

**EFFICIENT RETRIEVAL AND CATEGORIZATION FOR
3D MODELS BASED ON BAG-OF-WORDS APPROACH**

WANG YAN

NATIONAL UNIVERSITY OF SINGAPORE

2013

**EFFICIENT RETRIEVAL AND CATEGORIZATION FOR 3D
MODELS BASED ON BAG-OF-WORDS APPROACH**

WANG YAN

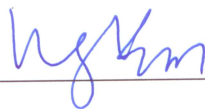
(B.Eng)

**A THESIS SUBMITTED
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SINGAPORE
2013**

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has not been submitted for any degree in any university previously.



WANG YAN

12 August 2013

ACKNOWLEDGEMENTS

First of all, I would like to the most sincere gratitude to my supervisors Prof. Jerry Fuh Ying Hsi and Prof. Lu Wen Feng, not only for their enormous support and guidance, but also for their kindly encouragement during times of difficulties along with my doctoral studies. This thesis cannot be completed without their timely feedback and careful revision.

I would also like to thank Prof. Wong Yoke San for his intensive discussions and many valuable suggestions throughout group meetings together. Many thanks also go to Prof. Cheong Loong Fah from the Department of Electrical and Computer Engineering, for his many useful suggestions, critical comments and encouragement during my second year of PhD study. I wish to thank Prof. Zhang Yunfeng for his comments and suggestions during my qualifying examination.

I would like to also thank the National University of Singapore for providing the research scholarship to support my doctoral studies.

My gratitude also goes to all the members in the labs of manufacturing group, especially Dr. Zhu Kunpeng, Dr. Wang Jinling, Dr. Wang Yifa, Dr. Li Min, Dr. Zheng Fei, Dr. Wang Xue, Ms. Zhong Xin and many others, for their encouragement, support

and creating a friendly environment. I wish thank all of my friends for their support and care.

Last, but not least, I would like to express my hearty gratitude to my parents and my husband for their love and continuous support and understanding.

Table of Contents

ACKNOWLEDGEMENTS	i
SUMMARY	vi
LIST OF FIGURES.....	ix
LIST OF TABLES.....	xi
Chapter 1 INTRODUCTION	1
1.1 Background	1
1.2 Research Motivation.....	2
1.3 Research Objectives	4
1.4 Organization of this Thesis.....	6
Chapter 2 LITERATURE REVIEW.....	7
2.1 Introduction	7
2.2 3D Model Retrieval based on Visual Similarity	10
2.3 3D Model Retrieval using Bag-of-Words Model	14
2.4 3D Model Categorization	21
2.5 Summary	22
Chapter 3 FRAMEWORK FOR RETRIEVAL AND CATEGORIZATION OF 3D MODELS USING BAG-OF-WORDS MODEL REPRESENTATION	24
3.1 Overview of this Research.....	24
3.2 Pose Alignment and Depth Image Extraction.....	27
3.2.1 Pose Alignment.....	27
3.2.2 Depth Image Extraction.....	30
3.3 Bag-of-Words Model Representation	32
3.3.1 Codebook Generation and Model Representation.....	32
3.3.2 Similarity Distance Comparison.....	33
3.4 Evaluation Measures for 3D Model Retrieval.....	34
3.5 Experimental Datasets.....	36
3.5.1 Purdue Engineering Shape Benchmark	36
3.5.2 Modified CAD dataset.....	38
3.5.3 NIST Generic Shape Benchmark	38
3.5.4 SHREC 2009 Partial Dataset.....	39
3.6 3D Model Retrieval Case Study	40

3.7 Summary	41
Chapter 4 MODIFIED DENSE SAMPLING AND MULTI-SCALE DENSE SAMPLING OF LOCAL FEATURES USING SIFT DESCRIPTION FOR 3D MODEL RETRIEVAL.....	43
4.1 Introduction	43
4.2 Scale Invariant Feature Transform (SIFT) Algorithm for Feature Detection and Description.....	45
4.3 Modified Dense Sampling and PHOW Sampling for Feature Extraction	47
4.5 Results and Discussions	51
4.4.1 Retrieval Results on ESB	52
4.4.2 Retrieval Results on NIST Generic Shape Benchmark	58
4.4.3 Retrieval Results on SHREC 2009 Partial Dataset.....	62
4.5 Summary	65
Chapter 5 REGION-BASED FEATURE DETECTION AND REPRESENTATION FOR 3D MODEL RETRIEVAL	66
5.1 Introduction	66
5.2 Region Speeded-Up Robust Feature (RSURF) and Histogram of Oriented Gradients (HOG) Descriptor	67
5.3 Results and Discussions	73
5.4 Summary	81
Chapter 6 LARGE-SCALE 3D MODEL CATEGORIZATION USING MULTI-CLASS SVM WITH LINEARLY APPROXIMATED KERNEL	82
6.1 Introduction	82
6.2 3D Model Categorization with Multi-class Kernel SVM.....	83
6.2.1 Bag-of-Words Representation for Categorization of 3D Models	83
6.2.2 Non-linear Kernel SVM Approximated by Linear Homogeneous Feature Maps	84
6.2.3 Multi-class SVM categorization	87
6.3 Results and Discussions	88
6.3.1 Classification Results on the NIST Generic Shape Benchmark	90
6.3.2 Classification Results on the Modified CAD Dataset	92
6.4 Summary	95
Chapter 7 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK.....	96
7.1 Conclusions	96
7.2 Recommendations for Future Works	99
7.2.1 Extension for an Improved Bag-of-Words Representation.....	99
7.2.2 Extension for an Incremental Bag-of-Words Learning for Classification	100
PUBLICATIONS	102

REFERENCES.....	103
Appendix A Lists of the Modified CAD Dataset.....	108

SUMMARY

Efficient retrieval and categorization of 3D models are in urgent need due to the rapid proliferation of 3-Dimensional (3D) digital models. Recently, bag-of-words approach based on the visual similarity for 3D model retrieval has received a lot of attention for its superior performance and scalability to various input formats. It represents 3D model as histogram of visual words according to a codebook generated from local features extracted from 2D depth images. However, existing salient feature extraction methods not only are time-consuming, but also require large computation and storage capacity. Besides, very little research work has addressed 3D model categorization problem compared to large amount of work for the 3D model retrieval tasks. The categorization of 3D models is of great importance because when the database is huge, it is impossible to compare the query example with all target models, so there is a need for a mechanism to classify the query models into categories. This research aims at achieving two main objectives. The first objective is to develop more discriminative but computationally less expensive feature extraction methods. The second objective is to develop a 3D model categorization system which is very little addressed in the past. Both of the two objectives are achieved based on the bag-of-words framework.

Firstly, a modified dense sampling and multi-scale dense (MSD) sampling strategy of local salient features are proposed to extract features from depth images of 3D models.

Dense sampling is to extract features on uniformly distributed grids and MSD sampling is to extract features at multiple scales on the same grids as dense sampling. The proposed sampling strategies extract local features over the full range of the depth images rendered from the 3D model and therefore more suitable for the 3D model description. With a flat window to substitute circular Gaussian window, the feature extraction speed for the proposed sampling strategies are in an order of magnitude faster than the original Scale Invariant Feature Transform (SIFT) detection. In combination with bag-of-words models, the proposed sampling strategies have shown superior performance over the original salient SIFT sampling.

Secondly, two region feature descriptors Region Speeded-Up Robust Features (RSURF) and Histogram of Oriented Gradients (HOG) features are proposed for 3D model description. The proposed RSURF and HOG features extract features on uniform grids over a local region. As they extract features with a pre-assumed scale and location, the proposed region-based feature detections are much faster and of lower dimension than the salient point detection. The region size, number of orientation bins and coarse spatial binning will influence the descriptiveness and distinctness of the region-based feature descriptor together. The proposed region feature descriptors are used as inputs for bag-of-words model and show a much better accuracy than salient feature description for the 3D model retrieval tasks.

Thirdly, a 3D model categorization scheme based on the bag-of-words representation

is proposed using kernelized multi-class SVM for classification. The chi-square kernel and histogram intersection kernel approximated by linear homogeneous map are adopted as they are inherently suitable for the histogram-based shape representation. The linearly approximated kernel SVM not only show significant improvement than the original SVM, but are also very efficient to compute. Example of the proposed 3D model categorization system will be given for classification of query examples on public shape benchmark.

LIST OF FIGURES

Figure 3.1 Overview of Retrieval and Categorization of 3D Models based on Bag-of-words Representation.	25
Figure 3.2 Procedures to compute bag-of-words representation for 3D models.	26
Figure 3.3 6-view camera positions with respect to the object.	31
Figure 3.4 Examples of CAD models from ESB dataset.	37
Figure 3.5 Partial and Range query models for SHREC 09 Partial Dataset.	40
Figure 4.1 Flow chart of sampling strategies of local features for bag-of-words model representation.	44
Figure 4.2 SIFT descriptor of 4×4 regions and 8 orientations in each region [43].	46
Figure 4.3 (a) SIFT features extracted from depth image of CAD part model, (b) Corresponding features, (c) SIFT features extracted from range image of 3D flying bird model, (d) Corresponding features.	47
Figure 4.9 Influence of distance metric for original SIFT sampling.	56
Figure 4.10 Retrieval examples of sampling methods: (a) original SIFT sampling, (b) modified dense sampling, and (c) MSD sampling.	56
Figure 4.11 Retrieval accuracy using SIFT, modified dense and MSD sampling.	57
Figure 4.12 Influence of codebook size for 6-view SIFT sampling.	59
Figure 4.13 Influence of codebook size for 6-view modified dense sampling.	60
Figure 4.14 Influence of codebook size for 6-view MSD sampling.	60
Figure 4.15 Overall comparison of precision-recall results for 6-view SIFT sampling, modified dense sampling and MSD sampling.	61
Figure 4.16 NN, FT, ST, E-measure and DCG measures for 6-view SIFT sampling, modified dense sampling and MSD sampling.	61
Figure 4.17 DCG measures for 6-view SIFT sampling, modified dense sampling and MSD sampling on SHREC 2009 Partial Dataset.	63
Figure 4.18 Overall comparison of precision-recall results for 6-view SIFT sampling, dense sampling and MSD sampling with optimal codebook size.	64
Figure 5.1 Haar wavelet responses for four patterns of image intensity changes [83].	69
Figure 5.2 Illustration of DSURF feature representation based on Haar wavelet responses of a 4×4 sub-region centered at the interest point.	70
Figure 5.3 Integral images makes the computation of summation of image gradients within the region ACDB is simple as subtracting the integral value at point B and C from point D, and plus the value at point A [84].	71
Figure 5.4 Convolution of depth image with 1D mask (-1, 0, 1).	72
Figure 5.5 DCG of RSURF features on modified CAD dataset for different codebook size K.	76
Figure 5.6 DCG of RSURF features on NIST generic shape benchmark for different codebook size K.	77

Figure 5.7 DCG of HOG features on modified CAD dataset for different codebook size K ...78

Figure 5.8 DCG of HOG features on NIST generic shape benchmark for different codebook size K 78

Figure 5.9 Precision recall curve for proposed region-based RSURF and HOG features compared to salient features SIFT and SURF on modified CAD dataset..... 79

Figure 5.10 Precision recall curve for proposed region-based RSURF and HOG features compared to salient features SIFT and SURF on NIST generic shape benchmark. 80

Figure 6.1 Categorization procedures of 3D models using bag-of-words representation. 84

Figure 6.2 Illustration of the multi-class classification problem [71]..... 87

Figure 6.3 Convergence of SVM energy for training..... 89

LIST OF TABLES

Table 3.1 List of 40 types of models for SHREC generic shape benchmark.....	39
Table 4.1 Feature Extraction Time (s)	53
Table 4.2 NN, FT, DCG, ST, E-measure, and MAP for 6-view SIFT sampling, dense sampling and MSD sampling with optimal codebook size.	63
Table 5.1 RSURF feature with different region size and number of sub-regions.....	74
Table 5.2 Feature extraction time (s) for RSURF vs. SURF feature detection.....	75
Table 5.3 Feature extraction time (s) for HOG feature detection	75
Table 5.4 Other evaluation measures for proposed features vs. SIFT and SURF on modified CAD dataset	80
Table 5.5 Other evaluation measures for proposed features vs. SIFT and SURF on.....	81
Table 6.1 Classification accuracy of SVM without kernel for different regularization parameters C	90
Table 6.2 Classification accuracy of histogram intersection kernel for different regularization parameter C and feature dimension.	91
Table 6.3 Classification accuracy of Chi-square kernel for different regularization parameter C and feature dimension.....	91
Table 6.4 Overall comparisons for optimal configuration for no kernel, HI and chi2 kernel...	92
Table 6.5 Classification accuracy of SVM without kernel for different regularization parameters C	93
Table 6.6 Classification accuracy of histogram intersection kernel for different regularization parameter C and feature dimension.	94
Table 6.7 Classification accuracy of Chi-square kernel for different regularization parameter C and feature dimension.....	94
Table 6.8 Overall comparisons for optimal configuration for no kernel, HI and chi2 kernel...	95

Chapter 1 INTRODUCTION

1.1 Background

The number of 3-Dimensional (3D) digital models has been rapidly growing due to the advancement in fields of 3D data acquisition, geometric modeling and visualization. A large number of 3D models are heavily involved in various applications such as augmented reality [1], Computer-Aided Design (CAD) [2], cultural heritage [3] and etc. With the explosion of 3D models both at Internet and in domain specific databases, there is an urgent need for automatic reuse and management of these models. One challenging issue is to develop an efficient and effective retrieval and categorization scheme to find similar models. Automatic retrieval and categorization of 3D models will not only facilitate the reuse of existing digital contents, but also save a lot of time and human efforts to create new models and save costs for design and development.

Content-based 3D model similarity search is to use the 3D model itself as query to match with existing models in a dataset. The similarity of 3D model defined in this thesis is purely based on shape, although similarity in other forms, e.g. functional similarity, is also of interest for different applications. In the content-based 3D model similarity search, both of the query and target models are represented as shape descriptors computed automatically such that similarity distance between similar models is small in the high-dimensional feature space. The shape descriptor is required to be both representative and discriminative in order to better characterize the 3D models for the

similar class and differentiate the models from different classes.

When the number of target models is small, retrieval can be achieved by one-to-one comparison between query model and target models. However, when the amount of target models hits a large number, one-to-one comparison becomes unaffordable. Therefore, one-to-class comparison scheme is needed which could reduce the number of comparisons only related to the number of categories of existing models. In this thesis, the one-to-one comparison scenario is named as 3D model retrieval and the one-to-class comparison procedure is called 3D model categorization. The input format of 3D models in this thesis is polygonal mesh, however, the methods proposed could be easily extended to any format of object, including 2D sketches, range scans, point clouds etc.

1.2 Research Motivation

Visual similarity based methods have received appealing retrieval accuracy than other methods for 3D model retrieval tasks. Among them, bag-of-words methods are most attractive not only because of their retrieval accuracy, but also of less storage space compared with other view-based methods. This is because only the codebook and histogram of visual words are kept without the details of descriptors for each model after the codebook generation. Due to these advantages, this thesis employs the bag-of-words representation of 3D models. However, there are two limitations to be overcome for existing approaches of bag-of-words representation of 3D models in order to develop efficient algorithms to search for similar 3D models in a large-scale dataset in this thesis.

Firstly, local salient features, such as Scale Invariant Feature Transform (SIFT) features, are often extracted for further shape description. These scale and rotation invariant salient features are often detected along corners and sharp changes. They might be more suitable for tasks like object recognition, where a number of notable features are extracted to build correspondence between two models. However, salient features often do not cover the whole content of the views of a 3D model, thus not descriptive enough for the representation of the 3D models. Therefore, there is a need to develop new feature descriptors which are more representative and discriminative than the previously proposed salient feature descriptors.

Secondly, when the amount of 3D models grows large to a certain extent, there are at least two practical issues to be considered for the 3D model similarity comparison. One is regarding the computation cost and storage. Although SIFT features are very descriptive in terms of saliency, it is of very high dimension at 128. Some work proposed to use 42 views of depth images, and extract around 1k features per image, the storage requirement becomes unaffordable. Therefore, there is a need to develop some feature detection and description methods, which not only need less storage space, but also more representative than the salient features. Another issue is the affordable computational expense for the 3D model comparison. Existing one-to-one comparison of models is too time consuming, and sometimes not practical for large-scale problems. Hence, a scalable system for large-scale 3D model comparison

system needs to be devised.

1.3 Research Objectives

From the above research motivation in Section 1.2, the objectives of this research are as follows:

- To develop feature sampling strategies which are descriptive enough for bag-of-words representation of 3D model. The sampled features should represent the 3D model by covering the full content of 3D models. Feature sampling parameters, such as scales and sampling step, will be investigated to find the optimal configurations for higher retrieval accuracy. The proposed feature sampling strategies should also compute the features in a much faster fashion.
- To develop two region-based feature descriptors which not only are compact in representation, but also simple and fast to compute. The Region-SURF (RSURF) feature is to use the SURF-like descriptor sum Haar wavelet responses over local image regions for shape representation. The Histogram of Oriented Gradients computes the derivative of a depth image and votes the gradients into orientation bins.
- To develop an algorithm for categorization of large-scale 3D models. A multi-class

Support Vector Machines (SVM) will be exploited for the categorization scheme. This learning-by-example approach obtain classifiers from existing models and assign a query example to a class of similar models without explicit comparison with all models in a dataset. As the 3D models are represented using the bag-of-words model, efficient non-linear kernels, such as the histogram intersection kernel and chi-square kernel that are suitable for the histogram-based data, can be incorporated with the SVM. The comparisons between the query model and target models are reduced from the total number of target models to the number of classes of the target models.

The proposed work of this thesis may have significant impacts for large-scale similarity comparison of 3D models. The proposed feature detection methods are not only simple and fast to compute than the salient features, but also more representative and discriminative. They require less storage space and computational power than the SIFT feature detection, and therefore more affordable for the generation of codebooks using K-means clustering. The proposed 3D model categorization system makes the large-scale comparison of 3D models practical. It may potentially handle thousands of 3D models and large number of categories thanks to the indirect one-to-class comparison and bridge the gap between single 3D model recognition and generic recognition. The proposed work has accommodated the needs of managing 3D models with a rapid growing amount.

1.4 Organization of this Thesis

This chapter presents the background and motivations of this research. A comprehensive literature review for content-based 3D model retrieval and categorization is given in Chapter 2. Chapter 3 outlines the framework of this thesis. The procedures of using bag-of-words approach to represent 3D models are also presented. Standard evaluation measures and four public available datasets for 3D model retrieval are also introduced in chapter 3. In Chapter 4, the modified dense sampling and multi-scale dense sampling of local features using SIFT description are proposed to incorporate with bag-of-words representation to improve the retrieval efficiency of 3D models. Chapter 5 proposes two region based descriptors, which are not only simpler in representation, but are also more discriminative for bag-of-words model based 3D model retrieval. In chapter 6, a multi-class SVM 3D model categorization system is proposed for the matching of large-scale 3D models. The histogram intersection kernel and chi-square kernel approximated with linear homogeneous maps are combined with the multi-class SVM have showed to improve the classification accuracy. The last chapter concludes this thesis and proposed recommendations for future work.

Chapter 2 LITERATURE REVIEW

2.1 Introduction

Recent advancements in techniques for modeling, digitizing and visualizing 3D models have led to an explosion in the number of available 3D models on the Internet and in domain-specific databases. Therefore, it is highly desirable to develop 3D model matching and retrieval algorithms to automatically annotate, recognize and classify 3D models in large-scale databases. In recent two decades, researchers in field of computer graphics and vision, geometrical modeling and pattern recognition, have conglomerated and dedicated enormous efforts to develop effective and efficient similarity search and retrieval algorithms. Several literature surveys can be found in [4-7]. According to the surveys, the existing 3D model retrieval approaches can be roughly categorized into four categories: statistical-based, spatial map-based, topology-based and view-based methods.

The statistical-based methods extract geometrical information of the object and then bin the measurements into histogram representation. These kinds of methods are generally easy to implement but not discriminative enough. Horn [8] first introduced the extended Gaussian Images to map the orientations of surface normal onto a Gaussian sphere and vote each triangle based on the normal direction. Other geometric measures, e.g., normal distance of the surface points to the object origin [9], are further investigated. Ankerst et al.

[10] introduced an intuitive representation of adaptive similarity distance function into spatial histograms. Ohbuchi et al. [11] partitioned the object into slices along the principle axes of the model and proposed the representation to extract the moment of inertia, the average distance of surface from axis, and the variance of distance of surface from the axis for each slice. The most popular work of this paradigm is shape distributions, proposed by Osada et al. [12]. The idea is simple, which is to measure distance between randomly sampled surface points, angle, area or volume properties and quantize them into histogram bins. The similarity is evaluated using earth mover's distance. Many extensions have been made based on shape distributions, for example generalized shape descriptor (GSD) [13] and shape distributions for solid CAD models [14].

Spatial map based methods represent the shape with its entries corresponding to physical locations of an object. Spherical representations are the most natural and common representations for 3D models. This representation is in general not invariant to rotations; therefore, a pose normalization step is critical to the exact description of the shape. Vranic et al. [15-17] proposed a seminal series of work to extract the coefficients of intersected ray extents with the sphere a 3D model and apply Spherical Fast Fourier Transform, known as Spherical Harmonic (SH) descriptors. SH descriptors can provide the multi-resolution representation of the shape and rotation invariant with respect to the z-axis. Kazhdan et al. [18] proposed to do pose alignment for the polygonal model first and then voxelise it in order to be more robust to local changes and artifacts. The resulted descriptor is not only rotation invariant but also has a lower dimensionality of the feature

vector. Novotni et al. [19] further proposed to use 3D Zernike moments computed as projection of the function defining the object as a set of orthonormal functions. This generalization considers the full volumetric information. The more compact 3D Zernike descriptors can capture extensions as a projection of the function onto a set of orthonormal basis functions within the unit ball. Papadakis et al. [20] decomposed a 3D model into set of spherical functions represented by intersections of emanating rays with the surfaces of 3D model. Later, the Generalized Radon Transform [21] and Spherical Trace Transform [22] have been applied in order to achieve better performance. The spatial map based descriptors basically show better results than some coarser histogram and distribution based approaches. These methods are intuitive in the meaningful interpretations with respect to the model's geometry but one main drawback is that only global information is encoded without specifying the relations between parts and features. Partial matching and deformable structures are not supported with these approaches.

The topology based methods build a graph according to the geometry meaning of a 3D shape, showing how parts are linked together. It is more intuitive to encode both the geometrical and topological shape properties, but is also more complex and difficult to obtain and index in general. For instance, Hilaga et al. [23] proposed topology matching to automatically calculate similarity between polyhedral models by comparing Multiresolutional Reeb Graphs (MRG). The MRG is computed via geodesic distance function to get the skeletal and topological structure of a 3D shape. Tung and Schmitt [24, 25] extended the Reeb graph with geometrical attributes for a more flexible

multiresolutional representation, known as augmented Reeb Graph. The inherent drawbacks of topology-based methods are it is too computational expensive for real applications and the resulted representations are very sensitive to noises and part perturbations. Therefore less work has been done in this area.

As this thesis mainly focused on visual-similarity based methods, and especially using bag-of-words approach, the visual similarity based approaches and that based on bag-of-words model are reviewed in more detail in the following sections.

2.2 3D Model Retrieval based on Visual Similarity

View-based methods are based on the fact that similar objects also look similar from different viewing angles. It not only opens up the way to use 2D query interfaces in typical 3D model retrieval systems, but also makes it possible to use the substantial amount of existing work from computer graphics and computer vision.

Earlier work on view-based methods, for instance [26, 27] , proposed the so-called shock graph descriptor which stores a number of views of a 3D model. Clustered views of the object are then represented in the shock graph. However, effective shock graph indexing is not addressed in these approaches and reduces the problem to a linear search over all views in the database.

This first prominent work based on visual similarity is Light Field Descriptor (LFD) by Chen et al. [28], which proposed to describe the objects by silhouettes from ten uniformly distributed viewing angles of a sphere. Zernike moments and Fourier transforms are applied to the silhouettes and the dissimilarity is determined by summing up the similarity scores over all corresponding views. This approach has won the superior precision-recall accuracy over all other matching methods till its publication. However, LFD still suffers the following drawbacks: (i) only silhouettes -the external outline of the geometry, are encoded, and inner structures are not considered; (ii) no rotation alignment is applied, therefore by choosing N views of one model, total $(N \times (N - 1) + 1) \times 60$ comparisons need to be done, which is computationally inefficient while leaving the critical problem of rotation invariance intact.

Vranic [17] has extended the silhouettes to the depth-buffer images, which could tackle the problem of inner structures, but they only use 6 views to calculate the shape descriptors. Chaouch et al. [29] presented a set of depth sequence information for a more accurate description of 3D boundaries from 20 depth images rendered of a 3D model. This description method classifies the regions into background regions and projected object regions and generates $2 \times N$ depth lines for a depth image of size $N \times N$. For the object regions, the first derivatives of the sequences are used for description. Similarity is computed via dynamic programming distance, which could lead to an accurate matching of sequences even in the presence of local shifting of the shape.

Axenopoulous and Daras [30] have proposed a Compact Multi-View Descriptor (CMVD) which compactly represents a 3D object as a set of multiple 2D views, both silhouettes and depth images. For each view, a set of 2D rotation-invariant descriptors, Polar-Fourier Transform, Zernike Moments and Krawtchouk Moments are extracted. 18 views from 32-hedron are extracted and the authors stated that 18 views can best compromise representativeness and compactness. The matching scheme effectively calculates the global shape similarity by combining the extracted information from the multi-view representation.

Makadia and Daniilidis [31] defined the similarity measure as the cross-correlation of the rendered silhouette image collections. This technique takes the advantage of that spherical correlation being equal to the multiplication in the spherical Fourier domain. A coarse-to-fine comparison strategy is achieved by using low-degree Fourier coefficients for coarse estimation and high-degree Fourier coefficients for finer estimation. The feature design is rotation invariant and $2L^2$ ($L = 3,5,17$) images are rendered respectively for consecutive fine-tuning. The results show that the matching similarity depends more on low-frequency coefficients.

Stavropoulos et al. [32] considers the query-by-range-image approach from a computer vision perspective. The concept is that there should be a virtual camera with certain intrinsic and extrinsic parameters that can produce an optimal range image from the 3D object to correspond with the query range image. Initially, salient features are extracted

for both query range image and 3D target model, and an objective error function is minimized based on the salient features of the object. A hierarchical search framework is applied to search for the optimal solution in the parameter space. The proposed framework is proved to be efficient and can be easily extended to use other kinds of models.

More recently, Papadakis et al. [33] proposed to use a set of panoramic views of a 3D object which could describe the position information and orientation of the object's surface in 3D space. The panoramic view is particularly descriptive because it can capture a large portion of an object, equivalent to information from several views using orthogonal projections. For each panoramic view, 2D Discrete Fourier Transform and 2D Discrete Wavelet Transform are applied. It is reported by the authors that using the wavelet features can increase the efficiency in terms of storage and computational time. A local relevance feedback scheme is also employed to increase the retrieval performance.

In the engineering domain, Pu et al. [34, 35] proposed to use 2.5D spherical harmonics transformation and 2D shape histogram to retrieve 2D drawings based on their shape similarity. The first approach uses the spherical function to transform the drawing from a 2D space into a 3D space. The second approach is based on statistical distribution between two randomly sampled points. A flexible sampling strategy is applied to allow users interactively emphasize certain local shapes. The results show the proposed methods have good discriminative ability and can be extended to free-hand sketches, vector drawings and scanned drawings.

In conclusion, the 2D visual-based similarity methods in common bear the advantages of being highly discriminative, and if applied appropriately, can work for articulated objects and partial matching. They are also beneficial for multimodal queries of 2D sketches, images, as well as 3D models. The state-of-art performance suggests that this is an appealing candidate for further investigations. The main drawback is that the valuable information, due to self-occlusion, is discarded. A potential research direction may combine shape descriptors both directly from 3D models and their 2D view projections in order to achieve satisfying results.

2.3 3D Model Retrieval using Bag-of-Words Model

Bag-of-words approach has been one of the most popular and effective methods in fields of document retrieval [27, 34, 36, 37] and image categorization [38-40] and content-based image retrieval [41]. In essence, it represents an object as histogram of feature occurrence frequency according to a codebook learned from sets of features extracted from all the models in a dataset. Each feature is encoded as a visual “Word” according to the codebook, and therefore this approach is called “Bag-Of-Words” approach. As both the spatial and geometric information of the features are discarded, and only the orderless histograms of visual “words” are kept as shape descriptors. The bag-of-words approach is not only efficient but also effective for matching of sets of local features.

Ohbuchi et al. [42] was among the earlier works to use bag-of-words model for 3D model retrieval. In their bag-of-SIFT features (BF-SIFT) approach [42], a set of range images, 6-view, 20-view and 42-view, are evenly sampled from vertices of polyhedrons for each model. Then, Scale Invariant Feature Transform (SIFT) [43] features are extracted from the range images and quantized into a visual codebook using unsupervised K-means clustering. The features are coded according to the codebook using direct quantization. Similarity distance is computed using Kullback-Leibler divergence (KLD). The influence of number of views and codebook size for the retrieval performance are tested on Princeton Shape Benchmark (PSB) of the rigid generic 3D models [44] and McGill Shape Benchmark (MSB) [45] of articulated 3D models. The BF-SIFT method shows better retrieval accuracy than both the Light Field Method (LFM) [46] and Spherical Harmonics Descriptor (SHD) [18] on MSB and no worth than peers on PSB. By increasing the vocabulary size from 100 to around 3000, the R-precision increases first, reaches at a peak and then decreases. In addition, it is also found out that with the increasing of number of views, the R-precision tends to increase as well. This is because there are more features extracted for each model with larger number of views, and therefore it is more robust because a local visual feature tends to be described by multiple visual words.

Based on above findings, Furuya et al. [47] proposed to extract a much larger number of local features by over densely sampled spatial grids and scales. To deal with the

thousands of features of high dimensions, there are two possible ways which could alleviate the difficulty of feature quantization and histogram indexing. The first method is to use a fast feature encoding method, e.g., tree-based encoder Extremely Randomized Clustering Trees (ERC-trees) [48] to accelerate the implementation speed. Another method is to reduce the dimensionality of the feature vectors. Ohbuchi et al. [49] proposed to use dimension reduction for the extracted SIFT features. Unsupervised Dimension Reduction (UDR), Supervised Dimension Reduction (SDR), and Semi-Supervised Dimension Reduction (SSDR) are proposed to learn features in a batch and encode the knowledge to a smaller m -dimensional subspace. Although the results suggest that the dimension reduction is able to compress the feature and achieves an improved retrieval performance, there is only empirical quantization levels mentioned.

Ohbuchi et al. [50] further proposed an unsupervised distance metric learning approach with a combination of both the local visual features and global features to improve the bag-of-words method. The motivation is to look for a compromise of shape representation using local features and global features. On one hand, it may happen that the local features are almost identical while the global shape is different, for example the pipes bent in U shape and S shape. On the other hand, shape with articulated parts may appear totally different using global feature description. Experiments using the adaptive distance metric have shown better retrieval accuracy across multiple benchmarks with different characteristics. However, the intention to

add one global descriptor, which is one SIFT feature at the center of each range image, with local feature descriptors does not show difference in the performance of using only local features. Interestingly, the 1SIFT descriptor itself performs well enough, e.g. better than the BF-SIFT approach.

Lian et al. [51-53] proposed a multi-view matching scheme, called Clock Matching Bag-of-Features (CM-BOF), by finding the minimum distance pair between all 24 possible matching pairs due to inexact pose alignment. No explicit description and explanation of advantages using CM-BOF over BF-SIFT are found in these two works, but if the histograms are generated for each view and compiled into a descriptor with certain order, the spatial relations between views are incorporated in this way. The CM-BOF performs slightly better than the BF-SIFT approach.

Except for SIFT features, there are other local feature descriptors that are used in combination with Bag-of-Words approach. Spin images [54] are applied to the 3D model directly to obtain local oriented gradients image as feature descriptor. Unlike SIFT features, spin image is a projection of normals within a certain range to basis points, therefore it can capture the details of concaves and self-hidden area in a mesh.

Li et al. [55] proposed a weak spatial constraint to encode the spatial information within concentric spheres. Instead of using a global dictionary to describe the histogram of words, the model is partitioned into M regions from outer sphere to inner sphere. The final feature descriptor is therefore of length $N*M$, where N is the

codebook size and M is the number of regions. The results in [55] show that spatially enhanced bag-of-words approach slightly outperforms than the bag-of-words approach. However, factors include the partition of number of regions, the support range r of spin image, the number of oriented points for each model are all non-trivial and not discussed in detail in [55].

Bag-of-words approaches which extract local features from 2D images are then extended to extract features from 3D mesh directly.

Fehr et al. [56] proposed to extract spherical patches in the 3D shape centered in respective sampling locations for local feature description. They stated that the selection of interest points in 3D model is far less crucial than the 2D case, because in 2D setting, the objects of interest may suffer from cluttered scenes. This may be true in certain cases; however, the authors only test the proposed approach on the well segmented Princeton Shape Benchmark (PSB) and have not compared the 3D Bag-of-Words method with its 2D equivalent. Tabia et al. [57] also proposed to extract local features, which are patches from the 3D mesh model directly, for non-rigid shape retrieval using bag-of-words approach.

Ohkita et al. [58] employed a shape-based 3D model representation, namely Local Statistical Features (LSF) to integrate with the bag-of-words model. LSF computes statistical values between sampling feature points within local sphere geometry. Thus

it is not only compliant to well-defined closed mesh, but also can be used for other types of shape models, for example polygon soup. From the results tested on MSB and PSB, the BF-LSF has achieved near or no better R-precision than the 2D version proposed in [47]. Kawamura et al. [59] proposed a novel local feature, which combines local geometrical information and spatial context, computed over mesh surface. As bag-of-words approach discards all the spatial information of local features, statistical diffusion distance is added to augment the contextual information. The combination of geometrical and spatial information is demonstrated to outperform either the local geometrical features alone or the spatial information. A single-scale version and a multi-scale version of the local features are both tested using bag-of-words model. The results still show no better than the dense 2D version of BF-SIFT in [47]. Tang et al. [60] conducted an extensive evaluation of different 3D shape descriptors with bag-of-words algorithm for 3D model retrieval using SHREC 2011 Non-rigid Watertight Meshes Dataset [61]. Six local descriptors evaluated using the method by Heider et al. [62], namely Distance to plane (DTP), Normal Distribution (ND), Mean Curvature (Mean), Gaussian curvature (Gauss), Shape Index (SI) and Curvature Index (CI) are extracted either randomly or using salient location detections are implemented within the bag-of-words framework. For random sampling, the best descriptors overall in terms of retrieval accuracy and high statistical values are Mean Curvature (Mean), Shape Index (SI), and Curvature Index (CI). Salient sampling of local shape descriptors needs slightly less number of features than random sampling in order to achieve a similar level of performance, but the advantage is very much limited. The

authors also examined combining descriptors by concatenating feature vectors and by concatenating histograms. The best combination comes from concatenating vectors, and concatenating histograms gives better performance overall. But there are also some combinations perform worse than using single descriptor.

To deal with articulated and partially occluded shape, Toldo et al. [63] proposed a hierarchical 3D object segmentation technique to partition objects into different segments. Sub-parts are then described by local region descriptors, which are properly clustered in order to be both discriminative enough and robust to irrelevant variations. Instead of using a single codebook, this method might need up to 108 different visual codebooks for classification of each particular 3D shape, which are very computationally expensive. The object is represented by a histogram assigning the object sub-parts to visual word, and SVM is used for classification. The part-based representation shows comparable retrieval accuracy with state-of-art approaches on SHREC 2007 Watertight models [64] and Tosca dataset [65].

Lavoue [66] proposed to uniformly sample local patches described on the mesh surface, which are computed by projecting the geometry of neighborhood onto the eigen-vectors of the Laplace-Beltrami operator. These descriptors are not only translation and rotation invariant, but also discriminative enough and robust to noise and connectivity changes. A hybrid representation of original and spatially-sensitive bag-of-features is proposed for final shape representation. Experimented on SHREC

2007 Watertight models, the hybrid bag of 3D features approach achieves almost equivalent accuracy as that of Toldo et al. [63] at a higher recall level, but more stable at a lower recall level. Although this method has achieved satisfying retrieval accuracy in most cases, it cannot find precise matching for corresponding subparts.

2.4 3D Model Categorization

Previous approaches have put very much focus on the retrieval of 3D models. However, the one-to-one comparison of 3D models in the 3D model retrieval algorithms is not scalable for large-scale datasets. Until very recently, there are a small amount of work turns to categorization system for large-scale similarity search of 3D models.

Toldo et al. [67] proposed a 3D model categorization system with part-based bag-of-words representation. The work has mainly put focus on the part-based representation with simple explanations for the categorization scheme with details undisclosed. It also mentions to adopt the histogram intersection kernel in the multi-class SVM and a one-against-all strategy is followed. However, the training process with the nonlinear kernel takes longer time than the proposed methods in this thesis.

Li et al. [68] proposed a non-parametric kernel discriminant analysis approach for 3D model classification. Invariable features are extracted by geometry projection-based

histogram model to represent the 3D models. The kernel discriminant analysis is based on a conceptual transformation of the features from the input space into the kernel space. The authors reported a high classification rate is on the Princeton shape benchmark.

Tabia et al. [69] proposed a belief function based approach for the categorization of 3D models. The training stage is processed on a set of representative parts for 3D models within the same category. Specifically, the labeled part is of evidence supporting the prediction of the category of the whole object. And it is especially able to handle objects which are “unclassifiable” by being able to reject it. However, the partitioning procedure is biased, as stated by the authors, in the categorization procedure. And the spatial relations between parts are not integrated in the matching process.

2.5 Summary

This chapter has surveyed existing methods for 3D model retrieval and few works for 3D model categorization. Among all the approaches, bag-of-words representation of 3D models based on the 2D visual similarity information proves to be the most promising approach for its superior performance and compactness in representation. However, there are still several limitations which hinder the bag-of-words representation for the further improvement of retrieval efficiency and scalability into large-scale retrieval problems. First, although salient feature detection methods might

be more suitable for object recognition, they are not efficient and representative enough for the 3D model retrieval tasks. Second, current 3D model retrieval systems can only handle several hundred of models for similarity and comparison and not scalable to deal with the huge amount of models. Therefore there is a gap between current single model comparison and generic model comparison. Therefore, the work in this thesis is proposed to address the two research gaps mentioned above.

Chapter 3 FRAMEWORK FOR RETRIEVAL AND CATEGORIZATION OF 3D MODELS USING BAG-OF-WORDS MODEL REPRESENTATION

This chapter gives an overview of this research. The framework of bag-of-words approach is outlined first. The links between this chapter and the following Chapter 4, Chapter 5 and Chapter 6 are addressed. The procedures of using bag-of-words approach for 3D model representation are introduced in more details. Similarity distance computation and evaluation measures for 3D model retrieval are also given in this chapter. Lastly, four public 3D model benchmarks that will be used in the following chapters are briefly introduced in this chapter.

3.1 Overview of this Research

This thesis aims at develop efficient retrieval and categorization algorithms of 3D models using bag-of-words model for 3D model representation. The concept of 3D model retrieval is to compare query model with each target model by calculating the similarity distance between them. When the stored number of existing models grows large, it becomes unaffordable for one-to-one comparison of query model with all the available target models. Therefore, there is a need to develop a system to reduce the number of comparisons. The categorization of 3D models is to compare the query

model only with a limited number of category classifiers and assign it to a category of similar models. Figure 3.1 depicts the structure of this thesis. Both of the proposed retrieval and categorization tasks are based on the bag-of-words approach for the 3D model representation. Chapter 4 and Chapter 5 of this thesis focus the case studies more on the retrieval tasks and Chapter 6 put the emphasis on the categorization system.

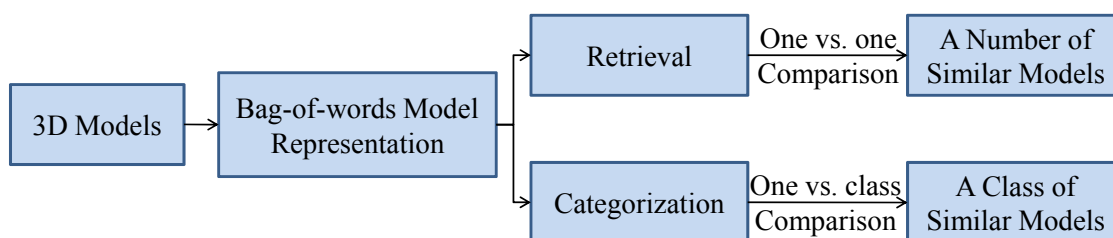


Figure 3.1 Overview of Retrieval and Categorization of 3D Models based on Bag-of-words Representation.

As bag-of-words model is used for the 3D model representation method for both retrieval and categorization tasks throughout this thesis, the procedures to compute bag-of-words representation of 3D models are introduced briefly in this section. The procedures are illustrated in Figure 3.2. Pose alignment is performed for each model to achieve position, rotation and scale invariance. This is followed by multi-view depth-buffer images extraction at specific viewing directions. Local visual features are extracted from the 2D depth images. A codebook is learned from the sets of local features extracted and each feature can be encoded as a visual word according to the codebook. Finally, each model can be represented as histogram of visual words

according to the occurrence frequency of features in the codebook, and the histogram is the final 3D representation.

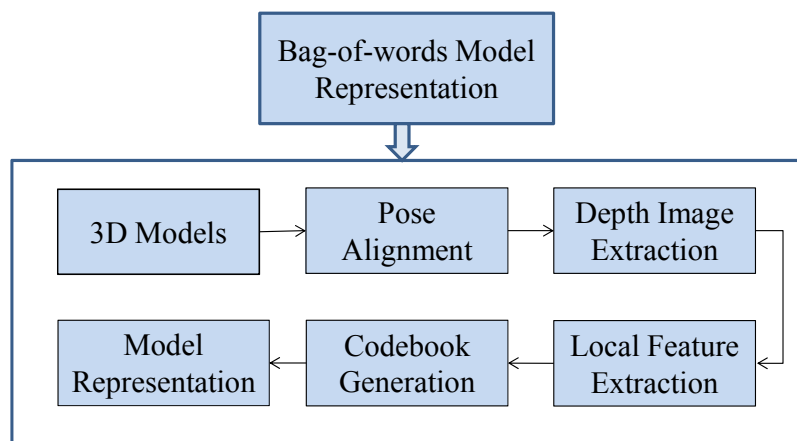


Figure 3.2 Procedures to compute bag-of-words representation for 3D models.

In this thesis, Chapter 4 and Chapter 5 are dedicated to improve the local feature extraction and description methods for 3D model retrieval using bag-of-words model representation. Specifically, Chapter 4 proposes modified dense sampling and multi-scale sampling strategies of local features using SIFT description for fast and more accurate 3D model representation. Chapter 5 proposes two region-based feature detection and description methods, which are both of lower dimension, efficient to compute and discriminative for 3D model retrieval. Chapter 6 develops a 3D model categorization system using multi-class kernelized SVM for classification. Linearly approximated histogram intersection kernel and chi-square kernel are incorporated in the SVM. These two kernel mappings are effective for histogram-based representation, and hence achieve better performance.

3.2 Pose Alignment and Depth Image Extraction

3.2.1 Pose Alignment

Objects represented as polygonal meshes are often given in arbitrary position, orientation, reflection and scale in the three dimensional space. Some feature descriptors, e.g., shape distribution, are invariant to the pose changes due to the design of feature representation methods. However, pose alignment is not a trivial problem for most of feature-based model representation methods because they are extracted with respect to the absolute position of the object. Therefore, in order to extract stable features each time, the 3D models must be transformed into a canonical position, and the process is called pose alignment.

In this thesis, pose alignment is applied to achieve translation, rotation, scale and reflection invariance. The translation, scale and reflection invariance are implemented use the methods proposed in [17]. For the rotation invariance transformation, we proposed to choose the best rotation among multiple rotation methods in this thesis. The details are followed.

Given the mesh model as a collection of triangles $M = \bigcup_{i=1}^m T_i$, where $T = \{T_1, T_2, \dots, T_m\}$, $T_i \in \mathbb{R}^3$ contains the connectivity of vertices for each face, and $P = \{v_1, v_2, \dots, v_m\}$, $v_i = (x_i, y_i, z_i) \in \mathbb{R}^3$ are the positions of vertices in the coordinate system.

First, the model is translated to the center of this mass to the origin of the coordinate system to achieve translation invariance. The following formula is applied:

$$M_1 := M - c = \{u \mid v - c, v \in M\} \quad (3.1)$$

Where $c = \frac{1}{S} \sum_{i=1}^m s_i c_i$ is the center of mass, s_i, c_i are the area and centroid of each triangle and S is the total surface area of the mesh model. The center of mass is calculated by taking into account the tessellations of the mesh model, and is therefore more robust to model degenerations.

Principal Component Analysis (PCA) [70, 71] is the most widely used approach for rotation estimation. It seeks a projection onto a lower dimensional space which can best represent the data. The main shortcoming for rotation alignment using PCA is that it can often produce poor alignments due to lack of consideration of object local structures and cannot produce pair-wise alignment of models within the same class.

In this thesis, Continuous PCA (CPCA) [17], Normal PCA (NPCA) [72] and Maximum Normal Distribution (MND) [35] are used to obtain the rotation matrix R and the method resulted in a minimum Axis-Aligned Bounding Volume (AABV) is finally chosen.

The rotation alignment is given as:

$$M_2 := R \cdot M_1 = \{u \mid u = R \cdot v, v \in M_1\} \quad (3.2)$$

where R is the rotation matrix and v is the vertices positions after translation alignment.

In standard PCA, R is obtained by performing Singular Value Decomposition (SVD) on the covariance matrix constituted by the vertices' directions directly. The first, second and third largest variance is the first, second and third principal directions of the rotation matrix.

In CPCA, the covariance matrix is computed by integral of a function over the model's surface, which demonstrates more robust rotation estimation than standard PCA applied on vertices position after this mapping. Because the CPCA has taken into account all points of the model with equal weight and is stable regardless of the degenerations of the triangulated meshes. CPCA is the most widely adopted orientation alignment step although it would still often result in 24 ambiguities of the placement of the orientations.

The covariance matrix is therefore given as:

$$\begin{aligned} C &= \frac{1}{S} \iint_{v \in M} (v - c) \cdot (v - c)^T ds \\ &= \frac{1}{12S} \sum_{i=1}^m (f(p_i(1)) + f(p_i(2)) + f(p_i(3)) + 9f(c_i)) S_i \end{aligned} \quad (3.3)$$

NPCA is calculated by performing PCA on the normal distributions of the mesh model.

The area of each triangle is taken as the weight factor for each facet normal. The covariance matrix is given as:

$$C = \frac{1}{S} \sum_i^N s_i \cdot n_i \cdot n_i \quad (3.4)$$

where n_i is the surface normal for each triangle.

Comparing to NPCA, the Maximum Normal Distribution (MND) method is to exhaustedly search for the maximum direction the normals of triangles are projected on. The intuition behind is larger triangles contributes more to the overall distribution of surface normals. It first calculates the normal of each triangle and normalizes it to the unit length, and followed by summing up the areas of all triangles with the same and opposite directions. The first principal axis is chosen by the direction of normal with the maximum areas, the second principal axis is then determined by searching the remaining distribution and orthogonal to the first principal axis, and the third axis is determined when the first two principal axes are fixed. Both NPCA and MND are more suitable to tackle the problem of objects with large flatten areas or sparse structures.

Although the PCA-based pose alignment methods and MND has the shortcoming of inaccuracies for pair-wise alignment of 3D models, studies have shown that descriptors designed with explicitly alignment are generally more accurate than encoding rotation invariance directly into descriptors [73]. Therefore, explicit pose alignment remains valuable and is needed for further investigation.

3.2.2 Depth Image Extraction

After pose alignment, a set of multi-view depth images need be rendered from the 3D

model. Vranic [17] proposed to use only 6-view images extracted from an 8-hedron and Ohbuchi et al. [42] suggested that more features generated from more views will achieve better retrieval accuracy and they therefore used 42 views of depth images extracted from an 80-hedron. As 42-view might be too time-consuming and redundant in representation, 18 views extracted from a 32-hedron are proposed by Daras et al. [73], which are expected to be symmetric with respect to 90 degrees rotations for the three orthogonal axes.

In this thesis, 6-view depth images are employed throughout the experiments as we only want to demonstrate the effectiveness of our proposed methods. More views will undoubtedly result in better retrieval accuracy, but also more computational cost. The 6-view depth images are generated from 6 vertices of an octahedron enclosing the model scaled to unit. The camera and object positions are illustrated as an example, as shown in Figure 3.3.

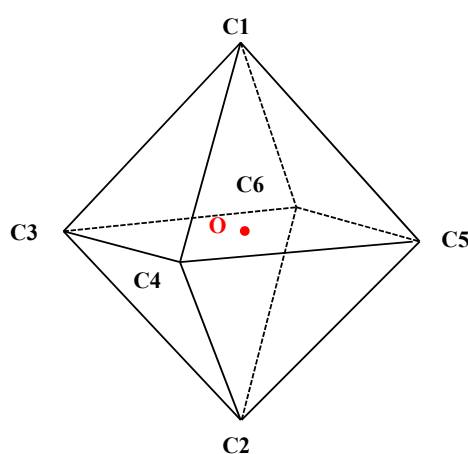


Figure 3.3 6-view camera positions with respect to the object.

To get the depth images, we first do mesh voxelization [74] for each model into grid of $256 \times 256 \times 256$. The mesh voxelisation implementation toolbox [75] is used in this work. Then the depth value of each voxel is projected onto the viewing plane orthogonally to generate depth images of resolution 256×256 .

3.3 Bag-of-Words Model Representation

To represent a 3D model using Bag-of-Words model, the collection of projected depth images of the 3D model can be treated as a document. The "words" of the depth images can be defined by generating a universal codebook from sets of local image features detected. The 3D model can then be represented as the occurrence of each "word" according to the codebook, where the ordering and spatial relations of the words are irrelevant.

3.3.1 Codebook Generation and Model Representation

The codebook can be generated via unsupervised K-means clustering by learning from thousands of features extracted. As we will detail the feature extraction stages in chapter 4 and chapter 5. In this section, we only put focus on the codebook generation and representation of 3D models according to the codebook.

Given the set of local features (f_1, f_2, \dots, f_n) , the k-means clustering is to find the k mean vectors $(\mu_1, \mu_2, \dots, \mu_k)$, such that the squared Euclidean distance for a feature to the nearest center is minimized

$$\arg \min \sum_{i=1}^k \sum_{j=1}^n \|f_j - \mu_i\|^2 \quad (3.5)$$

The k cluster means therefore constitute a codebook of k number of words. Then for any given feature, it can be encoded by the nearest center mean cluster obtained in equation 3.5. Thus, a 3D model of m number of local features can be described as a histogram of k bins where the binning elements are summed into m . The histogram H is then the shape descriptor of this 3D model.

3.3.2 Similarity Distance Comparison

In this section, two similarity distances for computation of model similarity distance are introduced. Given two shape histograms H_1 and H_2 , and a codebook of size K , the similarity distance can be computed using the following distance measures. The distance metrics are all normalized into the range of 0 to 1, when the distance is small it means the two models are more similar and vice versa.

Normalized L1 distance is a standard measure for the comparison of two feature vectors. In this work, it is calculated as the sum of absolute difference for all histogram bins.

$$D_{L1}(H_1, H_2) = \frac{\sum_{i=1}^K |H_2(i) - H_1(i)|}{\max(\sum_{i=1}^K H_1(i), \sum_{i=1}^K H_2(i))} \quad (3.6)$$

Maximum Histogram Intersection Distance (MHID) is developed by Swain et al. [76] to recognize image object to a large database of models. It is robust to image noise and occlusion, and therefore it is stable if the histogram representation has irrelevant variations. Lian et al. [51] first adopted this measure for bag-of-features 3D model

retrieval. It is given by

$$D_{MHI}(H_1, H_2) = \frac{\sum_{i=1}^K \min(H_1(i), H_2(i))}{\max(\sum_{i=1}^K H_1(i), \sum_{i=1}^K H_2(i))} \quad (3.7)$$

3.4 Evaluation Measures for 3D Model Retrieval

In order to make a confident evaluation of the retrieval performance of proposed algorithms, the following measures, namely Precision-Recall curve, Nearest Measure, First-Tier, Second-Tier, E-measure, and Discounted Cumulative Gain (DCG), are employed in this thesis. The measures are calculated based on a query and a collection of objects with known ground truth of relevancy. In response to a given set of queries, a retrieval algorithm searches the benchmark database and returns an ordered list of responses according to the similarity distance between the target objects and the query. Ideally, the 3D model retrieval system is expected to retrieve all relevant objects to the query objects in the ranked list. In practice, the following measures can be used to evaluate the efficiency of the retrieval algorithms.

- Precision-Recall curve

Precision-recall curve describes the relationship between precision and recall for a ranked list of matches. Precision is the ratio of relevant objects retrieved with the amount of all retrieved objects. Recall is the ratio of number relevant objects retrieved with respect to the amount of all relevant objects. Precision-recall curve is usually plotted along the vertical axis against recall along the horizontal axis. A perfect retrieval result produces a horizontal line at the top of the plot, indicating that all the

models within the query object's class are retrieved as the top ranked matches.

- Nearest Neighbor, First-Tier and Second-Tier

These evaluations measure the ratio of relevant models within the top M matches. For a class with C objects, when $M = 1$, it is Nearest Neighbor precision; when $M = |C| - 1$, it is First-Tier precision; and when $M = 2 * (|C| - 1)$, it is Second-Tier precision. Nearest Neighbor measure indicates the percentage of the closest matches that belong to the same class as the query. The First-Tier indicates the recall for the smallest number of M models that could possibly include 100% of the models and the Second-Tier is less stringent. Higher the values of these measures indicate better the retrieval accuracy.

- E-Measure

E-measure is to combine precision and recall into a single value for a fixed number of retrieved results to evaluate how well a retrieval system performs. The intuition is that a user of a search engine would be more interested in the first page of retrieval results than in later pages. It is given as $E = 1 - \frac{2}{\frac{1}{P} + \frac{1}{R}}$. The E-measure value for a perfect match is 1 and the higher values indicate better retrieval results.

- Discounted Cumulative Gain (DCG)

Discounted cumulative gain (DCG) measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the

result list to the bottom with the gain of each result discounted at lower ranks. This evaluation weights relevant objects retrieved in the front of list more than the relevant objects retrieved in the later of the list, assuming the user is less likely to examine the object in the end of ranked list. Specifically, the ranked list is first converted to a list G , where $G_i = 1$ denotes the object is relevant and $G_i = 0$ denotes the object is irrelevant. Thus, the cumulative gain vector CG is defined recursively as follows:

$$G_i = \begin{cases} G_1, & i = 1 \\ CG_i = CG_{i-1} + G_i, & \text{Otherwise} \end{cases} \quad (3.8)$$

A discounted factor is applied to progressively reduce the weight for object that ranks in the later of the list:

$$DCG_i = \begin{cases} G_1, & i = 1 \\ DCG_{i-1} + G_i/\log_2 i, & \text{Otherwise} \end{cases} \quad (3.9)$$

3.5 Experimental Datasets

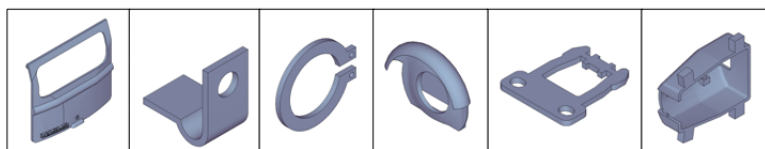
3.5.1 Purdue Engineering Shape Benchmark

The mechanical models are well-known for being characterized by presence of design features such as holes, cavities and helixes or they may often be resemblance of two or more parts. Different from the multimedia models, the Purdue Engineering Shape Benchmark (ESB) [2] is designed to cluster parts according to the engineering context. It consists of a primarily shape-based classification of models. There are a total of 801 models classified into 3 super classes, which are further divided into 42 finer categories. The three super classes are: solids of revolution, rectangular-cubic prism

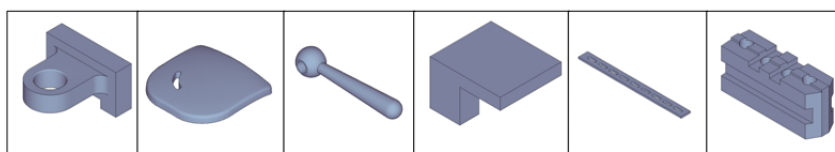
and thin-walled components. In this study, we exclude 66 models classified into "Miscellaneous" classes for experiments because they do not share similar shapes.

Figure 3.4 shows several examples of models for each super class.

Flat Thin-Wall Component



Rectangular-Cubic Prism



Solids of revolution

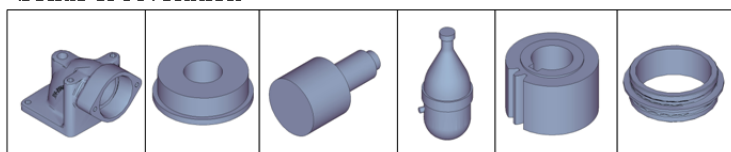


Figure 3.4 Examples of CAD models from ESB dataset [2].

The ESB dataset may have the following limitations. First, the number of models for each class varies from several to tens of them, which make it biased to evaluate when using certain retrieval algorithms. Second, for some categories, the models seem not be similar in shape, which make the ground truth not reliable. Third, some classes, for example thick plates, machined plates and machined blocks, are difficult to differentiate since there might be only particular fine details changed for them. Therefore, in this thesis, we have modified the ESB dataset to overcome the above mentioned limitations.

3.5.2 Modified CAD dataset

The modified CAD dataset has 424 models categorized into 46 classes. It was obtained by arranging and sorting up the ESB dataset. The number of models for each class now varies in a moderate range, from minimum 4 to maximum 10 for each class, with most of classes containing 10 models. Some classes contain a large of number of models in the ESB dataset have been portioned into smaller groups based on detailed consistent classification rules. The lists of models for the modified CAD dataset are given in appendix A.

3.5.3 NIST Generic Shape Benchmark

NIST Generic Shape Benchmark [77] is a public shape benchmark which has been used for 3D shape retrieval contest organized by AIM@SHAPE project [38]. There are equal number of models for each category to minimize the bias for evaluation. Each model in the dataset is triangulated, scaled to the same size, pose normalized and partially mesh errors corrected. The benchmark consists of 80 query models with two for each of the 40 classes and 720 complete target models, 18 for each of the 40 classes. The list of models is given in Table 3.1.

Bird	Fish	Nonflying Insects	Flying Insects	Biped
Quadruped	Apartment House	Skyscraper	Single House	Bottle
Cup	Glasses	Hand Gun	Submachine Gun	Musical Instrument
Mug	Floor Lamp	Desk Lamp	Sword	Cellphone
Desk Phone	Monitor	Bed	Non-Wheel Chair	Wheel Chair
Sofa	Rectangle Table	Round Table	Bookshelf	Home Plant
Tree	Biplane	Helicopter	Monoplane	Rocket
Ship	Motorcycle	Car	Military Vehicle	Bicycle

Table 3.1 List of 40 types of models for SHREC generic shape benchmark

3.5.4 SHREC 2009 Partial Dataset

Although many efforts have been devoted to complete 3D models matching, in practice, it is more often when sometimes complete model acquisition is not easily accessible or two models are only similar in partial. SHREC 2009 partial dataset [78] are obtained by modifying the query dataset from the NIST generic shape benchmark. It consists of two query sets. The first query set consists of 20 3D partial models which are obtained by cutting parts from complete models. This second query set contains 20 range images acquired by capturing range data of 20 models from arbitrary view directions. The partial models and range scan queries are shown in Figure 3.5.

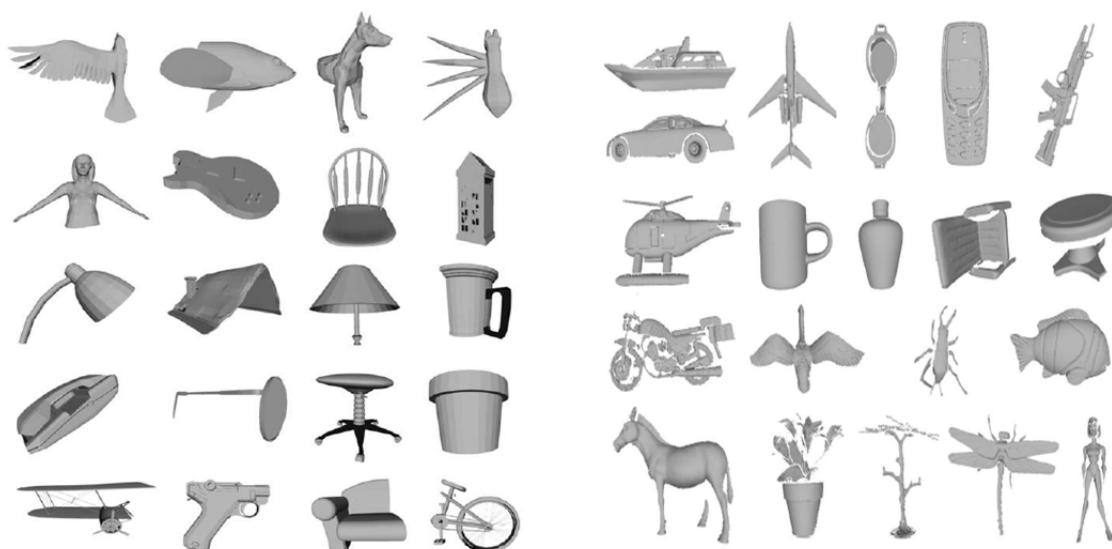


Figure 3.5 Partial and Range query models for SHREC 09 Partial Dataset [78].

The range images are captured using a desktop 3D scanner. The target database is the same as NIST generic shape benchmark, which contains 720 complete 3D models categorized into 40 classes.

3.6 3D Model Retrieval Case Study

In this section, a case study of the 3D CAD model retrieval based on bag-of-words model is illustrated, as shown in Figure 3.6. For a new query model, it is first pose normalized to achieve position, rotation and scale invariance as shown in step 1. This is followed by multi-view rendering (step 2) to extract depth-buffer images of the 3D model. The proposed sampling methods are then applied to extract all the features (step 3), which all lie in a high-dimensional feature space. A codebook is constructed via unsupervised learning of the high-dimensional feature space, as shown in step 4. Then each model can be represented as a histogram (step 5) and distance between

models are computed to retrieve the most similar models.

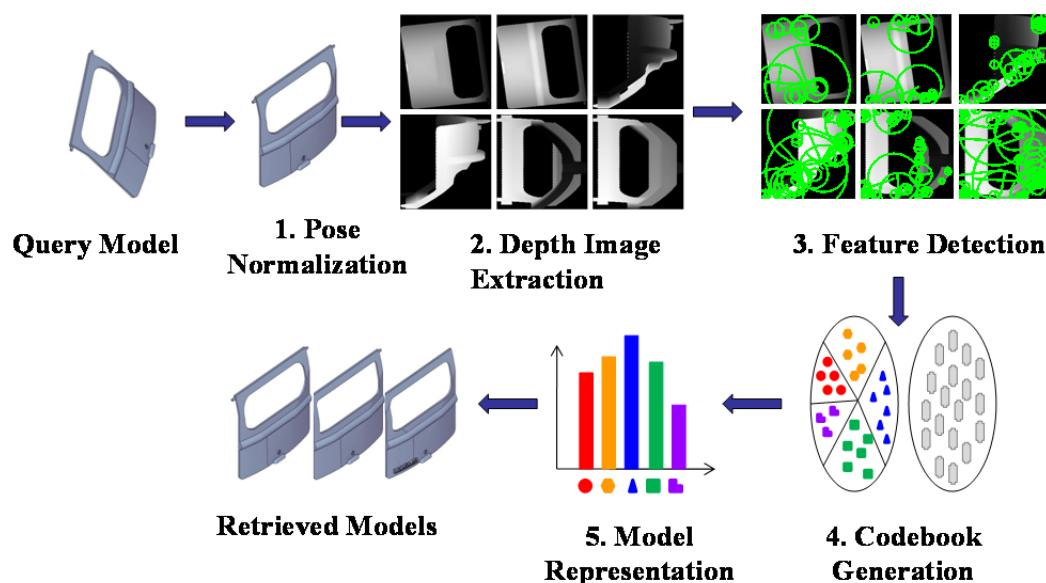


Figure 3.6 A case study of 3D model retrieval procedures.

For a specific query model, there is a precision value measured at each recall level. To evaluate the proposed algorithms on any given dataset, all the query models need to be fed into the retrieval system and compare against all target models in a dataset. The final algorithm performance is decided by the average precision values at each recall level for all query model are obtained.

3.7 Summary

In this chapter, the outline of this thesis is firstly given. Then the pre-processing of 3D models and standard procedures for bag-of-words model representation are introduced in details. The similarity distance computation and evaluation measures for 3D model retrieval are also described in this chapter while the 3D model categorization

evaluation will be detailed in chapter 6. Then, four public available datasets which will be used for tests are also briefly introduced in this chapter. Finally, a case study of 3D model retrieval procedures based on bag-of-words model representation is illustrated.

Chapter 4 MODIFIED DENSE SAMPLING AND MULTI-SCALE DENSE SAMPLING OF LOCAL FEATURES USING SIFT DESCRIPTION FOR 3D MODEL RETRIEVAL

4.1 Introduction

This chapter investigates the sampling strategies of local visual feature extraction in combination with bag-of-words model to improve the 3D model retrieval accuracy.

Scale Invariant Feature Transform (SIFT) [43] algorithm is popular salient local feature detection method in computer vision, with wide applications in object recognition, robotic vision, and more recently found in use for 3D model retrieval tasks. The SIFT algorithm searches for the most stable features across the image scale-space. The features detected are local, typically along edges and corners with sharp changes, and robust to scale and rotation variations. Although the SIFT might be good enough for tasks like object recognition, for which notable features need to be found to build correspondence between the image content and the object model, it is not sufficient to represent the 3D model for the purpose of retrieval tasks. In the 3D model retrieval scenario, a shape descriptor is required not only discriminative enough but also descriptive to faithfully represent a 3D model. As the SIFT algorithm only extracts features along with sharp changes and often ignores the smooth part and overall geometry of the shape, therefore dense sampling and multi-scale dense

sampling techniques are proposed in this chapter to address such problem. The proposed sampling techniques extract local features over the full range of the depth images rendered from the 3D model with different scale and sampling step. Experiments using the proposed feature sampling methods prove to be more suitable for the 3D model representation.



Figure 4.1 Flow chart of sampling strategies of local features for bag-of-words model representation.

The proposed sampling strategies are performed after the depth image extraction as shown in the Figure 4.1 flowchart. After the features are sampled and extracted, codebook can be generated using K-means clustering. Shape descriptors are therefore encoded as occurrence of visual words according to the dictionary. In the following sections, SIFT algorithm for local feature detection and description are firstly introduced. Then, the proposed modified dense sampling and multi-scale dense sampling of 2D local features using SIFT description are described in detail. The optimal parameters of modified dense and MSD sampling parameters and their influence for retrieval accuracy will be studied. Lastly, to evaluate the effectiveness of proposed methods, experiments have been conducted on 3D CAD models and 3D multimedia models respectively.

4.2 Scale Invariant Feature Transform (SIFT) Algorithm for Feature Detection and Description

The idea of Scale Invariant Feature Transform method is to describe local image data as histograms of orientation gradients according to scale and orientation invariant local coordinates at key locations.

First, it detects locations and scales of local extrema (maxima and minima) by searching for stable features across the neighboring scales in the scale-space. The image scale-space is generated by convoluting the image with the difference-of-Gaussian function, which is given as:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (4.1)$$

Where $I(x, y)$ is the image, $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ is the variable-scale Gaussian, $*$ is the convolution operator, and k is a constant multiplicative factor. A local extrema is found by comparing its 26 neighbors.

Then, the locations, scales, and ratio of principal curvatures of the keypoints are fit by the nearby data with Hessian and derivative of a 3D quadratic function. Candidate points with unstable extrema or poorly localized along an edge are rejected.

Finally, the keypoint is described by 4×4 sub-regions, with 8 orientation bins in each region, resulting in a histogram of orientations of dimension 128. Each sample is binned into the histogram and weighted by its gradient magnitude within a

Gaussian-weighted circular window relative to its scale. The geometry of the SIFT descriptor is illustrated in Figure 4.2.

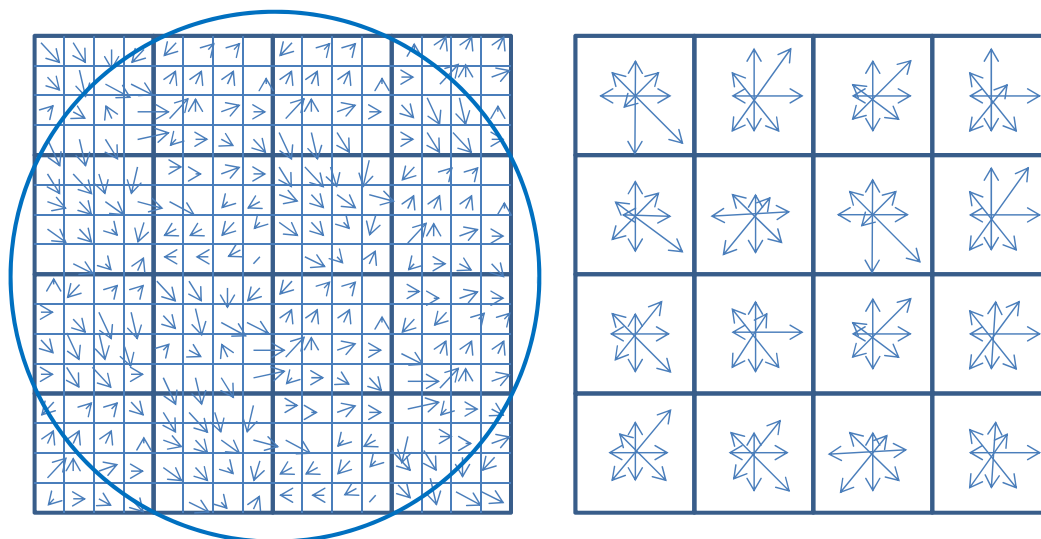


Figure 4.2 SIFT descriptor of 4×4 regions and 8 orientations in each region [43].

Fig 4.3 shows examples of SIFT features detected on the depth images of a door model from 3D CAD database Fig 4.3(a) and a 3D flying bird model from 3D generic dataset Fig 4.3(c). The detected SIFT features are outlined by frames with different orientations and scales. There are only 34 and 32 SIFT features detected from the door model, compared to 73 and 124 features extracted from the flying bird model. To find corresponding SIFT features, Lowe's nearest neighbor matching [43] is used to find the minimum distance between two features. The corresponding matches are shown in Fig 4.3(b) and Fig 4.3(d), where only 3 matches are found for the door model with two are false positives, and 17 matches are found for the flying bird model. The above findings suggest that SIFT algorithm detects less features on piece-wise smooth surfaces than shapes with smooth changes. Therefore, dense sampling and multi-scale

dense sampling of local features are proposed.

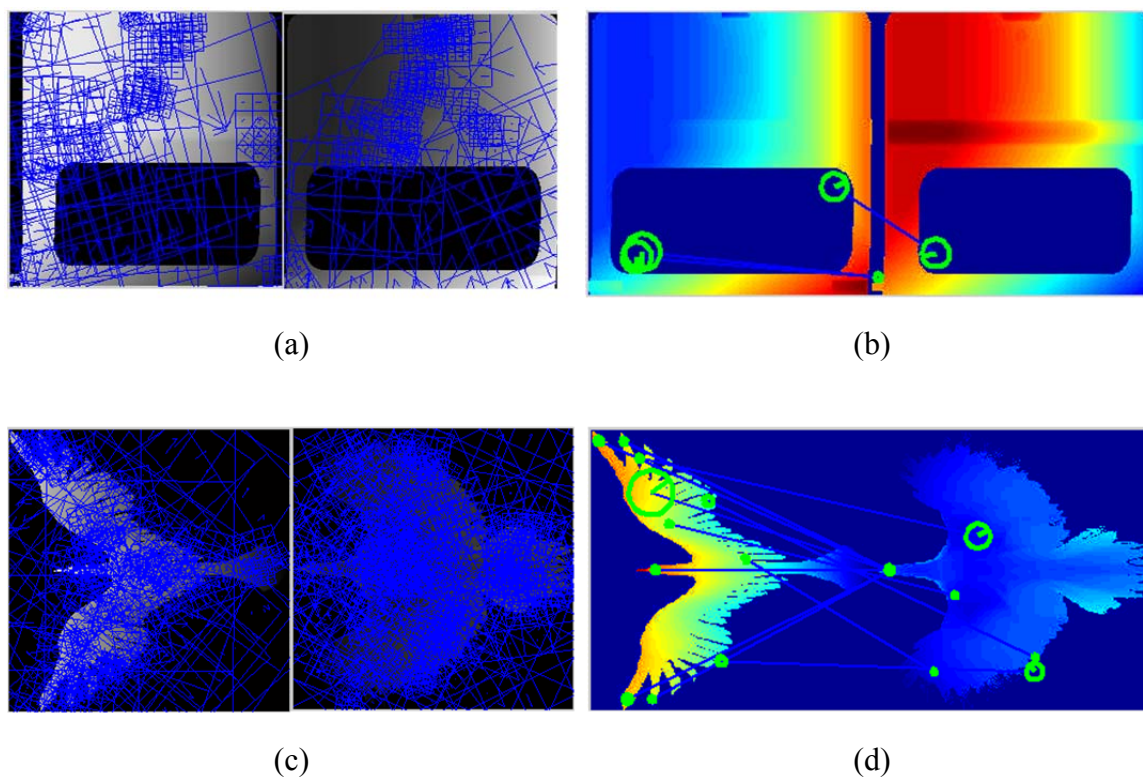


Figure 4.3 (a) SIFT features extracted from depth image of CAD part model, (b) Corresponding features, (c) SIFT features extracted from range image of 3D flying bird model, (d) Corresponding features.

4.3 Modified Dense Sampling and PHOW Sampling for Feature Extraction

It has been shown that features extracted on evenly sampled grid have shown superior performance than features extracted located at keypoints for natural scene categorization [40]. This suggests that the uniformly distributed local shape descriptors may produce shape representation for the purpose of object recognition and retrieval. Therefore, this thesis proposes modified dense sampling and multi-scale dense sampling of local features using SIFT description.

The idea of dense sampling is to extract local features on uniformly distributed grids with a moving window sliding over the image. Given a depth image $I(x, y)$, dense sampling extracts local features at location (x, y) and scale σ with uniform step s .

Two parameters are important for effective representativeness of object using the modified dense sampling. The scale σ determines the window size, which is the spatial range of window. Another is the spacing s between two adjacent windows, which determines the density of the sampled features. Larger window increases the richness of descriptor while reducing the discriminating power, and vice versa.

The original SIFT detection accumulates image gradients which are weighted inversely proportional to the distance of the gradients from descriptor centers within a Gaussian circular window, as shown in the left side of Figure 4.2. In this work, we propose to use a flat rectangular window to substitute the circular window in Figure 4.2, which is given as:

$$G(z) = \frac{1}{\sigma_{win}} \omega\left(\frac{z}{m\sigma}\right) \quad (4.2)$$

Where σ_{win} is the flat window size, m is the magnification factor, which determines the ratio to the relative keypoint scale σ , and $\omega(z) = \max(0, 1 - |z|)$ is the binning function for the histogram accumulation. The gradients within the sliding window are firstly weighted equally and accumulated into the 4×4 spatial bins. After the accumulation, the whole bin is weighted the second time using the average of the Gaussian circular window. The usage of above mentioned two steps to substitute the original Gaussian inversely proportionally weighting makes the feature extraction

speed much faster. The above mentioned procedures of dense sampling of local features using SIFT description can be summarized in Algorithm 1.

Algorithm 1 Modified Dense Sampling using SIFT description

Given an image space $I(x,y)$ of the spatial range of $256*256$

Step 1 Determine the window size σ and sampling step s

While the sliding window (from top left to bottom right)

Step 2 Compute the image orientation gradients within the window

Step 3 Weight the image gradients equally within the rectangular window region which is the same as sliding window

Step 4 Binning the gradients into histogram representation

Step 5 Re-weight the histogram using the average of Gaussian window

end

Algorithm 1 Modified Dense sampling using SIFT description

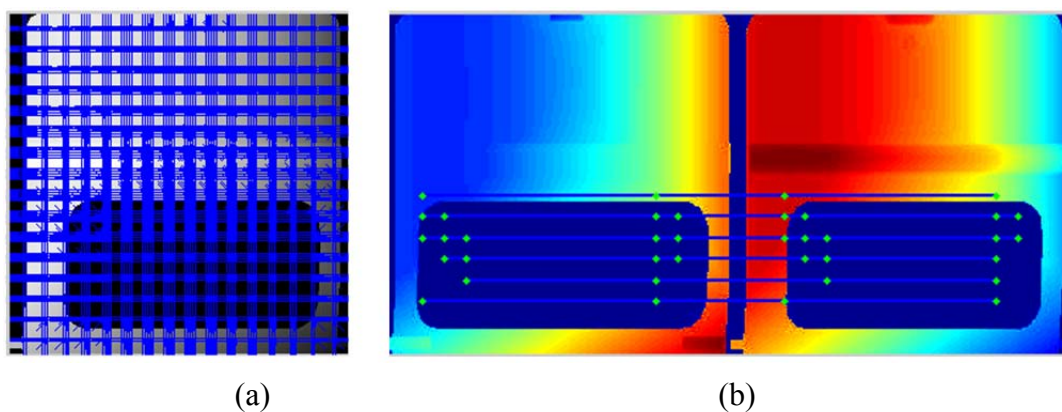


Figure 4.4 (a) Dense sampling of SIFT features of the door model, (b) Corresponding features.

Figure 4.4 shows SIFT features extracted using dense sampling, where the scale and spacing are both 16, in terms of pixel values. For a depth image of resolution 256×256 ,

there are 256 stable features extracted in this case compared to 19 matches found for the two door models in Figure 4.3 (a) and (b) with only one correct corresponding feature found using SIFT feature detection. As the amount of features is very important for the generation of codebook using bag-of-words model, therefore the modified dense sampling can extract more features than the original SIFT detection, especially for 3D CAD models with piece-wise smooth surfaces.

While the modified dense sampling extracts features at fixed scale σ on the spatial grids, the Multiple-Scale Dense (MSD) sampling extracts features at multiple scales $[\sigma_1, \sigma_2, \dots, \sigma_n]$ on the same evenly distributed spatial grids as dense sampling. The MSD descriptors are obtained by extracting densely sampled SIFT features on Gaussian smoothed image of different scale σ . Note, the difference between the MSD sampling method with the PHOW descriptor proposed by Bosch et al. [79] is that the proposed method extracts features on multiple scales, but do not construct the features in a pyramid structure.

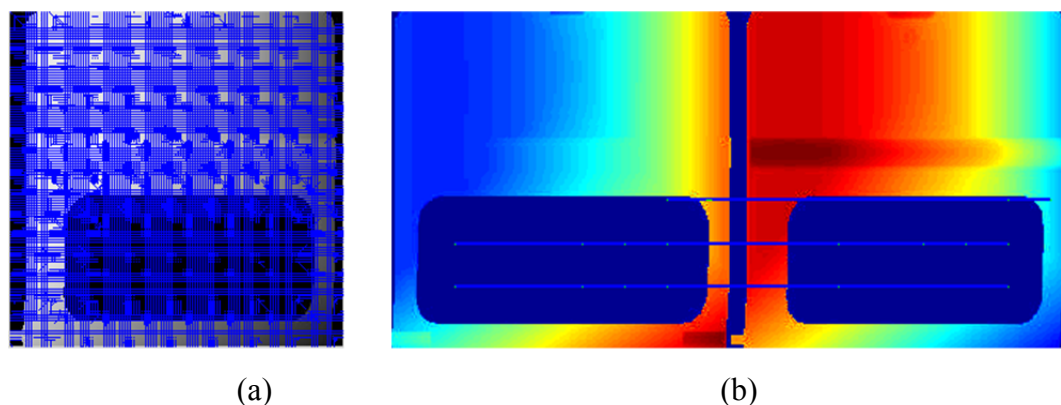


Figure 4.5 (a) MSD sampling of SIFT features of the door model, (b) Corresponding features.

Figure 4.5 (a) shows 110 features extracted using MSD sampling with sizes of 8, 16 and 32 and spacing 32. There are 10 correct matches found using nearest neighbor searching, as shown in Figure 4.4(b). If the spacing is changed to 16, there will be 413 features extracted for each image and 24 correct matches found.

4.5 Results and Discussions

In this section, we have tested the proposed sampling methods with bag-of-words model on the Purdue Engineering Shape Benchmark [6], NIST generic shape benchmark [77], and the SHREC 2009 Partial Dataset.

On all the three datasets, the influence of codebook size and sampling parameters for the modified dense sampling and MSD sampling are compared to the original BF-SIFT method. The experiments are run with Matlab R2010b on an Intel E8400 3.00 GHz CPU. VLFleat toolbox [80] is used for feature extraction and codebook generation.

The influence of sampling density for the retrieval accuracy is studied to achieve a good trade-off between computational and storage efficiency with the retrieval accuracy. The more densely the features are extracted, the bigger computation power and storage are required. As the sampling density can be determined by the scale and spacing, these two parameters are varied from 8 to 56 in terms of pixel values in the depth images to obtain features of varying sizes and density. It can be shown that the NN, DCG and MAP values increase first and then decrease dramatically when the

scale and spacing are 56. The optimal NN value is obtained at 32, while DCG and MAP are obtained at 24.

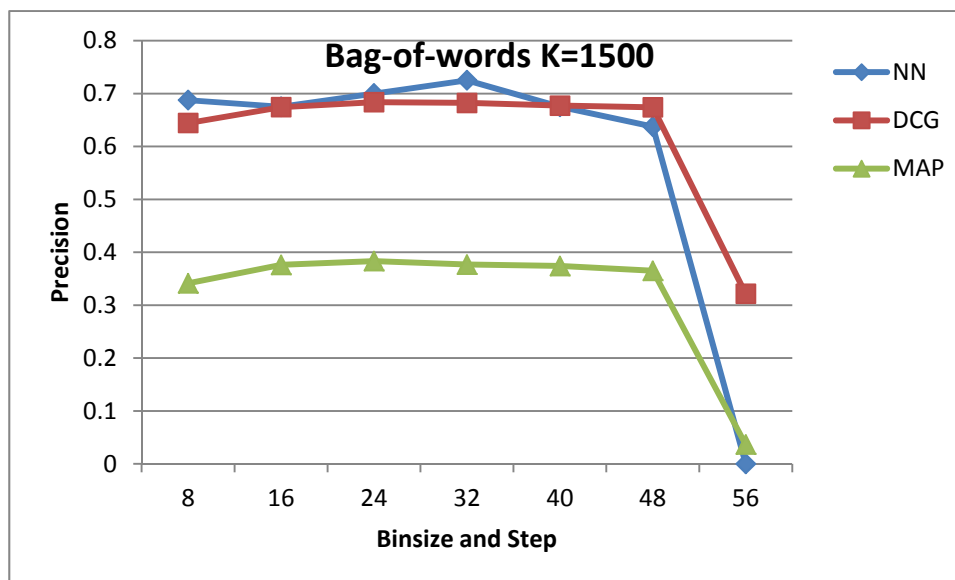


Figure 4.6 Influence of sampling density for the retrieval accuracy.

4.4.1 Retrieval Results on ESB

The feature extraction is performed on the 6-view depth images obtained as introduced in Section 3.4. The original SIFT algorithm detects an average of 164 features per model. For dense sampling, we choose scale σ and spacing s both to be 32, which gives rise to 150 features per model. For MSD sampling, we choose the scale to be 8, 16 and 32 and the step to be 32, which will gives 660 features per model. The feature extraction time is shown in Table 4.1. It can be observed that the modified dense sampling and MSD sampling with flat windowing are almost three times faster than original SIFT detection in terms of per feature extraction time.

SIFT	Dense	MSD
201.52	81.90	267.89

Table 4.1 Feature Extraction Time (s)

The codebook is generated using the Elkan's speedup [81] version of K-means clustering with robust initialization [82]. The computation time is of complexity of $O(N \cdot k)$, which is increasing linearly with the codebook size total number of features N and number of cluster centers k .

Then, for each sampling method, codebook sizes are chosen as 100, 200, 500 and 1000 respectively for experiments. Figure 4.7, Figure 4.8 and Figure 4.9 are the retrieval results of salient SIFT sampling, dense sampling and MSD sampling of different codebook sizes. It is shown that the optimal codebook sizes for different sampling methods are different. When $K=500$, it achieves best accuracy for the salient SIFT sampling method. And when $K=1000$ and $K=200$, dense sampling and MSD sampling achieve the best retrieval accuracy. Although the codebook size has certain impact on the retrieval accuracy, the results do not show significant difference. The optimal codebook sizes for each of the sampling methods are used for later comparison.

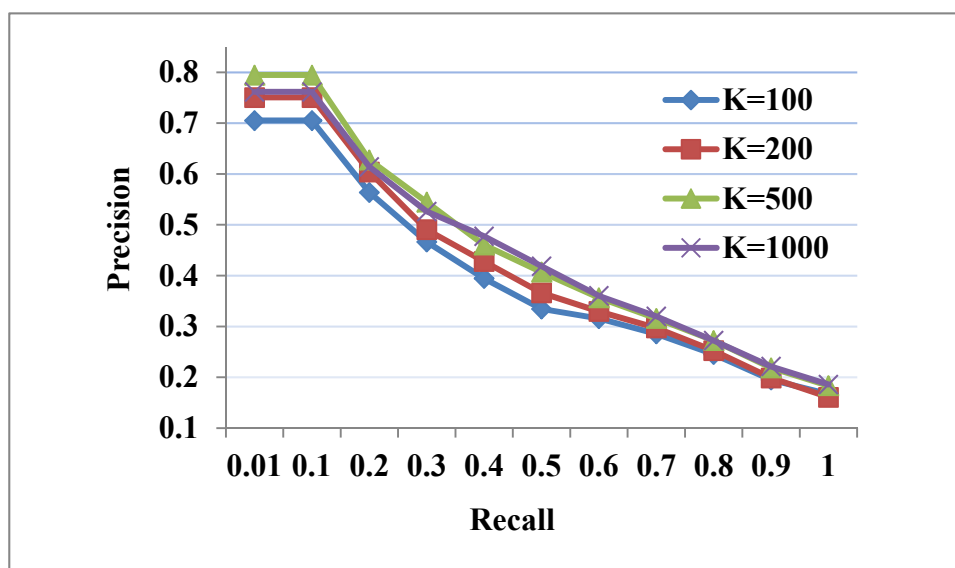


Figure 4.7 Influence of codebook size for original SIFT sampling.

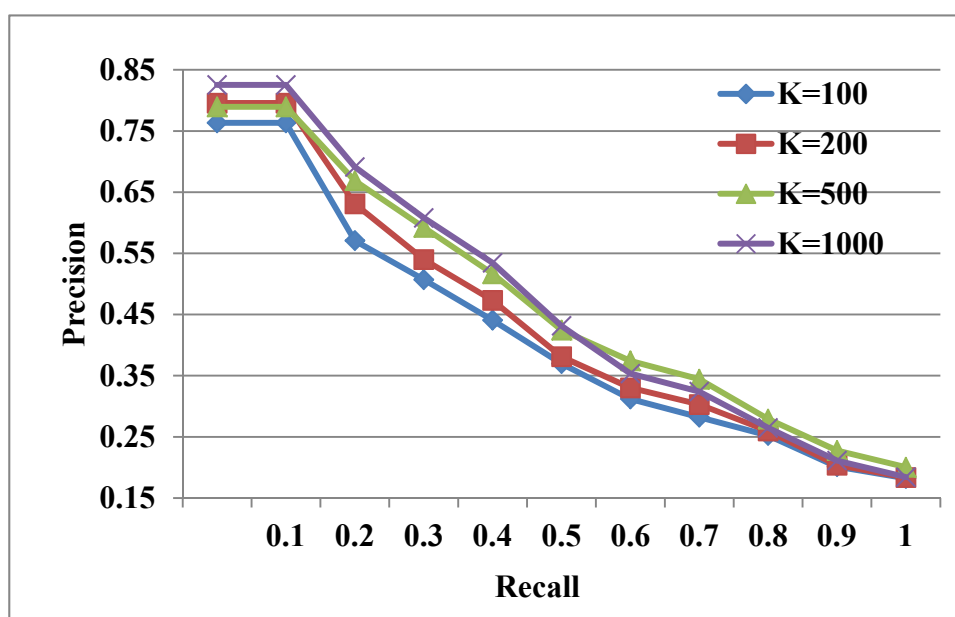


Figure 4.8 Influence of codebook size for modified dense sampling.

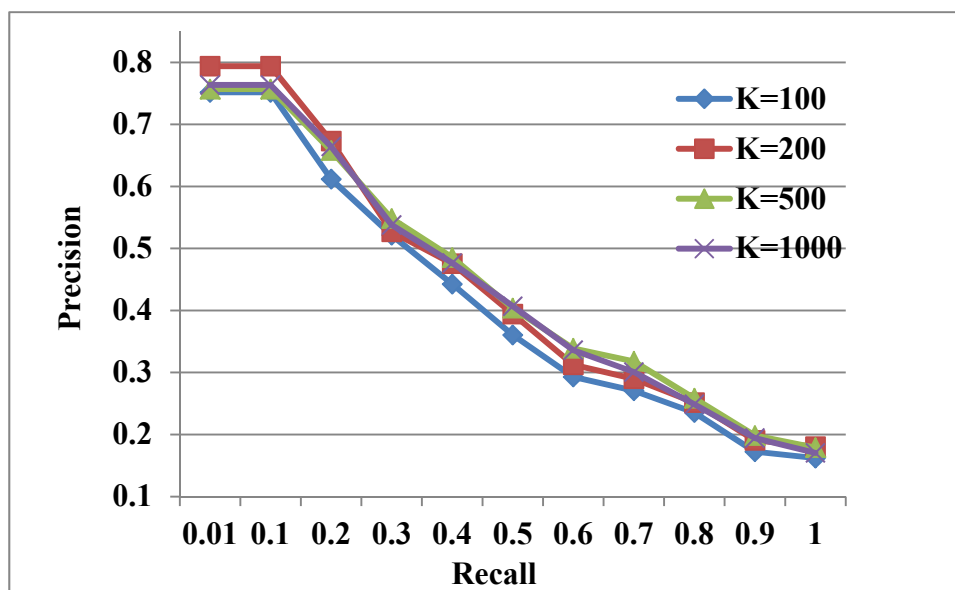


Figure 4.9 Influence of codebook size for MSD sampling.

Normalized L1 distance and Maximum Histogram Intersection Distance are compared for the dissimilarity computation. The results are shown in Fig. 4.10. The L1 distance and MHID are identical for the proposed dense sampling and MSD sampling methods. This is because L1 and MHID degrades to the same distance metric if the total sum of the histograms are the same, that is, when the number of features for each model is fixed. However, a big gap appears using MHID and L1 for the original SIFT detection method. MHID apparently outperforms Normalized L1 distance when the number of features for each model is different. This also suggests that the proposed dense sampling and MSD sampling are more robust to different distance metrics when the original SIFT sampling fails to do so.

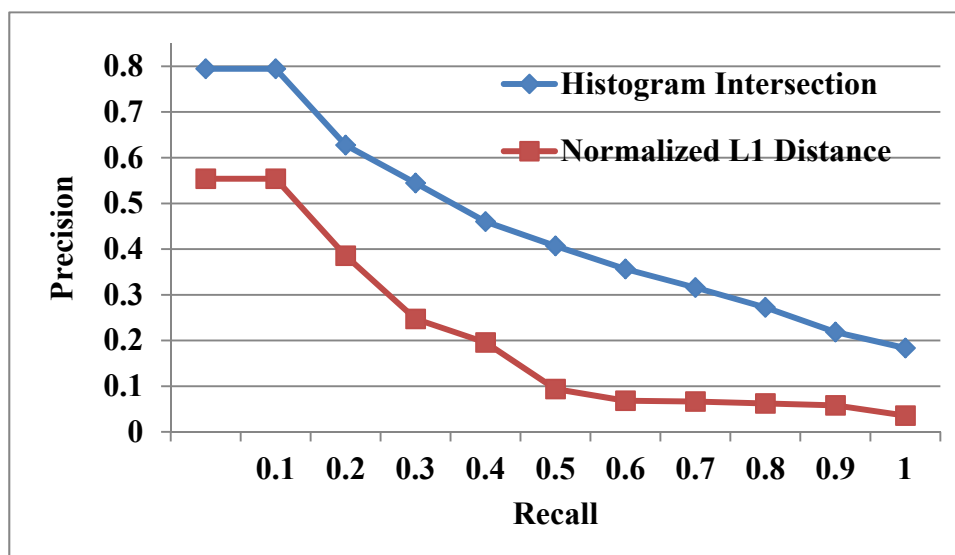


Figure 4.10 Influence of distance metric for original SIFT sampling.

Figure 4.11 gives an example of the retrieved items using a bearing block as a query example. It is shown that both of the proposed modified dense sampling and MSD sampling show better retrieval accuracy than the original SIFT sampling.

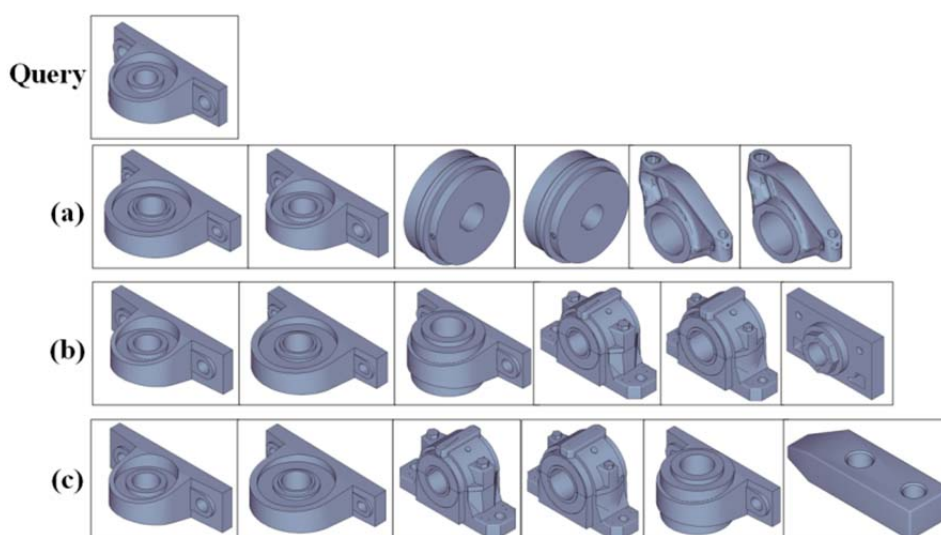


Figure 4.11 Retrieval examples of sampling methods: (a) original SIFT sampling, (b) modified dense sampling, and (c) MSD sampling.

The overall precision-recall retrieval accuracy is illustrated in Fig. 4.12. We choose the optimal codebook size for each of the three sampling methods, and use MHID as the distance metric. It is shown that modified dense sampling with codebook size of 1000 achieves the best retrieval accuracy. MSD sampling shows only slightly better retrieval accuracy than the original salient sampling method before the recall of 0.5, after which the retrieval accuracy decreases slightly.

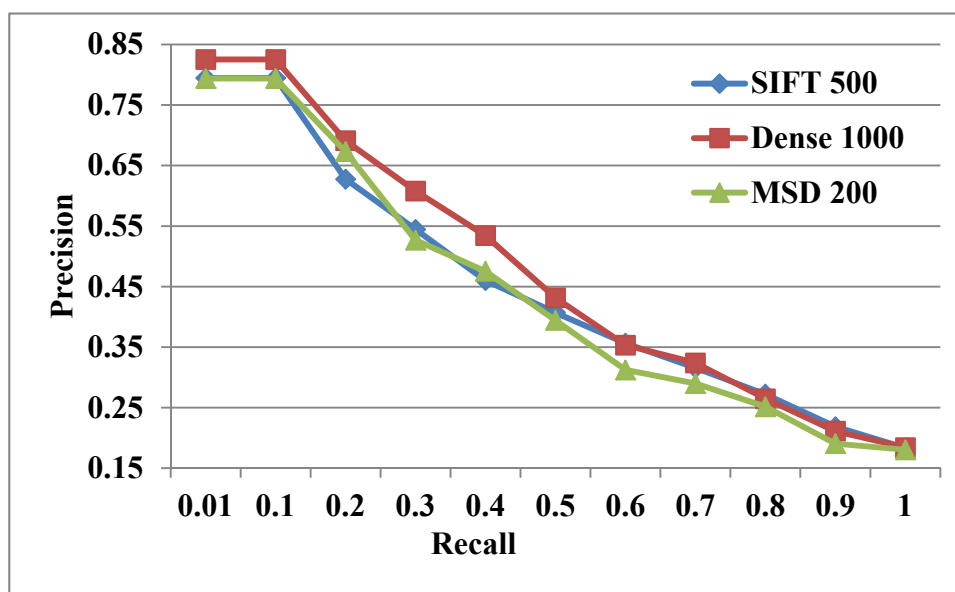


Figure 4.4 Retrieval accuracy using SIFT, modified dense and MSD sampling.

The results above show that dense sampling achieves generally better retrieval results than the original SIFT sampling; while the MSD sampling achieves similar retrieval accuracy as SIFT sampling. These findings suggest that modified dense sampling is more suitable for the tasks of 3D CAD model retrieval. This could be attributed to the reason that dense sampling detects the features covering full spatial range of the model at a fixed scale, and MSD sampling involves features at multiple scales.

To summarize, the proposed dense sampling and MSD sampling have effectively improved the richness of feature representation which could cover the full range of piece-wise smooth shapes. With a flat windowing function, modified dense sampling and MSD sampling are much faster than the original SIFT feature extraction using a Gaussian circular windowing function.

4.4.2 Retrieval Results on NIST Generic Shape Benchmark

In this section, original SIFT sampling, modified dense sampling and MSD sampling are tested on the NIST generic shape benchmark [77]. The 6-view depth images are generated from the 3D models for comparison. For original SIFT detection, there are average 481 features extracted for each model. The window size and sampling step for the modified dense sampling are 16, which results in 169 features per depth image and therefore 1014 features per model. MSD sampling extracts features at scales of 4, 6, 8 and 10 at every step of 24, which extracts 2400 features per model. The codebook is generated with different size from 100 to 2000. Figure 4.13 to Figure 4.15 show the precision-recall curves for original SIFT sampling, modified dense sampling and MSD sampling of 6-view depth images. The similarity distance is computed using Maximum Histogram Intersection Distance (MHID).

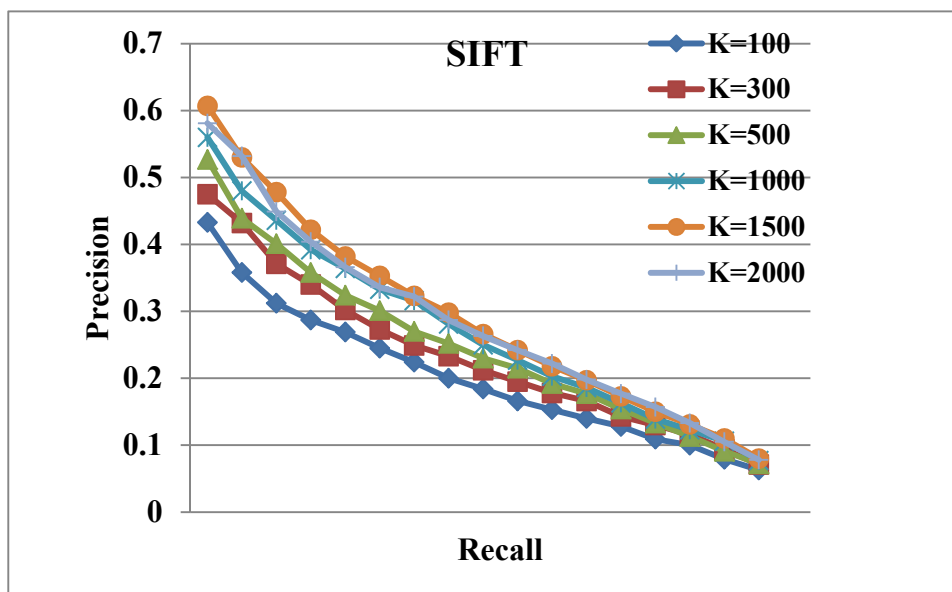


Figure 4.5 Influence of codebook size for 6-view SIFT sampling.

It can be seen that when the codebook size equals to 1500, the original SIFT sampling achieves best retrieval accuracy. For the modified dense sampling, when the recall level is less than 0.3, the K=1000 gives the better retrieval accuracy. When the recall level increases after 0.3 till 1, K=1500 shows the best retrieval accuracy among all curves. MSD sampling gives the best result when the codebooks size equals to 1000.

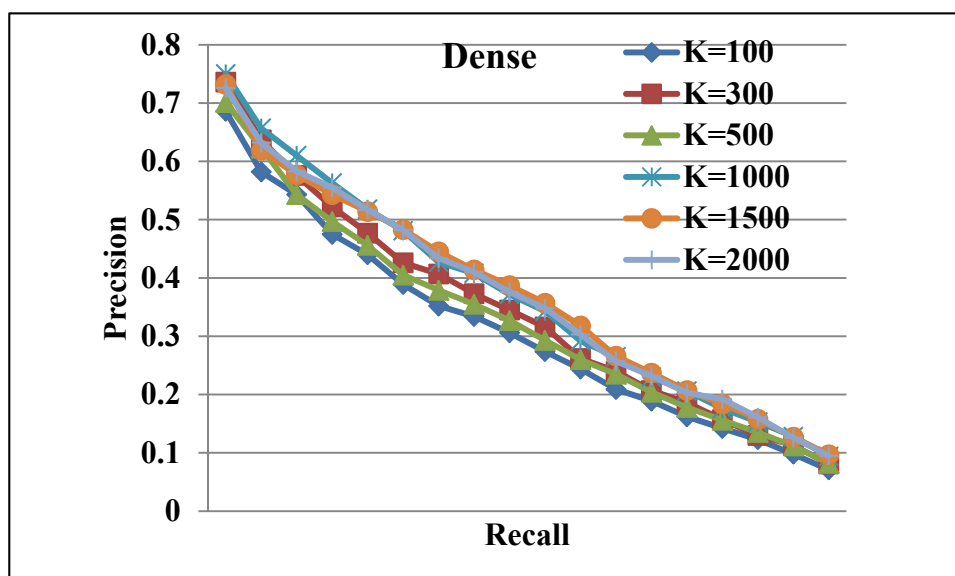


Figure 4.6 Influence of codebook size for 6-view modified dense sampling.

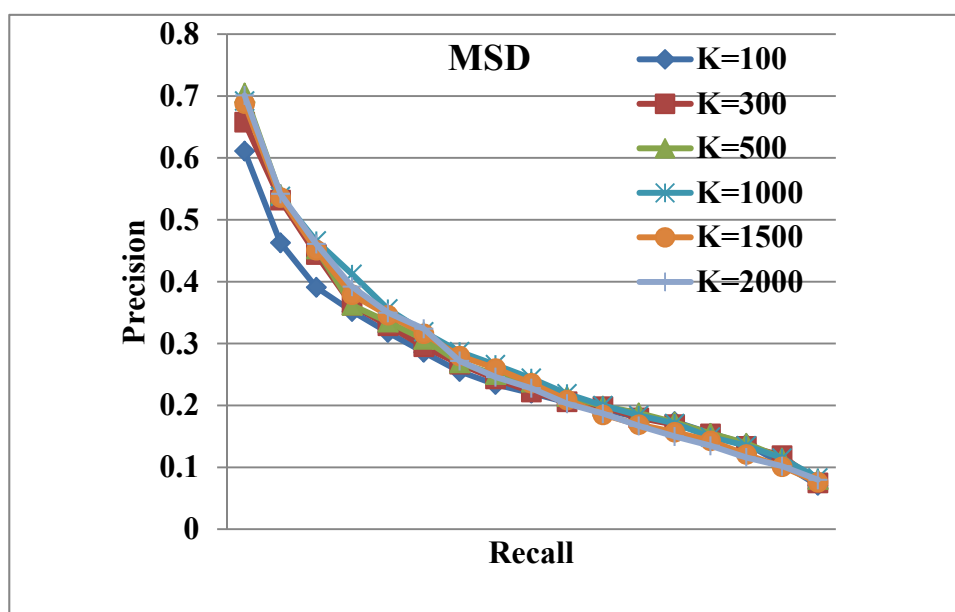


Figure 4.7 Influence of codebook size for 6-view MSD sampling.

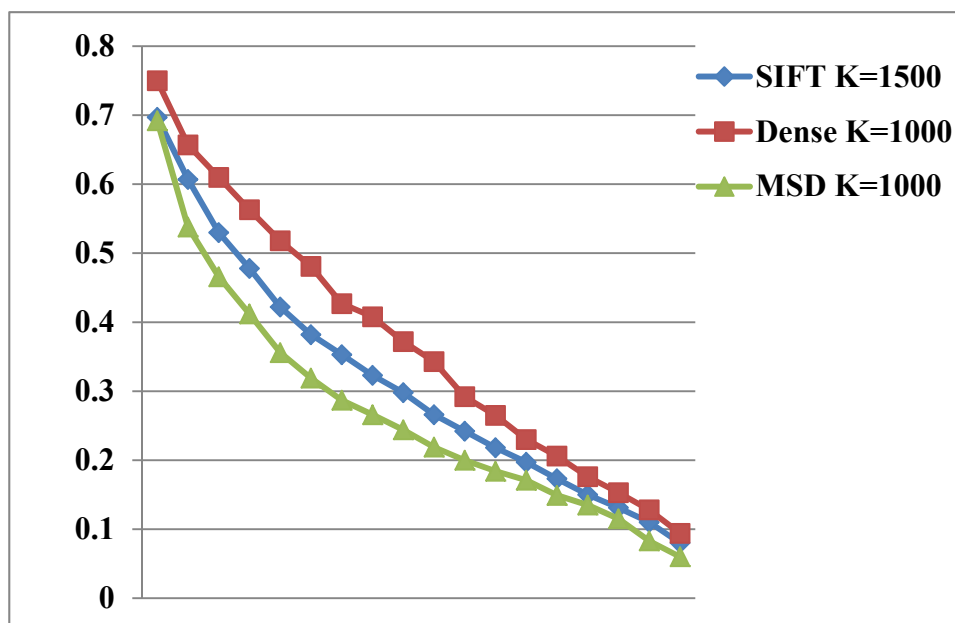


Figure 4.8 Overall comparison of precision-recall results for 6-view SIFT sampling, modified dense sampling and MSD sampling.

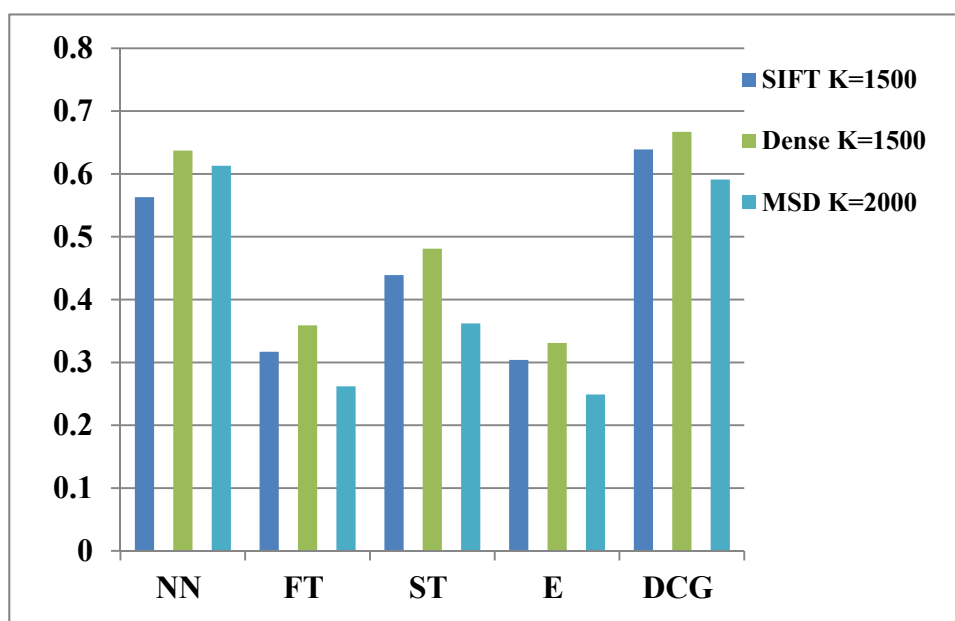


Figure 4.9 NN, FT, ST, E-measure and DCG measures for 6-view SIFT sampling, modified dense sampling and MSD sampling.

The three sampling methods with different codebook sizes are compared in terms of the Precision-recall curve (Figure 4.16) and other statistical values (Figure 4.17). The

results show that dense sampling has achieved the best performance and original SIFT sampling comes at the second.

4.4.3 Retrieval Results on SHREC 2009 Partial Dataset

This section shows examples of the proposed feature sampling strategies for 3D partial model retrieval using bag-of-words model. The parts query set from the SHREC 09 partial dataset is used to compare with the complete target models. These parts models are obtained by cutting parts from complete models. For SIFT sampling, there are an average of 447 features for each query model compared to 481 features of target models. The same parameters for modified dense sampling and MSD sampling are adopted for experiments. As the modified dense sampling and MSD sampling extract features on fixed uniform grids, they do not show different for extraction of features on partial models from complete models. Therefore, there are also average number of 1014 features and 2400 features for the modified dense sampling and MSD sampling as in section 4.4.2.

Experiments are conducted to investigate the optimal performance for the matching and retrieval of 3D parts models by varying dictionary size K , and sampling parameters.

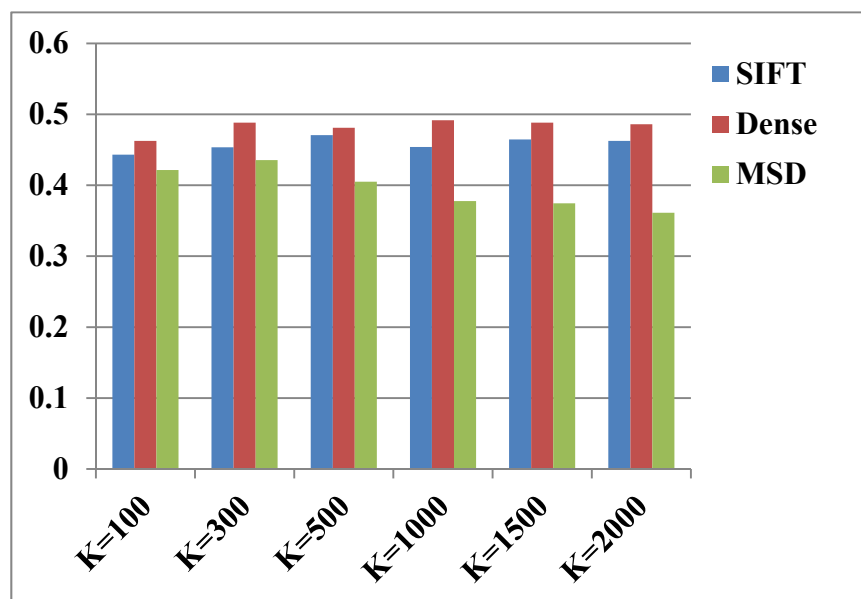


Figure 4.10 DCG measures for 6-view SIFT sampling, modified dense sampling and MSD sampling on SHREC 2009 Partial Dataset.

Figure 4.18 shows the DCG values of different codebook size K for the SIFT, modified dense and MSD sampling. It shows that the DCG value for modified dense sampling is higher than the other two methods for every codebook size K . The codebook size has shown less obvious trends for SIFT and modified sampling, and increase the DCG value first and decrease it after $K=300$ for the MSD sampling.

	K	NN	FT	ST	DCG	E	MAP
SIFT	500	0.1	0.1389	0.2389	0.4534	0.1540	0.1410
Dense	1500	0.3	0.1833	0.2778	0.4858	0.1940	0.1762
MSD	300	0.3	0.1111	0.1583	0.4368	0.1060	0.1147

Table 4.2 NN, FT, DCG, ST, E-measure, and MAP for 6-view SIFT sampling, dense sampling and MSD sampling with optimal codebook size.

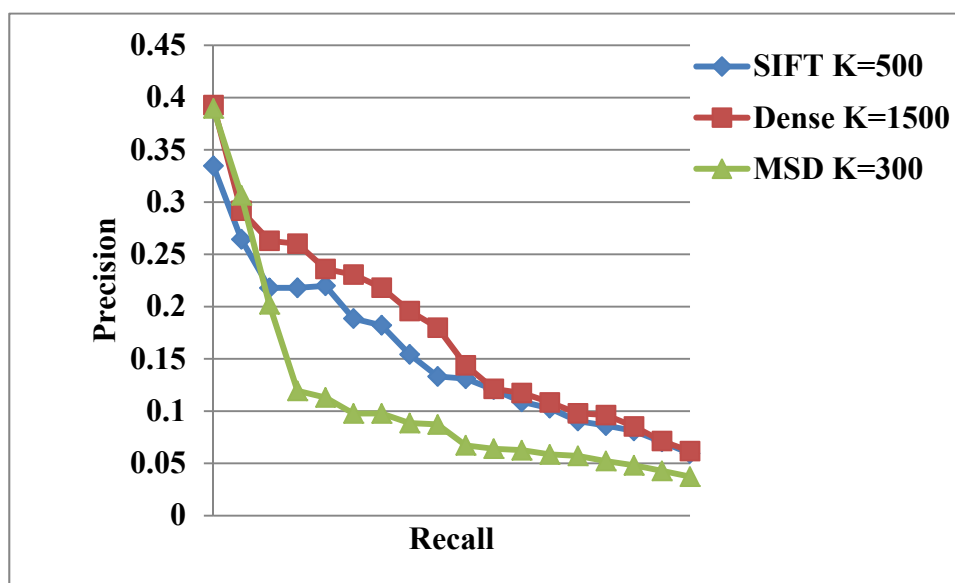


Figure 4.11 Overall comparison of precision-recall results for 6-view SIFT sampling, dense sampling and MSD sampling with optimal codebook size.

Table 4.2 provides other statistical values for the three sampling methods at their optimal codebook size K . It shows that the modified dense sampling has achieved highest statistical values except for the MSD gives best nearest neighbor results. The precision-recall curves of the three methods are shown in Figure 4.19, which also shows that the modified dense sampling has better retrieval accuracy. As sections 4.4.2 and 4.4.3 adopt the same target dataset, it makes the comparison sensible that the overall retrieval accuracy is less than the complete models. This could be explained that only partial information of the 3D models is provided.

In this section, we have tested the proposed methods on SHREC 09' parts query dataset which no previous methods have been tested on and achieved satisfactory

retrieval accuracy. By identifying the optimal sampling strategies for SIFT feature extraction, we find that the modified dense sampling have given the best retrieval accuracy.

4.5 Summary

In this chapter, modified dense sampling and multi-scale dense (MSD) sampling of local features using SIFT description are proposed to extract features from 3D mesh models. The modified dense sampling is to extract features on uniformly distributed grids and MSD sampling is to extract features at multiple scales on the same grids as it. In combination with bag-of-words models, the proposed modified dense sampling have shown better performance over the original SIFT sampling. With a flat window to substitute circular Gaussian window, the feature extraction time for dense sampling and MSD sampling are order of magnitude faster than the original SIFT sampling. Experiments on 3D CAD models, 3D multimedia models, and 3D partial models all have demonstrated the effectiveness of the proposed methods.

Chapter 5 REGION-BASED FEATURE DETECTION AND REPRESENTATION FOR 3D MODEL RETRIEVAL

5.1 Introduction

Using SIFT features with bag-of-words model has gained appealing results for 3D model retrieval tasks compared to other view-based methods. Besides for the efficiency of bag-of-words model, the SIFT feature itself is a rich descriptor as it captures substantial amount of information of spatial intensities. However, this kind of salient feature detection algorithm is not only of high dimensionality, but also very complicated to compute. The simplicity and better performance of modified dense sampling and MSD sampling of local features using SIFT description proposed in last chapter show hints that simple region based feature descriptor on uniform grids might be more suitable for 3D model representation for retrieval tasks. Given a dataset of 800 models and 6 depth images extracted for each model, the SIFT feature might require about 55MB storage. And if a higher sampling density is chosen for dense sampling, the computational cost could be unaffordable for the codebook generation using K-means clustering.

In this chapter, two region based feature descriptors are proposed. These two features are not only of lower dimension, but are simpler to compute than the SIFT features without degeneration of performance. The proposed feature detectors are used to

extract features from depth images of 3D models and are then used as inputs for bag-of-words model based representation of 3D models. The experimental results are encouraging. In the next section, the two region based feature detectors are introduced.

5.2 Region Speeded-Up Robust Feature (RSURF) and Histogram of Oriented Gradients (HOG) Descriptor

In order to compute these features very rapidly at many scales we introduce the integral image representation for images. The integral image can be computed from an image using a few operations per pixel.

The Region-SURF (RSURF) feature is to use the SURF-like descriptor to describe local image regions as features for shape representation. Unlike Speeded-Up Robust Features (SURF) detection [83], the RSURF feature does not involve complicated steps to detect the scale and orientation invariant locations of interest points. Instead, the RSURF feature can be constantly computed at any scale and location once given the sampling density and region size.

The idea of RSURF feature detection is to sum Haar wavelet responses over local image regions. The Haar wavelet response is natural choice for discretized image intensity computation. It is a discontinuous orthonormal function on the unit interval between 0 and 1, where the mother wavelet function $\psi(t)$ is given as

$$\psi(t) = \begin{cases} 1 & 0 \leq t < 0.5 \\ -1 & 0.5 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (5.1)$$

To extract the RSURF feature located at (x, y) on the image at scale σ , a square window of size 20σ is firstly centered at this location. Instead of computing the gradients inside the window as a whole, the window is split to 4×4 sub-regions to retain the local geometric and spatial information. For each sub-region, the Haar wavelet responses $F(\sum dx, \sum |dx|, \sum dy, \sum |dy|)$ can be computed at the 5×5 regularly sampled points, where dx and dy are the Haar wavelet responses in the horizontal and vertical directions respectively. Then, this RSURF feature can be obtained by concatenating the Haar wavelet responses for the 4×4 sub-regions together, which results in a descriptor of $4 \times 4 \times 4$ dimension. The Region-SURF feature detection and description is summarized in Algorithm 2.

Algorithm 2 Region-SURF Detection and Description

Given an image space $I(x, y)$ of the spatial range of 256×256

Step 1 Determine the window size σ and sampling step s

Step 2 Compute the integral image of I as I_Σ

While the sliding window (from top left to bottom right)

Step 3 Split the sliding window into a 4×4 sub-region

Step 4 Compute the Haar wavelet responses $(\sum dx, \sum |dx|, \sum dy, \sum |dy|)$ via subtractions of the integral image I_Σ for each sub-region

Step 5 Concatenate the Haar responses for the 4×4 sub-regions together to obtain a descriptor of 64 dimension

end

Algorithm 2 Region-SURF detection and description.

The four kinds of Haar responses are illustrated in Figure 5.1. To make the descriptor

more self-contained, the absolute value of responses $|dx|$ and $|dy|$ are also included to make a distinction between the gradual changes (Fig. 5.1 (b)). And alternating pattern (Fig 5.1 (d)).

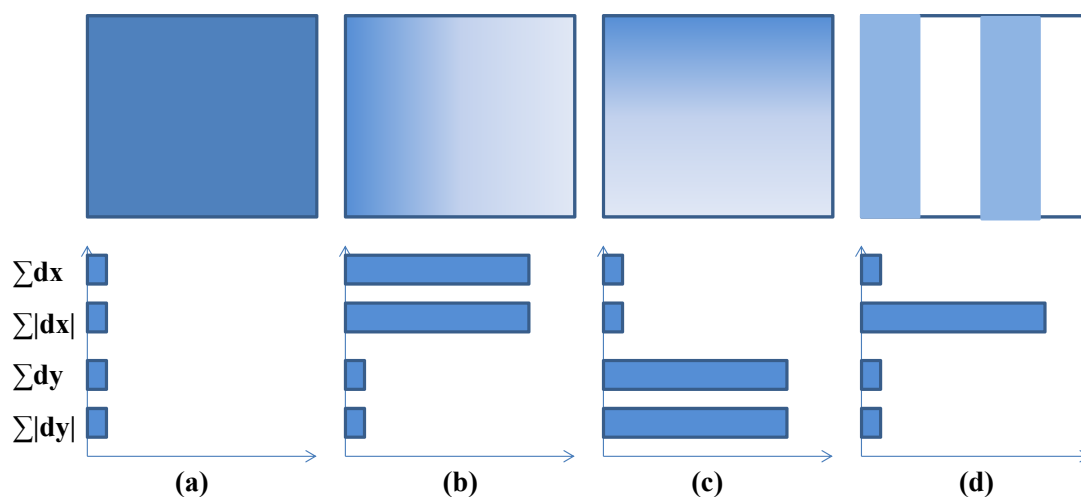


Figure 5.1 Haar wavelet responses for four patterns of image intensity changes [83].

The resulted shape descriptor is of dimension 64, where $(\sum dx, \sum |dx|, \sum dy, \sum |dy|)$ are concatenated for the 4×4 sub-regions. The feature description based on Haar wavelet responses is shown in Figure 5.2. The left figure depicts the 4×4 sub-region placed at the center of image point (x, y) . The right figure shows the image intensity gradients are computed over the 5×5 sub-regions, where examples of dx and dy are labeled.

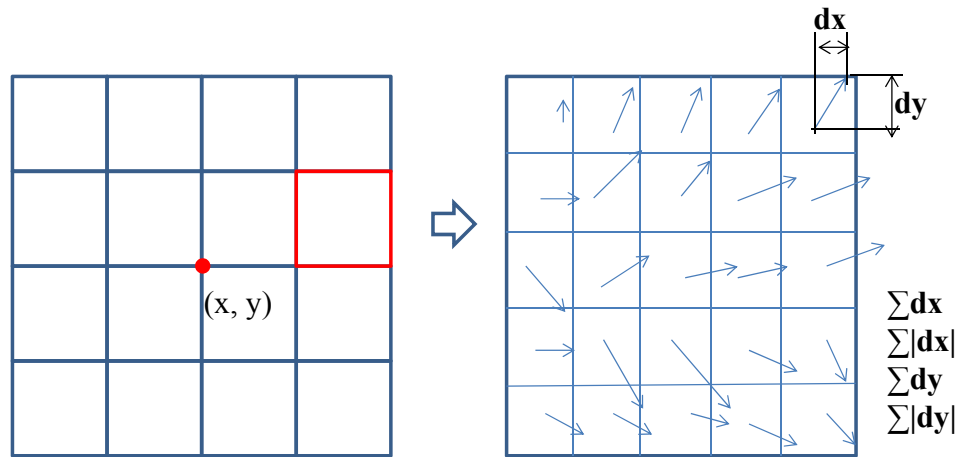


Figure 5.2 Illustration of DSURF feature representation based on Haar wavelet responses of a 4×4 sub-region centered at the interest point.

To make the computation much faster, the computation of integral image is introduced in this thesis. The above mentioned summations of Haar wavelet responses over a region can be easily obtained by several subtractions of rectangular region using integral image, which was firstly introduced in [84]. The integral image $I_{\Sigma}(x, y)$ at a location (x, y) is defined the summation of all pixel values within the rectangular region formed by the location (x, y) and origin of image, which is given as

$$I_{\Sigma}(x, y) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (5.2)$$

Figure 5.3 shows an example to explain the integral image. For example at a random point D, the integral value at that point $I_{\Sigma D}$ means the summation of all intensity gradient values from the origin to the blue shaded area. Therefore, the summation of values over the area of $ABCD$ can be easily computed via equation 5.3, which involves only simple additions and subtractions operation.

$$I_{ABCD} = I_{\Sigma D} - I_{\Sigma B} - I_{\Sigma C} + I_{\Sigma A} \quad (5.3)$$

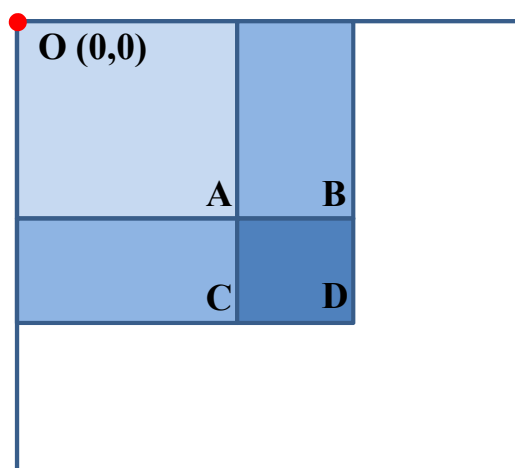


Figure 5.3 Integral images makes the computation of summation of image gradients within the region $ACDB$ is simple as subtracting the integral value at point B and C from point D , and plus the value at point A [84].

We compared our proposed descriptor with another region-based descriptor, Histogram of Oriented Gradients [85] for 3D model retrieval using bag-of-words model. The HOG was originally developed for human detection tasks in images. In this thesis, we modified it as a shape description method for 3D model retrieval tasks. Motivated from the idea that the local shape can be well characterized by the distribution of local intensity gradients, the HOG extracts features at uniformly placed cell blocks. Cell blocks are moved from left to right and top to bottom when forming the final descriptor. The feature size is easily controllable by varying the cell size. Thus the degree of feature robustness and representativeness to local shape deformations can be easily adjusted as well.

The Histogram of Oriented Gradients (HOG) descriptor extraction process is as follows. Assuming the cell size for each feature is s , the image is therefore

decomposed into $\left(\frac{256}{s}\right) \times \left(\frac{256}{s}\right)$ cells. To obtain the image gradient $\nabla g(x, y)$, a 1-D derivative mask $M = (-1, 0, 1)$ is convoluted with the image I , where no pre-smoothing of the image is required, as shown in equation 5.4.

$$\nabla g(x, y) = I(x, y) * M \quad (5.4)$$

Figure 5.4 (a) shows the original depth image, and Figure 5.4(b) gives the result of the depth image after convolution, which is actually the gradient map computed using equation 5.4.

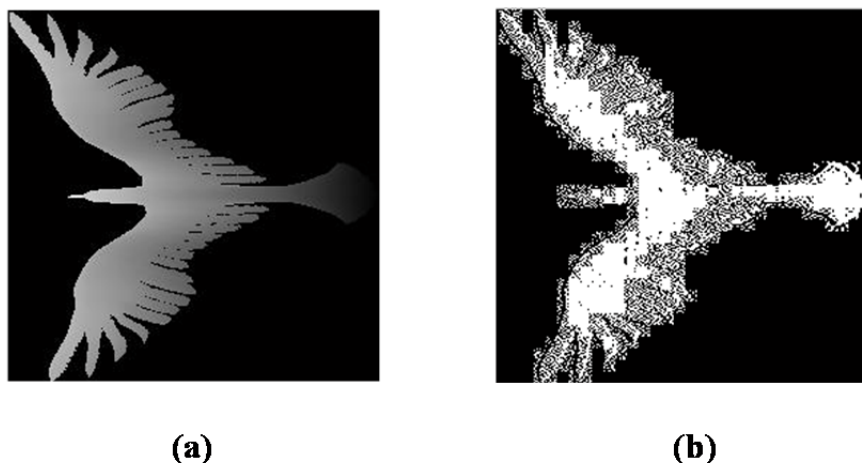


Figure 5.4 Convolution of depth image with 1D mask (-1, 0, 1).

Then the gradients are voted towards orientation bins for each cell weighted by the magnitude of each gradient. In practice, the descriptor achieves optimal performance with the increasing number of bins up to 9 orientation bins. Therefore, this work partitioned the span from 0° to 180° into 9 orientation bins throughout the experiments. To adjust the local illumination variations to the whole image region, normalization of gradient strengths are performed for each block. Adjacent cells are overlapped in order

to achieve good performance. Four normalization methods are adopted simultaneously, namely L2-norm, L2-norm followed by clipping, L1-norm and L1-norm followed by square root are used. The four normalization factors are stacked for each cell. Therefore, the HOG descriptor is of dimension 4×9 . The computation of HOG descriptor is summarized as follows.

Algorithm 3 Histogram of Oriented Gradients for Shape Description

Given an image space $I(x,y)$ of the spatial range of 256×256

Step 1 Compute the gradient image via the convolution of 1-D derivative mask $M=(-1,0,1)$

Step 2 Determine the cell size s

While the next cell(from top left to bottom right)

Step 3 Binning the orientation gradients weighted by gradient magnitude into 9 circular bins evenly spaced over 0° to 180° range

Step 4 Normalize the gradients within each cell using L2-norm, L2-norm flipping, L1-norm, and L1-norm squared root normalization and stack them together

end

Step 5 Collect the HOG descriptors over the cell blocks

Algorithm 3 HOG descriptor computation

5.3 Results and Discussions

In this section, the two region-based feature descriptors, RSURF and HOG, are tested as local depth image descriptors for bag-of-words model for the 3D model retrieval tasks. The experiments are tested on 3D multi-media models from the NIST generic shape benchmark and 3D CAD models from modified CAD dataset.

For the proposed RSURF features, there are two parameters to decide the descriptiveness and distinctness of the features. This first parameter is the region size, which determines the size and sampling density of RSURF feature. Increasing the descriptor region size results in less number of features, but with more information contained but less discrimination power and vice versa. Another parameter, the number of sub-regions for each description block, decides the number of dimensions of the RSURF feature. By default, the number of sub-regions is chosen as 4, which results a feature of dimension 64 as introduced in previous section. In this work, we also tests the situation when the number of sub-regions is reduced to 2, then the dimension of the feature is degraded to 16. Table 5.1 shows the resulted RSURF feature dimension and number of features extracted per image with respect to different region size and number of sub-regions.

	Region Size	Num of Sub-regions	Dimensions	Num of features per image
RSURF44	4	4	64	144
RSURF24	2	4	64	784
RSURF42	4	2	16	196

Table 5.1 RSURF feature with different region size and number of sub-regions

The feature extraction time for the proposed RSURF features and the SURF feature [83] are given in Table 5.2. It can be seen that the proposed RSURF features are much faster than the original SURF features, nearly by two orders of magnitude. This fast

feature extraction speed is very appealing in real applications when tons of models are available in a dataset. The feature extraction time also increases linearly with the sampling density, but shows little affect by the reduced number of sub-regions.

	RSURF44	RSURF24	RSURF42	SURF
SHREC dataset	42.3s	89.4s	36.5s	6894.4s
Modified CAD dataset	30.7s	29.8s	15.2s	3333.7s

Table 5.2 Feature extraction time (s) for RSURF vs. SURF feature detection

For the HOG descriptor, the cell size can be chosen as different values. And particularly we use only one HOG descriptor to describe an entire depth image. The cell sizes are chosen to be 8, 16 and 32 respectively, which will result in 1024, 256 and 64 features for each image. The HOG features are all of dimension 36. The feature extraction time for the three cell sizes are given in Table 5.3.

Cell Size	cs=8	cs=16	cs=32
SHREC generic dataset	317.0s	328.2s	326.1s
Modified CAD dataset	192.0s	189.0s	192.6s

Table 5.3 Feature extraction time (s) for HOG feature detection

Next, the proposed RSURF features and HOG feature with different region sizes are compiled with K-means clustering to generate a codebook of different size K varying from 100 to 3000.

For RSURF44, RSURF24 and RSURF42, the discounted cumulative gain (DCG) values obtained for the modified CAD dataset and the NIST generic shape benchmark are shown in Figure 5.5 and Figure 5.6 respectively. It can be seen that RSURF24 achieves the highest DCG values on both datasets. The DCG values for RSURF44 are more often slightly larger than RSURF24 with small codebook size K , and a little less accurate than RSURF24 for most cases. Although the RSURF42 has always got the smallest DCG values, but the difference is not too much. Besides, RSURF 44 and RSURF 42 apparently extract far less number of features than the RSURF24, 144 and 196 features per model compared to 784 features per model.

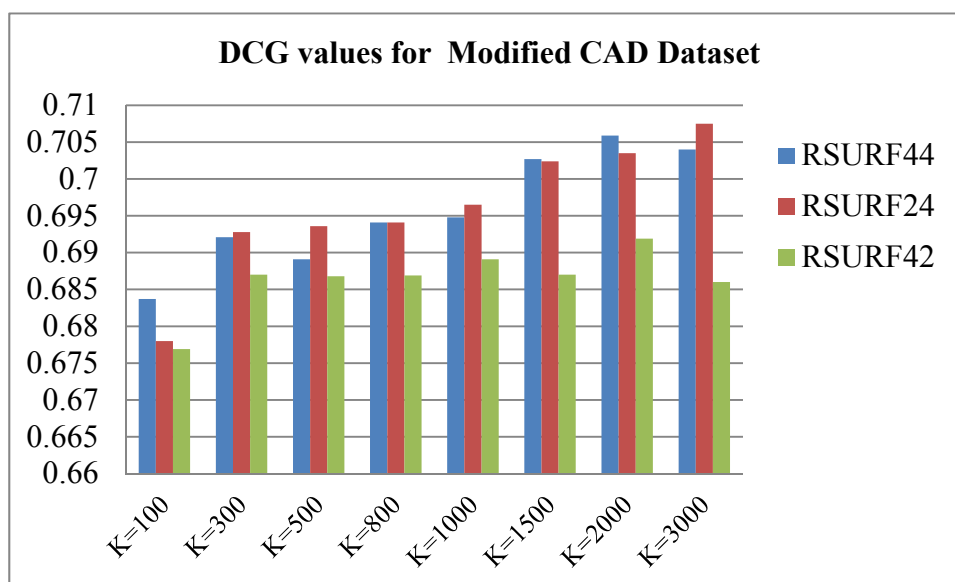


Figure 5.5 DCG of RSURF features on modified CAD dataset for different codebook size K .

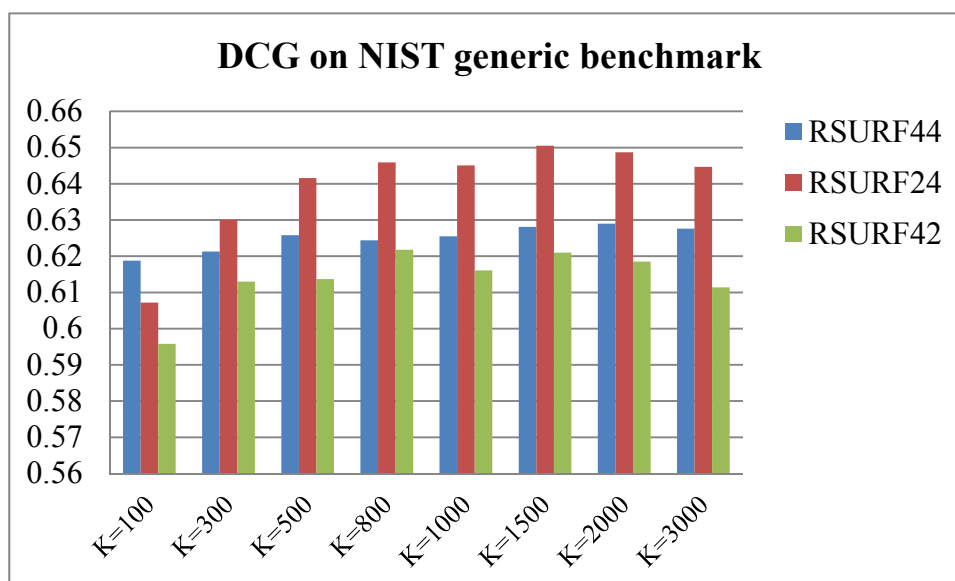


Figure 5.6 DCG of RSURF features on NIST generic shape benchmark for different codebook size K.

For HOG features, different cell sizes 8, 16 and 32 are also tested to obtain the optimal retrieval accuracy by varying the codebook size K from 100 to 3000 on modified CAD dataset and NIST generic shape benchmark. Figure 5.7 shows the DCG values on modified CAD. It can be seen that for cell sizes of 8, 16 and 32, the optimal codebook sizes K are 300, 500 and 500 respectively. Surprisingly, the cell size 32 outperforms all other parameters with much less features extracted, 64 features compared to 1024 features for cell size 8 and 256 features for cell size 16. The DCG values of HOG features tested on NIST generic shape benchmark are also given in Figure 5.8. The cell size 16 achieves the highest DCG value this time. This is probably because the 3D multi-media models contain more local shape variations than the CAD models, therefore smaller cell size can better characterize the local shape of 3D multi-media models.

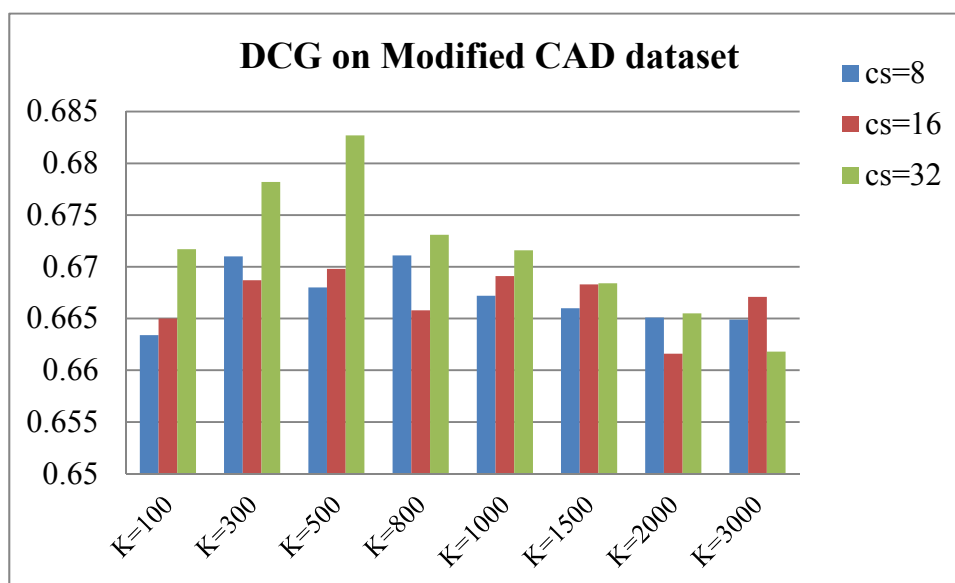


Figure 5.7 DCG of HOG features on modified CAD dataset for different codebook size K .

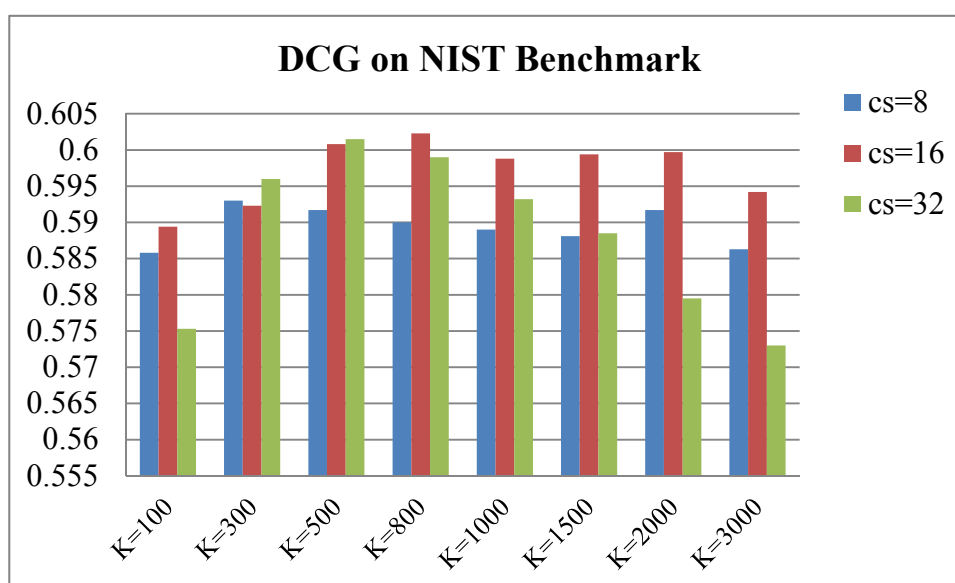


Figure 5.8 DCG of HOG features on NIST generic shape benchmark for different codebook size K .

To compare the two proposed region-based descriptors with salient feature detectors SIFT and SURF, Figure 5.9 shows the precision-recall curves for the four descriptors using bag-of-words model for the retrieval tasks on the modified CAD dataset. The region size and codebook size K which gives rise to optimal performance are used for comparison. The RSURF feature achieves the highest retrieval accuracy, SIFT feature comes at second. The precision for HOG features are lower than the SURF features, but outperform the SURF features after recall level of 0.5. The first-tier (FT), second-tier (ST), discounted cumulative gain (DCG), E-measure (E) and mean average precision (MAP) are also listed in Table 5.4. The results also indicate that RSURF44 has achieved the highest retrieval accuracy.

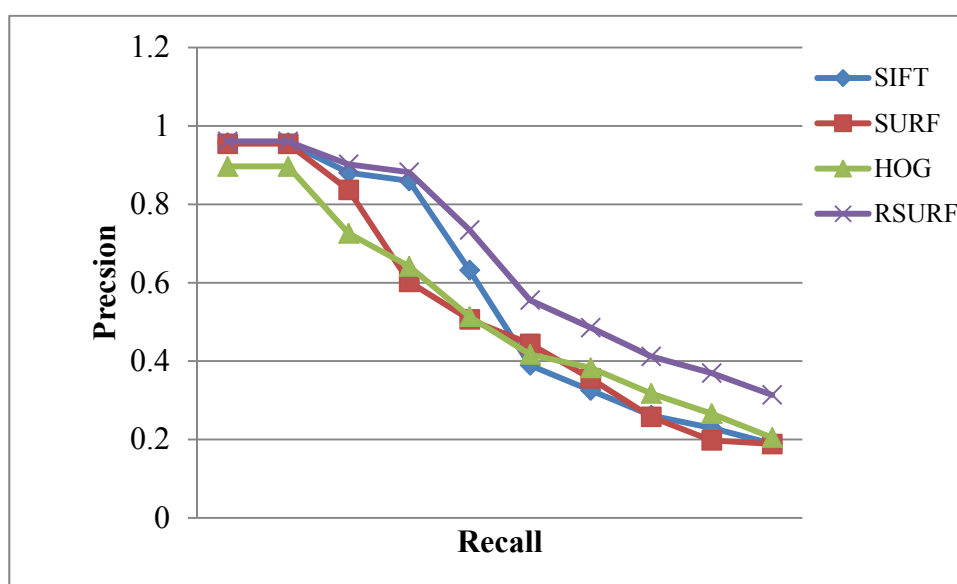


Figure 5.9 Precision recall curve for proposed region-based RSURF and HOG features compared to salient features SIFT and SURF on modified CAD dataset.

	K	FT	ST	DCG	E	MAP
SIFT	3000	0.379	0.468	0.682	0.228	0.425
SURF	1500	0.381	0.456	0.674	0.231	0.414
RSURF44	2000	0.409	0.492	0.706	0.247	0.452
HOG cs=32	500	0.376	0.465	0.683	0.236	0.427

Table 5.4 Other evaluation measures for proposed features vs. SIFT and SURF on modified CAD dataset

The same comparisons have been made on NIST generic shape benchmark. Figure 5.10 and Table 5.5 compares the precision-recall curves and other evaluation measures. The RSURF features again achieve the best retrieval accuracy while SIFT and HOG features come at second and third.

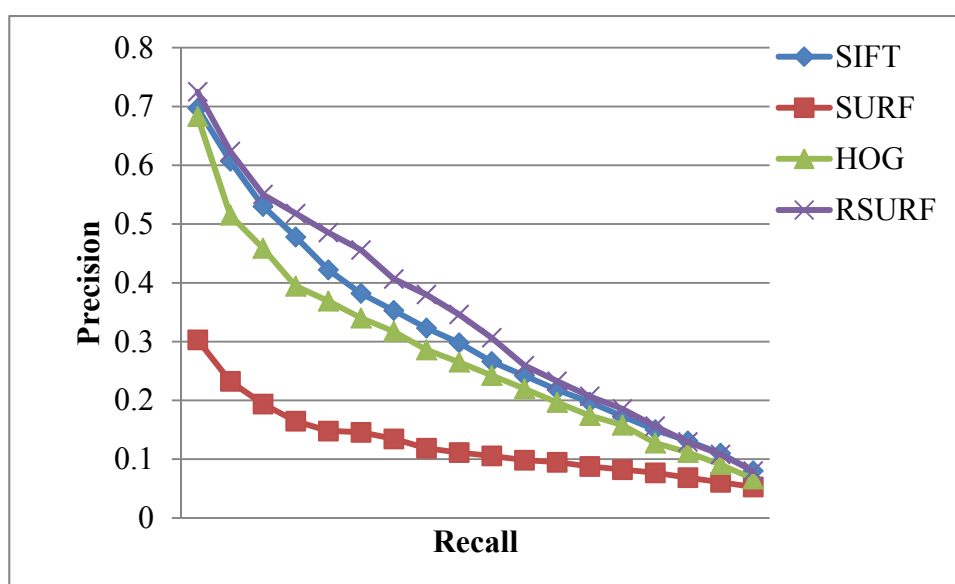


Figure 5.10 Precision recall curve for proposed region-based RSURF and HOG features compared to salient features SIFT and SURF on NIST generic shape benchmark.

	K	NN	FT	ST	DCG	E
SIFT	1500	0.563	0.317	0.439	0.639	0.304
SURF	3000	0.175	0.127	0.211	0.441	0.140
RSURF44	1500	0.650	0.340	0.444	0.651	0.308
HOG cs=16	1000	0.588	0.285	0.412	0.599	0.278

Table 5.5 Other evaluation measures for proposed features vs. SIFT and SURF on NIST generic dataset

5.4 Summary

To summarize, the region-based uniform sampled RSURF and HOG features show superior and similar performance than the salient feature detectors SIFT and SURF with bag-of-words model for 3D model retrieval tasks. The region-based descriptors assume the scale and location of the features are pre-defined, and therefore there is no need for scale-space construction to detect the salient points across different scales with respect to different orientations. This makes the region-based features extraction time much faster than the salient point detectors SIFT and SURF. Meanwhile, to choose optimal parameters for proposed region-based feature detector, suitable region size, fine orientation and coarse spatial binning together influence the descriptiveness and distinctness of the feature descriptor. Besides, the two feature descriptors are of great simplicity in terms of representation, and they are of much less dimensions compared to the SIFT descriptor with dimension of 128.

Chapter 6 LARGE-SCALE 3D MODEL CATEGORIZATION USING MULTI-CLASS SVM WITH LINEARLY APPROXIMATED KERNEL

6.1 Introduction

In Chapter 4 and Chapter 5, two feature detection methods using bag-of-words model for representation are investigated for the 3D model retrieval problem. In the retrieval scenario, given a query model, each of the stored models in the target dataset is compared with the query and the models with small similarity distance are retrieved as relevant models with respect to the query. But when the number of target models grows too large, it is not computationally affordable or efficient to compare the query model with every target model. Therefore, there is a need to develop a learning-by-example approach, which can assign a query example to a class of similar models from the knowledge learned from existing models without explicit comparison with all models in a dataset. Such process is called 3D model categorization.

In this chapter, a 3D model categorization scheme is devised. The 3D models are firstly represented as histograms of visual words obtained by bag-of-words representation. After that, the histogram shape descriptors are fed into multiple-class Support Vector Machine (SVM) with non-linear kernel, specifically, chi-square kernel

and histogram kernel are adopted. The non-linear kernels can be approximated by addition of linear homogeneous feature maps, which could significantly increase the training and classification speed.

Next, the 3D model categorization procedures and the linearly approximated kernelized multi-class SVM will be given in details. And examples for the categorization of 3D models will be followed.

6.2 3D Model Categorization with Multi-class Kernel SVM

6.2.1 Bag-of-Words Representation for Categorization of 3D Models

Given 3D models, which are pre-classified into n categories, they are firstly aligned into canonical pose. Then depth images are extracted from the aligned mesh models. Local features, which could be the features proposed in Chapter 4 and Chapter 5, are extracted and codebook can be learned via K-means clustering from all the features. Then the 3D models can be represented as histogram of visual words, which are the shape descriptors for the next stage of training classifier. As shown in Figure 6.1, after all pre-classified models are represented using the bag-of-words model, a classifier is learned for every two classes, which gives rise to a total number of $n * (n - 1) / 2$ classifiers. Similarly, for any new instances of 3D models with unknown classes, they will also be transformed to the bag-of-words representation first, using the codebook

learned for the training models. Finally, they are fed into the whole set of classifiers and a decision of class will be made. Figure 6.1 depicts the categorization procedure of 3D models using bag-of-words representation.

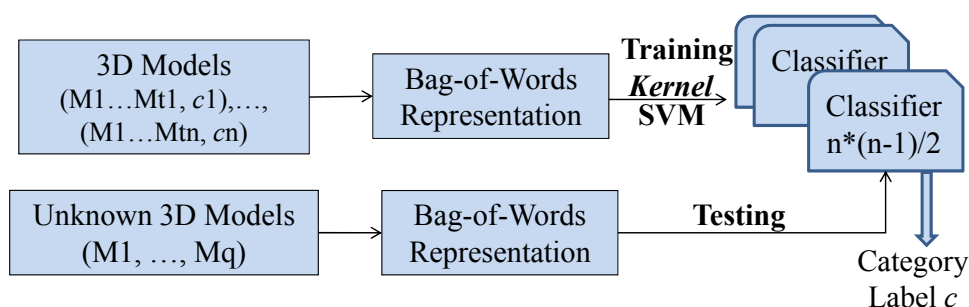


Figure 6.1 Categorization procedures of 3D models using bag-of-words representation.

6.2.2 Non-linear Kernel SVM Approximated by Linear Homogeneous Feature Maps

Support Vector Machine (SVM) is a simple and efficient tool to solve the linearly separable two-category classification problem. Given a set of training data with class labels, SVM trains a model to find a hyperplane which gives largest margin between two classes. Given a training data set D with n samples,

$$D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^d, y_i \in (-1, 1)\} \quad (6.1)$$

where \mathbf{x}_i is an training data in the d -dimensional space and y_i is the class label. SVM is to find the hyper plane $g(\mathbf{x})$ such that the support vectors give rise to a maximum-margin.

$$g(\mathbf{x}) = \mathbf{w} * \mathbf{x} + b \quad (6.2)$$

where $g(\mathbf{x}) = 1$ belongs to one class and $g(\mathbf{x}) = -1$ belongs to the other class. The distance between two hyperplanes $g(\mathbf{x}) = 1$ and $g(\mathbf{x}) = -1$ can be obtained as $\frac{2}{\|\mathbf{w}\|}$. Therefore, to maximize the distance is equal to minimize $\|\mathbf{w}\|$. The primal form of SVM is therefore to solve the minimization of $\frac{1}{2}\|\mathbf{w}\|^2$ subjecting to $g(\mathbf{x}_i) = \mathbf{w} * \mathbf{x}_i - b \geq 1$.

By introducing the Lagrange multipliers α_i , the constrained problem can be reformulated as

$$\min \max \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i (y_i(\mathbf{w} * \mathbf{x}_i - b) - 1) \quad (6.3)$$

The quadratic programming can be changed to the unconstrained dual form of maximizing the Lagrange function $L(\alpha)$, where the hyperplane can be obtained by maximizing $L(\alpha)$, which gives

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (6.4)$$

subjecting $0 \leq \alpha_i \leq C$ and $\sum_{i=1}^n \alpha_i y_i = 0$, where C is an important regularization parameter which controls the tradeoff between complexity of the SVM and number of non-separable points [86].

Whereas the original problem might not be linearly separable in the feature space, it is possible to make the separation easier to map the original finite-dimensional space into a higher-dimensional space. The histogram-intersection kernel and chi-square kernel are the natural candidates for the histogram based shape descriptors. As distance measures, the histogram intersection distance and chi-square distance can be

interpreted as comparing a test histogram to each of the supported histograms. The histogram intersection kernel is given by

$$K_{HI} = \min(\mathbf{x}_i, \mathbf{x}_j) \quad (6.5)$$

And the chi-square kernel is

$$K_{\chi^2} = 2 \frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{\mathbf{x}_i + \mathbf{x}_j} \quad (6.6)$$

By substituting the kernel $K(\mathbf{x}_i, \mathbf{x}_j)$ into the dual-form of Lagrange function, the Lagrangian equation is now as

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (6.7)$$

The decision function is also incorporated with the kernel function, resulted in $g(\mathbf{x}) = \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b$.

The non-linear chi-square kernel and histogram intersection kernel can be approximated with a finite series of additive homogeneous feature maps [87]. A homogeneous map of order n therefore can be used to encode the feature x into a higher dimension of $2n + 1$ $\Psi(x)$. It is equivalent to use a non-linear kernel with Support Vector Machine (SVM) for training and testing using the mapped data. Therefore, the kernel $K(x, y)$ can be interpreted as mapping the feature x into Hilbert space such that

$$K(\mathbf{x}, \mathbf{y}) \approx \langle \Psi(\mathbf{x}), \Psi(\mathbf{y}) \rangle \quad (6.8)$$

The homogeneous map $\Psi(x)$ can be constructed in a compact and closed form and data-independent. By making a kernel signature K periodic, it can be derived as a finite approximation to duplicate with period Λ . The periodicization \hat{K} can be

written as

$$\hat{K} = \sum_{k=0}^{+\infty} W(\lambda + k\Lambda)K(\lambda + k\Lambda) \quad (6.9)$$

Where $W(\lambda) = \lambda/\Lambda$ is a rectangular windowing function.

6.2.3 Multi-class SVM categorization

In this research, the multi-category classification is reduced to multiple one-versus-one classification problems. Suppose there are c classes, for every pair of classes, a classifier is learned as a two-class support-vector machine problem, and a total number of $c * (c - 1)/2$ classifiers needed to be trained. The multi-class categorization problem is illustrated in Figure 6.2. An example of four-class classification is shown, where 6 hyperplanes (H12, H13, H14, H23, H24, H34) are found.

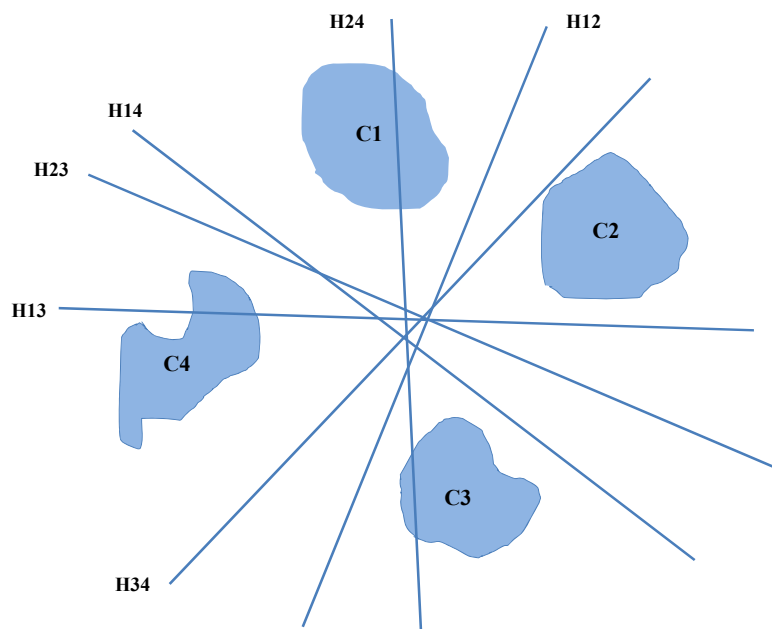


Figure 6.2 Illustration of the multi-class classification problem [71].

Given a new model x , it will be fed into each of the $c * (c - 1)/2$ classifiers and assigned to c_i if $g_i(x) > g_j(x)$ for all $j \neq i$, where the decision function is given as

$$g_i(x) == \sum \alpha_i y_i K(x_i, x_j) + b \quad \text{for } i = 1, 2, \dots, c \quad (6.10)$$

Then the maxi-win voting strategy is adopted. Each of the assigned class gets one vote and the instance is finally assigned to the class which gets the most votes.

The categorization of 3D models using homogeneous map approximated non-linear kernel multi-class SVM as described in section 6.2.2 and section 6.2.3 can be summarized in Algorithm 4.

Algorithm 4 3D Model Categorization using homogeneous map approximated kernel multi-class SVM

Given the 3D models for training/testing

Step 1 Compute the histogram descriptor of the 3D model using bag-of-words model representation

Step 2 Map the histogram descriptor into the $(2n+1)$ dimension of homogenous space, either by the histogram intersection kernel map or the chi-square kernel map

Step 3 Using the mapped data in step 2 as inputs for training/testing via SVM and obtain the (SVM weights/output class labels).

Algorithm 4 Categorization of 3D models using homogeneous map approximated kernel multi-class SVM

6.3 Results and Discussions

In this section, the proposed 3D model categorization system will be demonstrated on

the NIST generic shape benchmark for the categorization of 3D multimedia models and the modified CAD dataset of 3D CAD models. Although the dataset used for experiment is not extremely large-scale, we only want to demonstrate the effectiveness of our proposed methods. In fact, larger amount of training and testing models might result in better categorization results. The primal estimated sub-gradient solver for SVM [88] and the VLFeat implementation [89] is adopted for classifier training and testing.

The classification accuracy is evaluated by the percentage of correctly assigned models. Figure 6.3 shows the convergence of energy for the SVM training is achieved around 200 iterations. We forces the training of SVM converged or until a maximum iteration of 2000.

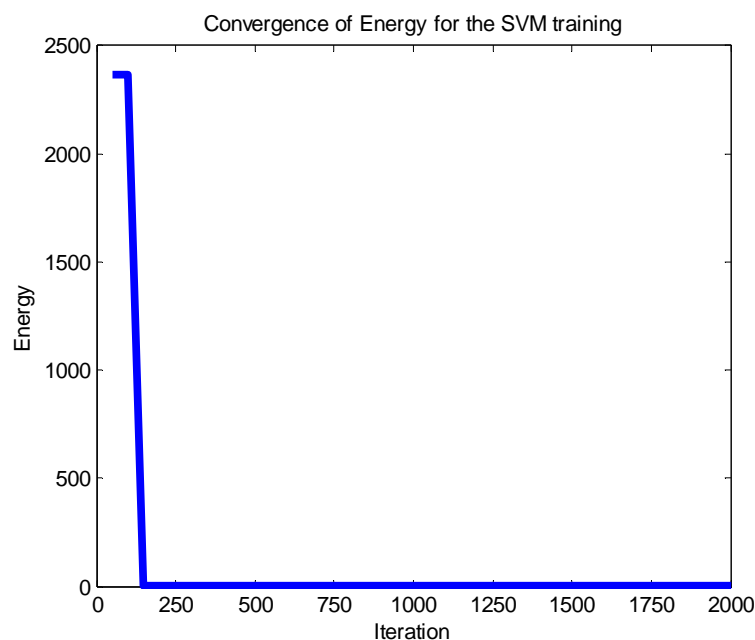


Figure 6.3 Convergence of SVM energy for training

6.3.1 Classification Results on the NIST Generic Shape Benchmark

The query models as testing examples and target models as training examples for the NIST generic shape benchmark,. Thus there are 18 models for each class to training and 2 models for testing.

Firstly, the regularization parameter C and feature map dimensions are studied to find the optimal parameters for the classification while the bag-of-words representation of 3D models are fixed at 500 visual words. Table 6.1 shows the classification accuracy using linear SVM with no kernel incorporated on the NIST generic shape benchmark for different regularization parameter C . It can be seen that when C equals 0.8, the classification accuracy is maximum 0.6125 compared to 0.49 for $C=0.1$.

No Kernel	C=0.1	C=0.5	C=0.8	C=1
Accuracy	0.49	0.55	0.6125	0.5875

Table 6.1 Classification accuracy of SVM without kernel for different regularization parameters C

The classification accuracy of using approximated Histogram Intersection (HI) kernel and Chi-square (Chi2) kernel with the multi-class SVM are given Table 6.2 and Table 6.3 for different regularization parameter C and order of approximation homogeneous map are varied. Note, the histogram descriptors is mapped into a dimension $2 * N + 1$

space given the order is N . The average classification accuracy of using HI kernel and Chi2 kernel is 10%-20% better than SVM with no kernel. The maximum classification rate for HI kernel is when $C=0.5$ and the homogeneous map of order 3. The Chi2 kernel achieves the best classification rate when $C=1$ regardless of the order of approximated map. HI kernel shows generally higher classification accuracy than the Chi2 kernel.

HI Kernel	N=3	N=5	N=10
C=0.1	0.675	0.6875	0.7000
C=0.5	0.75	0.7125	0.7125
C=0.8	0.7375	0.7125	0.7125
C=1	0.7125	0.7375	0.725

Table 6.2 Classification accuracy of histogram intersection kernel for different regularization parameter C and feature dimension.

Chi2 Kernel	N=3	N=5	N=10
C=0.1	0.6750	0.6750	0.6625
C=0.5	0.7000	0.7125	0.7125
C=0.8	0.7000	0.7125	0.7000
C=1	0.7250	0.725	0.725

Table 6.3 Classification accuracy of Chi-square kernel for different regularization parameter C and feature dimension.

Next, the number of histogram bins is varied while the regularization parameter and homogeneous order N are fixed at optimal. It can be seen that finer histogram binning may result in increasing shape representation, hence better classification accuracy. The maximum accuracy is reached at 0.8 when $K=2000$ and further increase of K produces a decay in the performance. Both of the HI kernel and Chi2 kernel produce better retrieval accuracy than the linear SVM.

	Kernel	K=500	K=1000	K=2000	K=2500
C=0.5 N=3	HI	0.7500	0.7750	0.8000	0.7750
C=1 N=3	Chi2	0.7250	0.7750	0.8000	0.7875
C=0.8	No Kernel	0.6125	0.625	0.6875	0.6875

Table 6.4 Overall comparisons for optimal configuration for no kernel, HI and chi2 kernel

The average training time for approximated kernelized SVM is about 11.6s for homogeneous map of order 3 compared to average training time 1.47s for non-kernel SVM. The average testing time for approximated kernelized SVM is 1.26s versus 0.48s for non-kernel SVM. The computation time also increases linearly with the order of homogeneous map and number of visual codebook size K , but shows no difference for the regularization parameter C .

6.3.2 Classification Results on the Modified CAD Dataset

For the modified CAD dataset, we firstly select the classes such that the number of models for each class is equal to 10. There are 34 classes chosen with 9 models for training and 1 model for testing for each class. As stated, although the number of models for training and testing is too small, we only test the proposed categorization system for 3D CAD models for the purpose of demonstration.

Similar as the experiments done for the NIST generic shape benchmark, we first use fixed number of histogram visual words to study the effect of regularization parameter C and homogeneous order N . First, the influence of C for non-kernel SVM is shown in Table 6.5. The classification accuracy is peaked at $C=0.8$.

No Kernel	C=0.1	C=0.5	C=0.8	C=1
Accuracy	0.3529	0.3824	0.4412	0.4118

Table 6.5 Classification accuracy of SVM without kernel for different regularization parameters C

Table 6.6 and Table 6.7 show the classification results of SVM with HI kernel and Chi2 kernel. The two parameters C and N are studied for the influence of classification. When $C=0.1$ and $N=3$, the classification rate for SVM with HI kernel is maximum at 0.5. When $C=1$, the classification rate for SVM with Chi2 kernel is best for $N=3, 5, 10$. The HI kernel also shows better classification results than the Chi2 kernel, while the non-kernel SVM comes at the last.

HI	N=3	N=5	N=10
C=0.1	0.5000	0.4418	0.4706
C=0.5	0.4706	0.4706	0.4412
C=0.8	0.4706	0.4412	0.4706
C=1	0.4706	0.4412	0.4412

Table 6.6 Classification accuracy of histogram intersection kernel for different regularization parameter C and feature dimension.

Chi2	N=3	N=5	N=10
C=0.1	0.4412	0.4412	0.4412
C=0.5	0.4412	0.4412	0.4412
C=0.8	0.4706	0.4412	0.4412
C=1	0.4706	0.4706	0.4706

Table 6.7 Classification accuracy of Chi-square kernel for different regularization parameter C and feature dimension.

Table 6.8 gives the overall comparison for the non-kernel, HI and Chi2 SVM for different number of visual words given optimal regularization parameter C and optimal homogeneous order N. It also shows that when K=2000, the classification rates are the highest for the three situations, and HI kernel SVM performs the best. The overall trend is also the HI kernel performs better than Chi2 kernel, and Chi2 kernel performs better than SVM with no kernel.

	Kernel	K=500	K=1000	K=2000	K=3000
C=0.1 N=3	HI	0.5000	0.5000	0.5588	0.4706
C=1 N=3	Chi2	0.4706	0.5000	0.5000	0.5000
C=0.8	No Kernel	0.4412	0.4118	0.5000	0.4412

Table 6.8 Overall comparisons for optimal configuration for no kernel, HI and chi2 kernel

The computational time for the non-kernel SVM is shortest than the kernel SVM, however, the kernel SVM also only takes a few seconds to train and test the classifier due to the use of linear homogeneous maps to approximate the non-linear kernel. It is reported in the literature [87] that the linearly approximated kernel SVM is an order of magnitude fast than the traditional non-linear SVM.

6.4 Summary

This chapter proposed a 3D model categorization system with multi-class SVM for classification. The 3D models are represented using bag-of-words model as the shape descriptors for training and testing. The histogram intersection kernel and chi-square kernel are approximated with linear homogeneous maps to be incorporated with the SVM. The proposed categorization scheme is demonstrated on the NIST generic shape benchmark and the modified CAD dataset. The results suggest that using the kernelized multi-class SVM always perform better than the linear SVM.

Chapter 7 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

7.1 Conclusions

This thesis employed the bag-of-words approach for efficient retrieval and categorization of 3D models. Two feature extraction strategies which are simpler, more computation efficient, and more discriminative than the salient feature detections are proposed to incorporate with the bag-of-words representation for better 3D model retrieval performance. To make the current 3D model retrieval system scalable to large-scale datasets, a multi-class SVM 3D model categorization system was proposed for the one versus class comparison.

The contributions of this research are mainly in the following areas:

Firstly, a modified dense sampling and multi-scale dense (MSD) sampling strategy were proposed to extract local features from depth images of 3D models. Both of the modified dense sampling and MSD sampling extract features on uniformly distributed grids and the modified dense sampling extract features at a single scale while MSD sampling extract features at multiple scales. The proposed sampling strategies cover the full range of the depth images rendered from the 3D model compared to that salient

feature detection algorithm only describes sharp changes. The feature extraction speed of proposed sampling strategies is an order of magnitude faster than the original Scale Invariant Feature Transform (SIFT) detection weighted with a flat window. In combination with bag-of-words models, the proposed sampling strategies not only have shown superior performance over the original salient SIFT sampling, but also much faster to compute. The proposed modified dense sampling have showed to outperform the salient features for 3D model retrieval tasks on Purdue engineering shape benchmark, NIST generic shape benchmark and SHREC 2009 partial dataset.

Secondly, encouraged by the success of uniformly sampled features, two region-based features, namely Region-SURF (RSURF) and Histogram of Oriented Gradients (HOG) were proposed. The RSURF and HOG feature detection sample features at uniform grids at fixed scales and locations. Suitable region size, fine orientation and coarse spatial binning will together influence the descriptiveness and distinctness of the region-based feature detector. The RSURF and HOG features not only are faster and simpler to compute, they only take half or less storage than the SIFT feature description. With RSURF and HOG features as inputs for bag-of-words model representation, they have shown superior performance than salient SIFT and SURF features for 3D model retrieval tasks on the modified CAD dataset and NIST generic shape benchmark.

Thirdly, a learning-by-example scheme was devised to accommodate the needs for

large-scale retrieval and categorization tasks of 3D models. This scheme is achieved by multi-class Support Vector Machine (SVM) learning of classifiers for every two classes. Histogram intersection kernel and chi-square kernel, which are suitable for histogram-based descriptions, were approximated by linear homogeneous maps and incorporated with the SVM learning procedures. The 3D models are represented using bag-of-words approach as the shape descriptors for training and testing. The proposed categorization scheme was demonstrated on the NIST generic shape benchmark and the modified CAD dataset and showed that using the kernelized multi-class SVM always performs better than the linear SVM. The proposed 3D model categorization scheme has showed promising applications in recognition, categorization and management of large-scale 3D model datasets.

The proposed approaches in this thesis may have significant contributions in the following aspects. Firstly, the proposed densely sampled features have proved to be more efficient and representative for shape representation than the salient features. They are not only simpler and faster to compute, but also save considerable storage capacity than existing salient feature descriptions. This may lead to affordable 3D model description and storage with increasing amount of 3D models both on internet and in domain-specific databases. Secondly, the 3D model categorization system is proposed to accommodate the importance of managing 3D models in large-scale. It may bring the existing 3D model retrieval and categorization algorithms to practical applications.

7.2 Recommendations for Future Works

7.2.1 Extension for an Improved Bag-of-Words Representation

Regardless the effectiveness of bag-of-words representation, it may still suffer two main disadvantages. The potential solutions are proposed in this section to address these insufficiencies.

The first disadvantage is due to that bag-of-words represents a 3D model as a resemblance of order-less local features. The spatial information of the local features is totally discarded. Although there are some existing work that have attempted to incorporate the spatial information by representing the histogram for layered concentric spheres [90] or segmented parts [63], the improvement is difficult to observe. We proposed to endow the local features to incorporate the locality constraints to preserve the shape context information in a neighborhood system. An objective function needs to be defined to encode features in the sense of shape context. The potential influence of the proposed future work may bring the use of low-level features to the middle-level with shape semantics for efficient 3D models representation.

The second disadvantage is that the histogram-based representation only described the

occurrence of local features according to the visual words of the codebook learned. However, the cluster centers themselves also contain rich geometric information of local intensity gradient distributions. Although the K-means clustering can assign a local feature to nearest cluster center, it does not model the cluster center information. One potential approach is to employ the Gaussian Mixture Model (GMM) [91] to model the geometric information of the visual words.

Given the set of local features f_1, f_2, \dots, f_N , each of the Gaussian Mixture Model is estimated using Expectation Maximization (EM) algorithm to obtain the parameters $\theta_i = (\pi_i, \mu_i, \Sigma_i)$,

$$p(f|\mu_i, \Sigma_i) = \frac{1}{\sqrt{2\pi\det(\Sigma_i)}} e^{-0.5(f-\mu_i)^T \Sigma_i^{-1} (f-\mu_i)} \quad (7.1)$$

where π_i is the prior probability, $\mu_i \in D$ and $\Sigma_i \in D \times D$ are the mean and positive-definite covariance matrix of the Gaussian component. The encoding of each feature to the Gaussian model is according to the geometry of the Gaussian component, where,

$$h_{ik} = \frac{p(f_k|\mu_i, \Sigma_i)\pi_i}{\sum_{j=1}^K p(f_k|\mu_j, \Sigma_j)\pi_j}, k = 1, 2, \dots, K \quad (7.2)$$

so the Gaussian Mixture Model can be fully characterized by parameters of $(2D+1)*K$ dimension.

7.2.2 Extension for an Incremental Bag-of-Words Learning for Classification

Current bag-of-words approach is based on the fixed sets of features to generate the codebook. As abundant of the data available may help the system to generate a robust

and rich codebook for more accurate representation of the 3D models, the current learning for fixed categories of models often fail when met with a new class or a new instance which has not been learned previously. Therefore, there is a need to develop an incremental learning approach for data collecting and learning simultaneously. A parametric latent model [92] can be used to incrementally accumulate knowledge and examples of new instances just like the human learning process. Given a small set of seed models and categories, the algorithm seeks to learn a model which can best describe a category. Then newly collected models and categories will add on to the dataset to improve the model. With this iterative process, the final categorization classifiers can have robust performance for any new instances.

PUBLICATIONS

Wang Y., Lu, W.F., Fuh, J.Y.H., Wong, Y.S., Cheong, L.F., 3D CAD Model Classification Using Ordinal Measures, *International CAD Conference and Exhibition, Taipei, Taiwan, 2011*

Wang Y., Lu, W.F., Fuh, J.Y.H., Wong, Y.S., Bag-of-Features Sampling Techniques for 3D CAD Model Retrieval, in *Proceedings of ASME IDETC&CIE, Washington D.C., USA, 2011*

Wang Y., Lu, W.F., Fuh, J.Y.H., Sampling Strategies for 3D Partial Shape Matching and Retrieval Using Bag-of-Words Model, *Computer Aided Design and Applications*, Accepted.

REFERENCES

1. Van Krevelen, D. and R. Poelman, *A survey of augmented reality technologies, applications and limitations*.
2. Jayanti, S., et al., *Developing an Engineering Shape Benchmark for CAD Models*. Computer Aided Design, 2006. **38**(9): p. 939-p53.
3. Koller, D., B. Frischer, and G. Humphreys, *Research challenges for digital archives of 3D cultural heritage models*. J. Comput. Cult. Herit., 2010. **2**(3): p. 1-17.
4. Loncaric, S., *A survey of shape analysis techniques*. Pattern Recognition, 1998. **31**(8): p. 983-1001.
5. Bustos, B., et al., *Feature-Based Similarity Search in 3D Object Databases*. ACM Computing Surveys, 2005. **37**(4): p. 345-387.
6. Iyer, N., et al., *Three Dimensional Shape Searching: State-of-the-art Review and Future Trends*. Computer-Aided Design, 2005. **37**(5): p. 509-530.
7. Tangelder, J.W.H. and R.C. Veltkamp, *A survey of content based 3D shape retrieval methods*. Multimedia Tools Applications, 2008. **39**: p. 441-471.
8. Horn, B.K.P., *Extended Gaussian images*. Proceedings of the IEEE, 1984. **72**(12): p. 1671-1686.
9. Kang, S.B. and K. Ikeuchi, *The complex EGI: a new representation for 3-D pose determination*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1993. **15**(7): p. 707-721.
10. Ankerst, M., et al., *3D Shape Histograms for Similarity Search and Classification in Spatial Databases*, in *Proceedings of the 6th International Symposium on Advances in Spatial Databases*. 1999, Springer-Verlag. p. 207-226.
11. Ohbuchi, R., et al. *Shape-similarity search of three-dimensional models using parameterized statistics*. in *Computer Graphics and Applications, 2002. Proceedings. 10th Pacific Conference on*. 2002.
12. Osada, R., et al. *Matching 3D models with shape distributions*. in *Shape Modeling and Applications, SMI 2001 International Conference on*. 2001.
13. Yi, L., Z. Hongbin, and Q. Hong. *The Generalized Shape Distributions for Shape Matching and Analysis*. in *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*. 2006.
14. Ip, C.Y., et al., *Using shape distributions to compare solid models*, in *Proceedings of the seventh ACM symposium on Solid modeling and applications*. 2002, ACM: Saarbrücken, Germany. p. 273-280.
15. Vranic, D.V., D. Saupe, and J. Richter. *Tools for 3D-object retrieval: Karhunen-Loeve transform and spherical harmonics*. in *In: Proc. IEEE workshop on multimedia signal processing*. 2001.
16. Vranic, D.V. *An improvement of rotation invariant 3D-shape based on functions on concentric spheres*. in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*. 2003.
17. Vranic, D.V., *3D Model Retrieval*. 2004, University of Leipzig.
18. Kazhdan, M., T. Funkhouser, and S. Rusinkiewicz. *Rotation invariant spherical harmonic representation of 3D shape descriptors*. in *Symposium on geometry processing, SGP 2003*. 2003.
19. Novotni, M. and R. Klein, *Shape retrieval using 3D Zernike descriptors*. Computer-Aided Design, 2004. **36**(11): p. 1047-1062.

20. Papadakis, P., et al., *Efficient 3D Shape Matching and Retrieval using a Concrete Radialized Spherical Projection Representation*. Pattern Recognition, 2007. **40**: p. 2437-2452.
21. Daras, P., et al., *E. Multimedia*, IEEE Transactions on, 2006. **8**(1): p. 101-114.
22. Daras, P., et al. *3D model search and retrieval based on the spherical trace transform*. in *Multimedia Signal Processing, 2004 IEEE 6th Workshop on*. 2004.
23. Hilaga, M., et al. *Topology matching for fully automatic similarity estimation of 3D Shapes*. in *In: Proc. ACM SIGGRAPH*. 2001.
24. Tung, T. and F. Schmitt. *Augmented Reeb graphs for content-based retrieval of 3D mesh models*. in *Shape Modeling Applications, 2004. Proceedings*. 2004.
25. TUNG, T. and F. SCHMITT, *THE AUGMENTED MULTIREOLUTION REEB GRAPH APPROACH FOR CONTENT-BASED RETRIEVAL OF 3D SHAPES*. International Journal of Shape Modeling, 2005. **11**(01): p. 91-120.
26. Cyr, C.M. and B.B. Kimia, *A similarity-based aspect-graph approach to 3D object recognition*. International Journal of Computer Vision, 2004. **57**(1): p. 5-22.
27. Macrini, D., et al. *View-based 3-D object recognition using shock graphs*. in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. 2002.
28. Ding-Yun, C., et al. *On visual similarity based 3D model retrieval*. 2003. UK: Blackwell Publishers for Eurographics Assoc.
29. Chaouch, M. and A. Verroust-Blondet. *A New Descriptor for 2D Depth Image Indexing and 3D Model Retrieval*. in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. 2007.
30. Daras, P. and A. Axenopoulos, *A Compact Multi-view Descriptor for 3D Object Retrieval*, in *Proceedings of the 2009 Seventh International Workshop on Content-Based Multimedia Indexing*. 2009, IEEE Computer Society. p. 115-119.
31. Makadia, A. and K. Daniilidis, *Spherical Correlation of Visual Representations for 3D Model Retrieval*. International Journal of Computer Vision, 2010. **89**(2): p. 193-210.
32. Stavropoulos, G., et al., *3-D Model Search and Retrieval From Range Images Using Salient Features*. Multimedia, IEEE Transactions on, 2010. **12**(7): p. 692-704.
33. Papadakis, P., et al., *PANORAMA: A 3D Shape Descriptor Based on Panoramic Views for Unsupervised 3D Object Retrieval*. International Journal of Computer Vision, 2010. **89**(2): p. 177-192.
34. Pu, J. and K. Ramani, *On visual similarity based 2D drawing retrieval*. Computer Aided Design, 2006. **38**: p. 249-259.
35. Pu, J., K. Lou, and K. Ramani, *A 2D Sketch-Based User Interface for 3D CAD Model Retrieval*. Computer-Aided Design & Applications, 2005. **2**(6): p. 717-725.
36. Lodhi, H., et al. *Text classification using string kernels*. in *NIPS (In Advances in Neural Information Processing Systems)*. 2001.
37. Squire, D.M., et al., *Content-based query of image databases: inspirations from text retrieval*. Pattern Recognition Letters, 2000. **21**: p. 1193-1198.
38. AIM@SHAPE. [cited; Available from: <http://www.aimatshape.net/>].
39. Fergus, R., et al. *Learning object categories from Google's image search*. in *Proc. ICCV 05*. 2005.
40. Fei-Fei, L. and P. Perona. *A Bayesian Hierarchical Model for Learning Natural Scene Categories*. in *Computer Vision and Pattern Recognition*. in *In CVPR 2005*. 2005.
41. Qiu, G., *Indexing chromatic and achromatic patterns for content-based colour image retrieval*. Pattern Recognition, 2002. **35**(8): p. 1675-1686.

42. Ohbuchi, R., et al. *Salient Local Visual Features for Shape-Based 3D Model Retrieval*. in *IEEE Int. Conf. on Shape Modeling and Applications*. 2008. Stony Brook, USA.
43. Lowe, D.G., *Distinctive Image Features from Scale-invariant Key points*. *International Journal of Computer Vision*, 2004. **60**(2): p. 91-110.
44. Shilane, P., et al. *The Princeton Shape Benchmark*. in *Shape Modeling Applications, 2004. Proceedings*. 2004.
45. Zhang, J., et al., *Retrieving Articulated 3-D Models Using Medial Surfaces and Their Graph Spectra*, in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, A. Rangarajan, B. Vemuri, and A. Yuille, Editors. 2005, Springer Berlin Heidelberg. p. 285-300.
46. Chen, D.Y., et al., *On Visual Similarity Based 3D Model Retrieval*. *Computer Graphics Forum*, 2003. **22**(3): p. 223-232.
47. Furuya, T. and R. Ohbuchi. *Dense Sampling and Fast Encoding for 3D Model Retrieval Using Bag-of-Visual Features*. in *ACM International Conference on Image and Video Retrieval*. 2009. Santorini, Greece.
48. Ansary, T.F., M. Daoudi, and J.-P. Vandeborre, *A Bayesian 3-D Search Engine Using Adaptive Views Clustering*. *Multimedia, IEEE Transactions on*, 2007. **9**(1): p. 78-88.
49. Ohbuchi, R., et al. *Squeezing Bag-of-Features for Scalable and Semantic 3D Model Retrieval*. in *Proc. 8th International Workshop on Context-Based Multimedia Indexing*. 2010. Grenoble, France.
50. Ohbuchi, R. and T. Furuya. *Distance Metric Learning and Feature Combination for Shape-Based 3D Model Retrieval*. in *Proceedings of the ACM workshop on 3D object retrieval*. 2010. Firenze, Italy.
51. Lian, Z., A. Godil, and X. Sun. *Visual Similarity based 3D Shape Retrieval Using Bag-of-Features*. in *IEEE Int. Con. on Shape Modeling and Applications*. 2010. Aix-en-Provence, France.
52. Lian, Z., et al. *Non-rigid 3D shape retrieval using Multidimensional Scaling and Bag-of-Features*. in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. 2010.
53. Lian, Z., et al., *CM-BOF: visual similarity-based 3D shape retrieval using Clock Matching and Bag-of-Features*. *Machine Vision and Applications*, 2013: p. 1-20.
54. Johnson, A. and M. Hebert, *Using spin-images for efficient multiple model recognition in cluttered 3-D scenes*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1999. **21**(5): p. 433-49.
55. Li, X. and A. Godil. *Investigating the Bag-of-Words Method for 3D Shape Retrieval*. in *EURASIP Journal on Advances in Signal Processing*. 2010. Aalborg, Denmark: Hindawi Publishing Corporation.
56. Fehr, J. and H. Burkhardt. *Harmonic shape histograms for 3d shape classification and retrieval*. in *IAPR conference on machine vision applications*. 2007.
57. Tabia, H., et al., *Deformable shape retrieval using bag-of-feature techniques*. 2011: p. 78640P-78640P.
58. Ohkita, Y., et al. *Non-rigid 3D Model Retrieval Using Set of Local Statistical Features*. in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*. 2012.
59. Kawamura, S., et al., *Local geometrical feature with spatial context for shape-based 3D model retrieval*, in *Proceedings of the 5th Eurographics conference on 3D Object Retrieval*. 2012, Eurographics Association: Cagliari, Italy. p. 55-58.
60. Tang, S. and A. Godil, *An evaluation of local shape descriptors for 3D shape retrieval*. 2012: p.

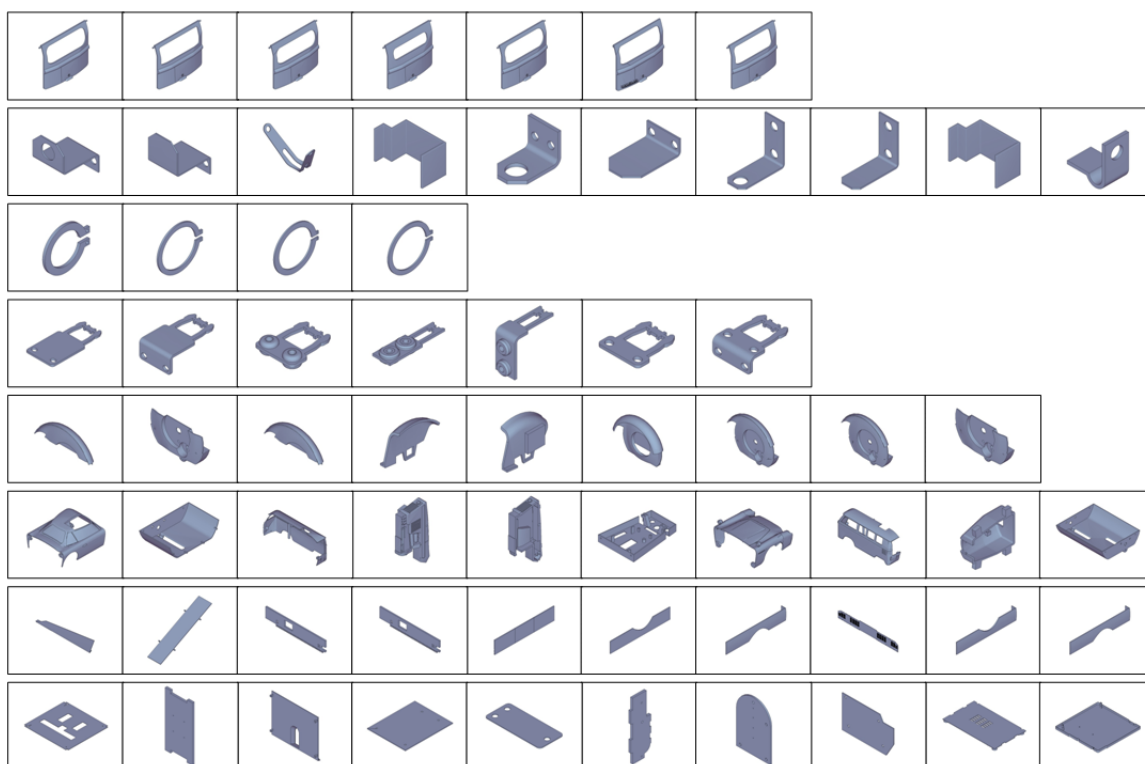
- 82900N-82900N.
61. Lian, Z., et al., *SHREC'11 track: shape retrieval on non-rigid 3D watertight meshes*, in *Proceedings of the 4th Eurographics conference on 3D Object Retrieval*. 2011, Eurographics Association: Llandudno, UK. p. 79-88.
 62. Heider, P., et al., *Local shape descriptors, a survey and evaluation*, in *Proceedings of the 4th Eurographics conference on 3D Object Retrieval*. 2011, Eurographics Association: Llandudno, UK. p. 49-56.
 63. Toldo, R., U. Castellani, and A. Fusiello. *Visual Vocabulary Signature for 3D Object Retrieval and Partial Matching*. in *Eurographics Workshop on 3D Object Retrieval*. 2009.
 64. Veltkamp, R.C. and F.B. ter Haar, *Shrec 2007 3d retrieval contest.*, in *Technical Report UU-CS-2007-015* 2007, Department of Information and Computing Sciences.
 65. Bronstein, A.M., et al., *Shape google: Geometric words and expressions for invariant shape retrieval*. *ACM Trans. Graph.*, 2011. **30**(1): p. 1-20.
 66. Lavoué, G., *Combination of bag-of-words descriptors for robust partial shape retrieval*. *The Visual Computer*, 2012. **28**(9): p. 931-942.
 67. Toldo, R., U. Castellani, and A. Fusiello, *A Bag of Words Approach for 3D Object Categorization*, in *Computer Vision/Computer Graphics Collaboration Techniques*, A. Gagalowicz and W. Philips, Editors. 2009, Springer Berlin Heidelberg. p. 116-127.
 68. Li, J.-B., et al., *3D model classification based on nonparametric discriminant analysis with kernels*. *Neural Computing and Applications*, 2013. **22**(3-4): p. 771-781.
 69. Tabia, H., et al., *A parts-based approach for automatic 3D shape categorization using belief functions*. *ACM Trans. Intell. Syst. Technol.*, 2013. **4**(2): p. 1-16.
 70. Jolliffe, I.T., *Principal component analysis*. 1986: Springer-Verlag.
 71. Duda, R., P. Hart, and D. Stork.
 72. Belongie, S., J. Malik, and J. Puzich. *Matching Shapes*. in *ICCV*. 2001.
 73. Daras, P. and A. Axenopoulos, *A 3D Shape Retrieval Framework Supporting Multimodal Queries*. *International Journal of Computer Vision*, 2010. **89**(2-3): p. 229-247.
 74. Patil, S. and B. Ravi. *Voxel-based Representation, Display and Thickness Analysis of Intricate Shapes*. in *Int. Conf. on Computer Aided Design and Computer Graphics*. 2005.
 75. Aitkenhead, A.H. *Polygon Mesh Voxelisation*. 2010 [cited; Available from: <http://www.mathworks.com/matlabcentral/fileexchange/27390-mesh-voxelisation>].
 76. Swain, M.J. and D.H. Ballard, *Color Indexing*. *International Journal of Computer Vision*, 1991. **7**(1): p. 11-32.
 77. Fang, R., et al. *A new shape benchmark for 3D object retrieval*. in *Proceeding ISVC '08 Proceedings of the 4th International Symposium on Advances in Visual Computing 2008*.
 78. *SHREC 2009 - Shape Retrieval Contest of Partial 3D Models*. [cited; Available from: <http://www.itl.nist.gov/iad/vug/sharp/benchmark/shrecPartial/>].
 79. Bosch, A., A. Zisserman, and X. Muoz. *Image Classification using Random Forests and Ferns*. in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. 2007.
 80. Vedaldi, A. and B. Fulkerson. *VLFleat: An Open and Portable Library of Computer Vision Algorithms*. 2008 [cited; Available from: <http://www.vlfleat.org/>].
 81. Elkan, C. *Using the Triangle Inequality to Accelerate k-Means*. in *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*. 2003. Washington, D.C.

82. Arthur, D. and S. Vassilvitskii, *k-means++: the advantages of careful seeding*, in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 2007, Society for Industrial and Applied Mathematics: New Orleans, Louisiana. p. 1027-1035.
83. Bay, H., et al., *Speeded-Up Robust Features (SURF)*. *Computer Vision and Image Understanding*, 2008. **110**(3): p. 346-359.
84. Viola, P. and M. Jones. *Rapid object detection using a boosted cascade of simple features*. in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. 2001.
85. Dalal, N. and B. Triggs. *Histograms of oriented gradients for human detection*. in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. 2005.
86. Haykin, S., *Neural Networks: A comprehensive foundation*. 2nd Edition ed. 1999: Prentice-Hall.
87. Vedaldi, A. and A. Zisserman, *Efficient Additive Kernels via Explicit Feature Maps*. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2012. **34**(3): p. 480-492.
88. Shalev-Shwartz, S., Y. Singer, and N. Srebro, *Pegasos: Primal Estimated sub-GrAdient SOLver for SVM*, in *Proceedings of the 24th international conference on Machine learning*. 2007, ACM: Corvallis, Oregon. p. 807-814.
89. Vedaldi, A. and B. Fulkerson, *Vlfeat: an open and portable library of computer vision algorithms*, in *Proceedings of the international conference on Multimedia*. 2010, ACM: Firenze, Italy. p. 1469-1472.
90. Li, X., A. Godil, and A. Wagan. *Spatially Enhanced Bags of Words for 3D Shape Retrieval*. in *Proceedings of the 4th International Symposium on Advances in Visual Computing*. 2008. Las Vegas, NV: Springer-Verlag.
91. Chatfield, K., et al. *The devil is in the details: an evaluation of recent feature encoding methods*. in *In BMVC*. 2011.
92. Li, L.-J. and L. Fei-Fei, *OPTIMOL: Automatic Online Picture Collection via Incremental Model Learning*. *International Journal of Computer Vision*, 2010. **88**(2): p. 147-168.

Appendix A Lists of the Modified CAD Dataset

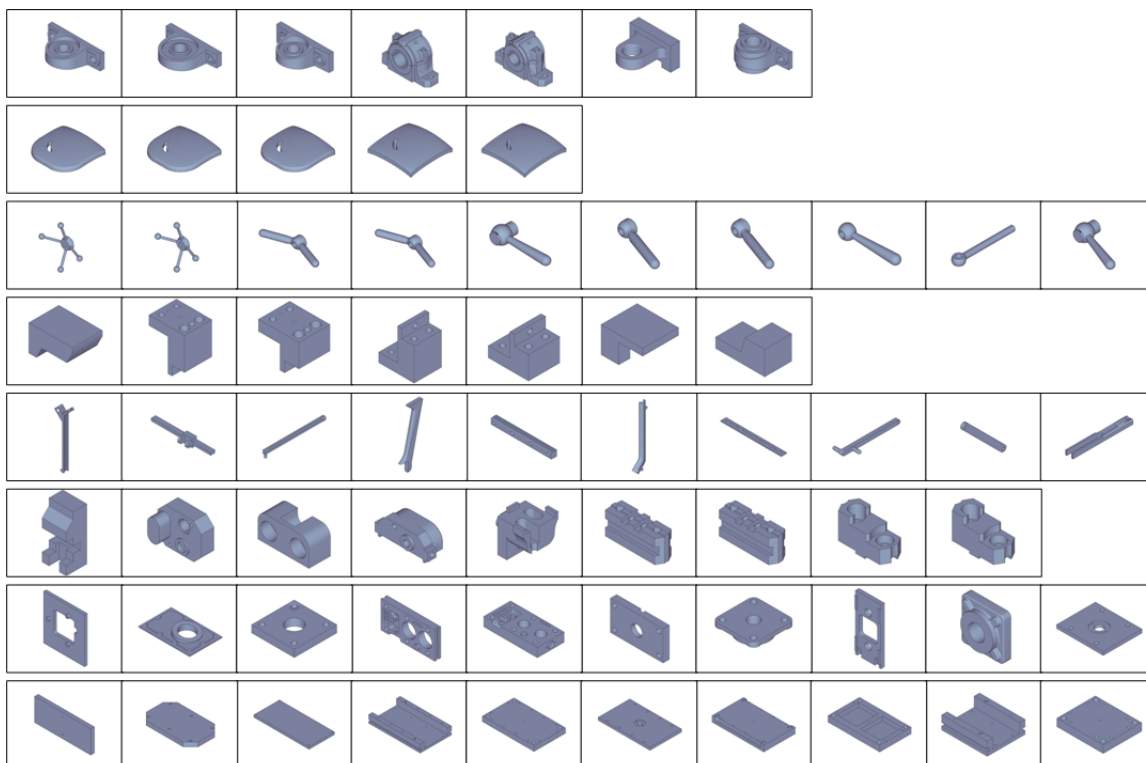
Part I: Flat-thin wall components: 8 classes, total 67 models.

Classes 1-8 are: 1-Back Doors (7); 2-Bracket Like Parts (10); 3-Clips (4); 4-Contact Switches (8); 5-Curved Housings (9); 6-Rectangular Housings (10); 7-Slender Thin Plates (10); 8-Thin Plates (10).

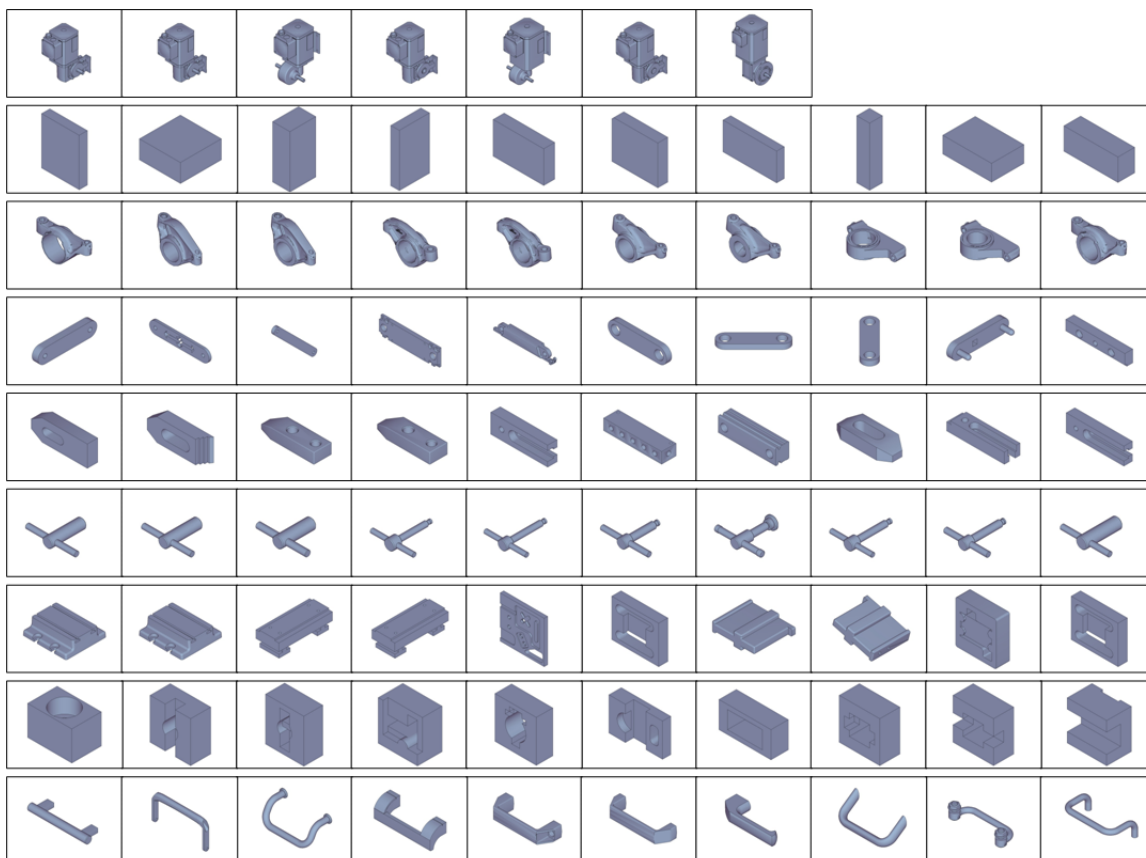


Part II: Rectangular-cubic Prism: Total 17 classes, 165 models.

Classes 9-16 are: 9-Bearing Blocks (7); 10-Contoured Surfaces (5); 11-Handles (10); 12-Blocks (7); 13-Long Machined Elements (10); 14-Machined Blocks (9); 15-Machined Plate with Significant Holes (10); 16-Machined Plate with Small Holes (10);

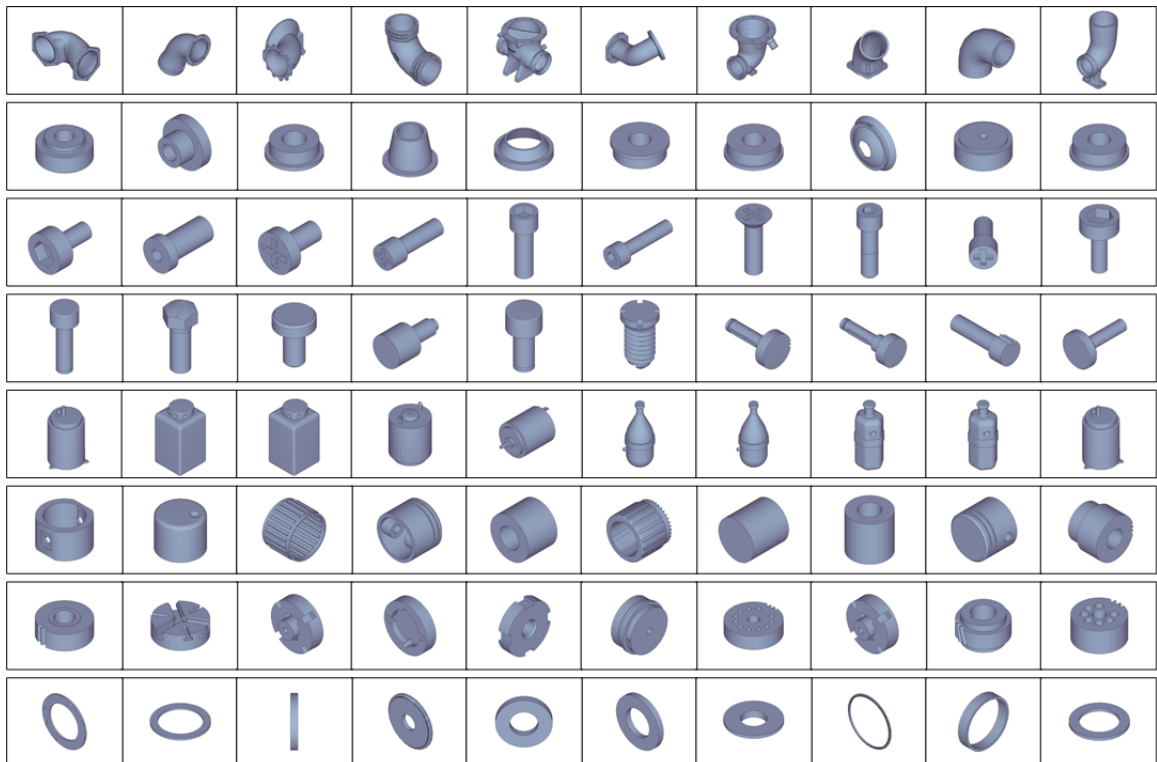


Classes 17-25 are: 17-Motor Bodies (7); 18-Prismatic Blocks (10); 19-Rocker Arms (10); 20-Slender Links (10); 21-Small Machined Blocks (10); 22-T-shaped Parts (10); 23-Thick Plates (10); 24-Thick Slotted Plates (10); 25-U-Shaped Parts (10).

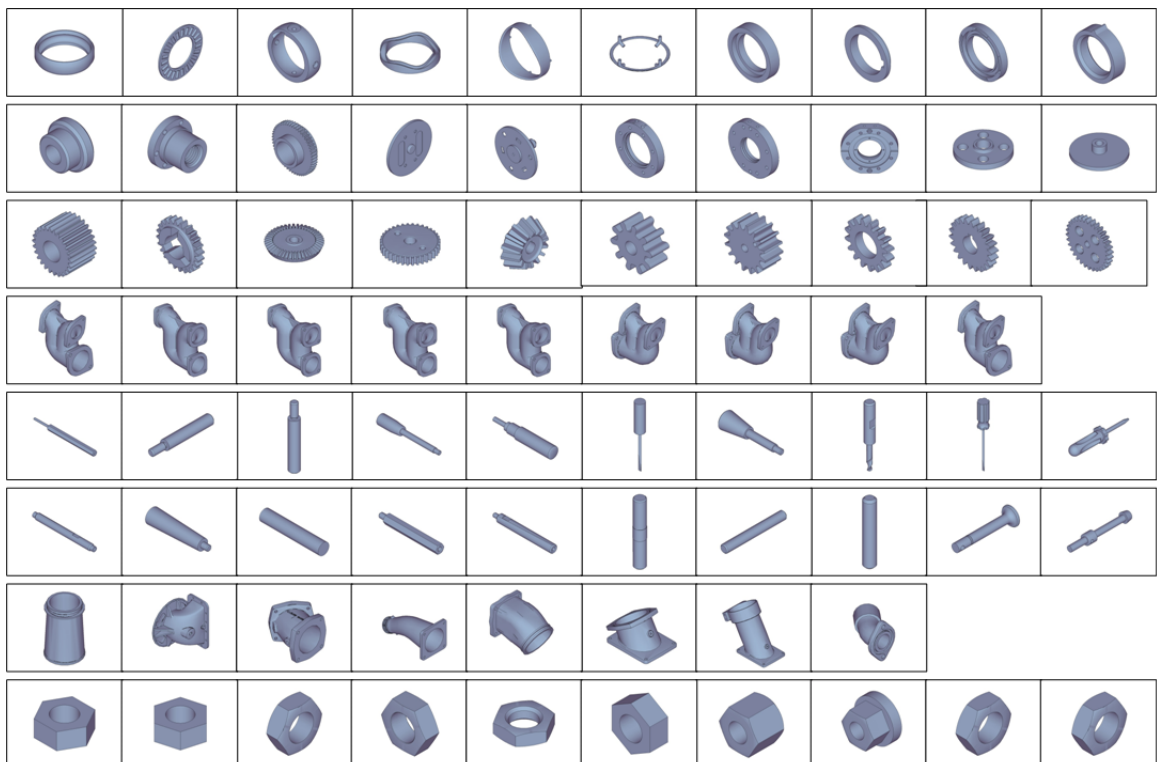


Part III: Solids of Revolution: Total 22 classes, 215 models.

Class 26-33 are: 26-90 Degree Elbows (10); 27-Bearing Like Parts (10); 28-Bolt with Closed Shape End (10); 29-Bolt with Open or No Shape End (10); 30-Container Like Parts (10); 31-Cylindrical-like Parts with Large H/R ratio (10); 32- Cylindrical-like Parts with Small H/R ratio (10); 33-Simple Discs (10).



Class 34-41 are: 34- Discs Others (10); 35-Flange Like Parts (10); 36-Gear Like Parts (10); 37-Intersecting Pipes (9); 38-Long Pins Screw Drives (10); 39-Long Pins Others (10); 40-Non-90Degree Elbows (8); 41-Nuts (10).



Class 42-47 are: 42-Oil Pans (8); 43-Posts (10); 44-Pulley Like Parts (10); 45-Round Change At End (7); 46-Simple Pipes (10); 47-Spoked Wheels (10).

