are categorized as either natural or man-made class. To the best of our knowledge, we find no work probing the coarser classification criteria. We have carried out some experiments about this issue which will be reported in the next chapter.

There are also some other generic databases, like NTU 3D Model Benchmark [15], CCCC database [48], a database used in [5], and the ITI database [3].

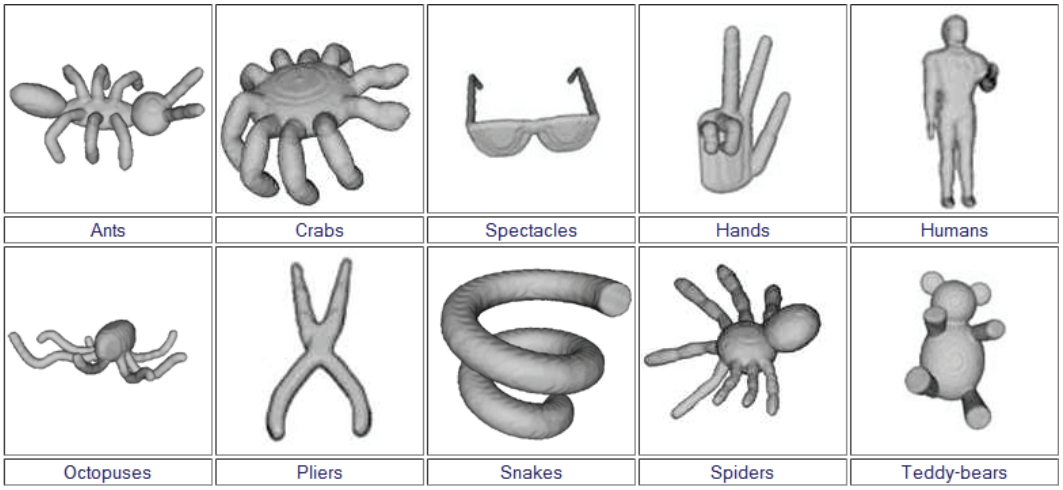



Figure 2.3: Objects in the McGill 3D Shape Benchmark with articulated parts.

### 2.2.2 A Database of Articulated Objects

Considering that the techniques to deal with rigid objects and articulated ones may be quite different, a database called McGill 3D Shape Benchmark [54] is designed for articulated 3D shape models on purpose. It is composed of 254 models lying in 10 categories, each with about 20 to 30 objects. Both voxel and polygon mesh forms are provided, and the meshes are all watertight. Figure 2.3 shows some example objects with articulated parts in the database. Note that another database full of objects with no or minor articulated parts, which may be regard as the generic database under our categorizing criterion in this thesis, is also released along with the articulated one.

We have proposed a novel shape descriptor particularly suitable for articulated object match [21] and will present the details in the next chapter. Our algorithm is also able to boost existing algorithm in retrieving generic objects.

### 2.2.3 Domain-Specific Databases

There are also some 3D databases designed for applications in some specific domains, such as CAD [37] and biological data [10]. An engineering shape

benchmark has been provided at Purdue University [23]. These databases are provided to facilitate research for some more specific problems and hence have their own characteristics on both the problems themselves and corresponding methods evolved.

### 2.2.4 Data Sets for the Shrec Contest

There has been an annual contest, the 3D Shape Retrieval Contest (Shrec) since 2007 [4]. It is organized within the AIMSHAPE project and usually in conjunction with the Eurographics Workshop on 3D Object Retrieval. The 3D object retrieval problem is studied in a finer level in this contest. It divides the problem into different tracks like Protein models, non-rigid shapes, range scans, CAD models, watertight models, and even a track of 3D face models. Besides the above mentioned databases, the contest also contributes its own data sets to facilitate related research.

## 2.3 Experimental Systems

After years of development, some experimental 3D search engines have been released:

- 3D Model Search Engine at Princeton University [6, 28];
- 3D Model Retrieval System built in National Taiwai University [1, 14];
- 3D Model Retrieval System at the University of Konstanz [2, 48];
- 3D Search Engine at the Informatics and Telematics Institute, Greece [3];
- 3D Shape Retrieval Engine [5, 45].

All these systems are developed experimentally based on one or more publicly available databases with a limit number of objects. Though they are in their infancies, it is promising for true applications where real-time processing and friendly user interface are required.

## 2.4 Challenges in 3D Object Retrieval

As a new type of multimedia information representation, 3D objects have their own characteristics and hence leads to some new challenges to the store, retrieval, analysis, and visualization. Comparing to other multimedia retrieval tasks, a 3D object retrieval system may have more challenges to overcome.

- Similar to other feature extraction criteria for traditional multimedia, shape descriptors ought to be as discriminative as possible. A good 3D shape descriptor should be expressive for a particular object, stable for a class of similar objects, and distinctive for objects of different categories.
- A 3D object retrieval system should be robust to objects' linear and non-linear transformation, deformation, and degeneration. This may be fulfilled by either canonical coordinate normalization, or some shape descriptors that are invariant to these influences, or both.
- Unlike text or image retrieval, a user-friendly query formulation and interface become important for a 3D object retrieval system, because users are often short of such data as query, and it is difficult for users to create a 3D object using current software and tools.
- Diversified representation of 3D objects may impede the applicable scope of a system. As introduced in the previous chapter, 3D objects may be represented by voxel, mesh, and range data. Some of them may also include appearance information in addition to the shapes. It is not a trivial task for a system to deal with such data simultaneously.

- Various tessellations or resolutions of 3D objects result in different levels of details of 3D objects. Descriptors that are able to describe an object in multi-resolution are promising for true application.

- "Mesh soup" is a popular and convenient way to represent and store an object. However, whether a mesh is closed or not may cause large changes of a shape descriptor. In other words, shape descriptors should not be restrictive to some particular structures of the mesh soup.

- Efficient descriptor extraction, compact representation, and real-time indexing are always what a retrieval system must satisfy.

# Chapter 3

# Boosting 3D Object Retrieval by Object Flexibility

This chapter elaborates our proposed 3D object flexibility feature and a shape descriptor based on flexibility using the bag-of-words model. We will show that this descriptor is stable to the deformation of objects' articulations, in addition to commonly concerned linear transforms, i.e., translation, scale, and rotation. Experiments demonstrate its superiority over a variety of existing 3D shape descriptors in the retrieval of articulated objects, as well as its enhancement of other shape descriptors to retrieve generic 3D objects.

## 3.1 Related Work

3D data are now widely recognized as the upcoming wave of digital media. 3D object retrieval rapidly becomes a key issue in this new multimedia content processing, and attracts more and more research interests. Some 3D object benchmarks and experimental retrieval systems have been made available, such as the Princeton shape benchmark and its associated search engine [40], and the NTU 3D model benchmark and its corresponding retrieval system [15]. The reader is referred to [46, 12, 52] for a comprehensive survey of this research.

The shape of a 3D object can be with arbitrary scale, location, and orientation. For generic 3D object retrieval, a retrieval method has to either perform a pose normalization process or use shape descriptors that are inherently invariant to linear transformations (translation, rotation, and scale). Much work has been presented in solving these problems so far. Shilane et al. compared twelve shape descriptors in [40]. Bimbo and Pala included five descriptors in their experimental analysis, which fall into point distribution based, volume based, and image based categories [16].

On the other hand, less attention has been paid to the problem of non-linear shape deformations caused by objects' articulations. Zhang et al. carried out the first work of 3D articulated object retrieval using medial surfaces and their graph spectra, and provided a 3D articulated object database, the McGill 3D shape benchmark [54]. The main problem in the graph-based method is that it is sensitive to topological changes which are common in generic 3D models. Jain and Zhang tried to achieve articulation invariance in [25] by using the spectral embedding of an affinity matrix. Ion et al. used the continuous eccentricity transform to make their method insensitive to shape articulations [22]. However, these two methods [25, 22] are sensitive to objects with disconnected parts or outliers.

In this chapter, we propose a novel feature, called *object flexibility*, at a point of a 3D object to describe how the neighborhood of this point is massively connected to the object. This feature is stable to both linear transformations and non-linear deformations caused by objects' articulations. Based on this object flexibility, we propose a new shape descriptor for 3D object retrieval. Extensive experiments show that it outperforms a variety of existing shape descriptors in the retrieval of articulated 3D objects, which are often natural objects like animals, plants, and humans. Besides, combined with existing shape descriptors, it also helps to obtain better performance of retrieving generic 3D objects.

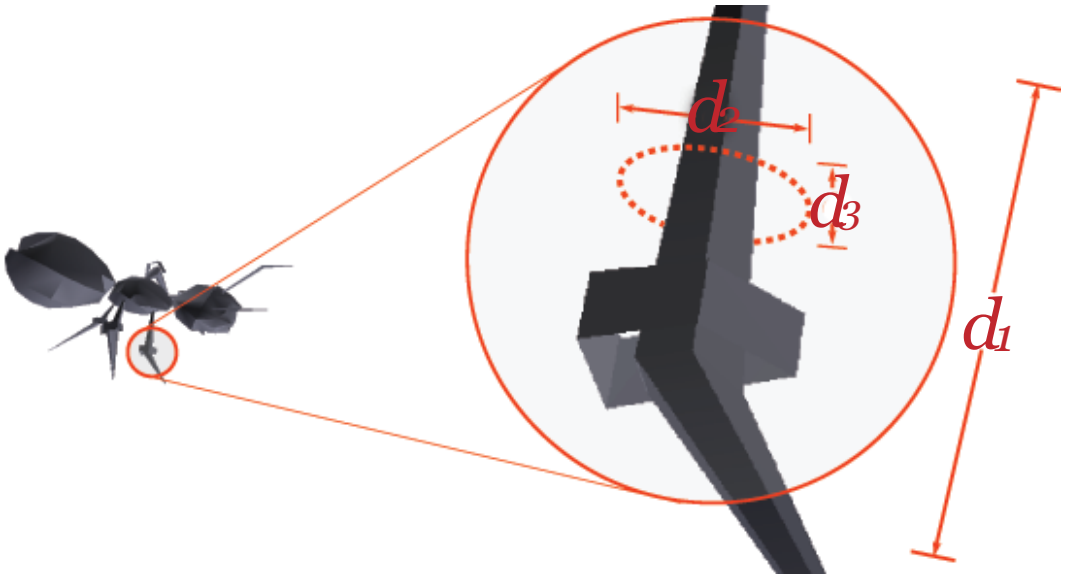



Figure 3.1: Illustration of the flexibility.

## 3.2 Object Flexibility

In this section, we first define the object flexibility mathematically, and then we discuss how to compute it and to form the final shape descriptor based on it.

### 3.2.1 Definition

**Definition 3.1.** *Given a radius $r$, let $C_{p,r} \subset \mathcal{O}$ be the set of points within a sphere $S_r^3$ centered at a point $p$ of a 3D object $\mathcal{O}$. The object flexibility at $p$ is defined as:*

$$\rho_r(p) = \frac{Eig_2(X^T X)}{r}, \tag{3.1}$$

*where $X$ is the data matrix consisting of the 3D coordinates of all the points in $C_{p,r}$, one point per row, and $Eig_2(\cdot)$ is a function that returns the second largest eigenvalue of a square matrix.*

Consider a small part enclosed by $S_r^3$, centered at a point $p$ on the leg of an ant in Fig. 3.1. Suppose that the three dimensions denoted by $d_1$, $d_2$, and $d_3$

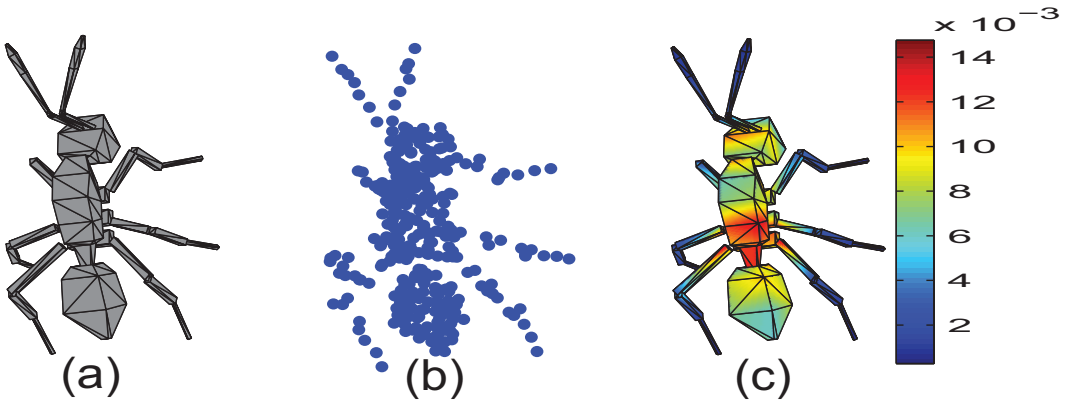



Figure 3.2: (a) A 3D model of an ant. (b) Sampled points of the ant. (c) The flexibility distribution on the ant.

are arranged as $d_1 \geq d_2 \geq d_3$ according to the variance on each dimension. In the definition, we estimate the normalized variance in the direction $d_2$, which can be a potential measure about how the local part (a segment of the ant's leg) is massively connected to the object (the ant), or how easily that part of the object can be bent. The smaller $\rho_r(p)$ is, the more tenuous that part is with more "bending ability".

Note that we do not use the variance in $d_1$ because it tends to be constant for different parts of an object when $r$ is fixed. We also discard the third eigenvalue since it may degenerate to zero when the points in $C_{p,r}$ lie in a plane.

### 3.2.2 Computation of the Flexibility

Before computing the flexibility, we need to determine the radius $r$ and the points of an object where their flexibilities are computed.

Since the flexibility describes local shape characteristics, we have to select enough points of a 3D model to obtain a complete shape description. For an object represented by voxels, we select all its surface points, each of which has less than 13 non-zero voxels among its 26 neighbor voxels. In our experiments,

590 surface points of a 3D model are left on average after filtering out inner points in the McGill database [54], where each model is represented by $128^3$ voxels. For an object represented by meshes in the Princeton shape benchmark [40], 2000 surface points are sampled using a scheme presented in [31] in the first round, and then 500 points are randomly selected from them. We use all the 2000 points to compute the flexibilities of the 500 points. One example of the sampled points from a mesh model is shown in Fig. 3.2(b).

Next, a proper radius $r$ is determined for each selected point. It is easy to see that

$$\lim_{r\to 0}\rho_r(p) = 0, \quad \lim_{r\to\infty}\rho_r(p) = 0, \tag{3.2}$$

for $p \in \mathcal{O}$, which further result in

$$\lim_{r\to 0}\eta(r) = 0, \quad \lim_{r\to\infty}\eta(r) = 0, \tag{3.3}$$

where

$$\eta(r) = Var_{p\in\mathcal{O}}(\rho_r(p)). \tag{3.4}$$

Therefore, there exists a $r^*$ such that the variance $\eta(r^*)$ is maximized. The goal of maximizing $\eta(r)$ is to obtain the richest descriptor on flexibility for a given 3D model.

Fig. 3.3(a) shows the flexibility variances $\eta(r)$ versus different radii, $0.1R_i$, $0.2R_i$, $\cdots$, $1.0R_i$, $1 \le i \le 10$, for 10 randomly selected objects (dashed lines), where $R_i$ is the radius of the $i$-th 3D model defined as the average distance from all the surface points to the center of mass. The solid curve in Fig. 3.3(a) denotes the average variances of 100 randomly selected objects. This figure shows that the maxima of the curves mainly fall into a relatively small range of $r$, indicating that the discriminative ability of $\rho_r(p)$ is insensitive to the choice of $r$. In practice, one can compute the flexibility of a point $p$ using some $r \in [0.2R, 0.4R]$. Fig. 3.2(c) shows the flexibility distribution of an ant computed using $r = 0.3R$. Alternatively, we can define a more elaborate

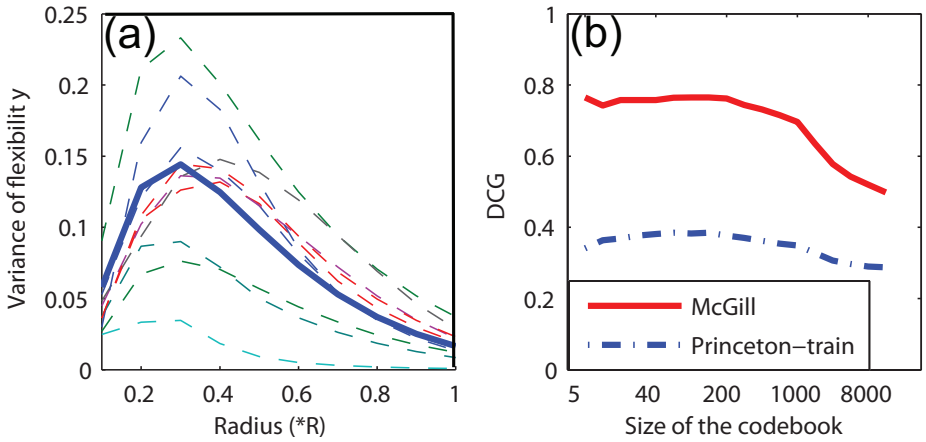



Figure 3.3: (a) Flexibility variance versus radius. (b) DCG versus size of the codebook.

measure to explore the flexibility characteristic thoroughly by using a series of radii to form a *flexibility vector* at a point. We discuss it in the next section.

## 3.3 A Flexibility Descriptor for 3D Object Retrieval

Let $\mathbf{r} = [r_1, r_2, \cdots, r_K]^T$, $r_1 < r_2 < \cdots < r_K$, denote the radii of a series of concentric spheres centered at some point $p$. A flexibility vector $\boldsymbol{\rho}_{\mathbf{r}}(p)$ at $p$ is then obtained by

$$\boldsymbol{\rho}_{\mathbf{r}}(p) = [\rho_{r_1}(p), \rho_{r_2}(p), \cdots, \rho_{r_K}(p)]^T. \tag{3.5}$$

Note that a 3D model generates a number of flexibility vectors and two models usually have different numbers of such vectors. To organize these vectors into a global shape descriptor, the bag-of-words model [18] is adopted here. We use the $k$-means algorithm to cluster the flexibility vectors from half of the objects in a database to $N$ clusters, the centers of which compose a codebook $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$. Each flexibility vector is then represented by a codeword $\mathbf{c}_i$ if it is closest to $\mathbf{c}_i$, and a 3D model is described by a histogram over

$\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$ obtained by counting the codewords representing all the flexibility vectors of the 3D model. Finally, the object flexibility descriptor is formed by normalizing the histogram.

To measure the dissimilarity between two flexibility descriptors $P$ and $Q$, we use the symmetric Kullback–Leibler divergence defined as

$$dis_{Fl}(P||Q) = D_{KL}(P||Q) + D_{KL}(Q||P), \tag{3.6}$$

where

$$D_{KL}(P||Q) = \sum_{i=1}^{N} P(i) \log \frac{P(i)}{Q(i)}. \tag{3.7}$$

With this descriptor, we can conduct 3D object retrieval by computing the dissimilarities between a query and every object in a database. Using every object as the query in the McGill database [54] and the Princeton training set [40], Fig. 3.3(b) shows the average retrieval performance (evaluated by the discounted cumulative gain (DCG) [40]) of the flexibility descriptor. We can see that DCG remains consistent in quite a wide range of the size of the codebook. We choose $N = 10$ for the McGill benchmark and $N = 60$ for the Princeton benchmark in our experiments.

## 3.4 Enhancing Existing Methods

The flexibility descriptor performs very well in retrieving articulated 3D objects (see Section 3.5). In addition, it can enhance existing shape descriptors when combined with them to retrieve generic 3D objects. A majority of existing shape descriptors represent a 3D shape without explicit geometric meanings, while the flexibility descriptor measures a particular geometric characteristic, the flexibility. So a more complete shape description of a 3D model can be obtained by combining it with other descriptors.

A natural way to combine two shape dissimilarity measures is

$$dis = \alpha \cdot dis'_{Fl} + (1 - \alpha) \cdot dis'_{Other}, \tag{3.8}$$

| | **FD** | LFD | SHD | AAD | D2 |
|---|---|---|---|---|---|
| FT | **0.560** | 0.508 | 0.478 | 0.439 | 0.419 |
| ST | **0.720** | 0.697 | 0.641 | 0.624 | 0.605 |
| EM | **0.530** | 0.497 | 0.464 | 0.433 | 0.414 |
| DCG | **0.844** | 0.831 | 0.804 | 0.768 | 0.764 |

Table 3.1: Retrieval performance of different shape descriptors for retrieving 3D articulated objects.

where $dis'_{Fl}$ is a normalized version of $dis_{Fl}$ defined in (3.6) such that it is in $[0, 1]$, $dis'_{Other}$ is some other dissimilarity measure, also normalized into $[0, 1]$, and $\alpha$ is a weighting factor to balance the two measures with $0 \leq \alpha \leq 1$.

## 3.5 Experiments

Three groups of experiments are carried out, each with emphasis upon different requirements for a shape descriptor. In these experiments, we compare our flexibility descriptor (FD) with four other shape descriptors, the source codes or executable programs of which have been provided by the authors. They are the point distribution based D2 [31] and one of its extended descriptors, the mutual absolute-angle distance histogram (AAD) [30], the volume based spherical harmonic descriptor (SHD) [27], and the image based light field descriptor (LFD) [15]. The retrieval results are quantified using the Princeton shape benchmark (PSB) evaluation tools of first tier (FT), second tier (ST), e-measure (EM), discounted cumulative gain (DCG), and the precision-recall plot [40].

### 3.5.1 Retrieving Articulated Objects

The first group of experiments is designed to measure the robustness of the shape descriptors to nonlinear shape transformations caused by objects' articulations, with the McGill articulated shape database. This database consists of 255 models in 10 classes. Every model is used as the query. Table 3.1 shows

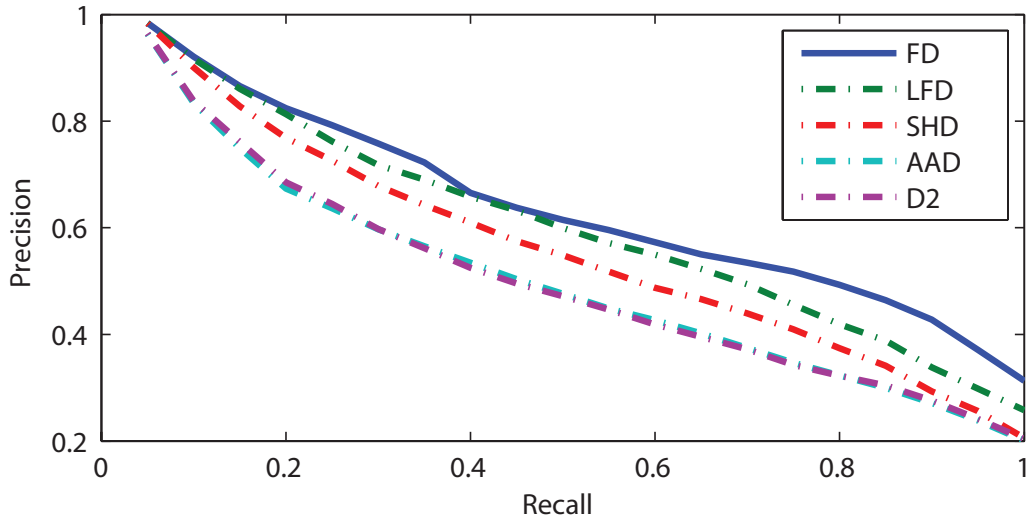



Figure 3.4: Precision-recall plots of different shape descriptors for retrieving 3D articulated objects.

the average retrieval results of the five descriptors evaluated by FT, ST, EM, and DCG. Fig. 3.4 is the precision-recall plots of the five descriptors. We can see that our FD outperforms the other descriptors.

### 3.5.2 Retrieving Generic Objects

In the second group of experiments, we use the whole McGill shape benchmark (MSB) which includes the 255 articulated objects used in the previous experiments and the other 200 objects with few or no articulations. Our FD itself does not perform very well in retrieving such generic objects. However, significant improvements can be achieved when it is combined with other descriptors using (3.8) ($\alpha = 0.5$ is chosen here). Fig. 3.5 shows the retrieval results of LFD, SHD, AAD, and D2 with and without our FD combined, evaluated by FT, ST, EM, and DCG. With the improved performance, all the four shape descriptors are enhanced by our FD under different evaluations. The improvements are due to the fact that the four shape descriptors describe 3D models from a general viewpoint only, without considering particular geometry properties, but the fusion of them with our FD provides a more complete

description.

The reader may wonder if the combination of two of the previous descriptors can also give similar or even better results than the combination of our FD with one of the previous. Since LFD works best among the previous four descriptors in our experiments as well as in [40] and [16], we use it as the baseline and combine each of the other descriptors with it. The retrieval results are given in Fig. 3.6(a), which are the precision-improvement-recall plots obtained from the precision-recall plots by subtracting the precision of LFD from the precisions of the four combinations. Fig. 3.6(a) indicates that our FD with LFD not only outperforms the other combinations, but also has improvement over the original LFD in a wide range of the recall.

### 3.5.3 Experiments on Larger Databases

The third group of experiments is conducted on the Princeton shape benchmark (PSB) training set [40], which contains 907 models in 90 classes. The majority of the models are rigid, man-made objects without much requirements for a shape descriptor to be articulation invariant. Even though it seems that the articulation invariant property of our FD is not necessary in such a case, it is still able to enhance the other shape descriptors to some extent. Fig. 3.6(b) shows the precision-improvement-recall plots by fusing FD, SHD, AAD, or D2 with LFD tested on the PSB training set. Obviously, LFD combined with FD improves the performance of LFD itself.

More specifically, there are improvements in 72 classes in the PSB training set where LFD with FD performs better than LFD itself. Among them, 17 classes are with more than 5% improvements. These improvements mainly happen to natural objects such as tree, apatosaurus, and face, and objects with large intra-class variances such as city, shoe, rectangular, roman building, and antique car.

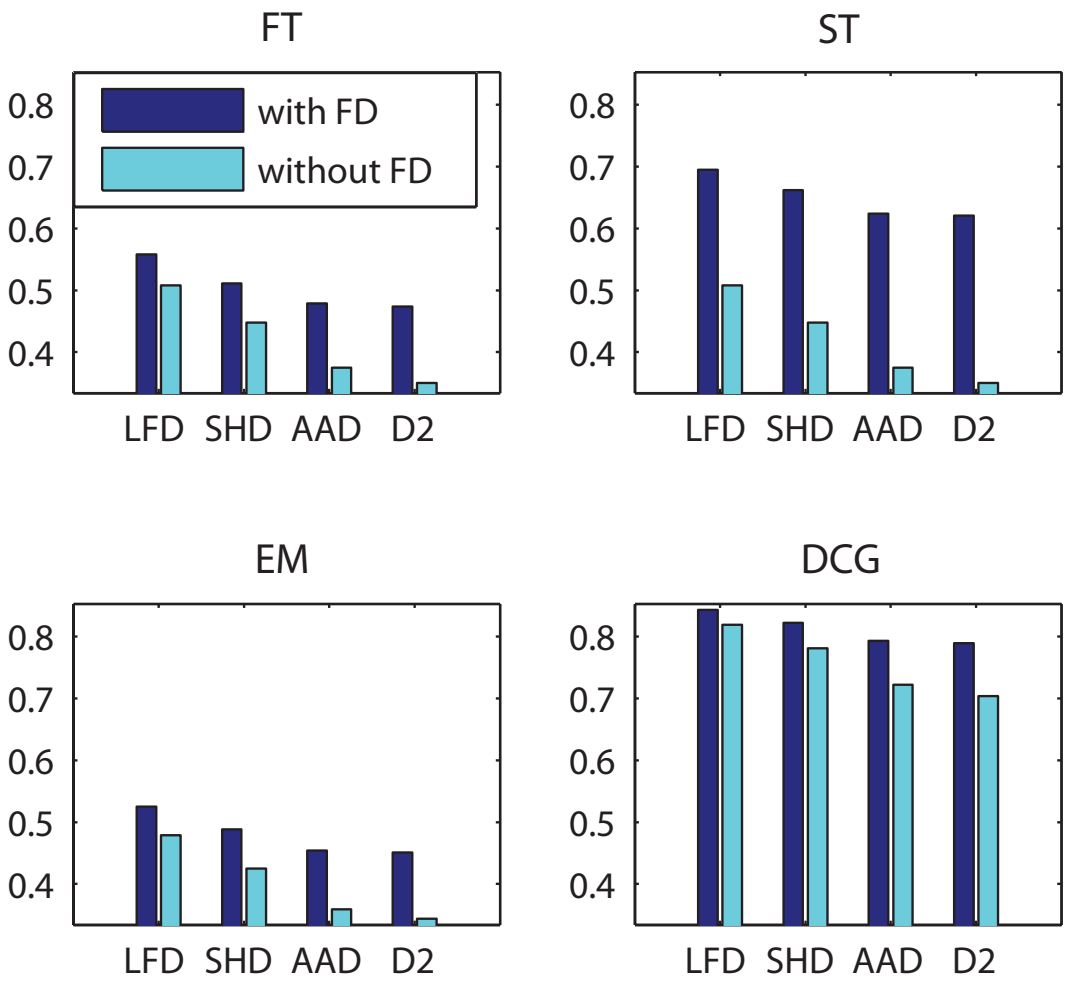



Figure 3.5: Retrieval performance of different shape descriptors with and without FD on MSB.

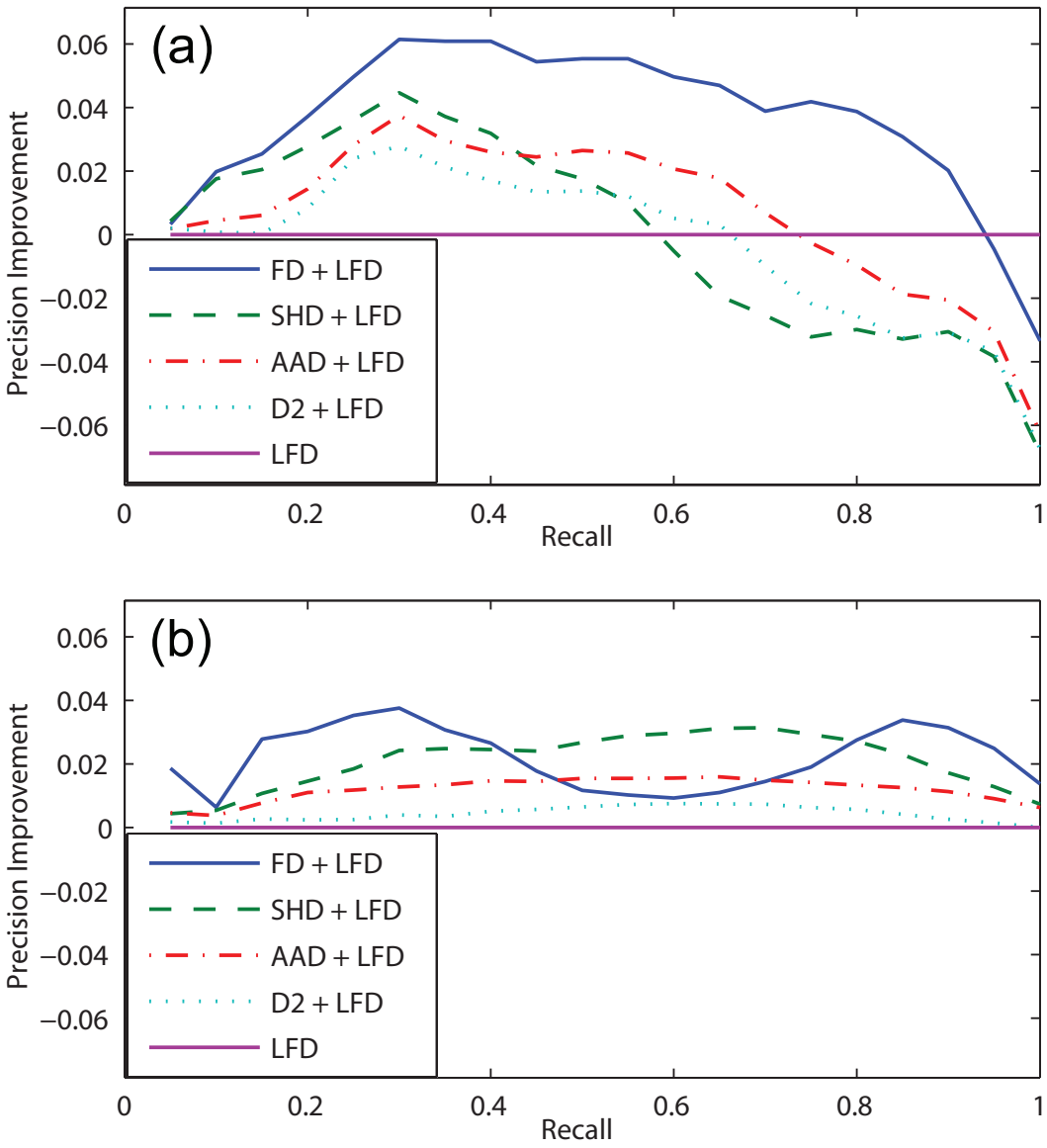



Figure 3.6: Precision improvement comparison when FD, SHD, AAD, or D2 is combined with LFD on MSB (a) and on the PSB training set (b).

The extensive experiments demonstrate that our flexibility descriptor is suited for retrieving articulated objects, especially natural objects. For generic object retrieval, it provides a favorable complementary to other shape descriptors.

### 3.5.4 Comparison of Times for Feature Extraction

The feature extraction times of FD, LFD, SHD, AAD, and D2 are about 1.12s, 2.69s, 2.25s, 0.46s, and 0.23s, respectively, on average for each object in the PSB training set. The programs are run on an Intel Pentium(R) 3.20GHz CPU with 2 GB RAM.

## 3.6 Conclusions & Analysis

We have proposed a new feature, called object flexibility, to measure local shape characteristics of an object about how a local part is massively connected to the object. Based on this feature, a new shape descriptor is obtained, which is stable to shape deformations caused by articulations. Extensive experiments show that this shape descriptor outperforms four previous popular shape descriptors in retrieving articulated objects. For generic 3D object retrieval, it can be combined with them to obtain better performance.

# Chapter 4

# 3D Object Retrieval with Referent Objects

The typical 3D object retrieval framework as shown in chapter 2 assumes no prior knowledge about both the database and the query, but in practice, a relatively size-fixed database can be organized into different clusters, and a query is convenient to be tagged with some labels by its user. This chapter shows some performance improvements by using this prior information, and a learning method to infer priors when they are not available.

## 4.1 3D Object Retrieval with Prior

Traditionally, 3D object retrieval is to compare a query to each object in the database and return a sorted list of objects according to the distances or similarities between them and the query. In other words, the distance $d(Q, O_i)$ between the query $Q$ and an object $O_i$ in the database is obtained by defining some metric on them. Therefore, object descriptor and the metric defined on them are the key factors influencing the final retrieval results.

However, we find that the performance can be improved significantly with a simple modification to this framework based on some prior about the objects. The motivation is that if some objects distinguish from the query quite a lot,

| | NN | FT | ST | EM | DCG |
|---|---|---|---|---|---|
| (EDT, L1) | 0.60 | 0.33 | 0.44 | 0.24 | 0.61 |
| (EDT, Prior, L1) | 0.64 | 0.38 | 0.51 | 0.28 | 0.65 |

Table 4.1: Performance evaluation of EDT on PSB training set with and without prior (the second and third row respectively).

they are likely not what a querier wants and ought to be ranked in the rear of the retrieval list. We demonstrate this by using the prior of whether the query and the object to be compared are natural or man-made. While we compare the query $Q$ and an object $O_i$ in the database, their labels of "natural" or "man-made" are firstly examined before going to the next step of calculating the distance between them. If they are with the same tag, the distance is computed using pre-defined method; and if they are with different labels which means that the query and the object are quite different, the distance is simply set as infinity. Thus the distance measure of $d(Q, O_i)$ is modified as

$$d(Q, O_i, Prior) = \begin{cases} d(Q, O_i) & \text{if } Pr(Q) = Pr(O_i), \\ \infty & \text{if } Pr(Q) \neq Pr(O_i), \end{cases} \tag{4.1}$$

where $Pr(\cdot)$ is the label of "natural" or "man-made" of an object here.

With such weak prior, a distinct improvement can be found in our experiment on the PSB training set using EDT descriptor and L1 metric as the distance measure. The first row in Table 4.1 are five kinds of performance evaluations (we refer readers to [40] for detailed definitions), and the second and third rows are retrieval performances of EDT with and without prior respectively. Figure 4.1 plots the recall-precision curves. We can see that by using the weak prior, one can reach much better retrieval results.

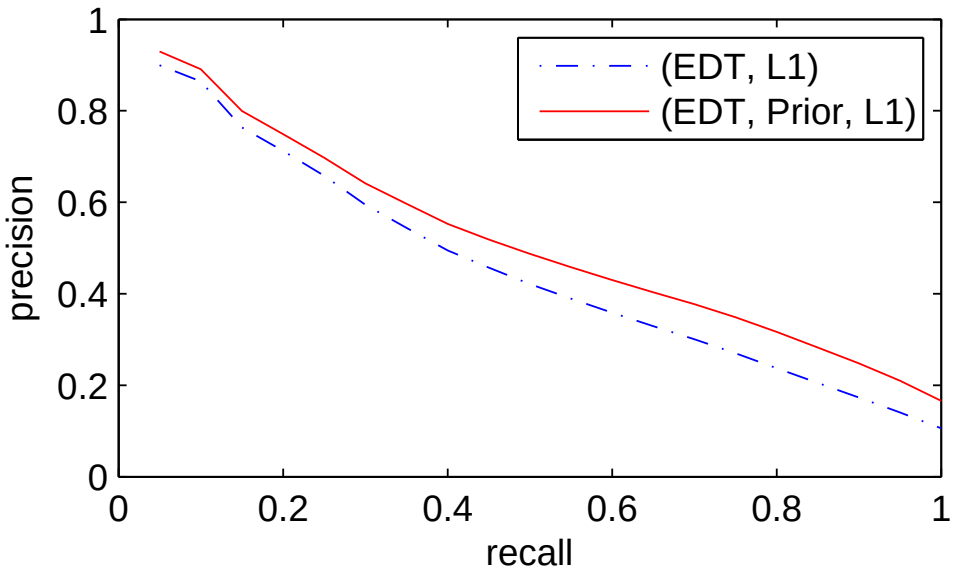



Figure 4.1: Precision ($y$ axis) versus recall ($x$ axis) on PSB training set.

## 4.2 3D Object Retrieval with Referent Objects

Note that the prior used in section 4.1 is provided when the PSB is released, but most objects are not with such labels of "natural" or "man-made" in real applications. We consider two scenarios here. One is that objects in the database are classified into natural or man-made ones, while the label of the query is not known. The other is that both objects to be retrieved and the query are not classified. The former scenario is suitable for applications where the size of the database is relatively fixed and hence some pre-processing to organize the objects in the database is reasonable, but when objects in the database are not classified, the latter scenario applies.

Although in both cases the advantage of object prior can not be used directly, we introduce a 3D object classification module to automatically tag labels to objects to make up this problem.

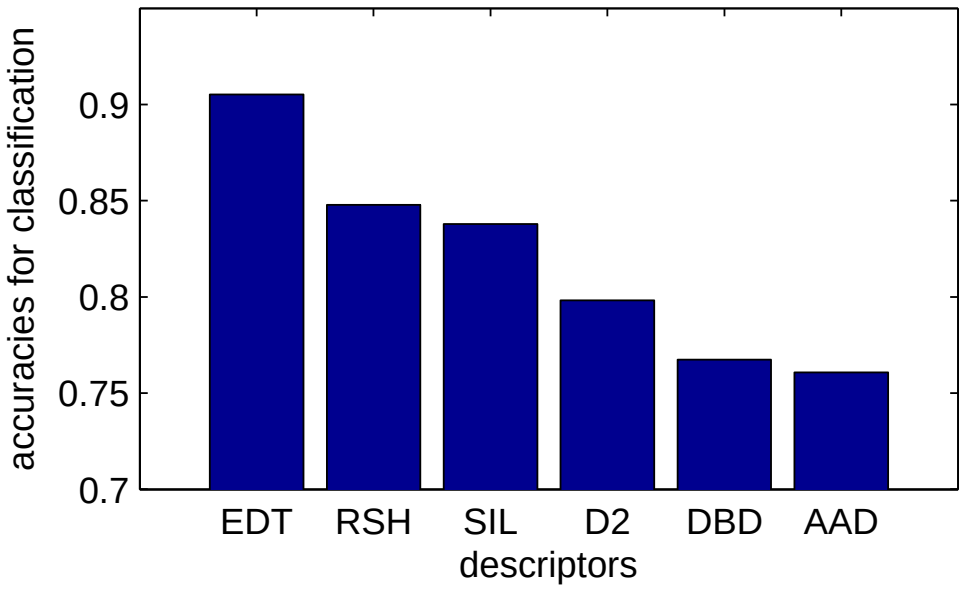



Figure 4.2: Performance comparison of descriptors for natural and man-made 3D object classification.

### 4.2.1 Natural and Man-made 3D Object Classification

We introduce a 3D object classifier to serve as the prior inferring module in the following retrieval algorithm. Though amount of performance comparisons of different 3D object descriptors has been reported before, no previous work can be referred to with respect to the problem of 3D object classification to the best of our knowledge. Therefore, we firstly compare three categories of existing 3D object descriptors in terms of their effectiveness in 3D object classification.

We use 10-fold cross validation to train a SVM classifier $\mathcal{C}$ [13] for natural and man-made objects (i.e., a typical binary classification problem) based on the PSB training set [40]. Six descriptors lying in three categories are chosen and compared for this purpose, i.e., point distribution based D2 [31] and AAD [30], view projection based DBD and SIL [49], and 3D space structure based RSH [49] and EDT [27].

From the observation of Figure 4.2, EDT outperforms the other descriptors

significantly for the 3D object classification problem, and we use EDT and its corresponding classifier trained on the PSB training set in the following experiments. Note that the PSB training set are taken as the referent objects here. Some more objects can be added.

Besides, an interesting byproduct worth pointing out is that, within the area of 3D natural and man-made object classification, space structure based methods (EDT and RSH) works better than view projection based ones (SIL and DBD), while point distribution based ones (D2 and AAD) are the worst. In our another experiment which is not reported in this chapter, we find that for 3D object retrieval problem, view projection based methods are better than space structure based ones and point distribution based ones are also the worst. Therefore, we may make an incomplete conclusion that descriptors extracted from the 3D object structure are the most expressive and robust representation of the objects and can be used in different applications with fairly good performance.

### 4.2.2 Inferring Priors Using 3D Object Classifier

Motivated by the distinct results in section 4.1, we propose to incorporate the outputs of the classifier $\mathcal{C}$ to have a refined distance modification for 3D object retrieval. For example, when compare the similarity of the query $Q$ and an object $O_i$ in the database, the distance between them may be defined as below,

$$d(Q, O_i, \mathcal{C}) = e^{-v_1 v_2} \cdot d(Q, O_i), \tag{4.2}$$

where $\mathcal{C}$ is the classifier learned in Section 4.2.1, $v_1$ and $v_2$ are the decision values for the query $Q$ and object $O_i$ respectively outputted by the classifier $\mathcal{C}$, and $d(Q, O_i)$ is the original distance defined based on the pure shape descriptors without any prior. Note that decision values are positive for natural objects and are negative for man-made objects.

| | NN | FT | ST | EM | DCG |
|---|---|---|---|---|---|
| (EDT, L1) | 0.57 | 0.318 | 0.423 | 0.245 | 0.589 |
| (EDT, $\mathcal{C}$, L1) | 0.577 | 0.330 | 0.437 | 0.253 | 0.596 |

Table 4.2: Performance evaluation of EDT on PSB test set with and without learned prior by SVM classifier $\mathcal{C}$. (the second and third row respectively).

Recall the two scenarios mentioned above. If objects in the database are not classified, both labels as well as decision values of the query and the object have to be predicted by the classifier $\mathcal{C}$. If objects in the database are classified in advance, only the query is necessary to be predicted. In the latter case, decision value $v_2$ of object $O_i$ is simply set as $+1$ (natural object) or $-1$ (man-made object).

However, experimental verification tells that this modification (Equation 4.2) performs bad. The main reason is that the predicted decision values are not quite right. Some false positives and false negatives generate noisy decision values.

### 4.2.3 Reducing False Positives

In our experiments, we find that false positives of the output of classifier $\mathcal{C}$ have a tremendously negative influence on the retrieval results. Therefore, a probability threshold $P_t$ is used to reduce the false positives, and the distance measure is modified as below,

$$d(Q, O_i, \mathcal{C}) = \begin{cases} \infty & \text{if } \mathcal{C}(Q) \neq \mathcal{C}(O_i) \ \& \ P(\mathcal{C}(O_i)) > P_t, \\ d(Q, O_i) & \text{otherwise,} \end{cases} \tag{4.3}$$

where $P_t = 0.95$ is used in the following experiments, and $\mathcal{C}(Q)$ and $\mathcal{C}(O_i)$ are the predicted labels of $Q$ and $O_i$ by the classifier $\mathcal{C}$ respectively.

Applying Equation 4.3 to the EDT descriptor on PSB test data set, similar performance enhancement to section 4.1 is observed (Table 4.2).

## 4.3 Conclusion and Future Work

This chapter demonstrated the effect of a weak prior, whether an object is man-made or natural, for 3D object retrieval. Motivated by such a big improvement based on the weak prior, we intended to propose a learning method to infer the prior. The SVM classifier was suitable for this task and similar improvement was observed. In the future, we will try to find some more elaborate methods to predict the weak prior. Besides, some more priors may be learned from the referent objects to boost the 3D object retrieval.

# Chapter 5

# 3D Facial Expression Recognition

Facial expression recognition has many applications in multimedia processing, and the development of 3D data acquisition techniques make it possible to identify expressions using 3D shape information. In this chapter, we propose an automatic facial expression recognition approach based on a single 3D face.

Our method firstly builds a reference face for each input 3D non-neutral face by a learning method, which well represents the basic facial shape component (BFSC). Then, based on the BFSC and the original expressional face, a facial expression descriptor is designed. Surface depth changes are considered in the descriptor. Finally, the descriptor is input into an Support Vector Machine (SVM) to recognize the type of expression.

## 5.1 Introduction

Automatic facial expression recognition is an important research topic with many applications. In human-computer interface (HCI) community, affective computing employs human emotion to build more flexible and natural multimodal systems [24]. In face recognition, researchers have to pay great attention to handling the effect of expressions [41]. In 2D or 3D face retrieval, automatic

facial expression recognition can serve as a particular kind of feature or a re-ranking algorithm to provide more accurate retrieval results. Moreover, 3D face models have been one of the five tracks of 3D Shape Retrieval Contest (SHREC) since 2007 [7].

In the past two decades, many efforts have been paid on 2D expression recognition [17, 34]. However, since 2D facial images are essentially projections of 3D human faces, facial expression recognition techniques based on them suffer from pose and illumination variations. With the rapid development and the dropping cost of 3D digital acquisition devices, 3D face data, which represent faces as 3D point sets or range data, can be captured more quickly and accurately. Several public 3D face databases have been available now [53, 36]. 3D face data contain explicit 3D geometry, so more clues can be used to encode data changes caused by expressions and handle the variations of face poses. Thus, the use of 3D information in facial expression recognition has attracted attention and some techniques have been presented in recent years [50, 42, 44].

Based on the observation that 3D surface features represent intrinsic facial surface structures associated with specific facial expressions, Wang et al. [50] proposed a primitive surface feature stemming from two surface geometric features, curvature and gradient. They partitioned a 3D face into seven regions guided by the neuro-anatomy knowledge and obtained the statistical primitive feature distribution in each region. They showed that their algorithm is better than two 2D appearance based methods using Gabor wavelets and topographic context. Note that their partitioned regions do not contain the mouth and eyes, which are often used as expressional regions in 2D image based methods [33], and their algorithm involves manually labeled feature points in order to obtain more accurate region partitions.

Soyel and Demirel [42] represented different facial actions by six characteristic distances using eleven manually labeled feature points. They compared

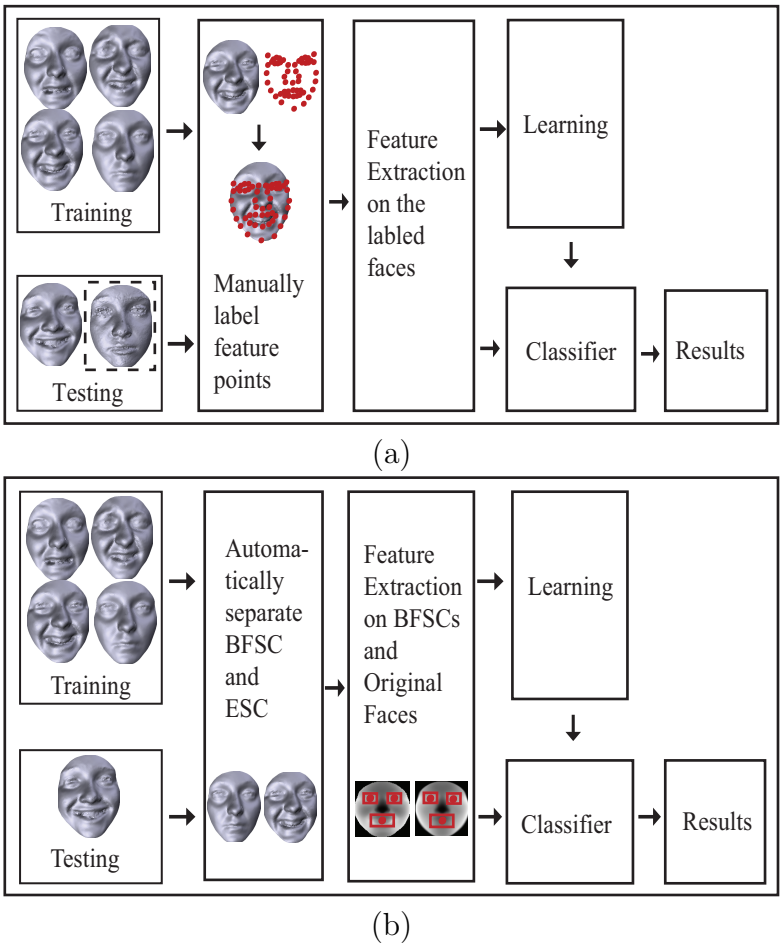



Figure 5.1: Comparison between previous framework (a) and ours (b) for 3D facial expression recognition. In the previous methods, a manually labeling process is required to determine the fiducial points on a 3D face model. Besides, some expression recognition algorithm also requires a neutral face together with the expressional face to be analyzed. In contrast, our algorithm is able to automatically align faces to a canonical coordinate frame, and the neutral face is synthesized as a basic facial shape component (BFSC) through a learning method.

their 3D distance vector based facial expression recognition (3D-DVFER) algorithm to the 2D appearance based Gabor-wavelet algorithm. The experiments showed the superiority of their 3D-DVFER. Tang and Huang [44] further explored the effect of the distance vectors using more manually labeled feature points. They presented both manually designed and automatically selected distance vectors using a feature selection algorithm based on Kullback-Leibler divergence. To achieve the person-independent requirement, Soyel and Demirel [42] normalized the distance vector of an expressional face by the width of the face, while Tang and Huang [44] normalized the distance vector of an expressional face by facial animation parameter units (FAPUs) in its corresponding neutral face as guided by the MPEG-4 standard.

The above methods have demonstrated the superiority of 3D face based methods over 2D image based ones. The main drawbacks of these methods are that they all need manually labeled facial key points for facial expression recognition in both training and testing processes. In addition, [44] also needs the help of neutral faces. These drawbacks make these methods can only work under some constrained conditions.

This chapter proposes an automatic facial expression recognition approach based on a single 3D face. An expressional 3D facial surface is assumed as an approximate sum of two parts. One is a basic facial shape structure which contains little information of expressions and is commonly person-specific. The other is expressional shape component (ESC) which includes rich information about expressions. The ESC is expression-specific, and thus ESCs caused by similar expressions are also similar among a large range of different facial samples. The basic facial shape component (BFSC) is estimated from a group of aligned training data and the input expressional face. After that, based on expressional regions of the BFSC and the original expressional face, the shape depth information is encoded as expression descriptors which are used for a Support Vector Machine (SVM) for classification. The whole framework of

our method is shown in Fig. 5.1(b), together with previous one for comparison in Fig. 5.1(a). It not only performs better than previous ones, but also is independent of the manual labeling of facial feature points. To the best of our knowledge, this is the first 3D facial expression recognition algorithm without the need of the manually labeling process.

## 5.2 Separation of BFSC and ESC

The ESC of an expressional 3D face is the surface deformation of the basic face shape, e.g., the neutral face. Since neutral faces are not always available in expression recognition problems, we propose a learning-based method to estimate the basic face shape of an input expressional 3D face. The estimation uses the information of a group of neutral faces and the input expressional face. Then, the ESC is separated by subtracting the basic face shape from the original expressional face. In addition, before estimations of the basic faces, all 3D face samples are put in a standard coordinate system by an automatic alignment method.

### 5.2.1 3D Face Alignment

We use the same preprocessing as Wang et al.'s in [51] to align every face in a standard coordinate system. Let $S$ denote the point set of a 3D face, and $S^m$ be its mirror set with respect to some plane $E_m$. Then the ICP algorithm [11] is adopted to register $S^m$ to $S$. A facial symmetry plane can be defined as the fitting plane of the midpoint set $B = \{p_i^b | p_i^b = (p_i + p_i^{m\prime})/2, p_i \in S, p_i^{m\prime} \in S^{m\prime}\}$, where $S^{m\prime}$ is the corresponding point set of $S^m$ after the registration. Based on this symmetry plane, the central profile is found and two key points, the nose tip and the top of nose bridge, are robustly extracted on the profile. we can define a standard coordinate frame, in which the origin is the nose tip and the three axes are placed as shown in Fig. 5.2. The detail can be found in [51].

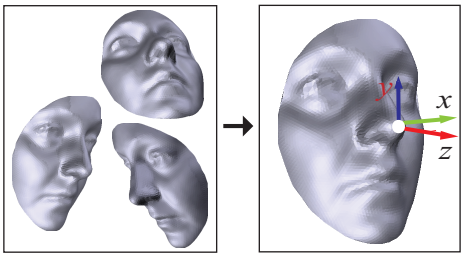



Figure 5.2: 3D face alignment in the standard coordinate frame.

After the alignment, a 3D face can be represented by a depth image, which is obtained by sampling the projection of the 3D face on the $x$-$y$ plane with a size of 100×100 in our experiments.

This process allows us to estimate the basic face structure from its corresponding expressional face placing in a standard coordinate frame, which is described in the next subsection. Furthermore, it makes our method without the need of manually labeling the feature points on the faces.

### 5.2.2 Estimation of BFSC

Our BFSC estimation is based on the assumption that given sufficient training samples, a new face can be recovered approximately by the linear combination of the training faces. Suppose that each depth image is represented by a vector $\mathbf{x} \in \mathcal{R}^N$ and there are $M$ training samples $\{\mathbf{x}_i\}_{i=1}^M$ which are all neutral faces. Then we approximate the basic face $\hat{\mathbf{x}}_e$ of an expressional face $\mathbf{x}_e$ by the linear combination of the training samples:

$$\hat{\mathbf{x}}_e \approx \sum_i c_i \mathbf{x}_i = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M][c_1, c_2, \cdots, c_M]^T = X\mathbf{c}, \tag{5.1}$$

where $\mathbf{c} = [c_1, c_2, \cdots, c_M]^T$ and $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M]$.

One possible solution to determining the weights $\{c_i\}_{i=1}^M$ is to use the discrete Karhunen-Loeve transform (KLT) [38]. Without loss of generality, suppose that $\bar{\mathbf{x}} = \sum_i \mathbf{x}_i = 0$[1] . Let $\{(\mathbf{e}_i, \lambda_i), i = 1, 2, \cdots, N\}$ be the eigensystem

[1]This can be obtained by subtracting the mean of all the training samples from every sample.

of $XX^T$, where $\lambda_1 \geq \lambda_2 \geq \cdots, \geq \lambda_N$. Also let $P = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_N]$. These eigenvectors span a linear face space. Many eigenvectors are devoted to individual differences in face structure, while noise, mainly facial expressions here, are represented orthogonal to these eigenvectors [33]. So we can reconstruct the neutral face structure if proper eigenvectors are selected, which are often the first $K$ ($K \leq M$) eigenvectors. Let $\hat{P} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_K]$. Thus, the reconstructed basic face can be approximately represented as

$$\hat{\mathbf{x}}_e \approx \hat{P}\mathbf{w}, \tag{5.2}$$

where $\mathbf{w} = \hat{P}^T\mathbf{x}_e$. Note that $\mathbf{x}_e$ is the expressional face being recognized while $\hat{\mathbf{x}}_e$ is its corresponding basic face being reconstructed by the projections of $\mathbf{x}_e$ on the selected eigenvectors. The matrix $P$ can be computed from $P = XVQ$, where $V$ is the matrix formed by the eigenvectors of $X^TX$, $Q = [D|O]$, $D = diag(\sqrt{\lambda_1^{-1}}, \sqrt{\lambda_2^{-1}}, \cdots, \sqrt{\lambda_M^{-1}})$ is a diagonal matrix, and $O$ is an $M \times (N-M)$ zero matrix [19]. So,

$$\hat{\mathbf{x}}_e \approx \hat{P}\mathbf{w} = X\hat{V}\hat{Q}\mathbf{w}, \tag{5.3}$$

where $\hat{V}$ is formed by the first $K$ columns of $V$ and

$$\hat{Q} = diag(\sqrt{\lambda_1^{-1}}, \sqrt{\lambda_2^{-1}}, \cdots, \sqrt{\lambda_K^{-1}}). \tag{5.4}$$

With (5.3) and (5.1) we have $\mathbf{c} = \hat{V}\hat{Q}\mathbf{w}$. Finally, the BFSC $\hat{\mathbf{x}}_e$ of an expressional face is computed by $X\mathbf{c}$.

## 5.3 Expressional Regions and An Expression Descriptor

In expressional face images, the eye regions and mouth regions are considered as expressional areas containing rich information of expressions [33]. Since the depth images are built based on the aligned 3D faces, these regions can

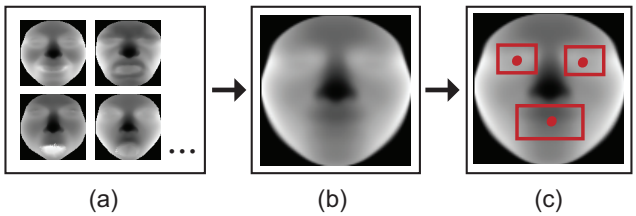



Figure 5.3: Expressional region mask. (a) A group of aligned depth images. (b) The average depth image. (c) The mask including the eye regions and the mouth region.

be easily extracted by a mask. As shown in Fig. 5.3(a), we randomly select 300 depth images from the training samples, and calculate an average depth image (see Fig. 5.3(b)). The centers of the eyes and mouth together with three rectangle regions are used as a mask for extracting the expressional regions. We find that the recognition results are not sensitive to the sizes of the rectangle regions. Thus, the size of each eye region is set to 32×20 and the size of the mouth region is set to 40×25, empirically. The main advantage of the mask is that it helps to find key regions without the need of manual labeling in the testing.

With the BFSC and the original expressional face, the deformation of facial surface can be captured by encoding the depth differences between them. By the expressional region mask, the gray levels of the depth images within the three regions in the BFSC and the original face are arranged with the same order to form two vectors. Then an expression descriptor of a 3D face is defined as:

$$\mathbf{F}(f_e, f_b) = \mathbf{F}(f_e) - \mathbf{F}(f_b), \tag{5.5}$$

where $\mathbf{F}(f_e)$ and $\mathbf{F}(f_b)$ are the vectors extracted from the original face and its corresponding BFSC, respectively. The feature vectors are used for training and testing by an SVM algorithm.

## 5.4 Experiments

In this section, we test our facial expression recognition method on a database named BU-3DFE [53]. The database contains 100 subjects with both males and females and a variety of ethnic ancestries and ages. 25 faces are captured for each subject, i.e., 6 prototypical expressions each with 4 different intensities and a neutral face (see the detailed description of the database in [53]).

The BU-3DFE database is used in existing facial expression recognition work [50, 42, 44]. The previous common setting is that 720 3D faces of 60 subjects are selected in their experiments, each subject with 12 expressional samples (2 higher intensities for every kind of expression). The 60 subjects are randomly partitioned into two subsets, a training set with 54 subjects (648 samples) and a test set with 6 subjects (72 samples). According to different partitions (54 vs. 6), 20 independent experiments are performed in [50], while in [42] and [44], 10 experiments are conducted. The reported results are the averages of the results of the independent experiments. In our experiments, we use the similar setting for comparison purpose.

### 5.4.1 Testing the Ratio of Preserved Energy in the BFSC Estimation

This experiment tests how different ratios of preserved energy in the BFSC estimation affect the recognition. Ten ratios from 0.1 to 1.0 with an interval of 0.1 are tested and Fig. 5.4 shows the average recognition rates. The highest recognition rate of 71.63% is achieved at the energy ratio of 0.4. This ratio is then used for the subsequent experiments.

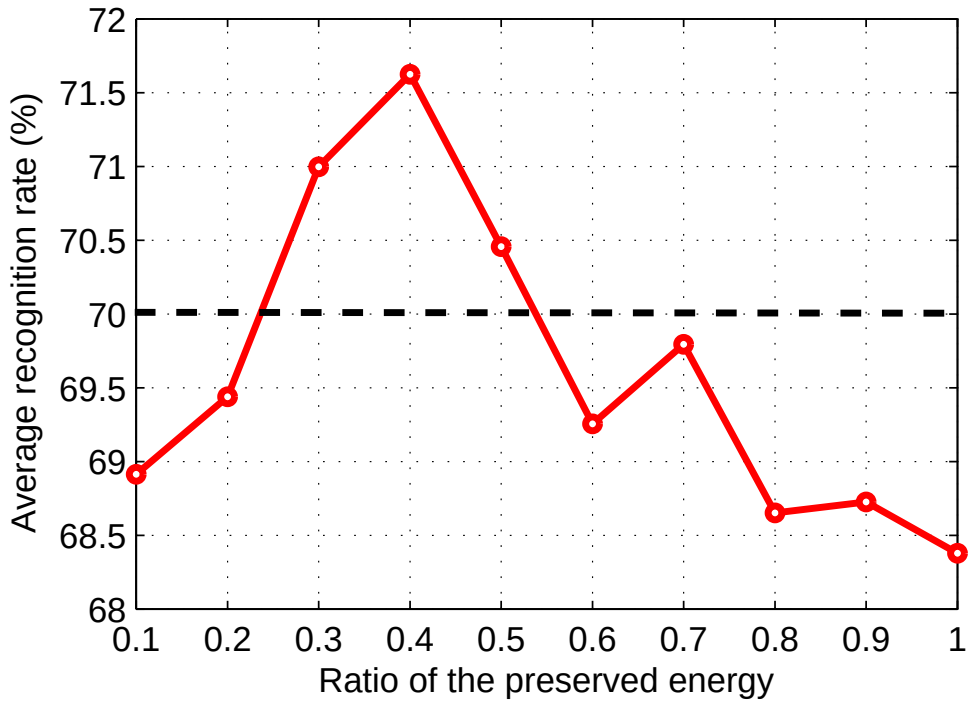



Figure 5.4: Average recognition rate vs. the ratio of preserved energy in the BFSC estimation.

### 5.4.2 Comparison with Related Work

In this section, we compare our method with the related work using the 54-versus-6-subject partitions. As pointed out in the beginning of this section, different partitions are independently trained and tested, and the average of all the results is shown as the final result. The partition process should guarantee that every subject is tested at least once. This subject-based partition aims to test how well the algorithm is with respect to the person-independent requirement in facial expression recognition. The selected 60 subjects are the same as those in [50]. However, we find that neither 20 [50] nor 10 [42, 44] times of experiments are enough to have a stable result. The average recognition accuracy obtained by 10 or 20 random experiments varies greatly, from about 50% to more than 90%. So we run the experiments 1000 times independently and obtain stable average recognition accuracies for all the methods. All the

| % | $\mathbf{F}(f_e)$ | $\mathbf{F}(f_e, f_n)$ | $\mathbf{F}(f_e, f_b)$ |
|---|---|---|---|
| dist-soyel [42] | 67.52 | – | – |
| dist-tang [44] | – | 74.51 | – |
| prim-curv [50] | 61.79 | – | – |
| ours (depth) | **68.77** | **76.22** | **71.63** |

Table 5.1: The comparison of different facial expression descriptors.

| % | AN | DI | FE | HA | SA | SU |
|---|---|---|---|---|---|---|
| AN | **71.41** | 12.28 | 2.92 | 0.00 | 15.30 | 2.48 |
| DI | 9.87 | **76.60** | 7.18 | 2.03 | 2.84 | 2.94 |
| FE | 3.91 | 4.92 | **62.48** | 15.33 | 2.87 | 4.76 |
| HA | 0.72 | 2.43 | 9.32 | **81.21** | 0.00 | 0.49 |
| SA | 14.06 | 1.11 | 4.56 | 0.00 | **77.49** | 1.19 |
| SU | 0.03 | 2.66 | 13.52 | 1.42 | 1.50 | **88.13** |

Table 5.2: The average confusion matrix obtained by our method.

results below are obtained using this experimental setting. The RBF kernel of SVM is used for classification in the four methods.

We compare different facial expression descriptions in Table 5.1, including Soyel and Demirel's distance vector (dist-soyel) [42], Tang and Huang's manually designed distance vector (dist-tang) [44], primitive surface feature (prim-curv) similar to [50] but obtained based on [43], and our depth feature (depth).

In Table 5.1, the second column records the recognition results of the features obtained from the expressional faces only ($\mathbf{F}(f_e)$), the third column shows the results of the features based on both the expressional faces and the neutral faces ($\mathbf{F}(f_e, f_n) = \mathbf{F}(f_e) - \mathbf{F}(f_n)$, where $f_n$ denotes a neutral face), and the fourth column shows the result from the expressional faces and the estimated BFSCs ($\mathbf{F}(f_e, f_b)$). Obviously, $\mathbf{F}(f_e, f_b)$ performs better than $\mathbf{F}(f_e)$, which indicates that the separation of BFSC and ESC is effective for expression recognition. When compared with [44] that requires neutral faces and the manual labeling (the third column), our method (also using the neutral faces) still obtains better result. It should be noted that the previous three methods

all need manually labeled facial key points for the recognition, while ours is automatic.

One common characteristic of all the descriptors is that they all recognize "Happiness" and "Surprise" better than other types of facial expressions. The average confusion matrix obtained by our algorithm is shown in Table 5.2, where AN, DI, FE, HA, SA and SU are short for "Anger", "Disgust", "Fear", "Happiness", "Sadness" and "Surprise", respectively.

## 5.5 Conclusion

In this chapter, we have developed an automatic 3D facial expression recognition algorithm requiring no manual facial keypoint labeling assistance. An expressional face is separated as a basic facial shape component (BFSC) and an expressional shape component (ESC). The description of facial expressions is designed based on both the original expressional face and its BFSC. Our algorithm obtains the highest average recognition rates in the comparison experiments. In addition, we find that the neutral face plays an important role in improving the facial expression recognition accuracy, which further shows that the separation of BFSC and ESC should be an important part of a facial expression recognition system.

# Chapter 6

# Conclusion

In this thesis we have carried out our initial attempts on 3D object retrieval and recognition, i.e., a novel shape descriptor for 3D articulated object retrieval and its enhancement for retrieving generic 3D objects, a modification to the typical retrieval framework by using a set of referent objects, and an automatic 3D facial expression recognition algorithm. In this chapter, we will summarize our contribution and important results, as well as some suggestive directions for future work.

Chapter 1 introduced voxel, mesh, and range image models for 3D object representation firstly, and then outlined our work in this thesis.

Chapter 2 reviewed related work on 3D object retrieval, including some main modules of a typical retrieval framework, publicly available databases, and some experimental systems. Query formulation and user interface are gaining much more prominence when we come to 3D object retrieval, because a common user is usually short of 3D models and might find it difficult to build a 3D model using current softwares or tools. Canonical coordinate normalization is adopted by many systems to achieve invariance to 3D object's translation, scale, and rotation, while some other systems fulfill this task by using invariant shape descriptors. Feature extraction is the essential part for a successful 3D object retrieval algorithm, and hence plenty of work can be found on this topic [46, 12, 52]. We also proposed a novel one in chapter 3 and showed its

superiority over existing descriptors. In this chapter, we also surveyed existing databases and experimental systems as thorough as we could. At the end of this chapter, we summarized some challenges on 3D object retrieval.

In chapter 3, we proposed a novel shape feature called object flexibility to describe the local characteristic of a 3D object. The bag-of-words model was introduced to formulate a final shape descriptor based on the local feature. We demonstrated that this descriptor was suitable for retrieving objects with articulated parts, and was a good supplement to generic object retrieval methods.

We showed our trial modification to the typical 3D object retrieval framework by involving a weak prior, whether an object was man-made or natural, in chapter 4. Motivated by the fact that this prior was able to improve the retrieval results on PSB database [40] distinctly, we proposed to use a 3D object classifier to learn this prior and incorporate it to traditional retrieval algorithms. Some promising experimental results were shown in this chapter. We suggest that some other prior information can be learned from the referent objects.

Chapter 5 was about our work on 3D object recognition, in particular, 3D facial expression recognition. An automatic posture alignment process was leveraged to replace the tedious manually labeling work in previous work. Another advantage of our work was that our algorithm synthesized a basic facial shape component of an expressional 3D face, which was a byproduct of our recognition algorithm, and was also an functionality of our algorithm when no neutral faces were available.

As the increase of 3D object number and the development of 3D scanning techniques, issues raised during this trend will attract more and more research efforts. 3D object retrieval and recognition will remain as the most challenging problems.

# Bibliography


[1] *Wikipedia:*, `http://en.wikipedia.org/`.

[2] S. Owen, A survey of unstructured mesh generation technology, in *7th International Meshing Roundtable*, volume 3, Citeseer, 1998.

[3] A. Shamir, A survey on mesh segmentation techniques, in *Computer Graphics Forum*, volume 27, pages 1539–1556, John Wiley & Sons, 2008.

[4] J. Peng, C. Kim, and C. Jay Kuo, Journal of Visual Communication and Image Representation **16**, 688 (2005).

[5] I. Akyildiz, X. Wang, and W. Wang, Computer Networks **47**, 445 (2005).

[6] A. Jaimes and N. Sebe, Computer Vision and Image Understanding **108**, 116 (2007).

[7] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, Proceedings of the IEEE **94**, 1948 (2006).

[8] B. Gong, C. Xu, J. Liu, and X. Tang, Boosting 3D object retrieval by object flexibility, in *Proceedings of the seventeen ACM international conference on Multimedia*, pages 525–528, ACM, 2009.

[9] B. Gong, Y. Wang, J. Liu, and X. Tang, Automatic facial expression recognition on a single 3D face by exploring shape deformation, in *Proceedings of the seventeen ACM international conference on Multimedia*, pages 569–572, ACM, 2009.

[10] J. Tangelder and R. Veltkamp, Multimedia Tools and Applications **39**, 441 (2008).

[11] *Princeton 3D Model Search Engine:*, `http://shape.cs.princeton.edu/search.html`.

[12] P. Min, J. Halderman, M. Kazhdan, and T. Funkhouser, Early experiences with a 3D model search engine, in *Proceedings of the eighth international conference on 3D Web technology*, ACM, 2003.

[13] *3D Search Engine:*, `http://3d-search.iti.gr/3DSearch`.

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

[15] J. Tedjokusumo and W. Leow, Normalization and Alignment of 3D Objects Based on Bilateral Symmetry Planes, in *International Multimedia Modeling Conference*, page 74, Springer Verlag, 2007.

[16] B. Bustos, D. Keim, D. Saupe, T. Schreck, and D. Vranić, International Journal on Digital Libraries **6**, 39 (2006).

[17] Y. Yang, H. Lin, and Y. Zhang, IEEE Transactions on Systems and Cybernetics part C Application and Reviews **37**, 1081 (2007).

[18] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, ACM Transactions on Graphics **21**, 807 (2002).

[19] R. Ohbuchi, T. Minamitani, and T. Takei, International Journal of Computer Applications in Technology **23**, 70 (2005).

[20] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung, Computer Graphics Forum **22**, 223 (2003).

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

[22] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser, The Princeton shape benchmark, in *Shape Modeling International*, pages 167–178, 2004.

[23] K. Jarvelin and J. Kekalainen, IR evaluation methods for retrieving highly relevant documents, in *ACM SIGIR conference on Research and Development in Information Retrieval*, Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA,, 2000.

[24] D. Vranic and D. Saupe, 3D model retrieval, in *Spring Conference on Computer Graphics and its Applications*, pages 3–6, 2000.

[25] *3D Shape Retrieval Engine:*, `http://www.cs.uu.nl/centers/give/multimedia/3Drecog/3Dmatching.html`.

[26] J. Zhang, K. Siddiqi, D. Macrini, A. Shokoufandeh, and S. Dickinson, Retrieving articulated 3-D models using medial surfaces and their graph spectra, in *International Workshop on Energy Minimization Methods in CVPR*, 2005.

[27] W. Regli and V. Cicirello, Computer Aided Design **32**, 119 (2000).

[28] H. Berman et al., Acta Crystallographica Section D: Biological Crystallography **58**, 899 (2002).

[29] N. Iyer, S. Jayanti, and K. Ramani, Proceedings of ASME IDETC/CIE , 293 (2005).

[30] *3D Shape Retrieval Contest (Shrec):*, `http://www.aimatshape.net/event/SHREC`.

[31] *3D Model Retrieval System:*, `http://3d.csie.ntu.edu.tw/~dynamic/`.

[32] D. Chen and M. Ouhyoung, *Three-dimensional model shape description and retrieval based on lightfield descriptors*, PhD thesis, Ph. D. dissertation, NTU CSIE, 2003.

[33] *3D Model Retrieval System:*, `http://merkur01.inf.uni-konstanz.de/CCCC/`.

[34] J. Tangelder and R. Veltkamp, Polyhedral model retrieval using weighted point sets, in *Shape Modeling International*, pages 119–129, 2003.

[35] A. D. Bimbo and P. Pala, ACM Transaction on Multimedia Computing, Communications, and Applications **2**, 20 (2006).

[36] V. Jain and H. Zhang, Computer-Aided Design **39**, 398 (2007).

[37] A. Ion et al., 3D shape matching by geodesic eccentricity, in *CVPR Workshop*, pages 1–8, 2008.

[38] L. Fei-Fei and P. Perona, A bayesian hierarchical model for learning natural scene categories, in *IEEE Conference on Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[39] R. Ohbuchi, T. Minamitani, and T. Takei, Int'l J. of Computer Applications in Technology **23**, 70 (2005).

[40] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, `http://www.csie.ntu.edu.tw/{\textasciitilde}cjlin/libsvm`.

[41] *SHREC: Shape Retrieval Contest*, http://www.aim-at-shape.net.

[42] B. Fasel and J. Luettin, Pattern Recognition **36**, 259 (2003).

[43] M. Pantic and L. Rothkrantz, IEEE Transactions on Pattern Analysis and Machine Intelligence **22**, 1424 (2000).

[44] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, A 3D Facial Expression Database for Facial Behavior Research, in *International Conference on Automatic Face and Gesture Recognition*, 2006.

[45] P. Phillips et al., Overview of the face recognition grand challenge, in *IEEE Conference on Computer Vision and Pattern Recognition*, pages 947–954, 2005.

[46] J. Wang, L. Yin, X. Wei, and Y. Sun, 3D Facial Expression Recognition Based on Primitive Surface Feature Distribution, in *IEEE Conference on Computer Vision and Pattern Recognition*, pages 17–22, 2006.

[47] H. Soyel and H. Demirel, Facial Expression Recognition Using 3D Facial Feature Distances, in *International Conference on Image Analysis and Recognition*, pages 17–22, 2006.

[48] H. Tang and T. S. Huang, 3D Facial Expression Recognition Based on Automatically Selected Features, in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.

[49] C. Padgett and G. Cottrell, Advances in Neural Information Processing Systems , 894 (1997).

[50] Y. Wang, X. Tang, J. Liu, G. Pan, and R. Xiao, 3D Face Recognition by Local Shape Difference Boosting, in *European Conference on Computer Vision*, 2008.

[51] P. J. Besl and N. D. McKay, IEEE Transactions on Pattern Analysis and Machine Intelligence **14**, 239 (1992).

[52] T. Russ, C. Boehnen, and T. Peters, 3D Face Recognition Using 3D Alignment for PCA, in *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 2006.

[53] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1990.

[54] H. Tanaka, M. Ikeda, and H. Chiaki, Curvature-Based Face Surface Recognition Using Spherical Correlation. Principal Directions for Curved Object Recognition, in *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 372–377, 1998.