3D Object Retrieval and Segmentation: Various Approaches Including 2D Poisson Histograms and 3D Electrical Charge Distributions.

By

Fattah Alizadeh, M.Sc.

A dissertation submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy (PhD)

to the

Dublin City University

Faculty of Engineering and Computing, School of Computing

Supervisor: Dr. Alistair Sutherland

August 2014

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: F. Alizadeh

(Candidate) ID No.: 59111925

Date: 11/08/2014

Table of Contents

Declaration i

Abstract xiii

Acknowledgement xiv

Dedication xvi

Chapter 1: Introduction 1

1.1. Introduction 1

1.2. 3D Model 3

1.3. 3D Model Creation Techniques 4

1.4. 3D Model Representation Format 5

1.5. Thesis Contributions 7

1.6. Thesis Outline 7

Chapter 2: Literature Review 9

2.1. Introduction 9

2.2. State-of-the-art in 3D Object Retrieval (3DOR) 9

2.2.1. 2D-Based Shape Descriptors 10

2.2.1.1. View-Based Descriptors (Spatial Domain) 11

2.2.1.2. Sketch-Based Descriptor 16

2.2.1.3. 2D Transform-Based Descriptor 20

2.2.2. 3D-Based Shape Descriptors 22

2.2.2.1. Histogram (statistical)-based approaches 22

2.2.2.2. Graph-based approaches 27
2.2.2.3. 3D Transform-based approaches 30
2.2.2.4. Heat Kernel-based approaches 32
2.2.3. Discussion on the Shape Descriptors 36
2.3. State-of-the-art in 3D model segmentation 38
2.3.1. Grouping Segmentation Algorithms 38
2.3.2. Individual Segmentation 40
2.3.2.1. Region Growing 40
2.3.2.2. Watershed 43
2.3.2.3. Interactive 46
2.3.2.3.1. Sketch-based interactive segmentations 46
2.3.2.3.2. Point-based interactive segmentations 48
2.3.2.4. Skeleton-based approaches 49
2.3.2.4.1. Skeleton graph-based approaches 49
2.3.2.4.2. Reeb Graph-based approaches 51
2.3.2.5. Feature Point-based approaches 53
2.3.2.6. Spectral approaches 55
2.3.2.7. Learning based approaches 57
2.3.2.8. Volume-based Approaches 58
2.3.2.9. Primitive Fitting-Based Approaches 59
2.3.2.10. Other approaches 61
2.3.2.10.1 Model-based approaches 61
2.3.2.10.2 Fuzzy-based approaches 61
2.3.2.10.3 Randomized Cut 62
2.3.3. Consistent Segmentation 63
2.3.4. Discussion on the Segmentation Approaches 64

Chapter 3: Proposed 3D Model Descriptors 67
3.1. Introduction 67
3.2. Motivation 67
3.3. Scientific Challenges 69
3.4. Applications 72
3.5. Proposed Approaches 73
3.5.1. 2D Poisson-Based Shape Descriptor 73
3.5.2. Background 74

3.5.2.1. Silhouette Poisson-Histogram Descriptor (SilPH) 75
3.5.2.2. Discussion of the SilPH Descriptor 78
3.5.3. 3D Charge Density-Based Shape Descriptor 78
3.5.3.1. Background 79
3.5.3.2. Discussion on Electrical Charge Distribution as a Descriptor 81
3.5.3.3. Bag-of-Feature Charge Density Descriptor (BoF-CDD) 82
3.5.3.4. Concentric Sphere-Electrical Charge Descriptor (CS-ECD) 83
3.5.3.4.1. First Phase of DP_CS-ECD construction 84
3.5.3.4.2. Second Phase of CS-ECD construction (DP_CS_ECD) 86
3.6. Chapter Summary 88

Chapter 4: Experimental Results of the Proposed Shape Descriptors 89
4.1. Introduction 89
4.2. Description of Datasets 89
4.3. Evaluation Metrics 91
4.4. Parameter Analysis 93
4.4.1. Parameter of the SilPH Descriptor 93
4.4.2. Parameters of the BOF-CDD Descriptor 94
4.4.2.1. Size of the Dictionary 94
4.4.2.2. Feature Point Selection Criteria 94
4.4.2.3. Feature Point Number 95
4.4.3. Parameters of the DP-CS-ECD Descriptor 95
4.5. Results on the McGill Dataset 96
4.6. Results on SHREC'11 Dataset 98
4.7. Partial Matching Ability on SHREC'07 Dataset 101
4.7.1. Related works 104
4.7.2. Partial Matching Results 105
4.8. Robustness 107
4.8.1. Robustness to Noise 108
4.8.2. Robustness to deformation 109
4.8.3. Sensitivity to Mesh Tessellation 111
4.9. Complexity 113
4.10. Conclusion 114

Chapter 5: 3D Model Segmentation 116

5.1. Introduction 116
5.2. Scientific Challenges 118
5.3. Applications 120
5.4. Related Work 122
5.5. Background 123
5.6. Proposed Approach 124
5.6.1. Pre-processing Phase 125
5.6.1.1. Simplification 125
5.6.1.2. Canonical Form Representation 125
5.6.1.3. Simulation of Charge Density distribution 126
5.6.2. Segmentation Phase 126
5.6.2.1. Terminology 126
5.6.2.2. Protrusion Parts (Initial Segments) Extraction 128
5.6.2.3. Boundary Faces Detection 129
5.6.2.3.1. Merging Strong Boundary Faces 130
5.6.2.3.2. Weak boundary enhancement 131
5.6.2.4. Convert to the Original models 134
5.6.2.5. Region Growing 134
5.6.3. Post-Processing Phase 134
5.6.3.1. Re-segmenting the mixed cores 134
5.6.3.2. Boundary Cut Refinement 135
5.7. Chapter Summary 138

Chapter 6: Experimental Results of the Proposed Segmentation Approach 140
6.1. Introduction 140
6.2. Description of Datasets 141
6.3. Evaluation Metrics 142
6.4. Parameter Analysis 147
6.5. Results on the SHREC’07 Dataset 149
6.5.1. Comparison to the Well-known Approaches 149
6.6. Results on SHREC’12 Dataset 152
6.6.1. Comparison to the State-of-the-Art Approaches 153
6.6.2. Cut-Refinement Effect 154
6.7. Robustness 155
6.7.1. Robustness to Noise 155

6.7.2. Robustness to deformation 156
6.7.3. Sensitivity to Mesh Tessellation 157
6.8. Comparison With the Other Charge Density-Based Approach 158
6.9. Limitations 160
6.10. Complexity 160
6.11. Conclusion 162

Chapter 7: Conclusions and Future Work 163
7.1. Introduction 163
7.2. Summary of contributions 163
7.2.1. SilPH Descriptor (Chapter 3) 164
7.2.2. BoF-CDD Descriptor (Chapter 3) 164
7.2.3. DP-CS-ECD Descriptor (Chapter 3) 164
7.2.4. Charge Density-Based Segmentation (Chapter 5) 165
7.2.5. Surveys of Available Descriptors and Segmentation Approaches (Chapter 2) 165
7.3. Directions for Future Research 165
7.3.1. Toward Partial Matching 165
7.3.2. Extending the Retrieval Algorithm to Support Relevance Feedback 166
7.3.3. Extending Segmentation Approach 166

Bibliography 167

Appendix A: Object File Format (OFF) .... A -1
Appendix B: Query Set for Partial Matching .... B -1
Appendix C: Segmentation Ground Truth .... C -1

List of Figures

Figure 1.1, Sample 3D models from different applications.. 4
Figure 1.2, sample 3D model acquisition devices. Left: laser scanner-based acquisition of Michelangelo's David model from Stanford University, Right: A CMM device for generating a Cat model. 4
Figure 1.3, 3D face models reconstructed from a set of 2D images............................. 5
Figure 1.4, Different representation of a sample Bunny model. (a) Point clouds, (b), (c) polygonal meshes, (d) set of parametric surface and (e) voxel................ 6

Figure 2.1, Taxonomy of 3D object retrieval approaches..10
Figure 2.2, a 3D model of an airplane (leftmost) and corresponding three silhouettes (first row) and depth-buffer images (second row) captured from three directions...11
Figure 2.3, The demonstration of Spin Image 13
Figure 2.4, A 3D cup models and the corresponding example SSCD images proposed by Gao et al. [30] 14
Figure 2.5, The illustration of the modified BOF method using SSCD proposed by Li et al. [31] 15
Figure 2.6, a sample interface of a retrieval system using the hand-drawn sketch as the query. 16
Figure 2.7, A typical sketch-based 3DOR system. 17
Figure 2.8, comparison of contour (left) and suggestive contour (right). 18
Figure 2.9, sample panoramic view of a car model 20
Figure 2.10, Computing the Light-Field descriptor for a chair model, by Chen et al.[27]. 21
Figure 2.11, 5 simple functions based on geometric measurements by Osada et al. [58]...23
Figure 2.12, Three decomposition techniques..24
Figure 2.13, two models with different volume and VEGI but the same EGI...........25
Figure 2.14, Examples of shot rays in a Cone-shape region. 26
Figure 2.15, A sample 2D histogram CDF of different poses of (a) human model and (b) dog model...27
Figure 2.16, a model and the corresponding multi-resolution Reeb graph [33]. ...28

Figure 2.17, The DAG graph of a sample dog model extracted by Sundar et al. [80]. 30
Figure 2.18, Computing the spherical harmonics shape descriptor by Kazhdan et al. 31
Figure 2.19, the scaled HKS at four sample points of a dragon model proposed by Sun et al. [92] 33
Figure 2.20, the SI-BOF histogram samples proposed by Bronstein et al. [95] 35
Figure 2.21, similar computed closed curves of [97] for different poses of human model, even models having missing parts. 35
Figure 2.22, Taxonomy of 3D model segmentation approaches. 39
Figure 2.23, sample steps of region growing scheme applied on a hand model. 40
Figure 2.24, two cutting contours generated by random walk algorithm for the same model [108]. 42
Figure 2.25, the watershed and basins in a topological surface. 43
Figure 2.26, segmentation result of Page et al. [120] on a sample mug model. (a) is the initial threshold of curvature, (b) shows the result after applying the morphological operator and (c) is the final segmentation result. 45
Figure 2.27, the segmentation steps of Benjamin et al. [122]. (b) is the heat kernel of the model using LBO, (c) and (d) are accumulator and dissipater regions identification, respectively. 45
Figure 2.28 (a-e), samples for different interactive segmentation algorithms. 46
Figure 2.29, segmentation steps of segmentation approach of [142]. (a) is the segmented medial surface, (b) is the readjusted segments and (c) is the resulted segments. 49
Figure 2.30, three smoothing iterations of skeleton extraction along with segmentation result of Au et al. [144] 50
Figure 2.31, a sample model in left and its related Reeb graph in right. 51
Figure 2.32, the segmentation results of Tierny et al. [147]; (a) input model, (b) extracted graph, (c) initial segmentation, (d) fine segmentation and (e) Coarse segmentation. 52
Figure 2.33, segmentation result of [149], (a) is the initial constructed Reeb graph and associated segmentation result and (b) shows the result after graph simplification. 53
Figure 2.34, the segmentation steps of the proposed algorithm of Katz et al. [151] 53
Figure 2.35: the pipeline used in segmentation work of Liu and Zheng [157] 56
Figure 2.36, Volume-based segmentation approach using SDF. 59
Figure 2.37, some examples of fitting primitive approach proposed by Attene et al. [169]. 60
Figure 2.38, the segmentation steps for Fuzzy-based approach proposed by Katz et al. [172] 62
Figure 2.39, segmentation steps of the randomized cut algorithm of [138], (a) shows sample random segmentations, (c) is the segmentation result for the Bunny model generated by the related partitioning function (c). 63
Figure 2.40, co-segmentation approach proposed by Meng et al. [173]. 64

Figure 3.1, A typical 3D model retrieval system 70

Figure 3.2, Centaur; A mythological Greek creature; half-horse and half-human. 71
Figure 3.3, the proposed model retrieval Framework; left part shows the Offline phase and right part is the Online phase....76
Figure 3.4, Some sample silhouettes and their solution to the Poisson equation. 77
Figure 3.5, the configuration for a charge point q placed at the point r' which is observed from the point r....80
Figure 3.6. Example coloured models from the McGill dataset; the redder parts specify the denser faces....81
Figure 3.7, A typical BoF framework in a 3DOR system....83
Figure 3.8, Various poses of a human model enclosed in 5 concentric spheres along with the associated histogram descriptors....84
Figure 3.9, Four different poses of an ant model and their corresponding canonical forms....85
Figure 3.10, Various poses of a human model represented in the canonical form enclosed in 5 concentric spheres along with the associated histogram descriptors....86
Figure 3.11, extracted dense patches on the surface of three different models. ...87

Figure 4.1, sample models from the McGill dataset....90
Figure 4.2, 30 classes of the SHREC'11 non-rigid watertight dataset along with 5 samples for each class....91
Figure 4.3, The relation between the DCG measure and number of feature points in BOF-CDD (Dic_Size=20)....95
Figure 4.4, the P-R curve for our and six other well-known approaches....97
Figure 4.5, Mean Curvature descriptor, left: normal model, right: noisy model..98
Figure 4.6, Precision-Recall curve of the proposed descriptors and the approaches participated in SHREC'11 contest....99
Figure 4.7, Precision-Recall curve for six sample classes of SHREC'11 dataset using DP-CS-ECD descriptor....102
Figure 4.8, the DP-CS-ECD descriptors for partial similar models (Q=100 and Ns=5)....102
Figure 4.9, First row: the silhouette extracted from identical viewpoints for the partial-similar models. Second row: solution of the views to the Poisson equation and Third row: SilPH Descriptor of the views....103
Figure 4.10, RPU approach of Tierny et al. [209]. The First row shows the segmentation process of a hand model into its Reeb charts and the Second row displays the Reeb chart unfolding process of its thumb....105
Figure 4.11, Three sample queries and associated ten retrieved models....106
Figure 4.12, NDCG plot for BOF-CDD and five other approaches....107
Figure 4.13, The SilPH histogram for identical view of the original and noisy versions of a Teddy model....108
Figure 4.14, Distribution of electrical charge on the surface of a model with different levels of noise....109
Figure 4.15, the SilPH histogram of human model in deferent poses....110
Figure 4.16, BOF-CDD for various poses of a human model....111
Figure 4.17, The SilPH descriptors along with the solution of their similar views of a human model in various resolutions to the Poisson equation....112

Figure 4.18, distribution of electrical charge on the surface of a human model in various resolutions. 113

Figure 5.1, Detailed steps of our segmentation framework. 118
Figure 5.2, sample 3D models (a) models having complex structure, (b) Hierarchically segmented, (c) noisy models and (d) Human model in different poses. 119
Figure 5.3, construction of a new chair model from other chairs' components. (Figure is taken from [130]) 120
Figure 5.4, Partial matching samples from [102]; the highlighted part of the models in the left-hand column is the query and the models in the other columns are the retrieved results. 121
Figure 5.5, Different poses of a horse model and consistent skeleton curves extracted by the approach of Shapira et al. [129] 122
Figure 5.6, The relationship between ρ, l and γ. 124
Figure 5.7, Coloured models based on the distribution of charge density on the surface; the redder parts specify the denser faces. 124
Figure 5.8, An Armadillo model at different levels of resolution based on the number of faces. 125
Figure 5.9, top row: Human model in various poses, middle row: the related canonical form representation and Last row: simulated charge density distribution over the models surfaces 126
Figure 5.10, (a) three sample faces (red colour) and the associated ring neighbourhood (blue faces), (b) three sample regions (blue region) and their associated ring neighbourhood (green region). 127
Figure 5.11. Examples of protruding parts along with the corresponding representative faces indicated by red points. 129
Figure 5.12, Detected Strong Boundary Face (Blue colour) and Weak Boundary Faces (Red) colour on ANT model. 130
Figure 5.13, The strong boundary faces located in boundaries of protruded parts of Octopus and Human models. 130
Figure 5.14, Strong boundary faces enhancement. 131
Figure 5.15, Weak boundary faces detected on a hand model with spurious boundaries. 131
Figure 5.16, Different annular regions in the middle finger of a Hand model (segment number=3) and the charge difference in successive regions. 133
Figure 5.17, Steps for segmenting models having mixed core: First and second rows show the process for an Ant and Spectacle models, respectively. (The colours are arbitrary) 135
Figure 5.18, inaccurate detected boundaries for three sample models identified by a rectangle. 136
Figure 5.19, (a) raw segmentation of Bird model and (b) three disjoint regions extracted for refining boundary of the left wing. 137
Figure 5.20, Detected boundaries (a) before and (b) after applying Cut Refinement. 137
Figure 5.21, Cut-refinement process, S and T denote Source and Target nodes of the extracted flow graph. 139

Figure 6.1, Sample models of watertight track of SHREC'07 dataset 141
Figure 6.2, 28 models of the segmentation track of SHREC'12. 142
Figure 6.3, Cow and Dinopet small part extraction by selecting small values for parameter $\Psi = 0.125$. 148
Figure 6.4. The relationship between parameter T and 3DPRI measure............... 148
Figure 6.5, segmentation results for sample models from the SHREC'07 dataset.. 149
Figure 6.6, Evaluation metrics for the proposed approach and some state-of-the-art methods applied to the SHREC'07 dataset. 151
Figure 6.7, segmentation results for the models of the SHREC'12 dataset. 153
Figure 6.8, 3D-NPRI indices of the 28 models of SHREC'12 dataset sorted in increasing order before and after applying Cut-Refinement process. 155
Figure 6.9. Segmentation of (leftmost) an original human model and (other three models) noisy human model at different noise levels. 156
Figure 6.10, 3D-NPRI indices of the 28 models of SHREC'12 dataset sorted in increasing order for different levels of noise 156
Figure 6.11, Similar segments extracted for a human model in different poses. 157
Figure 6.12, Similar extracted segments for a human model at various levels of tessellation. 158
Figure 6.13, First row: examples of detected boundaries (green parts) by Wu and Levine [171] for concave faces located in small rectangles. Second row: Segmentation results of our approach for the same models. 159
Figure 6.14, 3D-NPRI indices of the 28 models of SHREC'12 dataset sorted in increasing order for Our Method and a similar method of Wu and Levine 159
Figure 6.15, Inappropriately detected segment in a Mechanical model. 160

Figure A.1, A sample triangular-based pyramid model along with its .OFF file ... A-1

Figure B.1, partial matching query set of SHREC'07 contest....................................... B-1

Figure C.1, (a): The ground truth segmentation for sample models of SHREC'07 dataset, (b): Segmentation results produced by our approach............................ C-1
Figure C.2, (a): The ground truth segmentation for sample models of SHREC'12 dataset, (b): Segmentation results produced by our approach............................ C-2

List of Tables

Table 2.1, Taxonomy of shape descriptors. 37
Table 2.2, Taxonomy of segmentation approaches. 66

Table 4.1, The evaluation metrics for different bin numbers of the SilPH Descriptor. 94
Table 4.2, Evaluation factors for different size of dictionary in BOF-CDD (With 1000 random points). 94
Table 4.3, Evaluation metrics for different feature points selection criteria in BOF-CDD (Dic_Size=20). 95
Table 4.4, Evaluation metrics for different numbers of concentric spheres in the DP-CS-ECD 96
Table 4.5, Evaluation metrics of the proposed descriptors and the approaches participated in SHREC'11 contest. 100
Table 4.6, Complexity of the proposed descriptors 114

Table 6.1, Per-class Rand Index (%) for segmented models on SHREC'07 dataset 152
Table 6.2, 3DNPRI results (%) for segmented models on SHREC'12, Segmentation Track. 154
Table 6.3, the average required time and Rand Index metric for various approaches on the SHREC'07 dataset 162

Table B.1, The category ground-truth for each query model. B-2

Thesis Title: 3D Object Retrieval and Segmentation: Various Approaches Including 2D Poisson Histograms and 3D Electrical Charge Distributions.

Author: Fattah Alizadeh

Abstract

Nowadays 3D models play an important role in many applications: *viz.* games, cultural heritage, medical imaging etc. Due to the fast growth in the number of available 3D models, understanding, searching and retrieving such models have become interesting fields within computer vision.

In order to search and retrieve 3D models, we present two different approaches: one is based on solving the Poisson Equation over 2D silhouettes of the models. This method uses 60 different silhouettes, which are automatically extracted from different view-angles. Solving the Poisson equation for each silhouette assigns a number to each pixel as its signature. Accumulating these signatures generates a final histogram-based descriptor for each silhouette, which we call a SilPH (Silhouette Poisson Histogram).

For the second approach, we propose two new robust shape descriptors based on the distribution of charge density on the surface of a 3D model. The Finite Element Method is used to calculate the charge density on each triangular face of each model as a local feature. Then we utilize the Bag-of-Features and concentric sphere frameworks to perform global matching using these local features.

In addition to examining the retrieval accuracy of the descriptors in comparison to the state-of-the-art approaches, the retrieval speeds as well as robustness to noise and deformation on different datasets are investigated.

On the other hand, to understand new complex models, we have also utilized distribution of electrical charge for proposing a system to decompose models into meaningful parts. Our robust, efficient and fully-automatic segmentation approach is able to specify the segments attached to the main part of a model as well as locating the boundary parts of the segments.

The segmentation ability of the proposed system is examined on the standard datasets and its timing and accuracy are compared with the existing state-of-the-art approaches.

Acknowledgement

As most likely this is the last academic degree I will obtain, I would like to thank all those who supported me during my study life, especially during my Ph.D. study at Dublin City University.

My first and foremost thanks go to my supervisor, Dr. Alistair Sutherland for all of his friendly supervision and great comments. Alistair was not only an academic supervisor to me, but also he has been a mentor who supported me academically and spiritually. Thanks a million Alistair. You made my student life easier than what it was; I never forget you and your valuable lessons in my life.

I extend special thanks to Dr. Martin Crane for his beneficial consultation during my research. Also, additional thanks must go to Professor Noel O'Connor and Dr. Cathal Gurrin for their interesting comments on my transfer talk and useful advices about various parts of my thesis.

I also would like to thank the members of the committee and especially the reviewers of the manuscript for having accepted this significant task.

It was a pleasure for me to be a member of the 3D Vision Group at DCU University. The stimulating research environment and friendly atmosphere made my latest student life pleasant. I am pretty sure that without their scientific contributions this thesis would be impossible; many thanks to my warm hearted friends and colleagues, Alireza Dehghani and Pooyan Jamshidi, for their comments on general research and mathematical aspects of the work. Apart from work, we have had great leisure times together such as soccer games and barbeque parties on the rainy days of Dublin.

Additionally, there are some other colleagues that I would like to acknowledge; Dexmont Pena, Mohammed Farouk and Fiona Dermody for all their support and help.

Outside of the professional sphere, special thanks must go to my parents, brothers and sisters for their continued support and encouragement throughout.

Last but not certainly least; I also thank (my) Somayyeh for her love and kind support during my study. I have no words to express my gratitude to her. And my daughter, Karya, the cutest and most beautiful girl in the world, thank you very much darling! I hope this thesis shows that their supports and sacrifices weren't in vain.

Fattah Alizadeh

Dublin, Ireland

2014/05/10

Dedication

To the soul of my Father, the first who taught me lessons of patience and sacrifice.

To my beloved Mother, for her prayers for me during my life.

And

To my Wife and Daughter, for all of their support and help.

1

Introduction

1.1. Introduction

The world is changing and digital technology is growing rapidly to enhance the quality of human life. Nowadays, no one can imagine daily life without using technological achievements such as the internet, mobile phones, mp3 players and so on. The growth in technology is so fast that every day one can find a lot of new multimedia information using search engines such as Google, Yahoo and so on. 3D models are among the new types of multimedia information, which have recently opened a new door to computer users to enjoy the amazing world of 3D. This growth is in part due to the development of 3D scanners, modelling software, cameras and even mobile phones with fast CPUs. Nowadays, 3D models play an important role in many applications namely games, cultural heritage, medical imaging, biochemistry, art and mechanical manufacturing.

As the result of emerging new technologies for saving and manipulating multimedia information in the above-mentioned applications, the number of 3D models has mushroomed in recent years.

As the number of 3D models becomes larger and larger, automatic tools for the search, retrieval and understanding of such models are becoming more necessary. These tools should be able to help users, whether from academia or business, to manage available models. Management of the models includes a variety of tasks for different applications: an engineer or designer may like to decompose a mechanical model into its components so as to reuse them to create new models. An archaeologist may like to search among existing artefacts to find similar ones to a recently-discovered valuable antique. A medical researcher should be able to search among available CT-Scan images to find a similar pattern among related disorders. Other activities may be required to be carried out by the animators, chemists *etc*.

Following the growth of 3D models on the internet or other specific-domain datasets, the question of "how do we generate 3D models?" has evolved into "how do we find them?" [1]. It means that the trend in research is toward proposing new efficient systems to search, index and retrieve a desired model in a large pool of 3D models. On the other hand, due to the vast amount of 3D models, the textual-based search engines are not applicable anymore. Accordingly, the process of finding a query model in a target dataset has become a problematic issue.

An effective remedy for this problem is to design an automatic system to perform content-based matching and retrieval. The system should be able to interact with users by getting a model as query and finding and retrieving similar models from the target dataset. The foundation of such systems is a shape descriptor, by which the 3D models are represented as a set of numerical vectors or an appropriate graph. A practical 3D model retrieval system with an ideal shape descriptor should be able to retrieve similar models in a reasonable period of time. Additionally, the underlying shape descriptor is expected to be robust to various transformations and surface changes.

Therefore, the first aim of the current thesis is to introduce beneficial shape descriptors for 3D model retrieval, which offer high discrimination quality as well as robustness to deformation, noise and other surface changes.

On the other hand, decomposing 3D models into meaningful parts is of great interest to researchers in various domains. Reverse engineering, partial matching and model annotation are among the beneficial applications of decomposing 3D models into semantic sub-parts. Despite the presence of dozens of existing approaches to segmentation, proposing new, accurate and robust techniques is still a challenging issue in computer vision.

The optimal segmentation approach would be able to extract components of the complex models, which have emerged due to recent advances in graphical software and hardware. As 3D models are getting larger and larger, the need for more advanced segmentation tools is becoming more crucial. So, the purpose of the second part of the current work is to propose an accurate, robust, efficient and fully-automatic tool by which the available models, having more sophisticated structure, can be easily segmented.

All in all, there are several main objectives for the current work as follows:

1) Introducing a new view-based 3D shape descriptor based on the 2D Poisson Equation.
2) Proposing two new 3D based shape descriptors using a well-known fact from the physics of electricity about distribution of electrical charge on the surface of a solid.
3) Exploring and evaluating the efficiency and effectiveness of the proposed descriptors using different criteria on the models in the standard datasets, and comparing them to the state-of-the-art approaches.
4) Designing an automatic 3D model segmentation system using the same fact about distribution of electrical charge on the surface of models.
5) Comparing the ability of the proposed segmentation system to the available ones, using widely accepted benchmarks.

In the sequel, we give a brief description about some topics related to the 3D models *viz.* 3D model definition, creation and representation.

1.2. 3D Model

A 3D model is simply an abstract representation of an object by showing the relationships among its components. It represents a 3D object using a collection of points in 3D space, connected by various geometric entities such as triangles, lines, curved surfaces *etc*. A 3D model can be considered as the fourth generation of multimedia information, which has emerged after digital sound in the 1970s, digital images in the 1980s and digital videos in the 1990s. They have become popular in parallel with the development of 3D data acquisition, 3D graphics modelling and graphics hardware technologies. Nowadays, 3D models are widely adopted in various applications namely medical industry, movies, games technology, science, architecture etc. Figure 1.1 exemplifies some models utilized in different applications. We refer the readers to Chapters 3 and 5 to see more details about applications of the 3D models in different domains.

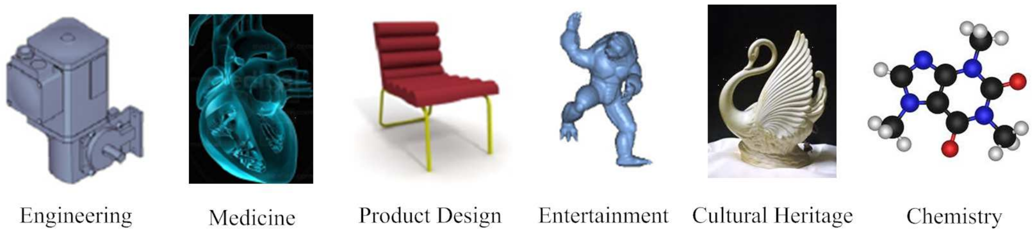


Figure 1.1, Sample 3D models from different applications.

1.3. 3D Model Creation Techniques

3D models can be generated using one of the following techniques:

- **3D Model Acquisition**: In this class of 3D modelling techniques, the real-world object is digitized to be saved and processed by computers. 3D Laser Scanners and Coordinate Measuring Machines (CMM) are two samples of modelling devices. In the former one, a real-world object is scanned, analysed and the raw data (typically an *x,y,z* point cloud) is used to generate an accurate polygonal or other mesh representation. Coordinate Measuring Machines, on the other hand, are designed to move a measuring probe to determine the coordinates of the points on the object. Figure 1.2 shows a 3D laser scanner and a CMM device used for construction of 3D models from real objects. 3D Model Acquisition techniques are mainly used in cultural heritage, animation and reverse engineering applications.

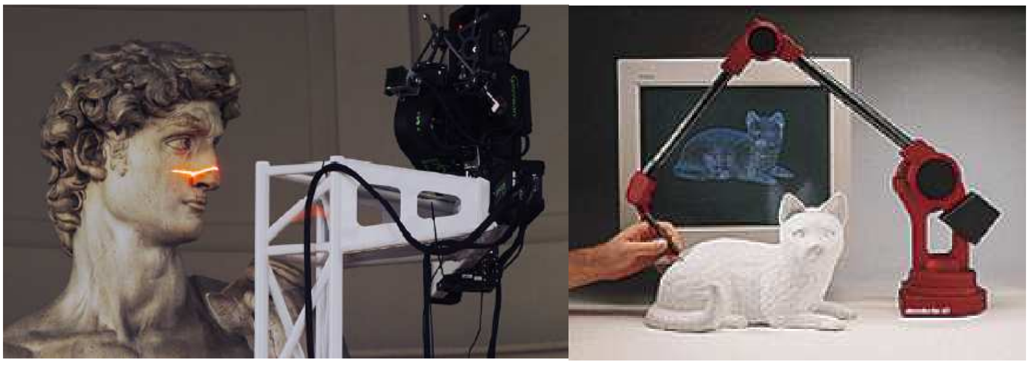


Figure 1.2, sample 3D model acquisition devices. Left: laser scanner-based acquisition of Michelangelo's David model from Stanford University, Right: A CMM device for generating a Cat model.

- **Manual Construction**: There is a wide range of software by which designers are able to construct their favourite 3D models. This class of software allows users to alter models by adding, subtracting and/or stretching desired parts of the models. A typical CAD software, for instance, enables engineers to generate a 3D plan of a building in a short period of time. Other 3D modellers such as Maya, 3D Max, Cheetah3D, Anim8or *etc.* are among beneficial modelling software packages, which have been designed for modelling and animating characters.

- **2D Image Combination**: 3D models can be constructed from a set of 2D images, when time and/or budgetary limitations do not allow the producers to manually generate a fully realized 3D model. In this situation, a 3D model is algorithmically derived from a set of static 2D images which are captured from different view angles. Then, an appropriate algorithm is applied to combine the images together to construct a 3D model. Figure 1.3 shows two samples of 3D face models constructed from a set of 2D images.

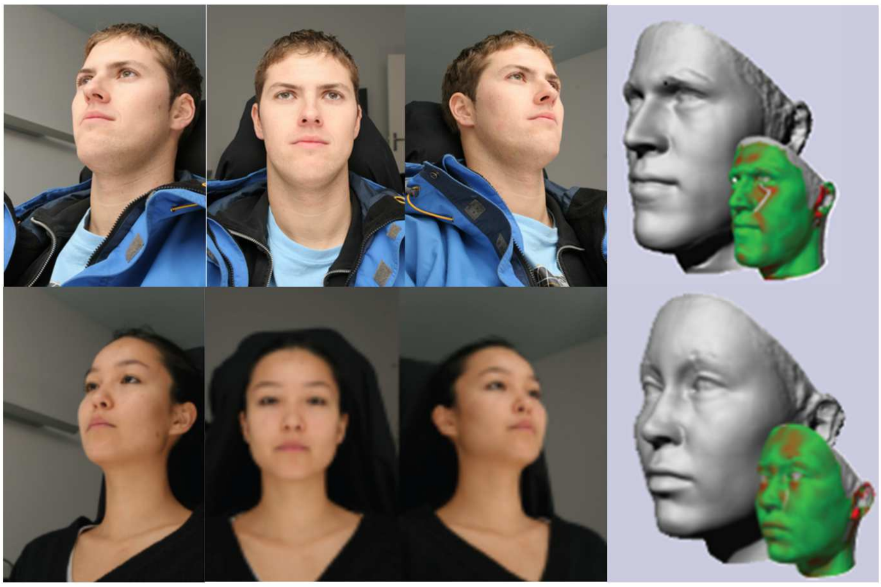


Figure 1.3, 3D face models reconstructed from a set of 2D images (image is taken from [2]).

1.4. 3D Model Representation Format

As the number of 3D models is becoming larger and larger, the demand for visualization of such data in different applications has aroused research interest. According to the application used, various formats of model representation have been proposed.

As displayed in Figure 1.4, 3D models can be represented based on their surface or volume information. The surface of a model can be represented using a point cloud, a polygonal mesh or a parametric surface, while the volumetric characteristics of an object are shown by voxels. These formats can be simply defined as follows:

- **Point Cloud representation**: Point Clouds are merely composed of a set of points and are often intended to represent the external surface of an object. So, the connectivity relations among the points are not considered in this representation format. A model represented in a cloud of points can be directly constructed by 3D scanners and usually is converted to the other formats to be used in graphic applications (Figure 1.4 (a)).

- **Mesh representation** (Triangular of Polygonal mesh): Polygonal meshes are among the most often adopted formats for object representation using a collection of vertices, edges and faces. They utilize connectivity (e.g. adjacency relationship between vertices), geometry (e.g. vertex location) and property data (e.g. normal vectors) to represent the objects. The boundary surface in this family of approaches is divided into a set of planar surfaces such as triangles or quadrilaterals. Figure 1.4 (b), 1.4(c) show a bunny model in two different face types.
- **Parametric Surface representation**: this format represents the object as a set of surfaces in Euclidean space R^3, which are defined by a parametric equation. This class of representation is the most efficient approach and unlike polygonal meshes, which only approximate the model surface, defines the exact model mathematically (Figure 1.4 (d)).
- **Voxel representation**: Voxels are another common representation of models in which the object is considered as a volumetric density defined on a 3D regular grid. This format aims at encoding the volumetric information of objects to be used mainly in medical applications. One can consider a voxel as a 3D counterpart of a pixel. Therefore, as in the case of pixels, the voxels themselves do not include their positions. Instead, the coordinates of a voxel in a model is specified based on the relative positions of the other voxels (Figure 1.4 (e)).

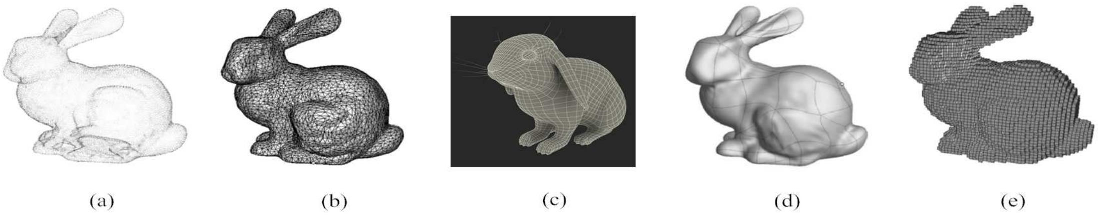


Figure 1.4, Different representation of a sample Bunny model. (a) Point clouds, (b), (c) polygonal meshes, (d) set of parametric surface and (e) voxel (figure is taken from [3]).

In the current work we mainly use the triangular mesh models saved in *.OFF* format, as it precisely shows model geometry and also can be converted to the other formats easily. We refer the reader to the appendix A to see more details about *.OFF* file formats and sample MATLAB code to read/write them.

1.5. Thesis Contributions

In this thesis we make three different contributions to the field of 3D model retrieval and segmentation. The first two contributions include introducing new shape descriptors, by which 3D models similar to a specific query from a target dataset are matched and retrieved. The last contribution is related to a segmentation technique for decomposing 3D objects into semantic meaningful parts:

- **2D-based 3D model descriptor**: For the first descriptor, we utilize silhouettes extracted from different view-points. The 2D Poisson Equation is applied to the silhouettes to assign a scalar value to each pixel. The pixel values taken together constitute the description of the silhouettes, by which the comparison between the models is carried out.
- **3D-based 3D model descriptor**: A well-known fact from physics about the distribution of electrical charge on the surface of a solid is borrowed to describe the models in a discriminative manner. After simulation of the charge distribution, the amount/density of charge on the faces of the model is employed as a local descriptor. Two different implementations of the fact are utilized to introduce two charge-based descriptors: in the first one, charge density and the well-known Bag-of-Feature framework describe the models, while the second one employs the amount of charge on the facets is combined with a term we call "Dense Patches". These two implementations offer considerable advantages, which are crucial in the retrieval domain.
- **3D model segmentation**: Charge distribution is also used to decompose 3D models into their meaningful parts. The simulated distributed charge will be utilized in both phases of segmentation: protruded parts extraction and boundary faces detection. Although the charge distribution has been utilized for 3D model segmentation previously, as will be discussed in Chapters 5 and 6, our approach offers great advantages over the previous method.

1.6. Thesis Outline

Following on from the above, the current thesis is organized as follows:

- **Chapter 2** discusses the related work in content-based 3D model retrieval as well as the state-of-the-art approaches proposed in the 3D model segmentation area.

- **Chapter 3** is dedicated to introducing the proposed shape descriptors including the 2D-based and 3D-based ones.
- **Chapter 4** provides extensive experimental results for the proposed shape descriptors on the standard datasets.
- **Chapter 5** details the proposed approach for segmentation of 3D models using distribution of electrical charge density.
- **Chapter 6** uses the proposed segmentation approach to decompose models in two standard datasets, SHREC'07 and SHREC'12.
- **Chapter 7** summarizes the implications and contributions, and discusses the possibilities for future research.
- **Appendix A** introduces Object File Format (.OFF) of 3D models along with a MATLAB code for read/write .OFF files.
- **Appendix B** presents information about the models on the query set, which is designed for the partial matching track of the SHREC' 07 contest.

2

Literature Review

2.1. Introduction

The fast growth in technology during the last two decades has led to an enormous volume of 3D models in databases or over the internet. Following this growth, the trend of research is going toward designing efficient systems to find a desired model in a large collection of 3D models. As a result, lots of effort has been made to propose new content-based techniques by which the desired models may be found. To review such efforts a section of this chapter is dedicated to the techniques, which have been proposed in the 3D object retrieval domain.

On the other hand, as the 3D models are becoming larger and more complex, decomposing such models into smaller and simpler meaningful parts is getting necessary in lots of graphics applications. Consequently, 3D mesh segmentation or decomposition has gained enormous attention from leading researchers in graphic domains. Therefore, the other part of this chapter aims at surveying all of the approaches, which fall in the segmentation field.

2.2. State-of-the-art in 3D Object Retrieval (3DOR)

Search and retrieval of 3D models started two decades ago and dozens of techniques have been proposed during this period of time. As the number of techniques has increased, a number of survey papers have been written on the topic [4, 5, 6, 7, 8, 9, 10, 11] among them [8] covers a wider range of available methods. But this later one is relatively old (published in 2008) and does not include the new efforts. However, due to the increasing number of proposed approaches, the new survey papers only cover a specific class of techniques in the literature, such as the review papers, which survey sketch-based approaches [9], partial matching [10], view-based approaches [11]. One of the most important goals of this section is to provide a comprehensive overview of retrieval techniques from all categories, focusing on the recently published ones.

Generally speaking, the 3D model retrieval techniques are grouped based upon the utilized shape descriptor by which the model matching problem is converted into numerical feature vector comparison or graph isomorphism. Different researchers have different ideas for the criterion by which the categorization is performed; local vs. global approaches, 2D vs. 3D approaches, spatial vs. frequency domain approaches, numerical vs. graph based approaches etc. For the sake of generality, we try to take into consideration all the available associated criteria to provide the readers with a full overview of the available approaches and discuss the advantages and drawbacks of each group. Figure 2.1 illustrates a graphical taxonomy of 3D shape descriptors utilized for model search and retrieval. As can be observed in this figure, we divide the available approaches into two main categories: 2D and 3D approaches, each of which has been classified into smaller groups.

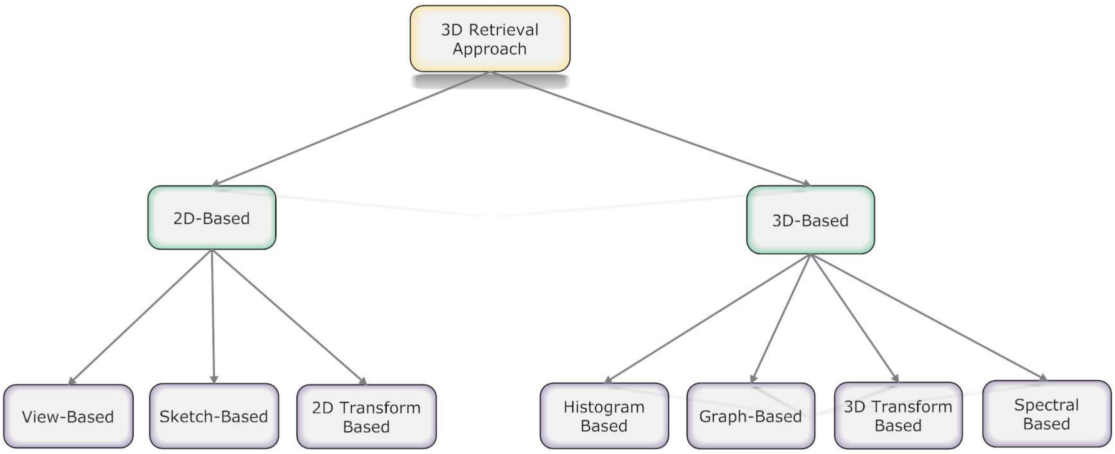


Figure 2.1, Taxonomy of 3D object retrieval approaches

2.2.1. 2D-Based Shape Descriptors

The main idea behind 2D-Based methods is that "two 3D models are similar, if they look similar from all viewing angles" [12]. Rather than using the 3D model itself for extracting shape descriptors, the methods, which lie in this category, generate shape descriptors using image projections. The projections may be silhouettes, depth buffers, contours or other kinds of image presentations. Using the above idea several researchers have tried to reduce 3D model-matching to a 2D image-matching process. Despite the fact that these techniques discard valuable information about the original 3D models, they have good discriminative power.

Based upon the query-type and the domain used, this class of shape descriptor is divided into three main sub-categories i.e. view-based and sketch-based and 2D Transform-based descriptors. In the sequel, we explain all of these sub-categories along with the proposed approaches, which fall in each group.

2.2.1.1. View-Based Descriptors (Spatial Domain)

The approaches of this group generally begin with extracting a desired number of 2D view images from each 3D model, including the query and the entire target models in the dataset. Then, in order to compute shape signatures, a 2D shape descriptor is applied on the extracted views. These signatures are finally exploited so as to measure the similarity between the query and target models to retrieve the best matches.

Since some of the view-based approaches use silhouettes and depth buffers, in the sequel we give a simple definition of these images before discussing the relevant retrieval systems.

Silhouette: A silhouette is an image represented as a solid shape of a single colour, usually black, its edges matching the outline of the subject. As shown in the first row of Figure 2.2, the interior of a silhouette is featureless, and the whole is typically presented on a light background, usually white, or none at all [13].

Depth buffer (Range Image or Depth Map): The depth buffer is a useful technique for model representation in which the spatial distance difference of 3D surfaces are encoded into different grey values of their 2D projection images. It contains information relating to the distance of the surfaces of scene objects from a viewpoint (see second row of Figure 2.2).

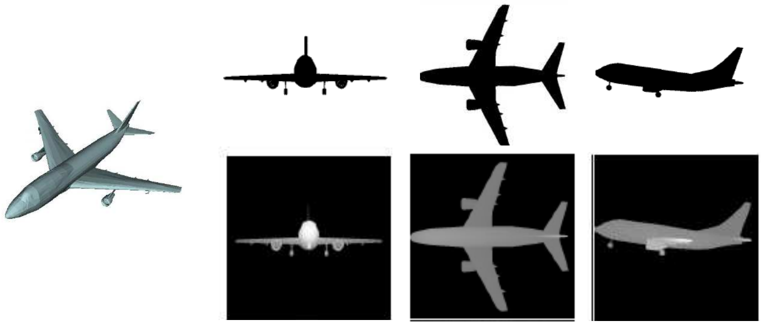


Figure 2.2, a 3D model of an airplane (leftmost) and corresponding three silhouettes (first row) and depth-buffer images (second row) captured from three directions (figure is taken from [14]).

There are several retrieval systems in which depth buffers play a critical role [15, 16, 17].

In the work of Shih *et al*. [16], depth-buffer images are used to introduce the elevation shape descriptor. Six depth-buffer views or elevations are generated from faces of the 3D model's bounding box and then each view is described by a grey level image, which is decomposed into several concentric circular areas. The sum of pixel values within areas between adjacent concentric circles generates the view descriptor. 3D models are compared based on the matching of the elevation descriptors. To match two groups of the elevation descriptors, the minimal distance is calculated to measure the distance between the two 3D models.

In a different aspect, Ohbuchi *et al*. [17], used depth-buffers combined with a Bag-of-Visual-Words-based retrieval framework. In this method, each 3D model is rendered into a group of depth images and SIFT features [18] are extracted from them. It uses the bag-of-features framework to integrate the local features into a feature vector for each model. Then the matching of these two feature vectors determines the distance between the two 3D models.

Recently the depth-buffer image is used in a very interesting method by Lian *et al*. [15]. After simplification of all models they applied Principal Component Analysis (PCA) and Multi-Dimensional Scaling (MDS) to achieve rotation-invariant and canonical form of models, respectively. Then, using different viewpoints of geodesic sphere, 66 depth-buffer views are extracted from the canonical form which are described using a combination of salient SIFT descriptors and Bag-of-Feature framework. The remarkable part of Lian's work lies in using a new comparison method, Clock Matching, which defines the similarity between each pair of models by using the minimum distance of their 24 matching pairs. Ignoring the matching time, their method attained achieved a very good ranking among the participants of SHREC'11 contest [19]

Differently, Napoleon *et al*. employed silhouette views in their proposed framework [20]. After normalization of the 3D models they extracted a set of silhouette views for each model and defined the distance between any two silhouettes as the number of pixels that are not common between the two (Hamming Distance). The final dissimilarity between each pair of models in their work is equal to the sum of the distances between their two sets of silhouettes.

In a different work, Johnson and Herbert [21] used spin images to compare 3D model surfaces. The spin image is defined as follows:

Spin Image: the spin image at each point *p* is a 2D histogram of the surface location around that point. As illustrated in Figure 2.3, this histogram accumulates the number of points located at the coordinate (α , β), where α and β are the length of the two orthogonal edges of the triangle formed by the oriented basis point p, whose orientation is defined by the normal vector n and support point q. the final size of the spin images is defined by the width *w* and the height *h* of the spin plane.

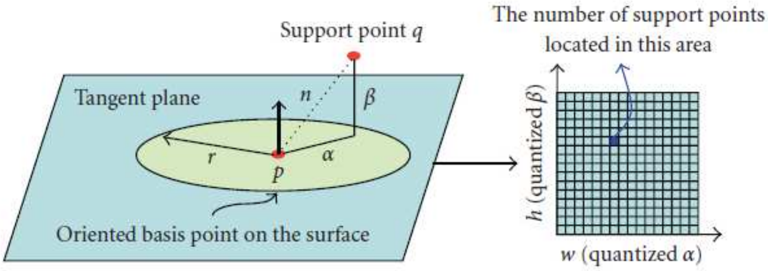


Figure 2.3, The demonstration of Spin Image (figure is taken from [22])

They applied the spin image descriptor to recognize and retrieve models among a cluttered scene. In spite of possessing valuable characteristics such as invariance to rigid transformations and flexibility, the spin image descriptor suffers from two main problems: firstly due to complexity, it is difficult to apply to 3D shape matching and secondly it is not clear how to define a dissimilarity function that satisfies the triangle inequality.

In an interesting work, panoramic views are combined with the SIFT descriptor so as to extract and describe the salient points in order to work with range image queries [23]. (The panoramic views are obtained by projecting the 3D model onto the lateral surface of a cylinder aligned with one of the object's three principal axes and centred at the centroid of the object (see Figure 2.9)). The authors extend the original PANORAMA [24] method to multiple axes in which each axis defines three panoramic view cylinders. Having panoramic views, the SIFT descriptor is calculated on the cylindrical depth images. Beside the retrieval quality of this method, it offers a fully unsupervised retrieval system.

Very recently, panoramic views are also used in a partial matching context by the same authors [25]. The Dense-SIFT (DSIFT) [26] extracts the feature points to describe the panoramic views. In the case of the query model, the depth-buffer image is taken from the model with a size of 256*256 pixels and the DSIFT descriptor is directly computed on the images. The similarity distance between models is computed as the minimum L2 distance of every DSIFT point of the range image to every DSIFT points of a 3D model's panoramic view. The performance of the method is only evaluated on a pottery dataset originating from the Hampson Archaeological Museum and no quantitative comparison to any other similar partial-matcher is provided.

The idea of a Light Field Descriptor [27] is used in the recent work of Ding *et al*. [28] in which 300 projected views are extracted from uniformly distributed cameras on a view sphere. To reduce the computational cost of view comparison, the authors use Gaussian Means (G-means) adaptive clustering [29] to select the most informative views. The Bag-of-view-word (BOVW) paradigm is used to generate model histograms in which each model is defined as a multi-resolution histogram that is composed of several BoVW descriptors with different numbers of bins. In addition to more accuracy, compared to the original LFD descriptor [27], the best-view selection approach of the proposed work leads to a faster retrieval process.

Gao *et al*. proposed a Spatial Structure Circular Descriptor (SSCD) [30], which can preserve the global spatial structure of 3D models and is invariant to rotation and scaling. All the spatial information of a 3D model can be represented by an SSCD which includes several SSCD images. In a SSCD, a minimal bounding sphere of the 3D model is computed, and all points on the 3D model surface are projected onto the bounding sphere. Attribute values are provided at each point to represent the surface spatial information. The bounding sphere is further projected onto a circular region of a plane. It can preserve the spatial structure of the original 3D model. This circular image was employed by each SSCD image to describe the surface information of a 3D model. Each spatial part of a 3D model is represented by one part of the SSCD individually (See Figure 2.4). Finally, histogram information is employed by the SSCD as the feature to compare two 3D models.

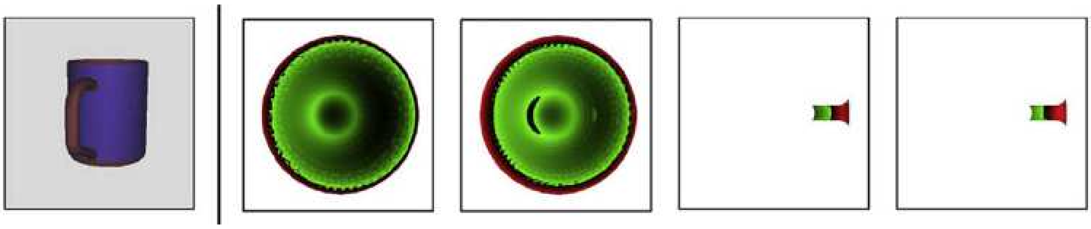


Figure 2.4, A 3D cup models and the corresponding example SSCD images proposed by Gao *et al*. [30]

A combination of SSCD descriptor and topological information of the models has been recently exploited to achieve higher retrieval performance [31]. In this method multiple views of 3D models are selected by topological structure. To do so, 20 topological salient points on the surface are extracted by using Radial Basis Functions (RBF) [32] in order to construct a Multiresolutional Reeb Graph (MRG) [33]. Then, unlike other view-based approaches, an image is rendered for each topological point, via an orthographic projection of the 3D model onto the plane tangent to the sphere at each point. In order to extract and describe features the SIFT descriptor is used which is quantized by using a modified bag-of-features (BOF). In this version of BOF the spatial structure information are preserved via combined shell-sector with logarithmic shell radii (see Figure 2.5). Finally, the similarity between two 3D models is computed using the Earth Mover Distance (EMD) [35].

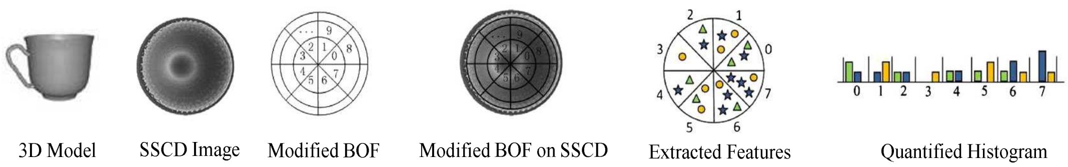


Figure 2.5, The illustration of the modified BOF method using SSCD proposed by Li *et al.* [31]

Some of the recent view-based methods select a subset of the extracted views as salient ones to reduce the computational cost of similarity measures [36] [37]. Instead of comparing all of the available rendered images, such methods use a criterion to choose some views to be employed in the final stage of retrieving the best matches.

Atmosukarto and Shapiro [36] applied a learning approach to specify the salient views to describe a 3D object. In this work, the initial feature points are extracted according to the Gaussian curvature and the Azimuth angle (angle between the positive *x*-*z* plane and the projection of normal vector *n* to the *x*) and elevation angle (angle between the *x* plane and vector normal vector *n*) for each point on the model surface [38]. Using a local histogram, the feature points are aggregated by taking a neighbourhood around each mesh point and accumulating the low-level feature values in that neighbourhood. Then, the histograms are fed to a Support Vector Machine (SVM) classifier to learn the salient points on the 3D surface mesh. Among the 100 extracted silhouette views, the top K (12 in their work) distinct views having most salient points are chosen as salient views to compute the similarity factor between query and target models. In addition to higher retrieval accuracy, compared to the similar work of Chen et al. [27], the authors claimed that their method can achieve a 15-fold speedup in feature extraction time.

In the same category, Ansary *et al*. [37] introduced an Adaptive Views Clustering (AVC) method. In AVC, there are 320 initial captured views, among which the representative views are optimally selected by adaptive view clustering with Bayesian information criteria. A probabilistic method is then employed to calculate the similarity between two 3D models, and those objects with high probability are selected as the retrieval results.

There are two parameters in the method, which are used to modulate the probabilities of objects and views respectively.

2.2.1.2. Sketch-Based Descriptor

Sketch-based 3D shape descriptors are of great interest to 3D model retrieval researchers. So that a considerable amount of research has be done in the literature [1, 39, 40]. The two recent contests on sketch-based 3D model retrieval in the SHREC'12 [41] and SHREC'13 [42] benchmarking initiatives show the importance of this emerging field.

This class of approaches aims at search and retrieving relevant 3D models using an input human-drawn 2D sketch as a query.

Figure 2.6 shows a snapshot of the query-by-sketch retrieval system designed by Daras and Axenopoulos [43].

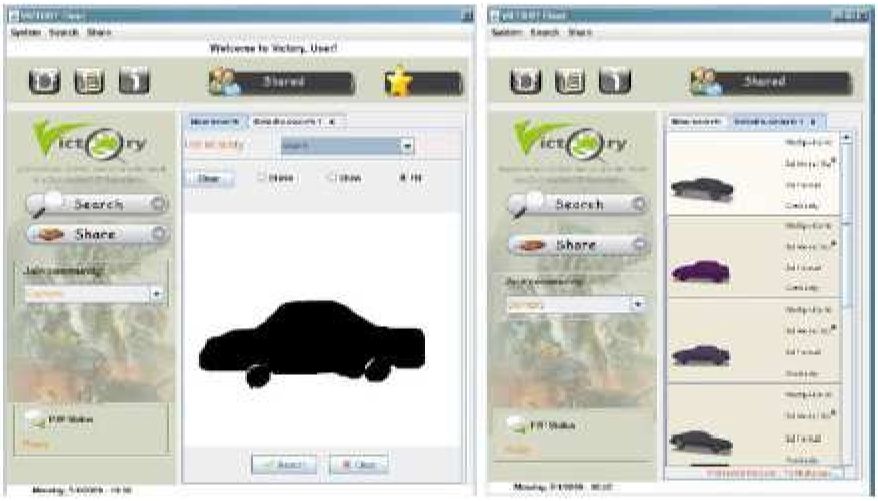


Figure 2.6, a sample interface of a retrieval system using the hand-drawn sketch as the query (figure is taken from [43]).

Actually, the motivation for developing this class of approach stems from the unavailability of all sample queries for the users. That is, when users do not have access to a sample desired model to perform query-by-example, they prefer to draw a sketch of the favorite model to show the visual properties of it and ask the system to retrieve similar models (query-by-sketch).

On the other hand, due to presence of a semantic gap between the sketches humans draw and the 3D models in the database [44], it is not a trivial task to design a retrieval framework in which the models are searched and retrieved according to their similarity to the sketch query. Therefore, a considerable amount of research works have been dedicated to ease this task [1, 39, 40, 45, 46].

Generally speaking, as demonstrated in Figure 2.7, this class of approach comprises the following steps:

i. A desired number of views are extracted from available models in the target dataset.
ii. The models are described using a shape descriptor applied on the extracted views.
iii. User is asked to draw a sketch as a query to feed the retrieval system by its descriptor.
iv. The best matches are retrieved from the target dataset according to similarity between the query sketch and extracted view.

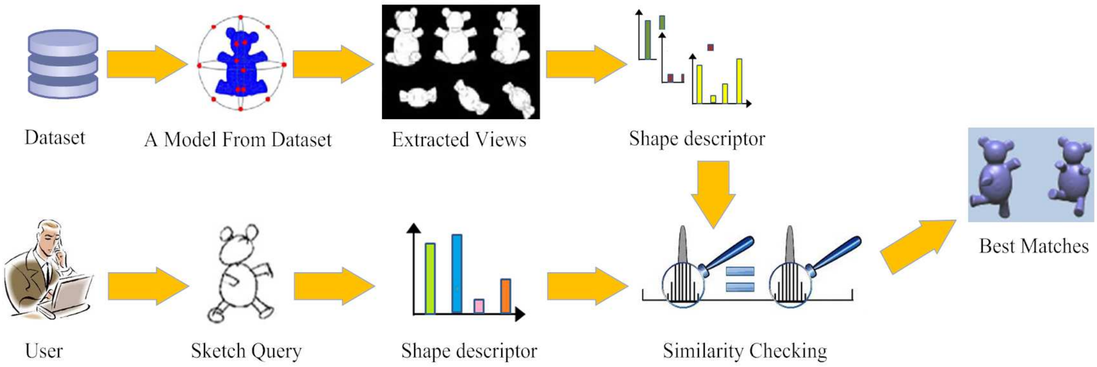


Figure 2.7, A typical sketch-based 3DOR system.

To the best of our knowledge, the first content-based query-by-sketch retrieval system for 3D models has been proposed by Funkhouser *et al.* [1]. It supports textual query along with both 2D and 3D queries based on a general orientation-invariant version of spherical harmonics [47]. First, they extract thirteen orthographic projection views of 3D models, for which the cameras are positioned on the edges and vertices of a bounding cube. Then, the user is asked to draw sketches from three principal view directions which are going to be matched against extracted views of the available models. Finally, by comparing the corresponding view descriptors to the boundary descriptor of the input sketch the best matches are found and retrieved.

The proposed work of Yoon *et al.* is another approach in this category [39]. The authors mention that those sketch-based retrieval systems, which employ silhouette or depth buffer views, have crucial drawbacks such as: inability of users in sketching depth buffer query or losing view-dependent information of silhouettes. To overcome these drawbacks, they propose to automatically extract suggestive contours [48] to construct descriptors from fourteen captured views. As shown in Figure 2.8, the suggestive contour rendering is a minimal way to convey a 3D model by a 2D image. It closely resembles the way most people sketch 3D objects: The silhouette as well as major ridges and valleys of the model are outlined. To describe the extracted views, the binary images are converted to their corresponding diffusion tensor field, which enables the authors to analyse the orientation of each pixel according to its neighbourhoods [49]. The similarity between models is defined via computing the minimal distance between the feature vectors of the query and each view image. To evaluate their approach, they built their own benchmark which contains 250 sketches for the 260 models of the Watertight Model Benchmark (WMB) dataset [50] categorized into 13 classes. The experimental result has shown the robustness of the proposed approach against variation of shape and pose of user drawn sketches.

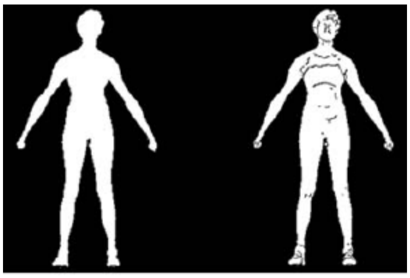


Figure 2.8, comparison of contour (left) and suggestive contour (right) (Figure is taken from [39]).

Very recently, the same authors extended the idea of using suggestive contour by a sparse coding representation [45]. Following computing the suggestive contour of the fourteen views extracted from virtual cameras equally distributed on a sphere around the object, a topological space of diffusion tensor fields (HOG-DFT) is used to describe the 2D images by a feature vector. The major difference between this work and their previous one [39] is related to the feature optimization using sparse coding applied on the generated feature vectors. Sparse coding is a well-known method [51] to find the optimal representation of input data using a linear combination of a smaller trained dictionary. The authors used such a method to optimally represent the original HOG-DFT feature vectors. Compared to the other participants of the sketch-based retrieval competition in the SHREC'12 contest, the experimental results demonstrate the ability of the proposed approach in terms of both time and accuracy [41].

Two of the most efficient approaches in the literature are proposed by Furuya and Ohbuchi [40] and Li and John [46]. They gained the best ranks among all fifteen state-of-the-art sketch-based approaches participated in the SHREC'13 contest [42][41].

Compared to the other similar ones, the considerable improvement of Furuya and Ohbuchi's work is related to the extra step which is inserted to the retrieval pipeline to fill the gap between sketches and rendered images of 3D models. That is, the user-drawn sketches contain stylistic variations, abstraction, inaccuracy and instability [44] which result in inappropriate comparisons of human-drawn 2D sketches with a set of rendered view images of a 3D model. To overcome this issue the authors have applied distance metric learning methods such as Uniform Manifold Ranking (UMR) and Cross-Domain Manifold Ranking (CDMR).

The UMR method is applied on the features extracted from both sketches and the rendered view images to learn a graph structure or a uniform manifold that reflects a low-dimensional representation of the features. Then, a manifold ranking algorithm [52] is employed to compute the diffusion distance between features of the sketch query and rendered views. The CDMR, on the other hand, can be used for all possible similarity measures e.g. sketch-to-sketch, sketch-to-3D and 3D-to-3D. Unlike the UMR which forms a manifold by using a single feature, CDMR uses multiple measures of similarities including both feature-based and semantic-label-based measures.

Therefore, the proposed work of Furuya and Ohbuchi can be summarized as follows: the 42 extracted range images along with the query sketch are described using Dense-SIFT [26]. Two manifolds are separately formed for the both the extracted views and user-drawn sketch. The BF-fGALIF feature [53] is then used to couple the two manifolds into a Cross-Domain Manifold (CDM). Once the CDM is constructed, relevance values from the sketch to images are computed as the similarity distances to retrieve the best matches to the sketch query.

The 2D-3D alignment-based approach of Li and John [46], on the other hand, comprises two stages of pre-computation and retrieval. In the pre-computing stage the 3D model features are computed for a set of sample views of target models, which are called View Context Features [54]. The View Context Features are utilized to select a set of candidate views to align a 2D sketch with a 3D model in the retrieval stage. This alignment step avoids brute-force direct matching between the sketch and many sample views which leads to a faster retrieval process. The computationally efficient integrated image descriptor named ZFEC is employed to represent global features from different aspects for the 81 extracted views. It combines Zernike moments (Z), Fourier descriptors (F), eccentricity (E) and circularity (C) in a feature vector. Finally, the sketch-to-model distances are obtained by comparing the sketch query with every candidate view using the relative shape context matching [34].

This approach along with the Cross-Domain Manifold-based approach of Furuya and Ohbuchi [40] outperforms all other approaches for both small-scale [41] and large-scale [42] benchmarks in terms of accuracy and scalability.

Generally speaking, this class of retrieval approaches suffers from the fact that the query and the target models are not of the same structure anymore; as mentioned before, the query is a free-form sketch containing stylistic variations, abstraction, inaccuracy and instability as sketched by the user. But the target objects, on the other hand, are graphical models defined according to the given file format. However, the recent advances in such approaches have demonstrated promising results to reach retrieval accuracy as high as their query-by-example counterparts.

It is worthwhile to note that there are some 2D-based approaches which support both hand-drawn sketches and rendered images as query [24, 27]. But, since they describe the models in the frequency domain they will be discussed in the 2D Transform-Based section (Section 2.2.1.3).

2.2.1.3. 2D Transform-Based Descriptor

Unlike the previous approaches, this group of methods tries to transform the working space to domains, mainly the frequency domain, other than the spatial one. The 2D Fourier Transform is one of the most utilized methods in this category. It is used alone or in combination with other Transformations (Such as Zernike moments and Krawtchouk moments) to represent the models.

Papakadis *et al*. [24] employ a set of panoramic views to capture the position and orientation of the model's surface as the 3D model descriptors. They obtain the views by projecting the 3D model onto the lateral surface of a cylinder aligned with one of the object's three principal axes and centred at the centroid of the object (see Figure 2.9). The panoramic views are described by the corresponding 2D Discrete Fourier Transform as well as 2D Discrete Wavelet Transform. Finally the Relevance Feedback technique is utilized to increase the retrieval performance.

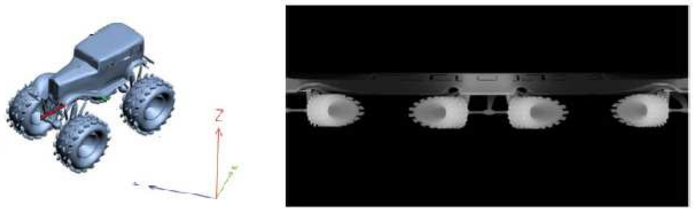


Figure 2.9, sample panoramic view of a car model (figure is taken from [24])

The well-know Light Field Descriptor (LFD) proposed by Chen *et al.* [27] is employed in an example-based retrieval framework that also supports query-by-sketch. In order to describe each model, one hundred silhouettes, excluding symmetry, are extracted from ten viewing angles distributed evenly on the viewing sphere (Figure 2.10). After applying both Zernike moments and Fourier transform on each silhouette, thirty five coefficients of Zernike moments along with ten coefficients of the Fourier transform are computed to constitute the feature vector of each silhouette. The dissimilarity of two objects is then defined as the minimal dissimilarity obtained by rotating the viewing sphere of one light-field descriptor relative to the other LFD. In order to make this method rotation-invariant, the authors consider ten images per viewing point obtained by uniformly varying all camera positions in the neighbourhood of the viewing point. Their experiments show that their approach has high retrieval quality and outperforms some other methods. Beside high computational cost of descriptor generation, the drawback of the proposed framework for the query-by-sketch systems is related to the fact that the LFD descriptor can be defined only for closed contour curves, which is not how humans sketch for shape retrieval [55].

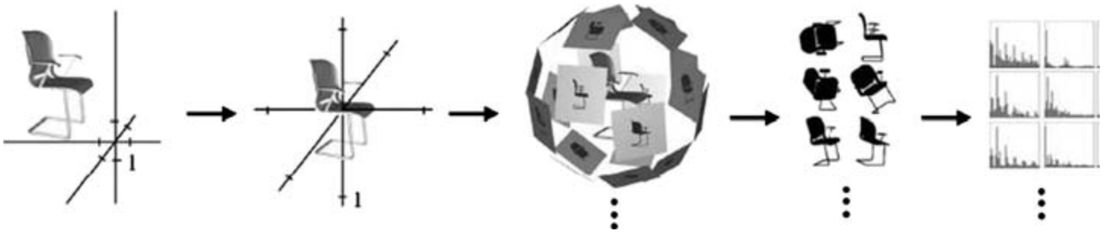


Figure 2.10, Computing the Light-Field descriptor for a chair model, by Chen *et al.*[27] (figure taken from survey Veltkamp).

Depth-buffer views can also be seen in the descriptor proposed by Ohbuchi *et al.* [56]. They use multiple depth-buffer views of the 3D model that are viewed from 42 viewpoints. Each depth buffer is then transformed to the $r - \theta$ domain and the Fourier transform is applied. The set of 42 feature vectors comprises the shape descriptor of the model, which is called the Multiple Orientation Depth Fourier Descriptor (MODFD). One of their problems is related to the speed of matching; they had to calculate the dissimilarity between two models by comparing all possible pairs of feature vectors which increases comparison time.

In the approach described by Vranic [14], each model is described by the coefficients of the Discrete Fourier transform applied on the model's silhouettes. In his approach, a 3D-object is first aligned using the Continuous Principal Component Analysis method

(CPCA) and then is projected perpendicularly onto the coordinate planes in order to generate three monochrome silhouettes.

In the same work using Depth Buffer images, Vranic introduced a new image-based descriptor. The new feature vector is obtained by projecting a 3D object to the six faces of a cube that encloses the object. The Depth Buffer-based approach uses the distances of the object's points from the projection planes. Each depth-buffer is then described by applying the respective 2D Fourier transform.

Another 2D transform-based retrieval framework is designed by Daras and Axenopoulos [43] which supports both sketch-based as well as example-based retrieval. The process starts with a pose normalization phase by applying Principal Component Analysis (PCA) and Visual Contact Area (VCA). Then, the models are described via three different descriptors: a) Polar Fourier Transform, b) 2D Zernike moments and c) 2D Krawtchouk moments which are applied on the eighteen different views extracted from a 32-hedron. To provide the system with the 2D query, the user is allowed to take an image either by drawing a sketch or from a digital camera. The Authors claim that since the quality of the input hand-drawn sketches depends on the users' drawing skills, even a few relevant hits among the retrieved results are enough. They suggest that such a retrieval framework should be combined with a relevance-feedback system, in which the user can select one of the relevant retrieved objects as a new query in order to retrieve more relevant results. While a qualitative evaluation demonstrates good retrieval results, the authors do not perform a quantitative evaluation for their sketch-based retrieval.

2.2.2. 3D-Based Shape Descriptors

Unlike the previous approaches, the 3D-based shape descriptors try to extract the distribution of 3D features to characterize the information about a 3D model. The 3D features can be either global or local. They can be grouped into four categories namely Histogram (Statistical)-based, Transform-based, Graph-based and Heat Kernel-based approaches.

2.2.2.1. Histogram (statistical)-based approaches

The methods in this category use a histogram which usually acts as an accumulator for collecting numerical values of specific property derived from the 3D models. Since these methods can be adopted as part of a machine learning framework, they hold a very important position in retrieval approaches [57].

A key work by Osada *et al*. [58] focuses on a set of functions based on geometric measurements (e.g., angles, distances, areas, and volumes) using some random points on the surface of the 3D model. They define their functions as follows (Figure 2.11):

- **A3**: Measures the angle between three random points on the surface of a 3D model.
- **D1**: Measures the distance between a fixed point and one random point on the surface. We use the centroid of the boundary of the model as the fixed point.
- **D2**: Measures the distance between two random points on the surface.
- **D3**: Measures the square root of the area of the triangle between three random points on the surface.
- **D4**: Measures the cube root of the volume of the tetrahedron between four random points on the surface.

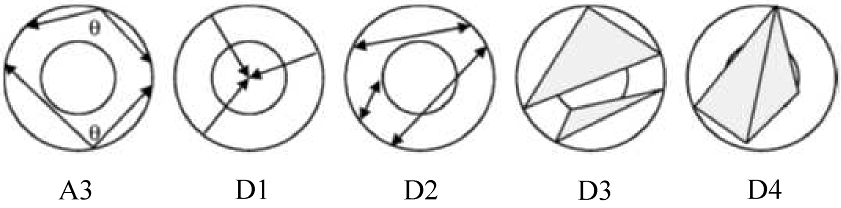


Figure 2.11, 5 simple functions based on geometric measurements by Osada *et al*. [58].

Then, the set of these functions form histograms, by which the similarity between the objects is specified. The accuracy of the histograms could be altered by changing the number of random points. Although D2 gets the best result among all of proposed methods, generally speaking, none of the functions have enough ability for describing 3D models.

The D2 function was extended later by Ohbuchi *et al*. [59] to create mD2. The difference between the original D2 and mD2 is related to the way that the sample points are selected; that is, the mD2 function uses quasi-random sequence of numbers instead of pseudo-random sequences. The authors claimed that with only a small increase in computational cost they could achieve significant performance improvements such as invariance to similarity transformation and be tolerant of topological and geometrical errors and degeneracies.

Another extension to the original D2 function has been recently proposed by Yuanhao *et al*. [60]. The main difference between the new D2 (Quick-D2) and the original one is related to the sampling phase; instead of selecting N^2 sample distances to extract the signature, as done in the original D2, the proposed Quick-D2 first selects N sample points on the surface and then calculates the distances between every pair of these points.

Therefore, taking samples on the points instead of the distances leads to reduce the number of points that the algorithm needs to get. Although the extracted shape descriptors for the both version of D2 are the same, the quick-D2 is much faster than the original version of D2 so that the average time cost of the quick-D2 is 75% faster than the original one.

The shape histogram proposed by Ankerst *et al.* [61] has been evaluated in the context of molecular biology and reached good accuracy and performance. They decomposed the 3D models using one of the following three techniques: Shell model, Bin model and spider-web or combined model (Figure 2.12).

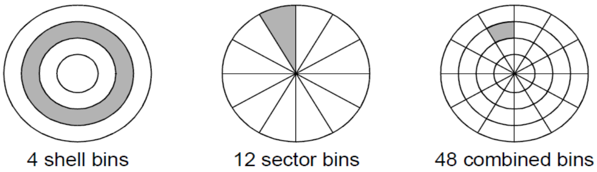


Figure 2.12**,** Three decomposition techniques (figure is taken from [61])

Their technique is not invariant to rigid transformations and so they had to do pose-normalization as a pre-processing step. Also since the approach proceeds with voxel data, 3D objects represented by polygonal meshes need to be voxelized prior to descriptor extraction.

The Extended Gaussian Image (EGI), introduced by Torn was initially proposed for machine vision applications [62]. The EGI is a spherical histogram in which bins accumulate the count of the spherical angles of the surface normal per triangle, usually weighted by triangle area.

The major drawback of the EGI is related to its inability in distinguishing non-convex and convex objects with the same EGI (see Figure 2.13). To overcome this issue Zhang *et al*. [63] extended the original EGI to compute the volume distribution of an object without canonical alignment. Additionally, the new EGI, which was called Volumetric Extended Gaussian Image (VEGI), was also able to maintain the properties of translation, scaling and orientation invariance. The experimental results of the proposed VEGI show that it outperforms the original EGI in terms of time and accuracy.

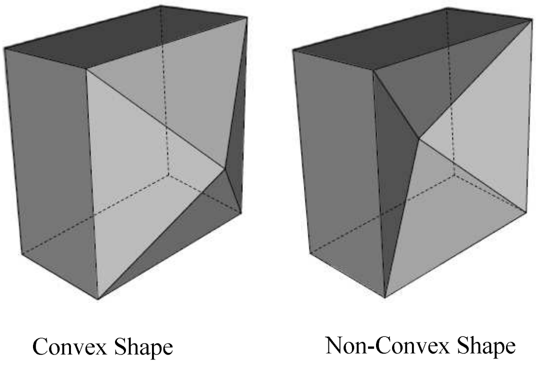


Figure 2.13, two models with different volume and VEGI but the same EGI (figure is taken from [64]).

Another extension to EGI was proposed by Wang *et al*. [65] so as to solve the aforementioned drawback of the original EGI. The proposed multi-shell Extended Gaussian Image (MSEGI) aims at differentiating between non-convex and convex objects by decomposing 3D models into multi-concentric shells, dividing the surface of each shell into cells and capturing the surface area distribution and surface orientation distribution of a 3D model in each concentric shell. Finally, the spherical harmonic transform of this surface area distribution function are adopted to extract shape descriptors. Although the retrieval ability of MSEGI was examined on the models in the PSB dataset, the authors did not compare their results to those of obtained by VEGI.

Paquet *et al.* [66, 67] use two kinds of geometric features as well as several photometric properties. The geometric features are cord and angle histograms. They define a cord as a vector from the centre of mass of the model to the centre of mass of a bounded region on the surface of the object in a triangulated surface. They define a set of statistics from the cords such as the distribution of angles between the cords and also the distribution of radii. Then histograms are employed to accumulate the distributions. Photometric properties include colour, reflectance, and texture. For these purpose they use voxelized models and assign a number to each voxel. This number is calculated using information from the texture, material reflectivity and vertex colour. Their techniques are easy to implement but since they only consider the global properties of the model, their proposed approach is not very discriminative about object details.

In contrast, Mademlis *et al*. [68] employ electrostatic fields to 3D model retrieval. They considered the complete voxelized 3D model as a distribution of electric charge. Their descriptor histograms comprise three different result fields: field potential value, field density Euclidean norms and the radial component of the field density. Changing the parameters of the descriptors enabled them to extract 24 histograms for each 3D model. The interesting characteristics of their descriptors are robustness with respect to object degeneracies and native invariance under rotation and translation. But it is notable that the proposed descriptors are sensitive to non-rigid transforms. They illustrate the shortcomings of the widely adopted Minkowski distance to compare two shape histograms. So, the normalized distance [69] and the diffusion distance [70] are utilized in their approach.

Akgul *et al*. [71] derived shape descriptor from the Probability Density Function (PDF) of a multivariate local feature computed on the surface of the object. Specifically, the vector of PDF values obtained by kernel density estimation (KDE) becomes the shape descriptor. They compared 3D models by the distance between their PDFs. The main advantages of their descriptor comprise a good accuracy rate as well as insensitivity to mesh resolution and small shape perturbation but since the descriptor is neither scale nor rotation-invariant pose normalization must be performed during the pre-processing step.

The histogram-based approach of Gal *et al*. describes the models using a histogram which is a combination of the distribution of two scalar functions defined on the boundary surface of the 3D shape i.e. Local Diameter Function and Centricity [72]. These two functions are defined as follows:

- **Local Diameter Function (LDF):** the function that measures the diameter of the 3D shape in the neighbourhood of each vertex. It is formed using a robust statistics measure of the diameters in a cone around the direction opposite to the normal of the point. As shown in Figure 2.14, 50 sample rays are shot in a cone shape region to the opposite side of the normal vector and the LDF is defined as a weighted average of the lengths of the rays.

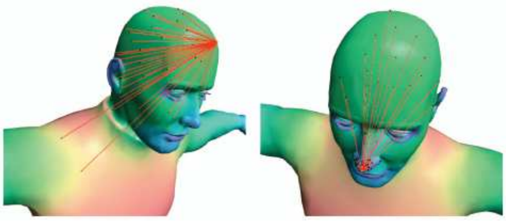


Figure 2.14, Examples of shot rays in a Cone-shape region (figure is taken from [72])

- **Centricity Function (CF):** The centricity of each vertex is defined as the average geodesic distance to all other vertices. The CF is normalized via dividing the centricity value of each vertex by the maximum centricity value on the mesh to arrive at a CF function value between 0 and 1.

Each model is finally described by a 2-D histogram (CDF) which is formed by a combination of the LDF and CF functions. As shown in Figure 2.15, this histogram offers a pose-oblivious descriptor which also remains the same under translation, scale and rotation transformations.

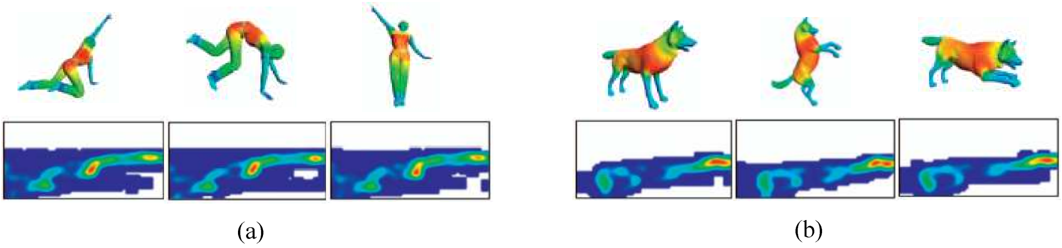


Figure 2.15, A sample 2D histogram CDF of different poses of (a) human model and (b) dog model (figure is taken from [72]).

Most recently, the Poisson-based descriptor on voxelized 3D models was introduced by Pan *et al*. [73]. They illustrated drawbacks of the distance transform and claimed that the Poisson equation is more stable because it essentially places a set of particles at each vertex and lets them move in a random walk until they hit the boundary. It measures the hitting time for a symmetric random walk to reach the boundary considering the values of a particle's neighbour. Then they created a histogram-based descriptor by accumulating the values of the Poisson equation in bins. The descriptor is robust to noise, simplification, smoothing and rotation transform. But similar to almost all of other statistical-based methods this approach has difficulty in distinguishing different models with similar structures.

The main advantage of histogram-based approaches is their simplicity of implementation. Almost all of the aforementioned methods are very straightforward to implement and understand. But their drawback is mainly related to low power of discrimination so that they do not have high ability in distinguishing similar models. They also suffer from sensitivity to rotation transformation which necessitates an extra pre-processing task. These methods can be combined with other methods such as a pre-processing step or an active filter to improve their performance.

2.2.2.2. Graph-based approaches

Unlike other approaches, techniques of the graph-based category utilize graphs to represent 3D model features. They convert the similarity matching problem among models into a graph isomorphism problem. Although approaches of this class are more elaborate and complex, they have the ability of encoding both geometrical and topological structures of models in a more natural manner. The similarity among models is defined using graph-matching algorithms. Reeb, Size and Skeletal graphs are among the techniques which are used to describe and match 3D models.

A Reeb graph is a topological skeleton defined using a scalar continuous function on a 3D object [74]. Three types of scalar functions have been used, namely Height function, Curvature function, and Geodesic distance. Since the Geodesic distance is invariant against rotation and noise, it has been used in many applications.

Hilaga *et al*. in [33] introduced a Multi-Resolution Reeb Graph (MRG) for 3D model matching. In this work, each model based on the value of the scalar function (Geodesic distance in their case) is divided into several levels. Connected components in each region constitute the nodes of the MRG graph and two nodes are linked together if their corresponding components contact each other in the model. Figure 2.16 shows a model in 3 levels of detail and their related MRG graphs. To speed up similarity measurement between the models they compare the graphs using a coarse-to-fine strategy while preserving the consistency of the graph structures.

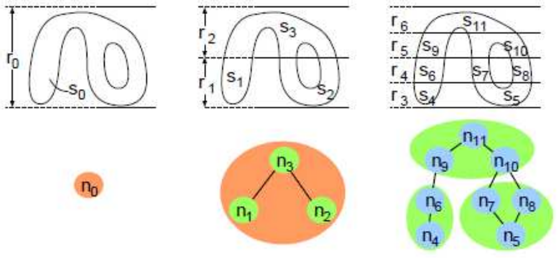


Figure 2.16, a model and the corresponding multi-resolution Reeb graph [33].

The MRG graph was later extended by Tung and Schmitt [75] to introduce the Augmented Reeb Graph (ARG) in which the topological information as well as some geometrical information such as volume, extent, curvature and orientation of the respective parts is taken into account. The retrieval accuracy of the ARG demonstrates reasonable improvement over the Multi-resolution Reeb Graph of Hilaga *et al.* [33].

The Size graph is another graph-based approach which is utilized in 3D model retrieval. After extracting the Reeb graph, Biasotti *et al*. [76] created a size graph for models using the centreline of each 3D model. Considering some mapping functions such as the distance from the centre of the bounding sphere and the integral geodetic distance, the authors extracted a centreline skeleton and measured a set of features of the corresponding region on the model. These features are used as a size graph to compute discrete size functions. They compared the models via matching between the size functions.

The Extended version of the Reeb graph (ERG) has been introduced in the work of Biasotti *et al*. [77]. The authors extracted the ERG by defining a finite set of contour levels which divide the model surface into distinct regions. The Reeb graph's scalar function in their work is identified according to the distance from the centre of mass and the protrusion function. Finally, the graph is oriented by considering the monotonicity of the scalar function used to construct the contour levels. The matching phase in this work is fulfilled via finding the largest maximal common sub-graph that minimizes the geometric and the structural differences of the two objects.

Very recently, Barra and Biasotti [78] used the Extended Reeb Graph (ERG) for the 3D retrieval purpose in which an adapted kernel is used to match each pair of Reeb graphs. They described each model by a combination of the overall shape structure (coded in a topological graph) with a local geometric description (the spherical harmonic indices of subparts). In order to check the similarity between the models a kernel is utilized which acts on bag of shortest paths defined from the graphs.

In contrast, the Directed Acyclic Graph (DAG) is used in [79] and [80] for voxelized models from two different aspects.

Zhang *et al*. from McGill University [79] utilizes the medial surface to represent 3D models and clusters the parts to form the DAG. The triangular form models are firstly voxelized before extracting the DAG via applying the topology-preserving tinning process. Each node of the DAG is associated with a set of geometric features and a signature vector that encodes topological information. Finally, the indexing and matching process of a 3-D object is guided by the graph spectra [81] for which the Eigen-decomposition of the graph's adjacency matrix is computed. The evaluation of the algorithm is done on a selection of models from the PSB dataset containing 320 models categorized into 13 classes. The retrial results demonstrate the superiority of the proposed approach compared to other approaches such as shape distribution [58] and harmonic sphere [47], especially for objects with articulating parts.

In a slightly different work, Sundar *et al*. [80] use the distance transform for thinning the models and extracting the DAG graph (Figure 2.17). Furthermore, they encode both geometrical and topological information of the models in terms of the graph and local shape descriptors which are held at each node in the graph. The proposed graph-based matching approach supports part-in-whole matching, that is, the object to be matched can be found as part of a larger object.

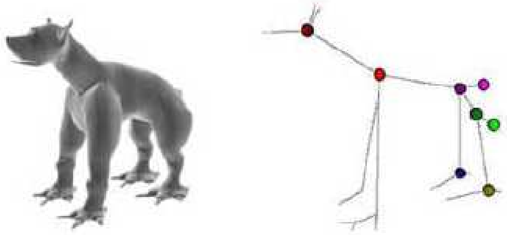


Figure 2.17, The DAG graph of a sample dog model extracted by Sundar *et al.* [80].

Another graph-based approach in the literature is a curve-skeleton which was proposed in the work of Cornea *et al.* [82]. For each voxel-based model, they use a generalized potential field [83] generated by charges placed on the surface of the model. The generalized potential at a point of the object, due to a nearby point charge, is defined as a repulsive force, pushing the point away from the charge with a strength that is inversely proportional to the distance between the point and the charge. The skeleton of the object is extracted from a vector field created by the potential field. Finally, the Earth Mover Distance (EMD) [35] is utilized to compute the dissimilarity between the models.

Although Graph-Based methods are generally very difficult to implement and graph-matching algorithms are needed to compare the models, they can represent both geometric and topological properties and are applicable for partial matching purpose. Another issue of graph-based approaches is related to their inability in supporting learning-based methods.

2.2.2.3. 3D Transform-based approaches

Similar to the 2D Transform-Based category, the goal of these methods is to work on the models in a new domain other than the spatial one. A mathematical transformation is applied on either the surface or volume of the model in order to switch to a new domain and then use the transformation coefficients as shape descriptors. The Spherical Harmonic, Fourier, Cosine and the Radial transform are the most well-known methods of this class of descriptors.

The Spherical Harmonics transform (SHT) is one of the most popular approaches in 3D model retrieval. Without losing too much shape information, SHT can reduce the descriptor size considerably. The ray-based or extent (EXT) descriptor proposed by Saupe and Vranic [84] gives the SH-transformed version of the maximal distance from the centre of mass as a function of the spherical angle. In this work, the authors try to describe each model by a function of rays on a sphere. To that end, they cast rays from the centre of mass of the model and then, for each ray, the distance from origin to the last point of intersection with the object surface is estimated. The extracted distances form a function that is called a spherical extent function. The shape descriptor is formed by a spherical harmonic transform on the extent function. Finally, the models are matched via comparing the desired number of the transform coefficients.

The idea of describing models by the spherical harmonic transform on the extent function was gradually improved by the same authors via taking samples of the spherical function at many points [85], taking in account the orientation of the surface, along with the extent vector [86] and considering a set of concentric spheres with different radii [86]. In all of the above-mentioned improvements, the accuracy of the descriptor can be managed by changing the parameters of sampling size and number of used spherical harmonic coefficients.

Kazhdan *et al*. [1, 47] proposed a rotation-invariant SHT on the voxelized model. They first voxelized the 3D model in a binary manner and then the intersections of the model with a set of concentric spheres are exploited to construct the spherical function from voxel values of each sphere. Collection of the spherical functions is employed to compute a rotation invariant descriptor by decomposing the function into its spherical harmonics and summing the harmonics within each frequency, and computing the L2-norm for each frequency component. (Figure 2.18)

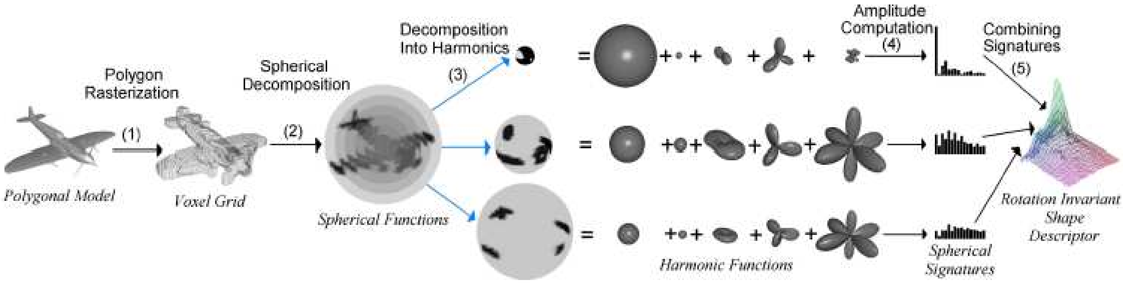


Figure 2.18, Computing the spherical harmonics shape descriptor by Kazhdan *et al*. (figure is taken from [1]).

Another approach which lies in this category is the 3D Fourier Transform (3DFT). After pose normalization by PCA, Vranic and Soupe [85] voxelized 3D models using a bounding cube (BC[1]). Then, they applied the 3DFT on voxel values to switch to the frequency domain and used the coefficients as shape descriptors.

The 3DTF is also employed by Dutagaci *et al*. [87] on voxelized models along with the 3D Radial Cosine Transform (3DRCT) by using two different voxel representations of 3D objects, namely, binary and continuous. At the expense of some information, they bypassed the PCA for pose alignment by considering the sum of magnitudes of 3DFT coefficients at the same frequency shell. Using the 3DRCT as an alternative to 3DFT enabled them to represent a 3D model with a small number of features. This led to it being easy and fast to calculate but with less power of discrimination.

The Extended 3D angular radial transform (ART) shape descriptor is another descriptor which is used for retrieving 2D colour images and 3D models. The original ART, recommended by the MPEG-7 standard [5], is only limited to binary images but Ricard *et al*. [90] extended and generalized it to work on 2D colour images as well as 3D models. Using 3D ART each 3D model is presented in spherical coordinates as the product of a radial basis function along the angular and two radial basis functions along the radial directions. The authors claimed that the 3D ART keeps properties of the original ART such as robustness to rotation, translation, noise and scaling. Furthermore, the 3D ART provides compact descriptor size which leads to short retrieval times.

Generally speaking, the transform-based approaches in 3D model retrieval have a more compact size of shape descriptor by keeping only the first few coefficients of the transformation. Also they have good power of discrimination. But the main drawback of them is that they usually have to discard the phase part of the transformation for pose normalization which leads to sacrificing some useful information of the 3D model.

2.2.2.4. Heat Kernel-based approaches

During last few years, the heat kernel-based descriptor has drawn considerable attention from leading researchers. Due to the importance of this class of descriptors, we dedicate the following section to give a brief discussion on the related retrieval approaches.

[1] The bounding cube (BC) of a 3D-model is defined to be the tightest cube in the canonical coordinate frame that encloses the model, with the centre in the origin and the edges parallel to the coordinate axes.

Suppose $\mathcal{M}$ be a compact Riemannian manifold possibly with boundary. The heat diffusion process u over $\mathcal{M}$ is governed by the heat equation in the following form:

$$\left(\Delta_{\mathcal{M}} + \frac{\partial}{\partial t}\right) u(x,t) = 0 \qquad (2.1)$$

Where $\Delta_{\mathcal{M}}$ denotes the *Laplace-Beltrami* operator, a Riemannian equivalent of the Laplacian, of $\mathcal{M}$. The fundamental solution of the heat equation is called the *heat kernel* and represented in the form of $\mathcal{K}_{\mathcal{M},t}(x,z)$ with a point heat source at x. It describes the amount of heat on the surface $\mathcal{M}$ at point x in time t.

On compact manifolds, the heat kernel can be presented as Equation 2.2 [91]:

$$\mathcal{K}_{\mathcal{M},t}(x,z) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x)\phi_i(z) \qquad (2.2)$$

Here, $\lambda_0, \lambda_1, \lambda_2, \dots > 0$ are eigenvalues and $\phi_{0,}\phi_{1,}\phi_2, \dots$ are the corresponding eigenfunctions of the Laplace-Beltrami operator satisfying $\Delta_{\mathcal{M}}\phi_i = \lambda_i \phi_i$.

Sun *et al.* [92] proposed the HKS descriptor using the diagonal of the heat kernel as the local shape descriptor. In their work, the HKS captures information about the neighbourhood of a point on a shape by recording the dissipation of heat from the point onto the rest of the shape over time (see Figure 2.19). The authors prove that a homeomorphism between two compact manifolds that preserves the diagonal of the heat kernel at each point must be an isometry. According to this later characteristic, the matching between the models is fulfilled via the local feature vectors formulated by the HKS. The proposed HKS has favourable advantages namely intrinsic (isometry-invariance), informative and multi scale which make it a popular approach compared with other approaches.

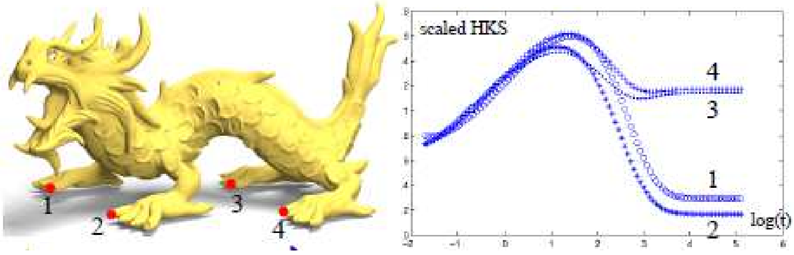


Figure 2.19, the scaled HKS at four sample points of a dragon model proposed by Sun *et al.* [92]

Ovsjanikov *et al.* [93] combine the HKS descriptor with the Bag-of-feature to construct a global shape descriptor. After computation of HKS at every point of the shape, the authors utilize the vector quantization approach to replace the HKS of every point with the index of the most similar entry in a geometric vocabulary consisting of representative heat kernel signatures or “geometric words”. Finally, the Bag of Feature paradigm counts the frequency of appearance of each geometric word to compare to the other models.

Despite all of the aforesaid valuable features, the HKS suffers from a challenging drawback of sensitivity to scale transformation. That is, when a model is globally scaled by $\mathcal{N}$ the corresponding HKS will be scaled by $\mathcal{N}^{-2}$[94].

Bronstein and Kokkinos discuss this issue and mention that model global pre-normalization (using unit bounding sphere) or the Laplace-Beltrami eigenvalues normalization are not able to solve the scale-sensitivity problem as they are not applicable to models having missing parts. They overcome this problem by local normalization of the heat kernel signature based on logarithmic scale space and magnitude of the Fourier transform. The new scale-invariant HKS (SI-HKS) is based on a logarithmically sampled scale-space in which shape scaling corresponds to a translation. This translation is undone using the magnitude of the Fourier transform. The experimental results of the SI-HKS in the Bag-of-Feature framework on the model in the *ShapeGoogle* database [93], consisting of 1061 models, show significant performance improvement over the original HKS. It also demonstrates high retrieval ability in the presence of transformations such as isometric deformations, missing data, topological noise, and global and local scaling.

Bronstein *et al*. [95] later enhanced the SI-HKS descriptor to overcome the major problem of the Bag-of-Feature (BOF) framework; it does not take into consideration the spatial information about the features. Instead of using a histogram of the features, models are described as histograms of pairs of features and the spatial relations between them (visual expressions). As shown in Figure 2.20, their Spatially-Sensitive BOF (SS-BOF) is a two-dimensional histogram in which cell $H(i,j)$ counts the frequency of co-occurrences of features i and j in the model. They also use the Term Frequency-Inverse Document Frequency (TF-IDF) weighing scheme [96] to improve the retrieval efficiency [95].

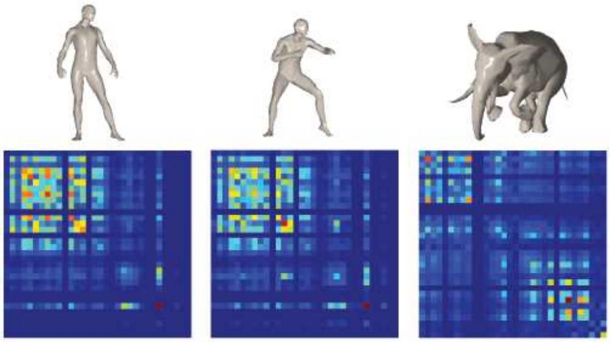


Figure 2.20, the SI-BOF histogram samples proposed by Bronstein *et al.* [95]

In a slightly different work, the closed curves along with the BOF framework is utilized for partial 3D model retrieval [97]. The process starts with extracting feature vectors using the diffusion distance in the spectral embedding space using heat kernels. Then, using the commute time distances which takes into account all paths connecting every pair of vertices on the mesh, a mapping function is defined. The function is computed using eigen-functions and eigenvalues of the *Laplace-Beltrami* operator and is employed to detect small regions which are encoded in the form of the closed curves. After defining 25 levels of closed curves for each model, the collection of all curves of all models are grouped to define the BOF framework. In the retrieval stage, the query model is described using a histogram based on the extracted closed curves. As can be observed in Figure 2.21, one of the notable advantages of the proposed closed curve over the indexed closed curves (ICC) [98] is related to robustness to the missing part degeneracy.

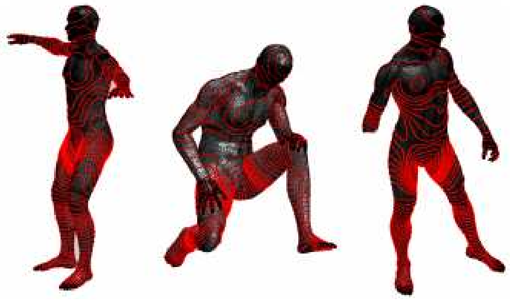


Figure 2.21, similar computed closed curves of [97] for different poses of human model, even models having missing parts.

To summarize, although this class of approach has only recently emerged in the literature, they have shown a superior retrieval quality among all other approaches. The transformation-invariance of such approaches is a valuable characteristic which guides the new trends of research toward such approaches. Furthermore, most of the techniques which lie in this category are able to be applied in the partial-matching purpose.

2.2.3. Discussion on the Shape Descriptors

As mentioned above, there are a considerable number of shape descriptors proposed in the area of model retrieval. All of them try to represent models in a good discriminative manner (Table 2.1 shows a summary of them). Although depending on the scope of the retrieval system, some shape descriptors can show better quality than the others, there is no shape descriptor that is clearly better than all of the other ones. High discriminative power, supporting partial matching, retrieving non-rigid objects, affine transformation invariance and robustness to noise distortion and simplification are among the main objectives that researchers are trying to achieve. Each class of descriptor tries to reach some of the mentioned goals. Thus, still most attention is drawn to proposing new shape descriptors with high quality of both efficiency and effectiveness. As will be detailed in Chapter 3, we propose three shape descriptors as follows: one is the SilPH descriptor which is a view-based approach (see Figure 2.1) and extracts the model descriptors using the 2D Poisson Equation. The other two descriptors, BoF-CDD and DP-CS-ECD, are fit to the histogram-based approaches of Figure 2.1 and utilizing distribution of electrical charge on the surface of the models. The proposed descriptors try to show high discriminative ability as well as robustness to various surface changes.

Table 2.1, Taxonomy of shape descriptors.

Segmentation Technique	References	Comment	Advantages	Disadvantages
View-based (Spatial domain)	15, 16, 17, 20, 21, 23, 25, 26, 28, 30, 31	A desired number of views are extracted and described via spatial-based descriptors.	-Easy to understand. -2D Image matching techniques are applicable.	-Time-consuming process of multi-view matching.
Sketch-based	1, 39, 40, 43, 45, 46	User sketches the query to be matched to the views of the target dataset.	-Easy to specify query as sometime the query model is not already available.	-Query and the target models are not of the same structure. -Retrieved models depend on the user's drawing skill.
2D Transform-based	14, 24, 43, 55, 56	The extracted views are described via 2D transform-based descriptors.	-2D Image matching techniques are applicable.	-May require pose-normalization process.
Histogram-based	58, 59, 60, 61, 62, 63, 65, 66, 67, 68, 71, 72, 73	Occurrences of extracted features are counted and shown in a histogram-based descriptor.	-Easy to understand. -Easy to implement.	-Mostly require pose-normalization phase.
Graph-based	33, 75, 76, 77, 78, 79, 80, 82,	An appropriate graph is extracted and graph-isomorphism is performed to match the models.	-Represent both geometric and topological properties -Applicable for partial matching.	-Difficult to implement.
3D Transform-based	1, 47, 84, 85, 86, 87, 90,	A 3D transformation is applied on the models and the coefficients are utilized to describe the 3D models.	-Offer more compact- size descriptors. -Good discriminative ability	-Losing some information as they discard phase part.
Heat kernel-based	92, 93, , 95, 97	The heat diffusion process is used to extract local descriptor of the models.	-High retrieval quality. -Can be applied for partial matching.	-May be sensitive to affine transformations (e.g. Original HKS).

2.3. State-of-the-art in 3D model segmentation

3D segmentation has a wide range of applications in different fields of computer science from compression [99], texture mapping [100] and model simplification [101] to content based retrieval [102]. Consequently, there are so many publications on segmentation algorithms in the literature. Thus, there is a need for grouping different ideas related to segmentation to help new researchers find their ways in this everyday growing field. This is exactly the purpose of this survey. There are a few other publications in the literature [103, 104, 105] surveying different types of mesh partitioning techniques among which [103] and [105] provide a better understanding of different types of segmentation algorithms, their different steps, and their pros and cons. However, they are quite old and don't cover the emerging group of segmentation algorithms discussed in this thesis. Furthermore, they do not include the latest achievements in the classical segmentation algorithms. Therefore, we devote this section to a survey on segmentation techniques in the literature.

The rest of this chapter is organized as follows: the next section introduces a new grouping schema for the segmentation algorithms. Based on this grouping the segmentation algorithms are divided into two classes. Therefore, the next following sections describe each of these classes.

It should be mentioned that this classification is not exhaustive; we try to give an overview of the most popular techniques and briefly discuss their advantages and drawbacks.

2.3.1. Grouping Segmentation Algorithms

Segmentation algorithms can be classified in many different ways using different factors. For example based on the number of the models to be segmented simultaneously (individual model segmentation vs. class segmentation or consistent segmentation), role of users in the process (interactive vs automatic approaches), the domains in which the features are extracted (surface-based vs volume-based approaches), segmentation goal (surface type vs part type) and etc.

The schema that is used in this chapter for grouping different segmentation algorithms is shown in Figure 2.22. This schema divides the segmentation algorithms into two different classes viz. Individual and Consistent segmentation approaches. The former one is [in turn] classified into ten sub-groups including Region growing, Watershed, Interactive, Skeleton based, Feature point based, Spectral, Learning based, Volume based, Primitive fitting based and others.

It should be noted that the aforementioned categories mainly belong to the part-type segmentation approaches. Later on in this chapter a brief discussion on surface-type approaches will be presented.

The previous surveys on mesh decomposition algorithms mainly cover the individual segmentation approaches [103, 104] and do not consider the consistent-based ones. The reason for this omission is perhaps that at the time of preparing these surveys there were not enough papers on the consistent-based methods. Nowadays, especially in the last few years, this class of segmentation algorithms has received a great interest among researchers. These methods are not covered in any previously published surveys on segmentation and are the main difference between this chapter of the thesis and those distinguished surveys [103, 104]. Furthermore, we try to mostly emphasise the recently-published papers in the literature as well as well-known older approaches. The following sections describe each of these groups of algorithms.

It is worthwhile to point out that, as some of the proposed approaches use more than one technique to perform segmentation, overlaps can be observed between different categories; that is one approach can fall into more than one category.

For the sake of clarity, from now on we interchangeably use terms "model segmentation", "model partitioning", "model decomposition" and "mesh partitioning" to refer to the same process.

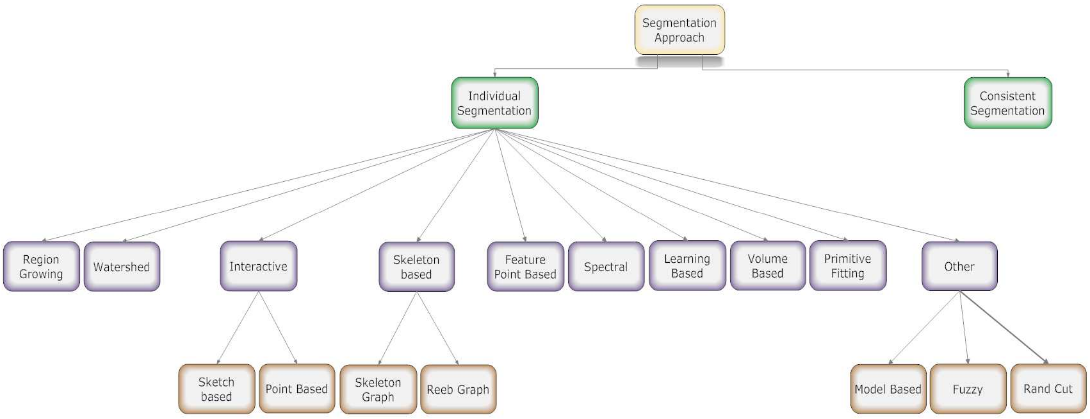


Figure 2.22, Taxonomy of 3D model segmentation approaches.

2.3.2. Individual Segmentation

The majority of mesh segmentation approaches fall into this category. This class of approach tries to find and extract semantically meaningful parts of every single model using local and/or global features of the model. Generally speaking, compared to the consistent-based methods, the individual segmentation approaches are older and comprise more mature approaches. In the following sections we present different sub-classes within this group and discuss their advantages and drawbacks.

2.3.2.1. Region Growing

The algorithms of this group are among the simplest and oldest approaches introduced in the literature. The philosophy behind all region growing algorithms is that all elements belonging to one segment are connected and similar according to some criteria. The techniques in this category try to decompose mesh elements (vertices and faces) via expansion of some pre-determined initial seeds. As illustrated in Figure 2.23, the elements are added to a segment only if a specific criterion is met. The main issues in these approaches are related to selecting the initial seeds as well as defining a criterion by which an element is assigned to a segment.

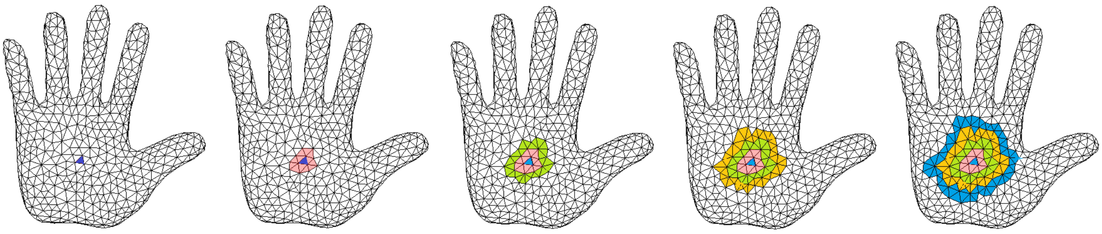


Figure 2.23, sample steps of region growing scheme applied on a hand model.

The proposed approach of Zhang *et al.* [106] consists of three major steps: curvature estimation, boundary detection and region growing. They firstly compute the Gaussian curvature for each vertex on the mesh. All of the vertices with highly negative curvature are classified as boundaries using a pre-defined threshold and the remaining vertices are considered as the seeds belonging to the potential object parts. Depending on the mesh resolution, the threshold is defined in a heuristic manner. Then, the region growing algorithm is applied on a random selection of the seed vertices to grow iteratively until they reach the boundaries. Finally, a post-processing step is applied to assign undetermined vertices to the closest regions. Also the regions possessing a few faces are merged with the adjacent ones. Despite the simplicity of the algorithm, the two pre-defined threshold parameters (high negative curvature value and minimum number of vertices of a segment) lead to a high dependency of the output quality on the choice of these parameters.

The curvedness signature used by Jagannathan and Miller is a rotation and translation invariant curvature-based shape descriptor and is computed at every vertex of the input mesh [107]. Their segmentation process begins with curvedness calculation followed by characterizing the input mesh as an attribute graph. Then, the adaptive threshold selection is utilized to determine the curvedness thresholds by which the authors define a new sub-graph extraction criterion. The introduced sub-graph is called a maximally connected attributed sub-graph (MCASG) and consists of a connected set of vertices whose curvedness values are in a desired interval, as specified by a pair of curvedness thresholds. The sub-graph extraction is done be iteratively applying a morphological process involving graph dilation and morphological filtering of outlier vertices.

The main advantage of the proposed algorithm is that the selection of curvedness thresholds is done without any user intervention, and the algorithm does not require the specification of the desired number of segments.

A random walk-guided segmentation algorithm proposed by Lai *et al*. [108] is another region growing approach in which the initial seeds are selected in accordance with user's agreement. The main ideas of the algorithm are as follows: a set of seed elements is first specified by the user. For all other faces, using an efficient process, the probability that a random walk starting at that face first reaches each particular seed is determined. The segmentation is performed by assigning the label of the seed first reached to the non-seed faces.

Instead of manually selecting the seeds, the authors generalize their segmentation approach by automatically placing the initial seeds using feature sensitive isotropic point sampling. In such a case, the number of seeds is usually more than the required number of regions and thus, following the region growing step a post-processing step of region merging based on similarities of neighbouring regions should be applied. The region-merging phase is done hierarchically based on the relative lengths of the intersections and total perimeters of adjacent segments. It will be repeated while the number of segments is greater than a user-specified number.

The main drawback of this approach, as with other seed-based approaches, relates to dependency on the initial position of the seeds so that, as shown in the Figure 2.24, different inputs always result in different cutting contours.

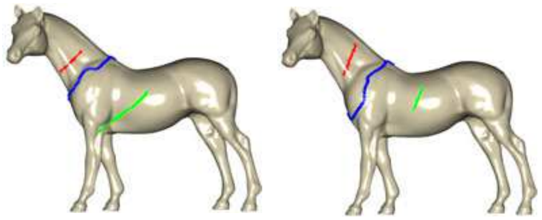


Figure 2.24, two cutting contours generated by random walk algorithm for the same model [108] (figure is taken from [109]).

Vieira and Shimada present an algorithm to segment dense, noisy meshes created from range scans of industrial objects [110]. Unlike the previous approaches, the emphasis of their work is on the criterion by which the seeds are selected. To do so, they follow a reverse paradigm. That is, after estimating noise and curvature of each vertex, the curvatures are filtered and the mesh is partitioned into regions with different shape characteristics. Instead of selecting the seeds arbitrarily, the regions are contracted to select the seed regions. They claim that this results in dealing with fewer seeds compared to the similar approaches and consequently, will result in achieving a faster segmentation process. By approximating a small neighbourhood around each vertex with a polynomial surface, they iteratively employ region growing and surface fitting to maximize the number of connected vertices approximated by a single underlying surface. Finally, the post-processing is applied so as to fill the holes emerging because of outlier noise.

The curvature tensor-based algorithm of Lavoue´ *et al*. [111] is another approach in this category. Like other approaches in this group it consists of two major steps of:

1) **Curvature based region segmentation**: after a pre-processing step of sharp edges and vertices identification, the curvature tensor is calculated for each vertex [112]. Then, depending on the principal curvature values the vertices are classified into clusters. These labelled vertices are used to assemble triangles into connected labelled regions by applying a region growing algorithm. Finally, based on size and curvature similarity, the similar regions are merged so as to specify final partitions.

2) **Boundary rectification**: in this step, the extracted boundaries are scored according to the curvature tensors and their relation to the principal curvature directions. The scores, which specify degree of correctness, are then used along with a fixed threshold to rectify the boundaries via a contour tracking algorithm.

Although this approach is specifically dedicated to segmentation of CAD models, the results on natural objects, show satisfactory results. The main drawback of this approach is related to a fixed threshold they exploit in the boundary rectification step. It may not work properly for models other than CAD ones.

It should be noted that the region growing scheme is sometimes combined with other approaches to create simpler, efficient and more accurate hybrid segmentation algorithms [110, 113, 114, 115].

2.3.2.2. Watershed

The watershed segmentation techniques have been originally exploited in 2D image segmentation [116]. The philosophy behind such approaches comes from the way water fills a geographic surface. That is, as water floods basins in a topological surface, there will be points where the flooded regions meet. These points are the watersheds and divide the surface into adjacent regions,

The analogy to this process in the model segmentation domain can be simply interpreted as follows:

Based on a pre-defined height function (e.g. surface curvature, dihedral angle ...), a scalar value is assigned to every element on the model surface. Then, the elements possessing the lowest values of the height function (local minima) are marked as "Catchment Basins". Finally, the adjacent basins are progressively grown until the neighbour basins touch each other. The dams separating the adjacent basins are called watersheds, which specify the segment boundaries. Therefore, a watershed can be described by the following definition:

Watershed: intuitively, the watershed of a function (seen as a topological surface) is composed of the locations from which a drop of water could flow towards different minima [117] (see Figure 2.25).

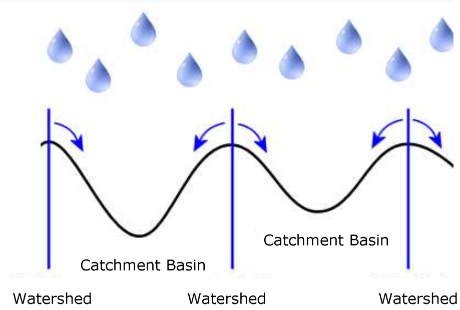


Figure 2.25, the watershed and basins in a topological surface.

The watershed segmentation technique has been extended to exploit in the 3D domain by Mangan and Whitaker [118] for the first time, where they used the total curvature for the height function as an indications of the region boundaries. Although the algorithm was only tested on a set of simple-structure models, the experiments on these models could illustrate the sensitivity of the algorithm to the exploited thresholds. To overcome the over-segmentation problem, they proposed an approach that merges the shallow region with minimum depth to one of its surrounding regions with the lowest boundary point.

The same height function to extract the boundaries has been employed by Chen *et al*. [119]. Since the curvature information is not suitable for detecting concave or convex corner vertices, the authors propose a concaveness detection algorithm based on enlargement of 1-ring neighbourhood to an eXtended Multi-Ring (XMR) neighbourhood. It will get more accurate geometric features for high-resolution meshes. The XMR concept consists of considering only the i^{th} level of neighbourhood vertices of a given vertex and ignoring all of the other adjacent vertices. After a feature extraction step, the watershed algorithm is applied to perform model partitioning into meaningful segments. The post-processing step of region-merging is finally employed to merge non-significant segments with the adjacent ones according to segment size and boundary length.

In contrast, dihedral angles (the angle between the normal vectors of the two adjacent facets) are used as the height function to perform watershed-based segmentation in the work of Zuckerberger *et al*. [115].

They investigated the ability of their approach in different applications viz. content-based retrieval, simplification and metamorphosis. Compared to the other approaches in this category, due to the small local support of dihedral angle, the probability of over-segmentation happening in the segmentation results of this approach is very low.

Another cure to avoid over-segmentation of such approaches was introduced by Page *et al*. [120] in which the normal curvature of the surface defines the height function. Following calculation of the height function and filling the basins until a certain point a 3D morphological operator [121] is applied on the minimum curvature. This leads to diminish the over-segmentation issue as shown in Figure 2.26.

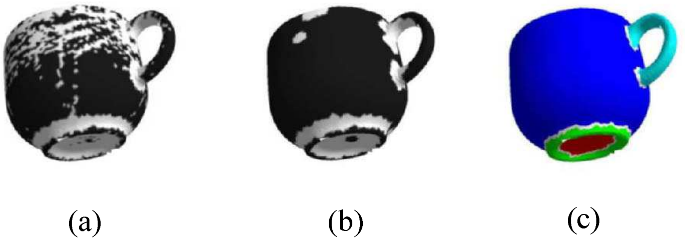


Figure 2.26, segmentation result of Page *et al*. [120] on a sample mug model. (a) is the initial threshold of curvature, (b) shows the result after applying the morphological operator and (c) is the final segmentation result.

An analogy to watershed approaches has been proposed in the spectral-based work of Benjamin *et al*. [122]. From the watershed point of view, the flow of water is analogous to heat flow in which the heat tends to accumulates in the heat walk-based accumulator basins. The authors claim that the available algorithms based on heat diffusion do not use the entire information that can be inferred from heat diffusion and therefore, they are sensitive to noise and perturbations. To overcome this problem and exploit the full information gained from the heat diffusion, they proposed a two-stage algorithm as follows:

1) Heat Flow mapping: using Laplace Beltrami Operation (LBO) this step generates a heat distribution function on the mesh surface (Figure 2.27 (b)).

2) Region identification: in this stage firstly, based on the distribution of heat, the accumulators (points which are very slow at dissipating heat compared to the rate at which they receive it) are classified and then using relative entropy the dissipaters (points in the flatter regions which tend to dissipate heat faster than they receive it) are identified. This is done depending on how close each point is to the uniform distribution of heat (Figure 2.27 (c) and (d)).

The experimental results support the author's claim about resistance of the proposed approach to any kind of perturbations such as: random noise, shot noise, short circuit, tessellation, scale and missing parts.

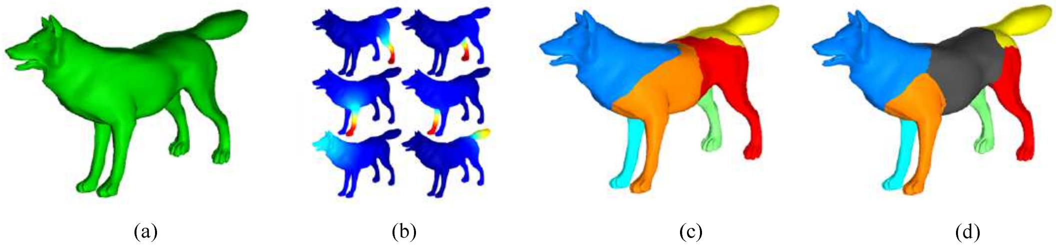


Figure 2.27, the segmentation steps of Benjamin *et al*. [122]. (b) is the heat kernel of the model using LBO, (c) and (d) are accumulator and dissipater regions identification, respectively.

In fact, the watershed approaches can be seen as a region growing algorithm with multiple initial seeds [103]. The initial seeds are the catchment basins which are defined according to a height function. Consequently, the ability of such approaches is highly dependent to the definition of the height function. That is, the improper height function may lead to an over-segmentation issue especially for the high-resolution complex models. Additionally, since this class of approach uses local features, they usually suffer from lack of tolerance to surface noises.

2.3.2.3. Interactive

The motivation for developing this class of approach stems from the fact that: due to complexity of available 3D models, automatic segmentation of such models remains a challenging issue in the computer graphic domain. Therefore, the user is asked to intervene in segmentation process to achieve more satisfactory results. These kinds of approaches usually provide users with an interface to communicate with the segmentation system via drawing some sketches or points on the desired parts of meshes so as to specify the rough approximation of boundary part location.

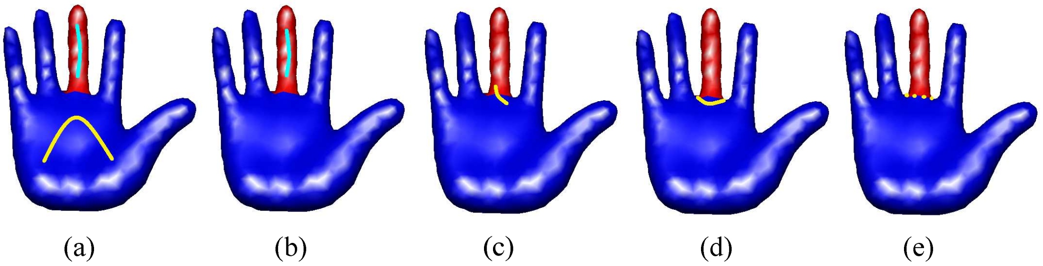


Figure 2.28 (a-e), samples for different interactive segmentation algorithms.

The interactive segmentation approaches can be roughly categorized into two main groups: sketch-based and point-based interactive approaches.

2.3.2.3.1. Sketch-based interactive segmentations

Recently, several works have been introduced for sketch-based mesh segmentation [118, 124, 125, 126]. Zhang *et al.* [125] introduced an interface called cross-boundary brush by which users are able to draw a sketch across the desired cutting boundaries (see Figure 2.28(c)). In most cases their segmentation system tries to minimize the user inputs by requiring only a single stroke to execute a cut. The strokes are drawn via a tool called part-brush or patch-brush. It uses a harmonic which is generated by the Poisson equation with constraints defined by the end points of the brush across the cutting boundary. Then, a multi-scale iso-line selection scheme is designed to select the best iso-line automatically as the final cutting boundary. It is notable that, if necessary, the user is asked to sketch additional strokes to refine a cut locally.

Conversely, the segmentation systems devised by [126] and [127] communicate with users by asking them to provide initial labels for some vertices as belonging to the desired part to be segmented (foreground) or to the rest of the model (background) (Figure 2.28(a)).

The first foreground/background sketch-based user interface for mesh segmentation was introduced by Ji *et al*. [126]. Their Easy Mesh Cutting (EMC) is based on a feature-aware isophotic metric, which employs a simple region-growing scheme to grow both background and foreground region seeds incrementally. All of the unlabelled vertices are iteratively assigned to the foreground or background regions based on the minimum distance to the regions, using the isophotic metric.

Similarly, Meng *et al.* [127] proposed a harmonic field-based method (HFM) for foreground/background interactive mesh segmentation. It starts with the user's strokes, specifying a foreground and a background seed set via solving a Poisson equation. The harmonic field is smooth and can be viewed as a smooth interpolation between the equation's constraints. The method then modifies the harmonic field to reflect the geometric features of the mesh. The graph-cut technique is then applied to produce the segmentation results that are consistent with the user's intention and the surface features.

A slightly different approach is considered as foreground sketch-based segmentation. As shown in Figure 2.28(b), unlike previous approaches, the users are only allowed to draw strokes on the foreground region and no need to paint over the background part or the boundary. A well-known Paint Mesh Cutting (PMC) approach proposed by Fan *et al*. [128] is a progressive painting-based tool provides an intuitive user interface where users cut out parts by directly painting the region of interest with a brush. Using the PMC system, users are able to continuously drag the mouse to expand the region of interest, until they are satisfied. Then, based on a Gaussian mixture models (GMM) on the shape diameter function (SDF) [129] metric of the shape, they align the cutting results with the part boundary.

The last group of sketch-based segmentation approaches are considered as close-boundary approaches. In these kinds of approaches the segmentation system asks the user to draw rough strokes close to the cutting boundary as shown in Figure 2.28(d).

[130, 131]. After rough cut selection by the users Funkhouser *et al*. [130] firstly assigned the edges a concavity weight based on the dihedral angle of edges to locate the precise boundaries. The concavity weight is equal to 1 for convex and is a low positive number for concave edges. Finally, the graph cut algorithm is applied on the dual graph of the model to locate the boundaries.

It is worthwhile to note that the generated boundary may not respect to the user's attention if the stroke is too short or if the back of model is too complex [132].

2.3.2.3.2. Point-based interactive segmentations

In a different aspect, during the early works on interactive segmentation, the users interact with the segmentation interface via specifying a few points on the desired cutting contour. These points are later employed so as to locating the cutting area by finding the shortest path between them [133, 134].

Wong *et al*. [133] introduced a tool which uses the concept of Intelligent Scissors for volume cutting along with interactive definitions of a cut contour onto the object's surface. After selecting some points on the surface, the system aims at computing the cutting surface by a dynamic algorithm of Dijkstra.

The point-based approach also has been utilized by Zockler *et al*. [134] to perform metamorphosis, which is a complex transformation between shapes via reduction to transformations between sub-patches.

Very recently, Zheng *et al.* [135] introduced an interactive segmentation framework using only one user-clicked point. The region near the clicked point is considered as a search space to locate the accurate boundary. The search space is determined by defining a set of harmonic fields that propagate in different directions and sample the isolines that pass through the clicked point as candidate cuts. Rather than superior segmentation quality, the simplicity of use is the great advantage of this approach compared to the other interactive ones.

The well-known benchmark proposed by Chen *et al*. [136] follows the point-based approaches in order to interact with users during their manual segmentation process.

There are some other interactive approaches in which users have trivial contribution such as selecting number of segments [137, 138], selecting representative points of each segment [139, 140] and *etc*.

2.3.2.4. Skeleton-based approaches

The segmentation methods in this category try to convert the mesh segmentation problem into a skeleton graph partitioning one. These approaches mainly comprise two main phases: skeleton graph extraction and graph partitioning. Depending on the graph types these approaches construct, they can be divided into two main groups: skeleton graph and Reeb graph-based approaches.

2.3.2.4.1. Skeleton graph-based approaches

The interactive segmentation approach proposed by Li *et al*. [141] uses a *skeletal tree* to extract mesh segments. They firstly extract the skeleton of the mesh via performing simplification of the surface using the edge contraction method. If the simplification process generates a disjoint tree, the *skeletal tree* is completed by inserting some *virtual* edges*.* Then, a set of critical points are identified from which the branches are cut. The critical points are extracted by sweeping a plane perpendicular to the skeleton branches. Using this scheme, the segmentation is defined implicitly by the creation of cuts. As the author investigated, the proposed approach showed good ability on collision detection problems.

The Medial surface which is a 3D counterpart of Medial Axis Transform (MAT) is used in the skeleton-based segmentation approach of Mademlis *et al*. [142]. After voxelizing the input mesh model the medial surface is extracted via thinning the model according to the Hamilton-Jacobbi equations presented in [143]. Next, the medial surface is segmented based on the connectivity of the neighbouring medial surface and background voxels. Then, the medial surface segments are readjusted based on the segment size and' degree of the node so that the resulting segments correspond to larger but more meaningful parts of the 3-D object. Figure 2.29 shows these steps for a simple dog model. The authors employ the segmentation results for content-based matching of 3D models in both partial and global retrieval.

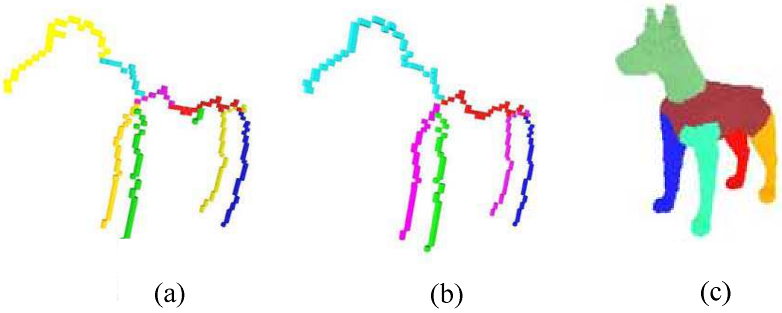


Figure 2.29, segmentation steps of segmentation approach of [142]. (a) is the segmented medial surface, (b) is the readjusted segments and (c) is the resulted segments.

The interesting work of Au *et al*. [144] uses a mesh contraction approach to construct the skeleton graph of the model. In their work the mesh contraction process is formulated as an energy minimization problem which aims at smoothing the object surface iteratively into an approximate zero-volume degenerate mesh via an implicit Laplacian with global positional constraints. As shown in Figure 2.30, the contraction leads to removal of the details of the mesh model so that a zero-volume mesh remains, that is convertible to the skeleton graph of the models. The converting step is done via a process they called "connectivity surgery" to remove all the collapsed faces from the degenerate mesh through a sequence of edge collapse operations. The authors employed the induced skeleton graph for mesh segmentation and skinning animation. To perform mesh segmentation the skeleton graph is combined with the thickness information of the models. The algorithm identifies a cutting node in each branch of the skeleton starting from the thickest branch. For each cutting node we search for segmentation boundary in a near boundary search region using the minimum-cut algorithm [145].

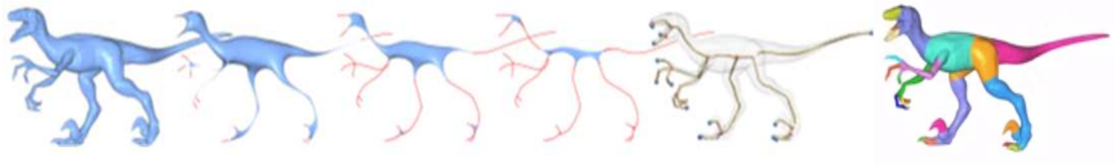


Figure 2.30, three smoothing iterations of skeleton extraction along with segmentation result of Au *et al*. [144]

A voxelized model segmentation approach of Reniers and A. Telea [140] is another approach which exploits skeletons of models to achieve meaningful partitions of the models. This hierarchical approach begins by computing the curve skeleton for the input model using a geodesic-based paradigm. The skeleton comprises branches which are associated with the components and the junctions which reflect the relationship among components. Then, a curve skeleton is constructed from the extracted skeleton by considering the shortest geodesic curves between automatically or manually-selected feature points. Using the junction points, a skeleton-to-boundary mapping is defined to divide the object surface into a set of components. This component set is finally utilized to perform hierarchical segmentation by defining a set of all components, which are generated by the junction points and their size is greater than a pre-defined threshold. The authors show that their approach is quite robust to noise and pose deformation.

2.3.2.4.2. Reeb Graph-based approaches

A Reeb graph is a topology-driven graph which was introduced in 1946 by Georges Reeb [74]. It describes the topological structure of the models using a scalar continuous function on a 3D object such as curvature function, geodesic function etc.

For a 3D mesh $\mathcal{M}$, the Reeb graph is constructed with respect to a continuous real function $\mathcal{F}$ on $\mathcal{M}$. The graph is a quotient space of $\mathcal{F}$ in $\mathcal{M} * \mathcal{R}$ generated by the equivalence relation. According to this relation, the equivalence relation for two points $v1$ and $v2$ holds if and only if both of the following conditions are fulfilled:

1) $\mathcal{F}(v1) = \mathcal{F}\ (v2)$
2) $v1$ and $v2$ belong to the same connected component of $\mathcal{F}^{-1}(F(v1))$

Figure 2.31 depicts a sample model and its associated Reeb graph.

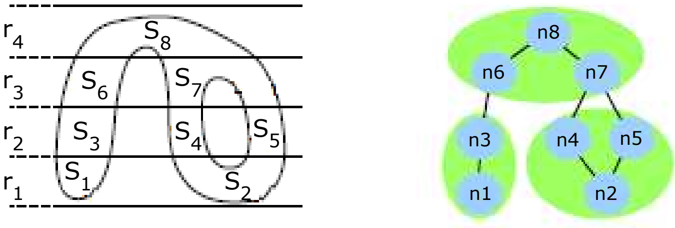


Figure 2.31, a sample model in left and its related Reeb graph in right (figure is taken from [33]).

The Reeb graph was exploited in the 3D retrieval domain as a pose- invariant shape descriptor for the first time. [33]. Hilaga *et al.* defined function $\mathcal{F}$ based on the Protrusion function or average geodesic distance between vertices of the model surface.

This idea has been used in mesh partitioning by Velette *et al.* [146] where they introduce an evolution process called "Protrusion Conquest". The process considers both the computed protrusion and the connectivity of the models. In order to extract the protruded parts of the mesh and separate them from the main body, they use the evolution of the protrusion function from its high towards its low values.

The Reeb graph-based approach of Tierny *et al.* [147] is based on the key idea that the topology of a feature is a more important decomposition criterion than its geometry. Their hierarchical algorithm which uses an enhanced topological skeleton [148] to delimit the object core and to identify the junction surfaces can be summarized as follows: the feature points on the mesh are firstly extracted according to two geodesic functions. Then, based on the geodesic distance to the closest feature point, a mapping function is defined for each vertex. This mapping function is used to construct a Reeb graph in which the nodes are classified into three main groups of Extremity (green nodes), tubular (blue nodes) and junction (red nodes), based on their degree of connection (see Figure 2.32(b)). Finally, by modelling components of the Reeb graph with an ordered collection of closed curves as well as estimating the curvatures of the curves the models are partitioned to its meaningful parts. Merging the adjacent components leads to achieve a hierarchical segmentation as shown in Figure 2.32 (c) and 2.32 (d). The experimental results verify the robustness of the algorithm against simplification, noise and deformation.

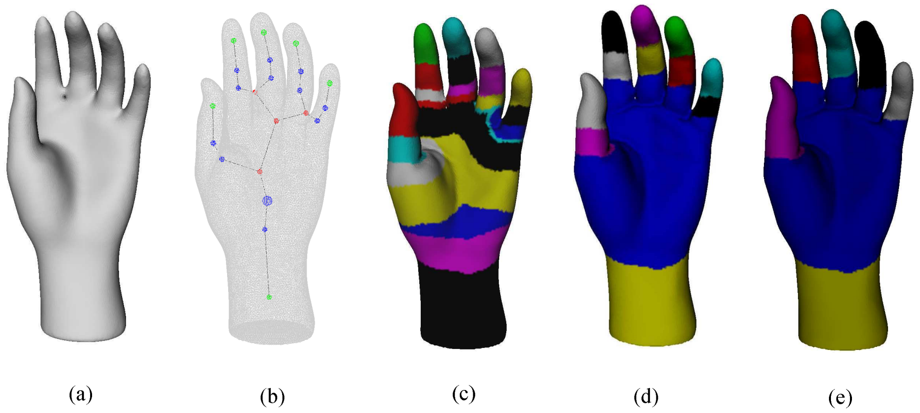


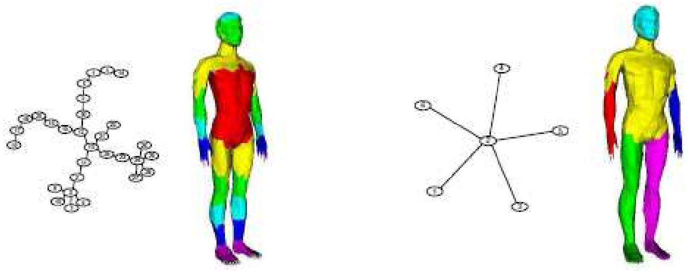
Figure 2.32, the segmentation results of Tierny *et al.* [147]; (a) input model, (b) extracted graph, (c) initial segmentation, (d) fine segmentation and (e) Coarse segmentation.

The protrusion function along with Reeb graph has been also exploited by Berretti *et al.* to perform mesh partitioning [149]. The proposed approach involves two steps as follows:

i. Reeb graph Construction: in this step, the topological and metric properties of the model are used to construct the graph.
ii. Refinement: in this step, the curvature information is exploited to refine object decomposition and adjust region boundaries so as to match deep surface concavities and to yield perceptually salient decomposition of objects.

It is important to note that the quantized values of the protrusion function lead to an over-segmentation problem of the models. To overcome this issue, the authors managed to merge regions by simplifying the Reeb Graph, i.e. Graph nodes are merged together based on their topological and mean curvature information. (See Figure 2.33).

This approach was later used for computing the similarity among 3D models based on their part resemblance. [150].



(a) (b)

Figure 2.33, segmentation result of [149], (a) is the initial constructed Reeb graph and associated segmentation result and (b) shows the result after graph simplification.

2.3.2.5. Feature Point-based approaches

Katz *et al*. [151] proposed a hierarchical algorithm based on feature points extraction which comprises the following steps:

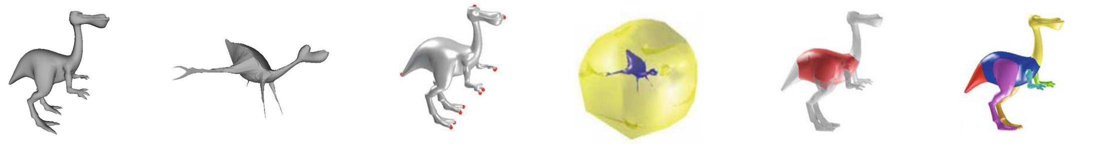
a) Transforming the target model into pose-invariant representation via computing the canonical form of the model using Multi-Dimensional Scaling [152] (Figure 2.34(b)).

b) Extracting a few feature points on the canonical form representation of the model and mapping them back into their corresponding points on the original model (Figure 2.34(c)). These feature points are located using a geodesic-based criterion.

c) Extracting the core part of the model using a new spherical mirroring operation as shown in Figure 2.34(d) and 2.34(e).

d) Computing and extracting the components attached to the core part, each representing at least one feature point (Figure 2.34(f)).

Finally, the segment boundaries are enhanced by applying a post-processing step of boundary refinement. The algorithm partitions segments hierarchically, stopping automatically when the current segment has no feature points or the number of vertices in the current segment is lower than a threshold.



(a) Input Model (b) Canonical Form (c) Feature Points (d) Spherical Mirroring (e) Extracted Core (f) Segmentation Result

Figure 2.34, the segmentation steps of the proposed algorithm of Katz *et al*. [151]

A protrusion-oriented segmentation approach introduced by Agathos *et al.* [153] exploits geodesic distance-based method to extract the feature points. They try to decompose models into a core and protruded parts attached to the core. Given a mesh, each vertex v is assigned a number computed by a protrusion function $pf(v)$ which is defined by the following equation.

$$pf(v) = \int_{p \in S} g(v, p) dS \tag{2.3}$$

Where, $g(v, p)$ is a geodesic distance between face v and all of the vertices p on the mesh surface S.

The protrusion function can be interpreted as the sum of the geodesic distances of the vertex to all of the vertices on the mesh. If the protrusion function related to a vertex is greater than a specific threshold, the vertex is marked as a feature point (the threshold is defined as the average of geodesic distances between each pair of vertices). The extracted feature points are merged together based on their geodesic distance to avoid over-segmentation. To extract the core part of the model, they use the minimum cost paths between the feature points. They utilize the fact that the minimum cost paths cover a significant area of each protruded part. So, expanding a set of vertices by this fact guarantees that it will reach the protruded parts and the region between them (the core part).

Since the core extraction scenario they follow does not guarantee that the core overlaps exactly the partition boundaries, similar to Katz *et al.* [151], they apply a post-processing step on the extracted parts to detect and refine the boundaries via the minimum-cut algorithm [145].

It is worthwhile to note that the proposed algorithm is not applicable in a recursive manner. Therefore, it has considerable limitations in decomposing models with mixed core parts[2]. That is, the algorithm is only able to decompose models with a star-like topology of connected components. Although, the authors investigate the effect of different parameters on their segmentation results, they have not shown any quantitative comparison to evaluate their work.

The feature point-guided algorithm proposed by Lin *et al.* [154] utilizes geodesic zones which are called locales. These locales are created by the extracted salient feature points and are used to define a border function which identifies those locales containing the

[2] *A mixed core (multipart core) is a detected core part of the model (by segmentation algorithm) which can be decomposed into simpler parts by human perception system.*

boundary of the protrusion. In their work, the faces of the core part are defined as the points of the mesh whose protrusion function values are lower than a fixed pre-defined threshold. Using a fixed threshold is one of the main drawbacks of their work, which may lead to over/under segmentation of the main body. On the other hand, strength of the current work is the ability to decompose noisy models into meaningful semantic parts.

As with all of aforementioned approaches in this category, the cut refinement approach has been applied on their extracted boundaries to generate segments having smoother borders.

2.3.2.6. Spectral approaches

The philosophy behind this class of approach is to convert the mesh segmentation problem into graph partitioning based on spectral graph theory [155]. Since the optimization graph partition problem is NP hard [99], the spectral methods are able to solve a relaxation of this problem by computing a few leading eigenvectors of an affinity or weighted graph Laplacian matrix. Those eigenvectors provide a new low dimensional embedding for which the clustering problem is more easily solved.

Most spectral-based approaches follow the following pattern:

- A square matrix $\mathcal{M}$ to represent a discrete operation on the input mesh is defined. This matrix can be interpreted as pairwise relation between mesh elements such as vertices or faces.
- The eigenvalues and eigenvectors of the matrix $\mathcal{M}$ are computed via an eigen-decomposition approach.
- The extracted eigenvalues and eigenvectors are utilized to solving the related problem.

The main effort on spectral mesh segmentation approach has been made by Liu and Zheng [156]. The authors employed the probability that two facets can be grouped in the same segment as the discrete function so as to generate the matrix $\mathcal{M}$. The spectral analysis of the matrix is later utilized to perform mesh segmentation by using some of its largest eigenvalues and eigenvectors.

This work has been expanded by the same authors to enhance the segmentation result [157]. Contour analysis along with spectral embedding has provided a tool to extract accurate boundaries (Figure 2.35). They firstly projected the 3D mesh models into a 2D plane using two spectral embeddings, e.g. Laplacian matrix to enhance the structural characteristics of the mesh (protruded parts) and the minimum curvature embedding of the vertices to enhance the geometrical characteristics of the mesh (concavities). Followed by contour extraction from the 2D projected image a decision about the possibility of cutting the image into two sections is made (based on the concavity criterion). Then, a sequence of facets is computed on each side of decomposable parts and finally, a one dimensional embedding is computed to perform a linear search over the sequence of facets to cut out the model into two sections.

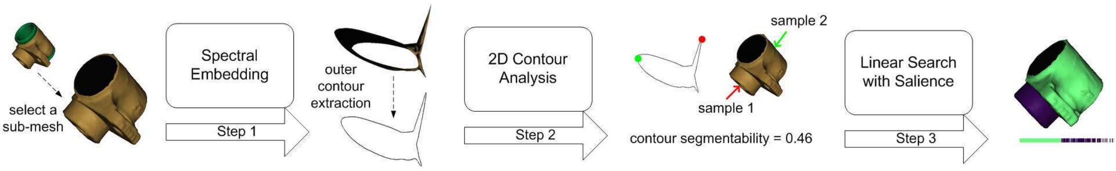


Figure 2.35: the pipeline used in segmentation work of Liu and Zheng [157]

The weighted Laplacian matrix constructed in the work of Lei *et al*. [158] uses the Shape Diameter Function [129] along with dihedral angles to formulate geometrical information of the shapes. After creating the matrix, they embed it in the spectral domain via eigendecomposition and finally, a multi-way normalized cut is employed to perform mesh segmentation by optimal partitioning of the Laplacian matrix. A min-cut methodology [145] is later on used to smooth the extracted boundaries. Although, they evaluate their results on Princeton Shape Benchmark [159] they show no evidence to support their claim about consistent segmentation ability of the proposed approach (see Section 2.3.3 of this chapter).

To the best of our knowledge, the very recent work of Chahhou *et al*. [160] is the last proposed approach in this category. Their hierarchical approach exploits both the minima rule and spectral clustering. The general form of spectral clustering (p-spectral) used in their work, is based on the non-linear operator of graph-Laplacian. Due to inability of spectral clustering and the standard adjacency matrix in cutting the parts placed on the regions of minima curvature, the authors defined new adjacency matrices to encode the concavity of the mesh using the minima rule and examined their capability in segmentation. The experimental experiments on the SHREC 2007 dataset have shown the higher segmentation quality of the proposed approach compared to other available ones.

The main drawback of spectral-based approaches is related to their high dependency on the definition of the Laplacian matrix, so that the inappropriate definition of the matrix can result in inaccurate extracted segments.

2.3.2.7. Learning based approaches

Most of the available segmentation approaches exploit geometrical or topological information about the models. This suffers from sensitivity to local features and/or poses changes. Furthermore, these approaches only use a single rule to perform segmentation and are not able to combine different criteria to achieve higher segmentation accuracy.

The aforementioned limitations along with the availability of ground-truth datasets for segmentation [36, 161, 162] have led to the emergence of a new class of segmentation algorithms based on learning criteria. This kind of approach employs a set of prior manually-segmented models to obtain more accurate results. Such algorithms generally comprise two main steps: an off-line step in which an objective function is learnt from a set of manually-segmented mesh models and an on-line step that uses the learnt function to decompose the input mesh models.

The supervised learning-based method proposed by Kalogerakis *et al*. [163] simultaneously segments and labels the models using a set of pre-analyzed training models. In their work, an objective function is formulated as optimization of Conditional Random Field (CRF) model with terms assessing the consistency of faces with labels and terms between labels of neighbouring faces.

Although, their approach outperforms other existing segmentation algorithms, it is worth mentioning that the requirement of training data and several features utilized in this approach leads to a slowdown of the segmentation process, a couple of minutes for a medium size model of about 5K faces.

Recently, Lv *et al*. have used a similar CRF objective function in a semi-supervised learning framework [164]. Since the CRF optimization is not applicable in a semi-supervised framework, the authors tried to introduce a new CRF model whose whole objective function is learned from function templates and can be optimized in a semi-supervised manner. Their proposed approach incorporates knowledge imparted by both segmented, labelled meshes, and un-segmented, un-labelled meshes. By adding an un-labelled conditional entropy into the objective function their framework is able to capture the information from the un-labelled meshes. The experimental results obtained on the Princeton Shape Benchmark [159] have shown almost 90% of segmentation accuracy, which is slightly better than the supervised approach of Kalogerakis *et al*. [164]

Similarly, an unsupervised, fully automatic 3D mesh segmentation algorithm based on boundary edge learning has been introduced by Benhabiles *et al*. [165]. The AdaBoost classifier [166] is utilized as a machine learning algorithm to automatically select the most relevant geometric features to detect candidate boundary edges. And, in addition to the AdaBoost classifier, several geometric features such as Dihedral angle, Gaussian Curvature, Global Curvature and Shape Diameter Function are utilized to create a 33-dimensional feature vector to characterize edges. The main difference between their proposed algorithm [165] and Kalogerakis's work [164] is related to the way that the boundaries are defined. That is, instead of determining the proper label for each face and then implicitly defining the segmentation result from these labels, we explicitly determine the boundary edges that allow obtaining smooth closed contours for defining the segments. Comparing the segmentation results generated by both approaches reveals that the proposed algorithm of Benhabiles *et al*. has shown a big enhancement in the segmentation quality than that of Kalogerakis *et al*. [164], so that the Rand Index error obtained by Kalogerakis *et al*. is 9.5% while this factor is 8.8% for the Benhabiles algorithm over the Princeton Shape Benchmark dataset.

2.3.2.8. Volume-based Approaches

Instead of using surface or skeleton features to perform model segmentation, the volume attributes are utilized for surface partitioning [67, 129].

Shapira *et al*. [129] claim that the surface attributes are often pose and topology-dependent and therefore, the resulting segments may vary as the model poses change. So, they propose a different volumetric feature called Shape Diameter Function (SDF) to fulfil segmentation. They believe that the only feature that remains invariant against pose and even topology changes is the volume of the shape.

As shown in Figure 2.36, their hierarchical approach exploits a volume-based scalar function (SDF), which can be interpreted as a volumetric counterpart of the Medial Axis Transform [168]. It measures the local diameter of the object at the surface points by sampling the rays fired from the surface point inward to the other side of the mesh in a cone-shaped region and averaging the length of those rays sampled. Finally, fitting *k* Gaussian function to the histograms of SDF values allows a hierarchical strategy for extracting mesh parts. The SDF signature is eventually employed in skeleton extraction as well as matching and retrieval purposes in [102].

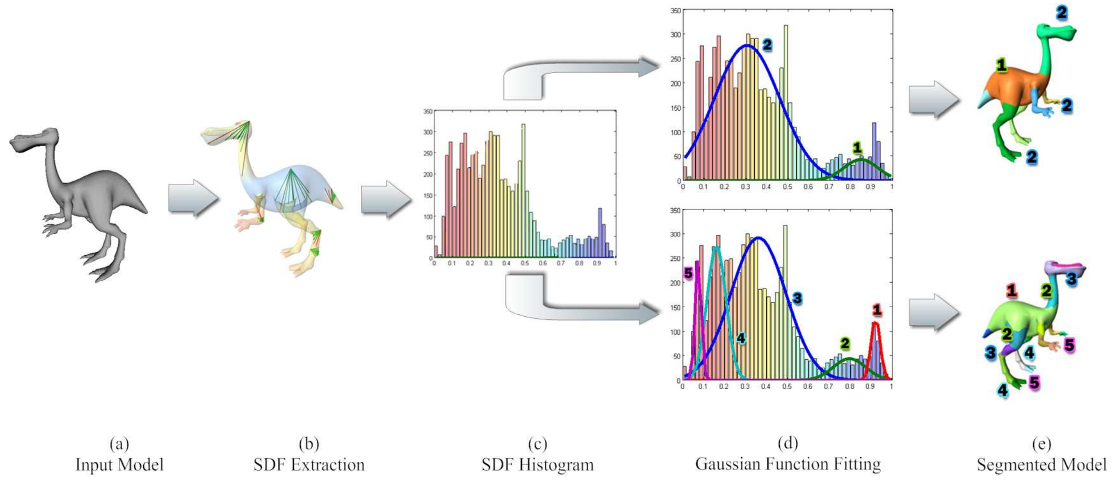

(a) Input Model
(b) SDF Extraction
(c) SDF Histogram
(d) Gaussian Function Fitting
(e) Segmented Model

Figure 2.36, Volume-based segmentation approach using SDF (Image is taken from [129])

Another volume-based partitioning approach on voxelized models was proposed by Kim *et al*. [167]. In order to extract the convex parts of the models, they applied the opening morphological operations using a ball-shaped structuring element. Their recursive approach starts with extracting body and branch classes of model parts via applying the opening operation. This process is recursively applied on the extracted parts while the extracted parts can be segmented into meaningful parts. Finally, the adjacent parts possessing the same convexity are merged together to avoid over-segmentation.

Despite generating acceptable segmentation results, they have not compared their results to the other available approaches quantitatively. The main drawback of this approach is related to its limitation in supported model types, i.e. it only is able to decompose models presented by voxel cubes.

2.3.2.9. Primitive Fitting-Based Approaches

The approaches falling into this category are aimed at selecting primitives from a user-defined set to extract the best matches which minimize the fitting error.

Another set of segmentation approaches is based on fitting primitives on the surface of models. Attene *et al*. [169] proposed a hierarchical algorithm, in which some pre-defined primitives such as sphere, cylinder and plane are fitted to the model to extract model segments. Firstly, they assign every face to a separate segment. Then, the faces in every pair of adjacent segments are utilized to approximate a best fitting geometric primitive. The algorithm is repeated recursively until a user-specified number of segments has been reached.

They employ the L^2 distance function to compute the fitting error between extracted segments and the fitted primitives. Figure 2.37 shows some examples of fitting cylinders into three sample models, Horse, Human and Hand. For the Horse model, two levels of resolution have been shown in their hierarchical approach.

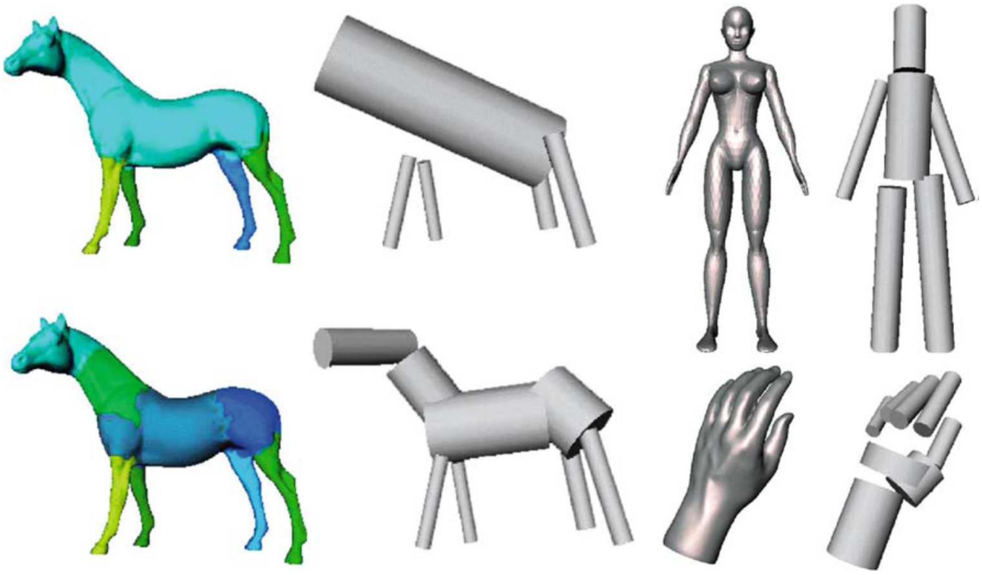


Figure 2.37, some examples of fitting primitive approach proposed by Attene *et al*. [169].

The very recent iterative work of Fayolle and Pasko [170] is applicable to point cloud meshes and employs a wide range of primitives such as sphere, cylinder, plane, cone, torus, and super-ellipsoid to extract high quality segments. The proposed algorithm comprises two main steps which are repeated after a de-noising step in the pre-processing phase:

- **Step 1**: the parameters of the primitives are optimized to fit a sub-part of the input point-set model. Then the best primitive is selected by comparing the fitted primitives and their related sub-part. The selected primitive is classified as a potential primitive candidate.
- **Step 2**: the points of the model located within a band around the surface of the corresponding potential primitive candidate are extracted.

These two steps are iterated until the size of the point-set is sufficiently small or the maximum number of iterations has been reached. As a post-processing step of the proposed approach, the unidentified points from the original point-set are assigned a primitive type and a label by iterating through the list of best primitives identified in the first step of the loop.

The authors claim that their algorithm is relatively general and can work with any type of parameterized primitives combined with transformations like tapering or twisting.

2.3.2.10. Other approaches

There are some other approaches proposed for 3D model segmentation which cannot be meaningfully assigned to the aforesaid categories. In the sequel, we discuss some of these approaches and expose their pros and cons.

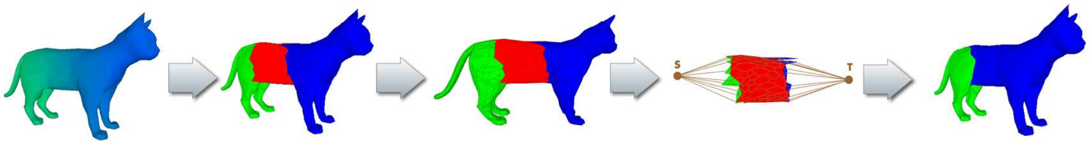
2.3.2.10.1 Model-based approaches

Wu and Levine [171] employ the charge density distribution to perform mesh decomposition. In their work, based on the minima rule, the boundary parts of models are located at the concave areas having local minimum charge density. They defined a concave face as a face possessing charge density lower than a predefined threshold as well as lower than all of its neighbours (the threshold in their work was: 1.5*lowest charge density on the model surface). To specify boundary parts they tried to locate a ring region of faces with the lowest charge density compared to their neighbours. Although their approach is applicable for some parts of specific models, it suffers from two main challenging issues; firstly, the entire boundary must lie in a concavity to be detected accurately, which is not the case for a lot of models. And secondly, for some boundary parts, there is no face meeting their requirement to be considered as a concave face.

Our proposed segmentation algorithm is based on the distribution of charge density on the surface of models too. But, our method differs from Wu and Levine's work in various aspects; unlike their algorithm, our approach is able to detect boundary parts not fully located in concave regions. Moreover, our approach can handle mesh models having a more complex arrangement. The pose-insensitive attribute of the proposed work allows the algorithm to generate similar segments for models in different poses. Finally, an extra post-processing step attached to our segmentation pipeline enables us to decompose those models having a mixed (multi-part) core structure.

2.3.2.10.2 Fuzzy-based approaches

To the best of our knowledge, the only hierarchical fuzzy approach for model segmentation was proposed by Katz *et al.* [172]. The key idea of the algorithm is to find key components via both geodesic distance and convexity using angular distance. To find fuzzy components, the authors relax the condition that every face should belong to exactly one patch for allowing fuzzy membership. Their algorithm proceeds from coarse-to-fine thanks to a binary tree of which each node corresponds to a segment. The binary tree is constructed by classification of the faces into two clusters. The likelihood of a face to belong in one of the two clusters will determine the membership of the face to each cluster. As shown in Figure 2.38, to perform hard clustering, the authors construct three parts from which two parts contain the faces belonging to only one cluster and the third part is a fuzzy part which contains the faces that are not certain to belong in a specific cluster. Finally, to compute the proper parts of faces in the fuzzy part, a minimum graph cut algorithm [145] is applied on the dual graph of that part. Based on the concavity, the fuzzy faces are assigned to one of the either sides (Green or Blue parts in Figure 2.38). The above procedure on each segment is recursively repeated until a desired condition is not satisfied anymore.



(a)
Membership Probability

(b)
Fuzzy Segmentation

(c)
Extracted Three Parts

(d)
Min-Cut on Dual Graph

(e)
Decomposition

Figure 2.38, the segmentation steps for Fuzzy-based approach proposed by Katz *et al.* [172]

2.3.2.10.3 Randomized Cut

The basic idea of the random-walk approach proposed by Golovinskiy and Funkhouser [138] is to characterize how and where a surface mesh is most likely to be cut by a segmentation approach into parts.

This approach uses a vast amount of previous work on randomized cuts for graph partitioning theory to find a minimum cut of a graph. It tries to generate a large set of randomized cut by combining several existing segmentation methods (K-Means, Hierarchical and Min-Cut) to define a "partitioning function" which measures the likelihood of each edge lying on a segmentation boundary in the randomized set. Using the function, the most consistent boundary cuts obtained by different automatic segmentation methods are identified and applied for some of their desired applications such as deformation, surface correspondence, visualization and segmentation. The experimental results on different models demonstrate the robustness of the partitioning function to the large set of transformations viz. noise, tessellation, pose and intra-class variation. Figure 2.39 shows the segmentation result on a sample Bunny model.

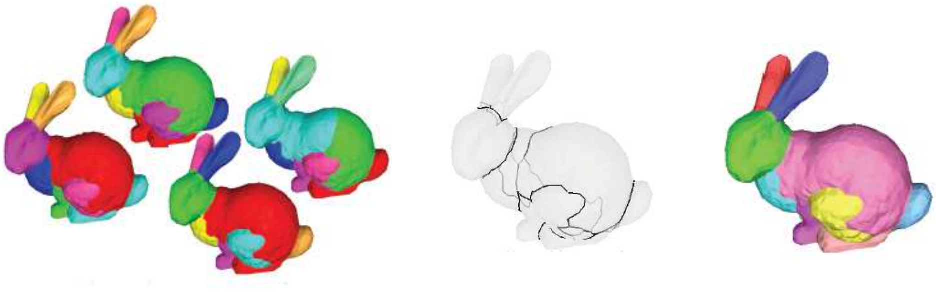


Figure 2.39, segmentation steps of the randomized cut algorithm of [138], (a) shows sample random segmentations, (c) is the segmentation result for the Bunny model generated by the related partitioning function (c).

The main disadvantage of the proposed framework is related to the time-consuming process of combining different approaches which has quite a large amount of implementation overhead.

2.3.3. Consistent Segmentation

Recently researchers have proposed a different approach for segmenting the same family of models, which is called co-segmentation [163, 173, 174]. It refers to decomposition of a set of shapes from the same family into consistent semantic parts with correspondence. The rationale behind co-segmentation comes from the fact that more knowledge can be inferred from multiple similar shapes rather than an individual shape [174]. The pipeline of such approaches comprises an initial over-segmentation of all models into primitive patches. Then, an initial co-segmentation of primitive patches is created using a clustering algorithm and finally the result of co-segmentation is achieved via improving the initial segment guesses by an optimization algorithm (see Figure 2.40).

Glovinsky and Funkhouser considered co-segmentation as a graph clustering problem [175]. They built reliable correspondences across segments of shapes using rigid shape alignment. However, due to a lack of shape semantics, their approach cannot correctly deal with models with large variations. To overcome this challenge, Xu *et al*. [176] firstly clustered the shapes based on their style depending on the scale of parts and then applied part correspondence in each group. Nonetheless, their approach still suffers from two main challenges; firstly, it is limited to models that can be properly aligned. And secondly, the expensive computational cost of their approach does not allow applying it on high resolution meshes.

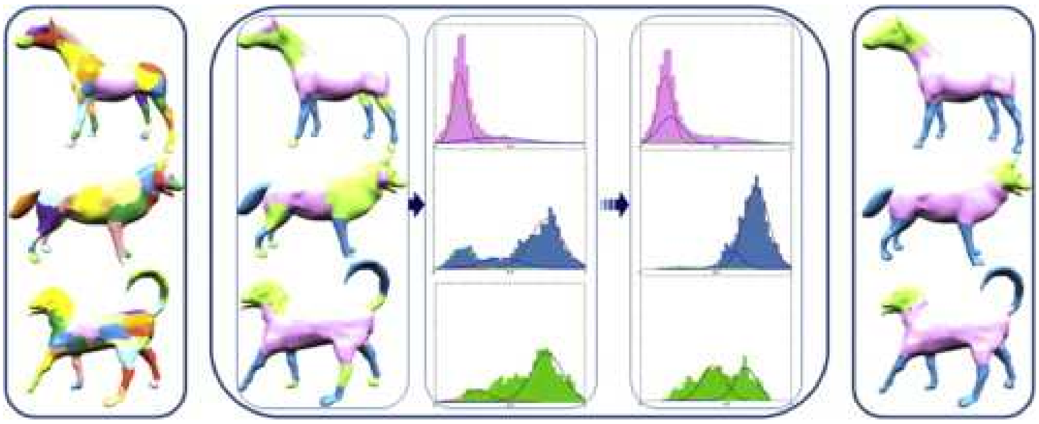


Figure 2.40, co-segmentation approach proposed by Meng *et al.* [173].

Very recently, Meng *et al*. proposed an unsupervised approach for consistent segmentation of similar models by clustering the predefined over-segmented patches using a multi-label optimization [173]. The main drawback of their approach lies in exploiting only two shape descriptors, which may result in a disability to generate proper segments in the case of dissimilar objects. In contrast, Hu *et al*. [174] and Wu *et al.* [163] use five shape descriptors to generate and match initial patches more accurately. Hu *et al*. fuse these five features using an optimization formulation with a consistent multi-feature penalty. Finally, a subspace clustering approach is employed in order to co-segment of models from the same class simultaneously. On the other hand, Wu *et al*. [163] generate a consistent segmentation by performing spectral clustering in a fused space of the five shape descriptors. They try to find an optimal combination of affinity matrices of different descriptors so as to alleviate the impact of unreliable and irrelevant features.

2.3.4. Discussion on the Segmentation Approaches

Mesh segmentation is a quite mature field of research for which dozens of approaches have been proposed. Table 2.2 summarizes discussed segmentation approaches from different classes. Each class of the approaches has its own pros and cons. For instance, the interactive approaches offer high quality segmentation at the expense of losing user-independence characteristics. Region growing approaches on the other hand are quite simple and easy to understand. But their extracted components are not as satisfactory as the human perception's ones.

As will be stated in Chapter 5, our approach presents a fully automatic algorithm using distribution of electrical charge on the surface of models. The proposed approach can be considered as a combination of model-based and region-growing sub-categories (see

Figure 2.22) which offers a robust algorithm for generating high quality segmentation of the models.

Generally speaking, since the segmentation results are quite application-dependent, no one can pick an approach as the best one for all applications. Therefore, despite the presence of various segmentation techniques, proposing a new approach, by which high quality segments are achieved in a fully-automatic manner, is of the great interest to researchers.

Table 2.2, Taxonomy of segmentation approaches.

Segmentation Technique	References	Comment	Advantages	Disadvantages
Region growing	106, 107, 108, 110, 111, 113, 114, 115	Choose set of seeds and grow them until reach the boundaries	- Simple approach - Easy to understand and implementation -Fast	-Highly dependent to the initial seed's location and selection criteria. - Over-segmentation problem
Watershed	115, 116, 118, 119, 120, 122	Select seed points using height function and then apply region growing	- Simple approach - Easy to implement - Not time consuming	-Very low tolerance to noise as they use local features. -Dependency to the defined height function
Interactive	118, 124, 125, 126, 127, 128, 130, 131, 133, 134, 135, 137, 138, 139, 140	Ask users to sketch boundary/protruded parts and then perform automatic segmentation	-More similar to the human perception	-Not fully automatic approach
Skeleton-based	33, 140, 141, 142, 144, 146, 147, 149	Extract skeleton graph and apply graph partitioning	- Hierarchical segmentation - Able to consider geometrical and topological features	- Skeleton extraction is not trivial task. - Quite skeleton-dependent results.
Feature Point- Based	151, 154	Select set of feature points using some criteria to extract segments	-Easy to understand. - Relatively high quality segmentation	- Highly dependent to the extracted feature points - Over-segmentation
Spectral-based	156, 157, 160	Extract spectral embedding of the mesh model for applying clustering algorithms.	-Convert segmentation to easier process of geometric space problem.	-highly dependent on definition of the Laplacian matrix
Learning-based	163, 164, 165	Train the segmentation algorithm using the ground-truth models.	-High accurate segmentation (similar to manual decomposition)	- Time consuming training process - Segment quality is dependent to the size of training data set. -Result are category-dependant.
Volume-based	102, 129, 167	Use volume features to apply segmentation	- Robust to noise	- Mainly working on voxelized models
Primitive fitting- based	169, 170	Fit a set of pre-defined geometrical primitive to find best segments	- Easy to understand	- Highly dependent on the pre-defined primitives
Consistent	163, 173, 174, 175, 176	Consistently segment models from the same class	- More accurate segmentation results -Similar segmentation for same class of models	- Require training phase - Over-segmentation problem - Time consuming

3

Proposed 3D Model Descriptors

3.1. Introduction

In this chapter the problem of 3D model retrieval is discussed. Firstly the motivations for working in this field are presented. Then, the challenges in this domain are described. Then the applications of 3D model retrieval and the proposed shape descriptors are discussed thoroughly. The experimental results associated with the proposed descriptors will be investigated in Chapter 4.

It should be mentioned that from now on, the expressions "3D model", "3D object" and "3D shape" are utilized interchangeably as they refer to the same term.

3.2. Motivation

The number of 3D models in databases is increasing exponentially as the growth in technology allows companies and researchers to create their own models easily. This growth is so fast that one can see a plethora of 3D models on the internet or in other specific databases such as those for Computer Aided Design (CAD), Molecular Biology (3D Protein Models), Computer Graphics, Medicine and Archaeology.

Nowadays, the advances in generating 3D models, such as modelling software, laser scanners and digital cameras have led to the emergence of lots of new models to be utilized in a variety of scientific domains. For instance, 3D laser scanners have enabled users to construct precise models from real objects. The Stanford University Digital Michelangelo [177] and Digital Formae Urbis Romae [178] projects are examples of such attempts to create archives of cultural heritage. Such models have other applications in the film industry and animation. Other kinds of domain-specific archives are also available. For example, the National Design Repository is an online database of CAD models [179] and the Protein Data Bank [180] is an online database of 3D biological macromolecule structures.

As the size of 3D model databases becomes larger and larger, the need for software tools to help people to navigate through these databases has drawn much attention. Such software should provide facilities by which the user is able to specify his/her desired model as a query and search and retrieve similar models in the target database.

On the other hand, the traditional text-based retrieval systems are not helpful anymore as the manual annotation of available models is almost impossible due to the daily mushrooming of such models. Furthermore, most of the time the file name or other textual data associated with the 3D models will not be enough to fully describe what the model actually is. Aware of this, during the last decade, researchers have proposed several approaches based on the content similarity of shapes to retrieve the sought 3D models. Although some of the available approaches perform content-based similarities checking quite successfully, their results are not yet as satisfactory as query results obtained by their textual counterparts. Consequently, a great deal of research should still be conducted to propose new content-based techniques, by which the desired models may be found.

The formal definition of 3D model/object retrieval can be simply presented as follows:

Suppose $\mathcal{Q}$ be a query model and $\mathcal{R}$ is target repository from which the desired models (models similar to the query $\mathcal{Q}$) are going to be searched and retrieved. If $\mathcal{D}$ is the distance function for measuring the dissimilarity between the query model $\mathcal{Q}$ and other models in $\mathcal{R}$, the retrieved objects $\mathcal{O}_i$ are listed according to their distance from the query so that:

$$\begin{aligned} Ret(\mathcal{R},\mathcal{Q}) = \{\mathcal{O}_1,\mathcal{O}_2,\mathcal{O}_3,..\} \mid \mathcal{D}(\mathcal{Q},\mathcal{O}_1) \leq \ \mathcal{D}(\mathcal{Q},\mathcal{O}_2) \\ \leq \mathcal{D}(\mathcal{Q},\mathcal{O}_3 \leq \cdots) \end{aligned} \tag{3.1}$$

The efficiency of the retrieval system depends on whether the starting elements of the retrieved list belong to the same class of the query model. Later in Chapter 4 we will briefly introduce several evaluation criteria for assessing the proposed retrieval algorithms.

The main step and foundation of any typical 3D retrieval system is to describe the query and other available models in a useful and discriminative way. Thus the major focus in this field is to find a descriptor by which all of the characteristics of the models can be represented. As presented in Chapter 2, many techniques have been proposed to represent the models using a numerical feature vector or an appropriate graph. In order to tackle the challenges of the 3D model retrieval domain (see Section 3.3) we will propose three different shape descriptors in this chapter: one is based on solving the Poisson Equation over 2D silhouettes of the objects and the two other approaches are based on computing an electrical charge distribution over the 3D surface of each object. These three descriptors are separately assessed in the next chapter by applying them to various standard datasets.

3.3. Scientific Challenges

A typical 3D retrieval system, as shown in Figure 3.1, consists of two main phases: the off-line and the on-line phase. In both of these phases, in order to represent the models in a discriminative manner, a numerical feature vector or a structural/topological graph is utilized to describe the models. Representation of models using a shape descriptor (numerical vector or graph) instead of the models themselves, results in an easier process of matching the query model to all the models in the target dataset.

Actually, one can consider a shape descriptor as a cornerstone of any retrieval system, by which the shape matching problem can be converted to a numerical vector and/or graph isomorphism comparison. Therefore, most of the challenges in this field are related to the utilized descriptors so that the performance of the retrieval system critically depends on its shape descriptor. In the sequel we list the current challenges in the literature concisely:

- **Accuracy:** In fact, accuracy is the major challenge of any retrieval system so that a significant proportion of research in the literature is dedicated to proposing new *effective* descriptors with the aim of finding and retrieving information more precisely. Like any other retrieval system, the 3D retrieval system and the underlying descriptor should be informative enough to offer high discrimination ability among the models. Despite the presence of numerous 3D model descriptors, proposing a more accurate one is still one of the greatest challenges faced by the leading researchers of this domain.

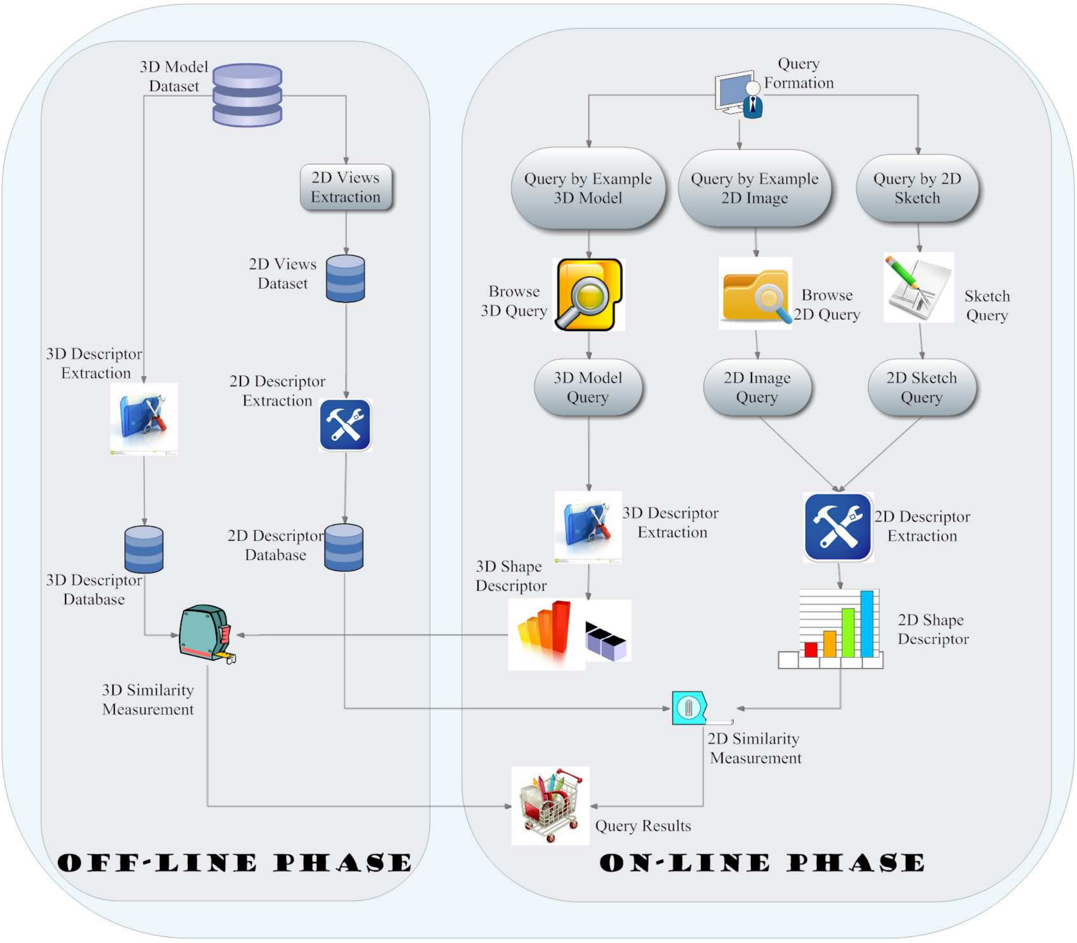


Figure 3.1, A typical 3D model retrieval system.

- **Speed**: another critical issue of any retrieval system is its *efficiency*; it is necessary that the employed shape descriptor be reasonable in size to allow fast extraction in practical systems. Although one may consider the accuracy and speed (effectiveness and efficiency) as conflicting terms, so that an increase in one of them results in a decrease in the other, the retrieval system should be able to make a trade-off between the two.
- **Robustness**: the utilized shape descriptor should be robust to small variations and defects in the model. This property becomes more valuable if we know that some of the 3D models are defective from the start as they have been constructed from damaged real objects (e.g. objects and sculptures discovered in archaeology or noisy sensors). Therefore, the underlying descriptor should be insensitive to a variety of distortions such as noise and small perturbations.
- **Pose Normalization Issue (Invariance)**: the proposed shape descriptor and the matching scheme should be invariant to transformations (e.g. rotation, scale

and translation) as much as possible. Although invariance characteristics can be obtained by applying a pose-normalization step in the pre-processing phase, it reduces the efficiency of the retrieval system by slowing down the total time of the retrieving process. Consequently, the ideal shape descriptor is invariant to geometrical transformations without losing any part of the shape information.

- **Supporting Partial Matching**: Most of the available systems for searching 3D models accept the whole model as a query and fulfil the matching process based on a similarity measure between the complete query and the models in the dataset. But, in the case of the availability of a part of a desired model as a query (e.g. a 3D arm of an antique throne), how do such systems perform the matching process among the available parts of the query model and the models in the dataset? This issue occurs in the interesting case of the Centaur; a Greek mythological race of half-human and half-horse creatures (Figure 3.2). Bronstein *et al*. say that "Arguing whether a Centaur is similar to a horse or to a man is as useless as asking whether a zebra is white or black!" [181]. In order to find the similarity between a Centaur and a human or a horse, the new mechanism of partial matching should be applied rather than whole-shape matching. Generally speaking, partially similar objects, despite having some similar parts, are dissimilar globally. As a result, proposing a new retrieval system supporting partial-matching is of great interest to the retrieval domain experts.

Figure 3.2, Centaur; A mythological Greek creature; half-horse and half-human.

Although there is no shape descriptor which overcomes all of the mentioned challenges in a high level of quality at once, in this chapter we try to tackle these challenges, as much as possible, by introducing three new shape descriptors from both the 2D and 3D domains.

3.4. Applications

The large collections of 3D models on the Internet and other sources have stimulated interest in the development of 3D model search/retrieval techniques in a variety of applications ranging from academia to the business domains. In the sequel we summarize the applications of the 3D model search/retrieval briefly:

a) **Engineering**: Computer Aided Design (CAD) components are widely used in the (re)-construction of objects in different domains viz. automobile, aircrafts, mechanical parts *etc*. When many CAD models are available, it is inefficient to manually search for required ones. So, designing suitable software to search such models is quite beneficial in the engineering field to reduce the cost of creating a new component by replacing it with the available similar ones. The Purdue Engineering Shape Benchmark (ESB) offers a collection of engineering components described as triangulated 3D meshes [182]

b) **Medicine**: The available 3D images obtained by CT and MRI scans are broadly used for diagnosis of organ deformations by matching actual images with medical databases of known deformations. A retrieval system for non-rigid models can help physicians to detect and diagnose organ disorders. This is one example of the application of 3D retrieval systems in the medical domain.

c) **Chemistry and Biology**: in order to design drugs and classify their structure, molecules and proteins are modelled as 3D objects. So, a deformable and/or partial object matching system is required to classify various types of proteins. On the other hand, in drug design, it is crucial to search and compare deformable 3D models of the query molecule to the others in the target molecular databases, so that appropriate drug molecular structures can be designed to cure specific diseases. The two special tracks in the SHREC'07 [183] and SHREC'10 [184] contests using a protein dataset, which includes more than 30000 protein structures, show the importance of model retrieval in these branches of science.

d) **Cultural Heritage**: another famous application of 3D model retrieval is related to digitized objects such as old buildings, statues, pottery *etc*. Performing global or partial matching among similar artifacts can reveal the relationships among historical cultures of ancient civilizations. The SCULPTEUR project [185] is a well-known example of such an application. It aims to develop a system to store, search and retrieve multimedia content and associated metadata that form the digital collection of a museum or art gallery.

e) **Entertainment**: Games and animation movies are utilizing 3D models for the purpose of holding considerable influence on the players and audiences. Using a database containing lots of animation components, a suitable search engine can save both time and cost in re-constructing animation characters.

f) **Face Recognition**: To combat the challenges in 2D face recognition (such as effects of lighting conditions, facial expressions, facial orientations *etc.*) 3D face recognition has become one of the popular biometric techniques in many security gateways namely airports, nuclear plants, hotels and so on. Despite the significant variations in the approaches taken by researchers in relation to this field, achieving the ideal 3D face recognition system is still of enormous interest to companies and research groups.

g) **Others**: Robotics, 3D spatial terrain *etc.* are among major applications in which 3D model retrieval plays a critical role in enhancing their performance and efficiency.

In the next following sections we will completely introduce our proposed shape descriptors along with the detailed description of their beneficial characteristics.

3.5. Proposed Approaches

In this thesis we have proposed three different shape descriptor; one 2D Poisson-based and two different 3D charge distribution-based descriptors. In this section we present these descriptors separately.

3.5.1. 2D Poisson-Based Shape Descriptor

The 2D-based shape descriptors usually extract the 2D projections of the 3D model such as silhouettes, depth buffers, contours or other kinds of image presentations and define each image by a set of numerical values in a beneficial feature vector.

Our motivation for proposing the 2D Poisson-based descriptor for 3D shapes is as follows: very recently Pan *et al.* [73] used the 3D Poisson equation for retrieving voxelized models and got significant results such as high retrieval quality and robustness to noise, smoothing and simplification. On the other hand, as mentioned in Chapter 2, the view-based approaches have shown relatively high retrieval quality in the 3D retrieval domain. Consequently, we decided to use the 2D Poisson Equation to boost the retrieval accuracy as well as exploit the advantages of the Poisson Equation.

A typical 2D-view-based retrieval system consists of the following steps:

1) **Pose Normalization**: The 3D models are aligned in the pre-processing phase.
2) **View Extraction**: A desired number of 2D view images are extracted from each 3D model.
3) **Descriptor Extraction**: Each view is described using an applicable shape descriptor.
4) **Matching**: The matching process is performed based on the comparison of corresponding views of the models.
5) **Query Results**: The best matches are retrieved as a result of retrieving process.

Most of the view-based methods need a pose normalization process before extracting the 2D views. In order to perform pose normalization, each 3D model should be normalized for Scale, Translation and Rotation. One possible approach for the translation is to translate the model so that the centre of mass of the model is at the origin. Usually scale invariance is accomplished by scaling the model so that it lies within the unit bounding sphere. Securing rotation invariance is usually more difficult than the others. Few methods have been introduced for achieving rotation invariance [43, 186, 187, 188]. The most prominent one is Principal Component Analysis (PCA) (which is also called as the Karhunen-Loeve transform or Hotelling Transform), or some variation of it such as Normal PCA (NPCA) [187], or ContinuousPCA [186].

In the coming section, prior to presenting the Poisson-based descriptor, a brief introduction to the Poisson Equation will be given. It makes this document more understandable and easy-to-follow.

3.5.2. Background

In mathematics the Poisson Equation is a second-order partial differential equation of elliptic type with broad utility in electrostatics, mechanical engineering and theoretical physics [189]. The Poisson Equation is defined by following formula:

$$\nabla^2 \theta = \mathscr{f} \tag{3.2}$$

Where ∇^2 is the Laplace operator, and $\mathscr{f}$ and θ are real or complex-valued functions on a manifold, (for example θ can represent electric potential and $\mathscr{f}$ is electric charge).

Application of the Poisson Equation in our approach is defined as follows:

Consider an extracted silhouette view $\mathcal{S}$, which is surrounded by a simple contour. We would like to assign a number to each internal point of $\mathcal{S}$ based on its location in the silhouette. One way is to solve the Poisson Equation in form of:

$$\nabla^2 \mathcal{U} = \mathcal{U}_{xx} + \mathcal{U}_{yy} = -1 \tag{3.3}$$

Where $\mathcal{U}_{xx}$ and $\mathcal{U}_{yy}$ are the second derivatives of $\mathcal{U}$ in x and y directions, respectively and $(x, y) \in \mathcal{S}$ with the boundary condition $\mathcal{U}(x, y) = 0$.

One can simply describe the Poisson Equation as placing a set of particles at each internal pixel point of the silhouettes and letting them move in a random walk until they hit the boundary. In this context U represents the mean time taken to reach the boundary.

3.5.2.1. Silhouette Poisson-Histogram Descriptor (SilPH)

As shown in Figure 3.3, our method includes two phases as follows:

- **Offline phase**: The desired number of views is extracted for all of the normalized 3D models in the dataset and then a Poisson equation solver is utilized to define their descriptors.
- **Online phase**: The system gets a 3D model as query and following the pose normalization step, the silhouette views are extracted. After calculating the shape descriptor for each view via the Poisson equation, the matching process is performed so as to find the similarity measurement between the query model and the other models. Then the results are retrieved based on the degree of similarity.

In the sequel, we present detailed descriptions of these two phases.

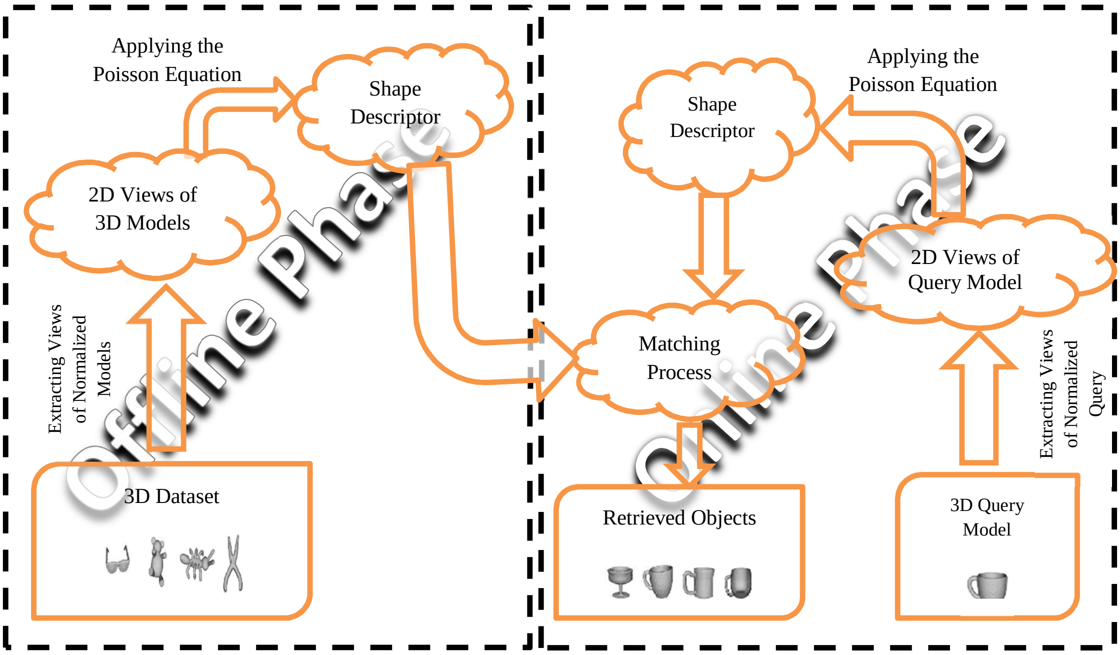


Figure 3.3, the proposed model retrieval Framework; left part shows the Offline phase and right part is the Online phase

As with any other view-based approach, the process is started with pose normalization; first all of the models are scaled by placing them into the bounding unit sphere while rotation invariance is accomplished by applying the NPCA approach to the scaled models. Because of the area weighted defect[3], the accuracy of PCA in 3D model normalization is reduced [190]. To solve this problem NPCA is proposed by Papadakis *et al*. [187] to extract the principal axes more precisely. With the aim of finding principal axes in NPCA, the collections of normal vectors for all of the triangular meshes are utilized in which the weight of each normal is equal to area of the triangle. It is important to note that since our descriptor is defined regardless of coordinate system, it is implicitly invariant to translation.

The next step consists of 2D view images extraction. To do this, we employ the 60 vertices of a truncated icosahedron. By placing the camera on the 60 vertices of the truncated icosahedron, 60 silhouettes with resolution of 128*128 pixels are extracted. The results show that the selected resolution is a good trade-off in terms of speed and accuracy. Now each silhouette view is described by a feature vector which consists of a histogram generated from the solution of the Poisson Equation.

[3] The area weighted defect is a phenomenon, in which a lot of small area meshes which create the fine parts of a surface, get larger weights than the other parts and thus greatly bias the accuracy of PCA.

Finally, in order to describe the silhouettes using the Poisson Equation, we utilized the average time required for a particle to hit the boundaries as a descriptor for each pixel. Based on the Poisson Equation, the value at each pixel is a constant plus the average value of its neighbours. It is quite similar to the distance transform [191] but unlike the distance transform, which only considers the nearest boundary point, the Poisson equation is affected by several internal points and then reflects more global structure of the silhouette. These global effects lead to more robustness to noise distortion. The experimental results in Chapter 4 support our idea about this class of robustness.

To solve the Poisson Equation we utilized the Poisson solver proposed by Gorelick *et al.* [192]. Figure 3.4 shows some silhouettes from the McGill dataset and their solutions to the Poisson Equation. These solutions become the descriptor for each shape.

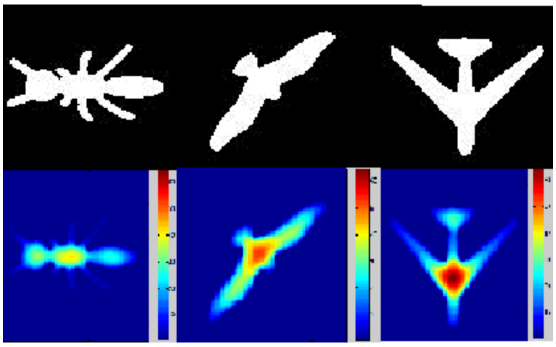


Figure 3.4, Some sample silhouettes and their solution to the Poisson equation.

In order to compute the similarity measure between two models, a view-to-view comparison among the entire 60 silhouettes is utilized. To do so, the pixel values are accumulated into a histogram called the Silhouette Poisson Histogram (SilPH). The histogram is constructed via clustering pixel values into 10 bins using a K-Means clustering approach.

The Dissimilarity $\mathcal{D}_{\mathcal{M},\mathcal{N}}$ between two corresponding view histograms $\mathcal{H}_{\mathcal{M}}$ and $\mathcal{H}_{\mathcal{N}}$ is defined by Equation 3.4:

$$\mathcal{D}_{\mathcal{M},\mathcal{N}} = \left[\sum_{i=1}^{10} \left|\mathcal{H}_{\mathcal{M},i} - \mathcal{H}_{\mathcal{N},i}\right|^{r}\right]^{1/r} \tag{3.4}$$

Where $r = 1$ for $\mathcal{L}_1$ and $r = 2$ for $\mathcal{L}_2$ distance function.

Finally, the dissimilarity $DIS(\mathcal{A}, \mathcal{B})$ between two different 3D models $\mathcal{A}$ and $\mathcal{B}$ based on their histograms is calculated by the sum of the dissimilarities between all of their corresponding views. Note that the comparison is only performed between corresponding views ($\mathcal{K}^{th}$ view of model $\mathcal{A}$ with $\mathcal{K}^{th}$ view of model $\mathcal{B}$) and does not perform brute-force matching which includes comparing each view of model $\mathcal{A}$ with all views of model $\mathcal{B}$.

3.5.2.2. Discussion of the SilPH Descriptor

The SilPH descriptor is a view-based approach, so it is expected to offer high retrieval capability compared to its 3D-based counterpart. Furthermore, it is easy to calculate and understand and also since the SilPH descriptor is defined regardless of coordinate system, it is robust to translation. Moreover, as mentioned above, the number assigned to each pixel as its signature in the SilPH, is the average time for particles to hit all of the boundaries. So, the effect of noise can be easily minimised by the contribution of all boundary points and accordingly the SilPH descriptor becomes less sensitive to noise. By the same reason it is expected to provide high matching ability for deformable models.

On the other hand, the SilPH descriptor suffers from some disadvantages: it is sensitive to scale and rotation transformations. Additionally, like other view-based approaches, due to multiple comparisons between the views of each pair of models, the matching speed is considerably reduced. Finally, since it describes each pixel via the effect of all boundary points, it considers the global structure of the shape and so it cannot be utilized for the purpose of partial matching.

The ability of the proposed SilPH descriptor will be experimentally tested in Chapter 4, by applying it to the models available in the various standard datasets. Furthermore, its robustness against noise along with the effects of utilized parameters will be investigated thoroughly.

3.5.3. 3D Charge Density-Based Shape Descriptor

Unlike the SilPH descriptor, the second and third proposed descriptors belong to the 3D-based class of shape descriptors. The underlying rationale for selecting this class of descriptor is simply as follows:

As stated in Chapter 2, the retrieval efficiency of 2D-based descriptors is primarily affected by pose normalization in the pre-processing phase. A good alignment technique can provide better potential for enhancing the discrimination ability. On the other hand, in addition to consuming extra time, proposing a reliable technique for aligning the models is a non-trivial task. Therefore, we tried to introduce our new descriptors from the 3D-based category, which is robust against affine transformation especially rotation.

There is a famous fact in the physics of electricity which describes a natural phenomenon and says: "the electric charges on the surface of a conductor tend to accumulate at the sharp convex areas and disappear at the sharp concave areas" [193]. Since natural physical phenomena are dependent on the nature of the objects, we think they work for any kind of surface and can describe the models in a distinguishable manner. Consequently, we aim at employing the density/amount of distributed electrical charge on the surface of models as the surface descriptors, and utilize them in the model retrieval framework.

In the next sections, first we briefly introduce the background of charge-distribution, which is the cornerstone of the proposed descriptor, and then its application in 3D model description will be presented.

3.5.3.1. Background

As stated before, when a pre-defined electrical charge Q is placed on the surface of a 3D model, which is viewed as a perfect conductor, the electrical charge tends to accumulate at sharp convexities and vanish at sharp concavities. In order to use this fact as the foundation of the proposed descriptor, the 3D models are treated as conductors which are placed in a free space (a space with no electric charge) and the electrical charge Q is distributed on their surface. Since the 3D models have arbitrary surfaces, it is not possible to calculate the charge density on the surface using an analytical approach. Thus, a Finite-Element-Method (FEM) is utilized to this end. We used the technique proposed by Wu and Levine to calculate the charge density [171]. Their approach can be briefly expressed as follows:

Each facet of the triangular mesh models is considered as a planar triangle $\mathcal{T}_{\mathscr{k}}$ which possesses a constant charge density $\rho_{\mathscr{k}}$ that should be calculated. To do so, firstly the reciprocal electrical potential of every pair of faces is identified by Equation 3.5:

$$\varphi(\boldsymbol{r}) = \frac{\mathscr{q}}{4\pi\varepsilon_0} \frac{1}{|\boldsymbol{r} - \boldsymbol{r}'|} \tag{3.5}$$

Where ε_0 is known as permittivity of free space and $\boldsymbol{r}$ is the vector position of observation point and $\boldsymbol{r}'$ is the position of charge $\mathscr{q}$, as shown in Figure 3.5.

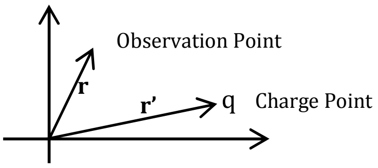


Figure 3.5, the configuration for a charge point q placed at the point $\boldsymbol{r}'$ which is observed from the point $\boldsymbol{r}$.

Since, all of the triangular faces contribute to the potential $\varphi(\boldsymbol{r})$, it can be re-written as follows:

$$\varphi(\boldsymbol{r}) = \frac{1}{4\pi\varepsilon_0} \int_{\mathcal{S}} \frac{\rho(\boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|} d_{\mathscr{s}}' \tag{3.6}$$

Where $\mathcal{S}$ is the total surface area, $\rho(\boldsymbol{r}')$ is the charge density at $\boldsymbol{r}'$ and $\mathscr{s}'$ is the area over $\mathcal{S}$. In order to calculate $\rho(\boldsymbol{r}')$ using FEM, the model surface $\mathcal{S}$ is considered as $\mathcal{N}_{\mathcal{T}}$ triangles $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3, \dots, \mathcal{T}_{\mathcal{N}_{\mathcal{T}}}$. Therefore, the Equation 3.6 can be expressed as:

$$\varphi(\boldsymbol{r}) = \frac{1}{4\pi\varepsilon_0} \sum_{\mathscr{k}=1}^{\mathcal{N}_{\mathcal{T}}} \left(\rho_{\mathscr{k}} \int\limits_{\mathcal{T}_{\mathscr{k}}} \frac{1}{|\boldsymbol{r_i} - \boldsymbol{r}'|} d_{\mathscr{s}}' \right), \qquad i = 1,2,\dots,\mathcal{N}_{\mathcal{T}} \tag{3.7}$$

On the other hand, the total charge $\mathcal{Q}$ is equal to the sum of charges on each triangle. Thus:

$$\mathcal{Q} = \sum_{\mathscr{k}=1}^{\mathcal{N}_{\mathcal{T}}} \rho_{\mathscr{k}}\, \mathscr{s}_{\mathscr{k}} \tag{3.8}$$

Here $\rho_{\mathscr{k}}$ and $\mathscr{s}_{\mathscr{k}}$ are charge density and surface area of triangle $\mathscr{k}$ respectively. Using Equations (3.7) and (3.8), a set of linear equations with $\mathcal{N}_{\mathcal{T}}$+1 unknown $\rho_1, \rho_2, \dots, \rho_{\mathcal{N}_{\mathcal{T}}}$ and $\varphi(\boldsymbol{r})$ in the form of $\mathcal{A} * \rho = \mathcal{Q}$ are obtained which $\mathcal{A}$ and $\mathcal{Q}$ are known terms and ρ is unknown variable. This set of equations can be easily solved using a method such as the conjugate gradient squared method [194]. As discussed in [171], the Matrix A contains the following variables which are extracted from equations 3.7 and 3.8.

$$A = \begin{pmatrix} A_{11} & A_{12} & \cdot & A_{1N} & -1 \\ A_{21} & A_{22} & \cdot & A_{2N} & -1 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ A_{N1} & A_{N2} & \cdot & A_{NN} & -1 \\ S_1 & S_2 & \cdot & S_N & 0 \end{pmatrix} \quad \text{Where:} \qquad A_{ik} = \int_{T_k} \frac{1}{|\mathbf{r}_i - \mathbf{r}'|} dS', \quad i,j = 1,\ 2,\ \dots, N$$

Solving this set of equations results in obtaining the density of distributed electrical charge $\mathcal{Q}$ on any individual triangular face $\mathcal{T}_i$ of model $\mathcal{M}$. Figure 3.6 depicts simulated distributions of electrical charge on the surface of some 3D models.

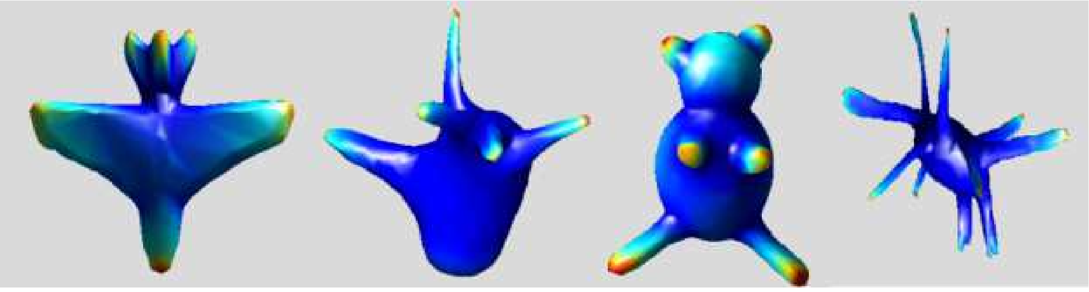


Figure 3.6. Example coloured models from the McGill dataset; the redder parts specify the denser faces.

We employed the simulated distribution of electrical charge so as to retrieve 3D models in two different approaches, namely the Bag-of-Feature Charge Density Descriptor (BoF-CDD) and the combination of Concentric Sphere-Electrical Charge Descriptor (CS-ECD) and Dense Patches (DP_CS-ECD).

3.5.3.2. Discussion on Electrical Charge Distribution as a Descriptor

The simulated charge density on the surface of models holds interesting properties which are very beneficial in the 3D retrieval domain. In the following we will briefly explain some of them.

- **Invariance to transformations:** the charge density distributed on the surface of a model only depends on the total amount of distributed charge (Q) and size of the model. (The charge amount of each triangular element is even independent from the model size). Since the density is calculated regardless of the coordinate system, it is completely robust to linear translations such as translation and rotation.
- **Gathering local and global information:** as Wu and Levine mention [171], the distributed charge on each surface is affected by all of the points on the model surface. On the other hand, the effects of all triangular elements in generating the potential are not equal: that is, their contributions depend on the distance between the source and observation points (see Figure 3.5). Therefore, the charge density holds both local and global information about the structure of the model. Wu and Levine considered these characteristics as "quasi-local" and "quasi-global" properties.
- **Insensitivity to noise and simplification:** the quasi-global property of charge density leads to less sensitivity to noise and simplification. As the charge density on each face is contributed to by all other faces, the small boundary changes which are caused by noise and simplification have almost no meaningful effects

on the density. This is a great advantage compared to the curvature-based approaches (e.g. mean-curvature and curvature-index); they are considerably affected by any surface perturbations (See Section 4.5 of Chapter 4). The charge density is robust to simplification as well; although during the simplification process, the sizes of triangular faces are increased, the amounts of distributed charges are changed by the same ratio. So, the charge density of the simplified faces remains the same.

- **Supporting partial matching:** the quasi-local property of charge density will result in defining each part in a finer manner. It enables us to use the local information around each part to describe it and hence is expected to support partial matching.

The aforementioned features are tested experimentally in Chapter 4.

3.5.3.3. Bag-of-Feature Charge Density Descriptor (BoF-CDD)

The Bag-of-Features (Words) was originally devised for use in text-based information retrieval systems. In the BoF framework, a text (such as document or sentence) is considered as an unordered collection of words disregarding grammar or even word order. Despite the simplicity of such a representation, the retrieval methods that use the BoF framework often have shown a high retrieval performance, so a great deal of research has been conducted to employ BoF in both 2D and 3D Image Retrieval [195, 196].

A typical content-based image retrieval system using the BoF framework has 3 major steps; (1): feature point selection, (2): visual dictionary building and (3): histogram generation. Figure 3.7 illustrates the simple BoF framework for three different objects. Thanks to the BoF framework, the local 3D shape descriptors can be utilized for global shape matching. Several works have described models using the BoF framework.

In the first proposed charge-based descriptor we utilize the density of distributed charge on each face in the BoF framework. It starts by selecting interest points on the surface of the models. So, for each model j we choose $\mathcal{N}_\mathrm{j}$ surface triangles (for all of $\mathcal{M}$ models in the dataset). Various criteria are employed to select the interest points, viz. random selection, local maxima points and the higher-than-mid. density points. These criteria are detailed and tested in Chapter 4. The densities of the surface triangles become the descriptors of each face.

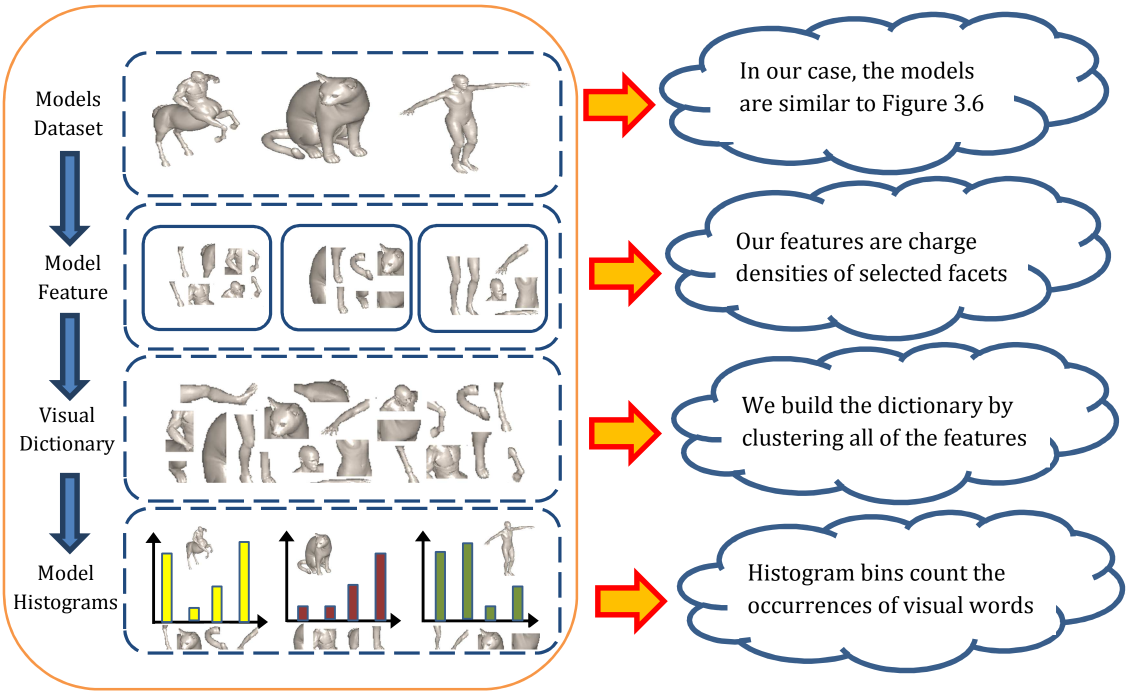

Models Dataset
Model Feature
Visual Dictionary
Model Histograms
In our case, the models are similar to Figure 3.6
Our features are charge densities of selected facets
We build the dictionary by clustering all of the features
Histogram bins count the occurrences of visual words

Figure 3.7, A typical BoF framework in a 3DOR system

The next step is to build a dictionary of visual words. To do so, the entire set of $\sum_{i=1}^{\mathcal{M}} \mathcal{N}_i$ feature point descriptors are clustered into $\mathcal{D}$ clusters by the K-Means algorithm using the VLFeat source code, which is publicly available on the internet [197]. The cluster centres are considered as the visual words. Then, for any model m_i, a feature vector is constructed by selecting $\mathcal{N}_i$ sample points for which the charge density acts as the local descriptor. Finally, each model is described by a histogram in which any individual bin counts the number of occurrences of the visual words in the model.

3.5.3.4. Concentric Sphere-Electrical Charge Descriptor (CS-ECD)

In the second electrical charge-based descriptor, instead of using density of charge, we utilize the amount of distributed charge on the triangular faces as the local descriptor. The conversion between amount and density of the simulated distribution of charge for any face $\mathcal{T}_\mathrm{i}$ can be easily carried out using Equation 3.9:

$$\rho(\mathcal{T}_\mathrm{i}) = \frac{\omega(\mathcal{T}_\mathrm{i})}{\Lambda(\mathcal{T}_\mathrm{i})} \qquad (3.9)$$

Where ρ, ω *and* Λ denote charge density, charge amount and the surface area of face $\mathcal{T}_\mathrm{i}$, respectively.

In order to describe each model using the Concentric-Sphere Electrical Charge Descriptor more precisely, a beneficial two-phase description scenario, including concentric spheres and dense patches, is employed. In the sequel these two phases are thoroughly presented.

3.5.3.4.1. First Phase of DP_CS-ECD construction

During the first phase, the model is enclosed by a set of concentric spheres with their centre at the centre of mass of the model. The radii of the spheres monotonically increase to enclose the model entirely. Assume that $\mathcal{N}_{\mathcal{S}}$ is the number of concentric spheres. So, we plan to use the total amount of charge in each layer between two adjacent spheres as an element of the feature vector to describe the models. Finally, the whole model is described using the $\mathcal{N}_{\mathcal{S}}$ dimensional feature vector, which is constructed by the $\mathcal{N}_{\mathcal{S}}$ numbers assigned to the layers. Figure 3.8 depicts a human model in various poses along with the related feature vectors shown as histograms (here $\mathcal{Q} = 100$ and $\mathcal{N}_{\mathcal{S}} = 5$). As illustrated in this figure, the deformation of the human model results in creating different descriptors which is a challenging issue in retrieving deformable models using the proposed descriptor.

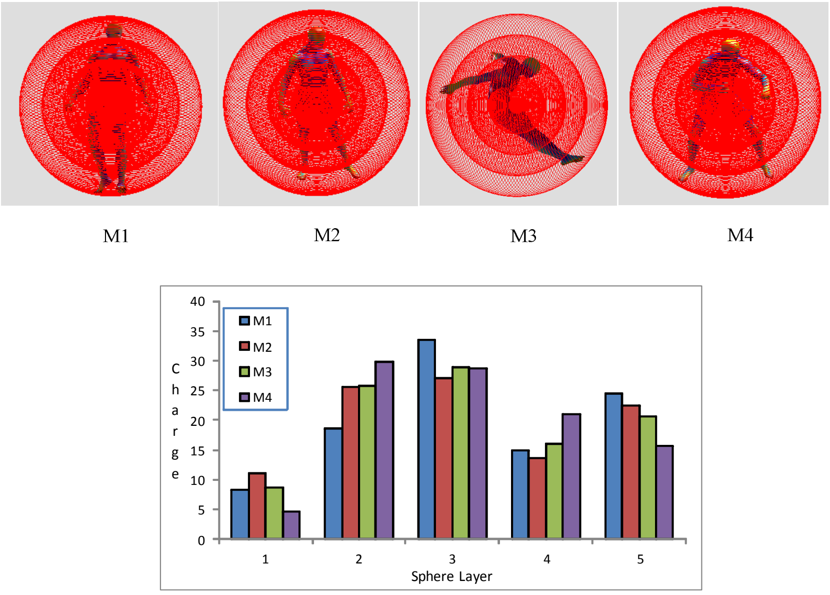


Figure 3.8, Various poses of a human model enclosed in 5 concentric spheres along with the associated histogram descriptors.

To overcome this problem, the canonical form representations of the models are utilized to construct the $\mathcal{N}_{\mathcal{S}}$ dimensional descriptor. The canonical form is a bending-invariant representation in which geodesic distances are approximated by Euclidian ones. Since bending has only a minor influence on the geodesic distance [198], different poses of similar objects have a similar canonical form representation. This idea was originally introduced by Elad and Kimmel [198]. They also compared three different MDS techniques to calculate the canonical form. In our work, we employ the Least-Square technique with the SAMCOF algorithm to compute the canonical representation of the models [198]. The SAMCOF is an iterative algorithm which starts with an initial approximation of possible solutions and then tries to update the results so that the final stress function becomes less than a threshold. Figure 3.9 displays the results for four different poses of an ant model. Although the poses are different their canonical form presentations are quite similar.

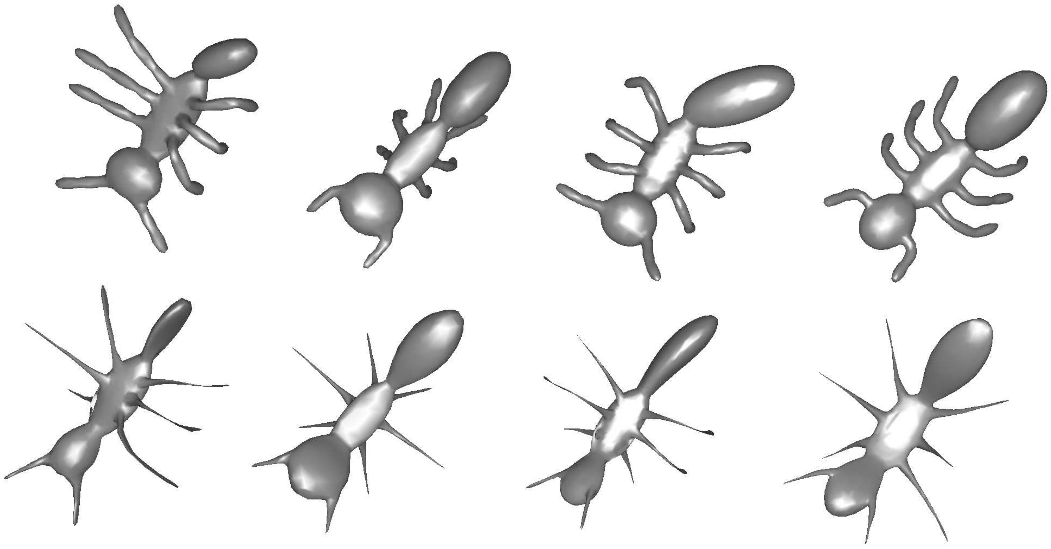
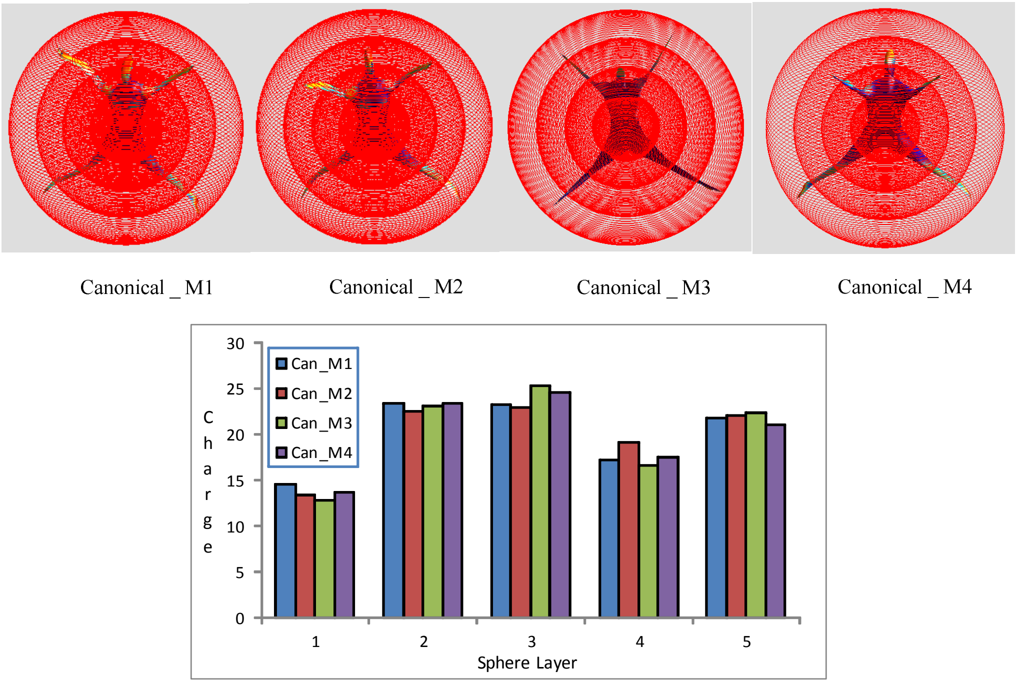


Figure 3.9, Four different poses of an ant model and their corresponding canonical forms.

Since both the SAMCOF algorithm and the geodesic distance extraction are time consuming tasks, we first simplify all of the models using MeshLab [199] so that they have 5000 faces. Later in Chapter 4 the robustness of the charge distribution against simplification is examined. As displayed in Figure 3.10, corresponding layers between adjacent spheres in the canonical form representations of various poses of a human model possess quite similar amounts of electrical charge. Therefore, the proposed descriptor is expected to offer a high quality of retrieval among deformable models.



Canonical _ M1 Canonical _ M2 Canonical _ M3 Canonical _ M4

Figure 3.10, Various poses of a human model represented in the canonical form enclosed in 5 concentric spheres along with the associated histogram descriptors.

3.5.3.4.2. Second Phase of CS-ECD construction (DP_CS_ECD)

In the second phase of construction the DP_CS-ECD descriptor, we aim at boosting the discrimination ability of the descriptor by considering the number of Dense Patches (DP) on the surface of each charged model and combining it with the feature vector extracted in the first phase.

To do so, each DP is defined as follows:

"***A Dense Patch (DP) is a local maximum point (a surface with higher electrical charge than its neighbours) along with a set of adjacent faces on the model surface which have a charge density more than a pre-defined threshold τ***"

The threshold τ is experimentally defined as Equation 3.10:

$$\tau = 0.3 * \max(\rho_i)\,, \quad i = 1,2, \dots, \mathcal{N}_{\mathfrak{f}} \qquad (3.10)$$

Where ρ_i is the charge density of face i and $\mathcal{N}_{\mathfrak{f}}$ is the number of faces on the model. Figure 3.11 shows some extracted dense patches on the models based on the density distribution

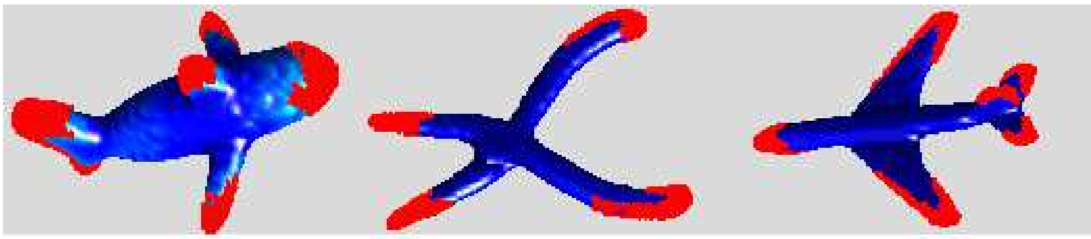


Figure 3.11, extracted dense patches on the surface of three different models.

The motivation for using the number of DPs is as follows: the models from the same class, regardless of their poses, have the same number of protruded parts. On the other hand, DPs are emerging at the very end sections of the protruded parts (see Figure 3.11). So, it is reasonable to expect models from the same class to possess the same number of DPs. Consequently, combining the number of DPs with the original CS-ECD to construct a hybrid descriptor is able to boost its discrimination ability.

Now, the dissimilarity between two models $\mathcal{M}_\mathrm{i}$ and $\mathcal{M}_\mathrm{j}$ is defined based on the combination of the two factors: the $\mathcal{N}_s$ dimensional descriptor extracted in the first phase and the number of DPs identified in the second phase. Equation 3.11 formulates the matching criteria using the two factors:

$$\mathcal{DIS}(\mathcal{M}_\mathrm{i},\mathcal{M}_\mathrm{j}) = Sph\mathcal{D}is(i,j) + \mathcal{DP}\_Diff(i,j) \qquad (3.11)$$

Here, $Sph\mathcal{D}is(.,.)$ denotes the $\mathcal{L}_1$distance between $\mathcal{N}_s$ dimensional descriptors of the two models and $\mathcal{DP}\_Diff(i,j)$ is the difference in the DP counts of two models i and j.

It is important to note that since the values of the two factors are not in the same range (e.g. for the models in the McGill dataset: $0 \leq Sph\mathcal{D}is(i,j) \leq 126$ and $0 \leq \mathcal{DP}\_Diff(i,j) \leq 10$), their linear combinations does not lead to achieve the desired goal. That is, the effect of one factor overcomes the other one. Therefore, the two factors should be transformed so that they get values in the same range. To do so, we use a kind of min-max normalization function which transforms the values of $\mathcal{DP}\_Diff(i,j)$ to the range of $Sph\mathcal{D}is(i,j)$:

$$New\_\mathcal{DP}\_Diff(i,j) = \lambda * \mathcal{DP}\_Diff(i,j) + Min(Sph\mathcal{D}\mathrm{is}(\mathrm{i},\mathrm{j})) \qquad (3.12)$$

Here, λ is the scale factor and is defined as Equation 3.13.

$$\lambda = \frac{Max\big(Sph\mathcal{D}is(i,j)\big) - Min(Sph\mathcal{D}is(i,j))}{Max(\mathcal{DP}\_Diff(i,j)) - Min(\mathcal{DP}\_Diff(i,j))} \tag{3.13}$$

So, the Equation 3.11 can be re-written as follows to measure the dissimilarity between two models $i \; and \; j$:

$$\begin{aligned} \mathcal{DIS}\big(\mathcal{M}_{\mathrm{i}}, \mathcal{M}_{\mathrm{j}}\big) = {} & Sph\mathcal{D}is(i,j) \\ & + New\_\mathcal{DP}\_Diff(i,j) \end{aligned} \tag{3.14}$$

3.6. Chapter Summary

This chapter was completely dedicated to the 3D model retrieval field. After giving an overview to the research challenges of 3D model retrieval and its application in various fields of computer vision, our proposed shape descriptors have been presented entirely. These descriptors include a 2D Poisson-based and two 3D electrical charge distribution-based descriptors.

The proposed descriptors possess beneficial characteristics and so they are expected to offer a high retrieval quality. In order to test their discrimination ability a complete investigation on the standard datasets will be held in the next chapter along with the effect of utilized parameters in a more in-depth analysis.

4

Experimental Results of the Proposed Shape Descriptors

4.1. Introduction

The previous chapter discussed our proposed shape descriptors from both the 2D and 3D categories of available model descriptors. In the current chapter, we present complete evaluations of the proposed descriptors on the various standard benchmarks.

The organization of the chapter is as follows: Section 4.2 introduces the underlying datasets to which the proposed descriptors will be applied. Then, the evaluation metrics to measure the efficiency of the descriptors are presented in Section 4.3. Before applying our descriptors to the datasets in Sections 4.5, 4.6 and 4.7, the utilized parameters will be set in Section 4.4. Then the robustness of descriptors is tested against different changes and transformations in Section 4.8. Finally, the complexity of the proposed descriptors will be discussed in Section 4.9.

4.2. Description of Datasets

Three different datasets have been utilized so as to evaluate the proposed shape descriptors more precisely. These datasets include:

a) **McGill 3D Shape Benchmark (MSB)**: The McGill dataset [79] mainly includes models with articulated parts such as Human, Hand, Spider *etc.* (255 models in 10 classes). It also offers various classes of models having moderate or no part articulations viz. Table, Chair, Airplane *etc.* (202 models in 9 classes). Some of the models are adapted from the PSB dataset [159] and others have been downloaded from different web repositories. Figure 4.1 displays sample models from each of the 19 classes of the benchmark.

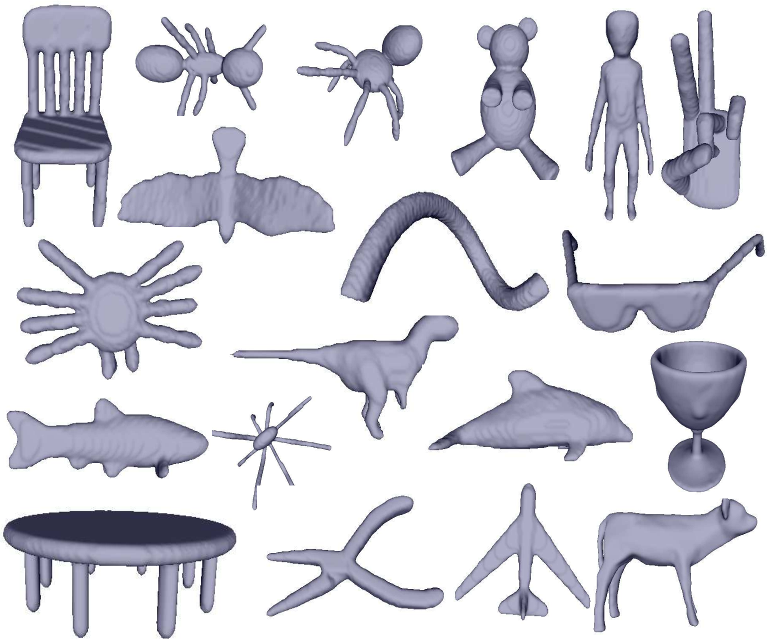


Figure 4.1, sample models from the McGill dataset.

b) **Watertight Models of the SHREC'07 dataset**: The 400 seamless models of the watertight[4] track of the SHREC'07 contest [162] are evenly categorized into 20 classes with 20 models in each class. The dataset offers a set of queries containing 30 models, which have been constructed via sub-part combinations of models from different classes. The query set has been designed with the aim of evaluating partial-matching algorithms. Appendix B displays the query set along with the information about the sub-parts.

c) **Non-Rigid 3D Watertight Meshes of SHREC'11 dataset**: This large-scale dataset includes 600 non-rigid models commonly seen in our surroundings. As shown in Figure 4.2, the dataset contains 30 classes in which 20 different poses of similar models are available. The reason for selecting this benchmark is related to the presence of challenging classes of models. That is, quite similar models have been classified into different groups. For instance, various poses of

[4] *A watertight model is a mesh model which has no whole, cracks or missing features on the surface.*

two kinds of bird (and dog) model are placed into different groups. Furthermore, the centaur class, which is partially similar to the man and horse classes, makes the dataset more interesting to examine.

Additionally, since some of the models in the SHREC'11 dataset have been borrowed from other well-known datasets (e.g. TOSCA [200] and PSB [159]), one can consider the dataset as a general benchmark, which covers a broad range of available models in the literature.

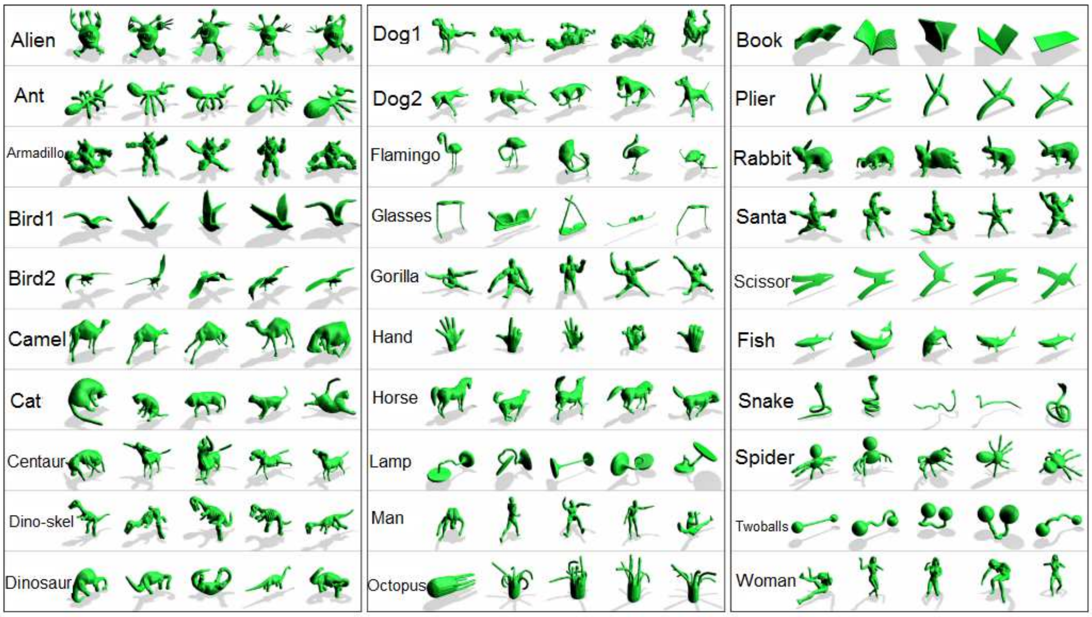


Figure 4.2, 30 classes of the SHREC'11 non-rigid watertight dataset along with 5 samples for each class.

4.3. Evaluation Metrics

The efficiency of a 3D model retrieval system can be evaluated by several factors namely Nearest Neighbour (NN), First Tier (FT), Second Tier (ST), E-Measure, Discounted Cumulative Gain (DCG) and the Precision-Recall Curve. In the paragraphs that follow, we briefly introduce these metrics:

- **Precision-Recall Curve (P-R)**: The P-R cure is the most common metric for the efficiency evaluation of the retrieval systems. The *Precision* and *Recall* factors are defined as follows:

$$Precision = \frac{\mathcal{N}}{\mathcal{A}} \qquad , \qquad Recall = \frac{\mathcal{N}}{\mathcal{Q}} \tag{4.1}$$

  Where, $\mathcal{N}$ is the number of relevant models retrieved, $\mathcal{A}$ is the number of retrieved models and $\mathcal{Q}$ is the number of relevant models in the target dataset. An ideal $Precision$- $Recall$ curve has $Precision$ equal to *1* for all values of $Recall$ (all relevant models are retrieved before any irrelevant ones).

It is worthwhile to mention that the parameters $Precision$ and $Recall$ can also be used separately, so as to evaluate the retrieval system (e.g., the graph of $Precision$ or $Recall$ vs. the number of retrieved models).

- **Discounted Cumulative Gain ($\mathcal{DCG}$)**: The $\mathcal{DCG}$ measure weights correct results returned earlier higher than those returned later within a ranked list. This is due to the assumption that the end user does not consider that the correct results appear at the lower positions of the ranked list of the retrieved models. If $\mathcal{G}$ is a vector and $\mathcal{G}_i$ corresponds to the i^{th} element in the ranked list of the retrieved models ($\mathcal{G}_i$ has a value of 1 if the result is of the query class and 0, otherwise), $\mathcal{DCG}$ is defined recursively by Equation (4.2).

$$\mathcal{DCG}_1 = \mathcal{G}_1 \quad , \quad \mathcal{DCG}_i = \mathcal{DCG}_{i-1} + \frac{\mathcal{G}_i}{\log_2(i)} \quad i > 1 \tag{4.2}$$

Then, the final measure of $\mathcal{DCG}$ is obtained be dividing the above result by the maximum possible $\mathcal{DCG}$. The maximum value of $\mathcal{DCG}$ is obtained, if all of the first $|\mathcal{C}|$ ranked models are correctly retrieved ($|\mathcal{C}|$ is the cardinality of the query class in the dataset) [3]:

$$\mathcal{DCG} = \frac{\mathcal{DCG}_{\mathscr{k}}}{1 + \sum_{j=2}^{|\mathcal{C}|} \frac{1}{\log_2(j)}} \tag{4.3}$$

Here, $\mathscr{k}$ is the number of models in the database.

- **Nearest Neighbour ($\mathcal{NN}$):** $\mathcal{NN}$ is a scalar measure to identify whether the first retrieved object belongs to the same class as the query object or not. It is computed as the mean percentage of the closest matches, which belong to the same class as the query model; an ideal score is 100%. It is obvious that the higher the $\mathcal{NN}$, the better the performance.
- **First Tier ($\mathcal{FT}$):** Similar to the $\mathcal{NN}$ metric, the $\mathcal{FT}$ aims at appraising whether the higher ranked retrieved models are the correctly-matched ones or not. Therefore, it is defined as the average percentage of the top $\mathcal{K}$ retrieved models that are of

the same class as the query. Where $\mathcal{K} = |\mathcal{C}| - 1$ and $|\mathcal{C}|$ is the number of models in the target dataset which belongs to the query class.

- **Second Tier ($\mathcal{ST}$):** The $\mathcal{ST}$(also is known as the Bull-Eye percentage) is quite similar to the $\mathcal{FT}$ factor but it examines the $\mathcal{K} = 2 * |\mathcal{C}| - 1$ top retrieved models.
- **E-Measure: This is a** composite measure based on the *Precision* and *Recall* factors which is defined by the following formula:

$$E = \frac{2}{{}^{1}/_{\mathcal{P}} + {}^{1}/_{\mathcal{R}}} \tag{4.4}$$

Where $\mathcal{P}$ and $\mathcal{R}$ are *Precision* and *Recall* factors for top $\mathcal{K} = 32$ matched objects (first page of retrieved objects), respectively. The rationale behind the E-Measure is the fact that the user is more interested in the high ranking positions of the retrieved objects.

Using the evaluation metrics, first we aim at setting the required parameters by which the best results can be achieved. Then, the proposed descriptors will be assessed using the above-mentioned factors.

4.4. Parameter Analysis

In the following sub-sections, we will investigate the effects of the available parameters in the proposed descriptors to find their fixed values by which the best performance can be achieved. It should be pointed out that the articulated models in the McGill benchmark have been utilized so as to examine the parameter effects. The reason for selecting the McGill dataset for setting the parameter is as follows: as will be discussed later in Sections 4.5 and 4.6, the models in the McGill dataset are more complex and challenging to retrieve. Therefore, setting the utilized parameters for the McGill dataset will guarantee to get the best results for the other datasets.

4.4.1. Parameter of the SilPH Descriptor

There is only one parameter to be set in the definition of SilPH descriptor, which is number of bins in the SilPH histogram.

We have examined four different values as the number of histogram bins including 5, 10, 20 and 50. As reported in Table 4.1, increasing the number of bins results in a higher retrieval quality. But it is obvious that the greater the number of bins, the greater the computational cost of computing the similarity measure. Therefore, we have set the bin number equal to 10 as there is no meaningful enhancement in the evaluation metrics for the values of 20 and 50.

Table 4.1, The evaluation metrics for different bin numbers of the SilPH Descriptor.

Bin Number	NN	FT	ST	E	DCG
5	0.5114	0.3136	0.3906	0.3218	0.4261
10	0.7345	0.4619	0.5512	0.4420	0.6177
20	0.7392	0.4853	0.5574	04817	0.6345
50	0.7516	0.4932	0.5741	0.4940	0.6671

4.4.2. Parameters of the BOF-CDD Descriptor

We tested the three parameters of the BOF-CDD descriptor namely the dictionary size, feature-point selection criteria and the number of feature points, so as to determine their best values. In order to quantify the effect of each parameter, the evaluation factors are listed in the appropriate tables.

4.4.2.1. Size of the Dictionary

Several different numbers are chosen for the size of the dictionary during the clustering algorithm of the BOF framework. As shown in Table 4.2, when the size of dictionary is equal to 20 the best retrieval quality is obtained. The smaller size of the dictionary is another powerful point of our approach and leads to a lower number of comparisons and a higher speed of the retrieval process.

Table 4.2, Evaluation factors for different size of dictionary in BOF-CDD (With 1000 random points)

Dictionary Size	NN	FT	ST	E	DCG
5	0.7188	0.4962	0.6393	0.4784	0.8186
10	0.8250	0.5349	0.6778	0.5049	0.8304
20	0.8563	0.5455	0.6610	0.5301	0.8649
50	0.8112	0.5147	0.6101	0.5441	0.8012
200	0.5125	0.4095	0.5641	0.4312	0.7465

4.4.2.2. Feature Point Selection Criteria

Three different approaches are utilized as the feature point selection criteria as follows:

1) **Random:** The feature points are randomly selected on the surface of the models.
2) **Local Maxima:** All of the surface points holding the local maxima of charge density are chosen as the feature points.

3) **Higher-Than-Mid Density:** The surface points $\mathscr{k}$ which possess the charge density satisfying the following criterion are selected:

$$\rho_{\mathscr{k}} \geq {}^{1}\!/_{2}\left[max(\rho_i) - min(\rho_i)\right] \qquad , \qquad i = 1,2,\ldots,\mathscr{m} \qquad (4.5)$$

Where, ρ_i denotes the charge density of face i and $\mathscr{m}$ is the number of faces on the model surface. As displayed in Table 4.3, the randomly selected points show the better ability than the others.

Table 4.3, Evaluation metrics for different feature points selection criteria in BOF-CDD (Dic_Size=20).

Selection Criterion	NN	FT	ST	E	DCG
Random	0.8563	0.5455	0.6610	0.5301	0.8649
Local Maxima	0.6415	0.3198	0.4602	0.3177	0.6505
Higher-Than-Mid.	0.3581	0.2700	0.4333	0.3176	0.5414

4.4.2.3. Feature Point Number

Different numbers of randomly selected feature points are employed for matching the models. As depicted in Figure 4.3, a choice of 1000 seems to be the best one for the feature point count per model.

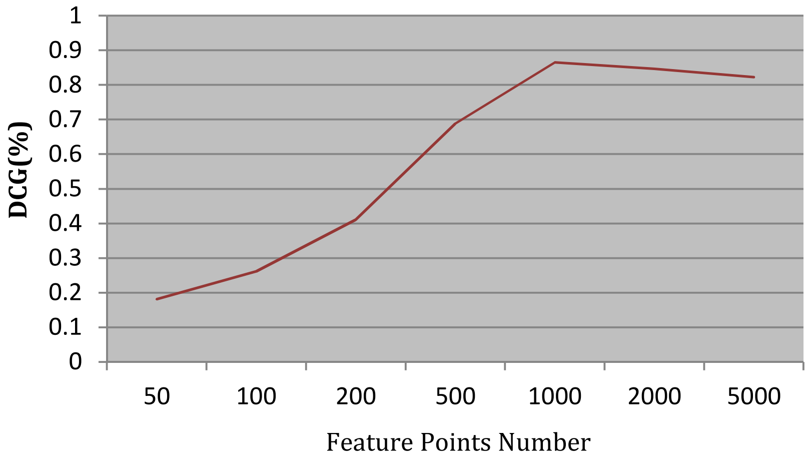


Figure 4.3, The relation between the DCG measure and number of feature points in BOF-CDD (Dic_Size=20)

4.4.3. Parameters of the DP-CS-ECD Descriptor

As detailed in Chapter 3, the number of concentric spheres is the only parameter to be set in the DP-CS-ECD descriptor. Therefore, we enclosed each model in different numbers of concentric spheres to describe the models. The resulted feature vectors are investigated to examine the retrieval abilities (see Table 4.4).

Table 4.4, Evaluation metrics for different numbers of concentric spheres in the DP-CS-ECD Descriptor.

Sphere Count	NN	FT	ST	E	DCG
5	0.7212	0.3414	0.4710	0.4012	0.6337
10	0.8563	0.4940	0.6359	0.4719	0.8172
20	0.8812	0.6352	0.8012	0.5319	0.8961
50	0.7375	0.4803	0.6433	0.4735	0.7968
200	0.4063	0.2641	0.3805	0.2879	0.5868

The data reported in the above table illustrate that value 20 is the best choice for concentric sphere count. So, we set $\mathcal{N}_{\mathcal{S}} = 20$ for the entire retrieval process.

4.5. Results on the McGill Dataset

After setting the parameters, the proposed descriptors have been applied to all the models in the McGill dataset with the aim of verifying their performance and comparing it to the well-known approaches. We compare our descriptors to the famous approaches for which the comparison data on the McGill dataset are available. These approaches are: Shape Distribution (D2) [58], Spherical Harmonic Descriptor (SHD) [47], Curvature-Based (Mean Curvature) [201], Light Field Descriptor (LFD) [27], 3D Poisson-Based (PH) [73] and the Clock Matching-Based Approach (MDDS-CM-BF) of Lian *et al*. [15]. The LFD had the best retrieval ability compared with the other 12 descriptors in [159]. In addition, the MDS-CM-BOF descriptor is one of the state-of-the-art approaches which, owing to a beneficial matching scheme (clock matching), achieved a very good ranking among the participants of SHREC’11 contest [19]. A detailed description of these approaches can be found in Chapter 2. Figure 4.4, depicts the *Precision-Recall* curve for these methods along with those obtained by our descriptors.

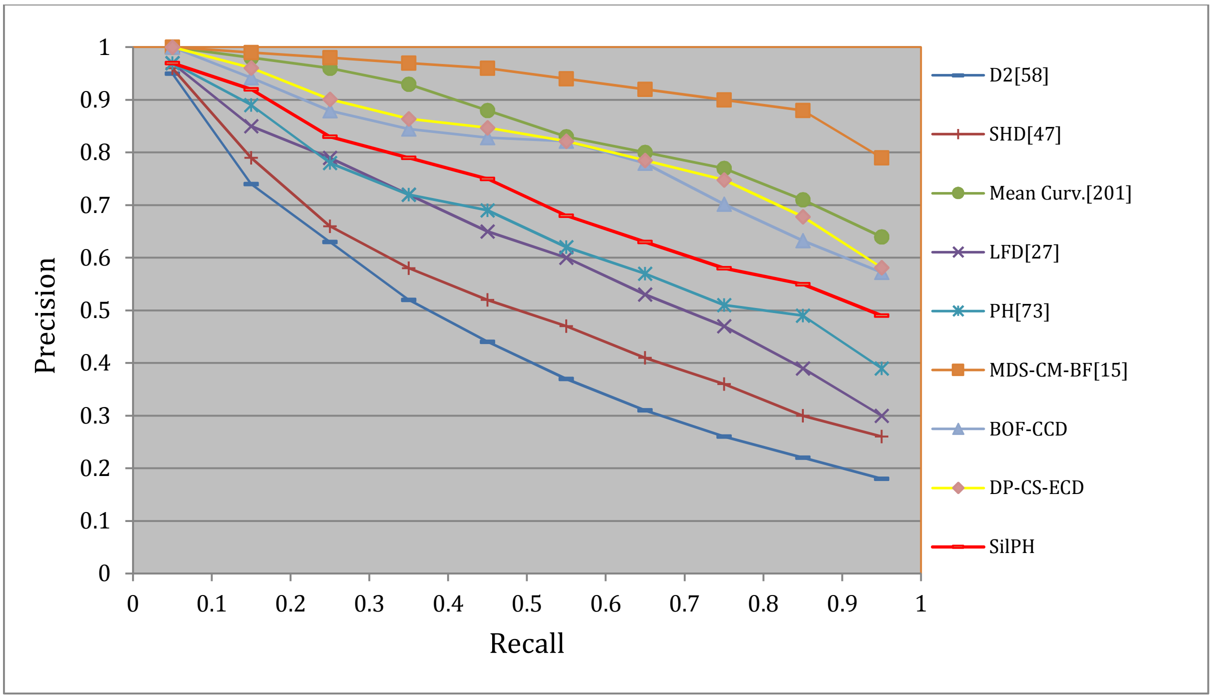


Figure 4.4, the P-R curve for our and six other well-known approaches.

As depicted in Figure 4.4, our methods distinctly outperform four other methods, but the MDS-CM-BOF descriptor is still the best by far especially for the higher values of Recall. It is important to note that although the MDS-CM-BOF shows the highest quality it cannot be utilized for partial matching, since it is a view-based approach. Furthermore, because of its special kind of matching scheme, it is not fast enough to retrieve similar models efficiently (its matching scheme is quite time-consuming). The curvature-based approach, on the other hand, is a histogram-based approach which performs better than our approaches. Since it only uses the local information to extract the signature for each point, it is noticeably sensitive to noise. Figure 4.5 depicts the effect of noise on the mean-curvature descriptors of a sample model. It shows that any change on the surface of model considerably affects its curvature which in turn affects the model descriptor. Additionally, the SilPH descriptor proposed in this work shows more discriminative ability than its 3D counterpart (descriptor PH in the above diagram). But, the SilPH descriptor is slower than the 3D PH one as the SilPH employs a time-consuming process of multi-view matching while the 3D PH computes the similarity measure between two models via a one-stage histogram comparison.

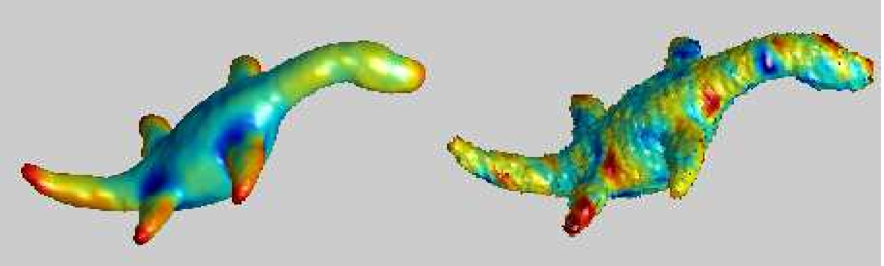


Figure 4.5, Mean Curvature descriptor, left: normal model, right: noisy model

Among descriptors proposed in this thesis, the charge distribution-based approaches (CD-BOF and DP-CS-ECD) show better retrieval quality than the Poisson-based one (SilPH). This superiority is reasonable as the charge-based descriptors take into account both local and global characteristics of the model surface, while the SilPH descriptor only encodes the global features of the models. On the other hand, the DP_CS-ECD descriptor outperforms the CD-BOF one. The reason for that may be related to the fact that the DP-CS-ECD is constructed on the canonical form of the models which makes it more applicable to deformable models in the dataset.

4.6. Results on SHREC'11 Dataset

In order to evaluate more precisely, we have applied the proposed descriptors to the 600 models of the SHREC'11 models and compared their performance to those of all the participants of the non-rigid watertight track of the SHREC'11 contest [19]. The contestants contain a variety of state-of-the-art approaches viz. Laplace Spectra approach of Reuter (Shape DNA) [202], Clock-Matching approach of Lian and Godil (MDS-CM-BOF) [15], Heat Kernel Signature utilized by Sipiran and Bustos (HKS) [203], fused approach of Geodesic Distance Matrix and MeshSIFT of Smeets *et al*. (GDM-MeshSIFT) [204], Densely-Sampled Local Visual Feature approach of Tabia and Daoudi (Patch-BOF) [205] *etc*. More descriptions about these approaches can be found in Chapter 2 and the report paper published on the SHREC'11 contest [19]. It should be noted that six among nine participants of SHREC'11 contest have utilized the BoF framework which confirms the ability and popularity of BoF in 3D retrieval domain. The result of the comparison is shown via the *Precision-Recall* curve in Figure 4.6. Furthermore, five other related standards metrics are shown in the Table 4.5.

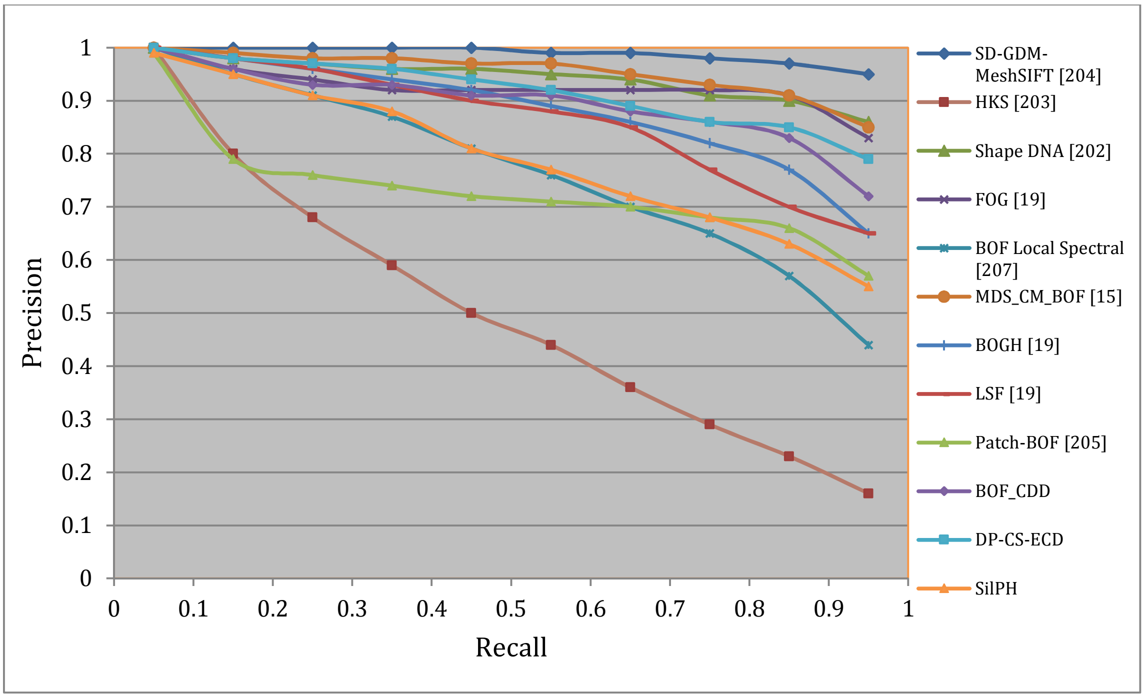


Figure 4.6, Precision-Recall curve of the proposed descriptors and the approaches participated in SHREC'11 contest.

The data reported in Figure 4.6 and Table 4.5 illustrate that the hybrid approach of SD-GDM-MeshSIFT outshines other approaches by far. The particular forte of SD-GDM-MeshSIFT, which gives it the best rank among all of the competitors, relates to its combined nature. That is, the approach considers both local and global characteristics of the models by fusing a global feature method (SD-GDM) with a local feature method (meshSIFT) to describe the models. Our electrical charge-oriented descriptors, BOF-CDD and DP-CS-ECD, are among the best approaches and their performances are quite comparable to the state-of-the-art ones. The SilPH descriptor on the other hand, gained lower rank among our approaches since it only considers the global features of the models.

It is worthwhile to point out that in general, the proposed descriptors show better discriminative ability for the SHREC'11 dataset than the McGill benchmark. It is also the case for other approaches such as the MDS-CM-BOF approach of Lian and Godil. This difference is due to the variation of the models in the two datasets and can be justified by the following reasons:

Table 4.5, Evaluation metrics of the proposed descriptors and the approaches participated in SHREC'11 contest.

Method Name	NN	FT	ST	E	DCG
FOG	0.968	0.881	0.946	0.696	0.959
BOF-Local Spectral	0.965	0.672	0.803	0.579	0.889
MDS-CM-BOF	0.995	0.913	0.969	0.717	0.982
BOGH	0.993	0.811	0.884	0.647	0.949
LSF	0.995	0.809	0.879	0.643	0.948
Shape DNA	0.997	0.915	0.957	0.705	0.978
HKS	0.837	0.406	0.497	0.353	0.730
GDM-MeshSIFT	1.000	0.972	0.990	0.736	0.996
Patch-BOF	0.748	0.642	0.833	0.588	0.837
SilPH	0.935	0.655	0.785	0.582	0.884
DP-CS-ECD	0.982	0.887	0.914	0.702	0.952
CD-BOF	0.965	0.833	0.896	0.668	0.944

- **Intra-Class Dissimilarity**: The models of each class in the McGill dataset have more diversity than the SHREC'11 ones. The Four-Limb class of the McGill for instance, includes 30 various poses of different animals ranging from Cat, Dog and Horse to wild animals such as Tiger and Boar. Each of these models has its own class in the SHREC'11 dataset. Generally speaking, each class of the SHREC'11 dataset includes an identical model in different poses, which are quite similar, while the models of the same class in the McGill dataset are not necessarily alike.
- **Inter-class Similarity**: Some distinct classes of the McGill dataset (e.g. Crab and Octopus classes) include models quite similar in structure and topology. This similarity is also present in two classes of Fish and Dolphin models.

Although the SHREC'11 dataset contains a set of various categories (e.g. Men and Women or Dog1 and Dog2 in Figure 4.2) which possess similar overall appearances and different details and/or topological structures, the above mentioned reasons make the models in the McGill dataset more challenging to match and retrieve.

We have shown in Figure 4.7, the *Precision-Recall* curve for six sample classes using the DP-CS-ECD descriptor as it has shown best retrieval ability among the three proposed descriptors. These classes include challenging groups of models such as similar classes of Men and Women.

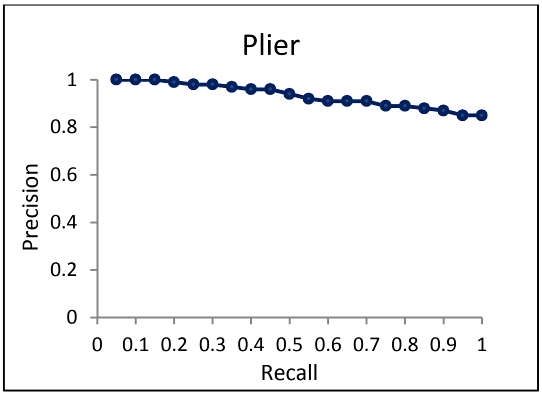
The Men and women are the problematic classes for which the retrieval quality is not satisfactory enough. The reason for this issue is related to the similarity of both local and global features of the models in these two classes. Moreover, the similar models in other classes (viz. Gorilla and Santa) deteriorate the results so that the retrieved models for different Man queries include several incorrect models from these similar classes.

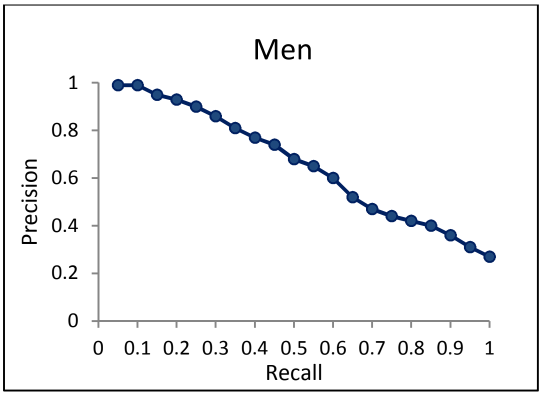
The Plier and Paper models on the other hand, are the classes for which the DP-CS-ECD has shown the highest discriminative quality. These two classes have quite different structure so that the DP-CS-ECD is able to differentiate them easily.

4.7. Partial Matching Ability on SHREC'07 Dataset

Partial matching is a quite challenging field in 3D model retrieval domain, in which the aim is to find and retrieve the models from a target dataset that share similar subparts with a sample query.

Among the three proposed descriptors, the BOF-CDD is expected to be able to support partial matching as it considers the local characteristics of the model surface. The other proposed electrical charge-oriented descriptor, DP-CS-ECD, utilizes the same information to describe the models, but it is obvious that the amount of electrical charge on the layers between the spheres for two partial-similar models is quite different. Figure 4.8 displays the different DP-CS-ECD descriptors of a mixed query model (query number=28 in the query set of Appendix B) and two of its partial-similar models (Plier and Octopus).





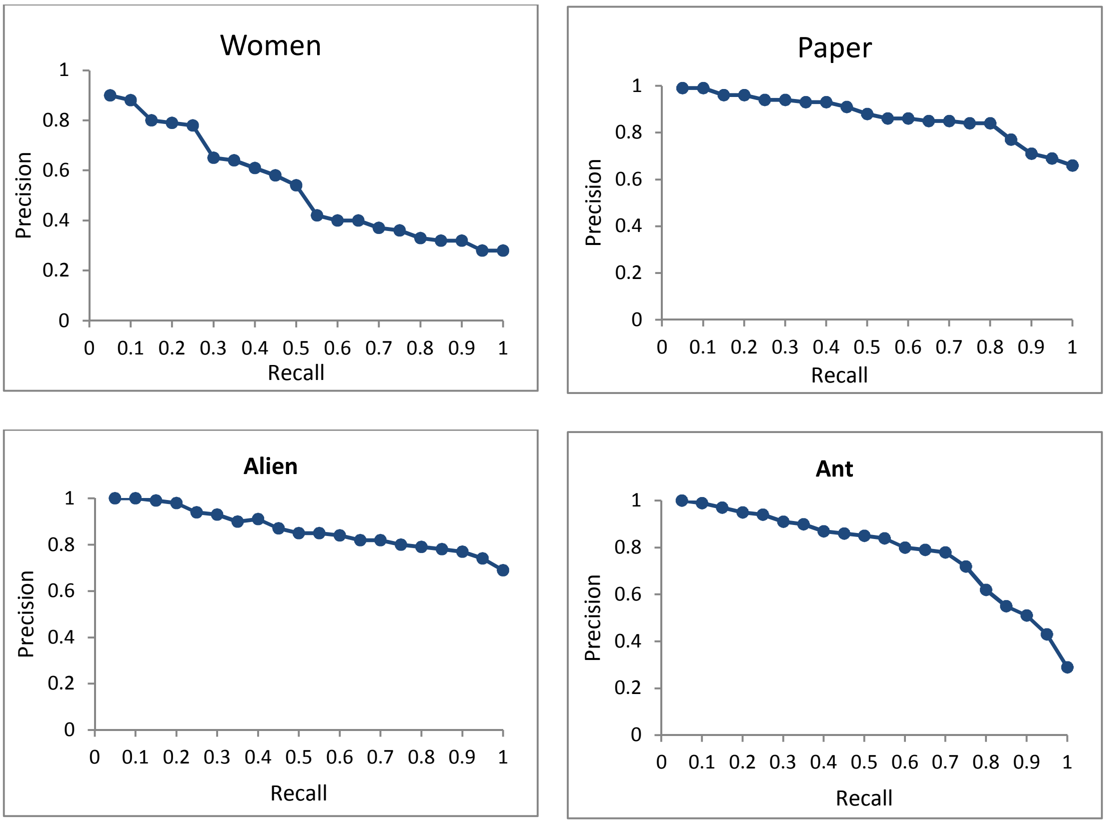


Figure 4.7, *Precision-Recall* curve for six sample classes of SHREC’11 dataset using DP-CS-ECD descriptor

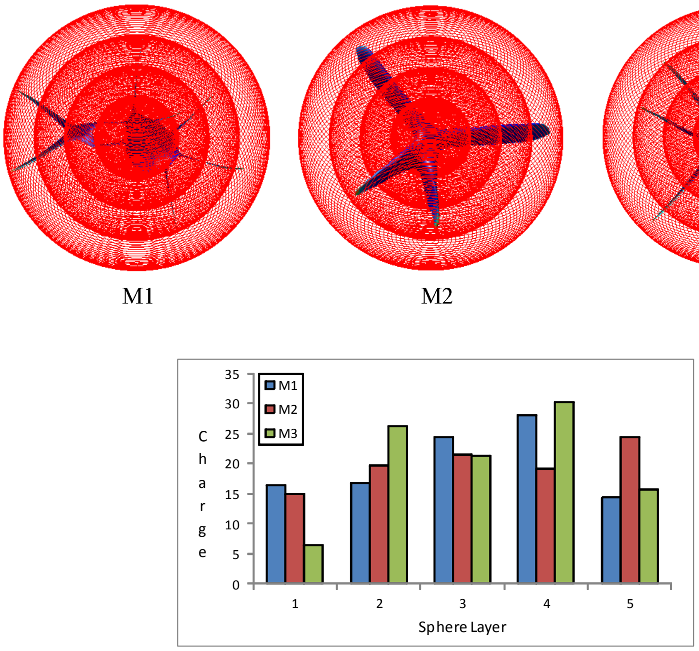


Figure 4.8, the DP-CS-ECD descriptors for partial similar models ($\mathcal{Q}$=100 and $\mathcal{N}_s$=5)

The SilPH descriptor, on the other hand, is an inappropriate descriptor of partial matching as it uses global features of the silhouettes for describing the models. As depicted in Figure 4.9, in a sample set of partial-similar models, the extracted silhouettes from similar viewpoints have quite different SilPH descriptors, which lead to an inability of the SilPH descriptor in partial matching.

In the following section we first give a concise overview of some available approaches for the partial matching of 3D models and then the ability of the BOF-CDD descriptor to support partial matching is tested on the models in the SHREC'07 dataset.

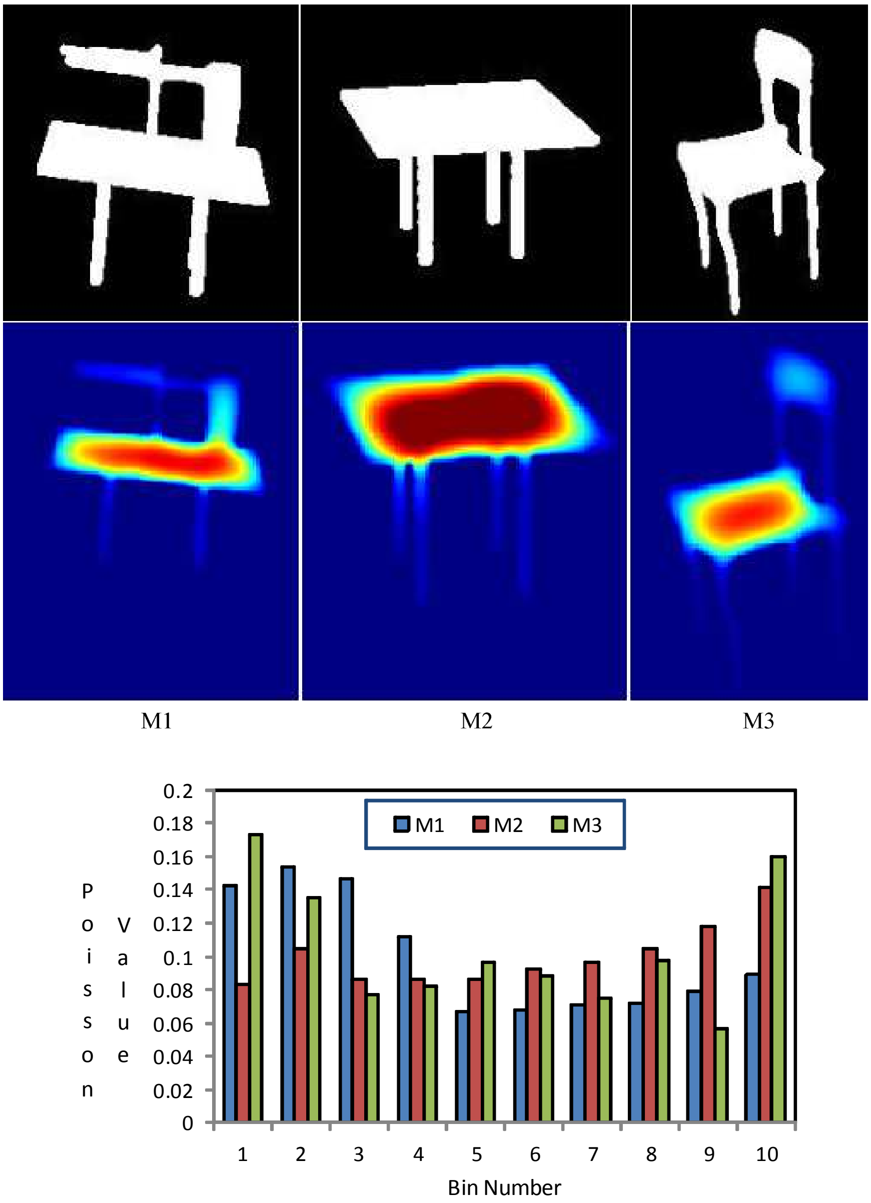


Figure 4.9, First row: the silhouette extracted from identical viewpoints for the partial-similar models. Second row: solution of the views to the Poisson equation and Third row: SilPH Descriptor of the views.

4.7.1. Related works

Despite the existence of dozens of approaches for global matching of 3D models, there are only a few methods proposed in the literature, which tackle the challenging issue of partial matching. In the sequel a brief discussion on some of these approaches will be presented.

Toldo *et al.* [206] proposed a multi-level BOF framework so as to describe the models. Their process starts by segmenting the models using the Shape Index (SI) descriptor. Then, four different shape descriptors namely Shape Index, Radial Geodesic Distance, Normal Direction and Geodesic context are employed to describe the extracted semantic segments. Based upon the type of descriptor and bin numbers in the BOF framework, several histograms are built to describe the models. The authors claim that their approach implicitly encodes partial matching since corresponding segments are likely to belong to the same bin of the signature histogram.

Recently, Guillaume employed a spectral oriented descriptor to deal with the partial matching problem [207]. His approach starts by selecting a set of random seed points on the surface of models. Then the Lloyd relaxation algorithm [208] is applied to the seeds to distribute them uniformly. These seeds are used to define patches centred on each seed. The method computes the Fourier spectra of the patch by projecting the geometry on the eigenvectors of the Laplace-Beltrami operator (LBO) and uses it as the local descriptor of the patches. Finally, a BoF framework augmented with the visual expression technique [95] is employed to support partial matching in a spatial-sensitive manner.

Differently, Tierny *et al.* [209] proposed a structural approach for partial matching using Reeb graphs. In their approach, Reeb Pattern Unfolding (RPU), the graph is constructed using the Geodesic distance and is augmented with the geometrical information of each node which is obtained by analysing the Gaussian curvature in each vertex of the mesh. The method first segments the models based on the graph and then the extracted parts are described based on a parameterization technique so as to be utilized in a partial matching framework (See Figure 4.10).

Another Reeb graph-based approach for partial matching has been proposed by Biasotti *et al.* [77]. It uses the Extended Reeb Graph (ERG) which is a combination of the multi-resolution Reeb graph (for structural information) and the Spherical harmonics of the nodes (geometrical information). In order to perform partial matching, the authors have adopted a directed attributed graph matching method to find the maximum common sub-parts between two ERGs.

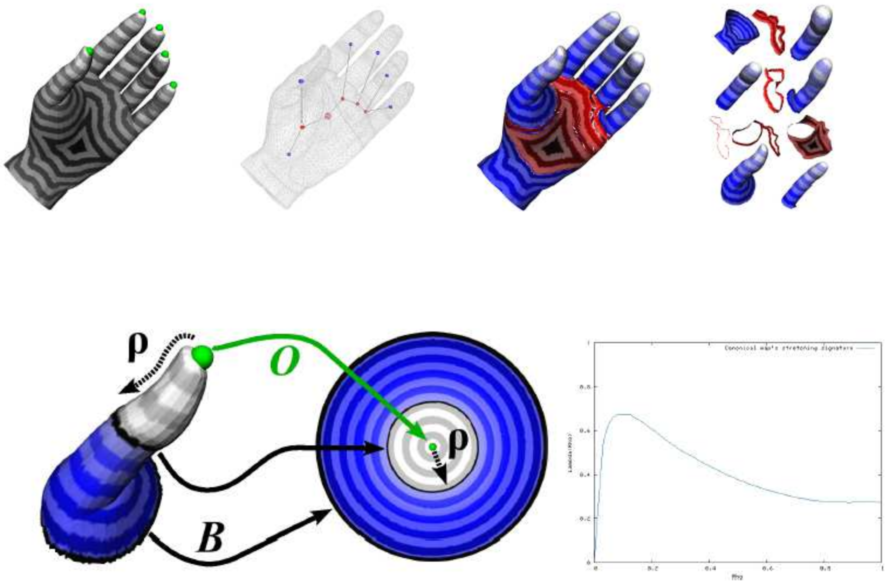


Figure 4.10, RPU approach of Tierny *et al.* [209]. The First row shows the segmentation process of a hand model into its Reeb charts and the Second row displays the Reeb chart unfolding process of its thumb.

Finally, Cornea *et al.* [82] introduced a skeleton graph-based approach for partial matching of 3D models. Their approach, as detailed in Chapter 2, is the extended version of skeleton extraction approach of Sundar *et al.* [80]. The skeleton graph is extracted using a generalized potential field [83]. The graph matching step is then carried out by the Earth Mover Distance (EMD) which naturally supports partial matching.

In the following section we compare the BOF-CDD descriptor to the above-mentioned approaches using the models of SHREC'07 dataset.

4.7.2. Partial Matching Results

All the models available in the query set of the partial matching track of SHREC'07 have been fed to the proposed retrieval system using the BOF-CDD descriptor. The query set contains 30 models which are obtained by merging or removing several subparts of models belonging to the SHREC'07 watertight dataset. Appendix B shows the models in the query set. In order to evaluate the performance of the BOF-CDD descriptor in terms of supporting partial matching, the results are compared to those of the approaches stated in the previous section. Figure 4.11 depicts the top ten retrieved models for three sample queries using the BOF-CDD descriptor. As displayed in this figure, the retrieved models contain relevant, marginally-relevant and even irrelevant models. The presence of irrelevant models among the top ten results reveals the weakness of the underlying descriptor in matching of partial-similar objects.

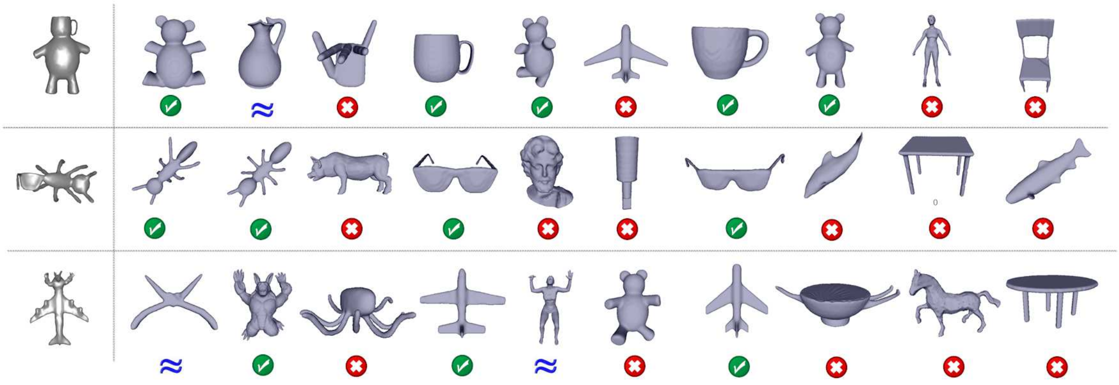


Figure 4.11, Three sample queries and associated ten retrieved models.

In order to evaluate the ability of the proposed approach quantitatively, we utilized the $\mathcal{NDCG}$ metric to compare it to the state-of-the-art 3D partial matchers. So, in order to compute $\mathcal{NDCG}$ the values of 2, 1 and 0 are assigned to the parameter $\mathcal{G}_i$ for the models which are highly-relevant, marginally-relevant and non-relevant, respectively (see Equation 4.2).

As illustrated in Figure 4.12, the BOF-CDD is not able to produce pleasing results in comparison to the other approaches for retrieving partial similar models. The weakness of BOF-CDD in terms of supporting partial matching can be explained as follows:

When a major change occurs in the structure of a model (e.g. attaching more sub-parts to the model or merging two models so as to generate a new mixed model), the altered parts accumulate some portions of distributed charge which will affect the amount/density of charge on the unchanged parts. Due to local property of BOF-CDD, one may expect to see similar charge distribution on the faces of identical sub-parts of partial-similar models. But since the total amounts of charge on these sub-parts are not the same, their faces on different models do not possess the exact similar amount of charge.

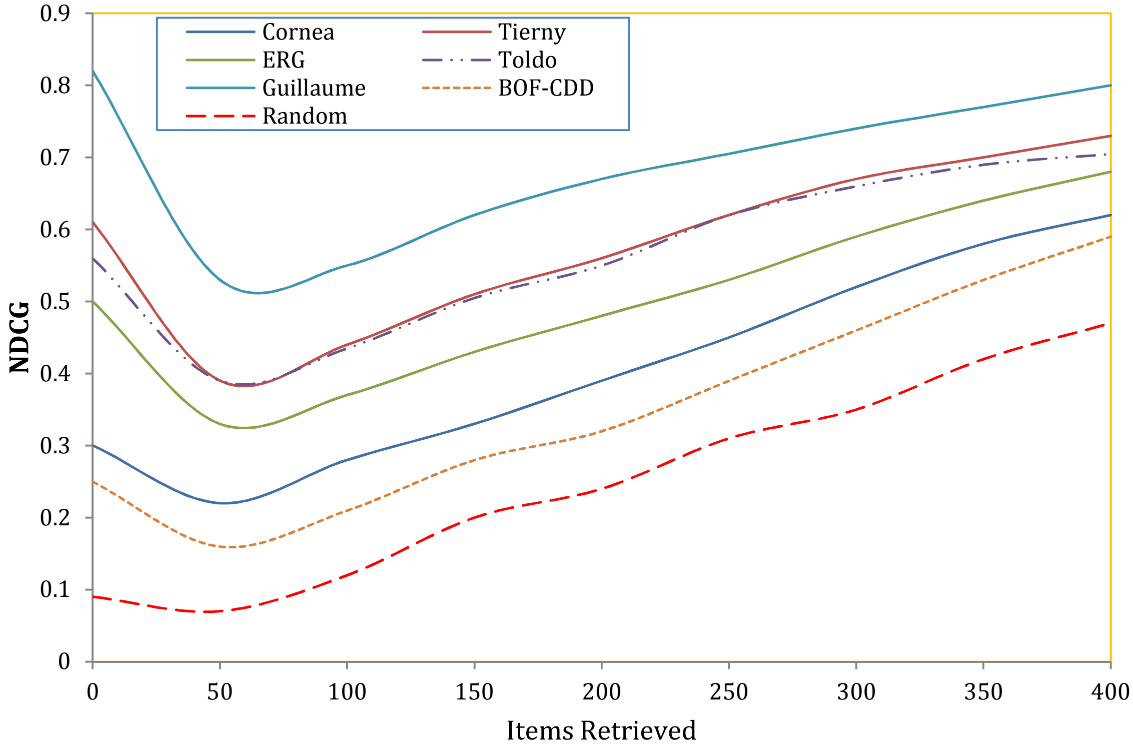


Figure 4.12, NDCG plot for BOF-CDD and five other approaches.

Note that the distribution of charge on the similar sub-parts follows the same pattern of distribution. Consider a finger model for instance, no matter whether the finger is connected to only a hand model or to a whole human, the amount/density of charge gradually declines as the faces get further from the fingertip. But the charge amount/density on the similar faces of the finger is not necessarily the same. As a result, a finger in various contexts cannot be correctly matched via the BOF-CDD descriptors.

Aware of this, we can conclude that the global features have a predominant role in the forming of BOF-CDD descriptor which results in inadequate ability of the BOF-CDD descriptor to match the models having similar sub-parts.

4.8. Robustness

In this section we examine the robustness of the proposed descriptors against a variety of surface changes such as noise, mesh tessellation and deformation.

4.8.1. Robustness to Noise

A variety of distortions such as noise and small perturbations may be seen on the surface of the models. So, the ideal descriptor should be insensitive to this class of distortions. Here we try to find how well the proposed descriptors are robust to noise.

SilPH Descriptor: As stated in Chapter 3, the SilPH signature is a global descriptor and is computed using the average time for particles to hit all of the boundaries of the extracted silhouette views. Therefore, it is expected to be insensitive to noise as each pixel signature is generated by the contribution of all boundary points.

Figure 4.13 depicts the SilPH descriptor for the normal and noisy versions of a Teddy model. As expected, the SilPH histograms of the same views of the two models are quite similar. It supports our claim about the robustness of the SilPH descriptor against noise.

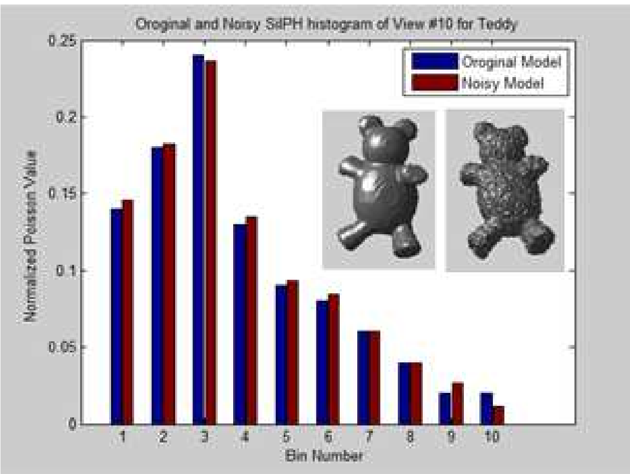


Figure 4.13, The SilPH histogram for identical view of the original and noisy versions of a Teddy model

Electrical Charge Distribution: since the foundation of both BOF-CDD and DP-CS-ECD descriptors is charge distribution, we examine how well the Electrical Charge Distribution is robust to noise so as to investigate the robustness of these descriptors. To do so, we distribute a certain amount of charge on the surface of a human model with different levels of noise (Noise Level or NL is the ratio of largest displacement to the longest edge of the object's bounding box). Figure 4.14 illustrates that the charge distribution on the surfaces of the noisy human models are quite similar to the normal one. This similarity is due to the fact that distribution of charge on each face is contributed by all the faces of the model. Consequently, noise and other small surface defects do not have considerable effects on the charge distribution. It is obvious that the higher the noise level, the greater the change in the distribution so that For NL=1.0 the distribution of charge (yellow bars of histogram in Figure 4.14) is not as satisfactory as for the lower levels of noise.

This class of robustness reveals the fact that the global features, which are encoded by the charge distribution-based descriptors, play a more critical role than the local features.

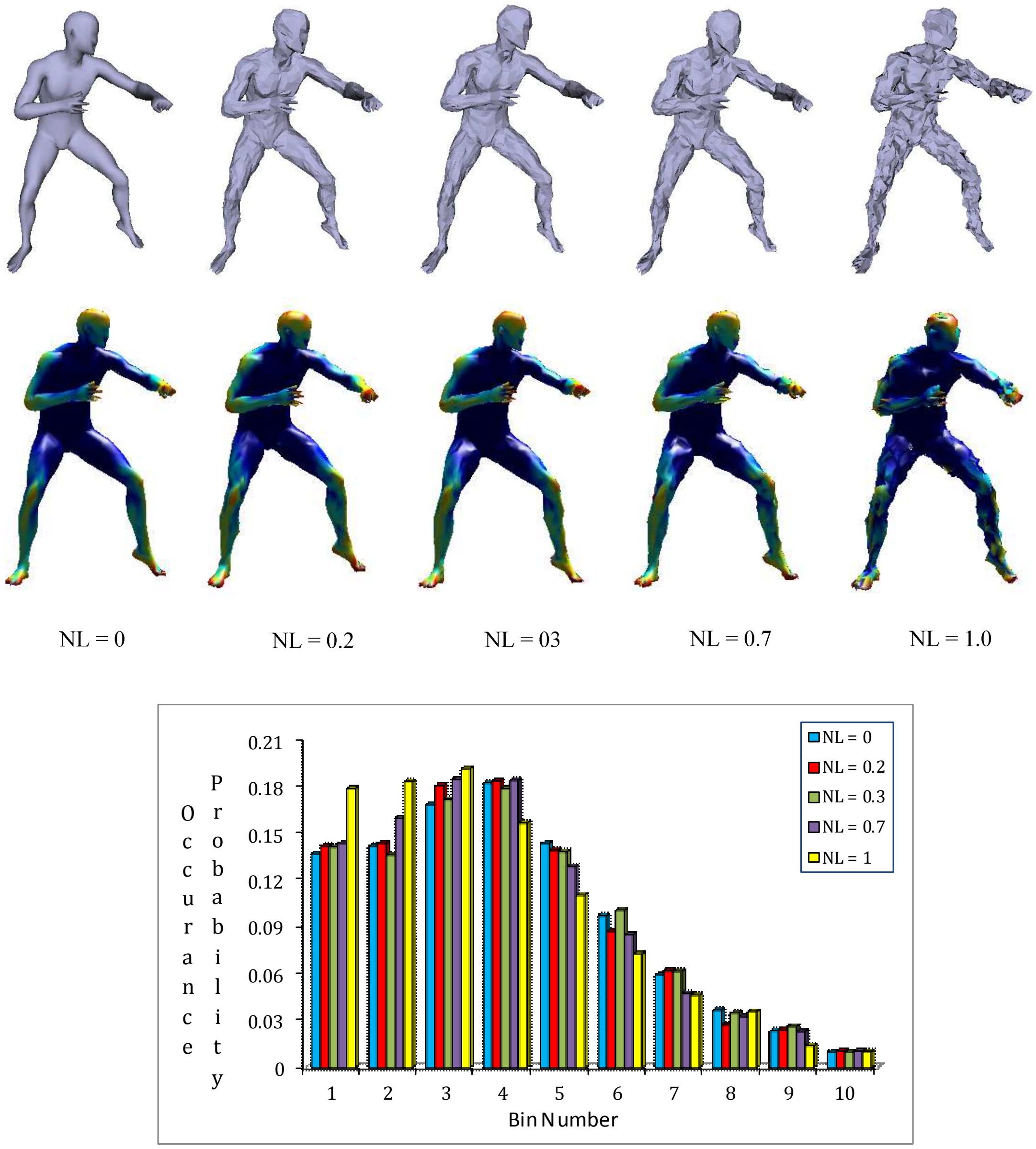


Figure 4.14, Distribution of electrical charge on the surface of a model with different levels of noise

4.8.2. Robustness to deformation

The ideal shape descriptors should be independent of the model pose to be able to effectively match and retrieve different models of the same class. In the paragraphs that follow, we examine experimentally the robustness of the proposed descriptors to pose deformations.

SilPH: the SilPH descriptor is expected to be relatively insensitive to mesh deformations as it utilizes the entire boundaries of the silhouettes. To examine it practically, two silhouette views of three different poses of a human model have been captured from similar viewpoints. The SilPH histograms of these silhouettes in Figure 4.15 illustrate that, although the histograms associated to view#1 of the models are comparable, it is not the case for those captured from view#15. Generally speaking, the robustness of the SilPH descriptor against mesh deformation depends on the amount of deformation of the model. That is, the more deformation in the model poses, the less robust the descriptor.

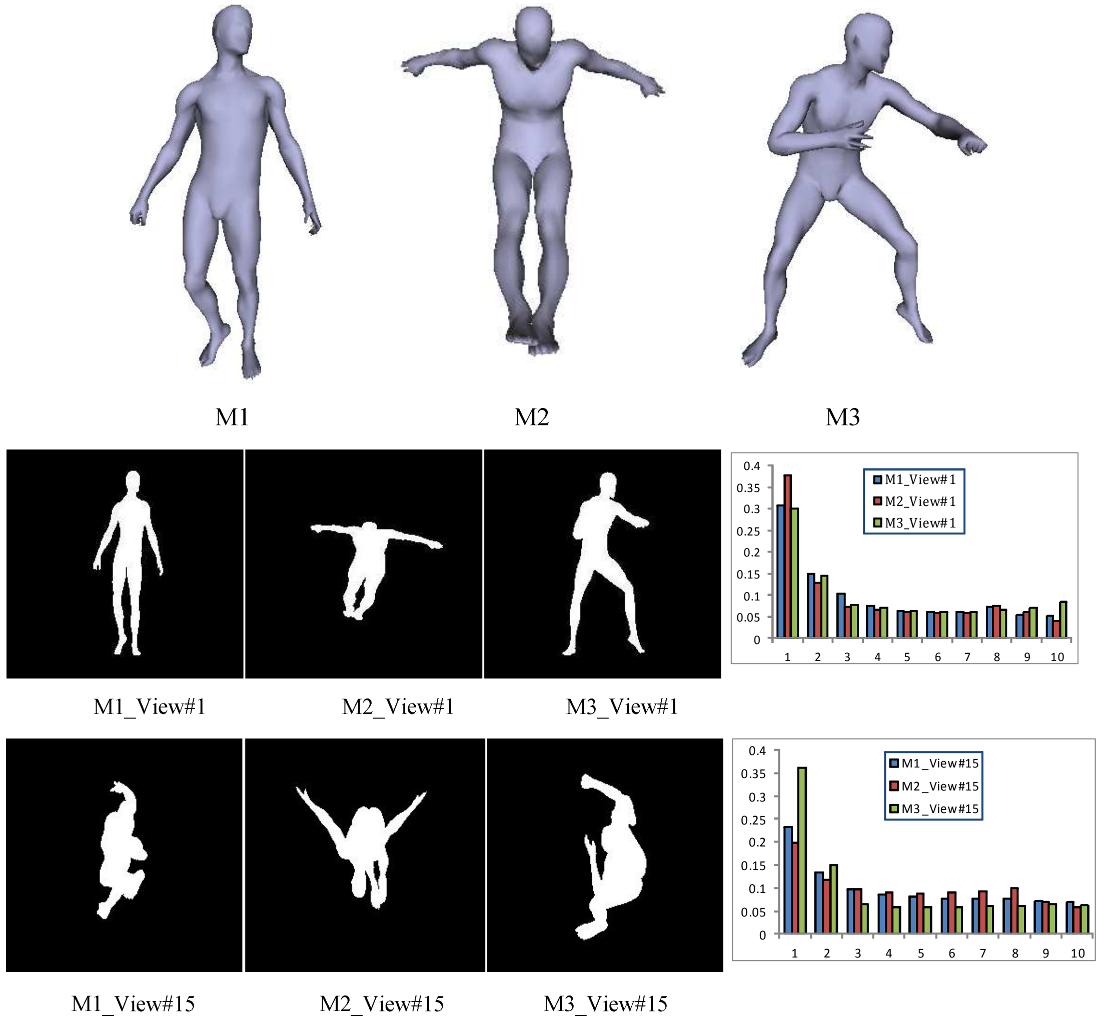


Figure 4.15, the SilPH histogram of human model in deferent poses.

Electrical Charge Distribution: It is obvious that the DP-CS-ECD is insensitive to pose deformations as it utilizes the canonical form representation of the models. In order to check this class of robustness for BOF-CDD descriptor we have shown the BOF-CDD descriptors of a human model in four different poses. The resemblance of descriptors for the human models reveals the insensitivity of the BOF-CDD against pose deformations (see Figure 4.16).

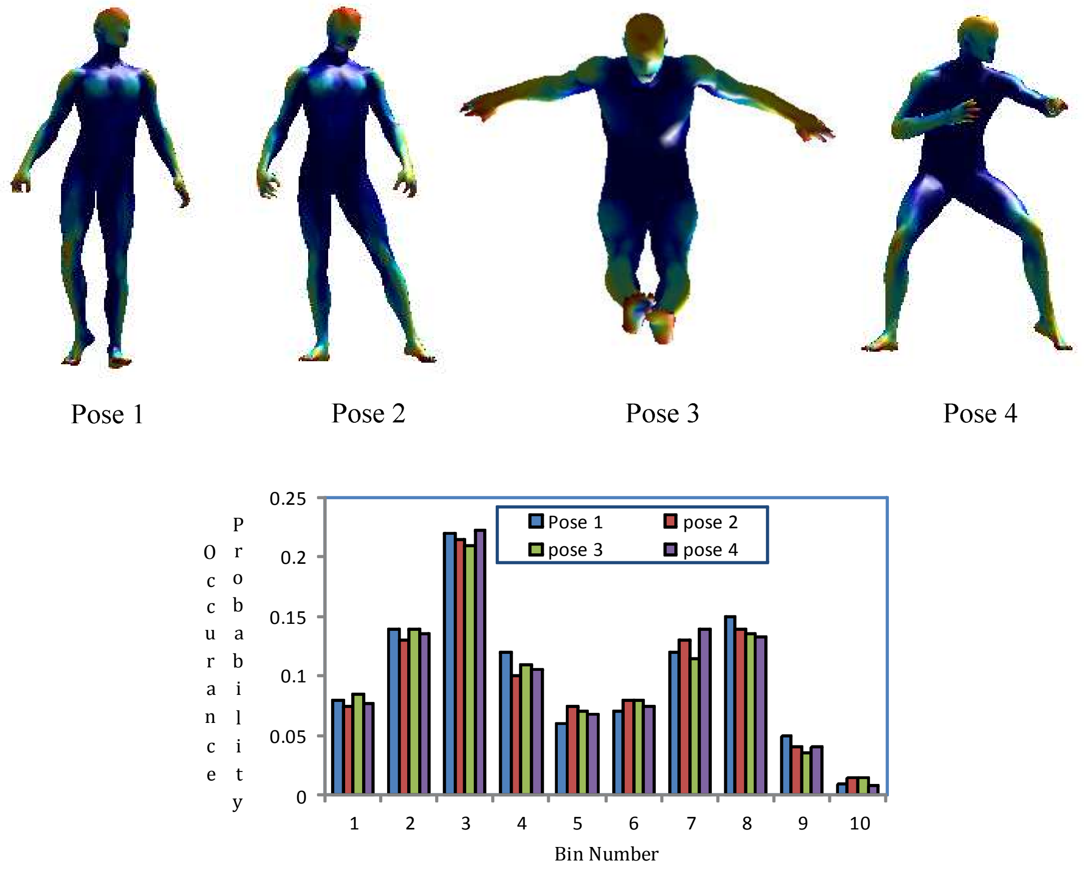


Figure 4.16, BOF-CDD for various poses of a human model.

4.8.3. Sensitivity to Mesh Tessellation

Working with lower resolution models (models with a lower number of faces) can result in achieving higher efficiency of the retrieval system. On the other hand, if the underlying shape descriptor is robust to the mesh tessellation, this efficiency is obtained without losing the effectiveness of the system. In the following sub-sections, In order to investigate the characteristics of the proposed descriptors more precisely, we examine their insensitivity to the mesh tessellation.

SilPH Descriptor: As detailed in Chapter 3, the SilPH descriptor is defined on the extracted silhouettes which are not affected by the mesh tessellation process. Therefore, the solution of these silhouettes to the Poisson equation is expected to be similar. One can see in Figure 4.17 that the extracted SilPH descriptors from similar viewpoints of a model in various resolutions are almost identical. Consequently, it is reasonable to conclude that the SilPH descriptor offers a resolution-independent signature which results in a faster retrieval process.

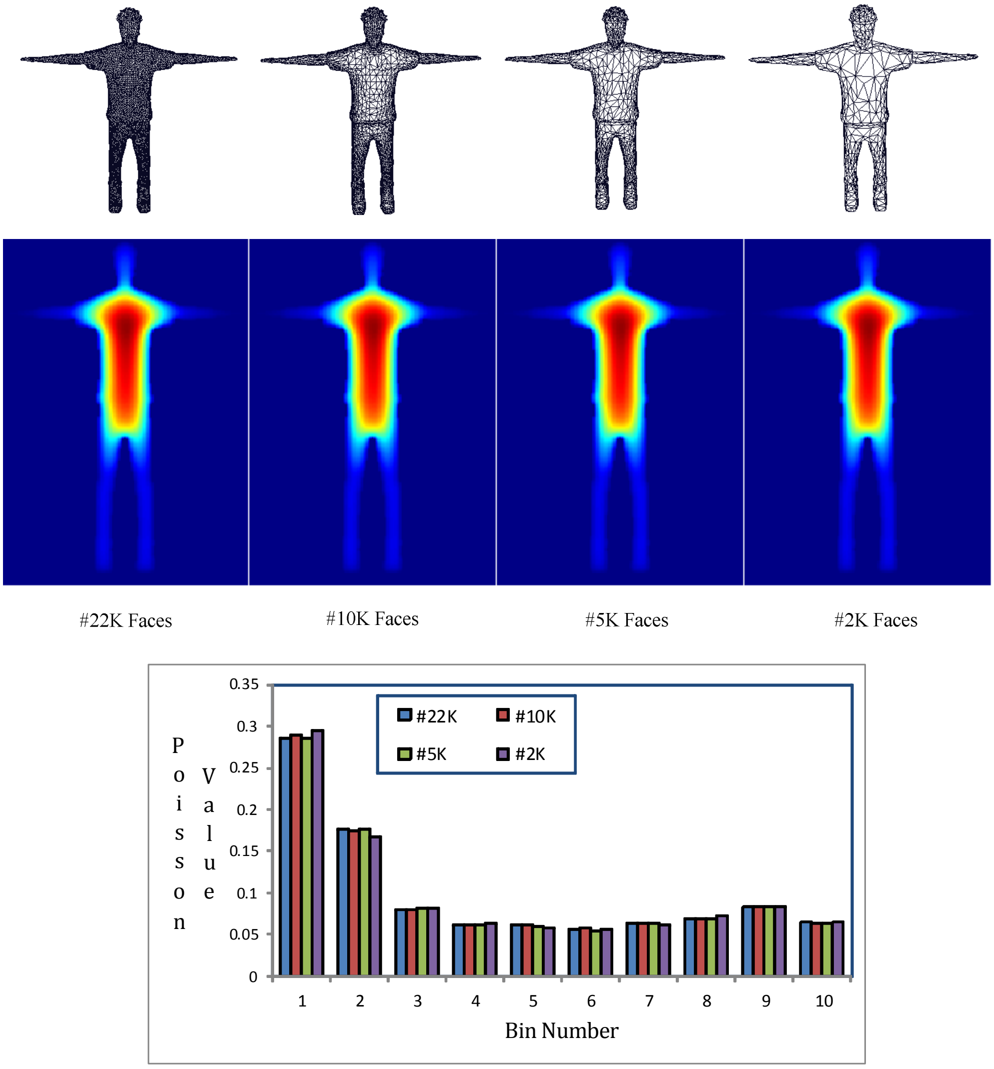


Figure 4.17, The SilPH descriptors along with the solution of their similar views of a human model in various resolutions to the Poisson equation

Electrical Charge Distribution: We have theoretically claimed in Chapter 3, that the distribution of electrical charge is not sensitive to the mesh resolution. In order to support our claim practically, we distribute electrical charge on the surface of an identical human model at four different levels of resolution, viz. 10K, 5K, 3K and 1K faces. As shown in Figure 4.18 the charge distribution on the surface of the models is completely independent of the mesh tessellation. As mentioned in Chapter 3, this is because of the fact that when the area of a triangular face gets larger (due to simplification), the amount of accumulated charge increases by the same ratio. Therefore, the charge density of the face remains the same. This makes the proposed electrical-charge-oriented descriptors tessellation-insensitive.

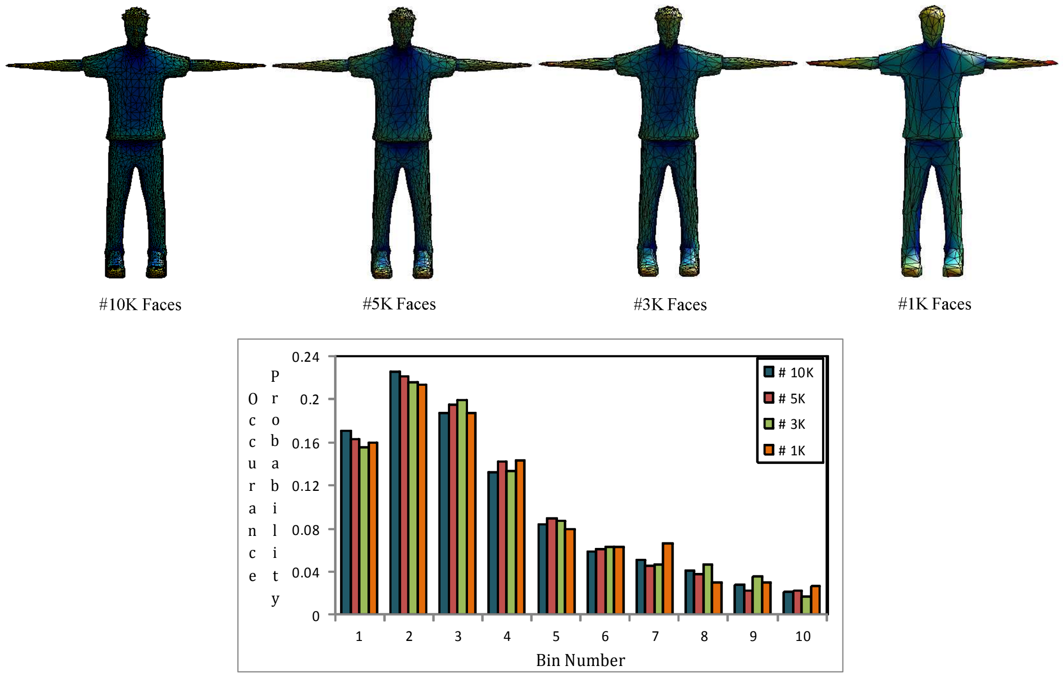


Figure 4.18, distribution of electrical charge on the surface of a human model in various resolutions.

4.9. Complexity

In this section, the computation costs of the proposed descriptors along with the required time for retrieving similar models from the target datasets are discussed.

SilPH: Numerical solution to the Poisson Equation requires $\mathcal{O}(\mathcal{N}_p{}^2)$ for a silhouette view having $\mathcal{N}_p$ pixels [192]. Additionally, the histogram construction for each view takes $\mathcal{O}(\mathcal{N}_p{}^2)$. So, the total time complexity of SilPH computation is $\mathcal{O}(\ \mathcal{N}_p{}^2 + \mathcal{N}_p{}^2) = \mathcal{O}(2 * \mathcal{N}_p{}^2) = \ \mathcal{O}(\mathcal{N}_p{}^2)$. On the other hand, assuming that $\mathcal{N}_{\mathcal{V}}$ projected views are extracted to represent a 3D model, comparing two 3D models would cost $\mathcal{O}(\mathcal{N}_{\mathcal{V}})$. If a database contains $\mathcal{K}$ models, then matching a 3D model query with the database would cost $\mathcal{O}(\mathcal{K}\mathcal{N}_{\mathcal{V}})$.

BOF-CDD: For a model having $\mathcal{N}$ faces, computation of simulated distribution of electrical charge costs $\ \mathcal{O}(\mathcal{N}^2\ )$ [171]. Additionally, the dictionary construction of the BOF process via K-Means clustering takes $\mathcal{O}\ (\mathcal{N}_c * \mathcal{N}_{Tot} * \ no\_iter)$; $\mathcal{N}_{Tot}$ is the total number of selected faces on the surface of all the models, $\mathcal{N}_c$ is the number of clusters (20 in our implementation) and no_iter in our work is 20. Finally, the histogram construction for describing each model requires $\mathcal{N}_c * \mathcal{N}_{Tot}$ operations.

It is worthwhile pointing out that, since the dictionary is created in the offline-phase, its time complexity does not affect the descriptor construction in the online-phase.

Therefore, the time complexity of descriptor generation for each model consists of the complexity of the charge distribution simulation and the histogram construction, which will be $\mathcal{O}(\mathcal{N}^2 + \mathcal{N}_c * \mathcal{N}_{Tot})$.

DP-CS-ECD: The canonical form calculation and the charge density distribution of the models take $\mathcal{O}(\mathcal{N}^2 * no\_iter)$ and $\mathcal{O}(\mathcal{N}^2)$ respectively, where $\mathcal{N}$ is the number of faces and no_iter is 10 in our implementation. Extraction of dense patches on the surface of models costs $\mathcal{O}(\mathcal{N})$. So, the overall complexity of DP-BOF-ECD descriptor computation is $\mathcal{O}(10\mathcal{N}^2 + \mathcal{N}^2 + \mathcal{N}) = \mathcal{O}(11\mathcal{N}^2 + \mathcal{N}) = \mathcal{O}(\mathcal{N}^2)$.

Table 4.6 reports the computation and retrieving times for the proposed descriptors on the target datasets running on a HP laptop with Intel 2.2GHz CPU. As illustrated in this table, the SilPH descriptor is the slowest one as it should generate and compare 60 silhouettes per model. Among the charge distribution-oriented descriptors, the computation time of DP-CD-ECD is slightly more than that of BOF-CDD. This is due to extra pre-processing phase of canonical form extraction which takes around four seconds for a 5000-face model. Although the dictionary creation of BOF-CDD is quite a time-consuming task (about 200 seconds), it is performed in the off-line phase and does not slow down the matching and retrieval tasks in the on-line phase. So, the retrieving times of DP-CD-ECD and BOF-CDD are almost similar on the both McGill and SHREC'11 datasets.

Table 4.6, Complexity of the proposed descriptors

Descriptor Name	Computation Complexity	Computation Time(Sec.)	Retrieving Time on McGill(Sec.)	Retrieving Time on SHREC'11(Sec.)
SilPH	$\mathcal{O}({\mathcal{N}_p}^2)$	7.1	21.1	26.4
BOF-CDD	$\mathcal{O}(\mathcal{N}^2 + \mathcal{N}_c * \mathcal{N}_{Tot})$	4.2	6.1	8.4
DP-CD-ECD	$\mathcal{O}(\mathcal{N}^2)$	7.4	7.9	10.2

4.10. Conclusion

In this chapter, we have experimentally investigated the capability of the proposed shape descriptors. After introducing the target datasets along with the utilized evaluation metrics to assess the descriptors, we first set the available parameters for all of the three descriptors. Then, the effectiveness of the proposed descriptors on two standard datasets namely, The McGill Shape Benchmark and SHREC'11 non-rigid watertight datasets has been examined. The results show that the electrical charge-based descriptors offer higher retrieval ability than the SilPH one.

Furthermore, the ability of the proposed approaches in terms of the challenging issue of supporting partial matching has been thoroughly investigated. Although the BOF-CDD descriptor was expected to provide good ability in matching similar sub-parts of the models, the encoded global features by the descriptors did not allow the similar parts to be correctly matched.

Additionally, we examined the robustness of the descriptors to different surface changes such as mesh deformations, tessellations and noise. As expected, the descriptors offer high robustness against these classes of changes.

Finally, the last part of the chapter was dedicated to the examination of time-complexity of the proposed descriptors. The required time for retrieving similar models shows that the view-based descriptor is quite slower than the other two electrical charged-based ones.

5

3D Model Segmentation

5.1. Introduction

As 3D models are becoming larger and more complex, decomposing such models into smaller and simpler meaningful parts is getting necessary in lots of graphic applications viz. compression [99], texture mapping [100], 3D shape retrieval [102], shape simplification [101] etc. Consequently, as also shown in Chapter 2, 3D mesh segmentation or decomposition has gained enormous attention from leading researchers in graphic domains.

The process of segmenting the model surface can be mathematically defined as follows:

Let $\mathcal{M}$ be a 3D mesh model. A set $\mathcal{S} = \{\mathfrak{s}_1, \mathfrak{s}_2, \mathfrak{s}_3, \dots, \mathfrak{s}_{\mathcal{N}} \}$ is considered as a valid segmentation on $\mathcal{M}$ if:

$$\mathfrak{s}_1 \cup \mathfrak{s}_2 \cup \mathfrak{s}_3 \cup \dots \cup \mathfrak{s}_{\mathcal{N}} = \mathcal{M} \text{ and}$$

$$\mathfrak{s}_i \cap \mathfrak{s}_j = \emptyset,\ 1 \le i,j \le \mathcal{N},\ i \ne j\ , \mathcal{N} \text{ is number of segments.}$$

In this chapter we introduce our robust, efficient and fully automatic approach for decomposing 3D models using a well-known fact from electrical physics about the tendency of charge to accumulate at sharp convex areas and to diminish at sharp concavities on the surface of a solid. That is, the 3D models are considered as perfect conductors which are placed in free space (a space with no electric charge). Then, we distribute a predefined electrical charge Q on the surface of the models. Guided by distribution of the charge, our algorithm will be able to specify the segments attached to the main parts of the models as well as locating the boundaries of the segments. Our unsupervised approach needs no prior knowledge about the models and thus, the entire information should be extracted from meshes themselves. As will be presented in Section 5.6, it solely exploits charge density distribution in both protruded part extraction and boundary detection. Figure 5.1 shows the pipeline detail of our proposed segmentation method. It is worthwhile to point out that the charge density distribution takes into account both the local and the global structure of the models, which is quite beneficial in segmentation applications. These valuable features are referred to as the "Quasi-Local" and "Quasi-Global" properties in [171]. The quasi-local property of charge density results in robustness to deformation, which has no effect on the local and fine features of the model surface such as face concavity and boundary detection. The quasi-global property, on the other hand, results in robustness of the proposed approach against noise and other surface perturbations.

The cornerstone of the proposed algorithm is based on two main concepts:

- The premise that a 3D model can be considered as a core body and some protruding parts connected to the core [153].
- A particular regularity in nature known as "transversality[5]" combined with a rule from cognitive science known as the "minima rule", which states that humans perceive boundaries at contours of minimum principal curvature.

These two premises will guide the segmentation process to locate the boundaries along concave regions in the direction of minimum principal curvature.

The organization of the rest of this chapter is as follows: in the next section we first discuss the scientific challenges available in the segmentation area. Next, we briefly explain the applications of 3D model segmentation in other fields of computer vision. Later in Section 5.4, similar related work will be presented. Section 5.5 provides a simple background on the facts, which we utilize in our approach, and finally we will offer a full detailed description of our approach in Section 5.6.

[5] It states that, "when two arbitrarily shaped surfaces are made to interpenetrate; they always meet at a contour of concave discontinuity of their tangent planes" [214].

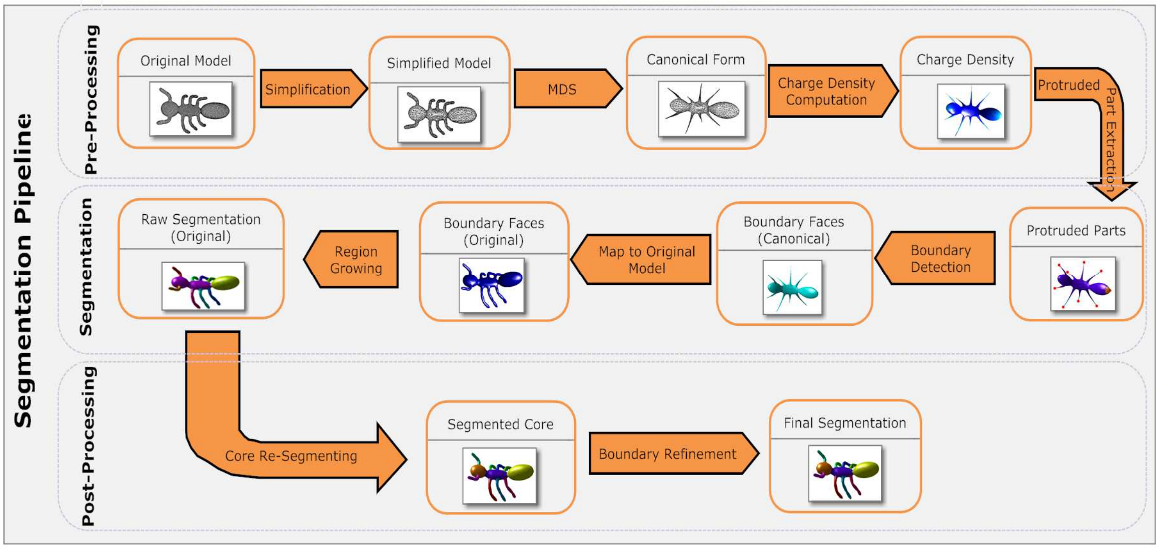


Figure 5.1, Detailed steps of our segmentation framework.

5.2. Scientific Challenges

As discussed in Chapter 2, 3D model segmentation is a relatively mature field so that, during the last decade, a considerable amount of research for decomposing models into meaningful parts has been conducted. Although a large portion of these approaches generate acceptable results, the emergence of new complex models (see Figure 5.2 (a)) due to the daily growth in digital technology and applications, justifies the need for new segmentation approaches. The new approach should be able to answer the challenges of the segmentation domain. These challenges can be listed as follows:

- **Accuracy:** accuracy of the extracted segments, including both boundaries and extracted parts, is the most considerable challenge to be dealt with. The segmentation algorithm should be able to decompose models into meaningful parts similar to the human perceptions as much as possible. Although some of the available approaches can generate acceptable and nice-looking segments (comparing to the manual ones), proposing new accurate techniques still remains a challenging issue.
- **Hierarchical Segmentation:** Hierarchical decomposition is a valuable characteristic which is only supported by a small portion of existing approaches. Based upon the application used, hierarchical segmentation algorithms can proceed until a desired level of detail has been extracted. Considering the fact that the segmentation level is highly application-dependent, the hierarchical segmentation methods are applicable in a wide range of graphic applications. (Figure 5.2 (b))

- **Robustness:** Robustness can be considered as an umbrella term that encompasses a wide range of transformations; the insensitivity to noise and perturbations is a desired feature for the approaches dealing with low resolution models. Also, pose deformation-insensitivity is another constructive aspect of robustness, which is quite beneficial for content-based model retrieval and partial matching. Therefore, designing robust segmentation algorithms will be of enormous value to anyone interested in the computer graphics domain. (Figure 5.2 (c), (d))
- **Speed:** The new emerging complex and high resolution models need time-consuming approaches to be segmented correctly. On the other hand, as with all other computer algorithms: the faster, the better. Despite the emerging fast processors with high speed GPUs, proposing faster algorithms is still of great interest to researchers and graphics companies (Figure 5.2 (a)).

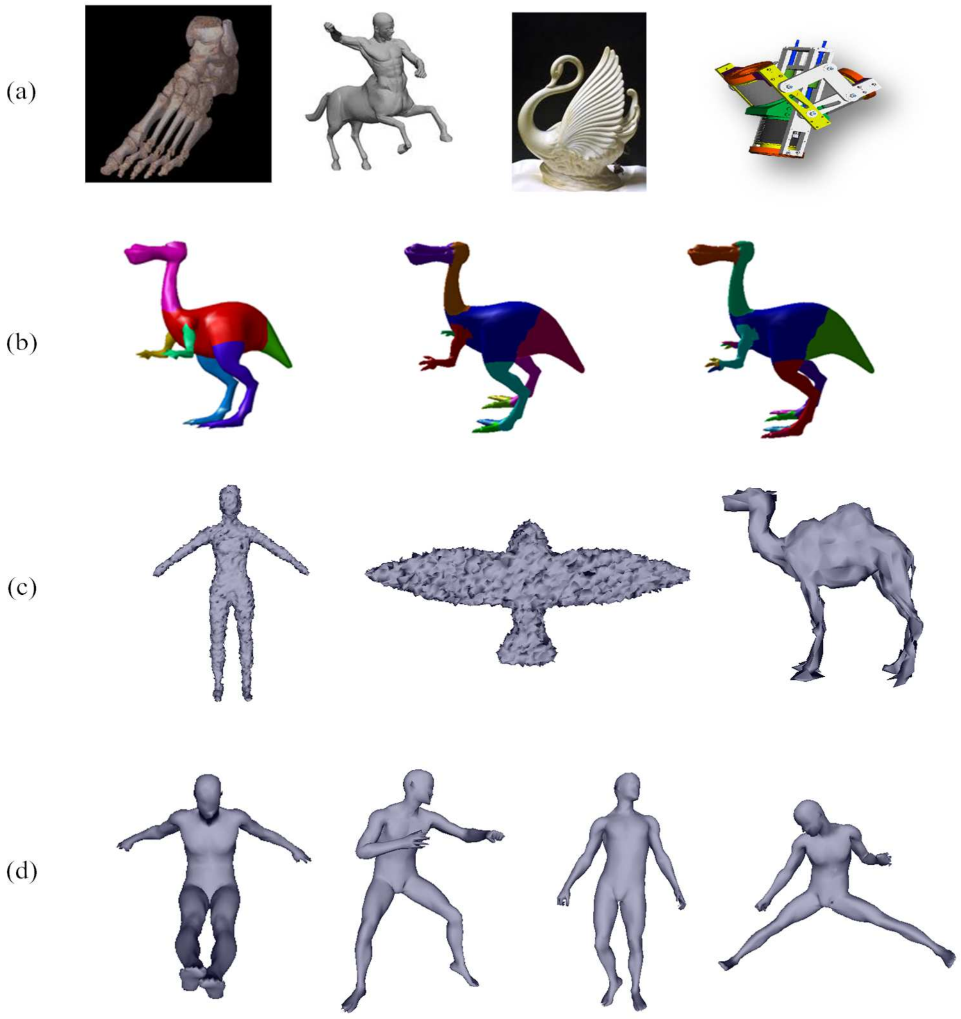


Figure 5.2, sample 3D models (a) models having complex structure, (b) Hierarchically segmented, (c) noisy models and (d) Human model in different poses.

5.3. Applications

Segmentation of a 3D model into semantically meaningful parts has various beneficial applications in the computer graphics domain. In the sequel we briefly list some of the applications.

a) **Reverse Engineering**: Reverse engineering of segmentation is mostly seen in CAD models. By extracting the parts of a mechanical model, one can re-use the components to construct new, different models. As is possible in the real word, pre-constructed components empower users to create free-form models faster and easier. A modelling-by-example framework proposed by Funkhouser *et al.* [130] is a sample of such applications, in which a desired part of the segmented models can be replaced by any of the similar parts in the dataset. Figure 5.3 displays a new chair constructed by assembling different parts of available models.

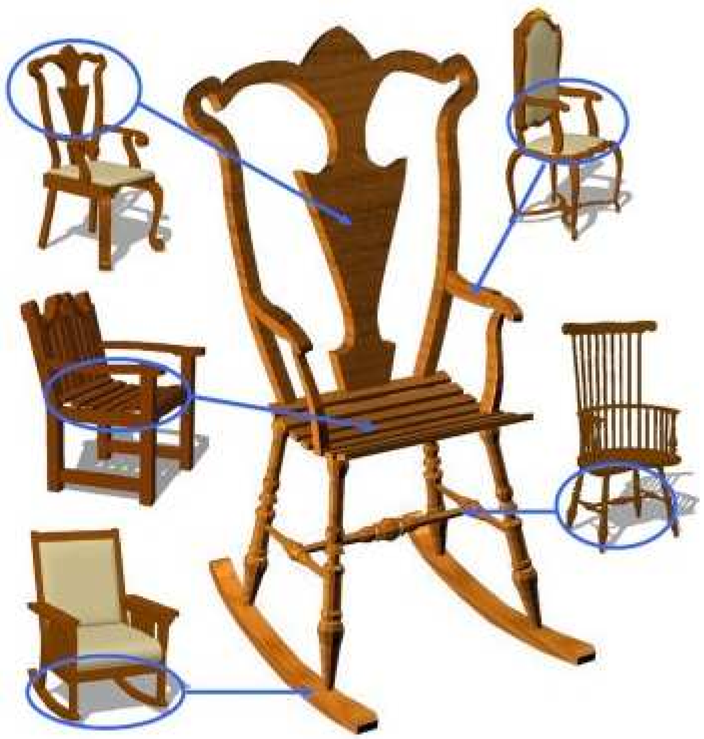


Figure 5.3, construction of a new chair model from other chairs' components. (Figure is taken from [130])

b) **Partial Matching:** Comparing models based on the resemblance of their parts, as well as finding shapes that contain a specific part, can be enhanced by decomposing the models prior to performing a search algorithm. Agathos *et al.* [210] employed a graph-isomorphism methodology to perform partial matching. They extracted the attribute relational graph using extracted segments of the models. Shapira *et al*. [102] on the other hand, use Shape Diameter Function (SDF) as the shape descriptor and allow the user to load a model and select a desired part to retrieve models having similar parts. Figure 5.4 exemplifies some queries and the retrieved models in their work.

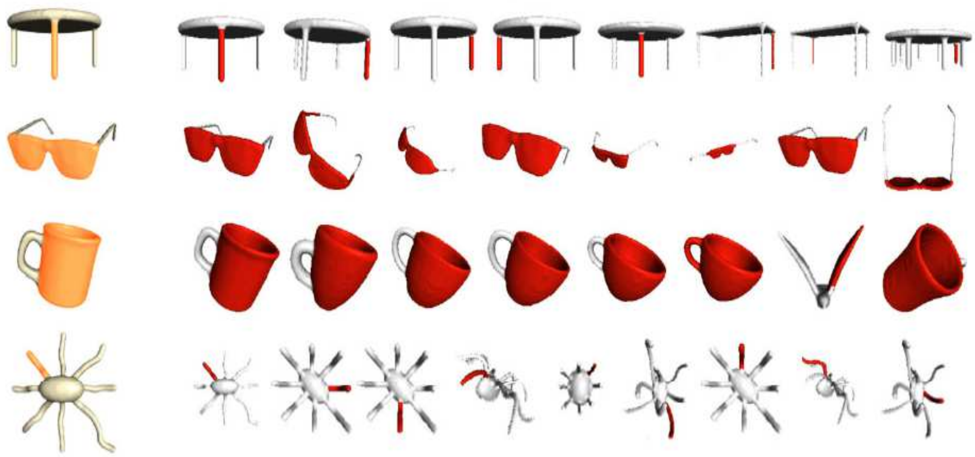


Figure 5.4, Partial matching samples from [102]; the highlighted part of the models in the left-hand column is the query and the models in the other columns are the retrieved results.

c) **Skeleton Extraction:** the skeleton is a one-dimensional graph representing the structure of the model at hand. Due to their simplicity compared to the original models, skeletons can speed up some applications such as deforming the pose of models in producing animated movies. Although the skeletons are usually used to perform segmentation (see Chapter 2), the reverse direction of extracting the skeleton of models from extracted segments has been seen in some works [129][211]. Lien *et al*. [211] follow a hierarchical paradigm to decompose models and extract its skeleton, while Shapira *et al*. [129] extract a consistent skeleton curves, which remain the same for different poses using the SDF descriptor (see Figure 5.5).

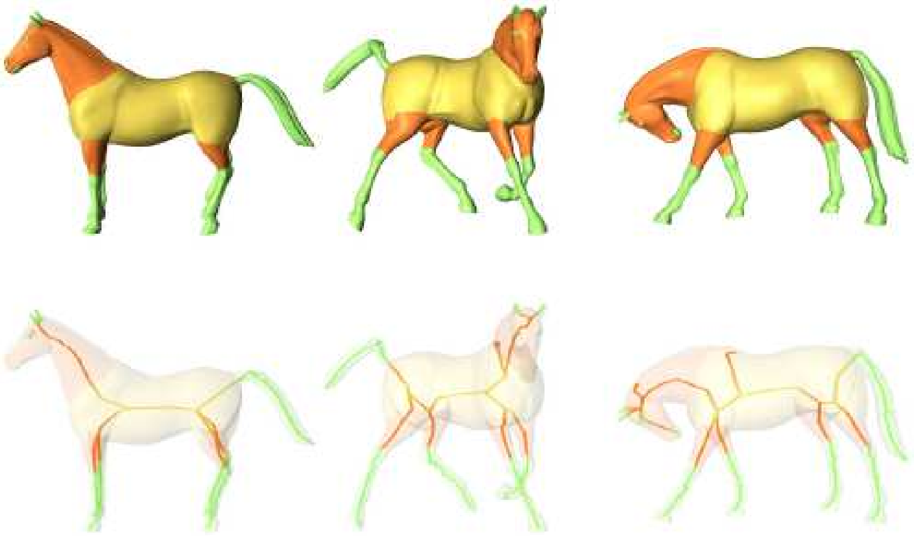


Figure 5.5, Different poses of a horse model and consistent skeleton curves extracted by the approach of Shapira *et al.* [129].

d) **Model Automatic Annotation:** Annotating the model sub-parts can easily convert content-based search to the simpler and possibly faster task of text-based search. In the work of Shapira *et al.* [102], the user is allowed to select a desired part of a segmented model, which is going to be annotated automatically. The part is fed into the retrieval system as a query and a set of results is retrieved. Then, the tags attached to the top 20 retrieved parts are considered as assigning a set of weighted tags to the query part. In a different aspect, Attene *et al.* [193] designed a "shape annotator" system by which the extracted parts of a segmented model are annotated using ontology-based concepts.

5.4. Related Work

According to our knowledge, the only work for 3D model decomposition similar to ours has been proposed in 1997 [171]. Wu and Levine employed a charge density distribution to perform mesh decomposition. In their work, based on the minima rule, the boundary parts of models are located at concave areas having local minimum charge density. They defined a concave face as a face possessing charge density lower than a predefined threshold as well as lower than all of its neighbours (the threshold in their work was: 1.5 * lowest charge density on the model surface). To specify boundaries they tried to locate a ring-like region of faces with the lowest charge density compared to their neighbours. Although their approach is applicable for some parts of specific models, it suffers from two main challenging issues: firstly, the entire boundary must lie in a concavity to be detected accurately, which is not the case for a lot of models. And secondly, for some boundaries, there is no face meeting their requirement to be considered as a concave face. In the next chapter, we will experimentally demonstrate these issues by testing their approach on the available models.

Our method differs from Wu and Levine's work in various aspects. Unlike their algorithm, our approach is able to detect boundary parts not fully located in concave regions. Moreover, our approach can handle mesh models having a more complex arrangement. The pose-insensitive attribute of the proposed work allows the algorithm to generate similar segments for models in different poses. Finally, an extra post-processing step attached to our segmentation pipeline enables us to decompose those models having a mixed (multi-part) core structure.

5.5. Background

Electrical charge density at sharp corners has been studied completely by Jackson [212]. He showed that the charge density at a sample point P on the surface has a direct relation with two factors: i): the distance between point P and the closest sharp edge or corner (ℓ) and ii): the angle of two adjacent planes creating the edge or corner (γ). So that, the larger the angle γ and the smaller the distance ℓ, the greater the charge density at point P. Theoretically, at ℓ =0 the charge density is infinite and zero when the angle defined at the corners is convex and concave respectively. Figure 5.6 illustrates the relationship between charge density ρ, angle γ and distance ℓ. It is worth to note that, for a corner with angle γ, the density ρ declines monotonically, when ℓ increases gradually. It supports a well-known fact in the physics of electricity which says: "the electric charges on the surface of a conductor tend to accumulate at the sharp convex areas and diminish at the sharp concave areas".

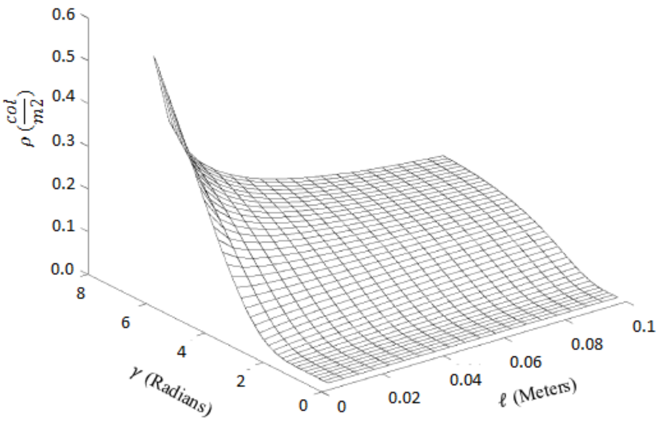


Figure 5.6, The relationship between ρ, $\boldsymbol{\ell}$ and γ. (The image is taken from [212])

Inspired by this fact, each triangular face has its own scalar charge density value which will be used in the segmentation process. As detailed in Chapter 3, the Finite Element Method (FEM) is utilized to calculate the charge density on each triangular face of the models. Figure 5.7 shows some different coloured model samples based on their charge density distribution. As depicted in this figure, convex faces located in the sharp corners possess higher electrical charge than those of placed in the concave corners.

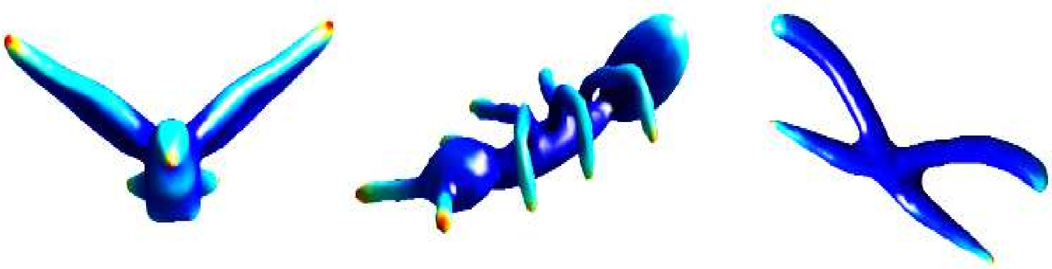


Figure 5.7, Coloured models based on the distribution of charge density on the surface; the redder parts specify the denser faces.

Having computed the simulated charge density for each triangular surface of the models, in the next section we will discuss the details of the proposed segmentation algorithm.

5.6. Proposed Approach

As depicted in Figure 5.1, the entire segmentation process is divided into three main phases; the Pre-processing, Segmentation and Post-processing phases. In the pages that follow, we offer a detailed description of each phase and its position in the proposed framework.

5.6.1. Pre-processing Phase

This section is devoted to the first block of the proposed framework and fulfils the initial functions, which are necessary in the subsequent phases. It comprises the simplification, canonical form presentation and charge density distribution steps.

5.6.1.1. Simplification

In order to speed up the segmentation process and also to remove some of irregularities on the model surfaces, we simplify the models so that they have 5000 faces. Figure 5.8 depicts an Armadillo model at four levels of resolution. As shown in this figure, the model with 5000 faces details the descriptions of the model surface precisely. It is worth mentioning that simplifying to the lower number of faces has also been utilized in this field [15, 151]. Meshlab is an advanced mesh processing software, which is able to simplify each model in a fraction of a second [199].

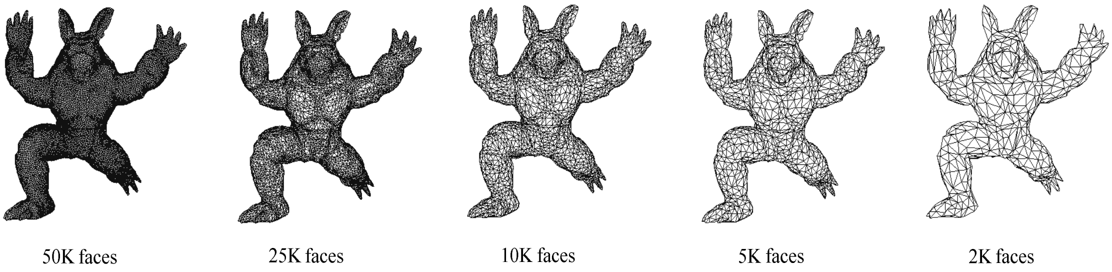


Figure 5.8, An Armadillo model at different levels of resolution based on the number of faces.

5.6.1.2. Canonical Form Representation

In order to achieve a pose-invariant representation, the Multi-Dimensional Scaling (MDS) paradigm is utilized to calculate the canonical form of the models. MDS aims at representing the geodesic distances (i.e. distance along the surface on which the points reside) between points in an m-dimensional Euclidean space.

As will be discussed later in Section 5.6.2.2, straightening bended parts of the models, accomplished by MDS, has a crucial effect on our segmentation results. It helps to locate the representative point of each segment at the very end of segments, at points possessing the most convexity.

To accomplish this goal, we employ the Least-Squares technique and the SAMCOF algorithm to compute the canonical representation of the models [198]. The models shown in the middle row of Figure 5.9 are the canonical form representations of the ones in the first row.

5.6.1.3. Simulation of Charge Density distribution

As stated before, the proposed protrusion-oriented segmentation approach focuses on the nature of charge density distribution. So, after simplification and computation of canonical form representation of the models, the Finite Element Method is used to simulate the distribution of electrical charge on the surface of the models. The last row in Figure 5.9 depicts this distribution on the three different poses of human model.

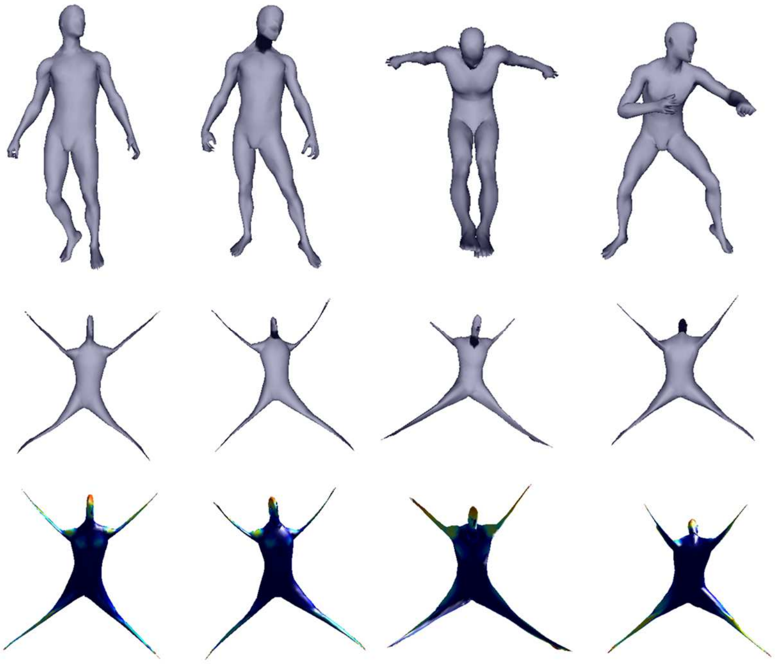


Figure 5.9, top row: Human model in various poses, middle row: the related canonical form representation and Last row: simulated charge density distribution over the models surfaces

5.6.2. Segmentation Phase

This phase is, in fact, the heart of our proposed segmentation framework. This phase tries to locate the boundaries and extract the protruded parts connected to the core part of model. The protruded parts are treated as initial segments. This section discusses the second block of the framework shown in Figure 5.1.

Before proceeding to the detail of the proposed boundary detection technique, first we describe the terms used in the following.

5.6.2.1. Terminology

a) **Region:** a set of connected faces of $\mathcal{M}$ which share similar properties**.**

b) **Ring_Neighbourhood of face $\aleph$, RN($\aleph$):** is a subset $\Finv$ of faces on $\mathcal{M}$ such that:

$$\Finv = \{\mathfrak{H}_1, \mathfrak{H}_2, \dots\} \mid \text{GeoDist}(\aleph, \mathfrak{H}_i) \leq \text{AvgEdgLen}, \ \forall \text{ faces } \ \mathfrak{H}_i \in \Finv.$$

Where, GeoDist(.,.) is the Geodesic distance between two faces and AvgEdgLen is the average length of edges on the surface $\mathcal{M}$. (See Figure 5.10 (a)). It can be simply considered as a set of connected faces around face $\aleph$ which have common vertices with the face $\aleph$.

c) **Ring_Neighbourhood of region $\mho$, RN($\mho$)**: is region Ω if:

i. $\mho \cap \Omega = \emptyset$

ii. $\forall$ faces $\aleph \in \mho, \exists$ face $\mathfrak{H} \in \Omega$ so that $\text{GeoDist}(\aleph_{Cntr}, \mathfrak{H}_{Cntr}) \leq \text{AvgEdgLen}$

Here, $\aleph_{Cntr}, \mathfrak{H}_{Cntr}$ are centres of faces $\aleph$ *and* $\mathfrak{H}$. (See Figure 5.10 (b)). It simply means a set of faces around region $\mho$ that have common vertices with the faces in the region $\mho$.

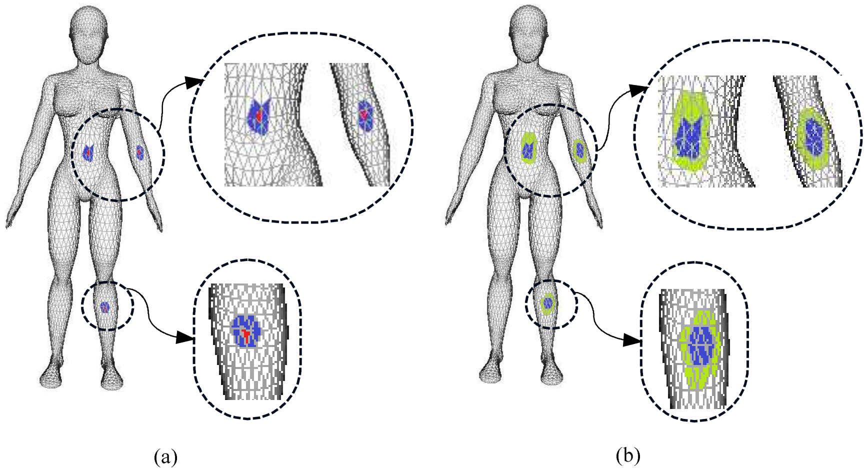

(a) (b)

Figure 5.10, (a) three sample faces (red colour) and the associated ring neighbourhood (blue faces), (b) three sample regions (blue region) and their associated ring neighbourhood (green region).

d) **Concave Face:** face $\aleph$ is concave if $ChrgDns(\aleph) \leq ChrgDns(\mathfrak{H}), \forall\ \mathfrak{H} \in RN(\aleph)$. In other words, face $\aleph$ is concave if its charge density is lower than the density of all faces in its Ring_Neighbourhood.

e) **Semi-concave Face:** face $\aleph$ is semi-concave if there is a region Ω so that:

i. $\Omega \subset \text{RN}(\aleph)$

ii. $\text{ChrgDns}(\aleph) \leq \text{ChrgDns}(\mathfrak{H}), \forall\ \mathfrak{H} \in \Omega$

iii. $|\Omega| \geq N * |\text{RN}(\aleph)|$, $|.|$ denotes the number of faces in the region.

It means: face $\aleph$ is semi-concave if its charge density is lower than density of N percent of faces in its Ring_Neighbourhood. (N will be set to a suitable value experimentally.)

f) **Concave Region:** region $\mho$ is concave if $AvgChrgDns(\mho) \leq ChrgDns(\aleph), \forall \aleph \in RN(\mho)$ where $AvgChrgDns(.)$ is the average of charge densities of faces in the region. In other words, a region is concave, if the average of the charge densities of its faces is lower than density of all of faces placed in the Ring_neighbourhood of the region.

g) **Semi-concave Region:** region $\mho$ is semi-concave if there is a region Ω such that:
   i. $\Omega \subset \mathrm{RN}(\mho)$
   ii. $\mathrm{AvgChrgDns}(\mho) \leq \mathrm{ChrgDns}(\mathfrak{H}), \forall \mathfrak{H} \in \Omega$
   iii. $|\Omega| \geq \mathrm{N} * |\mathrm{RN}(\mho)|$, $|.|$ denotes the number of faces in the region

   That means: a region is semi-concave if the average of the charge densities of its faces is lower than density of N percent of faces in the Ring_neighbourhood of the region (N will be set to a suitable value experimentally).

5.6.2.2. Protrusion Parts (Initial Segments) Extraction

Based on a premise in [153], segments can be considered as protruding areas connected to the main part of the model. So, we need to extract the protruding parts as the first step of the segmentation phase. We identify Ψ connected faces having charge density higher than the other faces on the model surface so as to extract these parts (Parameter Ψ will be set to an appropriate value in the next Chapter). Each disjoint set of identified faces creates a protruding part, which is an initial extracted segment. Then, among the faces of each protruding part χ, a face possessing the highest charge density is identified as the representative face of the segment. Figure 5.11 shows some examples of protruding parts connected to the main body in different colours along with their representative faces. Note that, we will ignore the extracted parts having a low number of faces. These parts appear due to noise or the presence of some bumps on the surface of the models.

We expect to see the representative faces at the very end of segments, at the points which have the most convexity. But for some of the segments having bending regions, it is possible to locate some faces, which have the most charge density at the bending region. Two representative faces of the ant model identified by rectangles in Figure 5.11 (left), are examples of representative faces which are not located on the very end of the protruding parts. To avoid this discrepancy we utilize the canonical form representation of models in which the geodesic distances are mapped into Euclidean ones. As presented in Figure 5.11 (right), in the canonical form presentation all of the representative faces are located at the very ends of each segment.

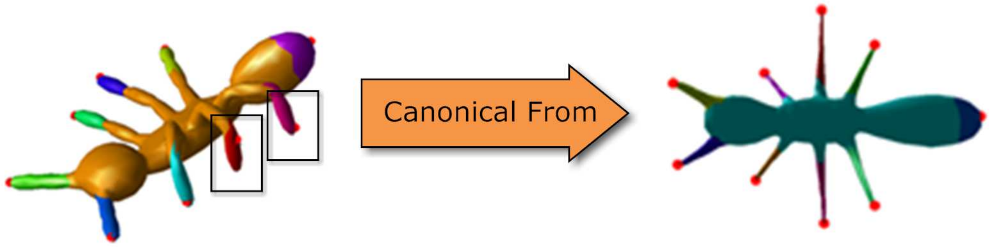


Figure 5.11. Examples of protruding parts along with the corresponding representative faces indicated by red points.

5.6.2.3. Boundary Faces Detection

Based on the minima-rule, concave faces are good candidates to locate the boundaries. On the other hand, as stated above in Section 5.5 (Background), concave faces have minimum charge density on the surface of a solid. Thus, the local minima of charge density on the surface of the models can be a helpful tool to locate the boundary faces.

With the aim of detecting boundaries, based on the distribution of electrical charge, we define two types of boundary faces.

- **Strong Boundary face:** face $\mathfrak{H}$ is a strong boundary face, if it is a concave face and its Ring_neighbourhood creates a concave region. Being located in a concave region for a detected boundary guarantees that the boundary face is a real one and is not due to noise or presence of bumps.
- **Weak Boundary face:** face $\mathfrak{H}$ is weak boundary face, if it is a semi-concave face and its Ring_neighbourhood creates a semi-concave region. The reason for using weak boundary faces is to find candidate boundary faces of the segments, for which no strong boundary has been detected.

Our examinations on dozens of models about distribution of electrical charge on the surface of models revealed the fact that Strong Boundary Faces only emerge on the extreme concave corners of the models located in the real boundaries (see Figure 5.13). Weak Boundary Faces on the other hand, may be detected on places other than the real surface boundaries. Therefore, the Weak ones are only boundary candidates and so an examination process should be applied to verify them. Figure 5.12 shows detected strong and weak boundary faces on an Ant model.

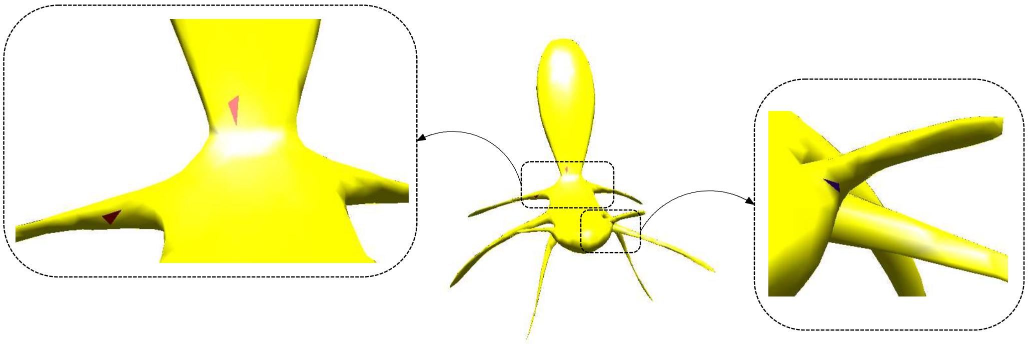


Figure 5.12, Detected Strong Boundary Face (Blue colour) and Weak Boundary Faces (Red) colour on ANT model.

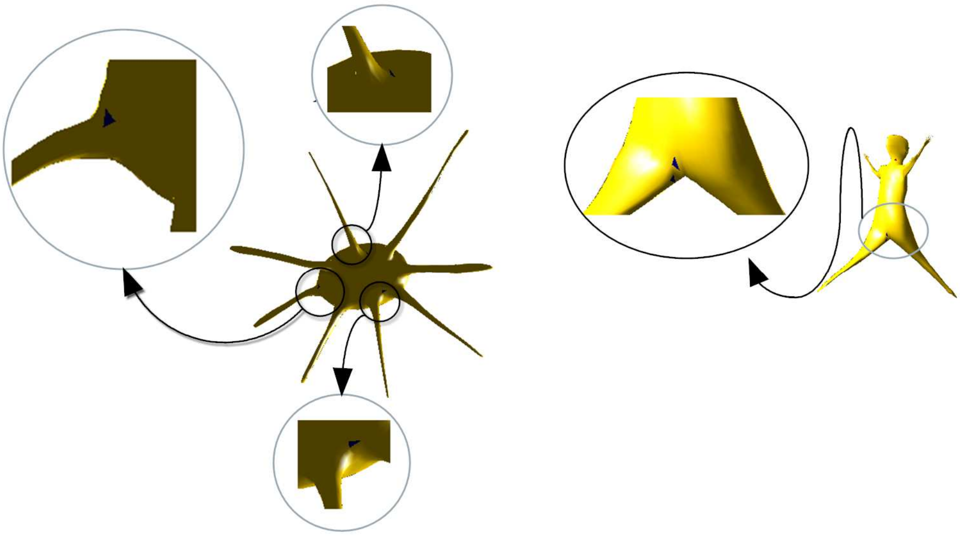


Figure 5.13, The strong boundary faces located in boundaries of protruded parts of Octopus and Human models.

5.6.2.3.1. *Merging Strong Boundary Faces*

It is worth mentioning that, depending on model shape and geometry, the proposed algorithm may detect more than one boundary face for a protruded part. To overcome this problem, after finding the strong boundary faces, an enhancement process is required to merge related strong boundary faces. This leads to a speed-up of the remaining process of segmentation. For this reason, if there are several strong boundary faces having almost the same distance from the representative face of a segment, we filter all of them except the one that possesses the lowest charge density (See Figure 5.14).

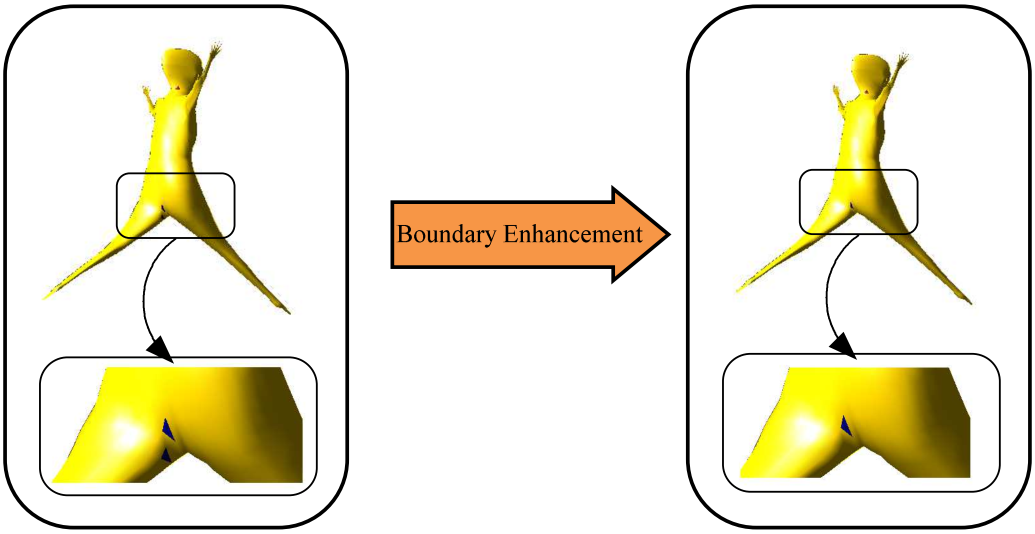


Figure 5.14, Strong boundary faces enhancement.

5.6.2.3.2. Weak boundary enhancement

After merging the strong boundary faces, we try to detect boundary faces for the segments whose strong boundaries are not determined yet. Therefore, the weak boundary faces are detected with the intention of locating the candidate boundary faces. Depending on the surface geometry and also noise and un-smoothness of the model surface, some spurious weak boundary faces may be found located in different places other than boundary parts (see Figure 5.15). In order to remove these outliers, we employ some simple and efficient heuristics as follows:

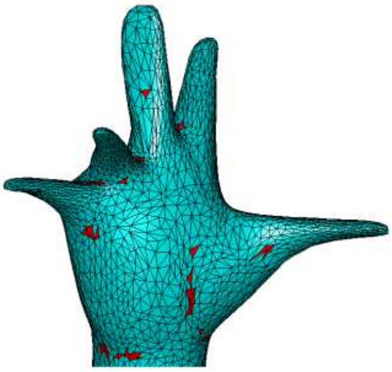


Figure 5.15, Weak boundary faces detected on a hand model with spurious boundaries.

- All the weak boundary faces, whose geodesic distances from each other are less than a threshold, will be ignored except the one having the lowest charge density. We experimentally set the threshold as 3*(AvgEdgeLen).

- If a weak boundary face is placed in a protruded part (segment) for which the strong boundary face has been already defined, the weak boundary face will be removed.
- For the remaining weak boundary faces, we utilize the relationship available between distance ℓ from the sharp corner, the angle γ defined at the corner and the rate of change of charge density ρ (see Section 5.5 and Figure 5.6). As stated before, for target points around a sample corner, the charge density has marginal and monotonic change, so that the density gradually decreases as the distance of the target point from the corner grows. Figure 5.16 shows changes in the amount of charge in successive annular regions in a sample segment of the middle finger of a Hand model. As shown in this figure, the amount of electrical charge in successive annular regions of a desired protruded part is only slightly different. On the other hand, for two successive annular regions (regions 19 and 20) having constant width on both sides of the boundary, the change in electrical charge has a significantly larger difference, compared to changes in the protruding area. Accordingly, in order to remove all the remaining spurious weak boundary faces, we define and assign the parameter *StepDiff(k)* for any protruded part χ_k using the following terms. This parameter, as will be detailed in the sequel, is utilized to assess the weak boundaries and remove the outliers.

a) **AR(i , k):** The i^{th} annular region in protruded part χ_k having representative face rp_k:

$$AR(i,k) = \{ faces\ \aleph \in model\ \mathcal{M}\ |\ GeoDist\ (\aleph, rp_k)\ is\ in\ range\ [\ i,i+1\ )\ *\ AvgEdgeLen\ \}$$

b) **ChDiff (i , k):** Difference in the amount of charge in i^{th} and $(i+1)^{th}$ successive annular regions of part χ_k:

$$ChDiff\ (i,k)=Abs(\ Ch(\ AR(i,k)\ )-Ch(\ AR(i+1,k)\ )\ )$$

Here Ch (.) denotes the amount of electrical charge in a desired annular region.

c) **DiffSet (k):** set of charge differences in successive annular regions in the protruded part χ_k :

$$DiffSet(\ k\ ) = \{\ ChDiff\ (\ i,k\ )\ , \qquad i=1,2,3,...,5\}$$

It is worthwhile to note that 5 successive annular regions are examined to calculate the DiffSet parameter.

d) **StepDiff (k):** The distinctive factor of charge difference per step for protruded part χ_k is:

$$StepDiff(\,k\,)=StdDiv(\,DiffSet\,(\,k\,)\,)$$

Where, StdDiv(.) is the standard deviation of the desired set.

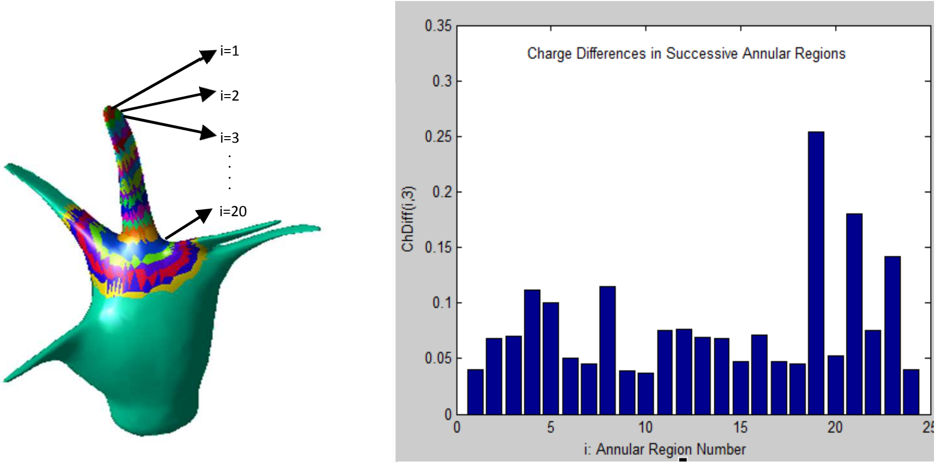


Figure 5.16, Different annular regions in the middle finger of a Hand model (segment number=3) and the charge difference in successive regions.

Now, on either side of any detected weak boundary face in protruding part χ_k, two annular equal-width regions are extracted and the charge amount at each region is observed (the width of the annular regions is set equal to the average of edge lengths on the current model). If the change in electrical charge of these two regions is not significantly different from the parameter *StepDiff* of the desired protruding part (less than T * *StepDiff*), the current boundary face is ignored and the process is repeated with the next weak boundary face. Otherwise, the current weak boundary face is considered as a real boundary for the current segment and the process is restarted with the representative face of another initial segment. Parameter T will be set to an appropriate value in the next Chapter. This process is repeated until the boundary faces for all initial segments have been found.

To speed up the aforementioned process and to avoid checking detected spurious faces located in the core part of model, before starting the enhancement process, the detected weak boundary faces are sorted in ascending order of their geodesic distance to the representative face of the protruded segment. And the process of checking the boundary faces is started from the top of the list.

5.6.2.4. Convert to the Original models

Since all of the aforesaid procedures have been applied on the simplified canonical form of the models, a reverse mapping is required to convert the model to its original format. To do so, all of the extracted critical faces (boundary and representative faces) are mapped to the original representation of the models. Mapping from the canonical form into the simplified original representation can be easily accomplished by finding the faces of the original model associated to the critical ones on the canonical form model. On the other hand, in order to perform a mapping from the simplified into the high-resolution original model, the faces on the high-resolution original model having identical coordinates to those of the critical faces on the simplified one, are simply considered as the final boundary and representative faces of the segments, which are employed in a region growing scheme to finalize the process.

5.6.2.5. Region Growing

Having the start points (representative faces) and the boundary faces for all segments, the region growing scheme can be easily applied on the faces to construct the complete segments. To do so, for each extracted segment, the geodesic distance g between its representative and boundary faces is computed. Then, all of the faces on the model, whose geodesic distances from the representative face are less than or equal to g, will be assigned to that segment.

5.6.3. Post-Processing Phase

The outcomes of the segmentation process so far, can be considered as raw results, which need to be boosted by applying some extensions to the segmentation pipeline such as re-segmenting the extracted mixed cores as well as locating more accurate boundary faces. These extensions are discussed in the following sections.

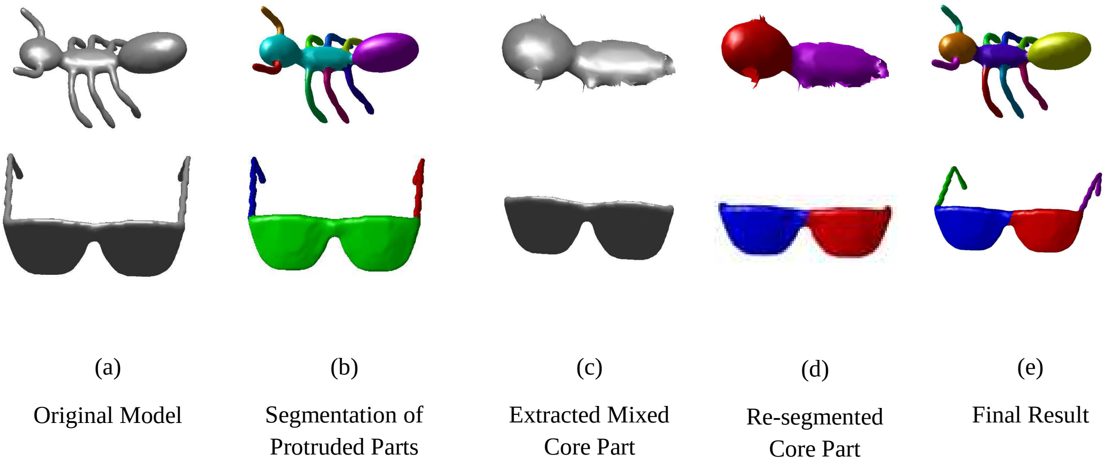
5.6.3.1. Re-segmenting the mixed cores

Similar to other protrusion-based approaches, the proposed algorithm suffers from a lack of ability in detaching mixed segments connected to the main part. We refer to this kind of segment as a ‘mixed core’ and it is defined as follows:

"***A mixed core (multipart core) is a detected core part of the model (by segmentation algorithm) which can be decomposed into simpler parts by human perception system.***"

Consider the head and body parts of the Ant model in the first row of Figure 5.17 (b), which have been detected as a mixed core of the model by our proposed approach. But, the human perception system identifies the head as a separate segment attached to the antenna and main body segments. The same issue occurs in spectacle models, where both eye frames are considered as only one segment.

To solve this problem, the extracted mixed core of such models can be dealt with as a full model and the same segmentation algorithm is recursively applied on it. In order to accomplish this goal, electrical charge Q is distributed over the surface of the extracted core parts and using the proposed algorithm, the representative and boundary (Strong or Weak) faces are detected to decompose the model into the core and attached segment. Repeating this process leads to detaching the mixed core parts into a main core part and some segments attached to it (See Figure 5.17 (d and e)).



(a) Original Model

(b) Segmentation of Protruded Parts

(c) Extracted Mixed Core Part

(d) Re-segmented Core Part

(e) Final Result

Figure 5.17, Steps for segmenting models having mixed core: First and second rows show the process for an Ant and Spectacle models, respectively. (The colours are arbitrary)

5.6.3.2. Boundary Cut Refinement

The last step of the segmentation phase in our approach (region growing in Section 5.6.2.5) uses geodesic distance to find the boundary parts. Therefore, it only extracts a soft and raw segmentation, which is a rough approximation of the real boundaries. Thus, the extracted boundaries are not fully located in the real boundary areas for some models. Figure 5.18 depicts the inaccurate detected boundaries for Bird, Vase and Bust models. This problem may happen because not all of the boundary faces necessarily have the same distance from the representative face of the segment.

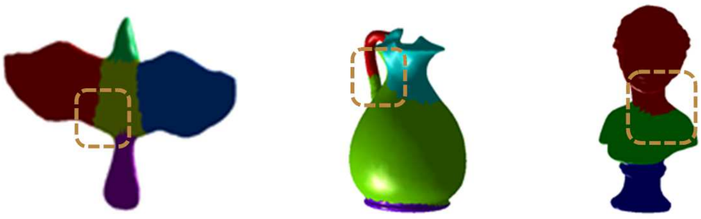


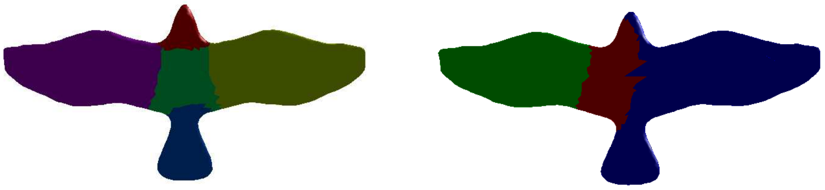
Figure 5.18, inaccurate detected boundaries for three sample models identified by a rectangle.

In some segmentation approaches, a post processing cut-refinement step is performed to overcome this issue. The snake and min-cut algorithms are two of most common approaches in the literature [154, 172, 213]. Similar to [172], due to its simplicity in implementation and its guarantee in finding the solution, we employ the min-cut methodology on the dual graph of the models. To do so, the search region and capacity function should be defined on the extracted dual graph as follows:

For creating the search region, firstly three disjoint regions are extracted on the mesh namely:

- **Region A**: Includes the faces of the protruded area. (Green part of Figure 5.19(b))
- **Region B**: is going to contain the boundary faces. This region can be defined by selecting all of the faces having a geodesic distance in the interval $[0.9 * AvgGeoDist, 1.1 * AvgGeoDist]$ where, $AvgGeoDist$ is the average geodesic distance of a segment boundary face to the related representative face. (Red part of Figure 5.19(b))
- **Region C**: contains all of the faces on the rest of the model. (Blue part of Figure 5.19(b))

Next, the dual graph of region B is extracted to define the search region. The search region is constructed via adding two extra nodes, start S and target T, to either side of the dual graph. This graph is a flow graph for which the least cost path should be computed. The capacity function that we use is similar to that of Katz and Tal [172]:



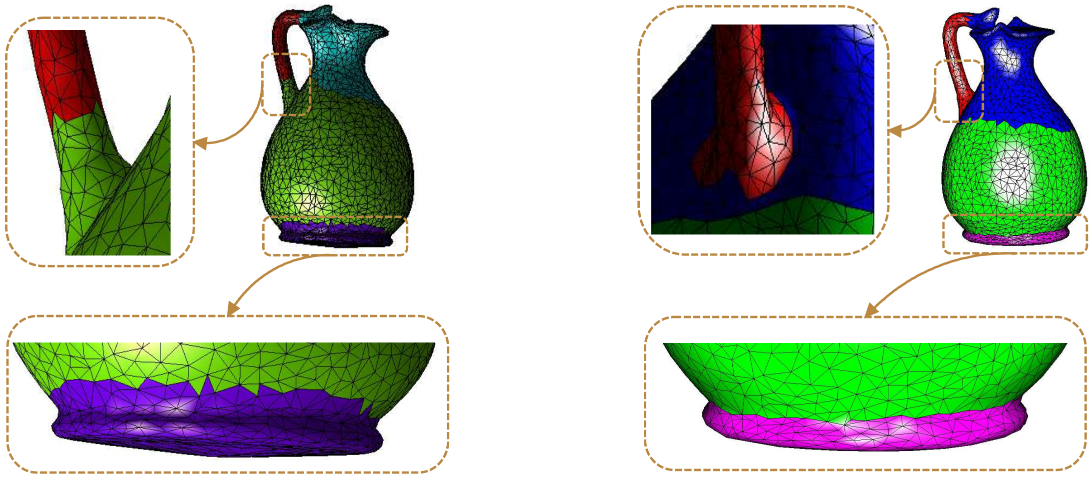
(a) (b)

Figure 5.19, (a) raw segmentation of Bird model and (b) three disjoint regions extracted for refining boundary of the left wing.

$$Cap(i,j) = \begin{cases} \dfrac{1}{1 + \dfrac{Ang\_Dist(\alpha_{i,j})}{Avg(Ang\_Dist)}} & if\ i,j\ \neq s,t \\ \infty & otherwise \end{cases}$$

Here, $\alpha_{i,j}$ is the dihedral angle of two faces which share the edge (i, j), $Ang\_Dist(\alpha_{i,j})$ is angular distance and defined as: $Ang_{Dist(\alpha_{i,j})} = \mathcal{B}(i - \cos(\alpha_{i,j}))$ and $Avg(Ang_{Dist})$ is the average angular distance. Note that for $\mathcal{B} = 1$, both concave and convex angles are treated equally. But, since we are looking for concave angles, a small positive number is assigned to $\mathcal{B}$ for convex angles and $\mathcal{B} = 1$ is for the concave ones. Figure 5.21 illustrates the boundary refinement process.

Applying the min-cut_max-flow algorithm on the extracted flow graph leads to extracting better locations for the boundary parts (see Figure 5.20).



(a) (b)

Figure 5.20, Detected boundaries (a) before and (b) after applying Cut Refinement.

5.7. Chapter Summary

This chapter was devoted to a detailed discussion of the proposed algorithm for segmenting 3D models into meaningful parts. A well-known fact from electrical physics about the distribution of electrical charge on the surface of a solid, which was also used as a shape descriptor in Chapter 3, guided our search to locate boundaries of segments of 3D models. The result of the proposed approach has been boosted via a boundary cut refinement algorithm as well as re-segmenting the extracted core parts having complex structure. Finally, a similar approach of Wu and Levine for 3D model segmentation using distribution of electrical charge was debated and its drawbacks in comparison to our approach were considered.

In the next chapter we will perform a thorough analysis of the ability of the proposed approach along with its limitations on the models available in different standard datasets.

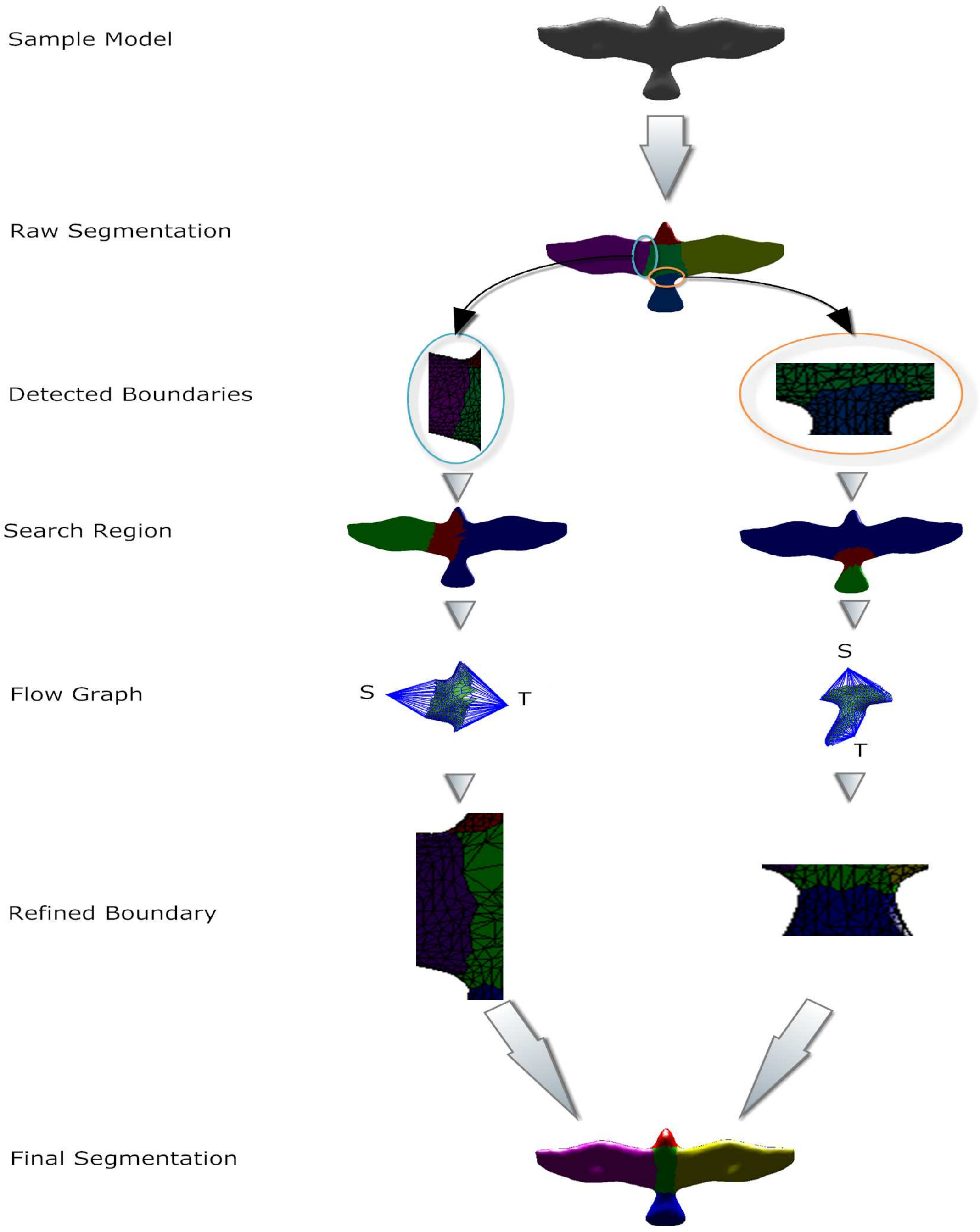


Figure 5.21, Cut-refinement process, S and T denote Source and Target nodes of the extracted flow graph.

6

Experimental Results of the Proposed Segmentation Approach

6.1. Introduction

The previous chapter discussed the proposed segmentation framework using electrical charge distribution. As mentioned there, the proposed system consists of three main phases viz. the Pre-Processing phase (for simplification and canonical form representation of models), the Segmentation phase (for extracting the protruded parts and boundary faces) and the Post-Processing phase (for refining the extracted boundaries and re-segmenting the extracted multi-part cores).

In this chapter, we concentrate on our charge density-based segmentation framework in experimental terms. We apply our algorithm to the various models available in the watertight track of the 2007 Shape-based Retrieval Contest (SHREC'07) [162] datasets and 28 models, which have been exploited in the 3D mesh segmentation track of the SHREC'12 contest [115].

After discussing the parameters, which were used, and introducing some standard 3D model segmentation evaluation factors, we will give a comprehensive quantitative comparison with the state-of-the-art approaches. We also compare our approach to the similar approach introduced by Wu and Levine [171]. Furthermore, the robustness of the proposed framework against different transformation will be discussed. The timing and complexity of the entire segmentation process is another crucial factor that will be discussed in this chapter. Finally, we will consider the limitations of the proposed approach in decomposing models with certain special structures.

6.2. Description of Datasets

In order to evaluate the proposed approach, we utilized two widely used standard benchmarks from the SHREC'07 and SHREC'12 contests. The datasets contain a large variety of models from rigid objects (Chair and Table classes) to deformable models (Ant and Human categories). Besides the different number of models in each benchmark, the SHREC'07 set contains facet-based segmentations, whereas the SHREC'12 set contains vertex-based segmentations:

a) **SHREC'07 Benchmark:** The watertight track of SHREC'07 contest comprises 400 high resolution models evenly categorized into 20 classes including Ant, Chair, Glasses, Hand, Human, Octopus *etc*. We have excluded the Spring class from the corpus as it is almost impossible to be segmented meaningfully. So, the experiments have been done on the remaining 380 models. The SHREC'07 dataset comes with 4300 manually generated segmentations (an average of eleven segmentations for each model) to provide a rich distribution over "How humans decompose each model into functional parts" [136]. Figure 6.1 depicts sample models from each class of the SHREC'07 dataset.

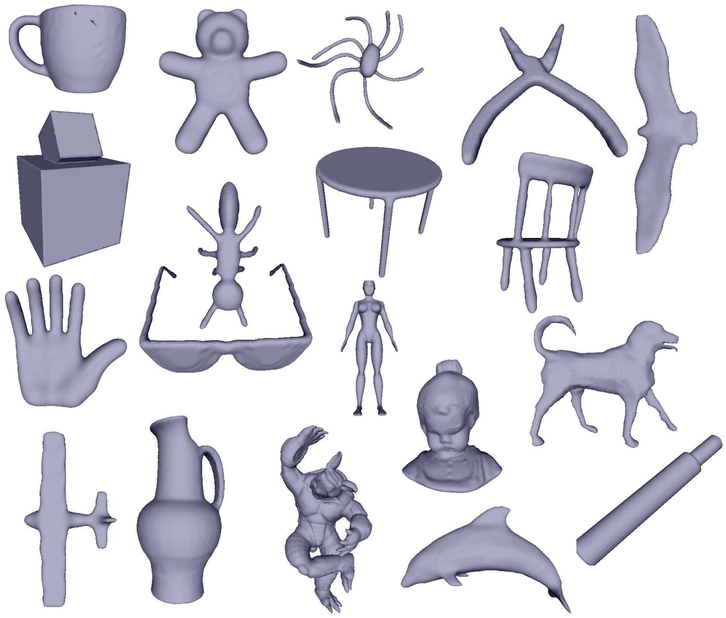


Figure 6.1, Sample models of watertight track of SHREC'07 dataset

b) **SHREC’12 Benchmark:** This dataset comprises 28 models (as triangular meshes) categorized in five groups namely Hand, Furniture, Animal, Human and Bust. The dataset is accompanied by 112 ground-truth segmentations done by 36 human volunteers (four manual segmentations for each model). Figure 6.2 shows the entire 28 models of this dataset. Among the models of the dataset, three classes contain objects which are fairly easy to segment (Hand, Human and Animal), while the models of the other two groups (Bust and Furniture) are very challenging. As the authors claim, the relatively small size of this dataset allows us to rapidly evaluate segmentation algorithms without running on 380 objects, as is done in SHREC’07 benchmark.

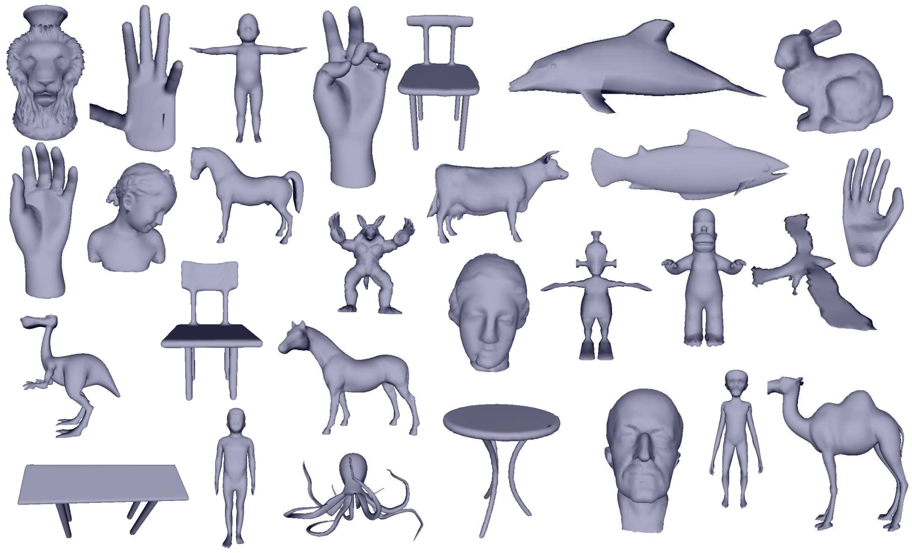


Figure 6.2, 28 models of the segmentation track of SHREC’12.

6.3. Evaluation Metrics

Chen *et al.* [136] have proposed four different metrics to evaluate segmentation algorithms quantitatively. Actually, these metrics had been used previously for evaluating 2D image segmentation approaches and then adapted for 3D purposes. These widely accepted factors are Cut Discrepancy (CD), Hamming Distance (HD), Rand Index (RI), and Consistency Error (CE) which are defined as follows:

a) **Cut Discrepancy (CD):** Among all of the four metrics the CD is the only boundary-based one that provides a simple and intuitive measure of how well boundaries align. As Chen *et al.* [136] mentioned: "It sums the distances from points along the cuts in the computed segmentation to the closest cuts in the ground truth segmentation, and vice-versa". It actually measures the distances between cuts [216] which can be formulated as follows:

Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be two segmentations of a given 3D model and two sets, $\mathcal{C}_1$ and $\mathcal{C}_2$, are sets of all points located on the segments boundaries. The geodesic distance from a point $\mathcal{p}_1 \in \mathcal{C}_1$to a set of boundary faces $\mathcal{C}_2$ is defined as:

$$d_{g}\ (\mathcal{p}_1\ ,\mathcal{C}_2) = \min\{\ d_{g}\ (\mathcal{p}_1\ ,\mathcal{p}_2)\ ,\forall\mathcal{p}_2 \in\ \mathcal{C}_2\} \tag{6.1}$$

Now the Cut Discrepancy between $\mathcal{S}_1$ and $\mathcal{S}_2$ is defined as Equation 6.2:

$$CD\ \ (\mathcal{S}_1\ ,\mathcal{S}_2) = \frac{DCD(\mathcal{S}_1 \Rightarrow\ \mathcal{S}_2) + DCD(\mathcal{S}_2 \Rightarrow\ \mathcal{S}_1)}{AvgDis} \tag{6.2}$$

Where $AvgDis$ is the average Euclidean distance from a surface point to the centroid of the mesh and DCD is a directional function defined as $DCD(\mathcal{S}_1 \Rightarrow \mathcal{S}_2) = Mean\,\{\ d_{g}\ (\mathcal{p}_1\ ,\mathcal{C}_2)\ \ ,\forall\mathcal{p}_1 \in\ \mathcal{C}_1\}$. A value of 0 for the CD metric indicates a complete matching between the two segmentations and greater values indicate imperfect ones. It is worthwhile to mention that despite its simplicity, the CD factor suffers from sensitivity to granularity. The CD decreases to zero when more cuts are added to the ground-truth segmentation and also it is undefined when the model has zero cuts.

b) **Hamming Distance (HD):** this region-based factor tries to measure the overall difference between two segmentation results by comparing the average of two parameters, missing rate ($\mathcal{R}_{f}$) and false alarm rate ($\mathcal{R}_m$), of the segmentation to the ground truth. These two rates are defined using a parameter, Directional Hamming distance ($\mathcal{D}_{\mathcal{H}}$), on the given two segmentations $\mathcal{S}_1 = \{\ \mathcal{S}_1^1, \mathcal{S}_1^2, \dots, \mathcal{S}_1^m\}\ and\ \mathcal{S}_2 = \{\ \mathcal{S}_2^1, \mathcal{S}_2^2, \dots, \mathcal{S}_2^n\}$. Directional Hamming distance ($\mathcal{D}_{\mathcal{H}}$) is defined as:

$$\mathcal{D}_{\mathcal{H}}\ \ (\mathcal{S}_1 \Rightarrow\ \mathcal{S}_2) = \sum_i ||\mathcal{S}_2^i \backslash \mathcal{S}_1^{i_t}|| \tag{6.3}$$

Where operator \ is the set difference operator and ||.|| is the cardinality of a set and $i_t = argmax\ \left|\left|\mathcal{S}_2^i \cap \mathcal{S}_1^k\right|\right|$ is the closest segment in $\mathcal{S}_1$ to the segment $\mathcal{S}_2^i$

in $\mathcal{S}_2$. Suppose $\mathcal{S}_2$ is the ground-truth segmentation. Then, the parameters, missing rate ($\mathcal{R}_m$) and false alarm rate ($\mathcal{R}_f$), are defined as follows:

$$\mathcal{R}_m\ (\mathcal{S}_1, \mathcal{S}_2) = \frac{\mathcal{D}_{\mathcal{H}}\ (\mathcal{S}_1 \Rightarrow\ \mathcal{S}_2)}{||\mathcal{S}||} \tag{6.4}$$

$$\mathcal{R}_f\ (\mathcal{S}_1, \mathcal{S}_2) = \frac{\mathcal{D}_{\mathcal{H}}\ (\mathcal{S}_2 \Rightarrow\ \mathcal{S}_1)}{||\mathcal{S}||} \tag{6.5}$$

Here $\left||\mathcal{S}|\right|$ is the total surface area of the model. Finally, the Hamming Distance (HD) is simply defined as the average of $\mathcal{R}_m$ and $\mathcal{R}_f$ as Equation 6.6:

$$\text{HD}\ (\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{2}\ (\ \mathcal{R}_m\ (\mathcal{S}_1, \mathcal{S}_2) +\ \mathcal{R}_f\ (\mathcal{S}_1, \mathcal{S}_2)) \tag{6.6}$$

As with Cut Discrepancy, the value of 0 for the HD metric indicates perfect matching between the two segmentations.

c) **Rand Index (RI):** this factor is the most important evaluation factor and attempts to convert the problem of comparing two segmentations into the problem of pair-wise likelihood checking of face labels in two segmentations [217]. Suppose we denote the number of faces on the model by $\mathcal{N}$ and the two segmentations by $\mathcal{S}_1$ and $\mathcal{S}_2$. If $\mathcal{S}_i^1$ and $\mathcal{S}_i^2$ are the segment IDs of face i in segmentations $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively, then the two parameters $\mathcal{C}_{ij}$ and $\mathcal{P}_{ij}$ are defined as: $\mathcal{C}_{ij} = 1 \text{ iff } \mathcal{S}_i^1 = \mathcal{S}_j^1$, and $\mathcal{P}_{ij} = 1 \text{ iff } \mathcal{S}_i^2 = \mathcal{S}_j^2$. The region-based measure based on the Rand Index (RI) is defined as the following equation:

$$\text{RI}\ (\mathcal{S}_1, \mathcal{S}_2) = \binom{N}{2}^{-2} \sum_{i,j,i<j} \left[\mathcal{C}_{ij}\mathcal{P}_{ij} + \left(1 - \mathcal{C}_{ij}\right)\left(1 - \mathcal{P}_{ij}\right)\right] \tag{6.7}$$

$\mathcal{C}_{ij}\mathcal{P}_{ij} = 1$ means that the faces i and j have the same ID in both segmentations, $\mathcal{S}_1$ and $\mathcal{S}_2$, and $\left(1 - \mathcal{C}_{ij}\right)\left(1 - \mathcal{P}_{ij}\right) = 1$ indicates that they have different IDs in $\mathcal{S}_1$ and $\mathcal{S}_2$. And the denominator is the number of possible unique pairs among $\mathcal{N}$ faces. The RI factor ranges from 0 to 1 where value of 1 shows complete matching. To be similar to the other metrics, $1 - RI\ (\mathcal{S}_1, \mathcal{S}_2)$ is usually considered as the evaluation metric so that the lower number of RI the better the segmentation matching results.

d) **Consistency Error (CE):** based on a theory about human perceptual organization, the CE factor is responsible for nested hierarchical similarities and differences in segmentations [218]. This measure is based on the computation of Local Refinement Error (LRE) of a face $\mathscr{f}_i$ between two segmentations $\mathcal{S}_1$ and $\mathcal{S}_2$. If $\mathrm{R}(\mathcal{S}, \mathscr{f}_i)$ is a segment in segmentation $\mathcal{S}$, which contains the face $\mathscr{f}_i$, then the LRE is defined as Equation 6.8:

$$\mathrm{LRE}\ (\mathcal{S}_1, \mathcal{S}_2, \mathscr{f}_i) = \frac{||\mathrm{R}(\mathcal{S}_1, \mathscr{f}_i) \backslash \mathrm{R}(\mathcal{S}_2, \mathscr{f}_i)||}{||\mathrm{R}(\mathcal{S}_1, \mathscr{f}_i)||} \tag{6.8}$$

Similar to Equation 6.3, \ is the set difference operator and ||x|| is the cardinality of set x. The LRE measure can be combined for all the faces on the mesh in two different manners to create the Global Consistency Error (GCE) and the Local Consistency Error (LCE) as follows:

$$\begin{aligned} &\mathrm{GCE}\ (\mathcal{S}_1, \mathcal{S}_2) \\ &= \frac{1}{\mathcal{N}} \min\{\sum_i \mathrm{LRE}\ (\mathcal{S}_1, \mathcal{S}_2, \mathscr{f}_i)\ , \sum_i \mathrm{LRE}\ (\mathcal{S}_2, \mathcal{S}_1, \mathscr{f}_i)\} \end{aligned} \tag{6.9}$$

$$\mathrm{LCE}\ (\mathcal{S}_1, \mathcal{S}_2) = \frac{1}{\mathcal{N}} \sum_i min\{\mathrm{LRE}\ (\mathcal{S}_1, \mathcal{S}_2, \mathscr{f}_i)\ , \mathrm{LRE}\ (\mathcal{S}_2, \mathcal{S}_1, \mathscr{f}_i)\} \tag{6.10}$$

The main difference between these two symmetric measures is related to the direction of matching, i.e. the GCE forces all local refinements to be in the same direction, while the LCE allows refinements in different directions. The generated values of GCE and LCE range from 0 to 1 where 0 indicates perfect segmentation matching and 1 indicates no similarity between segmentations. Although these two factors take into account the nested, hierarchical differences in segmentations, they suffer from a challenging issue of incorrectly generating a zero value as the result for two specific scenarios: all faces belonging to one segment or each segment containing only one face.

e) **3D Probabilistic Rand Index (3DPRI):** As can be understood from its name, this factor has been derived from the Rand Index measure by Benhabiles *et al.* [219] and is introduced to perform a quantitative comparison between a mesh segmentation algorithm and a set of ground-truth segmentations (instead of only one ground-truth). Suppose $\mathcal{S}_\mathrm{a}$ be an automatic segmentation to be compared with the set of ground-truths $\mathcal{S}_{k}$. If $\mathcal{S}_a^i$ and $\mathcal{S}_{k}^i$ are the segments IDs of face i in segmentations $\mathcal{S}_a$ and ground-truth $\mathcal{S}_{k}$ respectively then, the 3DPRI is defined as follows:

$$\text{3DPRI}\ (\mathcal{S}_\text{a}, \{\mathcal{S}_{k}\}) = \binom{\mathcal{N}}{2}^{-2} \sum_{i,j,i<j} \left[\mathcal{C}_{ij}\mathcal{P}_{ij} + (1 - \mathcal{C}_{ij})(1 - \mathcal{P}_{ij})\right] \tag{6.11}$$

Where $\mathcal{N}$ is the number of faces on the mesh model and $\mathcal{C}_{ij}$ is defined (as above in the definition of Rand Index factor) as $\mathcal{C}_{ij} = \boldsymbol{I}(\mathcal{S}_a^i \ = \mathcal{S}_a^j \ )$. But the definition of $\mathcal{P}_{ij}$ is quite different from that presented in definition of RI factor. It is defined as the probability of the faces i and j belonging to the same segments of the set of ground-truth segmentations $\{\mathcal{S}_{k}\}$:

$$\mathcal{P}_{ij} = \frac{1}{k}\sum_{k} \boldsymbol{I}(\mathcal{S}_{k}^i \ = \ \mathcal{S}_{k}^j \ ) \tag{6.12}$$

Unlike all other metrics, the 3DPRI metric generates values between 0 and 1 where a value of 0 specifies no matching and 1 is for the perfect matching. It is worthwhile to point out that, Benhabiles *et al.* [219] also provided a normalized version of 3DPRI (3DNPRI), which is widely utilized and defined for each segmentation $\mathcal{S}_\text{a}$ as Equation 6.13:

$$\text{3DNPRI}\ (\mathcal{S}_\text{a}) \ = \frac{\text{3DPRI}\ (\mathcal{S}_\text{a}, \{\mathcal{S}_{k}\}) - \text{E}[\text{3DPRI}\ (\mathcal{S}_\text{a}, \{\mathcal{S}_{k}\})]}{1 - \text{E}[\text{3DPRI}\ (\mathcal{S}_\text{a}, \{\mathcal{S}_{k}\})]} \tag{6.13}$$

Where $\text{E}[\text{3DPRI}\ (\mathcal{S}_\text{a}, \{\mathcal{S}_{k}\})]$ is the expected probabilistic Rand Index and is defined using a random segmentation $\mathcal{S}_{r}$as:

$$\text{E}[\text{3DPRI}\ (\mathcal{S}_\text{a}, \{\mathcal{S}_{k}\})] = \frac{1}{\mathcal{M}}\sum_{r=1}^{\mathcal{M}} \text{3DPRI}\ (\mathcal{S}_{r}, \{\mathcal{S}_{k_{r}}\}) \tag{6.14}$$

Here $\mathcal{M}$is the number of models in the dataset and $\{\mathcal{S}_{k_{r}}\}$ is the set of ground-truths of the model concerned by $\mathcal{S}_{r}$. The 3DNPRI generates values in range [-1, 1] where -1 indicates no similarity and +1 shows the perfect match between the automatic segmentation and the ground-truth of the model. Note that, as with

Benhabiles *et al.* mentioned in [219], the values of 3DNPRI are usually sorted in ascending order for each algorithm. So, the jth model may not be the same across the algorithms being examined.

Since each of the above metrics may provide different information about the segmentation quality, we present evaluation results for our approach based on all of these metrics.

6.4. Parameter Analysis

In this section, we investigate the effects of the utilized parameters on the segmentation results of the proposed algorithm. Examining different values for the parameters, we try to find the optimum values by which the best segmentation results are obtained. The main effective parameters used in our algorithm relate to the following concepts: Semi-concave face, Protruded parts and Weak boundary enhancement. In the following sections we analyse the effect of each parameter and select a fixed value for them in order to achieve consistent segmentation results.

a) **Semi-concave face:** Based on our definition, we consider a face as semi-concave, if its charge density is lower than density of N percent of faces in its Ring_neighbourhood (see Section 5.6.2.3 of Chapter 5). We attempt to find the best value for parameter N. Investigating different values for parameter N revealed, that the value of N has no meaningful effect on the segmentation results. However, the smaller values for N will lead to more semi-concave faces, which leads in turn to a greater post-processing task in the enhancement step. Therefore, the lower the value for N, the more weak boundary faces and thus the slower the segmentation process. It is obvious that selecting values close to 1 for N can cause the algorithm to miss a large number of boundary faces, which may result in the inability to detect proper boundaries for some boundary parts. So, N=0.7 seems to be the best choice in our segmentation algorithm.

b) **Protruded parts:** By definition, so as to detect initial segments (protruded parts), we extracted Ψ connected faces having charge density higher than the other faces on the model surface. The smaller values for Ψ will lead to the detection of smaller parts on the model. Figure 6.3 shows the horn, teats and talons detected on the surfaces of Cow and Dinopet models, respectively ($\Psi = 0.125$). On the other hand, selecting a very small value for Ψ results in over-segmentation, i.e. every small protrusion will be extracted as a distinct segment. So, we experimentally selected 0.33 for this parameter.

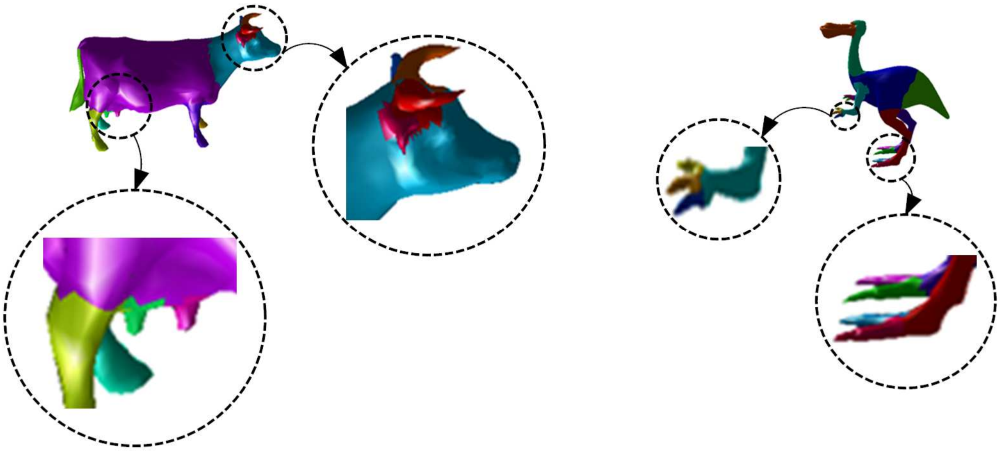


Figure 6.3, Cow and Dinopet small part extraction by selecting small values for parameter $\Psi = 0.125$.

c) **Weak boundary enhancement:** As discussed in Chapter 5, weak boundary faces are considered as acceptable boundaries, if a significant difference occurs in the amount of electrical charge on the annular regions located on the either side of the face. Several values of T have been examined so as to quantify the meaning of "significant difference" in Section 5.6.2.3.2 of Chapter 5. Figure 6.4 shows the 3DPRIs for different values of *T* on the SHREC 2012 dataset. As depicted in the figure, T=3 seems to be the choice for gaining the best result.

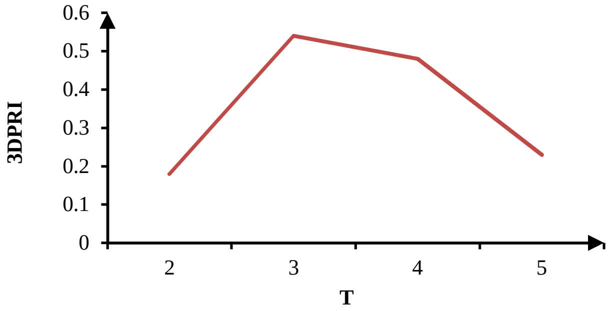


Figure 6.4. The relationship between parameter T and 3DPRI measure.

In the following sections, we will examine the evaluation metrics on the models available in two standard benchmarks (SHREC'07 and SHREC' 12) using the obtained fixed parameters.

6.5. Results on the SHREC'07 Dataset

In order to evaluate the segmentation results, we employ our proposed approach for segmenting the 380 models of SHREC'07 dataset. Figure 6.5 depicts the segmentation results on a wide variety of models from this dataset (Appendix C shows the ground truth segmentation for these models). Five other well-known approaches including Randomized Cut [138], Shape Diameter Function [129], Normalized Cut [138], Core Extraction [151] and Random Walk [108] along with the human-generated segmentations are compared to our approaches via the five evaluation factors. A complete description of these approaches can be found in Chapter 2.

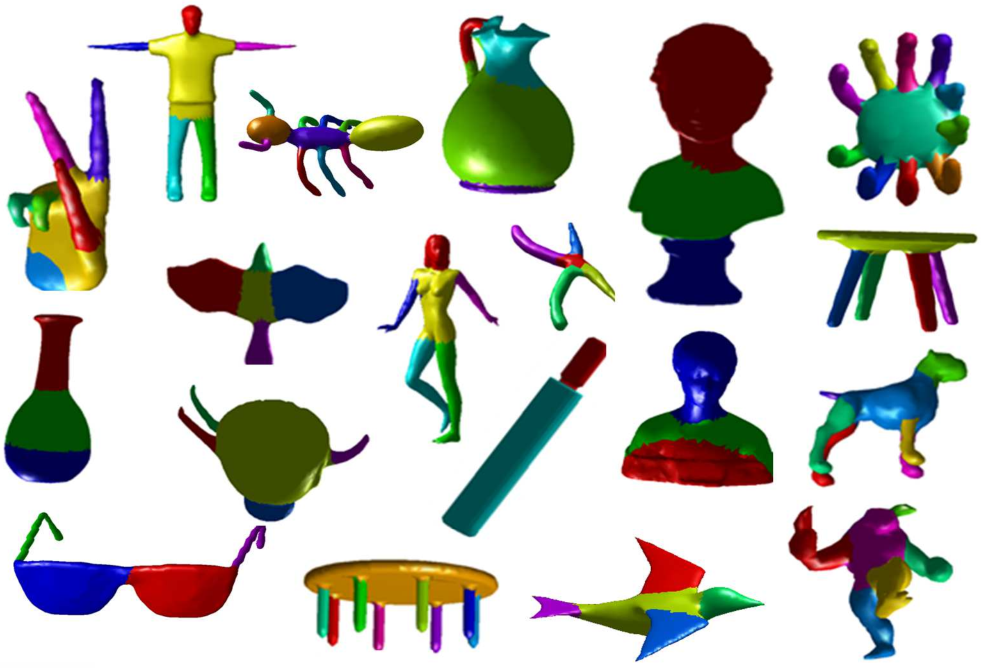
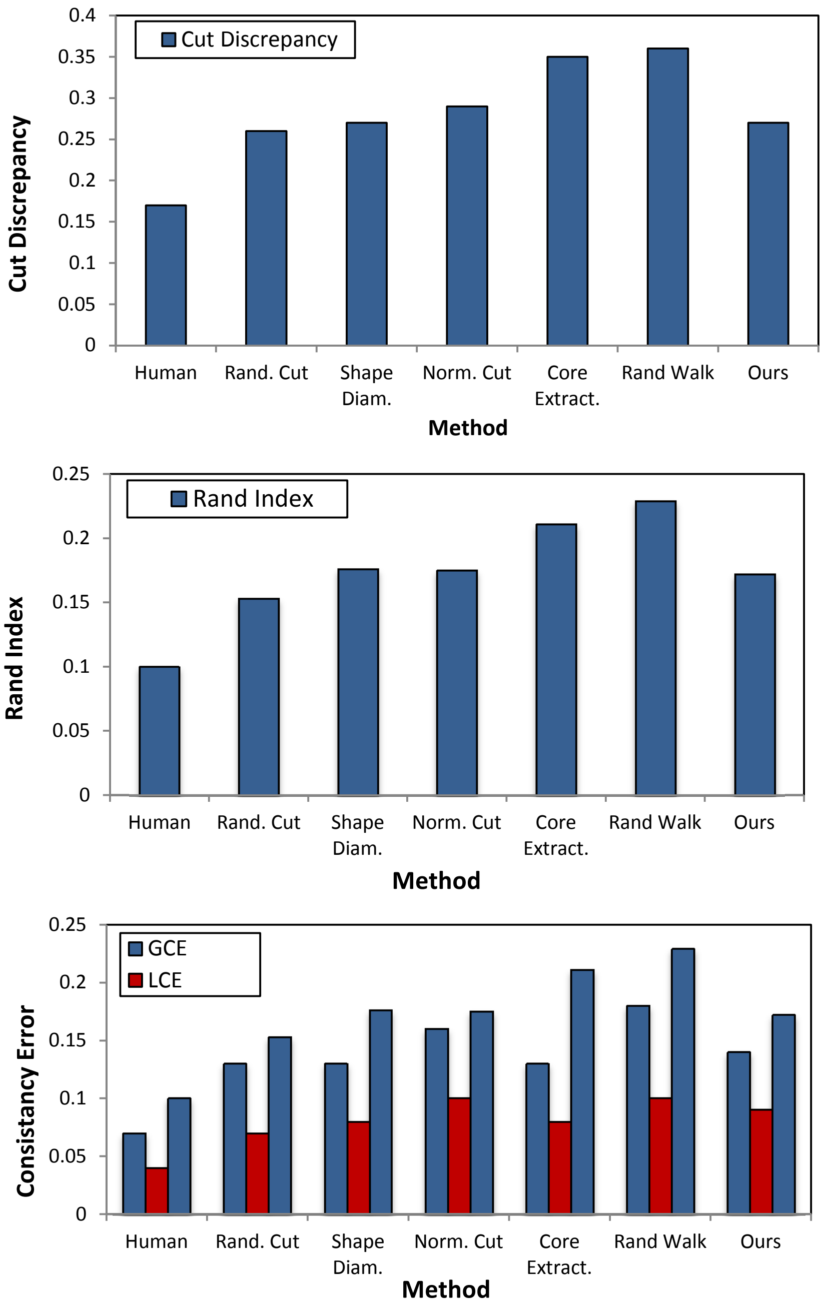


Figure 6.5, segmentation results for sample models from the SHREC'07 dataset.

6.5.1. Comparison to the Well-known Approaches

The five charts in Figure 6.6 depict the average value of the evaluation metrics for different approaches. Generally speaking, as shown in Figure 6.6, our approach performs slightly worse than the best approach reported in [136]. Although one of the metrics (CD) computes a boundary-based error and the others focus on region dissimilarity, our approach shows a consistent behaviour for all of the metrics and gains the third rank among the examined algorithms (after the Human-generated and Rank Cut approaches). However, it is notable that the Rank Cut algorithm, which outperforms all other approaches, is not a fully automatic algorithm and the user is asked to enter the number of desired segments as an input parameter. Furthermore, as mentioned in [136], this latter approach leverages a quite time-consuming process of checking both boundary and region properties, so that more than 83 seconds on average is required to segment each model on a 2GHz PC. In comparison to this approach, our algorithm partitions the models in a quite reasonable manner in terms of both time and accuracy (as will be discussed later in this chapter, ours takes an average of 24 seconds on a 2.2GHz Laptop).



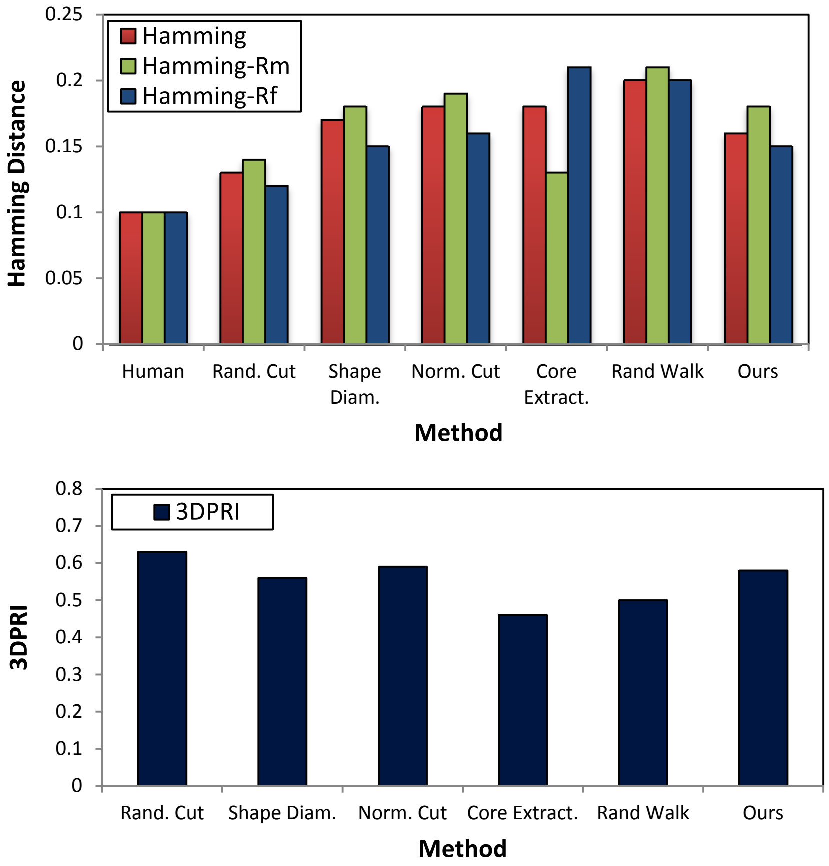


Figure 6.6, Evaluation metrics for the proposed approach and some state-of-the-art methods applied to the SHREC'07 dataset.

To perform a more accurate evaluation of the proposed approach, we have investigated the Rand Index metric for all 19 model categories separately. The data provided in Table 6.1 show a comparison of the performance of all six algorithms for each category. This table reveals that, like the other approaches, our method does not perform equally for all of the classes. That is, our approach outperforms almost all other approaches in the classes, for which the protruding parts are clearly distinguishable, e.g. Human, Octopus, Bird and Plier. In contrast, for the Bust and Vase classes, due to lack of clear boundaries between core and protruding parts, the results are not quite satisfactory. As will be discussed in Section 6.9, the main drawback of the proposed approach is related to the models having planar surfaces such as CAD and Mechanical models. The Rand Index metric of our approach for the Mechanical class illustrates this drawback, as our approach underperformed almost all other approaches for models in this class.

Table 6.1, Per-class Rand Index (%) for segmented models on SHREC'07 dataset

Algorithm Category		Rand Cut	Shape Diam.	Norm Cut	Core Ext.	Rand.Walk	Ours
Human	RI	13.1	17.9	15.8	22.5	29.5	16.7
	Rank	1	3	4	5	6	2
Cup	RI	21.9	35.8	23.6	30.7	33.4	33.2
	Rank	1	6	2	3	5	4
Glasses	RI	10.1	20.4	14.2	30.1	31.6	13.4
	Rank	1	4	3	5	6	2
Airplane	RI	12.2	9.2	18.6	25.6	26.1	16.6
	Rank	2	1	4	5	6	3
Ant	RI	2.5	2.2	4.7	6.5	6.8	2.8
	Rank	2	1	4	5	6	3
Chair	RI	18.4	11.1	9.3	18.7	16.7	17.8
	Rank	6	2	1	5	3	4
Octopus	RI	6.3	4.5	6.3	5.1	6.9	4.9
	Rank	5	1	4	3	6	2
Table	RI	38.3	18.4	9.8	24.4	13.9	17.2
	Rank	6	4	1	5	2	3
Teddy	RI	4.5	5.7	12.1	11.4	12.7	4.6
	Rank	1	3	5	4	6	2
Hand	RI	9.0	20.2	15.6	15.5	22.2	15.1
	Rank	1	5	4	3	6	2
Plier	RI	10.9	37.5	18.3	9.3	23	8.6
	Rank	3	6	4	2	5	1
Fish	RI	29.7	24.8	39.9	27.3	40.6	26.6
	Rank	4	1	5	3	6	2
Bird	RI	10.7	11.5	20.2	12.4	28	10.5
	Rank	2	3	5	4	6	1
Armadil	RI	9.2	9.0	12.0	14.1	10.7	11.2
	Rank	2	1	5	6	3	4
Bust	RI	23.2	29.9	33.2	31.5	33.5	32.4
	Rank	1	2	5	3	6	4
Mech.	RI	27.7	23.8	17.5	38.7	24.4	35.9
	Rank	3	2	1	6	4	5
Bearing	RI	12.4	11.9	18.3	39.8	27.1	16.8
	Rank	2	1	4	6	5	3
Vase	RI	13.3	23.8	23.6	22.6	28.7	26.4
	Rank	1	3	5	2	6	4
Four leg	RI	17.4	16.1	21.3	19.1	20.8	16.7
	Rank	3	1	4	5	6	2
Average RI		**15.3**	**17.6**	**17.5**	**21.1**	**22.9**	**17.2**
Total Rank		**1**	**4**	**3**	**5**	**6**	**2**

6.6. Results on SHREC'12 Dataset

To evaluate our approach more precisely, we have applied our approach to the models in the 3D mesh segmentation track of the SHREC'12 contest. Figure 6.7 displays the segmentation results on all the models of this dataset (Appendix C shows the ground truth segmentation for these models).

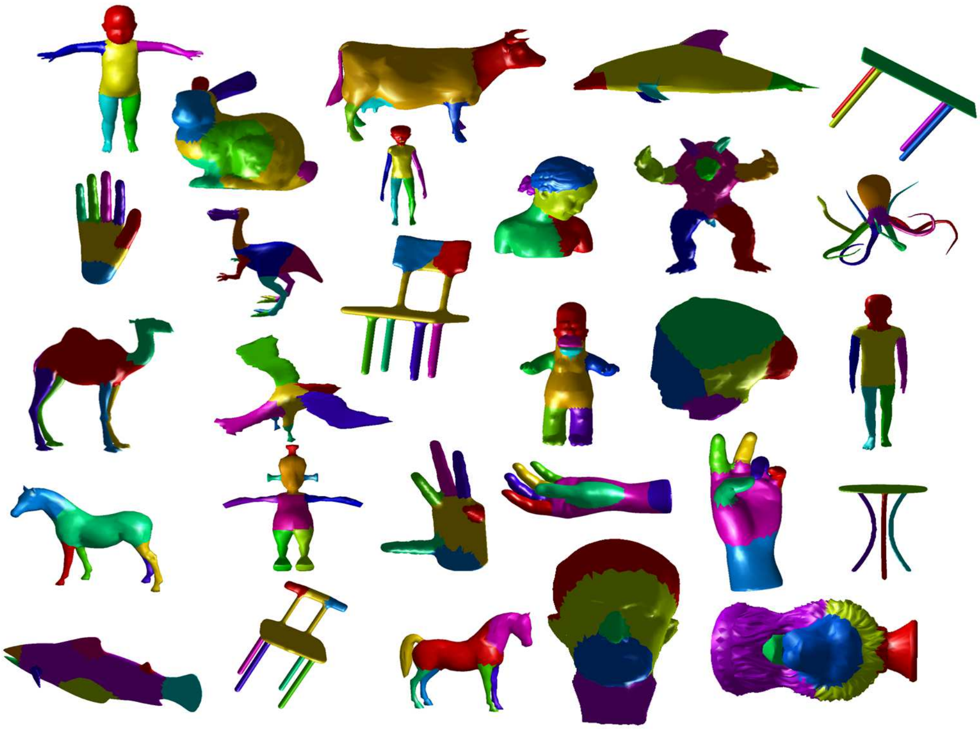


Figure 6.7, segmentation results for the models of the SHREC'12 dataset.

6.6.1. Comparison to the State-of-the-Art Approaches

The 3DNPRI metric is utilized to compare the ability of our approach to the seven state-of-the-art approaches participated in the segmentation track of SHREC'12 contest [215]. These approaches include the recently introduced approach of Box Approximation and Decomposition proposed by Huebner [220], the Plumber approach from CNRI-MATI of Genova [221], the Boundary Learning-based approach of Benhabiles *et al.* [165], Curvature Classification of Lavoué *et al.* [111], Fitting Primitive-based [169], the Reeb graph- based method of the Topology Driven approach proposed by Tierny *et al.* [147] and the famous approach of Shape Diameter Function introduced by Shapira *et al.* [129]. The detailed descriptions of these approaches are available in Chapter 2.

Table 6.2 reports the 3DNPRI factor of all of the 8 approaches participating in the contest for each category. As illustrated in this table, none of the approaches works best for all of the categories. In other words, the performance of the methods is highly class-dependent and no method has the same quality for all classes [215]. The data in the table also reveals that our approach outperforms most of the contestants of SHREC'12 and is quite comparable to the best approach, the Boundary Learning approach [165]. But it is worthwhile to point out that the Boundary Learning approach needs an off-line step, in which an objective boundary function should be learned using a set of manually segmented models. This learning step involves a time-consuming pre-processing phase, which results in slowing down the total process. The learning step takes around 10 minutes and the running time for the online step is between 1 and 9 minutes [165]. Also, it is notable that for the Human class our approach gains the first rank among all of the other approaches as its core and protruded parts are easily distinguishable. On the other hand, the segmentation results of our approach for the Bust class clarify that the proposed method cannot generate perfect results for the same reasons stated for Vase and Bust classes in the SHREC'07 dataset.

Table 6.2, 3DNPRI results for segmented models on SHREC'12, Segmentation Track.

Method Name	Animal	Bust	Furniture	Hand	Human	Global Mean	Rank
Box Approx. and Decomposition	0.52	-0.08	0.08	-0.09	0.37	**0.16**	8
Plumber	0.36	0	0.54	0.27	0.33	**0.30**	7
Boundary Learning	0.68	0.41	0.79	0.68	0.69	**0.65**	1
Curvature Classification	0.43	0.1	0.38	0.45	0.29	**0.33**	6
Fitting Primitives	0.45	0.09	0.56	0.52	0.61	**0.45**	4
Topology Driven	0.51	-0.07	0.36	0.78	0.5	**0.41**	5
Shape Diameter Function	0.62	0.24	0.85	0.19	0.66	**0.51**	3
Ours	0.66	0.08	0.72	0.57	0.70	**0.55**	2

6.6.2. Cut-Refinement Effect

As detailed in the Section 5.6.3 of the previous chapter, after locating the boundary faces in the segmentation phase, we utilize a post-processing phase of Cut-Refinement to locate boundaries more precisely. We have reported the 3DNPRI measure of segmented models before and after applying the Cut-refinement so as to examine the effect of this process. One can see the enhancement of the segmentations after the refinement process in the Figure 6.8. As illustrated in this figure, the refinement process has a more meaningful effect at the lower values of the 3DNPRI. This phenomenon can be explained as follows: for the models having easily distinguishable parts, such as Human and Hand classes (higher values of 3DNPRI), the algorithm has already generated high quality segments and so the refinement process did not have much effect on the detected boundaries. But for the challenging models, for which the 3DNPRI metric is low (such as models in the Bust class), on the other hand, the refinement process is able to increase the 3DNPRI measure by locating the boundaries more accurately.

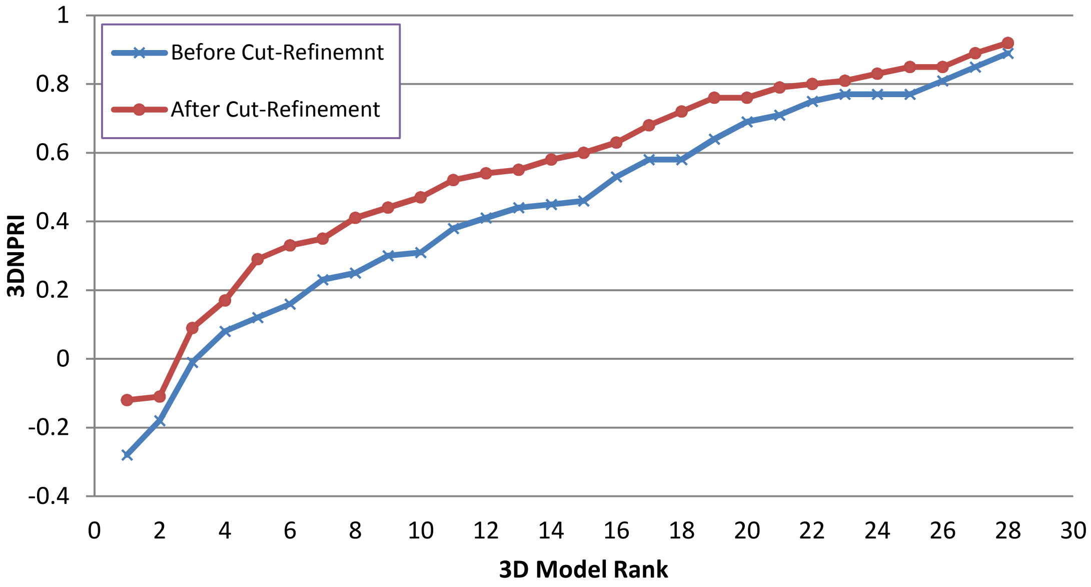


Figure 6.8, 3D-NPRI indices of the 28 models of SHREC'12 dataset sorted in increasing order before and after applying Cut-Refinement process.

6.7. Robustness

In this section we will investigate the robustness of the proposed approach to different changes and transformations namely noise, deformation and tessellation.

6.7.1. Robustness to Noise

Since the main core of our approach is based on the charge density distribution over the model surface, our segmentation approach can benefit from all of the characteristics of charge density, especially its robustness to noise (See Chapter 3). To test this, noisy models were created by randomized displacement of the vertex coordinates determined by noise level $n\ell$ ($n\ell$ is the ratio of largest displacement to the longest edge of the object's bounding box) and we applied our segmentation approach to them. One can see in Figure 6.9 the segmentation results of our approach on a human model having different levels of noise. The generated segments show the ability of the applied approach in decomposing noisy models into meaningful parts in comparison to the original models.

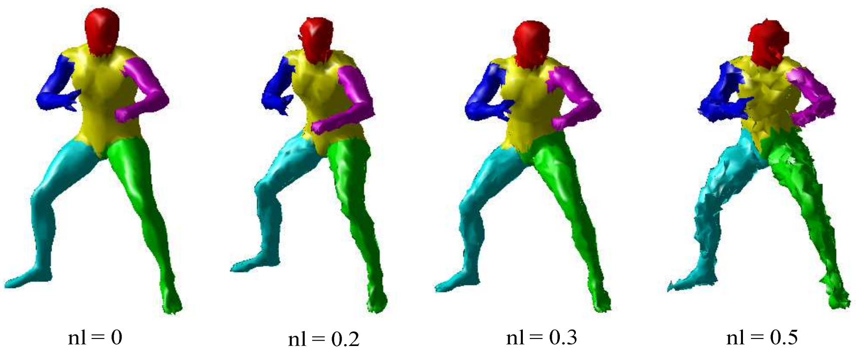


Figure 6.9. Segmentation of (leftmost) an original human model and (other three models) noisy human model at different noise levels.

Also, the 3DNPRI measures for various levels of noisy models of SHREC'12 dataset, as reported in Figure 6.10, supports our claim that the proposed algorithm is largely insensitive to noise.

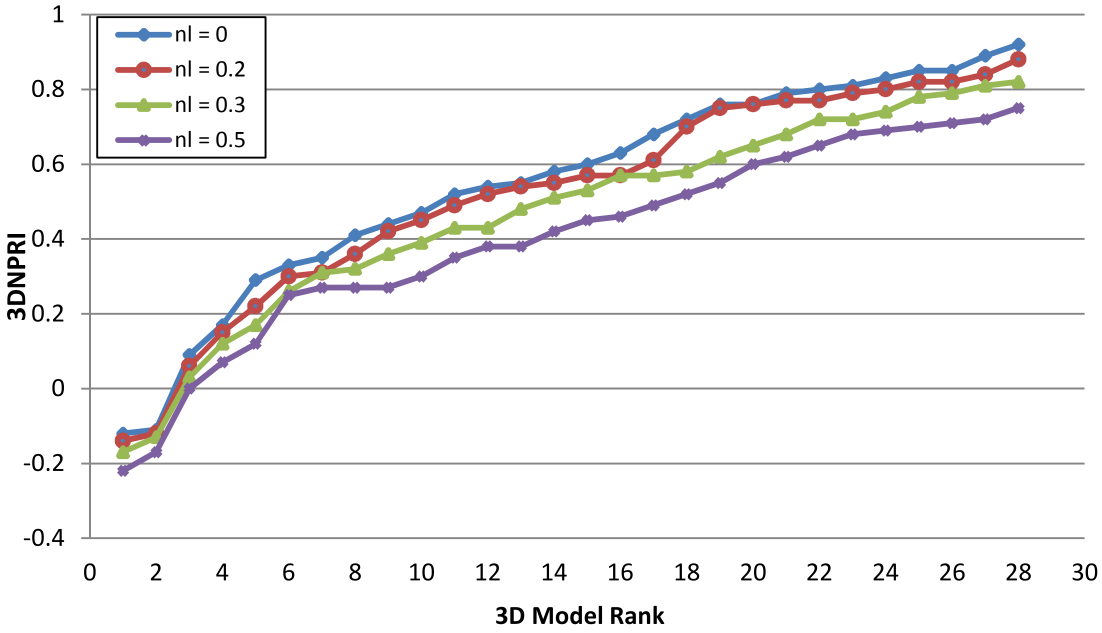


Figure 6.10, 3D-NPRI indices of the 28 models of SHREC'12 dataset sorted in increasing order for different levels of noise

6.7.2. Robustness to deformation

One of the major applications of part decomposition algorithms, as stated in Chapter 5, is to utilize the extracted segments in comparing the models based on resemblance of their parts [210, 102]. Therefore, it is crucial for the segmentation algorithm to be insensitive to model deformations. The charge density distribution, which has been utilized in this work, conveys both local and global structures of the models [171]. On the other hand, the local features have usually no meaningful effect on the coarse characteristics of the model surface, such as face concavity and boundary detection. As a result, the proposed algorithm is expected to be quite insensitive to different deformations. The similarity of the extracted parts of a human model in different poses, as shown in Figure 6.11, confirms the robustness of the proposed approach to model deformations. As a result, it is reasonable to expect the associated part-based retrieval system to offer high matching ability among similar models in different poses.

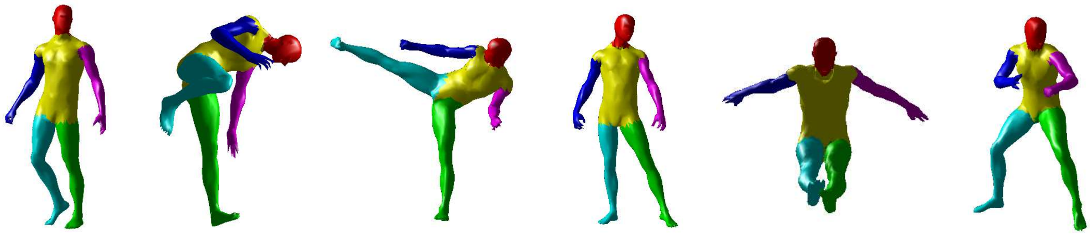


Figure 6.11, Similar segments extracted for a human model in different poses.

6.7.3. Sensitivity to Mesh Tessellation

Since our segmentation approach works based upon both local and global features of the model surface, it is very interesting to see that the combination of these features results in insensitivity to the mesh tessellations. As depicted in Figure 6.12, the simplified models possessing different numbers of faces have been segmented quite similarly. Although the higher the face number, the more accurate the segmentation, the segmentation result for a very coarse model (the right-most model in Figure 6.12 having 1000 faces) is still acceptable. However, the jagged boundaries present in the neck part of the very coarse models (models with 1000 and 2000 faces) result from the effect of the simplification process on concave faces. These irregularities can be easily smoothed by the cut-refinement process to enhance the boundaries. As the result, the proposed method can be considered as a tessellation-insensitive approach, which leads to segment the simplified models in a fast and efficient manner.

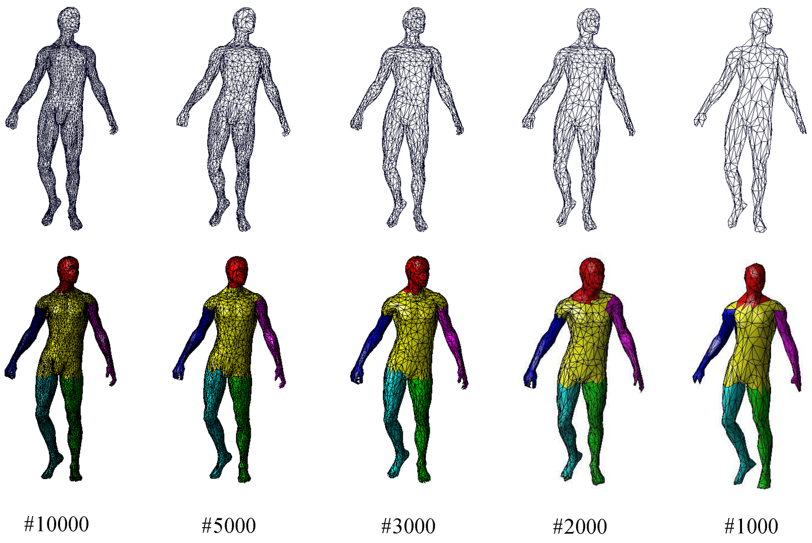


Figure 6.12, Similar extracted segments for a human model at various levels of tessellation.

6.8. Comparison With the Other Charge Density-Based Approach

As discussed before in Chapter 5, the charge density distribution has been previously leveraged for model decomposition in the work of Wu and Levine [171]. Their approach focuses only on the concavity of the mesh for detecting the boundary parts, i.e. the boundary parts must fully contain concavities to be detected accurately, which is not the case for lots of models. We have implemented and applied their algorithm on a variety of models in recently published datasets. Figure 6.13 shows the detected boundaries by their algorithm on two simple models. As expected, due to the unavailability of concavities in the boundary part, the algorithm cannot extract the boundary parts properly. Comparing these results to those of ours shown in the second row of Figures 6.13, reveals the superiority of our approach in boundary detection and then in model decomposition.

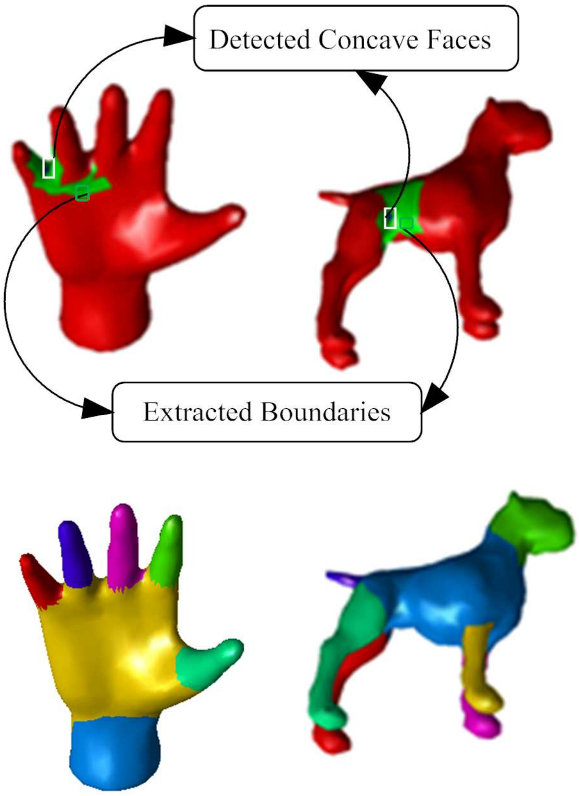


Figure 6.13, First row: examples of detected boundaries (green parts) by Wu and Levine [171] for concave faces located in small rectangles. Second row: Segmentation results of our approach for the same models.

We have also applied their approach on the 28 models of the SHREC'12 dataset. The 3DNPRI factor depicted in Figure 6.14, illustrates the huge difference in performance of our and their approaches.

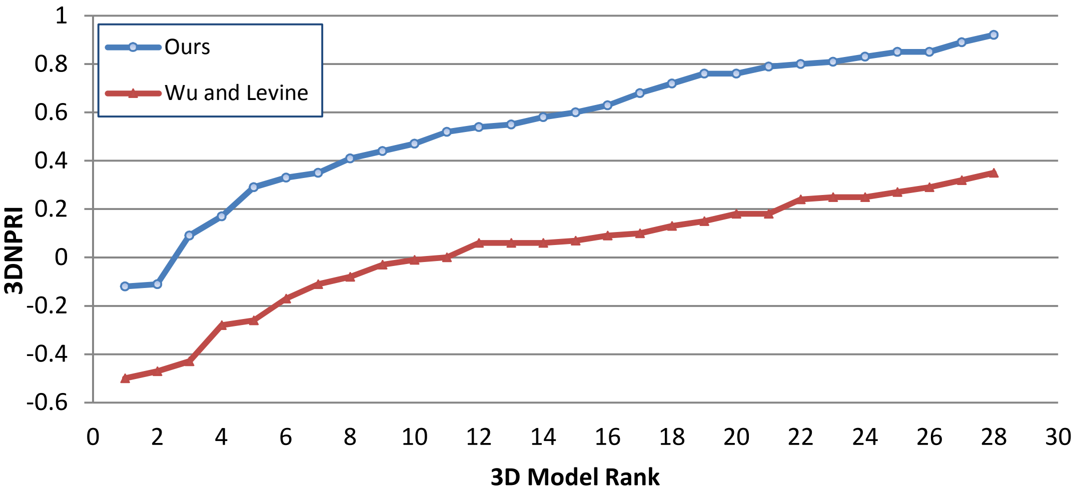


Figure 6.14, 3D-NPRI indices of the 28 models of SHREC'12 dataset sorted in increasing order for Our Method and a similar method of Wu and Levine [171].

6.9. Limitations

Despite the strong ability of the proposed algorithm in decomposing complex mesh models into meaningful parts, it suffers from some limitations:

- Like other protrusion-oriented approaches [154], our approach has been designed for decomposing models having relatively strong protrusions attached to the core part of model. As a result, the current work has limitations in decomposing models containing small protrusions such as the Bust and Vase classes of the SHREC'07 dataset as well as the Bunny model in SHREC'12 dataset. (See Figures 6.5 and 6.7 as well as Tables 6.1 and 6.2).
- Furthermore, since the region growing step of our algorithm exploits geodesic distance between the representative and boundary faces of each segment, for models possessing very planar surfaces (or non- cylindrical parts), such as CAD models, the detected boundary areas may not be perfectly fine (see Figure 6.15). The Rand Index factor of the Mechanical class in Table 6.1 exposes the weakness of our approach in such a class of models.

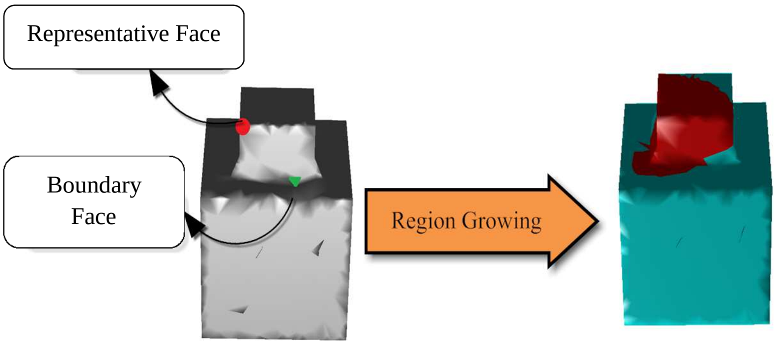


Figure 6.15, Inappropriately detected segment in a Mechanical model.

6.10. Complexity

The robustness of the proposed approach has been shown against tessellation of the model surface in Section 6.7.3. So, we simplified models by reducing the number of faces to 5000. It enables us to do all of the process in a faster and more efficient way. We implemented our algorithm in MATLAB and some of its critical parts were written in C++, connected to MATLAB using the MEX interface.

The total time for the whole segmentation process for each simplified model, including charge density distribution simulation, canonical form calculation, protruded part extraction, boundary detection and boundary refinement, takes an average of 24 seconds, out of which the canonical form calculation took 7.2 seconds and the boundary refinement about 11 seconds running on a HP laptop with Intel 2.2GHz CPU.

Let N be the number of faces in the simplified models (5000 in our case). So the complexity of the proposed approach contains the following steps: the canonical form calculation of the models takes $O\ (N^2*no\_iter)$ where *no_iter* is 10 in our implementation. The charge density distribution costs $O(N^2)$ [171]. The protrusion area extraction takes $O\ (N)$ for initial protruded area detection and $O(N)$ for the representative faces identification. The boundary detection phase comprises two steps: semi-concave face detection, which is bounded by $O(N)$, and the boundary refinement step for finding the final boundary faces, which takes $O(M*R)$, where M is the number of detected semi-concave faces and R is the number of representative faces identified on the surface of models (or the number of initial disjoint segments). Finally, the complexity of boundary refinement is $O(N_{ng}^2 \log(N_{ng}))$, where N_{ng} is the number of nodes in the network graph [153]. Thus, the overall time complexity for the current approach is $O(N^2*no\_iter + N^2 + N + N + N + M*R + N_{ng}^2 \log (N_{ng})) = O(11 N^2 + 3N + M*R + N_{ng}^2 \log (N_{ng})) = O(N^2 + M*R + N_{ng}^2 \log (N_{ng})) = O(N^2)$.

Table 6.3 reports the average required time for segmentation of models available in the SHREC’07 dataset along with the Rand Index measure of the examined algorithms. One can conclude from this table that the proposed approach can perform segmentation in a quite reasonable time. The fastest approach, Random Walk, is not fully-automatic and the user should provide the system with the number of desired segments. It also suffers from inaccurate parts extraction as identified by the Rand Index metric. The Randomized Cut approach, on the other hand, is the slowest approach for which the segmentation quality is significantly higher than the others. Our approach provides an acceptable result in terms of both speed and quality.

It is worthwhile to point out that by considering both time and accuracy the Shape Diameter Function is the best approach among all of the examined algorithms.

Table 6.3, the average required time and Rand Index metric for various approaches on the SHREC'07 dataset

Method Name	Average Time (Sec.) Run on 2 GHz PC	Rand Index (%)
Randomized Cut	83.8	15.3
Shape Diameter Function	8.9	17.6
Normalized Cut	49.4	17.5
Core Extraction	19.5	21.1
Random Walk	1.4	22.9
Ours (Run on 2.2 GHz Laptop)	24.0	17.2

6.11. Conclusion

In this chapter, several experiments were conducted to assess the performance of the proposed segmentation algorithm. Various evaluation metrics were utilized so as to evaluate the generated segments on the models available in two standard benchmarks of theSHREC'07 and SHREC'12 contests.

After setting the required parameters experimentally, we have performed a thorough investigation on the results in order to tackle the challenging problems in the segmentation field (e.g. accuracy, speed, robustness and hierarchy).

We have utilized five metrics with the intention of assessing the accuracy of the extracted segments. Our approach achieved quite acceptable results in comparison to the existing approaches as well as to a similar approach proposed by Wu and Levine [171]. Furthermore, the robustness against noise, deformation and mesh tessellation is another remarkable characteristic of our method, which makes it applicable in a variety of domains such as partial matching and others. In terms of offering hierarchical segmentation, although our approach does not provide a complete hierarchy of segments, for the models having mixed cores, it can be applied hierarchically to decompose the extracted cores into their components.

Generally speaking, the experimental results presented in this chapter indicate that the proposed approach outperforms most of the state-of-the-art methods and is quite comparable to the best ones.

The drawbacks of the proposed approach have also been examined, which include inadequate ability in decomposing models having planar surfaces (such as CAD and Mechanical models) and the models containing small protrusions such as the Bust and Vase models.

7

Conclusions and Future Work

7.1. Introduction

In this dissertation, we have proposed new approaches for processing 3D models. We have introduced three different shape descriptors by which 3D models can be effectively and efficiently searched and retrieved from standard datasets. Additionally, a robust, efficient and fully automatic approach for decomposing 3D models into their meaningful components has been presented.

In the pages that follow, for the last time, we simply discuss the proposed approaches for retrieval and segmentation, as well as other contributions we have made. Finally, we will describe the possibilities for future research.

7.2. Summary of contributions

The content of the current thesis can be divided into two main parts:

- **Part I**: 3D Model retrieval (Chapters 3 and 4): in this part three shape descriptors have been introduced and their performances on the models in standard datasets have been investigated.
- **Part II**: 3D Model Segmentation (Chapters 5 and 6): the second part of the current work was dedicated to introducing a new technique for decomposing 3D models into semantically meaningful parts. The proposed approach can be considered as an improvement on the simple approach proposed by Wu and Levine [171].

The proposed techniques for both retrieval and segmentation of 3D models can be detailed as follows:

7.2.1. SilPH Descriptor (Chapter 3)

This first contribution is the **Sil**houette **P**oisson **H**istogram (SilPH) descriptor which belongs to the view-based category of descriptors (See Chapter 2 for classification of available descriptors). It extracts a set of 2D silhouettes from pre-defined view angles. The solutions to the 2D Poisson Equation over the silhouettes assign to each pixel a number, which acts as a local descriptor. A histogram accumulates the assigned numbers and describes the views efficiently. As practically investigated in Chapter 4, the SilPH descriptor offers valuable characteristics viz. higher retrieval accuracy than its 3D counterpart, robustness to noise and mesh tessellation.

7.2.2. BoF-CDD Descriptor (Chapter 3)

For the second and third descriptors, a well-known fact from electrical physics has been utilized: "when electrical charge is distributed on the surface of a solid, it tends to accumulate on the sharp convexities and diminish in sharp concavities". To simulate this fact, the Finite Element Method (FEM) was employed to specify charge distribution on the facets of the models.

The second contribution employs density of distributed charge on each facet of a mesh model and considers it as a local descriptor. The Bag-of-Feature framework enabled us to compare and match the models globally, using the local features. The proposed **B**ag-**o**f-**F**eature **C**harge **D**ensity **D**escriptor (BoF-CDD) presents high retrieval ability as well as insensitivity to Noise, Deformation and Tessellations. It can also be extended, by some post-processing phases, to support partial matching.

7.2.3. DP-CS-ECD Descriptor (Chapter 3)

This contribution is a hybrid shape descriptor which uses the above fact of charge distribution to formulate 3D model features. Unlike the BoF-CDD, the **D**ense **P**atch-**C**oncentric **S**phere-**E**lectrical **C**harge **D**istribution (DP-CS-ECD) descriptor exploits the amount of charge in each facet. After the amount of charge has been computed, the models are enclosed in a set of concentric spheres and the total amount of charge between adjacent spheres creates a numerical shape descriptor. Finally, the discrimination ability of the descriptor is improved by considering the number of Dense Patches on the surface of models (See Chapter 3 to more detail about the Dense Patches).

The descriptor presented higher discrimination ability then the BoF-CDD. But, as shown in Chapter 4, it cannot be employed to support partial matching.

7.2.4. Charge Density-Based Segmentation (Chapter 5)

Unlike the previous contributions, this is related to decomposing 3D models into their semantic components. It employs the same fact of charge distribution on the surface of models, which has been used in contributions 2 and 3. By using this fact, the convex and concave parts of the models are detected, which are able to locate the start and end parts of each segment, respectively. The ability of the technique has been boosted using some extensions added to the algorithm's pipeline. The proposed approach has quite beneficial advantages over similar approaches in the literature [171]. Additionally, since the charge distribution acts as a fundamental cornerstone of the approach, it inherits all of its characteristics such as insensitivity to Noise and Pose deformations.

7.2.5. Surveys of Available Descriptors and Segmentation Approaches (Chapter 2)

In Chapter 2, we have presented complete and quite up-to-date surveys on both 3D shape descriptors and segmentation techniques. The available approaches have been classified into various groups for which sample works try to clarify the idea behind each class. Furthermore, the advantages and shortcomings of each class give a clear guide to the researcher to find their proper way of work.

Additionally, later in Chapter 4, a literature review on available Partial Matching algorithms has been discussed. The aforementioned surveys are able to help new researchers to find their proper ways in these fields.

7.3. Directions for Future Research

The ideas and concepts in this research offer interesting avenues for future research. In the sequel, we identify some of these possibilities as the most important ones:

7.3.1. Toward Partial Matching

The practical test in Chapter 3 showed that among the proposed shape descriptors, only the BoF-CDD has the potential of supporting partial matching. But, since the charge density of each face has contributions from all of the other faces, altering a part of a model can affect the amount/density of charge on the whole model. This led to achieving results which were not as satisfactory as expected. One possibility for supporting partial matching is using a proper statistical pattern recognition approach to discover available patterns in the charge distribution among similar subparts. Statistical pattern recognition could be expected to provide users with a high quality partial matcher, as the similar subparts in dissimilar models follow the same patterns of charge distribution. In a finger, for example, the amount/density of charge increase as faces get closer to the finger tip.

Another possible technique for matching partial similar models could be measuring the global similarity using the similarity of their subparts. That is, the models are firstly segmented into their components and then each component is treated as an individual model for which the charge distribution is simulated. Finally, the similarity between two models is scored based on the number of common components they share. The more common subparts, the higher the similarity between models. Although problems such as the spatial relationship among the components should be considered in the retrieval process, weighting schemes such as TF-IDF [96] are able to boost the retrieval ability. The TF-IDF is shown to be superior to simple, non-weighted approaches [88].

7.3.2. Extending the Retrieval Algorithm to Support Relevance Feedback

Relevance Feedback (RF) is a valuable feature of some retrieval systems such as CBIR and 3DOR to enhance the discriminative ability in such systems. The idea behind it is to bridge the semantic gap between the abstract, high level user intention and the low level data representation and processing. It has shown its ability in retrieving more accurate results compared to similar original retrieval systems [24, 123, 89]. Consequently, the other possible improvement to the proposed retrieval techniques is to propose a system to utilize RF in order to include the user's needs in a beneficial way.

7.3.3. Extending Segmentation Approach

As discussed in Chapter 6, the major limitation of the proposed segmentation approach is related to the models possessing very planar surfaces, such as CAD and Mechanical models. Fitting primitives (such as cube, cylinder and sphere) to the components of such models could be expected to overcome this shortcoming. Therefore a hybrid approach, in which a primitive-fitting approach acts as a post-processing phase, seems to be a very promising approach.

Bibliography

[1] Funkhouser, Thomas, Patrick Min, Michael Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. "A search engine for 3D models."*ACM Transactions on Graphics (TOG)* 22, no. 1 (2003): 83-105.

[2] Amberg, Brian, Andrew Blake, Andrew Fitzgibbon, Sami Romdhani, and Thomas Vetter. "Reconstructing high quality face-surfaces using model based stereo." In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1-8. IEEE, 2007.

[3] Dugelay, Jean-Luc, Atilla Baskurt, and Mohamed Daoudi, eds. *3D object processing: compression, indexing and watermarking*. John Wiley & Sons, 2008.

[4] Bustos, Benjamin, Daniel Keim, Dietmar Saupe, and Tobias Schreck. "Content-based 3D object retrieval." *Computer Graphics and Applications, IEEE* 27, no. 4 (2007): 22-27.

[5] Bimbo, Alberto Del, and Pietro Pala. "Content-based retrieval of 3D models."*ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 2, no. 1 (2006): 20-43.

[6] Iyer, Natraj, Subramaniam Jayanti, Kuiyang Lou, Yagnanarayanan Kalyanaraman, and Karthik Ramani. "Three-dimensional shape searching: state-of-the-art review and future trends." *Computer-Aided Design* 37, no. 5 (2005): 509-530.

[7] Qin, Zheng, Ji Jia, and Jun Qin. "Content based 3D model retrieval: A survey." In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pp. 249-256. IEEE, 2008.

[8] Tangelder, Johan WH, and Remco C. Veltkamp. "A survey of content based 3D shape retrieval methods." *Multimedia tools and applications* 39, no. 3 (2008): 441-471.

[9] Li, Bo, Yijuan Lu, Afzal Godil, Tobias Schreck, Benjamin Bustos, Alfredo Ferreira, Takahiko Furuya *et al*. "A comparison of methods for sketch-based 3D shape retrieval." *Computer Vision and Image Understanding* 119 (2014): 57-80.

[10] Liu, Zhen-Bao, Shu-Hui Bu, Kun Zhou, Shu-Ming Gao, Jun-Wei Han, and Jun Wu. "A survey on partial retrieval of 3D shapes." *Journal of Computer Science and Technology* 28, no. 5 (2013): 836-851.

[11] Liu, Qiong. "A Survey of Recent View-based 3D Model Retrieval Methods."*arXiv preprint arXiv:1208.3670* (2012).

[12] Riesenhuber, Maximilian, and Tomaso Poggio. *Computational models of object recognition in cortex: A review*. No. AI-MEMO-1695. MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 2000.

[13] http://en.wikipedia.org/wiki/Silhouette

[14] Vrani´c DV (2004) 3D model retrieval. PhD thesis, University of Leipzig

[15] Lian, Zhouhui, Afzal Godil, and Xianfang Sun. "Visual similarity based 3D shape retrieval using bag-of-features." In *Shape Modeling International Conference (SMI), 2010*, pp. 25-36. IEEE, 2010.

[16] Shih, Jau-Ling, Chang-Hsing Lee, and Jian Tang Wang. "A new 3D model retrieval approach based on the elevation descriptor." *Pattern Recognition* 40, no. 1 (2007): 283-295.

[17] Furuya, Takahiko, and Ryutarou Ohbuchi. "Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features." In *Proceedings of the ACM international conference on image and video retrieval*, p. 26. ACM, 2009.

[18] Lowe, David G. "Distinctive image features from scale-invariant keypoints."*International journal of computer vision* 60, no. 2 (2004): 91-110.

[19] Lian, Zhouhui, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita *et al*. "SHREC'11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes." *3DOR* 11 (2011): 79-88.

[20] Napoléon, Thibault, Tomasz Adamek, Francis Schmitt, and Noel E. O'Connor. "Multi-view 3D retrieval using silhouette intersection and multi-scale contour representation." (2007).

[21] Johnson, Andrew E., and Martial Hebert. "Using spin images for efficient object recognition in cluttered 3D scenes." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 21, no. 5 (1999): 433-449.

[22] Li, Xiaolan, and Afzal Godil. "Investigating the bag-of-words method for 3D shape retrieval." *EURASIP Journal on Advances in Signal Processing* 2010 (2010): 5.

[23] Sfikas, Konstantinos, Ioannis Pratikakis, and Theoharis Theoharis. "3D object retrieval via range image queries based on SIFT descriptors on panoramic views." In *Proceedings of the 5th Eurographics conference on 3D Object Retrieval*, pp. 9-15. Eurographics Association, 2012.

[24] Papadakis, Panagiotis, Ioannis Pratikakis, Theoharis Theoharis, and Stavros Perantonis. "PANORAMA: A 3D shape descriptor based on panoramic views for

unsupervised 3D object retrieval." *International Journal of Computer Vision* 89, no. 2-3 (2010): 177-192.

[25] Sfikas, Konstantinos, Ioannis Pratikakis, Anestis Koutsoudis, Michalis Savelonas, and Theoharis Theoharis. "3D Object Partial Matching Using Panoramic Views." In *New Trends in Image Analysis and Processing–ICIAP 2013*, pp. 169-178. Springer Berlin Heidelberg, 2013.

[26] Furuya, Takahiko, and Ryutarou Ohbuchi. "Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features." In *Proceedings of the ACM international conference on image and video retrieval*, p. 26. ACM, 2009.

[27] Chen, Ding-Yun, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. "On visual similarity based 3D model retrieval." In *Computer graphics forum*, vol. 22, no. 3, pp. 223-232. Blackwell Publishing, Inc, 2003.

[28] Ding, Ke, Wei Wang, and Yunhui Liu. "3D model retrieval using Bag-of-View-Words." *Multimedia Tools and Applications* (2013): 1-22.

[29] Hamerly, Greg, and Charles Elkan. "Learning the k in k-means." In *NIPS*, vol. 3, pp. 281-288. 2003.

[30] Gao, Yue, Qionghai Dai, and Nai-Yao Zhang. "3D model comparison using spatial structure circular descriptor." *Pattern Recognition* 43, no. 3 (2010): 1142-1151.

[31] Li, Pengjie, Huadong Ma, and Anlong Ming. "Combining topological and view-based features for 3D model retrieval." Multimedia tools and applications 65.3 (2013): 335-361.

[32] Ma, Wan-Chun, Fu-Che Wu, and Ming Ouhyoung. "Skeleton extraction of 3d objects with radial basis functions." In *Shape Modeling International, 2003*, pp. 207-215. IEEE, 2003.

[33] Hilaga, Masaki, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. "Topology matching for fully automatic similarity estimation of 3D shapes." In*Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp. 203-212. ACM, 2001.

[34] Belongie, Serge, Jitendra Malik, and Jan Puzicha. "Shape matching and object recognition using shape contexts." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, no. 4 (2002): 509-522.

[35] Rubner, Yossi, Carlo Tomasi, and Leonidas J. Guibas. "The earth mover's distance as a metric for image retrieval." *International Journal of Computer Vision* 40, no. 2 (2000): 99-121.

[36] Atmosukarto, Indriyati, and Linda G. Shapiro. "3D object retrieval using salient views." International journal of multimedia information retrieval 2.2 (2013): 103-115.

[37] Ansary, Tarik Filali, Mohamed Daoudi, and Jean-Philippe Vandeborre. "A bayesian 3-d search engine using adaptive views clustering." Multimedia, IEEE Transactions on 9.1 (2007): 78-88.

[38] Besl, Paul J., and Ramesh C. Jain. "Three-dimensional object recognition."*ACM Computing Surveys (CSUR)* 17, no. 1 (1985): 75-145.

[39] Yoon, Sang Min, Maximilian Scherer, Tobias Schreck, and Arjan Kuijper. "Sketch-based 3D model retrieval using diffusion tensor fields of suggestive contours." In *Proceedings of the international conference on Multimedia*, pp. 193-200. ACM, 2010.

[40] Furuya, Takahiko, and Ryutarou Ohbuchi. "Ranking on cross-domain manifold for sketch-based 3D model retrieval." In *Cyberworlds (CW), 2013 International Conference on*, pp. 274-281. IEEE, 2013.

[41] Li, Bo, Tobias Schreck, Afzal Godil, Marc Alexa, Tamy Boubekeur, Benjamin Bustos, Jipeng Chen *et al*. "SHREC'12 track: sketch-based 3D shape retrieval." In *Proceedings of the 5th Eurographics conference on 3D Object Retrieval*, pp. 109-118. Eurographics Association, 2012.

[42] Li, Bo, Yijuan Lu, Afzal Godil, Tobias Schreck, Masaki Aono, Henry Johan, Jose M. Saavedra, and Shoki Tashiro. "SHREC'13 Track: Large Scale Sketch-Based 3D Shape Retrieval." In *3DOR*, pp. 89-96. 2013.

[43] Daras, Petros, and Apostolos Axenopoulos. "A 3D shape retrieval framework supporting multimodal queries." International Journal of Computer Vision 89.2-3 (2010): 229-247.

[44] Li, Bo, Yijuan Lu, Afzal Godil, Tobias Schreck, Benjamin Bustos, Alfredo Ferreira, Takahiko Furuya *et al*. "A comparison of methods for sketch-based 3D shape retrieval." Computer Vision and Image Understanding 119 (2014): 57-80.

[45] Yoon S.M, Gang-Joon Yoon G. J., Schreck T. "User drawn sketch based 3D object retrieval using sparse coding", Multimedia Tools and Applications (2014).

[46] Li, Bo, and Henry Johan. "Sketch-based 3D model retrieval by incorporating 2D-3D alignment." Multimedia tools and applications 65.3 (2013): 363-385.

[47] Kazhdan, Michael, Thomas Funkhouser, and Szymon Rusinkiewicz. "Rotation invariant spherical harmonic representation of 3D shape descriptors." In*Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 156-164. Eurographics Association, 2003.

[48] DeCarlo, Doug, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. "Suggestive contours for conveying shape." In *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 848-855. ACM, 2003.

[49] Basser, Peter J., James Mattiello, and Denis LeBihan. "MR diffusion tensor spectroscopy and imaging." *Biophysical journal* 66, no. 1 (1994): 259-267.

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

[51] Lee, Honglak, Alexis Battle, Rajat Raina, and Andrew Y. Ng. "Efficient sparse coding algorithms." *Advances in neural information processing systems* 19 (2007): 801.

[52] Zhou, Dengyong, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. "Learning with Local and Global Consistency." In *NIPS*, vol. 16, pp. 321-328. 2003.

[53] Eitz, Mathias, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. "Sketch-based shape retrieval." *ACM Trans. Graph.* 31, no. 4 (2012): 31.

[54] Li, Bo, and Henry Johan. "View context: a 3D model feature for retrieval." In*Advances in Multimedia Modeling*, pp. 185-195. Springer Berlin Heidelberg, 2010.

[55] Eitz, Mathias, Ronald Richter, Tamy Boubekeur, Kristian Hildebrand, and Marc Alexa. "Sketch-based shape retrieval." ACM Trans. Graph. 31, no. 4 (2012): 31.

[56] Ohbuchi, Ryutarou, Masatoshi Nakazawa, and Tsuyoshi Takei. "Retrieving 3D shapes based on their appearance." In *Proceedings of the 5th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 39-45. ACM, 2003.

[57] Li, Xiaolan, Afzal Godil, and Asim Wagan. "Spatially enhanced bags of words for 3D shape retrieval." In *Advances in Visual Computing*, pp. 349-358. Springer Berlin Heidelberg, 2008.

[58] Osada, Robert, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. "Shape distributions." ACM Transactions on Graphics (TOG) 21, no. 4 (2002): 807-832.

[59] Ohbuchi, Ryutarou, Takahiro Minamitani, and Tsuyoshi Takei. "Shape-similarity search of 3D models by using enhanced shape functions." International Journal of Computer Applications in Technology 23.2 (2005): 70-85.

[60] Wu, Yuanhao, Ling Tian, and Chenggang Li. "High efficient methods of content-based 3D model retrieval." Chinese Journal of Mechanical Engineering 26.2 (2013): 248-256.

[61] Ankerst, Mihael, Gabi Kastenmüller, Hans-Peter Kriegel, and Thomas Seidl. "3D shape histograms for similarity search and classification in spatial databases." In *Advances in Spatial Databases*, pp. 207-226. Springer Berlin Heidelberg, 1999.

[62] Horn, Berthold Klaus Paul. "Extended gaussian images." *Proceedings of the IEEE* 72, no. 12 (1984): 1671-1686.

[63] Zhang, Jiqi, Hau-San Wong, and Zhiwen Yu. "3D model metrieval based on volumetric extended gaussian image and hierarchical self organizing map." In*Proceedings of the 14th annual ACM international conference on Multimedia*, pp. 121-124. ACM, 2006.

[64] Alfredo Ferreira, "A Thesaurus based Approach to 3D Shape Retrieval", PhD Thesis, Technical University of Lisbon, 2009.

[65] Wang, Dingwen, Jiqi Zhang, Hau-San Wong, and Yuanxiang Li. "3D model retrieval based on multi-shell extended Gaussian image." In *Advances in Visual Information Systems*, pp. 426-437. Springer Berlin Heidelberg, 2007.

[66] Paquet, Eric, and Marc Rioux. "Nefertiti: a query by content software for three-dimensional models databases management." In *3-D Digital Imaging and Modeling, 1997. Proceedings., International Conference on Recent Advances in*, pp. 345-352. IEEE, 1997.

[67] Paquet, Eric, Marc Rioux, Anil Murching, Thumpudi Naveen, and Ali Tabatabai. "Description of shape information for 2-D and 3-D objects." *Signal Processing: Image Communication* 16, no. 1 (2000): 103-122.

[68] Mademlis, A., P. Daras, D. Tzovaras, and M. G. Strintzis. "G.: 3D object retrieval based on resulting fields." In *In Eurographics 2008 Workshop on 3D Object Retrieval*. 2008.

[69] Daras, Petros, Dimitrios Zarpalas, Dimitrios Tzovaras, and Michael G. Strintzis. "Efficient 3-D model search and retrieval using generalized 3-D radon transforms." *Multimedia, IEEE Transactions on* 8, no. 1 (2006): 101-114.

[70] Ling, Haibin, and Kazunori Okada. "Diffusion distance for histogram comparison." In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, pp. 246-253. IEEE, 2006.

[71] Akgul, Ceyhun Burak, Bülent Sankur, Yücel Yemez, and Francis Schmitt. "3D model retrieval using probability density-based shape descriptors." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, no. 6 (2009): 1117-1133.

[72] Gal, Ran, Ariel Shamir, and Daniel Cohen-Or. "Pose-oblivious shape signature."*Visualization and Computer Graphics, IEEE Transactions on* 13.2 (2007): 261-271.

[73] Pan, Xiang, Qian You, Zhi Liu, and Qi Hua Chen. "3D shape retrieval by Poisson histogram." *Pattern Recognition Letters* 32, no. 6 (2011): 787-794.

[74] Reeb G.: "On the singular points of a Platform completely integrated or digital function." Sessions of the Academy of Sciences 222. 1946.

[75] Tung, Tony, and Francis Schmitt. "Augmented reeb graphs for content-based retrieval of 3d mesh models." In *Shape Modeling Applications, 2004. Proceedings*, pp. 157-166. IEEE, 2004.

[76] Biasotti, Silvia, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. "Size functions for 3D shape retrieval." In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pp. 239-242. Eurographics Association, 2006.

[77] Biasotti, Silvia, Simone Marini, Michela Spagnuolo, and Bianca Falcidieno. "Sub-part correspondence by structural descriptors of 3D shapes." *Computer-Aided Design* 38, no. 9 (2006): 1002-1019.

[78] Barra, Vincent, and Silvia Biasotti. "3D shape retrieval using Kernels on Extended Reeb Graphs." Pattern Recognition 46.11 (2013): 2985-2999.

[79] Zhang, Juan, Kaleem Siddiqi, Diego Macrini, Ali Shokoufandeh, and Sven Dickinson. "Retrieving articulated 3-d models using medial surfaces and their graph spectra." In *Energy minimization methods in computer vision and pattern recognition*, pp. 285-300. Springer Berlin Heidelberg, 2005.

[80] Sundar, Hari, Deborah Silver, Nikhil Gagvani, and S. Dickinson. "Skeleton based shape matching and retrieval." In *Shape Modeling International, 2003*, pp. 130-139. IEEE, 2003.

[81] Shokoufandeh, Ali, Diego Macrini, Sven Dickinson, Kaleem Siddiqi, and Steven W. Zucker. "Indexing hierarchical structures using graph spectra." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27, no. 7 (2005): 1125-1140.

[82] Cornea, Nicu D., M. Fatih Demirci, Deborah Silver, Ali Shokoufandeh, Sven J. Dickinson, and Paul B. Kantor. "3D object retrieval using many-to-many matching of curve skeletons." In *Shape Modeling and Applications, 2005 International Conference*, pp. 366-371. IEEE, 2005.

[83] Chuang, Jen-Hui, Chi-Hao Tsai, and Min-Chi Ko. "Skeletonisation of three-dimensional object using generalized potential field." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, no. 11 (2000): 1241-1251.

[84] Saupe, Dietmar, and Dejan V. Vranić. "*3D model retrieval with spherical harmonics and moments*." Springer Berlin Heidelberg, 2001: 392-397.

[85] Dejan V. Vrani´c and Dietmar Saupe. "3D shape descriptor based on 3d fourier transform. " In Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001), 271–274, Budapest, Hungary, 2001.

[86] Vranic, Dejan 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*, vol. 3, pp. III-757. IEEE, 2003.

[87] Dutagaci, Helin, Bülent Sankur, and Yücel Yemez. "Transform-based methods for indexing and retrieval of 3d objects." In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pp. 188-195. IEEE, 2005.

[88] Sivic, Josef, and Andrew Zisserman. "Video Google: A text retrieval approach to object matching in videos." In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1470-1477. IEEE, 2003.

[89] Elad, Michael, Ayellet Tal, and Sigal Ar. "Content based retrieval of VRML objects-an iterative and interactive approach." In *Multimedia 2001*, pp. 107-118. Springer Vienna, 2002.

[90] Ricard, Julien, David Coeurjolly, and Atilla Baskurt. "Generalizations of angular radial transform for 2D and 3D shape retrieval." Pattern Recognition Letters26.14 (2005): 2174-2186.

[91] Jones, Peter W., Mauro Maggioni, and Raanan Schul. "Manifold parametrizations by eigenfunctions of the Laplacian and heat kernels."*Proceedings of the National Academy of Sciences* 105, no. 6 (2008): 1803-1808.

[92] Sun, Jian, Maks Ovsjanikov, and Leonidas Guibas. "A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion." In *Computer Graphics Forum*, vol. 28, no. 5, pp. 1383-1392. Blackwell Publishing Ltd, 2009.

[93] Ovsjanikov, Maks, Alexander M. Bronstein, Michael M. Bronstein, and Leonidas J. Guibas. "Shape Google: a computer vision approach to invariant shape retrieval." *Proc. NORDIA* 1, no. 2 (2009).

[94] Bronstein, Michael M., and Iasonas Kokkinos. "Scale-invariant heat kernel signatures for non-rigid shape recognition." In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1704-1711. IEEE, 2010.

[95] Bronstein, Alexander M., Michael M. Bronstein, Leonidas J. Guibas, and Maks Ovsjanikov. "Shape google: Geometric words and expressions for invariant shape retrieval." *ACM Transactions on Graphics (TOG)* 30, no. 1 (2011): 1.

[96] G. Salton and M. J. McGill. "Introduction to modern information retrieval". 1983.

[97] El Khoury, Rachid, Jean-Philippe Vandeborre, and Mohamed Daoudi. "3D-model retrieval using bag-of-features based on closed curves." In *Eurographics 2013 Workshop on 3D Object Retrieval*. 2013.

[98] Khoury, Rachid El, Jean-Philippe Vandeborre, and Mohamed Daoudi. "Indexed heat curves for 3d-model retrieval." In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1964-1967. IEEE, 2012.

[99] Shi, Jianbo, and Jitendra Malik. "Normalized cuts and image segmentation."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22, no. 8 (2000): 888-905.

[100] Lévy, Bruno, Sylvain Petitjean, Nicolas Ray, and Jérome Maillot. "Least squares conformal maps for automatic texture atlas generation." In *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 362-371. ACM, 2002.

[101] Gelfand, Natasha, and Leonidas J. Guibas. "Shape segmentation using local slippage analysis." In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pp. 214-223. ACM, 2004.

[102] Shapira, Lior, Shy Shalom, Ariel Shamir, Daniel Cohen-Or, and Hao Zhang. "Contextual part analogies in 3D objects." *International Journal of Computer Vision* 89, no. 2-3 (2010): 309-326.

[103] Shamir, Ariel. "A survey on mesh segmentation techniques." In *Computer graphics forum*, vol. 27, no. 6, pp. 1539-1556. Blackwell Publishing Ltd, 2008.

[104] Fan, Lubin, Min Meng, and Ligang Liu. "Sketch-based mesh cutting: A comparative study." *Graphical Models* 74, no. 6 (2012): 292-301.

[105] Attene, Marco, Sagi Katz, Michela Mortara, Giuseppe Patané, Michela Spagnuolo, and Ayellet Tal. "Mesh segmentation-a comparative study." In*Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pp. 7-7. IEEE, 2006.

[106] Zhang, Yan, J. Paik, Andreas Koschan, Mongi A. Abidi, and David Gorsich. "Simple and efficient algorithm for part decomposition of 3-D triangulated models based on curvature analysis." In *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 3, pp. III-273. IEEE, 2002.

[107] Jagannathan, Anupama, and Eric L. Miller. "Three-dimensional surface mesh segmentation using curvedness-based region growing approach." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29, no. 12 (2007): 2195-2204.

[108] Lai, Yu-Kun, Shi-Min Hu, Ralph R. Martin, and Paul L. Rosin. "Fast mesh segmentation using random walks." In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, pp. 183-191. ACM, 2008.

[109] Zhang, Juyong, Chunlin Wu, Jianfei Cai, Jianmin Zheng, and Xue-cheng Tai. "Mesh snapping: Robust interactive mesh cutting using fast geodesic curvature flow." In *Computer Graphics Forum*, vol. 29, no. 2, pp. 517-526. Blackwell Publishing Ltd, 2010.

[110] Vieira, Miguel, and Kenji Shimada. "Surface mesh segmentation and smooth surface extraction through region growing." *Computer aided geometric design*22, no. 8 (2005): 771-792.

[111] Lavoué, Guillaume, Florent Dupont, and Atilla Baskurt. "A new CAD mesh segmentation method, based on curvature tensor analysis." *Computer-Aided Design* 37, no. 10 (2005): 975-987.

[112] Cohen-Steiner, David, and Jean-Marie Morvan. "Restricted delaunay triangulations and normal cycle." In *Proceedings of the nineteenth annual symposium on Computational geometry*, pp. 312-321. ACM, 2003.

[113] Zhou, Yinan, and Zhiyong Huang. "Decomposing polygon meshes by means of critical points." In *Multimedia Modelling Conference, 2004. Proceedings. 10th International*, pp. 187-195. IEEE, 2004.

[114] Razdan, AnShuman, and Myungsoo Baeн. "A Hybrid Approach to Feature Segmentation of 3-Dimensional Meshes." PhD diss., MS Thesis, Arizona State University, 2001.

[115] Zuckerberger, Emanoil, Ayellet Tal, and Shymon Shlafman. "Polyhedral surface decomposition with applications." *Computers & Graphics* 26, no. 5 (2002): 733-743.

[116] Vincent, Luc, and Pierre Soille. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations." *IEEE transactions on pattern analysis and machine intelligence* 13, no. 6 (1991): 583-598.

[117] Couprie, Camille, Leo Grady, Laurent Najman, and Hugues Talbot. "Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest." In *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 731-738. IEEE, 2009.

[118] Mangan, Alan P., and Ross T. Whitaker. "Partitioning 3D surface meshes using watershed segmentation." *Visualization and Computer Graphics, IEEE Transactions on* 5, no. 4 (1999): 308-321.

[119] Chen, Lijun, and Nicolas D. Georganas. "An efficient and robust algorithm for 3D mesh segmentation." *Multimedia Tools and Applications* 29, no. 2 (2006): 109-125.

[120] Koschan, A. F. "Perception-based 3D triangle mesh segmentation using fast marching watersheds." In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II-27. IEEE, 2003.

[121] Rossl, Christian, Leif Kobbelt, and Hans-Peter Seidel. "Extraction of feature lines on triangulated surfaces using morphological operators." In *Proceedings of the AAAI Symposium on Smart Graphics*, pp. 71-75. 2000.

[122] Benjamin, William, Andrew Wood Polk, S. V. N. Vishwanathan, and Karthik Ramani. "Heat Walk: Robust Salient Segmentation of Non-rigid Shapes." In*Computer Graphics Forum*, vol. 30, no. 7, pp. 2097-2106. Blackwell Publishing Ltd, 2011.

[123] Atmosukarto, Indriyati, Wee Kheng Leow, and Zhiyong Huang. "Feature combination and relevance feedback for 3D model retrieval." In *Multimedia Modelling Conference, 2005. MMM 2005. Proceedings of the 11th International*, pp. 334-339. IEEE, 2005.

[124] Shlafman, Shymon, Ayellet Tal, and Sagi Katz. "Metamorphosis of polyhedral surfaces using decomposition." In *Computer Graphics Forum*, vol. 21, no. 3, pp. 219-228. Blackwell Publishing, Inc, 2002.

[125] Zheng, Youyi, and Chiew-Lan Tai. "Mesh Decomposition with Cross-Boundary Brushes." In *Computer Graphics Forum*, vol. 29, no. 2, pp. 527-535. Blackwell Publishing Ltd, 2010.

[126] Ji, Zhongping, Ligang Liu, Zhonggui Chen, and Guojin Wang. "Easy mesh cutting." In *Computer Graphics Forum*, vol. 25, no. 3, pp. 283-291. Blackwell Publishing, Inc, 2006.

[127] Meng, M., Z. Ji, and L. Liu. "Sketching mesh segmentation based on feature preserving harmonic field." *Journal of Computer-Aided Design and Computer Graphics. v20 i9* (2008): 1146-1152.

[128] Fan, Lubin, and Kun Liu. "Paint mesh cutting." In *Computer Graphics Forum*, vol. 30, no. 2, pp. 603-612. Blackwell Publishing Ltd, 2011.

[129] Shapira, Lior, Ariel Shamir, and Daniel Cohen-Or. "Consistent mesh partitioning and skeletonisation using the shape diameter function." *The Visual Computer*24, no. 4 (2008): 249-259.

[130] Funkhouser, Thomas, Michael Kazhdan, Philip Shilane, Patrick Min, William Kiefer, Ayellet Tal, Szymon Rusinkiewicz, and David Dobkin. "Modeling by example." In *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 652-663. ACM, 2004.

[131] Lee, Yunjin, and Seungyong Lee. "Geometric snakes for triangular meshes." In*Computer Graphics Forum*, vol. 21, no. 3, pp. 229-238. Blackwell Publishing, Inc, 2002.

[132] Lee, Yunjin, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, and Hans-Peter Seidel. "Mesh scissoring with minima rule and part salience." *Computer Aided Geometric Design* 22, no. 5 (2005): 444-465.

[133] Wong, Kevin Chun-Ho, Tommy Yu-Hang Siu, Pheng-Ann Heng, and Hanqiu Sun. "Interactive volume cutting." In *Graphics Interface*, vol. 98, no. 6. 1998.

[134] Zöckler, Malte, Detlev Stalling, and Hans-Christian Hege. "Fast and intuitive generation of geometric shape transitions." *The Visual Computer* 16, no. 5 (2000): 241-253.

[135] Zheng, Youyi, Chiew-Lan Tai, and OK-C. Au. "Dot scissor: a single-click interface for mesh segmentation." *Visualization and Computer Graphics, IEEE Transactions on* 18, no. 8 (2012): 1304-1312.

[136] Chen, Xiaobai, Aleksey Golovinskiy, and Thomas Funkhouser. "A benchmark for 3D mesh segmentation." In *ACM Transactions on Graphics (TOG)*, vol. 28, no. 3, p. 73. ACM, 2009.

[137] Lei, Haopeng, Jianqiang Sheng, and Shujin Lin. "3D Mesh Segmentation Based on Multiclass Spectral Graph Partition." In *Digital Home (ICDH), 2012 Fourth International Conference on*, pp. 449-455. IEEE, 2012.

[138] Golovinskiy, Aleksey, and Thomas Funkhouser. "Randomized cuts for 3D mesh analysis." In *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5, p. 145. ACM, 2008.

[139] Bergamasco, Filippo, Andrea Albarelli, and Andrea Torsello. "A graph-based technique for semi-supervised segmentation of 3D surfaces." *Pattern Recognition Letters* 33, no. 15 (2012): 2057-2064.

[140] Reniers, Dennie, and Alexandru Telea. "Hierarchical part-type segmentation using voxel-based curve skeletons." *The Visual Computer* 24, no. 6 (2008): 383-395.

[141] Li, Xuetao, Tong Wing Woon, Tiow Seng Tan, and Zhiyong Huang. "Decomposing polygon meshes for interactive applications." In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pp. 35-42. ACM, 2001.

[142] Mademlis, Athanasios, Petros Daras, Apostolos Axenopoulos, Dimitrios Tzovaras, and Michael G. Strintzis. "Combining topological and geometrical features for global and partial 3-D shape retrieval." *Multimedia, IEEE Transactions on* 10, no. 5 (2008): 819-831.

[143] Siddiqi, Kaleem, Sylvain Bouix, Allen Tannenbaum, and Steven W. Zucker. "Hamilton-jacobi skeletons." *International Journal of Computer Vision* 48, no. 3 (2002): 215-231.

[144] Au, Oscar Kin-Chung, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. "Skeleton extraction by mesh contraction." In *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 44. ACM, 2008.

[145] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.*Introduction to algorithms*. Vol. 2. Cambridge: MIT press, 2001.

[146] Valette, Sébastien, Ioannis Kompatsiaris, and Michael G. Strintzis. "A polygonal mesh partitioning algorithm based on protrusion conquest for perceptual 3D shape description." In *Workshop towards Semantic Virtual Environments*, pp. 68-76. 2005.

[147] Tierny, Julien, Jean-Philippe Vandeborre, and Mohamed Daoudi. "Topology driven 3D mesh hierarchical segmentation." In *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*, pp. 215-220. IEEE, 2007.

[148] Tierny, Julien, Jean-Philippe Vandeborre, and Mohamed Daoudi. "3D mesh skeleton extraction using topological and geometrical analyses." In *14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006)*. 2006.

[149] Berretti, Stefano, Alberto Del Bimbo, and Pietro Pala. "3D Mesh decomposition using Reeb graphs." *Image and Vision Computing* 27.10 (2009): 1540-1554.

[150] Antini, Gianni, Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. "3D mesh partitioning for retrieval by parts applications." In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pp. 1210-1213. IEEE, 2005.

[151] Katz, Sagi, George Leifman, and Ayellet Tal. "Mesh segmentation using feature point and core extraction." *The Visual Computer* 21, no. 8-10 (2005): 649-658.

[152] Cox, Trevor F., and Michael AA Cox. *Multidimensional scaling*. CRC Press, 2000.

[153] Agathos, Alexander, Ioannis Pratikakis, Stavros Perantonis, and Nickolas S. Sapidis. "Protrusion-oriented 3D mesh segmentation." *The Visual Computer* 26, no. 1 (2010): 63-81.

[154] Lin, Hsueh-Yi Sean, H-YM Liao, and Ja-Chen Lin. "Visual salience-guided mesh decomposition." *Multimedia, IEEE Transactions on* 9, no. 1 (2007): 46-57.

[155] Spielman D. A. "Spectral graph theory and its applications. " In 48th Annual IEEE Symposium on Foundations of Computer Science FOCS07 (2007), pp. 29–38.

[156] Liu, Rong, and Hao Zhang. "Segmentation of 3D meshes through spectral clustering." In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pp. 298-305. IEEE, 2004.

[157] Liu, Rong, and Hao Zhang. "Mesh segmentation via spectral embedding and contour analysis." In *Computer Graphics Forum*, vol. 26, no. 3, pp. 385-394. Blackwell Publishing Ltd, 2007.

[158] Lei, Haopeng, Jianqiang Sheng, and Shujin Lin. "3D Mesh Segmentation Based on Multiclass Spectral Graph Partition." In *Digital Home (ICDH), 2012 Fourth International Conference on*, pp. 449-455. IEEE, 2012.

[159] Shilane, Philip, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. "The princeton shape benchmark." In *Shape Modeling Applications, 2004. Proceedings*, pp. 167-178. IEEE, 2004.

[160] Chahhou, Mohamed, Lahcen Moumoun, Mohamed El Far, and Taoufiq Gadi. "Segmentation of 3D meshes using p-spectral clustering." 1-1.

[161] Benhabiles, Halim, J-P. Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. "A framework for the objective evaluation of segmentation algorithms using a ground-truth of human segmented 3D-models." In *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on*, pp. 36-43. IEEE, 2009.

[162] Giorgi, Daniela, Silvia Biasotti, and Laura Paraboschi. "Shape retrieval contest 2007: Watertight models track." *SHREC competition* 8 (2007).

[163] Kalogerakis, Evangelos, Aaron Hertzmann, and Karan Singh. "Learning 3D mesh segmentation and labeling." *ACM Transactions on Graphics (TOG)* 29, no. 4 (2010): 102.

[164] Lv, Jiajun, Xinlei Chen, Jin Huang, and Hujun Bao. "Semi-supervised Mesh Segmentation and Labeling." In *Computer Graphics Forum*, vol. 31, no. 7pt2, pp. 2241-2248. Blackwell Publishing Ltd, 2012.

[165] Benhabiles, Halim, Guillaume Lavoué, Jean-Philippe Vandeborre, and Mohamed Daoudi. "Learning Boundary Edges for 3D-Mesh Segmentation." In *Computer Graphics Forum*, vol. 30, no. 8, pp. 2170-2182. Blackwell Publishing Ltd, 2011.

[166] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences* 55, no. 1 (1997): 119-139.

[167] Kim, Duck Hoon, Il Dong Yun, and Sang Uk Lee. "A new shape decomposition scheme for graph-based representation." *Pattern Recognition* 38, no. 5 (2005): 673-689.

[168] Choi, Hyeong In, Sung Woo Choi, and Hwan Pyo Moon. "Mathematical theory of medial axis transform." *pacific journal of mathematics* 181, no. 1 (1997): 57-88.

[169] Attene, Marco, Bianca Falcidieno, and Michela Spagnuolo. "Hierarchical mesh segmentation based on fitting primitives." *The Visual Computer* 22, no. 3 (2006): 181-193.

[170] Fayolle, Pierre-Alain, and Alexander Pasko. "Segmentation of discrete point clouds using an extensible set of templates." *The Visual Computer* 29, no. 5 (2013): 449-465.

[171] Wu, Kenong, and Martin D. Levine. "3D part segmentation using simulated electrical charge distributions." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19, no. 11 (1997): 1223-1235.

[172] Katz, Sagi, and Ayellet Tal. *Hierarchical mesh decomposition using fuzzy clustering and cuts*. Vol. 22, no. 3. ACM, 2003.

[173] Wu, Zizhao, Yunhai Wang, Ruyang Shou, Baoquan Chen, and Xinguo Liu. "Unsupervised co-segmentation of 3D shapes via affinity aggregation spectral clustering." *Computers & Graphics* 37, no. 6 (2013): 628-637.

[174] Hu, Ruizhen, Lubin Fan, and Ligang Liu. "Co-Segmentation of 3D Shapes via Subspace Clustering." In *Computer Graphics Forum*, vol. 31, no. 5, pp. 1703-1713. Blackwell Publishing Ltd, 2012.

[175] Golovinskiy, Aleksey, and Thomas Funkhouser. "Consistent segmentation of 3D models." *Computers & Graphics* 33, no. 3 (2009): 262-269.

[176] Xu, Kai, Honghua Li, Hao Zhang, Daniel Cohen-Or, Yueshan Xiong, and Zhi-Quan Cheng. "Style-content separation by anisotropic part scales." In *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6, p. 184. ACM, 2010.

[177] http://www.graphics.stanford.edu/data/mich/

[178] http://www.formaurbis.stanford.edu/index.html

[179] http://edge.mcs.drexel.edu/repository/frameset.html

[180] http://www.rcsb.org/pdb/home/home.do

[181] Bronstein, Alexander M., Michael M. Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer, 2008.

[182] Jayanti, Subramaniam, Yagnanarayanan Kalyanaraman, Natraj Iyer, and Karthik Ramani. "Developing an engineering shape benchmark for CAD models." *Computer-Aided Design* 38, no. 9 (2006): 939-953.

[183] Temerinac, Maja, Marco Reisert, and Hans Burkhardt. "SHREC'07-Protein Retrieval Challenge." *Proceedings of Shape Modelling International, Lyon, France* (2007).

[184] Mavridis, Lazaros, Vishwesh Venkatraman, D. W. Ritchie, N. Morikawa, Rumen Andonov, Alexandre Cornu, Noël Malod-Dognin *et al*. "Shrec-10 track: Protein models." In *3DOR: Eurographics Workshop on 3D Object Retrieval*, pp. 117-124. 2010.

[185] http://www.sculpteur.ecs.soton.ac.uk/aims.html

[186] D. Vranic, D. Saupe, J. Richter "Tools for 3D object retrieval: Karhunen–Loeve transform and spherical harmonics" Proceedings of the IEEE 2001 Workshop on Multimedia Signal Processing, pp. 293–298, 2001.

[187] Papadakis, Panagiotis, Ioannis Pratikakis, Stavros Perantonis, and Theoharis Theoharis. "Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation." *Pattern Recognition* 40, no. 9 (2007): 2437-2452.

[188] Lian, Zhouhui, Paul L. Rosin, and Xianfang Sun. "Rectilinearity of 3D meshes."*International Journal of Computer Vision* 89, no. 2-3 (2010): 130-151.

[189] en.wikipedia.org/wiki/Poisson's_equation.

[190] Shih, Jau-Ling, Chang-Hsing Lee, and Chao-Hung Chuang. "A 3D model retrieval approach based on the combination of PCA plane projections." *Journal of Information Technology and Applications* 5, no. 2 (2011): 46.

[191] Breu, Heinz, Joseph Gil, David Kirkpatrick, and Michael Werman. "Linear time Euclidean distance transform algorithms." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 17, no. 5 (1995): 529-533.

[192] Gorelick, Lena, Meirav Galun, Eitan Sharon, Ronen Basri, and Achi Brandt. "Shape representation and classification using the poisson equation." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 28, no. 12 (2006): 1991-2005.

[193] Attene, Marco, Francesco Robbiano, Michela Spagnuolo, and Bianca Falcidieno. "Characterization of 3D shape parts for semantic annotation."*Computer-Aided Design* 41, no. 10 (2009): 756-763.

[194] Barrett, Richard, Michael W. Berry, Tony F. Chan, James Demmel, June Donato, Jack Dongarra, Victor Eijkhout, Roldan Pozo, Charles Romine, and Henk Van der Vorst. *Templates for the solution of linear systems: building blocks for iterative methods*. Vol. 43. Siam, 1994.

[195] Bronstein, Alexander M., and Michael M. Bronstein. "Spatially-sensitive affine-invariant image descriptors." In *Computer Vision–ECCV 2010*, pp. 197-208. Springer Berlin Heidelberg, 2010.

[196] Grauman, Kristen, and Trevor Darrell. "Efficient image matching with distributions of local invariant features." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 627-634. IEEE, 2005.

[197] VEDALDI A., FULKERSON B.: VLFeat : "An Open and Portable Library of Computer Vision Algorithms". (2008).

[198] Elad, Asi, and Ron Kimmel. "On bending invariant signatures for surfaces."*Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25, no. 10 (2003): 1285-1295.

[199] MeshLab1.1.0, meshlab.sourceforge.net (2008).

[200] TOSCA, Non-rigid shape comparison and analysis. http://tosca.cs.technion.ac.il.

[201] Tang, Sarah, and Afzal Godil. "An evaluation of local shape descriptors for 3D shape retrieval." In *IS&T/SPIE Electronic Imaging*, pp. 82900N-82900N. International Society for Optics and Photonics, 2012.

[202] Reuter, Martin. "Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions." *International Journal of Computer Vision* 89, no. 2-3 (2010): 287-308.

[203] Sipiran, Ivan, and Benjamin Bustos. "Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes." *The Visual Computer* 27, no. 11 (2011): 963-976.

[204] Smeets, Dirk, Thomas Fabry, Jeroen Hermans, Dirk Vandermeulen, and Paul Suetens. "Isometric deformation modelling for object recognition." In *Computer Analysis of Images and Patterns*, pp. 757-765. Springer Berlin Heidelberg, 2009.

[205] Tabia, Hedi, Mohamed Daoudi, J. P. Vandeborreb, and Olivier Colot. "Deformable shape retrieval using bag-of-feature techniques." In *SPIE Electronic Imaging Symposium D*, vol. 3, p. 115. 2011.

[206] Toldo, Roberto, Umberto Castellani, and Andrea Fusiello. "Visual vocabulary signature for 3D object retrieval and partial matching." In *Proceedings of the 2nd Eurographics conference on 3D Object Retrieval*, pp. 21-28. Eurographics Association, 2009.

[207] Lavoué, Guillaume. "Combination of bag-of-words descriptors for robust partial shape retrieval." *The Visual Computer* 28, no. 9 (2012): 931-942.

[208] Lloyd, Stuart. "Least squares quantization in PCM." *Information Theory, IEEE Transactions on* 28, no. 2 (1982): 129-137.

[209] Tierny, Julien, Jean-Philippe Vandeborre, and Mohamed Daoudi. "Partial 3D shape retrieval by reeb pattern unfolding." In *Computer Graphics Forum*, vol. 28, no. 1, pp. 41-55. Blackwell Publishing Ltd, 2009.

[210] Agathos, Alexander, Ioannis Pratikakis, Panagiotis Papadakis, Stavros Perantonis, Philip Azariadis, and Nickolas S. Sapidis. "3D articulated object retrieval using a graph-based representation." The Visual Computer 26, no. 10 (2010): 1301-1319.

[211] Lien, Jyh-Ming, John Keyser, and Nancy M. Amato. "Simultaneous shape decomposition and skeletonization." In *Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pp. 219-228. ACM, 2006.

[212] J.D. Jackson, "Classical Electrodynamics. " New York: Wiley, 1975.

[213] Lee, Yunjin, Seungyong Lee, Ariel Shamir, Daniel Cohen-Or, and H-P. Seidel. "Intelligent mesh scissoring using 3d snakes." In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on*, pp. 279-287. IEEE, 2004.

[214] V. Guillemin and A. Pollack, Differential Topology. Englewood Cliffs, N.J.: Prentice Hall, 1974.

[215] Lavoué, Guillaume, Jean-Philippe Vandeborre, Halim Benhabiles, Mohamed Daoudi, Kai Huebner, Michela Mortara, and Michela Spagnuolo. "SHREC'12 Track: 3D mesh segmentation." In *Proceedings of the 5th Eurographics conference on 3D Object Retrieval*, pp. 93-99. Eurographics Association, 2012.

[216] Huang, Qian, and Byron Dom. "Quantitative methods of evaluating image segmentation." In *Image Processing, 1995. Proceedings., International Conference on*, vol. 3, pp. 53-56. IEEE, 1995.

[217] Rand, William M. "Objective criteria for the evaluation of clustering methods."*Journal of the American Statistical association* 66, no. 336 (1971): 846-850.

[218] Martin, David, Charless Fowlkes, Doron Tal, and Jitendra Malik. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 416-423. IEEE, 2001.

[219] Benhabiles, Halim, Jean-Philippe Vandeborre, Guillaume Lavoué, and Mohamed Daoudi. "A comparative study of existing metrics for 3D-mesh segmentation evaluation." *The Visual Computer* 26, no. 12 (2010): 1451-1466.

[220] Huebner, Kai. "BADGr—A toolbox for box-based approximation, decomposition and GRasping." *Robotics and Autonomous Systems* 60, no. 3 (2012): 367-376.

[221] Mortara, Michela, Giuseppe Patané, Michela Spagnuolo, Bianca Falcidieno, and Jarek Rossignac. "Plumber: a method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies." In *Proceedings of the ninth ACM symposium on Solid modeling and applications*, pp. 339-344. Eurographics Association, 2004.