Study of a View-based 3-D Object Retrieval Method for 3-D Object Reconstruction

（3 次元再構成のためのビューベース 3-D オブジェクト復元に関する研究）

Yongsheng Zhang
Graduate School of Information Science and Technology
Hokkaido University
Sapporo, Hokkaido, Japan

Table of Contents

Abstract

With rapid advances in computer techniques and the popularity of the camera, a large number of photographs can be obtained. How to obtain a novel 3-D object or 3-D scene from 2-D images is a challenging task.

To create a 3-D object, two methods can be applied. One is to create 3-D object by using some 3-D modeling methods. The other is to create 3-D object by combining or revising existing designs. According to a research report, 20% of designs should be started from the very beginning reconstruction, and 80% of designs can be obtained by combining or revising existing 3-D models.

Many research about 3-D modeling has been investigated, but it is still a high-cost and laborious task to model highly realistic 3-D models. We have also proposed a super-pixel based method to get a 3-D model from image sequence. With the development of 3-D technology and 3-D applications, people can easily get a lot of existing 3-D model. We can use these existing model instead of a direct 3-D modeling from images. So we focus on the view-based 3-D object retrieval.

Object retrieval has attracted much research attention in recent years. Confronting object retrieval, how to estimate the relevance among objects is a challenging task. We focus on view-based object retrieval and propose a multi-scale object retrieval algorithm via learning on graph from multimodal data. In our work, shape features are extracted from each view of objects. The relevance among objects is formulated in a hypergraph structure, where the distance of different views in the feature space is employed to generate the connection in the hypergraph. To achieve better representation performance, we propose a multi-scale hypergraph structure to model object correlations. The learning on graph is conducted to estimate the optimal relevance among these objects, which are used for object retrieval. To evaluate the performance of the proposed method,

we conduct experiments on the National Taiwan University dataset and the ETH dataset. To evaluate the 3-D object retrieval performance of our method, we employ the state-of-the-art methods of ED, AVC and QVS for comparison. In order to measure the 3-D object retrieval performance, the criteria of NN, F, DCG and ANMRR are employed to compare different methods in our experiments. Experimental results and comparisons with the state-of-the-art methods demonstrate the effectiveness of the proposed method.

List of Figures

Chapter 1

1. Introduction

In this chapter, we provide the background for understanding image-based 3-D object reconstruction and 3-D object retrieval. We then briefly define the motivation and goal of our research. The last section of this chapter provides an overview of contents of the rest of this book.

1.1 Background

With rapid advances in computer techniques and the popularity of the camera, a large number of photographs are obtained. How to obtain a novel 3-D object or 3-D scene from 2-D images is a challenging task. In the past several decades, 3-D object reconstruction has been investigated extensively. In recent years, graphics hardware and image processing techniques have made remarkable progress and 3-D techniques have been applied in various fields, such as computer-aided design(CAD), medical diagnosis, online shopping, virtual reality(VR), and entertainment. For example, 3-D navigation of city and museum have become much more popular nowadays.

The long history of 3-D technology can be drawn the way back to the start of photography. Stereoscopic photography, or the technique of creating a "third dimension", was first invented in 1838 by the English scientist Sir Charles Wheatstone. Stereoscopic photography is a special motion picture camera system that records images from two different perspectives. Eyewear is then used to combine these perspectives and create the illusion of depth. In 1965, a team led by Charles Lang from Cambridge University

started conducting 3-D CAD modeling research. And 3-D modeling became popular and has been applied to various design works. In 1981, Hideo Kodama of Nagoya Municipal Industrial Research Institute published his account of a functional rapid prototyping system using photopolymers. A solid, printed model was built up in layers, each of which corresponded to a cross-sectional slice in the model. On 31 January 2010, BSKYB became the first broadcaster in the world to show a live sports event in 3-D when Sky Sports screened a football match between Manchester United and Arsenal to a public audience in several selected pubs.

With the development and wide application of 3-D technology, various methods have been proposed to produce a lot of 3-D data. Simultaneously, 3-D data are increased in both local data storage and online data storage. To create a 3-D object or 3-D scene, two methods can be applied as Figure 1.1 shows. One is to create 3-D object by using some 3-D modeling methods including active methods and passive methods. The other is to create 3-D object by combining or revising existing designs. According to a report from Gunn[1], 20% of designs should be started from the very beginning reconstruction, and 80% of designs can be obtained by combining or revising existing designs. Although lots of methods have been investigated, 3-D modeling is still a high cost and laborious task. The combination and revision of existing 3-D models can improve model design performance.

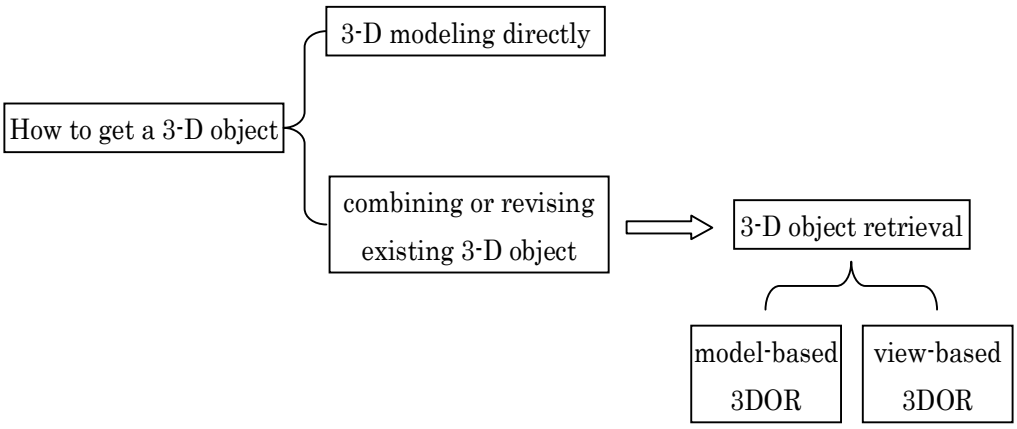

Figure 1.1 The structure of 3-D object rebuild

1.2 Image-based 3-D Modeling(IBM) and View-based 3-D Object Retrieval(VBOR)

The motivation of this paper is to explore solutions to get 3-D object from image views. Image-based modeling methods rely on a set of 2-D images of a scene to generate a 3-D model. In recent years, great progress has been made with applications in the field, and various methodologies have been proposed to deal with related problems. Some of these methods have produced outstanding results, and can be roughly divided into four categories [2]. The first involves computation of a 3-D volume and extraction of an optimal surface from it [3]. The accuracy of this approach is limited by the resolution of the voxel grid. The second involves the use of voxels, level sets or surface meshes, and iterative evolution of a surface to reduce or minimize the cost function. Although this approach is widely used in medical image 3-D reconstruction, its applicability is limited. The third [4, 5, 6] is based on a set of depth maps. The solutions are flexible, but require a set of such maps to be merged into a 3-D scene with consistency constraints. The fourth involves extracting feature points and matching followed by the fitting of a surface to the reconstructed points. With this method, wide-baseline stereo matching is applied to an MVS model [7] to recover salient 3-D

features, and a visual hull model can be shrunk so that the recovered points lie on the surface. The results are then refined using cost function minimization. Although 3-D model reconstruction has been investigated for decades, it is still a challenging task and costs much computation load.

In recent years, 3-D object retrieval[12] has attracted much research attention from both research and industrial fields. Extensive research attention [13,14,15] has been dedicated in such an emerging field[16-19], from either model-based [20,21,22]or view-based directions [23-26], based on the representation methods of 3-D objects. Most of early objects are model-based, where each object is represented by a virtual 3-D model, such as triangle mesh. Model-based methods have shown advantages when describing the global spatial information. However, one main limitation of such methods lie in situation of lack of model data. Model-based methods highly depend on the virtual model, where such model information may be not available in many practical applications.

Different from model-based methods, view-based methods [23,30,24,31] have become more useful in recent years. In view-based methods, each object is represented by a set of views from different directions. Such methods are much more flexible than model-based methods, as the model information is not mandatorily required. Also, view-based methods can be beneficed from existing image processing achievements, such as image feature extraction and comparison. Figure 1.2 provides examples of views from objects. Daras et al. [32] introduced that view-based methods [33, 34] can be more discriminative than model-based methods, and view-based methods have been investigated in recent decade.

In view-based methods, generally, object comparison is based on multi-view matching. It is noted that it is still a challenging task to compare two objects via two groups of views, which is different from traditional image comparison. In this paper, we focus on view-based object retrieval and propose a multi-scale object retrieval algorithm

via learning on graph. Figure 1.3 shows the framework of our proposed method. In our work, shape features are extracted from each view of objects. The relevance among objects is formulated in a hypergraph structure, where the distance of different views in the feature space is employed to generate the connection in the hypergraph. To achieve better representation performance, we propose a multi-scale hypergraph structure to model object correlations. The learning on graph is conducted to estimate the optimal relevance among these objects, which are used for object retrieval. To evaluate the performance of the proposed method, we conduct experiments on the National Taiwan University dataset and the ETH dataset. Experimental results and comparisons with the state-of-the-art methods demonstrate the effectiveness of the proposed method.

Figure 1.2 Example views of 3-D objects

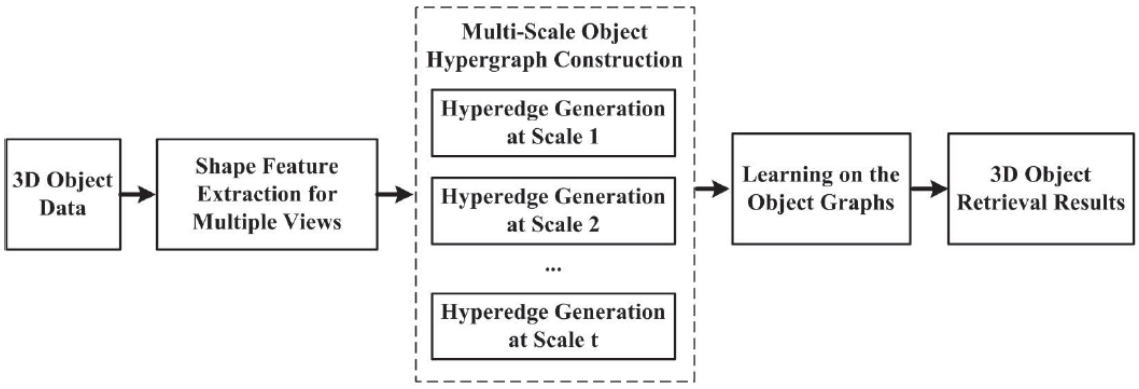

Figure 1.3 The framework of the proposed method.

1.3 Thesis Overview

This thesis is organized as follows. This chapter presents the background, research motivation and structure of this thesis. We describe the history and progress of the development of 3-D technology. We also introduce the IBM and 3DOR as well as the solutions and our contributions to get 3-D object from image views.

In Chapter 2 we provide a brief overview of previous research that can be considered background for the proposed method. We do some experiment in image-based modeling, then introduce the recent progress in 3-D object retrieval.

Chapter 3 presents the 3-D object retrieval method that we use in our research. We focus on view-based object retrieval and propose a multi-scale object retrieval algorithm via learning on graph from multimodal data.

In Chapter 4 we introduce the testing datasets , comparing methods, evaluation criteria and experimental results. Experiments are conducted on two public datasets the National Taiwan University dataset and the ETH dataset. Methods of ED, AVC and QVS are employed for comparison. The criteria of NN, F, DCG and ANMRR is employed to evaluate the performance of different methods.

Chapter 5 is the concluding remarks and future work.

Chapter 2

2. Related Work

From the 2-D image sequence to get the 3-D model, the general approach is the method of 3-D modeling. Many research about 3-D modeling has been investigated, but it is still a high-cost and laborious task to model highly realistic 3-D models. We have also done some work in this area. The technology and the analysis will be introduced in section 2.1. With the development of 3-D technology and 3-D applications, people can easily get a lot of existing 3-D models. We can use these existing models instead of a direct 3-D modeling from images. So we focus on the view-based 3-D object retrieval. From Section 2.2 we will introduce recent progress on 3-D object retrieval.

2.1 From Images to 3-D models

2.1.1 A General IBM Method

The acquisition methods of the real object model can be roughly divided into two categories: active method and passive method. The typical representative of the active approaches is the method of using scanner. It can get the 3-D model of objects accurately, while its cost is very high and it is difficult to acquire enough data to reconstruct models in all applications. The passive method is to reconstruct 3-D model based on images of the scene or objects one wishes to reconstruct, which is economical and can get features with color directly. The baseline of the image-based method is shown as the figure 2.1. Provided with a sequence of images of a scene or an object, it

can generate a realistic 3-D model in five steps. Firstly, for an image is just a large collection of pixels with their own intensities, features matching should be applied. That is to say, to reconstruct the model of this object based on images, it should find the corresponding features, such as points in different images, which may be completed by comparing intensities over a small local window centered with a point. Secondly, the motion and the 3-D structure should be recovered. If there is not a priori calibration of the camera used to get the image sequences, a projective skew should be contained in this stage to recovery the calibration of camera. Thirdly, with the knowledge of the camera parameters that we have got in the second step or based on priori, we can get a depth estimate for almost each pixel of an image to match all pixels of an image with pixels in its neighboring images. In this way, all points of the object can be reconstructed. Fourthly, 3-D model is built by fusing the results together. Finally, texture mapping should be applied to achieve the final photo-realistic model.

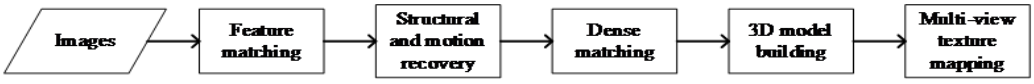

Figure 2.1 The baseline of the image-based 3-D model reconstruction algorithms.

2.1.2 Super-pixel based IBM

Generally, IBM processing is considered high cost and inefficient for texture-less images. We propose a super-pixel based IBM method to solve the problem. The pixels of the reference images are clustered into super-pixels in preprocessing before reconstruction. Then the reconstructing is based on these super-pixels. Figure 2.2 gives an outline of the super-pixel based approach. First, an image sequence is input, camera parameters are derived for the subsequent processing based on a general structure and motion method. At the same time, good images are selected for reference and split into small segments (super-pixels). Feature points are then extracted for each image, but correspondent relative matching points are found only between images relating to the references. From the salient matching points and camera parameters, 3-D points are

derived via triangulation and used to construct salient-basis 3-D surface patches. However, the number of these patches is small. It is necessary to expand the patches to nearby voxels in line with the geometric relationships linking the super-pixels. A photometric discrepancy function is used to remove the incorrect 3-D surface patches at the post-expansion stage. Finally, the 3-D patches are used to reconstruct the polygonal surface model via an optimization procedure.

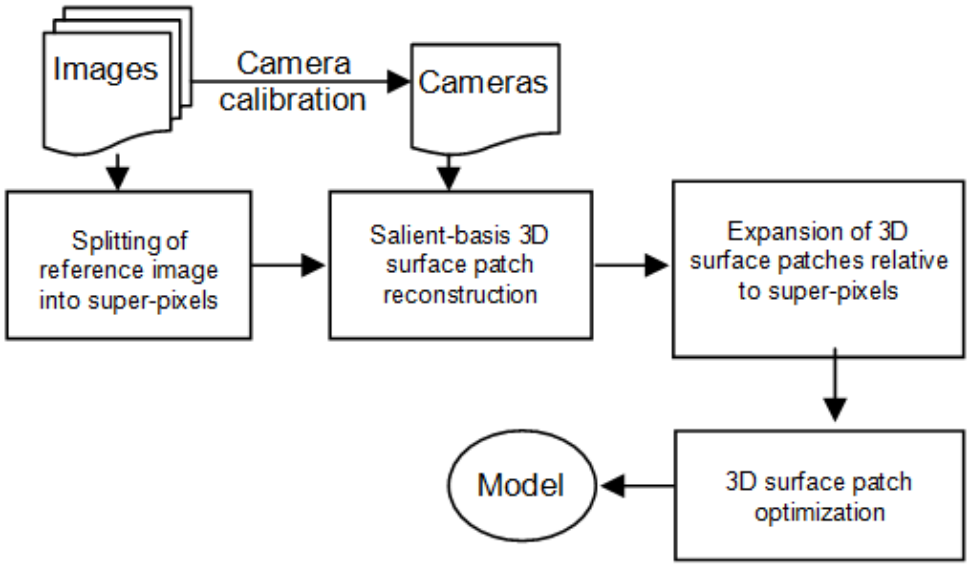

Figure 2.2 Outline of the super-pixel based 3-D modeling approach

Camera calibration is a basic technique in multi-view image processing. Many methods have been developed and applied in related areas. The bundle method [61] is used to simulate the process of structure and motion analysis. After this step, a sparse set of 3-D points and camera parameters describing the relationship between the camera and the scene are occupied.

The concept of super-pixel is proposed by Ren and Malik[58], originally defined as a kind of over-segmentation. Super-pixel in this study is defined as a group of many adjoining pixels, where all pixels have similar properties. For the application on 3-D reconstruction, super-pixel has four specific properties:

Most of feature points should be on the boundaries of super-pixels.

Surfaces without textures should also be split into small clusters (no specific rule is applied for such splitting).

Brightness and texture differences in super-pixels are homogeneous.

The number of super-pixels should be large enough to preserve more features in one image.

In this study, we arrange initial seeds under a lattice grid, where one initial seed is selected in each grid with the most significant features. If there are no significant features in a grid, the center point of the grid is selected. The feature is defined using the Harris and SIFT method.

Followed by the seed initialization, the next step is to generate the local optimal path connected neighborhood of each seed vertically and horizontally. The searching algorithm of shortest path is used to find the path. We define an region undirected graph according to the position and gradient of pixels. In the undirected graph, the node denotes the pixel, and each node has a weight. The weight is defined as:

$$w = 1.0/(1.0 + s * t) \tag{2.1}$$

$$t = d_x * d_x + d_y * d_y \tag{2.2}$$

Where d_x and d_y is horizontal and vertical derivatives. Sensitivity of s is a tuning parameter.

Dijkstra algorithm is employed to generate the shortest path, and seed connection can be accomplished. We use several points sequence on the edge of seed connection to define it. In our work, no more than 5 Harris or SIFT feature points are selected for an

edge. Therefore, every super-pixel is depicted by a feature points sequence. Figure 2.3 shows an example of super-pixels. It is obviously that most of feature points in this image, such as object boundaries, are located on the edges of super-pixels.

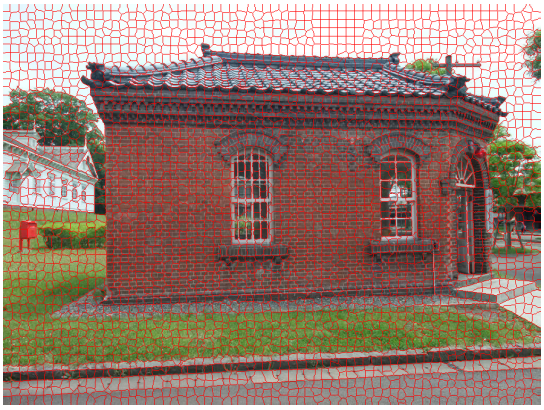

Figure 2.3 An example of resulting super-pixels.

After the definition of super-pixel, 3-D surface patch is defined as following. A 3-D surface patch model p is essentially a local tangent plane approximation of a surface with specifically boundary corresponding to a super-pixel in the reference image. The projection of a 3-D surface patch onto the reference image forms a super-pixel in the image. It has four properties: position $pos(p)$; unit normal vector $n(p)$; vertexes $node(p)$ corresponding to the super-pixel polygon vertexes; and a reference images $r(p)$ in which p is visible. In contrast to the usual definition of a patch [7], the patch referred here may be polygonal rather than rectangular (Figure 2.4).

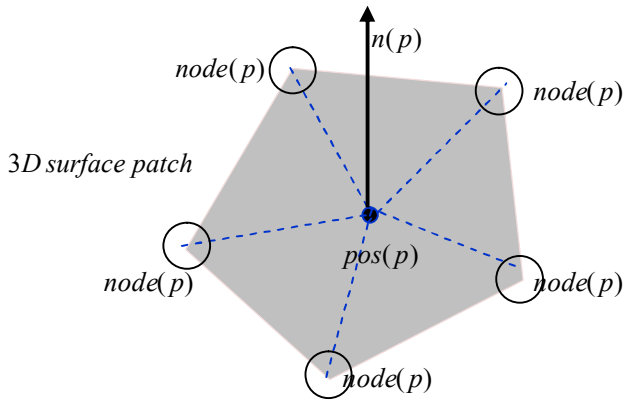

Figure 2.4 Patch definition

3-D surface patch is corresponding to 2-D super-pixel. First, as a normal dense matching method, feature points in each image are detected using SIFT and Harris operators. In this study, super-pixel is applied to reduce the processing time and increase computing efficiency by control the threshold number of features in super-pixels. Only 4 feature points are computed for a super-pixel with the size of 256 pixels in this study. For each feature point f_i in the super-pixel of the reference image I_r, the best matching point f_i' in the other image is searched, and 3-D points $c(i)$ associated with the pairs of matching points are triangulated as shown in Eq.2.3. Multiple 3-D points may be derived because multiple feature points (the most is 4) in a single super-pixel may be derived. The mean coordinates of these points are set for the initial position of this 3-D surface patch corresponding to the super-pixel (see Eq. 2.4).

$$c(i) \leftarrow \{triangulation\ from\ f_i\ and\ f_i'\} \tag{2.3}$$

$$pos(p) \leftarrow \sum c(i) \tag{2.4}$$

The direction vector of the patch is defined as a unit vector from a position $c(i)$ orienting toward the camera $O(I_r)$ of the reference image as

$$n(p) \leftarrow \overrightarrow{pos(p)O(I_r)} / |\overrightarrow{pos(p)O(I_r)}| \tag{2.5}$$

It is clearly that a point and a normal direction can define a plane. The nodes of 3-D surface patches are defined by the intersection points via back projection of the nodes of super-pixel polygon with the patch plane (see Figure 2.5).

For the initialization of reference image $r(p)$, it is first initialized from the outcome of camera calibration, and optimized by removing some images with bigger photometric discrepancy score compared to others. Indeed some patches are also removed when all the discrepancy scores are big.

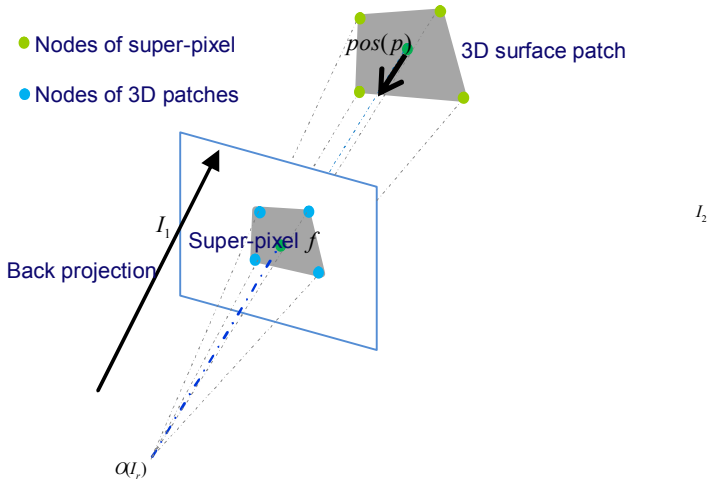

Figure 2.5 The procedure of back projection, during this procedure3-D surface patch nodes are acquired.

As some images are texture-less images, and some patches are removed for big discrepancy score. The initial 3-D surface patches are sparse, not all super-pixels have relative 3-D surface patches. This study uses an expansion procedure to create more patches. The goal of the expansion is to reconstruct corresponding patch for each super-pixel. In the study, the super-pixel is called fixed super-pixel, when it has a relative 3-D patch. This spreads fixed super-pixel to nearby super-pixels and makes as many super-pixels associated with a corresponding determinate 3-D patches as possible. For a 3-D surface patch, all the neighboring super-pixels of its fixed super-pixel are

identified in the reference image. If the neighbor is already associated with a patch, expansion in this orientation is terminated. For a known patch p and its corresponding fixed super-pixel s, a new patch p' corresponding to the neighboring super-pixel s' is generated as following. Unit normal vector $n(p')$ and reference images $r(p')$ are initialized by replicating the values of the known patch p. Position $pos(p')$ is initialized based on the intersecting point where a viewing ray passing through the center of the super-pixel s' intersects the plane of patch p. Depth testing is also performed to prevent expansion from the depth with dramatic changing. Until now, the expanded patches are some coarse ones because their normal vectors may be with big mistakes. As Figure 2.6 shows, when the expanding surface plane is curved surface, the normal vector of patch must be corrected as Figure 2.7 shows. The problem can be thought as a nonlinear minimization problem. Quasi-Newton method with a relevance evaluation is applied to get the right normal vectors.

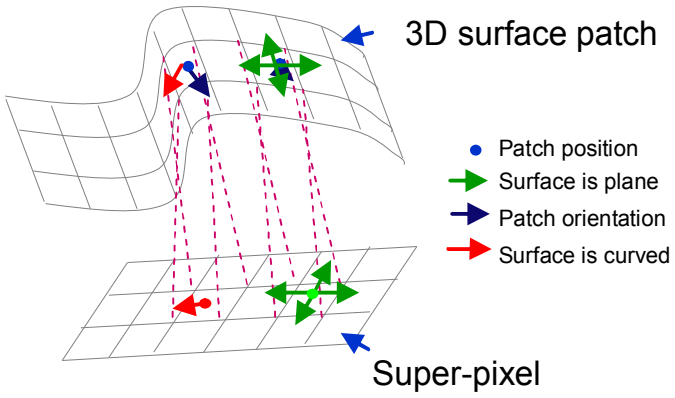

Figure 2.6 Patches expanded on curved surface. Normal vector correct is necessary.

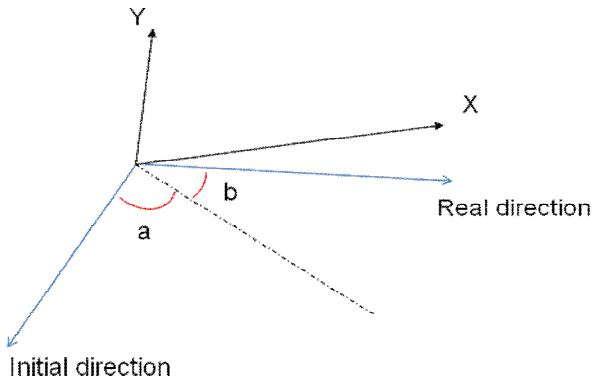

Figure 2.7 The purpose of normal vector correction is to find the angles a and b, it is a nonlinear minimization problem.

The proposed algorithm was tested with a number of datasets, and was found to be valid as Figure 2.8 and Figure 2.9 shows. Its applicability to the processing of complicated surface objects is currently limited (e.g. leaf, grass etc).

Figure 2.8 Sample images from the input dataset

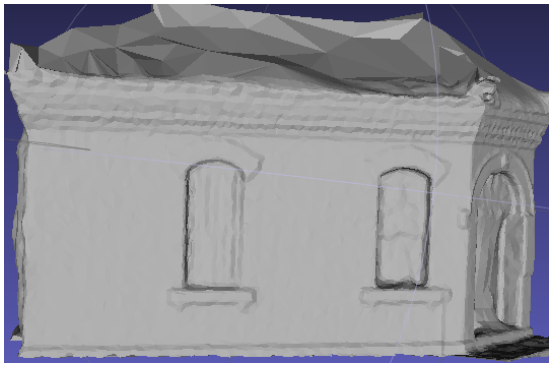

Figure 2.9 Final polygon model simulated from 3-D surface patches using meshlab software.

2.2 3-D Object Retrieval

In this section, we will introduce recent progress on 3-D object retrieval. 3-D object retrieval can be divided into two types of methods, i.e., model-based methods and view-based methods.

2.2.1 Model-based Methods

Most early 3-D object retrieval methods are model-based method. For model-based methods, existing virtual 3-D model information is required. For these methods, low level features, such as volumetric descriptor [27], surface distribution [21] and geometry[20, 28, 29] can be extracted for object description. Other methods [63, 64] extract high level feature for object structure description. To extract model-based feature, Papadakis et al.[35] introduced a panoramic view, i.e., panoramic object representation for accurate model attributing (PANORAMA), where were generated by projecting the model to a lateral surface of a cylinder. To compare two 3-D models, the distance was measured by matching between two PANORAMA images. In [30], Gao et al. introduced a spatial structure circular descriptor (SSCD), which employed the projected model information in a circular region to represent the 3-D model. In SSCD, the projected image is able to preserve the global spatial information, and the comparison between two 3-D models is achieved by the histogram distance for each SSCD view. Vranic et al. [37] introduced an Extension Ray-based Descriptor (ERD) method, where the concentric spheres were used to extract the surface information. In this method, each sampling surface point had a value on the corresponding sphere surface. The disadvantage of model-based 3-D object retrieval is that the 3-D model information is required for the 3-D object retrieval. In the case where no 3-D model is available, the 3-D model reconstruction is needed to generate a model first.

2.2.2 View-based Methods

View-based method is much more flexible compared with model-based methods, because it does not need the virtual model information. For view-based methods, each 3-D model is represented by clusters of multiple views and the feature is extracted from views. The general process is composed of four steps: view capture, view selection, feature extraction and object matching as shown in Figure 2.10.

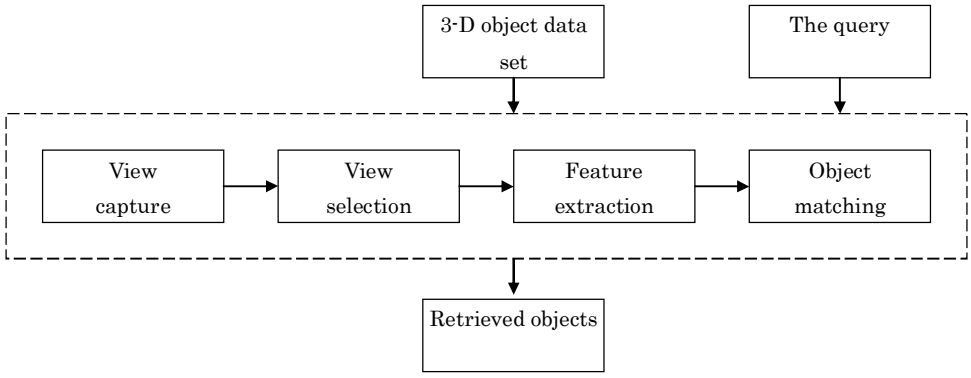

Figure 2.10 The general framework of view-based 3-D object retrieval

Chen et al. [38] proposed the first view-based 3-D object retrieval method, i.e., Lighting Field Descriptor (LFD). In LFD, several groups of 10 views are used to represent each 3-D object. For these views, the Zernike moments and Fourier descriptors were employed as the features. To compare two 3-D objects, the minimal distance between two groups of views were used in [38]. Shih et al.[39] introduced an Elevation Descriptor (ED), which employed six range views from different directions to represent 3-D objects. In ED, the depth histogram was extracted as the ED feature and matching between two ED histograms was measured as the distance between two 3-D objects. To represent 3-D objects via a set of views, five sets of images, from four vertical and one horizontal loop directions, were employed in [65]. In this method, each group of views were formulated as a Markov Chain (MC). The comparison between two 3-D models can be divided into the comparison in the view set level and the comparison in the model level, and the objective of 3-D model retrieval was to find the maximal a posterior (MAP) given the query model. Daras et al. [32] introduced the Compact

Multi-View Descriptor (CMVD), which contained 18 views from the vertices of a 32-hedron. Mahmoudi et al. [40] proposed a model to employ the curvature scale space as the view descriptor. The curvature scale space was combined with Zernike moments to compare 3-D models. Adan et al. [41] proposed a depth gradient image (DGI) model, which employed both the surface and the contour information to avoid restrictions concerning the layout and visibility of 3-D models.

To select representative views from the large view pool, a query view selection(QVS) method was introduced in [45]. In QVS, a small set of candidate views were first selected via view clustering. Then the user relevance feedback was involved to interactively select representative views. In this way, the views were incrementally selected, which can be more discrimitive to the query information. Ansary et al. [23] proposed the Adaptive Views Clustering (AVC), where 320 initial views were firstly captured and about 20 to 40 representative views were selected. The objective for 3-D model retrieval was formulated as a probabilistic approach to measure the posterior probability for one object given the query object, and the higher the posterior probability of one 3-D model, the higher relevance between it and the query. In [34], a probabilistic matching method was introduced, where a positive matching model and a negative matching model were generated individually. For each target object, the positive matching ratio and the negative matching ratio were measured and then combined for retrieval. To measure the distance between two groups of views, Gao et al. [31] proposed a Hausdorff distance learning method, where a view-level Mahalanobis distance metric for Hausdorf distanc was learnt through relevance feedback. Hypergraph learning has also been investigated in 3-D object retrieval [30], where the relationship of 3-D objects were formulated in a hypergraph structure.

In recent years, the bag-of-words methods have been investigated in 3-D object retrieval [46, 47, 48]. Ohbuchi et al. [24] introduced a bag-of-visualfeature (BoVF) method, where the local SIFT features [49] were extracted from the images and visual

words were generated using a clustering-based visual vocabulary. A histogram of visual words was built as the feature for the 3-D object, and the Kullback-Leibler divergence (KLD) was used to measure the distance between 3-D objects.

Chapter 3

3. Multi-Scale Object Retrieval via Learning on Graph

In this Chapter, we introduce the proposed multi-scale object retrieval via learning on graph. The proposed method is composed of three components, i.e., multi-view object distance, multi-scale object graph construction and learning on the graph, as shown in Figure 1.3

3.1 Shape Feature Extraction

Feature extraction is an integral part of multimedia information retrieval. For view feature extraction, several effective features can be used, such as Fourier descriptors[8] and Zernike moments[50,51,23,62]. In our work, the Zernike moments are employed as the visual descriptors.

Moments are considered as popular pattern representation methods. Zernike moments are a class of orthogonal moments and have been shown effective in terms of image representation. Zernike moments are rotation invariant and can be easily constructed to an arbitrary order. Zernike moments are constructed using a set of complex polynomials which form a complete orthogonal basis set defined on the unit circle. Zernike polynomials are expressed in polar coordinates by $\{P_{st}(x,y)\}$ as

$$P_{st}(x,y) = P_{st}(\rho,\theta) = R_{st}(\rho)e^{jt\theta}, \tag{3.1}$$

Where $s = 0,1,2,...,\infty$ and defines the order, t is an positive or negative integer depicting the angular dependence, or rotation, subject to the conditions:

$$s - |t| = even, \qquad |t| \le s \tag{3.2}$$

And (ρ, θ) are defined over the unit circle, $j = \sqrt{-1}$ and $R_{st}(\rho)$ is the orthogonal radial polynomial, which is defined as

$$R_{st}(\rho) = \sum_{z=0}^{\frac{s-|t|}{2}} (-1)^z F(s,t,z,\rho) \tag{3.3}$$

$$\text{where} \quad F(s,t,z,\rho) = \frac{(s-z)!}{z!(\frac{s+|t|}{2}-z)!(\frac{s-|t|}{2}-z)!} \rho^{s-2z} \tag{3.4}$$

Zernike moments are the projections of the image function onto the orthogonal basic functions. The $(s+t)th$ Zernike moment for an image function $f(x,y)$ is defined as

$$ZM_{st} = \frac{s+1}{\pi} \sum_x \sum_y f(x,y) P_{st}(\rho,\theta) \tag{3.5}$$

The number of moments for the k th order is $k(\frac{k}{2}+1)(k+1)$.

3.2 Multi-view Object Distance

Distance metric plays an important role in multimedia information retrieval. For the image retrieval task, which is based on the matching of two single images. 3-D object retrieval is more complex as the high-order information contained in the multiple views of 3-D object. Multi-view 3-D object matching is a many to many matching problem.

Here, we let $O_i = \{v_{i1}, v_{i2}, ...v_{in}\}$ denote one object O_i with n views, and let $O_j = \{v_{j1}, v_{j2}, ...v_{jn}\}$ denote another object O_j. For each view v_{ia}, a shape feature is extracted for view representation. The Zernike moments are employed as the visual descriptors. To measure the distance between O_i and O_j based on multiple views, the following distance measure is employed in our work:

$$d(O_i, O_j) = \frac{1}{n}\sum_{a=1}^{n} d_{\min}(v_{ia}, O_j) \tag{3.6}$$

where $d_{\min}(v_{ia}, O_j)$ is the minimal distance between v_{ia} and all the views in O_j:

$$d_{\min}(v_{ia}, O_j) = \min\{d(v_{ia}, v_{jb}) | b \in [1, n]\} \tag{3.7}$$

Here, $d(v_{ia}, v_{jb})$ is the Euclidean distance between v_{ia} and v_{jb} based on the Zermike moments feature.

3.3 Multi-scale Object Graph Construction

In our work, the relationship among objects are formulated in a object hypergraph. Here, each object is regarded as a vertex in the object hypergraph $\varsigma = (V, E, \omega)$, and how to construct the vertex connection is important. Here, the star expansion method in [52] is employed to generate the hyperedges. Each time, one vertex is selected as the centroid vertex and a hyperedge is generated by connecting its nearest neighbors. A parameter K is used to select the number of nearest neighbors for each hyperedge. Figure 3.1 illustrates the construction of hyperedge via star expansion. We note that different selections of K indicates different representation scales for object formulation. A large K value indicates a large amount of objects can be connected by one edge and a small K value will lead to strict constraint on the similarities of

connected objects by each edge. As it is difficult to identify the optimal K value for hyperedge construction, multiple K values are used in our work to construct edges on the hypergraph, which leads to a multi-scale object graph. As shown in Figure 3.1, multiple K values can construct multiple hyperedges.

The multi-scale object hypergraph $\varsigma = (V, E, \omega)$ is composed by a vertex set V, a hyperedge set E, and the weights of the edges ω. Here, each vertex $v \in V$ denotes one object, the each edge $e \in E$ is constructed via star expansion, and the weight ω is assigned with the equal wight 1.

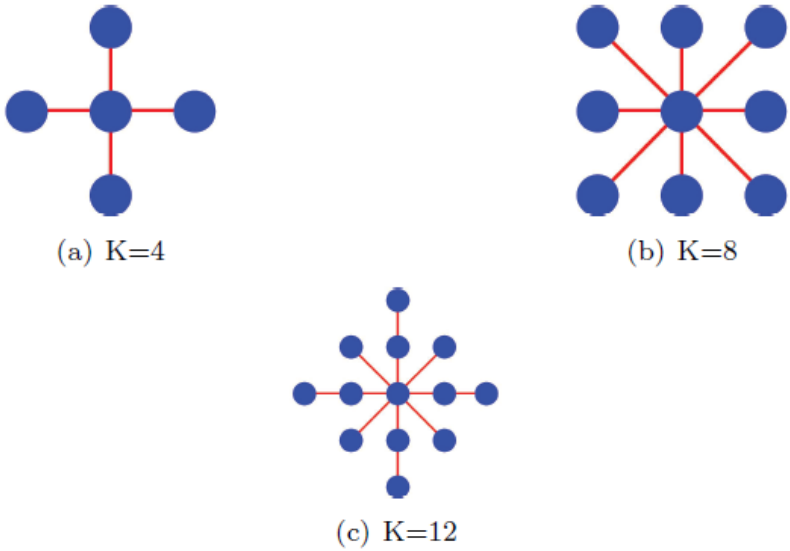

Figure 3.1 Hyperedge generation using different K values.

The structure of the object hypergraph ς is presented by an incidence matrix H, where the entry of H is calculated by:

$$H(v,e) = \begin{cases} 1 & if\ v \in e \\ 0 & if\ v \notin e \end{cases} \tag{3.8}$$

For each vertex, the vertex degree $d(v)$ is defined as:

$$d(v) = \sum_{e \in E} \omega(e) h(v,e) \tag{3.9}$$

For each hyperedge, the edge degree $\delta(e)$ is defined as:

$$\delta(e) = \sum_{v \in V} h(v, e) \tag{3.10}$$

For the vertex degree and the edge degree, two matrices D_v and D_e are used to denote the diagonal matrices of the vertex degrees and the edge degrees respectively.

3.4 Graph Learning for Object Retrieval

To estimate the relevance among objects, it is important to explore the relationship of the vertices in the hypergraph structure. In recent years, learning methods [53, 54] have been performed on hypergraphs, such as data clustering, classification and ranking. In our work, we formulate the object relevance exploration task as a binary classification task, i.e., whether one object belongs to the query type or not.

The following normalized Laplacian hypergraph learning framework is employed here,

$$\arg\min_{f} \{\lambda R(f) + \Omega(f)\} \tag{3.11}$$

where f is the relevance from each object to the query, $R(f)$ is the empirical loss of hypergraph learning on the labeled data, i.e., the query, $\Omega(f)$ is a regularizer on the hypergraph structure, and λ is a parameter to balance different components on the objective function.

The empirical loss $R(f)$ is defined as:

$$R(f) = \|f - y\|^2 = \sum_{u \in V} (f(u) - y(u))^2 \tag{3.12}$$

where y is the labeled vector. In y, all the entries are zeros expect the query, which is one.

The regularizer on the hypergraph $\Omega(f)$ is defined as:

$$\begin{aligned}\Omega(f) &= \sum_{e\in E}\sum_{u,v\in V}\frac{\omega(e)h(u,e)h(v,e)}{\delta(e)}\left(\frac{f^2(u)}{d(u)}-\frac{f(u)f(v)}{\sqrt{d(u)d(v)}}\right)\\ &= \sum_{u\in V}f^2(u)\sum_{e\in E}\frac{\omega(e)h(u,e)}{d(u)}\sum_{v\in V}\frac{h(v,e)}{\delta(e)}\\ &- \sum_{e\in E}\sum_{u,v\in V}\frac{f(u)h(u,e)\omega(e)h(v,e)f(v)}{\sqrt{d(u)d(v)}\delta(e)}\\ &= f^T(I-\Theta)f\end{aligned} \tag{3.13}$$

where $\Theta = D_v^{-\frac{1}{2}}HWD_e^{-1}H^TD_v^{-\frac{1}{2}}$.

Here we let $\Delta = I - \Theta$, equation can be rewritten as

$$\Omega(f) = f^T\Delta f \tag{3.14}$$

Now, the cost function on the learning task can be rewritten as:

$$\Phi(f) = f^T\Delta f + \lambda\left\|f-y\right\|^2 \tag{3.15}$$

The relevance among objects can be learned by the minimizing the objective function, and the optimal f can be solved via:

$$f = (1+\frac{1}{\lambda}\Delta)^{-1}y \tag{3.16}$$

To further reduce the computational cost, Eq.3.16 can be solved using an iterative process as shown in the follows. We first initialize f with $t=0$. Then, we update f

by

$$f^{t+1} = \frac{1}{1+\lambda}(1-\Delta)f^{(t)} + \frac{\lambda}{1+\lambda}y \tag{3.17}$$

We let $t = t+1$ and then go back to update f. This process is repeated until the cost function does not further significantly reduced. With the learned relevance vector f, all the objects can be ranked in a descending order, which generates the object retrieval results.

3.5 Computational Cost

In this part, we analyze the computational cost of our proposed method. The main computational load lies in the hypergraph construction and the learning part. It can be obtained that the hypergraph construction process costs $O(n_K n_O)$, where n_K is the number of employed K in the hyperedge construction part, and n_O is the number of objects in the dataset. The computational cost for the learning part is $O(n_O^2 n_{it})$, where n_{it} is the iteration number for the alternating optimization process.

Chapter 4

4. Experiment

In this section, we introduce the testing datasets, compared methods, evaluation criteria and experimental results.

4.1 Evaluation Datasets

To evaluate the performance of the proposed method, we have conducted 3-D object retrieval experiments on two public datasets, i.e., National Taiwan University 3-D Model database (NTU) [38], and Eidgenössische Technische Hochschule Zürich 3-D object dataset (ETH) [55]. Figure 4.1 and Figure 4.2 shows examples in the NTU and the ETH datasets.

4.1.1 National Taiwan University 3-D Model Database(NTU)

The NTU 3-D model database contains two parts, one is the NTU 3-D model benchmark and the other is the NTU 3-D model database. NTU provides 3-D models for research purpose in 3-D model retrieval, matching, recognition, classification, clustering and analysis. The benchmark contains a database of 1,833 3-D models, which are free downloaded from 3-DCafe (http://www.3-Dcafe.com) in Dec. 2001, but removes several models with failed formats in decoding. The benchmark was clustered into 47 classes including 549 3-D models mainly for vehicle and household items, and

all the other 1,284 models classified as "miscellaneous". 3-D models in the miscellaneous class are not the same function but noise for correct retrieval. The database contains a database of 10,911 3-D models, which are free downloaded from the Internet in July 2002. All 3-D models are converted into Wavefront file format (.obj) in the database. Thumbnail images of each 3-D model are also included in the database.

The first dataset used in our work is the NTU benchmark. In the NTU dataset, each object contains a corresponding 3-D model. Here, the virtual cameras are set to capture multiple views for each object. In our work, a virtual camera array with 60 cameras are used, which locate on the vertices of a polyhedron with the similar structure of Buckminsterfullerene as Figure 4.2 shows. Using these virtual cameras, 60 views can be obtained for each 3-D object.

Figure 4.1 3-D object examples in the NTU database

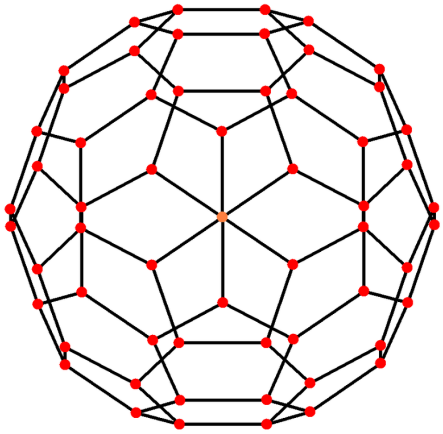

Figure 4.2　Structure of Buckminsterfullerene

4.1.2 The Eidgenössische Technische Hochschule Zürich Database(ETH)

The ETH dataset is a real world 3-D object dataset with multiple views. There are 80 objects from 8 categories in the ETH dataset, such as apple, pear, tomato, dog, cow, cup, car, and horse. In this dataset, each object contains 41 views, which are captured spaced evenly over the upper viewing hemisphere. All the cameras are located on the positions by subdividing the faces of an octahedron to the third recursion level. All images have been taken with a Sony DFW-X700 progressive scan digital camera with 1024×768 pixel resolution and a Tamron 6-12mm varifocal lens (F1.4). For every image, a high-quality segmentation mask is provided.

Figure 4.3　3-D object examples in the ETH database

4.2 Compared Methods

To evaluate the 3-D object retrieval performance of our method, we employ the following state-of-the-art methods for comparison.

4.2.1 Elevation Descriptor(ED)

Elevation descriptor (ED) [39] is a view-based 3-D object retrieval method, which proposed a new feature for 3-D model retrieval, for 2-D silhouettes represented by binary images do not describe the altitude information of the 3-D model from different views well. Based on the new feature, a new content-based multimedia retrieval system of 3-D models retrieval is designed. One 3-D model should be represented with six elevations to describe its altitude information from six different views. Each evaluation is represented by a 2-D gray-level image decomposed into several concentric circles and obtained by taking the difference between the altitude sums of two successive concentric circles.

For each 3-D model, a tightest bounding box is constructed and decomposed into a $2\mathrm{L}\times 2\mathrm{L}\times 2\mathrm{L}$ voxel grid. $voxel(m,n,h)$ represents a voxel located at (m,n,h). Based on whether there is a polygonal surface located within the voxel, $voxel(m,n,h)=1$ or $voxel(m,n,h)=0$. As a result, each voxel with a polygonal surface is weighted equally. Let the model's center move to the location (L,L,L), the average distance from all voxels with $voxel(m,n,h)=1$ to the center is linearly. In this way, ED is robust for rotation and invariant to translation and scaling of 3-D models. In addition, an effective way for extracting features from each gray-level image is employed in order to make them less sensitive to rotations.

Each elevation is decomposed into L concentric circles $C_j, j=1,2...,L$ around the center point to extract the ED from six elevations. For the k th elevation, $g_k(j)$ is

the sum of gray values of pixels in the j th circle and calculated by

$$g_k(j) = \sum_{(r,c)\in C_j} f_k(r,c), \quad j = 1,2,...,L \tag{4.1}$$

The difference between two successive concentric circles is defined as:

$$d_k(j) = g_k(j) - g_k(j-1) \tag{4.2}$$

and the sum of all $d_k(j)$ values for the k th elevation is

$$D(k) = \sum_{j=1}^{L} d_k(j) \tag{4.3}$$

The ED X of a 3-D model is defined as:

$$X = [(x_1)^T, (x_2)^T, (x_3)^T, (x_4)^T, (x_5)^T, (x_6)^T]^T \tag{4.4}$$

where $x_k = [x_k(1), x_k(2), ..., x_k(L)]^T$.

For a 3-D model, six elevations are obtained to describe the altitude information of 2-D projections from six different views: front, top, right, rear, bottom and left which are notated successively as $E_k, k = 1,2...,6$. Taking the relative positions of the elevations into account, it can divide the six elevations into three pairs $(E_1, E_4), (E_2, E_5), (E_3, E_6)$ and reduce the matching time between two models to $3!\times 2^3 = 48$ matching operations. For the i th permutation $p_i, 1 \le i \le 48$ matching operation, the distance between $X = [(x_1)^T, (x_2)^T, (x_3)^T, (x_4)^T, (x_5)^T, (x_6)^T]^T$ and $Y = [(y_1)^T, (y_2)^T, (y_3)^T, (y_4)^T, (y_5)^T, (y_6)^T]^T$ is defined as

$$Dis_{X,Y}^{i} = \sum_{k=1}^{6} \left\| x_k - y_{pi}(k) \right\| = \sum_{k=1}^{6} \sum_{r=1}^{L} \left| x_k(r) - y_{pi(k)}(r) \right| \tag{4.5}$$

and the distance between these two models is

$$Dis_{X,Y} = \min_{1 \le i \le 48} Dis_{X,Y}^{i} \tag{4.6}$$

The similarity measure between X and Y is defined as

$$Sim_{X,Y} = \frac{1}{Dis_{X,Y}} \tag{4.7}$$

The larger the similarity value is, the more similar a model is. Take the model with the largest similarity value as the retrieved model.

4.2.2 Adaptive Views Clustering(AVC)

Adaptive views clustering (AVC) [23] provides a probabilistic Bayesian method for 3-D model retrieval from these views.

A set of characteristic views $V = \{V_1, V_2, ..., V_c\}$ represent a model in model collection D_b, with C the number of characteristic views. Corresponding to a 3-D request model Q, the target retrieval model M_i is the closest one in D_b, with the highest probability of $P(M_i|Q)$. $P(M_i|Q)$ can be written as

$$P(M_i|Q) = \sum_{k=1}^{K} P(M_i|V_Q^k) P(|V_Q^k|Q) \tag{4.8}$$

Where K is the number of characteristic views of the model Q. Let H be the set of all the possible hypotheses of correspondence between the request view V_Q^k and a

model M_i, $H = \{h_1^k \vee h_2^k \vee ... \vee h_N^k\}$. A hypothesis h_p^k means that the view p of the model is the view request V_{cQ}^k. The sign $\vee$ represents logic *or* operator. If an hypothesis h_p^k is true, all the other hypotheses are false. $P(M_i \left| V_{cQ}^k \right.)$ can be expressed by $P(M_i \left| H^k \right.)$.

$$P(M_i \left| H^k \right.) = \sum_{j=1}^{N} P(M_i, V_{M_i}^j \left| h_j^k \right.) \tag{4.9}$$

The sum $\sum_{j=1}^{N} P(M_i, V_{M_i}^j \left| h_j^k \right.)$ can be reduced to the only true hypothesis $P(M_i, V_{cM_i}^j \left| H_j^k \right.)$. In fact, a characteristic view from the request model Q can match only one characteristic view from the model M_i. The characteristic view is chose with the maximum probability.

$$P(M_i \left| Q \right.) = \sum_{k=1}^{K} Max_j (P(M_i, V_{M_i}^j \left| h_j^k \right.)) P(\left| V_Q^k \right| Q) \tag{4.10}$$

Using the Bayes theorem:

$$P(M_i \left| Q \right.) = \sum_{k=1}^{K} Max_j \left(\frac{P(h_j^k \left| V_{M_i}^j, M_i \right.) P(V_{M_i}^j \left| M_i \right.) P(M_i)}{\sum_{i=1}^{N} \sum_{k=1}^{K} P(h_j^k \left| V_{M_i}^j, M_i \right.) P(V_{M_i}^j \left| M_i \right.) P(M_i)} \right) P(\left| V_Q^k \right| Q) \tag{4.11}$$

With $P(M)$ the probability to observe the model M. $P(M_i) = \alpha e^{(-\alpha . |M_i| / \sum_{i=N}^{i=1} |M_i|)}$. Where $\left| M_i \right|$ is the number of characteristic views of the model M_i. α is a parameter to hold the effect of the probability $P(M_i)$. The algorithm conception makes that, the complex is the geometry of the 3-D model, the greater is the number of its characteristic views.

On the other hand $P(V_{M_i}^j | M_i) = 1 - \beta e^{(-\beta . N(V_{rM_i}^j)/320)}$ Where $N(V_{rM_i}^j)$ is the number of views represented by the characteristic view j of the model M. The greater is the number of represented views $N(V_{rM_i}^j)$, the more the characteristic view $V_{cM_i}^j$ is important and the best it represents the 3-D model. The β coefficient is introduced to reduce the effect of the view probability.

The value $P(h_j^k | V_{M_i}^j, M_i)$ is the probability that, knowing that we observe the characteristic view j of the model M_i, this view is the k view of the 3-D query model $Q: P(h_j^k | V_{M_i}^j, M_i) = 1 - D_{(Q^k, h_{v_{M_i}^j})}$ With $D_{(h_q, h_{v_{M_i}^j})}$ the Euclidean distance between the 2-D Zernike descriptors of Q and of the $V_{cM_i}^j$ characteristic view of the 3-D model M_i.

In AVC, the representative views are selected either from the 60 views in the NTU dataset or the 41 views in the ETH dataset. Then, the probabilistic matching is conducted to measure the relevance between each object to the query.

4.2.3 Query View Selection Method(QVS)

For query view selection (QVS) [45], the query views are interactively selected and incrementally increased. In this method a group of views are provided to describe the query object $Q = \{q_1, q_2, ..., q_m\}$ that contains m views. View clustering is first conducted to group views into clusters using a hierarchical agglomerative method[66]. The central view is selected from each cluster to generate a candidate view set $\widetilde{Q}$ with r candidate query views. After generating the candidates, a view graph is constructed based on the relationship among the candidate query views, and a random walk process is employed to select the initial query view. In the view graph, each node denotes a

candidate and the edge between two candidates v_i and v_j is defined as their visual similarity:

$$s_{ij} = \exp(-\frac{d(v_i, v_j)}{\sigma}) \tag{4.12}$$

The transition probability between the i th and the j th candidates is defined as

$$p_{ij} = \frac{s_{ij}}{\sum_{k \neq i} s_{ik}} \tag{4.13}$$

and the random walk process is actually repeating

$$\tau_i^{(t+1)} = \alpha \sum_{k \neq i} \tau_k^{(t)} p_{ik} + (1-\alpha)\tau_i^{(0)} \tag{4.14}$$

until convergence. The candidate $\tilde{q}_K$ with the highest score is selected as the initial query view. If users are not satisfied with the initial retrieval results, a relevance feedback process can be conducted to refine the search results by minimizing its distance to the positive samples and maximizing its distance to the negative samples. A distance metric can be learned for the selected query view with the criterion that minimizes the distance between the selected query view and the relevant objects and maximizes its distance to the irrelevant objects. All the selected query views are combined using the learned weights for next search. The distance estimation of two multi-view 3-D objects in QVS plays the central role in the retrieval process. Given two objects Q and O_i, the distance $d(Q, O_i)$ is defined as

$$d(Q, O_i) = \sum_{j=1}^{K} \rho_j d(\tilde{q}_j, O_i) \tag{4.15}$$

where K is the number of the views of the query object, and $d(\tilde{q}_j, O_i)$ is defined by

$$d(\tilde{q}_j, O_i) = \min_P ((\tilde{q}_j - v_{i,p})^T W_j (\tilde{q}_j - v_{i,p})) \quad (4.16)$$

Here $\tilde{q}_j$ is the j th selected query view of Q , and $v_{i,p}$ is the p th view of the i th object in the database. And W_j is the corresponding weight metric of the Mahalanobis distance metric for $\tilde{q}_j$. In our experiments, we only compare our proposed method with the QVS method without relevance feedback.

In our experiments, we implemented ED, AVC and QVS following the introductions in [39, 23, 45]. For our method, λ is set as 100, and the number K of nearest neighbors is selected as {5; 10; 15; 20}. In 3-D object retrieval experiments, each time one object is selected as the query, and this process is repeated until all the objects are severed as the query once. The average retrieval performance for all the objects in each dataset will be used for comparison.

4.3 Evaluation Criteria

For measuring the 3-D object retrieval performance, evaluation criteria is important to evaluate the different methods. Given a view of the query object, 3-D object retrieval method can be applied to calculate the similarity with other objects in object database. After gotten the final ranking list of retrieved 3-D objects, the outcome should be justified by some evaluation criteria. We employ the following criteria to compare different methods in our experiments. The criteria includes nearest neighbor precision, F-measure, discounted cumulative gain(DCG), and average normalized modified retrieval rank(ANMRR).

4.3.1 The Accuracy of The Nearest Neighbor(NN)

The accuracy of the nearest neighbor (NN) evaluates the retrieval accuracy of the first returned result. NN ranges from 0 to 1, a higher value indicates better performance. The NN is a simple but effective criteria which is defined as (4.17) shows.

4.3.2 F-Measure(F)

F measure[67] is a composite measure of both the precision and the recall for a fixed number of returned results.

The precision of retrieved objects can be defined as follows:

$$\text{precision} = \frac{|\{\text{relevant objects}\} \bigcap \{\text{retrieved objects}\}|}{|\{\text{retrieved objects}\}|} \tag{4.17}$$

Where $\{\text{retrieved objects}\}$ are the retrieved objects given the query, $\{\text{relevant objects}\}$ are the relevant objects for the query(the groundtruth). $|X|$ is the number of objects in X, and $X \bigcap Y$ is the intersection which specify the objects in both sets X and Y. The precision value ranges from 0 to 1. Precision of 1 indicates

all the retrieved results are correct.

The recall evaluates the recall of retrieved objects compared with the groundtruth. The recall is defined as follows:

$$\text{recall} = \frac{|\{\text{relevant objects}\} \bigcap \{\text{retrieved objects}\}|}{|\{\text{relevant objects}\}|} \tag{4.18}$$

The recall value ranges from 0 to 1. Recall of 1 indicates that all the relevant results have been correctly retrieved.

A commonly used performance measure that combines Precision and Recall is the F-measure, also known as the balanced F-score:

$$\text{F} = \frac{2 \times P_K \times R_K}{P_K + R_K} \tag{4.19}$$

Where K is the number of selected top returned results, P_K is the precision for the top K results, and R_K is the recall for the top K results. F ranges from 0 to 1, a higher value indicates better performance. In our experiment K is set as 20. F is defined as

$$F = \frac{2 \times P_{20} \times R_{20}}{P_{20} + R_{20}} \tag{4.20}$$

where P_{20} and R_{20} are the precision and the recall of the top 20 retrieval results, respectively.

4.3.3-Discounted Cumulative Gain(DCG)

DCG[56] measures the ranking performance of the retrieved result list, which is a statistic that gives relevant object high score. DCG works under the assumption that a

user is less like to consider lower results. The DCG value is then defined as

$$DCG[i] = \begin{cases} G[1] & \text{if } i = 1 \\ DCG[i-1] + \dfrac{G[i]}{\log_2(i)} & \text{otherwise} \end{cases} \tag{4.21}$$

where

$$G[i] = \begin{cases} 1 & \text{if the } i\text{th result is correct} \\ 0 & \text{otherwise} \end{cases} \tag{4.22}$$

We explore the behavior of DCG as the relative weight given to highly relevant objects varies. By manipulating this weight we can closely approximate either evaluation by all relevant objects or evaluation by highly relevant objects only. DCG ranges from 0 to 1, a higher value indicates better performance. Assuming the number of all relevant objects is τ, and the number of all objects is n, the maximal DCG is computed as

$$DCG_{\max} = \frac{DCG_n}{1 + \sum_{i=2}^{\tau} \dfrac{1}{\log_2(i)}} \tag{4.23}$$

4.3.4 Average Normalized Modified Retrieval Rank(ANMRR)

ANMRR[57] measures the rank performance given a ranking list, which considers the ranking information of relevant objects among the top-retrieved objects. ANMRR ranges from 0 to 1, and the smaller the value of this measure the better the matching quality of the query is. ANMRR is defined as follows:

To calculate ANMRR, the average retrieval rank $AVR(Q_k)$ for a given k th query Q_k is depicted as:

$$AVR(Q_k) = \sum_{i=1}^{NR(Q_k)} \frac{RANK(i)}{NR(Q_k)} \tag{4.24}$$

where $NR(Q_k)$ is the number of relevant objects for the query Q_k. If the ith result is relevant to the query then $RANK(i)$ is the ranking position; otherwise $RANK(i) = 1.25 \times S_k$. S_k is the top-ranked returned retrievals, where:

$$S_k = \min\{4 \times NR(Q_k), 2 \times GMT\} \tag{4.25}$$

and GMT is the maximal number of relevant objects for all queries.

The modified retrieval rank is:

$$MRR(Q_k) = AVR(Q_k) - \frac{1 + NR(Q_k)}{2} \tag{4.26}$$

Then the modified retrieval rank can be normalized to compute the normalized MRR as follows:

$$NMRR(Q_k) = \frac{MRR(Q_k)}{1.25 \times S_k - \frac{1 + NR(Q_k)}{2}} \tag{4.27}$$

Finally, the average NMRR(ANMRR) can be calculated by averaging the NMRR values over all queries:

$$ANMRR = \frac{\sum_{k=1}^{n} NMRR(Q_k)}{n} \tag{4.28}$$

Where n is the number of queries.

4.4 Experimental Results

Experimental results on the NTU dataset and comparison of different methods are shown in Figure 4.4. As shown in these results, our proposed method, i.e., multi-scale object graph learning (MSOGL), achieves the best performance compared with other methods. Based on NN, the proposed method achieves an improvement of 146.91%, 53.72%, and 12.66%, compared with ED, AVC, and QVS, respectively. In terms of F, the improvement from the proposed methods is 99.40%, 27.80%, and 7.77%, respectively. For DCG, the gain is 55.32%, 8.06%, and 6.01%, respectively. Regarding the ranking performance, the proposed method achieves an improvement of 9.96%, 4.03%, and 3.31% in terms of ANMRR compared to ED, AVC, and QVS, respectively. We can also observe similar results on other criteria.

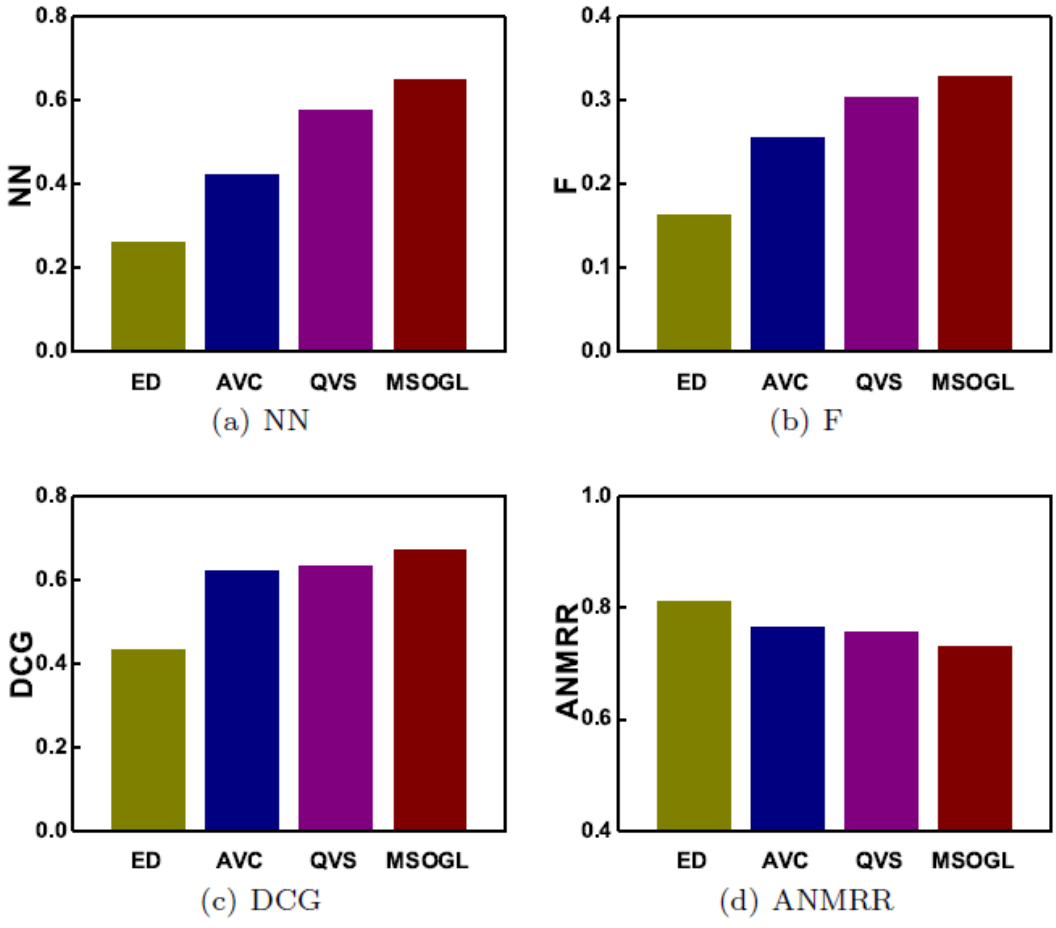

Figure 4.4 Experimental results on the NTU dataset.

Experimental results on the ETH dataset and comparison of different methods are shown in Figure 4.5. As shown in these results, our proposed method achieves the best performance compared with other methods. Based on NN, the proposed method achieves an improvement of 23.8%, 23.8%, and 11.8%, compared with AVC and QVS, respectively. In terms of F, the improvement from the proposed methods is 19.69% and 11.94%, respectively. For DCG, the gain is 10.08% and 4.12%, respectively. Regarding the ranking performance, the proposed method achieves an improvement of 25.1% and 14.3% in terms of ANMRR compared to ED, AVC, and QVS, respectively. Similar results can be observed from other criteria.

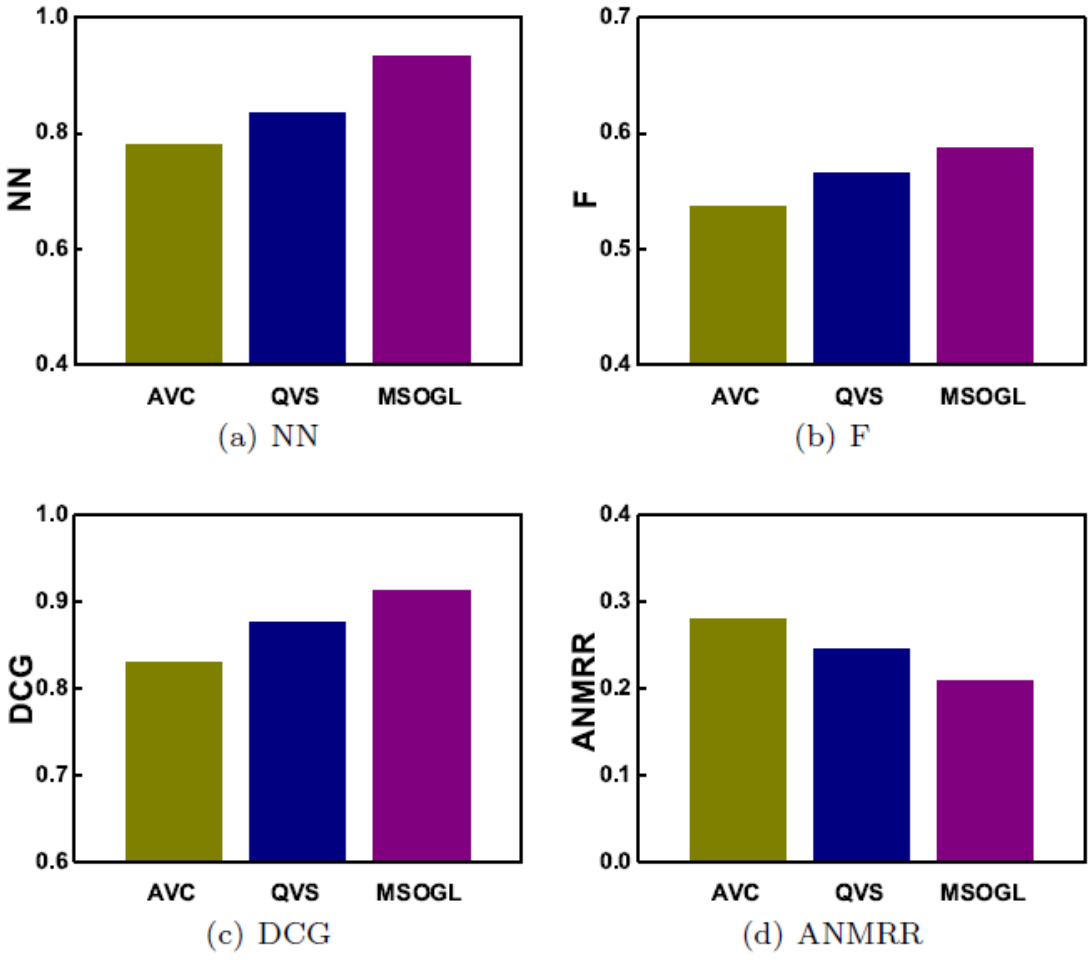

Figure 4.5 Experimental results on the ETH dataset.

As shown in the results, we can have the following observations.

· The proposed achieves the best performance compared with all other compared methods. This satisfactory result can be dedicated to the better formulation of our method on object relationship. The proposed method is able to formulate the connections among objects from multiple scales. Compared with existing formulation methods, our proposed method can be much more robust to object variations. In this way, different connections among objects can be modeled in the hypergraph structure and the learning on hypergraph can explore the optimal relevance among these objects.

· The gain on the ETH dataset is smaller than that on the NTU dataset. This can be dedicated to the high performance of all methods on the ETH dataset.

The better performance can be dedicated to the flexible structure of the proposed method. In the multi-scale relationship formulation, i.e., multi-scale hypergraph construction, the proposed method is able to explore all possible view matching results, which is hard to find the best one in traditional methods. In this way, the proposed method can achieve better performance through the optimal data distribution modeling.

4.5 Analysis

4.5.1 On the Multi-scale Hypergraph Connections

In this subsection, we evaluate the influence of the multi-scale hypergraph connections. The parameter K indicates how many nearest neighbors can be connected by one hyperedge. When K is too large, the connected objects will be much dissimilar. When K is too small, the constructed hyperedge will lose the discriminative performance. Here we evaluate different K values and the combination of multiple K values, which is used in our work. More specifically, we vary K from 5, 10, 15, to 20, and the experimental results on the two datasets are demonstrated in Figure 4.6 and Figure 4.7, respectively.

As shown in these two figures, the performance with single K value varies a lot with respect to the selection of K, and different K values have different performance in the two datasets. When multiple K values are employed, as shown in our method, the overall performance becomes steady and better than that of each single one. These results can demonstrate the effectiveness of our proposed multi-scale hypergraph connection method.

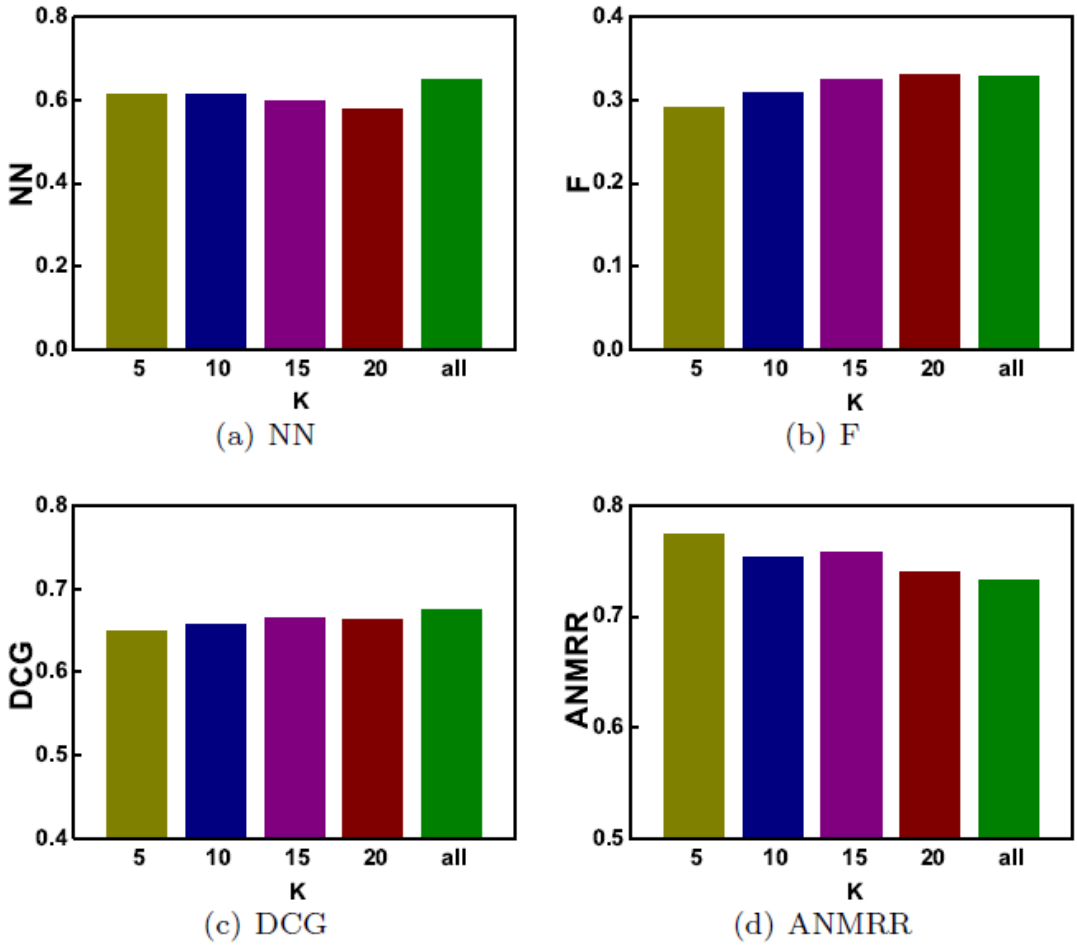

Figure 4.6 Experimental results with respect to different connection numbers on the NTU dataset.

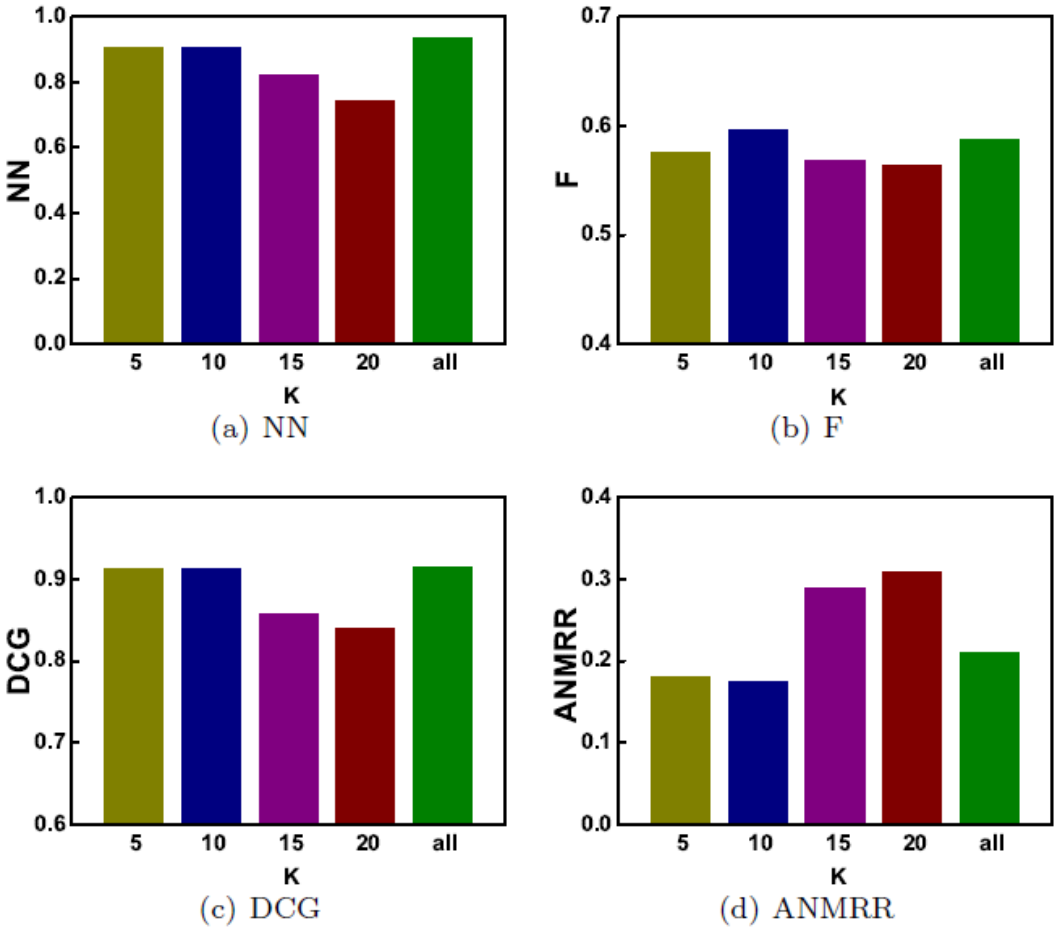

Figure 4.7 Experimental results with respect to different connection numbers on the ETH dataset.

4.5.2 On Parameter λ

In this section, we evaluate the influence of the parameter λ on the 3-D object retrieval performance. λ is a parameter to balance different components in the learning formulation. In this part, we vary λ from 0.001 to 10000, and evaluate the performance of our method. Experimental results on the two datasets are demonstrated in Figure 4.8 and Figure 4.9, respectively.

As shown in these results, when λ is too small, such as 0.001, the performance is not very satisfactory. When λ varies in a large range, such as from 1 to 10000, the performance is steady and satisfactory. These results can demonstrate that the proposed method is robust to the selection of the parameter λ.

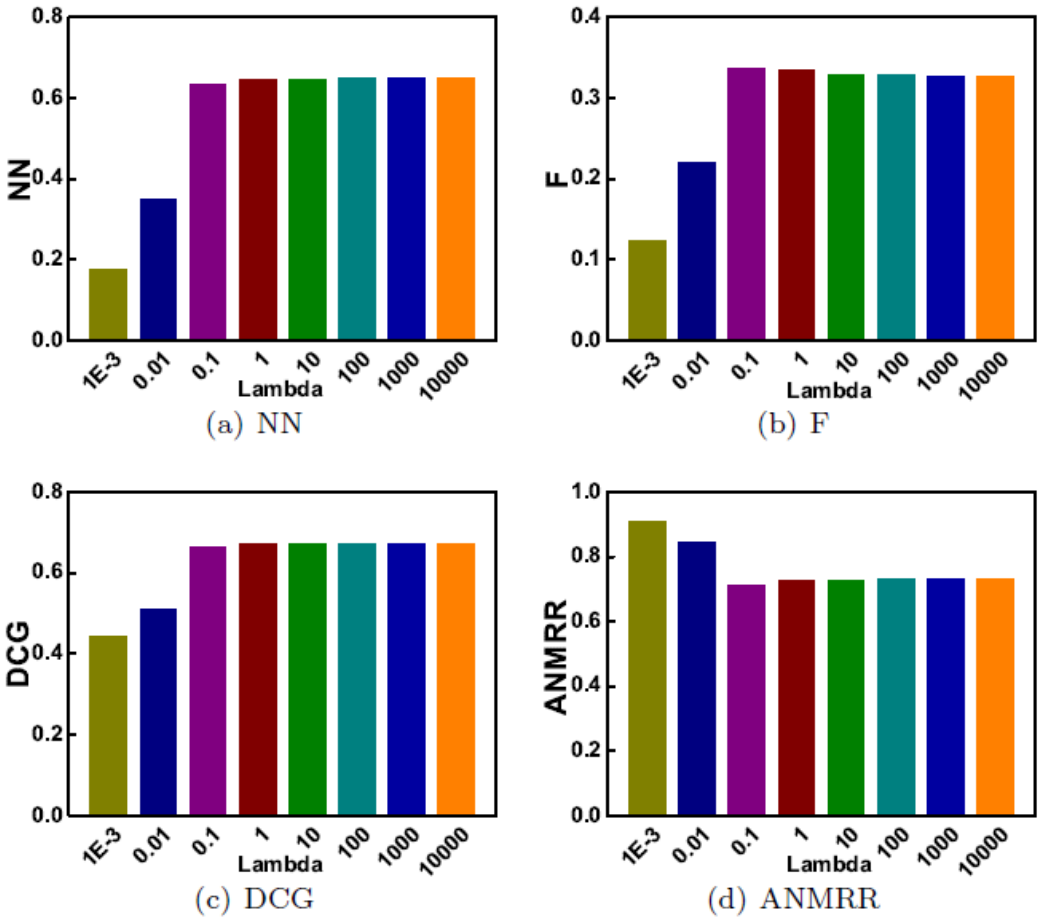

Figure 4.8 Experimental results with respect to different λ values on the NTU dataset.

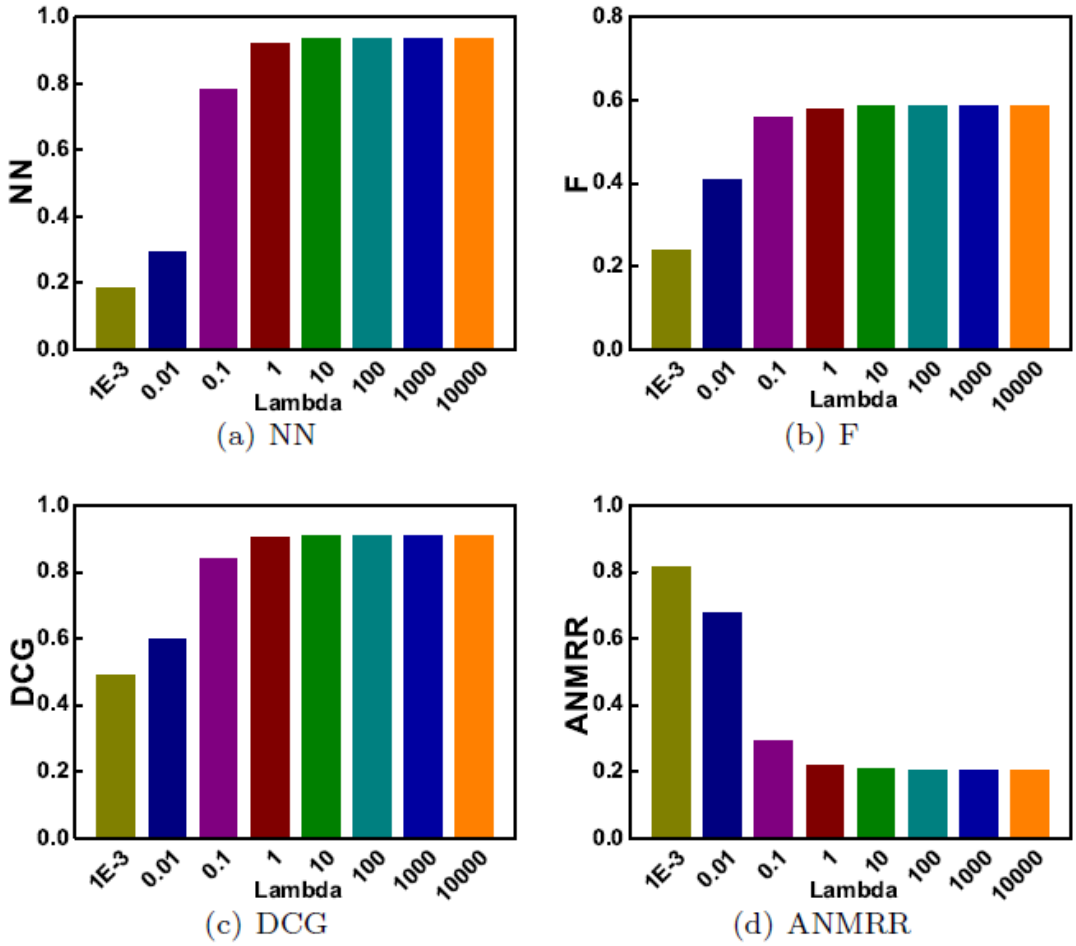

Figure 4.9 Experimental results with respect to different λ values on the ETH dataset.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this paper we are committed to figuring out how to reconstruct a 3-D object or 3-D scene from 2-D image sequence. Two methods are depicted, image-based 3-D modeling method and 3-D object retrieval for combining or revising existing object. For image-based 3-D modeling, many methods have been proposed to solve the problem. However, it is still considered as a laborious task. We recommend the super-pixel into 3-D modeling in order to reduce the calculation cost. The advances in 3-D data acquisition techniques, graphics hardware, and 3-D data modeling and visualizing techniques have led to the growth of 3-D models. This has made the 3-D model retrieval a vital issue. From the retrieved objects, new model or 3-D scene can be combined. We do some research in the area of view-based 3-D object retrieval.

we propose a multi-scale object retrieval algorithm via learning on graph. This work focuses on the view-based object representation and proposes to formulate object relationship in a multi-scale connection hypergraph. Given the multiple views of objects, shape features are firstly extracted and all the objects are formulated in a hypergraph structure. The distance of different views in the feature space is employed to generate the connection in the hypergraph. The learning on hypergraph is conducted to estimate the optimal relevance among these objects, which can be employed for object ranking for the query. Experiments are conducted on two public datasets the National Taiwan University dataset and the ETH dataset. We employ ED, AVC and QVS for comparison. The criteria of NN, F, DCG and ANMRR are employed to evaluate the performance of

different methods. Experimental results and comparisons with the state-of-the-art methods demonstrate the effectiveness of the proposed method.

5.2 Future Work

Although the proposed method has shown satisfactory performance compared with existing methods, there are still several limitations. The plan for the corresponding future works is as follows:

First, how to employ multi-modal data is an important issue. In current work, only the shape feature is employed for view representation. It is noted that different features can have more powerful representation ability for 3-D objects. Therefore, it is important to further investigate how to combine multiple features for views.

Second, most of existing methods only focus on model-to-model or view-to-view retrieval. How to conduct cross modal object retrieval, such as model-to-view and view-to-model, is another important issue.

Third, we have done some research in both image-based 3-D modeling and view-based 3-D object retrieval, however, it is the theories and key technologies research at the present stage. In future, we will develop a workstation including the 3-D modeling and model revising and compiling based on the 3-D object retrieval.

Bibliography

[1] Gunn TG. The mechanization of design and manufacturing. Sci Am 1982; 247:114-30

[2] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein and Richard Szeliski, “A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms,” 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR '06).

[3] A. Treuille, A. Hertzmann and S. Seitz, “Example-based stereo with general BRDFs,” in ECCV, Vol. II, pp. 457 – 469, 2004.

[4] R. Szeliski, “A multi-view approach to motion and stereo,” in CVPR, Vol. 1, pp. 157 – 163, 1999.

[5] V. Kolmogorov and R. Zabih, “Multi-camera scene reconstruction via graph cuts,” in ECCV, Vol. III, pp. 82 – 96, 2002.

[6] C. Zitnick, S.-B. Kang, M. Uyttendaele, S. Winder and R. Szeliski, “High-quality video view interpolation using a layered representation,” ACM Trans. on Graphics, 23 (3): 600 – 608, 2004.

[7] Y. Furukawa and J. Ponce, “Accurate, Dense, and Robust Multiview Stereopsis,” Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR ’07), June 2007.

[8] Bracewell R. The Fourier transform and its applications. 31999. New York: McGraw-Hill; 1986

[9] Del Bimbo, P. Pala., Content-based retrieval of 3-D models, ACM Transactions on Multimedia Computing, Communications, and Applications 2 (1) (2006) 20‑43.

[10] B. Bustos, D. Keim, D. Saupe, T. Schreck, D. Vranic, Feature-based similarity search in 3-D object databases, ACM Computing Surveys 37 (4) (2005) 345‑387.

[11] Y. Zhao, Y. Liu, Y. Wang, B. Wei, J. Yang, Y. Zhao, Y. Wang, Regionbased saliency estimation for 3-D shape analysis and understanding, Neurocomputing 197 (2016) 1‑13.

[12] Y. Gao, Q. Dai, View-based 3-D object retrieval: Challenges and approaches, IEEE Multimedia Magazine 21 (3) (2014) 52‐57.

[13] J. W. H. Tangelder, R. C. Veltkamp, A survey of content based 3-D shape retrieval methods, Multimedia Tools and Applications 39 (2008) 441‐471.

[14] Y. Yang, H. Lin, Y. Zhang, Content-based 3-D model retrieval: A survey, IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews 37 (2007) 1081‐1035. 16

[15] Y. Gao, A. Liu, W. Nie, Y. Su, Q. Dai, F. Chen, Y. Chen, Y. Cheng, S. Dong, X. Duan, et al., 3-D object retrieval with multimodal views, in: Proceedings of the 2015 Eurographics Workshop on 3-D Object Retrieval, Eurographics Association, 2015, pp. 129‐136.

[16] N. Bayramoglu, A. A. Alatan, Comparison of 3-D local and global descriptors for similarity retrieval of range data, Neurocomputing 184 (2015) 13‐27.

[17] Z. Liu, S. Bu, J. Han, Locality-constrained sparse patch coding for 3-D shape retrieval, Neurocomputing 151 (2015) 583‐592.

[18] X. Wang, W. Nie, 3-D model retrieval with weighted locality-constrained group sparse coding, Neurocomputing 151 (2015) 620‐625.

[19] W.-Z. Nie, A.-A. Liu, Y.-T. Su, 3-D object retrieval based on sparse coding in weak supervision, Journal of Visual Communication and Image Representation.

[20] A. E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3-D scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 433‐449.

[21] R. Osada, T. Funkhouser, B. Chazelle, D. Dobkin, Shape distributions, ACM Transactions on Graphic 21 (4) (2002) 807‐832.

[22] E. Paquet, M. Rioux, Nefertiti: A query by content system for three dimensional model and image databases management, Image Vision Computing 17 (1999) 157‐166.

[23] T. Filali Ansary, M. Daoudi, J. P. Vandeborre, A Bayesian 3-D search engine using adaptive views clustering, IEEE Transactions on Multimedia 9 (1) (2007) 78‐88.

[24] R. Ohbuchi, K. Osada, T. Furuya, T. Banno, Salient local visual features for

shape-based 3-D model retrieval, in: Proceedings of IEEE Conference on Shape Modeling and Applications, 2008, pp. 93‑102.

[25] W. Li, G. Bebis, N. Bourbakis, 3-D object recognition using 2-D views, IEEE Transactions on Image Processing 17 (11) (2008) 2236‑2255.

[26] W.-Z. Nie, A.-A. Liu, Z. Gao, Y.-T. Su, Clique-graph matching by preserving global & local structure, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4503‑4510.

[27] J. Tangelder, R. Veltkamp, Polyhedral model retrieval using weighted point sets, International Journal of Image and Graphics 3 (1) (2003) 209‑229.

[28] C. Ip, D. Lapadat, L. Soeger, W. C. Regli, Using shape distributions to compare solid models, in: Proc. ACM Symposium on Solid Modeling and Applications, 2002, pp. 273‑280. 17

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

[30] Y. Gao, M.Wang, D. Tao, R. Ji, Q. Dai, 3-D object retrieval and recognition with hypergraph analysis, IEEE Transactions on Image Processing 21 (9) (2012) 4290‑4303.

[31] Y. Gao, M. Wang, R. Ji, X. Wu, Q. Dai, 3-D object retrieval with Hausdorff distance learning, IEEE Transactions on Industrial Electronics 61 (4) (2014) 2088‑2098.

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

[33] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The Princeton shape benchmark, in: Proceedings of Shape Modeling International, 2004, pp. 167‑178.

[34] Y. Gao, J. Tang, R. Hong, S. Yan, Q. Dai, N. Zhang, T. Chua, Camera constraint-free view-based 3-D object retrieval, IEEE Transactions on Image Processing 21 (4) (2012) 2269‑2281.

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

[36] Y. Gao, Q. H. Dai, N. Y. Zhang, 3-D model comparison using spatial structure circular descriptor, Pattern Recognition 43 (3) (2010) 1142–1151.

[37] D. Vranic, An improvement of rotation invariant 3-D shape descriptor based on functions on concentric spheres, in: Proceedings of IEEE International Conference on Image Processing, 2003, pp. 757–760.

[38] D. Y. Chen, X. P. Tian, Y. T. Shen, M.Ouhyoung, On visual similarity based 3-D model retrieval, Computer Graphics Forum 22 (3) (2003) 223–232.

[39] J. L. Shih, C. H. Lee, J. T. Wang, A new 3-D model retrieval approach based on the elevation descriptor, Pattern Recognition 40 (2007) 283–295.

[40] S. Mahmoudi, M. Benjelloun, T. Filali Ansary, 3-D objects retrieval using curvature scale space and zernike moments, Journal of Pattern Recognition Research 6 (1).

[41] A. Ad´ an, P. Merch´ an, S. Salamanca, 3-D scene retrieval and recognition with depth gradient images, Pattern Recognition Letters 32 (9) (2011) 1337– 1353. 18

[42] Y. Zhang, F. Jiang, S. Rho, S. Liu, D. Zhao, R. Ji, 3-D object retrieval with multi-feature collaboration and bipartite graph matching, Neurocomputing 195 (2016) 40–49.

[43] M. Ji, Y. Feng, J. Xiao, Y. Zhuang, X. Yang, J. J. Zhang, Efficient semisupervised multiple feature fusion with out-of-sample extension for 3-D model retrieval, Neurocomputing 169 (2015) 23–33.

[44] H. H. Mohamed, S. Belaid, Algorithm boss (bag-of-salient local spectrums) for non-rigid and partial 3-D object retrieval, Neurocomputing 168 (2015) 790–798.

[45] Y. Gao, M. Wang, Z. Zha, Q. Tian, Q. Dai, N. Zhang, Less is more: Efficient 3-D object retrieval with query view selection, IEEE Transactions on Multimedia 11 (5) (2011) 1007–1018.

[46] R. Ohbuchi, T. Furuya, Accelerating bag-of-features sift algorithm for 3-D model retrieval, in: Proceedings of the SAMT 2008 Workshop on Semantic 3-D Media, 2008, pp. 22–30.

[47] R. Ohbuchi, T. Furuya, Scale-weighted dense bag of visual features for 3-D model

retrieval from a partial view 3-D model, in: Proceedings of IEEE ICCV 2009 workshop on Search in 3-D and Video (S3-DV), 2008.

[48] T. Furuya, R. Ohbuchi, Dense sampling and fast encoding for 3-D model retrieval using bag-of-visual features, in: Proceedings of ACM International Conference on Image and Video Retrieval, 2008.

[49] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91‐110.

[50] A. Khotanzad, Y. H. Hong, Invariant image recognition by zernike moments, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (5) (1990) 489‐497.

[51] W. Y. Kim, Y. S. Kim, A region-based shape descriptor using Zernike moments, Signal Processing: Image Communication 16 (1-2) (2000) 95‐102.

[52] Y. Huang, Q. Liu, S. Zhang, D. Metaxas, Image retrieval via probabilistic hypergraph ranking, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3376‐3383.

[53] D. Zhou, J. Huang, B. Schokopf, Learning with hypergraphs: Clustering, classification, and embedding, in: Proceedings of Advances in Neural Information Processing Systems, 2007, pp. 1601‐1608.

[54] Y. Huang, Q. Liu, D. Metaxas, Video object segmentation by hypergraph cut, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2009, pp. 1738‐1745. 19

[55] B. Leibe, B. Schiele, Analyzing appearance and contour based methods for object categorization, in: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2003, pp. 409‐415.

[56] K. Jarvelin, J. Kekalainen, Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems 20 (4) (2002) 422‐446.

[57] Description of core experiments for MPEG-7 color/texture descriptors, in: ISO/MPEGJTC1/SC29/WG11 MPEG98/M2819, MPEG video group, 1999. 20

[58] Xiaofeng Ren and Jitendra Malik, "Learning a Classification Model for Segmentation" In ICCV 2003

[59] J.Shi and J.Malik. “Normalized cuts and image segmentation”, IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8):888-905, August 2000

[60] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows," TPAMI, vol. 31, no. 12, pp. 2290-2297, 2009.

[61] Noah Snavely, Steven M. Seitz, Richard Szeliski. Photo Tourism: Exploring image collections in 3-D. ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006), 2006.

[62] Heloise Hse and A. Richard, “ Newton Sketched Symbol Recognition using Zernike Moments”,

[63] B. Leng, Z. Qin, X. Cao, T. Wei, Z. Zhang, Mate: a visual based 3-D shape descriptor, Chinese Journal of Electronics 18 (2) (2009) 291‑296.

[64]B. Leng, Z. Xiong, Modelseek: an effective 3-D model retrieval system, Multimedia Tools and Applications 51 (3) (2011) 935‑962.

[65] Y. Gao, J. Tang, H. Li, Q. Dai, N. Zhang, View-based 3-D model retrieval with probabilistic graph model, Neurocomputing 73 (10-12) (2010) 1900‑1905.

[66] M. Steinbach, G. Karypis, and V. Kumar, “A comparison of document clustering techniques,” in *Proc. KDD Workshop Text Mining*, 2000.

[67] Van Rijsbergen CK, Information retrieval, London: Butterworths; 1973

Acknowledgements

I would like to express my deep gratitude to Prof. Tsuyoshi Yamamoto for his guidance and support as my adviser.

I would like to express my deep gratitude to Assoc. Prof. Yoshinori Dobashi for his guidance and support.

I would like to express my deep gratitude to the associate examiners Prof. Kenji Araki, Prof. Miki Haseyama and Prof. Yuji Sakamoto.

Thanks to my mother for her love and support.

Thanks to all the people who have helped me in the past.

Research Achievements

Journal Paper

Yongsheng Zhang, Tsuyoshi Yamamoto, Yoshinori Dobashi “Multi-Scale Object Retrieval via Learning on Graph from Multimodal Data” Neurocomputing, DOI:10.1016/j.neucom.2016.05.053

International Conferences

Yongsheng Zhang, Tsuyoshi Yamamoto, Yoshinori Dobashi “A Prototype of A 3-D Scene Reconstruction System Based on Super-Pixel Segmentation” International Symposium on Multimedia and Communication Technology 2013 (ISMAC 2013) pp.172-175(2013-02)

Yongsheng Zhang, Tsuyoshi Yamamoto, Yoshinori Dobashi “An Application of Super-pixel for Image-based 3-D Reconstruction” International Conference on Engineering, Technology and Applied Science 2016(ICETA2016) volume 2, pp.228-235(2016-04)