An Isometry-invariant Spectral Approach for Macro-molecular Docking

by

De Youngster, Dela

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In partial fulfillment of the requirements

For the M.Sc. degree in

Computer Science

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Abstract

Proteins and the formation of large protein complexes are essential parts of living organisms. Proteins are present in all aspects of life processes, performing a multitude of various functions ranging from being structural components of cells, to facilitating the passage of certain molecules between various regions of cells. The 'protein docking problem' refers to the computational method of predicting the appropriate matching pair of a protein (*receptor*) with respect to another protein (*ligand*), when attempting to bind to one another to form a stable *complex*.

Research shows that matching the three-dimensional (3D) geometric structures of candidate proteins plays a key role in determining a so-called docking pair, which is one of the key aspects of the Computer Aided Drug Design process. However, the active sites which are responsible for binding do not always present a rigid-body shape matching problem. Rather, they may undergo sufficient deformation when docking occurs, which complicates the problem of finding a match.

To address this issue, we present an isometry-invariant and topologically robust partial shape matching method for finding complementary protein binding sites, which we call the *ProtoDock algorithm*. The ProtoDock algorithm comes in two variations. The first version performs a partial shape complementarity matching by initially segmenting the underlying protein object mesh into smaller portions using a spectral mesh segmentation approach. The Heat Kernel Signature (HKS), the underlying basis of our shape descriptor, is subsequently computed for the obtained segments. A final descriptor vector is constructed from the Heat Kernel Signatures and used as the basis for the segment matching. The three different descriptor methods employed are, the accepted Bag of Features (BoF) technique, and our two novel approaches, Closest Medoid Set (CMS) and Medoid Set Average (MSA).

The second variation of our ProtoDock algorithm aims to perform the partial matching by utilizing the pointwise HKS descriptors. The use of the pointwise HKS is mainly motivated by the suggestion that, at adequate times, the Heat Kernel Signature of a point on a surface sufficiently describes its neighbourhood. Hence, the HKS of a point may serve as the representative descriptor of its given region of which it forms a part. We propose three (3) sampling methods—Uniform, Random, and Segment-based Random sampling—for selecting these points for the partial matching. Random and Segment-based Random sampling both prove superior to the Uniform sampling method.

Our experimental results, run against the Protein-Protein Benchmark 4.0, demonstrate the viability of our approach, in that, it successfully returns known binding segments for known pairing proteins. Furthermore, our ProtoDock-1 algorithm still still yields good results for low resolution protein meshes. This results in even faster processing and matching times with sufficiently reduced computational requirements when obtaining the HKS.

Acknowledgements

Profound thanks to God for his favour, grace, mercies and unrelenting love. To my supervisors, Eric Paquet, Herna Viktor and Emil Petriu; my deep and unfettered appreciation for your insightful analysis, unwavering support and patient guidance. I have really grown under your care.

To all my friends, particularly Daniel Antwi, Michael Mireku Kwakye, Naki Ocran, Jennifer Bonsu, Fayzah AlShammari, Anita Darkoh, Mavis Manu, Conrad Kyei, Lorraine Sottie, Rita Teiko, and Deborah Osei-Owusu, thank you for your support and for always being there when I needed you. To my loving father, Mr Anthony Kwame De Youngster, my uncles Frank and Lawrence Akakpo, my dear sister Wendy De Youngster-Bastine, my niece Esi Yamoah and my entire family; thank you for always believing in me.

Dedication

In loving memory of Mary Ama De Youngster.

You are gone, but never forgotten. Mother, I will always love you.

Contents

List of Tables

List of Figures

Part I

INTRODUCTION

Chapter 1

Introduction

The 'protein docking problem' refers to the computational method of predicting the appropriate matching pair of a protein (*receptor*) with respect to another protein (*ligand*), when attempting to bind to one another to form a stable *complex*. Proteins and the formation of such protein complexes are an essential part of living organisms. Proteins are present in all aspects of life processes, performing a multitude of various functions ranging from, being structural components of cells, to facilitating the passage of certain molecules between various regions of cells.

Research shows that matching the three-dimensional geometric structures of candidate proteins plays a key role in determining a so-called docking pair. However, the active sites which are responsible for the binding do not always present a rigid-body shape matching problem. Rather, they may undergo sufficient deformation when docking occurs, which complicates the problem of finding a match.

To address this issue, we present an isometry-invariant and topologically robust partial shape descriptor method for finding complementary protein binding sites, which we call the *ProtoDock algorithm*. Our method employs the *Heat Kernel Signature* shape descriptor which is based on the diffusion of heat on surfaces. The Heat Kernel Signa-

ture has been shown to be highly informative and intrinsic, thereby requiring no external reference frame for defining its descriptor. We also propose two novel compact descriptor vector methods known as the *Closest Medoid Set (CMS)* and the *Medoid Set Average (MSA)*, both of which show promising shape matching performance for rigid and non-rigid shapes.

1.1 Motivation

Proteins play a major role in several life processes. Assemblies of aggregated protein can do mechanical work such as muscle contraction and relaxation. Proteins also act as catalysts which accelerate other biological processes. They further form the structural components of cells and organelles, and serve as regulators of biochemical activities in certain cells. Likewise, interactions between proteins yield useful benefits to living organisms [54]. As a result, the study of their structure and function—*proteomics*—presents more insights to the benefits accrued from their functions in living biological systems.

Two main categories of methods exist in addressing the protein docking problem, namely, experimental methods and computational methods. Experimental methods employ the use of in-vitro laboratory experimentation with biological samples to find the docking sites. Although comprehensive and ultimately conclusive, such methods require massive financial requirements to acquire both the expert personnel and the equipment [85].

Computational methods, on the contrary, address the problems using core biological principles and mathematical theories implemented as computer software applications. Two categories of computational approaches exist, that is, ones that solve the docking as a low energy optimization problem, and geometric matching approaches which aim to

find complementary sections of two prospective protein structures [68, 85].

Approaches that use the low energy optimization problem, though often more exhaustive, are computationally very costly as a result of the combinatorially high recursive comparisons of a ligand over a target receptor protein. Another key drawback to the energy optimization approach stems from the fact that, given a known conformation for a protein pair, once any one of the pairing proteins undergoes considerable structural transformation, the entire recursive comparison has to be recalculated. That is, this approach lacks a simple method of quickly re-evaluating modified protein structures.

Geometric methods, on the other hand, address the problem by using the geometric information embedded in the structural representation of the proteins. Although less conclusive in its results, as compared to the optimization methods, geometric methods provide a means of describing the geometry of the proteins using compact descriptors, and therefore enabling fast comparisons between target proteins. These approaches consequently serve as suitable pruning techniques to the more exhaustive low energy optimization methods [85]. The relatively low computational cost of the geometric methods may further be reduced by considering smaller portions of the large marco-molecules at a given time. Reducing the resource cost necessary for analysing protein-protein interactions allows for researchers and students with limited computing hardware to be able to perform such experimentations for the purposes of, for instance, Computer Aided Drug Design (CADD). Furthermore, this enables faster evaluation of the more comprehensive protein docking analysis methods by facilitating quicker pruning phases, and therefore quicker selections of possible protein docking pairs.

1.2 Thesis Goals

To address the protein docking problem as a geometric matching problem, this thesis proposes the ProtoDock algorithm, which is an isometry-invariant and topologically robust deformable shape descriptor method for finding shape matches. The descriptor utilized here employs the Heat Kernel Signature, which in itself is based on the propagation of heat over an object's surface. The Heat Kernel Signature has been shown to be highly informative, thus sufficiently capturing the geometry of a given surface. Also, due to its invariance to isometric changes, it enables the description of deformable shapes.

The contributions of this thesis are as follows;

1. We first explore and evaluate spectral approaches to mesh object segmentations. Spectral methods have been shown to "understand" geometry, and therefore often produce salient segments on a given three-dimensional mesh object. For such techniques, the graph representation of an underlying object mesh forms the basis for the segmentation process. We investigate the quality and properties of the segmentations induced by such spectral partitioning process.

2. We present an analysis of our proposed ProtoDock-1 algorithm. Our approach performs a partial deformable shape matching by first partitioning the mesh into salient segments, obtaining appropriate descriptors for each segment, and then finally comparing the descriptors of these segments to find possible matches for docking. We also duly present a formalization of our algorithm. We additionally evaluate and discuss the viability of our algorithm in addressing the protein docking problem.

3. We also present an alternate ProtoDock-2 algorithm, which is likewise based on the Heat Kernel Signature. However, unlike the ProtoDock-1 algorithm, this ap-

proach does not undergo any partitioning but attempts a partial shape matching by performing a point-wise comparison of sampled points over the possible target object meshes. For the purpose of the sampling phase, we propose three sampling techniques, the last of which presents an intuitive and isometrically robust method for consistently selecting the points on specified segments in the mesh.

4. Resulting from our ProtoDock-1 algorithm, we present two (2) novel compact descriptor methods for describing shapes, namely, the Medoid Set Average (MSA) and Closest Medoid Set (CMS) techniques. These two methods utilize the representative Heat Kernel Signature vectors—(the *medoids*)—obtained after a K-Means clustering on the Heat Kernel Signature of each point on the Surface. The MSA method obtains a final compact descriptor vector by finding the column-wise average of the medoid set. The CMS method, on the other hand, finds the summation of the squared average of the closest distances between each pair of medoids in two prospective object medoid sets. We further evaluate the performance of the our descriptors when compared with the established Bag of Features (BoF) method which first performs a soft vector quantization, and then finds the feature distribution of all the points over three-dimensional mesh with respect to the prior obtained medoids.

1.3 Thesis Outline

This thesis is organized into nine (9) chapters. Chapter 2 presents an introduction to proteins, protein-protein interaction, the protein docking problem and a review of the literature on existing methods that address the problem. Chapter 3 discusses the concept of mesh segmentation and provides a review of the current methods employed to this

purpose. Special attention is paid to Region growing, Graph and Spectral methods. Chapter 4 presents a discussion on the Laplace-Beltrami operator and its properties. Further in-depth discussion on the various proposed discrete laplacians is given in the latter sections of the chapter. Chapter 5 presents the last of the review chapters on three-dimensional mesh descriptors. Several proposed descriptors are explained, with a detailed review of the Heat Kernel Signature given at the final sections.

Chapter 6 presents the core of our contribution in the form of a detailed analysis of our proposed ProtoDock algorithms. Our algorithm is presented in phases, with an overview of each phase duly analysed. The chapter concludes by providing a formalization of each of the ProtoDock algorithms. Chapter 7 reviews our procedures, datasets, software and hardware platforms adopted for the implementation of our experimentation. Chapter 8 introduces the criteria adopted for the evaluation of our experiments. Analysis of the results is also presented in the chapter. The final Chapter 9 presents the concluding remarks, our contributions and a short overview of our future work.

Part II

BACKGROUND AND LITERATURE REVIEW

Chapter 2

Proteins

2.1 Elements of Life

Cells form the basic building blocks of all living organism. They are part of the 'hierarchical organization of life' [54] which may be given as: *atom* $\rightarrow$ *molecules* $\rightarrow$ *macromolecules* $\rightarrow$ *organelles* $\rightarrow$ *cells* $\rightarrow$ *tissues* $\rightarrow$ *organs* $\rightarrow$ *whole organism*. Atoms are at the root of this hierarchy with carbon, hydrogen, nitrogen, oxygen, phosphorus and sulphur contributing to more than 97% of the weight of most organisms [54]. These six elements are popularly abbreviated as CHNOPS. The progressive formation of the different levels of the hierarchy is achieved through different forms of interactions. From the atom level through to the tissues level, most interactions that occur are chemical reactions which often involve chemical bonds, or parts of molecules called *functional groups* (which serve as the active sites for molecular interactions). One or more atoms may interact through bonds or linkages such as, *covalent bonds or Hydrogen bonds*, to form a molecule. Several molecules may then subsequently be involved in linkages with similar bonds to form macromolecules. Most macromolecules are organic *polymers*.

Polymers are usually formed from joining many smaller molecules, or *monomers*,

through the removal of the elements of water (a process called *condensation*). In certain cases, polymers may be formed from the repetition of the same monomer, for instance, as is found in certain carbohydrates. In other cases, such as in proteins and nucleic acids, several different monomers may be aggregated in a particular order to form the polymer [54]. An example of a macromolecule is starch. It is a polymer of the sugar glucose. This example also brings to light a key principle in the hierarchical organization of life, in that, properties and characteristics of each level cannot be directly inferred from the properties of its preceding level. Using the glucose $\rightarrow$ starch polymer as an example, while glucose is sweet and soluble in water, the starch polymer is not [54].

The sequence of aggregation through bonds and linkage continues up the hierarchical organization of life. However, at the cells level and above, the definition of each level is determined less by the presence of actual chemical bonds, but rather mainly by their form and collaborative function. For example, several cells of similar structure, and performing similar functions make up an organelle.

2.2 Proteins

Proteins are found at the macromolecule level in the hierarchical organization of life. They are essential to the sustenance of living organisms. They are present in all aspects of life processes performing a multitude of different roles ranging from functioning as enzymes (biochemical catalysts) which rapidly accelerate biological reactions, to being structural components of cells and organisms which provide support and also facilitate the passage of certain molecules between different regions of cells [54, 86].

Assemblies of aggregated proteins can do mechanical work such as muscle contraction [86], while some other proteins also play a role in the transfer of nutrients and information in living cells [15]. However, key protein functions of major interest are their ability

to, regulate biochemical activities in target cells, serve as receptors for hormones and various ligands, and act as modifiers to cell-cell interactions [54]. These are of principal significance because they are essential in the drug design process for finding cures to diseases and ailments such as designing antibodies that defend against infections [54].

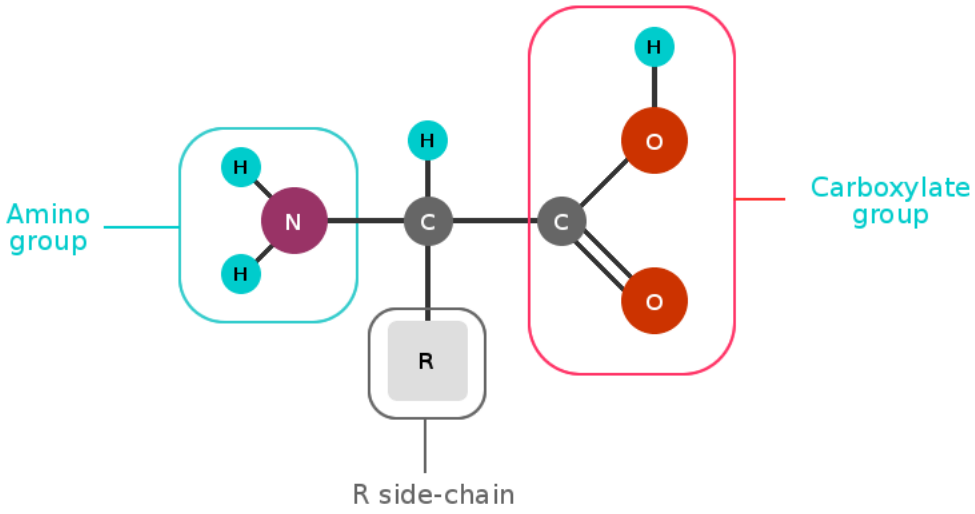


Figure 2.1: Structure of an amino acid depicting the amino group (left), carboxylate group (right) and the R side-chain (bottom)

Proteins are primarily made up of chains of amino acids. Amino acids have three major parts— *amino group, carboxylate group,* and a *side alkyl chain (R group)* which is the prime determinant of the classification of the protein. Figure 2.1 shows the typical structure of an amino acid with the generic formula given as $H_2NCHRCOOH$. The amino acid chains are formed during protein synthesis where a condensation occurs between the amino group of one amino acid, and the the carboxylate group of another amino acid. This results in an *amide linkage* or a *peptide bond* forming between the carbon atom and the nitrogen atom of the two constituent amino acid residues. The resulting chaining of

such amino acids is called a *polypeptide*, with lengths ranging from 30 to over 30,000 amino acid residues [15, 54]. A single polypeptide may serve as a functional protein. However, most proteins exist as complex structures formed from the sequential chaining of several distinct tightly bound polypeptides which often fold into distinctive three-dimensional (3D) shapes or *conformations* [54]. The unique conformations formed are mainly driven by the protein's reaction to water. In that, the main goal of the protein is to pack the *hydrophobic* (water-fearing) residues into the non aqueous interior of the molecule in order to avoid contact with the water, while *hydrophilic* (water-loving) residues usually remain on the solvent exposed surface [15, 54].

2.3 Protein Structure and Function

The three-dimensional protein shapes formed during protein synthesis underlies an extremely key principle of biological science: *structure determines function*. In that, the three-dimensional shape of any protein is the core determining factor of what function the protein will perform. As an example, it is now known that many enzymes contain a cleft (or groove) which serves as the active site of the enzyme that binds the substrates of a chemical reaction [54].

The three-dimensional structures of proteins (and polymers as whole) may be represented in several forms *e.g.*,, backbone, ball and stick, van der Waal, Surface models *etc.*. Figure 2.2 gives an illustration of a few of the different representations of an example protein proteinase inhibitor SSI (3SSI) [81, 98].

As of this writing, there are several thousands of experimentally determined protein structures. The RCSB Protein Data Bank records 88,714 structures in its database [81]. ExPasys UniProtKB/Swiss-Prot contains 539,616 sequence entries, comprising 191,569,459 amino acids abstracted from 217,393 references [28, 29, 91]. The rapid in-

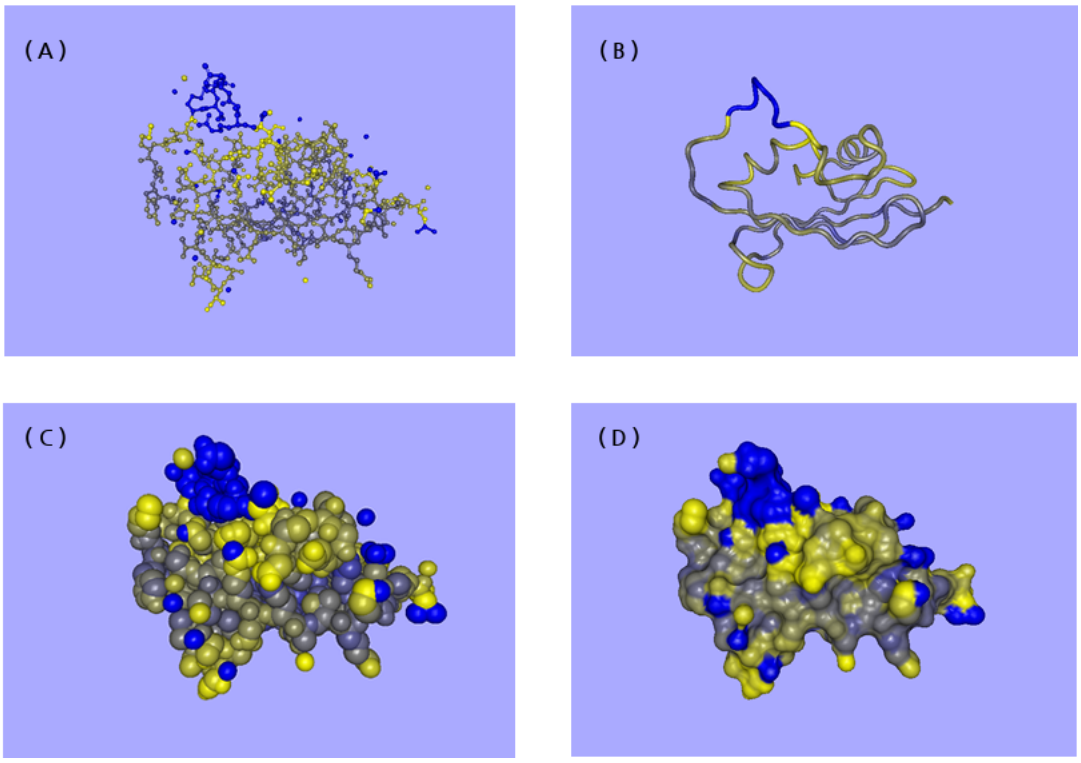


Figure 2.2: Different structural representations (models) of the protein proteinase inhibitor SSI (3SSI). (A) Ball and Stick model (B) Backbone (C) Van Der Waal (VDW) model (D) Solvent-Excluded Surface (SES).

crease is due ot the varied means of structure determination including methods such as, X-ray crystallography and Nuclear Magnetic Resonance (NMR) spectroscopy [75]. There are a vast number of experimentally determined sequence entries now known and continually being added to database [88].

As noted prior, structure determines function, and this has been experimentally confirmed by showing that, proteins of similar structure will perform similar functions [88]. To this effect, there has been significant research into protein structure prediction employing various approaches ranging from inference from statistical information from known structures [84], to the relatively more accurate template or homology modelling approach which predicts the structure by comparing it to a similar sequence [53, 121, 122].

Likewise, much research has also been undertaken in the field of protein function prediction based on the three-dimensional structural similarities to proteins whose function and structure are known. Such methods vary in their approaches. Akbar *et al* [2], Cui *et al* [22], and Yeh *et al* [115] use structure alignment as their main technique for comparison. Paquet *et al* [75] use a combination of both two-dimensional and scale, rotation and translation invariant three-dimensional signatures of the protein structures for comparisons. While other works, such as Chi *et al* [36], Huang *et al* [41], and Park *et al* [76], use local approaches, some of which employ two-dimensional distance matrices as representations of the topological structure of the proteins.

2.4 Protein-Protein Interaction

Another area of particular research interest in proteomics (the large-scale study of proteins, with specific concentration on their structures and functions [54, 111]) is the study of the interactions between proteins. This interaction between proteins is often referred to as *protein-protein interactions* and spans studies carried out in biological, biochemical and biophysical fields. Protein-protein interactions are present across many aspects of cell function, from being the transport mechanism for various biological membranes to the regulation of gene expression [88].

One specific form of protein-protein interaction which is under active research is ***protein-protein docking***, also known as *protein-ligand docking*, or generically called *the protein docking problem*. The protein docking problem refers to the method of predicting the appropriate alignment or orientation of a protein molecule (*receptor, host* or *lock*) with respect to another protein (*ligand, guest* or *key*) when bound to each other to form a stable *complex* or *dimer* [101, 110]. Finding two binding proteins is a key step in identifying prospective drug candidates during the process of drug design [3, 110].

The problem of finding appropriate matches is not only made severe by the complex structure of the proteins and the several thousands of protein sequence entries to be considered, but also by the several degrees of freedom in terms of position and orientation which must be accounted for. Figure 2.3 shows a simplified cartoon illustration of protein-protein docking.

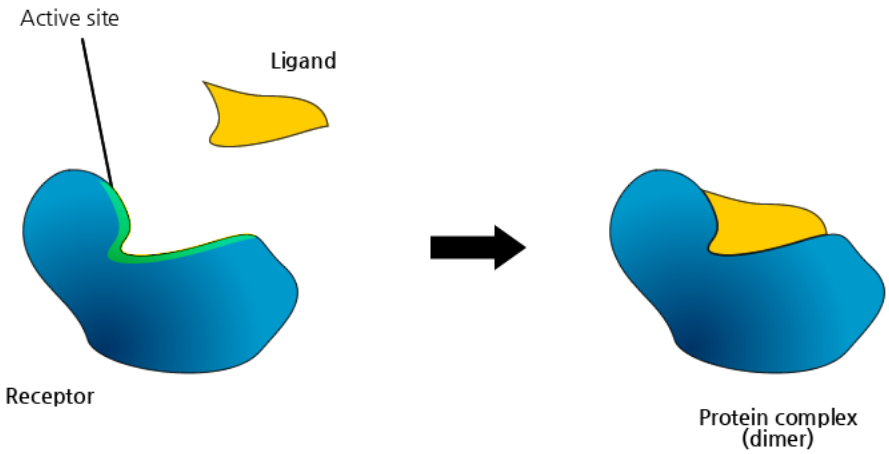


Figure 2.3: A simplified cartoon illustration of a ligand (yellow) docking to a receptor protein (blue) to form a stable complex or dimer.

Several methods have been researched for addressing the protein docking problem. These methods fall under two main areas, namely, *biological* and *computational*. Biological methods mainly deal with *in vitro* (laboratory) experimentations. Though it provides definitive results on docking pairs, due to the large and ever increasing domain of protein sequences being discovered, such methods require very expensive laboratory equipment, time and human resources. Computational methods, on the other hand, seek to abate such costly necessities by using the underlying biological principles, mathematical theories and computing applications to address the protein docking problem.

Computational methods may be placed under two broad categories, *matching methods*

and docking *simulation methods* [85]. Matching methods aim to dock a target inhibitor structure into a created model of the receptor active site, by comparing that with the inhibitors structural geometry. The docking methods, on the other hand, attempt to model the docking process by randomly exploring different translations, orientations and conformations of a target ligand to a receptor protein, with the goal of finding an ideal site [68, 85]. Several of these computational methods have been implemented as stand-alone computer simulation and prediction application software with varying levels of success and resource requirements.

2.4.1 Existing Works

Morris *et al*

One such popular matching-based computer application is Autodock, which is a result of the work of Morris *et al* [68]. The authors use three stochastic search methods— Monte Carlo simulated annealing and traditional genetic algorithm— for predicting the bound conformations. Genetic algorithms employ ideas based on principles of natural genetics and biological evolutions. They sufficiently address problems that suffer from combinatorial explosions arising from the presence of many degrees of freedom [68]. Later versions of AutoDock incorporate the Lamarckian genetic algorithm which is shown to be the most efficient of the three optimization methods and also better handles ligands with more degrees of freedom.

Atilgan *et al*

Atilgan *et al* [3] introduced AutoDockX which is based on the work of [68] but seeks to address the deficiencies arising from local optima and premature convergence issues in the simulated annealing and the traditional genetic algorithms, respectively. The authors

note that these issues are usually offset by conducting multiple runs in order to obtain reasonable results, but end up being computationally more expensive and consequently time consuming. The authors therefore present the Age-Layered Population Structure (ALPS) algorithm— a sustainable genetic algorithm— to address the shortcomings of the core algorithms implemented in AutoDock. The ALPS algorithm takes its roots from the Hierarchical Fair Competition (HFC) Framework introduced by Hu *et al* [40]. Atilgan *et al* [3] report improved performances in running times and robustness over the traditional genetic search methods used in [68].

Hashmi and Shehu

Hashmi and Shehu [37] present another matching-based stochastic search method for addressing the protein docking problem. Their approach attempts to solve the problem by finding an optimal *Ranking*. Ranking is the process of classifying ligands according to the likelihood of interacting favourably with a given receptor based on the predicted free-energy of binding. Their method employs a basin hopping algorithm which repeatedly evaluates the energy minimization of structural perturbations to obtain approximate estimates of the energy surface in terms of local minima [37]. The authors propose that increasing the surface area for the free-energy sampling produces the prefered docking configurations with great accuracy, with many energy minima close to the regions of native configurations. Like many optimization approaches, this method is computationally expensive, is subject to local minima issues like the earlier version of Autodock [68], and is also highly dependent on the sampling range selected.

Paschalidis *et al*

Paschalidis *et al* [77] likewise explore the free-energy approach by exploiting all 6-dimensional degrees of freedom in terms of translations and rotations of a ligand with respect to a given receptor. A global optimization method called Semi-Definite Underestimation (SDU) is employed in the continual series of translational and oriental changes between a given ligand and a target receptor. The total energy change over several iterations of the SDU optimization algorithm shows sufficient incremental reductions in the free-energy levels at each step. Paschalidis *et al* [77] show improved performances in energy evaluations over traditional Monte Carlo optimization.

Axenopoulos *et al*

Other approaches, such as that proposed by Axenopoulos *et al* [5] adopt a geometry-based shape matching technique, instead of the optimization problem of finding paired regions of the receptor and ligand which minimize the free-energy of binding. Axenopoulos *et al* [5] propose the use of a Shape Impact Descriptor (SID). This is a rotation-invariant 3D shape descriptor which alleviates the need for repeated searches for initial alignment. The removal of this requirement reduces the computational needs as compared to the relatively more exhaustive search methods such as [37, 77]. However, given that the structures of ligands do not remain fixed, but may bend or contort to bind to a target receptor, this approach remains relatively deficient as it is only suitable for rigid body comparisons. Also, the work of Kahraman *et al* [45] suggest that on the geometrical complementarity basis of shape and size matching, shape is the major contributing factor to matching two prospective active sites. Nonetheless size still plays an adequate role. Hence, another drawback to this method arises from the fact that, matching complementary descriptors are only obtained for shapes of the same sizes.

Complementary Works

Other works, such as Tong and Zhiping [102] propose methods for refining and improving only the results of other algorithms predictions. Work by Lo [61], on the other hand, also uses existing algorithms but modifies the computing platform in order to utilize recent improvements in computing technologies such as, improved parallel graphics processing and leveraging the benefits of modern off-site cloud computing facilities.

While promising results have been achieved by the above mentioned methods, a number of issues still remain, particularly in terms of their computational complexity. While the optimization methods produce relatively more definitive docking pair predictions, they come at significantly high computational costs. Some of the geometric matching methods also suffer from certain preprocessing requirements such as, scale equivalence, mesh uniformity, pose normalization, amongst others. Our work, as presented in this thesis, aims to address these shortcomings.

2.5 Summary

Proteins play a vital role in living organism and, most especially, in the creation of medical drugs. Studies into the prediction of their three-dimensional structures from their constituent amino-acid chain, their biological function, and also in the location of their active sites which serve as the binding regions for protein-protein interactions is an active area of research. Several methods that attempt to solve the protein docking problem have been proposed and implemented in various computer applications. These methods range from stochastic free-energy optimization methods, to methods of geometric complementarity through shape matching. Existing approaches have been shown to address the docking problem with varying results of success and computational requirements.

For example, the optimization methods are more accurate but computationally more expensive, while the matching methods are less precise but notably faster. Our work, which focuses on finding accurate matches, addresses this need.

In the next chapter, we discuss object segmentation techniques, which we employ during our protein-protein matching process.

Chapter 3

Object Segmentation

The goal of object segmentation is to divide a given object into '*meaningful*' smaller parts or segments. Such an object may be a digital image or a three-dimensional (3D) object. Segmenting the large protein meshes into smaller portions aids in reducing the computational requirements when predicting possible protein docking pairs. Methods used in 3D object segmentations borrow and extend quite a number of their techniques from the well researched and mature field of two-dimensional (2D) image segmentation methods. We discuss several works employed in the literature which fall under the three major categories of, *region growing*, *clustering*, and *spectral based methods*. We then discuss the two major works carried out on the formalization of 3D mesh segmentation evaluations. We finally give a summary of this discussion.

3.1 Object Segmentation

Object segmentation is the process of dividing or partitioning a given object into non-overlapping or disjoint components [1, 63]. Usually, the segmentation is based on a specific criteria *e.g.*, equal or ranged color equivalence in 2D Image segmentation—regions

of equal color or colors within a set spectrum are returned as a single segment. Likewise, three dimensional (3D) segmentation is performed on a mesh representation of 3D objects. Here, a segmentation process attempts to partition the mesh into disjoint regions of connected components (set of vertices or faces), of either geometric or semantic relevance [8]. Three-dimensional object meshes may be obtained from scanning an actual physical object using a laser, and connecting the corresponding 3D points which represent the surface of the object [1]. These object meshes may also be computer generated from virtual models created using softwares such as, Computer Aided Design (CAD) tools.

The aim of of most mesh segmentation algorithms is to try to segment the mesh into '*meaningful*' parts, which have an underlying representation of the structure of the object. This semantic meaning, including the relative size and organization, of the parts or segments is often dependent on the domain of application, and varies as such [63]. One such popular semantic criteria for 3D mesh segmentation is based on the minima rule [94]. The minima rule posits that, in the process of visual recognition, the human visual system divides 3D shapes into parts at negative minima of curvature [39, 63]. Simply put, humans perceive parts of an object at areas of concavity. The minima rule itself is also primarily motivated by the principle of *transversality*. This states that: two generic 3D shapes intersect almost surely in a concave crease [39]. This criteria for determining semantic parts is often adopted as the main aim for most algorithms in addressing the mesh segmentation problem [1, 12]. Figure 3.1 demonstrates the principle of transversality as a natural human perception of object segmentation. Figure 3.2 shows a set of 3D segmentations based on the minima rule.

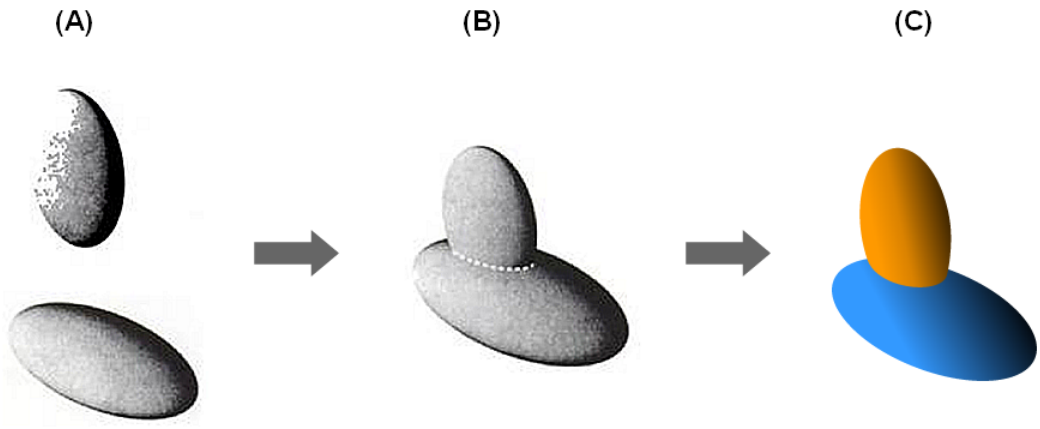


Figure 3.1: The principle of transversality. (A) shows two ellipsoidal objects. (B) depicts the perceived line of concavity formed from joining the two objects. (C) shows the perceived segments obtained along the line of concavity from the two merged objects. *Adapted from [94].*

3.2 Properties of Mesh Segmentation Methods

Much research has been carried out in the field of three-dimensional mesh segmentation into functional parts, not only because it gives semantic information about the base structure of which they form a part, but also serves as a helping process for several types of mesh processing algorithms *e.g.*, texture mapping [57], shape-based retrieval [125], morphing [35, 124], skeleton extraction [11, 48], amongst others. Works in mesh segmentation have employed a varied number of techniques including methods based on, graph cuts, hierarchical clustering, primitive fitting, spectral clustering, and random walks [20]. Agathos *et al* [1] propose two (2) groups under which these methods can be classified—*surface-based (geometric)* and *part-based (semantic)*. In surface-based methods, the mesh is segmented into distinct representative parts which are based on surface geometric information such as curvature and planarity [89]. Part-based methods, on the

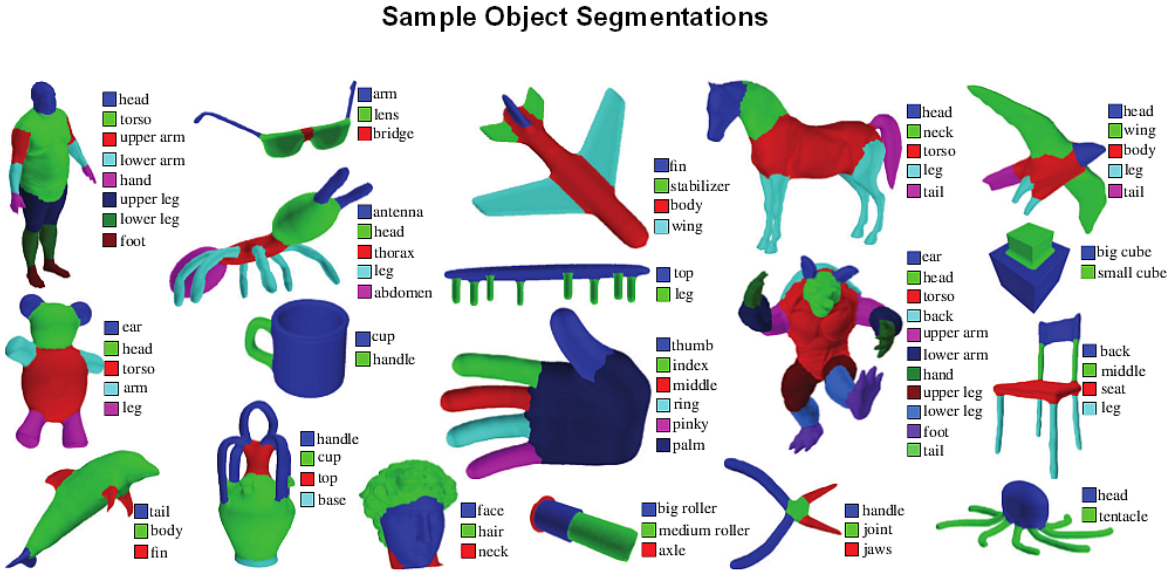


Figure 3.2: Examples of three-dimensional object segmentations based on the minima rule [46]

other hand, segment the mesh into volumetric parts corresponding to relevant features of the object which can be approximated by volumetric primitives (cones, spheres *etc.*) [1, 8].

The quality of the segmentations produced by these methods are measured by different factors and are also domain dependent. Several works, including Attene *et al* [4] and Agathos *et al* [1], present user-based evaluation metrics where the readers are encouraged to do a side-by-side analysis of images of the segmented 3D meshes, with each segment colour-coded for easy visualization. The metrics included, boundary smoothness, segmentation type, suitability of inputs, sensitivity to pose, computational complexity, and control parameters. Agathos *et al* [1] likewise propose a set of similarly domain-subjective segmentation quality metrics, particularly, ones useful in Computer Aided Design (CAD) software tools. Their measures of quality are, smoothness of segment boundaries, ability to approximate segmented regions by smooth surfaces, ability of shared region bound-

aries to allow certain types of continuity for the approximating surfaces, and closeness of segments to ones based on the minima rule. However, recent works by Benhabiles *et al* [8] and Chen *et al* [20] present a relatively more formal and objective set of segmentation evaluation metrics based on a ground-truth corpus. The ground-truth corpus of segmentations were obtained by surveying the results of manually segmented 3D surface meshes made by humans (eighty people in total).

Chen *et al* [20] define four evaluation metrics, namely, *Cut Discrepancy, Hamming Distance, Rand Index,* and *Consistency Error.*

Cut Discrepancy is a boundary-based method which measures the distances between cuts [8]. It achieves this by summing the distances from points along the segments of an obtained segmentation to the closest segments found in the ground-truth segmentation [20]. One noted merit of this metric is that it gives a simple intuitive measure of how close the alignments are between obtained and ground-truth segmentations. However, a weakness of the Cut Discrepancy is its sensitivity to the granularity of segmentation. This means that the Cut Discrepancy metric decreases to zero as more segments are included in the ground truth segmentation, and also becomes undefined when neither of the 3D meshes being compared to has no segments at all [20].

Hamming Distance, also employed in image segmentation metrics, measures the overall differences in the segment regions between two segmentation results [8, 20]. Given two mesh segmentations $S_1 = \{S_1^1, S_1^2 ... S_1^m\}$ *(computed segmentation)* and $S_2 = \{S_2^1, S_2^2 ... S_2^n\}$ *(ground-truth segmentation)* with m and n segments respectively, the core idea is to obtain and sum up the set difference of the closest equivalent segment in S_1 for each segment in S_2. Given that this metric finds the correlation between segments, it has the main advantage of giving a more meaningful evaluation criteria when corre-

spondences between a given pair of mesh models are *'correct'*. On the other hand, this same metric yields an undesirable effect of adding noise to the evaluation metric when the correspondences are *'not correct'*. It is also moderately sensitive to variations in segmentation granularity [20].

Rand Index is the third metric employed by [20]. This measure was originally introduced by W. Rand [80] as a metric for evaluating the performance of clustering methods. The Rand Index measures the likelihood that a given pair of equivalent mesh faces either belong to the same segment, or are in separate segments in two segmentations. Therefore, the Rand Index of two segmentations gives the proportion of pairs of mesh faces that jointly 'agree or disagree' on their segment group identities within the segmentations. Chen *et al* [20] note that the chief advantage of this metric is its ability in modelling segment area overlaps without the need of first obtaining the segment correspondences.

Consistency Error is based on the works of Martin et al [64]. The authors propose a region-based consistency metric which avoids penalizing variations in the hierarchical granularity. The idea stems from the theory that, the perceptual organization of humans imposes a hierarchical tree structure on observed objects [20, 64]. Although this metric is able to adequately account for the differences in the nested hierarchy of the segmentations, it however induces a weakness of tending to give better scores when two mesh models have different segmentation numbers. This situation is most evident in instances approaching extreme segmentation counts where, for example, an error of zero is given if one of the meshes only has a single segment, or conversely, where each face within the mesh belongs to a different segment.

Benhabiles *et al* [8] propose a metric called the *3D Normalized Probabilistic Rand Index (3D-NPRI)*. They build on the work of [20] by presenting a probabilistic approach

to the Rand Index metric discussed above. The core of the 3D-NPRI metric aims to address the correct variations in the granularity of segmentations that exist within the ground-truth mesh model segmentations generated by humans. They state that, a good segmentation metric should possess certain key properties including, (i) Ability to measure proportionality in segmentations when compared to the ground-truths, (ii) Ability to account for the differences in granularity of segmentation by humans, and (iii) Cardinality independence, i.e. where quality of segmentations shouldnt be irrespective of the number of segmentations between generated segments and the ground-truth, amongst others. Benhabiles *et al* [8] present results which show improved segmentation evaluation performances for the 3D-NPRI metric in terms of the aforementioned properties and discriminative power.

As suggested by several works, including Chen *et al* [20], Benhabiles *et al* [8] and Shamir [89], although there exist consistencies in the segmentations created by humans which often abide by the minima rule of lying along sections of concavity, no single segmentation algorithm suits all objects and conditions. This further emphasizes the point that, the requirements in the quality of segmentations are very much dependent on the domain of application and the specific requirements therein.

3.3 Mesh Segmentation Methods

We first present a formalization of the mesh segmentation problem in the same vein as Shamir [89]. Let a three dimensional mesh M be defined as a tuple V, E, F of,

$$vertices \quad V = \left\{ p_i \mid p_i \in \mathbb{R}^3, 1 \leq i \leq m \right\}, \tag{3.1}$$

$$edges \quad E = \{ e_{ij} = (p_i, p_j) \mid p_i, p_j \in V,\, i \neq j \}, \tag{3.2}$$

$$faces \quad F = \{ f_{ijk} = (p_i, p_j, p_k) \mid p_i, p_j, p_k \in V,\, i \neq j, i \neq k, j \neq k \}. \tag{3.3}$$

Let S be the set of mesh elements, V, E or F. The faces F are usually triangles, but could include other types of planar polygons. The definition of meshes in this discussion will be limited to a *manifold (2-manifold) watertight* mesh representation. A *2-manifold* denotes a meshing where every edge is shared by 2 triangular faces, and *watertight* refers to a mesh without any boundary edges and which hence forms a complete enclosure. Let $S' \subset S$ be a subset of the mesh elements, V' be the set of all vertices found in S'. A subsequent sub-mesh $M' \subset M$ is defined as the mesh

$$M' = \{V', E', F'\}, \tag{3.4}$$

where E', the set of all edges which have both vertices in V', is defined as,

$$E' = \{(p_i, p_j) \in E \mid p_i, p_j \in V'\} \tag{3.5}$$

and F', the set of all faces which have all vertices in V', is defined as,

$$F' = \{(p_i, p_j, p_k) \in F \mid p_i, p_j, p_k \in V'\} \tag{3.6}$$

A mesh segmentation $\mathcal{M}$, is a set of sub-meshes $\mathcal{M} = \{M_0, \ldots, M_{k-1}\}$ induced by partitioning of M into k disjoint subsets of vertices or faces. Segmentations using faces does not pose the trivial boundary element ambiguity issues as is present when segmenting meshes using vertices. This issue arises from having either one or two vertices of a given triangle (or polygon) forming part of a particular segment, while the other vertices are part of another. This is usually solved by simply including a face with conflicting vertices in one of the adjacent parts.

Several 3D mesh segmentation methods have been presented in the literature. Shamir

[89] propose one of several groupings of these methods based on their core methodology, as follows, *Region growing, Hierarchical clustering, Iterative clustering, Spectral analysis and Implicit methods*. Agathos *et al* [1] also present a slightly different but inclusive set of categories resulting in, *Region Growing, Watershed-based, Reeb graphs, Model-based, Skeleton-based, Clustering, Spectral Analysis, Explicit Boundary Extraction, Critical points-based, Multiscale Shape Descriptors, Markov Random Fields,* and *Direct Segmentation*. However, both Agathos *et al* [1] and Shamir [89] duly point to the fact that, their classifications are not necessarily strict, and could easily have overlappings, mergers and omissions of the categories due to the similarities of the core methodologies used *e.g.*, region growing and clustering both exhibit similar properties of increasing colonies.

We employ a combination of both classifications and discuss the three major classes, namely, Region growing methods *(Watershed methods inclusive)*, Clustering based methods, and Spectral methods. Shamir [89] further presents three (3) major conditions which serve as the main guide for most mesh segmentation objectives, *cardinality constraints, geometric constraints*, and *topological constraints*. Cardinality deals mainly with the proportion of elements (vertices or faces) in each segment of a given mesh model. Such constraints include, the maximum and minimum number of elements in each segment, in all segments, and the total number of segments induced (where applicable). Geometric constraints imposed on sub-meshes include, maximum and minimum area covered by the segment, diameter of the segments mesh, closeness to specified volumetric shapes and concavity or convexity of segments with respect to the underlying mesh model. Then finally, topological constraints may be a restriction where each segment obtained must form a single connected component (graph).

3.3.1 Region Growing methods

The Region growing technique is a local-greedy approach where segmentations are generated from initial points of growth. These points are termed *seed elements* (or *seeds*), and could be either points or faces [1] belonging to the set S of elements which grow incrementally as a sub-mesh, adding neighbouring elements to its region with every iteration. Figure 3.3 shows an illustration of the region growing process from a initial seed element.

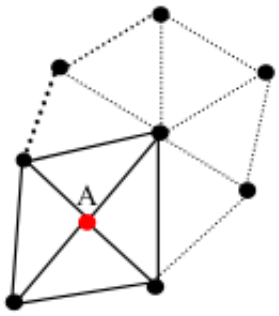


Figure 3.3: A Point **A**, in a Region growing approach, grows by adding neighbouring vertices

The process of repeatedly acquiring neighbours is done until a given growing criteria is reached. The key variations in the region growing techniques are determined by the growing criteria, which dictates whether any more mesh elements should to added to a given region [89]. Other issues that influence the various algorithms are, the selection of the initial seed elements which affects the number of segments induced, the need for post-processing smoothing or straightening of segmentation boundaries [89].

Vieira and Shimada [104] present a region growing algorithm which is based on the image-based version introduced by Besl and Jain [9]. Besl and Jain [9] perform the region growing process in range images. This work is extended by Vieira and Shimada [104] to three dimensional meshes where initial data points, labelled using the mean and Gaussian

curvature, are set as the seed elements from which growing will occur. Fitting a variable order bi-variate polynomials to the growing region is the growing criteria. Neighbouring elements are added to the region according to their compatibility with the approximating polynomial. The growing process terminates when elements can no longer be added to a given region [1].

Kalvin and Taylor [47] introduce the super-face algorithm which uses a set of representative planes for each region (or *cluster*) approximated by an ellipsoid. One consequent growing criteria used is the L_∞ *face-distance*. This face-distance is obtained by calculating the maximum distance from a set representative plane to any given vertex v in the mesh. The seed element for this technique is chosen randomly, and a post-smoothing process is often required to straighten out the borders between the final regions. Chazelle *et al* [17] and Lavoue *et al* [55] employ convex decomposition and surface curvature respectively as their criteria for growing regions. Lavoue *et al* [55] identify curvatures by detecting *sharp edges* and *sharp vertices*. Sharp edges are those which have the dihedral or torsion angle of the incident faces being greater than a set threshold. Sharp vertices are vertices which belong to sharp edges. However, in order to achieve better segmentations, Chazelle *et al* [17] use an additional constraint on the segment sizes.

Other region growing methods propose the use of multiple seed elements, such that each region grows simultaneously. Eck *et al* [26] present a method which attempts to create segments of a Voronoi pattern. A reiterative use of the dual of a given region as the seed face where other faces are added to it, is adopted throughout the region growing process. The growing criteria for adding faces is a prioritised approximation of geodesic distance between faces. Segmentations obtained by this method are bound by the following constraints, (i) Any given segment must be homeomorphic to a disk, in that, each segment must form a single connected component, (ii) Two segments cannot

share one consecutive boundary, and finally, (iii) The maximum number of segments that can share a single vertex is three.

Watershed Segmentation

Although watershed-based segmentation methods are put under their own category by Agathos *et al* [1], Shamir [89] duly notes that watershed-based approaches are simply a type of multi-seed region growing method. The watershed segmentation algorithm was originally used for two and three dimensional image segmentation. At the core of the technique is the analogy of how water fills geographic surfaces. Water continuously flows into the basins of the surface until they meet. The points where flooded basins (called *catchment basins*) meet form a segmentation boundary with each catchment representing a segment [1]. Figure 3.4 demonstrates the catchment flooding analogy.

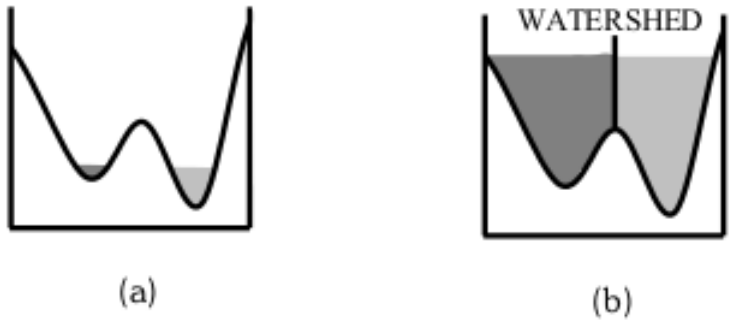


Figure 3.4: Watershed flooding analogy (a) shows the initial catchment basins. (b) shows filled catchments and their corresponding boundaries [1].

For 3D mesh segmentations, the criteria for determining the catchment basins is not obtained directly from the 3D mesh, but from a transformed version derived from a watershed function $f : R^3 \longrightarrow R$. The algorithm begins by first finding and labelling points of local minima of this function f (also called the *height function* or the *feature*

energy function). There exists a one-to-one correspondence between the local minima and the catchment basins, which consequently serves as the initial seed elements for a prospective region. A region incrementally grows until it reaches a maxima (or *ridge*) in the watershed function. The resulting basins formed correspond to the segments of an induced segmentation [89].

The notable differences in the watershed techniques arise from the choice of watershed function employed. A majority of the watershed algorithms, however, use surface curvature as the watershed function [1]. Pulla *et al* [79] make use of several forms of surface curvature including, *Gaussian, Mean* and *Absolute curvature*. Mangan and Whitaker [63] use a vertex discrete curvature function which represents the square root of the deviation from surface flatness. This deviation D is given by,

$$D = \sqrt{k_1^2 + k_2^2}, \tag{3.7}$$

where k_1 and k_2 are principal curvatures. These surface curvature methods often suffer from issues of only segmenting along regions of high curvature, and also will often require post-smoothing processes to correct the rough segmentation boundaries.

Zhou and Huang [123] use the *Average Geodesic Distance (AGD)* as the basis of definition of the watershed function. Wu and Levine [113], on the other hand, exploit the distribution of electrical charge on the mesh surface. It is an inherently surface-based curvature criteria, in that, the density of charge is higher along surfaces of sharp *convexities*, and lower on parts of sharp *concavities*. The corresponding basins are located at regions of charge density minima. Koschan [52] attempts to abate the prevalent problem of over-segmentation found within several of the watershed approaches. In that, the basins are filled up until a certain limit is reached. This limit is determined by both a threshold and the use of 3D morphological operators to prevent the spilling over from

one basin to another.

In region growing-based approaches, the issue of determining the initial seed elements is handled quite adequately by the watershed methods which use elements of the watersheds function minima. However, in cases where a watershed function cannot be computed, random seeds are often used.

3.3.2 Clustering methods

The class of clustering-based segmentation methods can be divided into two, namely, *Iterative clustering* and *Hierarchical clustering* [1, 89]. Iterative clustering is a parametric search process where a given k number of clusters is set a-priori. The process of finding the k clusters can be presented as an iterative search of identifying the best segmentation for the given k number clusters. The *K-Means algorithm* [25] is the most often used iterative clustering algorithm, which repeatedly attempts to classify mesh elements as a member of either of the k initial representative clusters. After each iteration of adding an element of an initial set to a cluster, the representative element (also known as the *medoid*) for each cluster is re-computed and the element addition process is repeated. When the representatives of the clusters stop changing, the clustering process terminates. Shlafman *et al* [93] utilize the k-means clustering algorithm computed on the dual graph of the mesh model. They define the distance $Dist$ between two faces, f_1 and f_2 as the weighted summation of the geodesic distance between the barycenters of the faces, and the angle (δ) between their corresponding faces [1],

$$Dist(f_1, f_2) = (1 - \delta)cos^2(\alpha) + \delta BaryDist(f_1, f_2), \tag{3.8}$$

where $BaryDist(f_1, f_2)$ is an approximate geodesic distance defined as the sum the distances from the center of the given faces f_1 and f_2, to the center of their shared edge e_{12}

[93]. Representatives are chosen for each cluster, and the process of selecting the best cluster representatives is repeated by finding the faces which minimize the sum of the distances to all the other faces which belong to its cluster.

Hierarchical clustering methods are a quasi–'global-greedy' region growing method. However, unlike region growing methods, the seed element selected does not remain as the cluster representative (or reference point of region growth), but is continually optimized to obtain the best merging operation for all the clusters [89]. Hierarchical methods also start with each face as its own segment. A merging cost value is assigned to each pair of clusters. The lowest computed cost between any given pair results in the merging of those two clusters. Garland *et al* [33] propose a hierarchical segmentation method which is well suited for mesh models with highly planar segmentation areas. They perform a cluster expansion on the dual graph of the mesh. The cluster expansion cost is based on the L_2 distance and a measure of planarity computed from orientation norms of representative planes. Also, the resultant shape of the obtained clusters are biased towards a circular compact shape by factoring the correlation between the square of clusters perimeter and $4\pi A$, where A is the surface area of the clusters [33, 89].

Attene *et al* [4] adopt a similar hierarchical clustering technique as used by Garland *et al* [33]. The mesh faces are classified into clusters based on the measure of their approximation to a finite set of fitting primitives—spheres, planes, cylinders. The cluster growth cost is computed by the addition of the face that results in the least error in fitting a resultant cluster to the aforementioned primitives. The method is well suited to CAD (Computer Aided Design) mesh segmentations, due to the abundance of the fitting primitives found within three-dimensional CAD objects. It also provides a semantically meaningful interpretation of the high-level mesh segmentations it computes [1, 4].

Gelfand and Guibas [34] likewise present another hierarchical clustering technique

which is based on *slippage analysis* or *slippable motions*. Slippable motions are rigid motions which cause a gap-less sliding motion of a stationary shape and its transformed version (obtained by applying the slippable motion to it). A segment is therefore slippable, if its set of connected components (vertices and faces) can be approximated by this motion. Like Attene *et al* [4], it is also suitable for CAD models such that, it's able to identify spheres, planes, and cylinders, amongst others.

3.3.3 Spectral Analysis methods

Spectral Analysis methods perform segmentations not based directly on the three-dimensional mesh embedding, but on the spectra of the underlying graph. The main mathematical operator which is used in Spectral graph theory is the graph *Laplacian*. Given a graph G induced by mesh M, let A be the *adjacency matrix* of graph G, and D be the *diagonal matrix* with the *degree* (*valence*) of vertex i as elements of its *leading diagonal* $d_{i,i}$. Then the Laplacian L of G is defined as the matrix,

$$L = D - A. \tag{3.9}$$

The combinatorial graph partitioning problem may be reduced to a geometric space-partitioning problem by embedding the graph G into the space R_k using the first k eigenvectors of the Laplacian [89]. We present an in-depth discussion of the Laplacian in Chapter 4.

Work by Liu and Zhang [58] construct a Laplacian using the affinity matrix W containing the elements $0 \leq w_{ij} \leq 1$, where w_{ij} corresponds to the likelihood of the faces i and j being placed in the same cluster or segment from an induced segmentation. The diagonal matrix, D, is constructed with its columns comprising of the normalized first k eigenvectors of the Laplacian. Due to the close relationship between the character-

istics of a graph and the algebraic properties of its Laplacian, semantically meaningful segmentations are produced by this approach. The segmentations are achieved by applying K-Means clustering to the row vectors of V, each of which is considered a point in k dimensional space. Given the technique's keen attention to concavity, areas of subtle concavity are still segmented. This sometimes produces poor quality segmentations [1, 58].

Zhang and Liu [120] use a graph bi-partitioning approach which is based on the *Normalized Cut algorithm* from the work of Shi and Malik [92]. They also adopt the affinity matrix used by Liu and Zhang [58]. Zhang and Liu [120] employ a novel sampling scheme (based on the *Nystrom approximation* [30]) in constructing their affinity matrix W with a sample size of two faces. The corresponding eigenvectors of W are computed, and the two leading eigenvectors obtained are used in bi-partitioning the mesh by finding the most salient cut. Apart from being able to also produce meaningful segmentations, their approach is relatively computationally less expensive due to the partial construction of the affinity matrix W.

Liu and Zhang present an extension of their work from [58] in [59]. Here, the mesh segmentation is guided by the outer contour of the two dimensional embedding of the mesh. However, two distinct affinity operators are defined based on the *structural segmentability* and *geometrical segmentability*. The affinity matrix used for the structural segmentability is the regular graph adjacency matrix defined as,

$$W_{ij} = \begin{cases} 1 & \text{if} e_{ij} \in E \\ 0 & \text{otherwise} \end{cases}. \tag{3.10}$$

The geometrical segmentability uses the minimum principal curvature and is sensitive to concavities on the mesh. If there exists an edge $e_{ij} < E$ between each pair of vertices

i,j, then W_{ij} is defined as,

$$W_{ij} = \begin{cases} (|\overrightarrow{\mathrm{K}_i}| + |\overrightarrow{\mathrm{K}_j}|) \cdot |\langle \overrightarrow{e}, \overrightarrow{\mathrm{K}} \rangle \cdot l & \text{if } \mathrm{K}_i < 0 \text{ or } \mathrm{K}_j < 0 \\ \varepsilon & \text{otherwise} \end{cases}, \tag{3.11}$$

where, $\overrightarrow{\mathrm{K}_i}$ and $\overrightarrow{\mathrm{K}_j}$ are the minimal principal curvatures at vertices i and j respectively, e denotes the direction of the edge e_{ij}, l is the normalized length of the edge e_{ij}.

3.3.4 Discussion

While most of the Region Growing methods have the benefit of being unsupervised segmentation methods, a post-processing phase is often required to correct prevalent issues such as over-segmentation. Clustering methods, on the other hand, present a semi-supervised approach to segmentation which are generally only well suited for CAD models, due to the criteria of clustering the mesh faces to primitives. Finally, Spectral based segmentations frequently produce semantically meaningful partitions arising from the close correlation between the underlying spectra (or eigensystem) of a given mesh object and its geometry, thereby making spectral methods more invariant to the type of 3D models being segmented. Furthermore, certain parameter choices for obtaining the spectra and their subsequent clusters further enhance the quality of segmentations produced by such methods. We thus employ this approach in our work.

3.4 Summary

Not only are there several algorithms for handling image segmentations, there also exist several works which have established viable benchmarks for evaluating the quality of segmentations induced by these image segmentation algorithms. Being a relatively less

matured field of research, three dimensional mesh segmentations algorithms have been able to adapt many image segmentation methods to the mesh segmentation field. In this chapter, we provided an introductory discussion on segmentation and mesh segmentation. We also presented a formalization of the mesh segmentation problem in the vein of [89]. We discussed works that aim to present a set of formalized objective evaluation metrics based on ground-truth segmentations generated by humans. However, the conclusion is still made that, the notion of 'quality' of segmentation is very much subjective to the area of application. We finally provided a discussion of the background works in three (3) major classification of mesh segmentation algorithms, namely, Region growing, Clustering, and Spectral based mesh segmentations.

The next chapter presents a detailed discussion of the variations of the graph Laplacian or Laplace Beltrami operator and its corresponding properties. The Laplacian is a mathematical operator which possesses several useful properties and is used in deriving the spectra of a given mesh object, as will be discussed next.

Chapter 4

Laplace-Beltrami Operator

The *Laplace-Beltrami Operator* (also known as the *Laplacian* or *Laplace matrix* for certain cases) is a mathematical operator that has been shown to possess extremely useful properties which addresses several problems in varying fields of application. The solution to the Laplace equation helps model the behaviour in electric and fluid potentials in the area of electromagnetism and fluid dynamics/mechanics, respectively. Particularly in computer graphics and related fields, three-dimensional shape signatures and descriptors, graph segmentation, data representation, geometric optimization and dimensionality reduction are some of the areas within which the Laplace-Beltrami operator has proved highly beneficial. To this effect, we discuss the Laplace-Beltrami operator, its mathematical and spectral properties, and finally, its various discrete representations for use in three-dimensional mesh operations.

4.1 The Laplace-Beltrami Operator

The *Laplace operator* is a well researched mathematical operator which possesses many useful properties. Sometimes referred to as the *Laplacian*, the Laplace-Beltrami operator

is indeed a generalization of the Laplace operator to operate on functions defined on surfaces in manifolds—*Riemannian* and *pseudo-Riemannian manifolds* (including Euclidean space) [106, 108]. Its use has grown in popularity particularly in Computer Graphics, Computational Geometry and other related fields of study where mesh processing tasks are dominant due to the representation of objects of interest as polygonal meshes. Such mesh processing applications include, shape interpolation, mesh filtering, simulation, surface smoothing using mean curvature flow, pose transfer, parametrization, compression, surface reconstruction, shape representation and segmentation [7, 106]. Also, computing the eigenfunctions of the Laplace operator of a surface presents a foundation of close representation of the given surface's geometry [24]. Recall from the discussion in Chapter 3 that, Levy [56] and Reuter *et al* [82], amongst many others, utilize this Laplacian property of '*understanding geometry*' as the principal feature for their shape analysis and segmentation. Another interesting characteristic of the Laplace-Beltrami operator is its close correlative behaviour to the propagation of heat. This property of heat propagation is based on the heat equation and the diffusion of heat as determined by the geometry of target object surfaces. The heat propagation property has consequently been exploited by several research works for varying purposes, including employing it as the basis for shape signatures and representations for object matching and retrieval [13, 73, 83, 90, 96]. (These works are discussed in detail in the following Chapter 5).

The initial key step required to utilize the Laplace-Beltrami operator in applications is the conversion from a continuous to a discrete operator *i.e.*, *discretization*. The discretization computed by any given method is expected to present an accurate representation of the underlying surface Laplacian for any given mesh. Such that, it retains the essential properties inherent in the continuous context [7, 108]. The structural properties often required to hold true of the discretized Laplacian in these applications are,

symmetry, sparsity, linear precision, positivity, and *convergence* [108].

To this effect, several discretizations of the Laplace-Beltrami operator for meshes have been proposed. Reuter *et al* [82], in their analysis of the robustness of the eigenfunctions of the Laplacian obtained from several proposed discretization methods, group them into two main classes, namely, *Finite Element Methods (FEM)* and *Geometric Methods*. Classification into either is mainly based on the methods used in computing the constituent matrices of the corresponding Laplacian matrix. However, it is noted that majority of the discretization methods are a variation of the so-called '*cotangent scheme*', which was first introduced by Pinkall and Polthier [78]. Although very popular as a result of adequately approximating the continuous operator for sufficiently uniform meshes, the cotangent scheme suffers from a convergence issue (L_2 and *pointwise*) for non-uniform meshes, as theoretically and experimentally investigated by Xu [114] in the analysis of the convergence of the different Laplacian discretizations. Recent works, including Belkin et al [7], attempt to address this issue by proposing a discretization scheme which guarantees L_∞ convergence (and implicitly L_2 convergence). Also, results of an empirical study on the convergence of the Laplacian by [6] show that, convergence of the Laplacian translates into convergence of its corresponding eigenfunctions. Further discussions and definitions of the various discrete Laplacian approaches are presented in the following sections.

4.2 Definition of the Laplace-Beltrami Operator

Let S be a smooth manifold, possibly with boundary, with a Riemannian metric. Let ∇ denote the gradient. The Laplace-Beltrami operator $\triangle$ (or ∇^2), of a given twice continuously differentiable function $f \in C^2$, is the *divergence* (div) of the *gradient* (grad)

of S in Euclidean space [7, 106, 108],

$$\triangle f = \nabla^2 f = \operatorname{div} \operatorname{grad} f = \operatorname{div}(\nabla f) = \nabla \cdot \nabla f. \tag{4.1}$$

The gradient of a scalar field is a corresponding vector field which measures the size and direction of the highest rate of change (*steepest slope*) of the scalar field. Divergence can be thought of as the net flow (*outflow*/*inflow*, or *expansion*/*compression*) of a vector field at a given point. The divergence is a positive value when representing an expansion, and a negative value when otherwise [70, 107]. Figure 4.1 demonstrates a very simplified illustration of divergence on a three-dimensional vector field.

4.3 Properties of the Continuous Laplacian

As noted before, the continuous Laplace-Beltrami operator possesses certain key properties which are essential in its use in applications. The properties, as presented by Wardetzky *et al* [106], include, *null*, *local support*, *symmetry*, *Linear precision*, *maximum principle* and *positive semi-definiteness*. For the definitions of these properties, let L_2 be the intrinsic inner product of functions u and v on a smooth manifold S be given by,

$$(u, v)_{L^2} = \int_S uv dA \tag{4.2}$$

Null : When u is constant, then,

$$\triangle u = 0. \tag{4.3}$$

Symmetry : When u and v vanish along the boundary of S and are sufficiently smooth, then,

$$(\triangle u, v)_{L^2} = (u, \triangle v)_{L^2}. \tag{4.4}$$

Local Support : Given two points x and y on S, where $x \neq y$, and where x and y are some distance apart, the function value of u at x, *i.e.*, $u(x)$, is unaffected by the resultant operation of the Laplacian locally, $\triangle u(y)$.

Linear Precision : When u is a linear function on the Euclidean plane, $u = ax+by+c$,

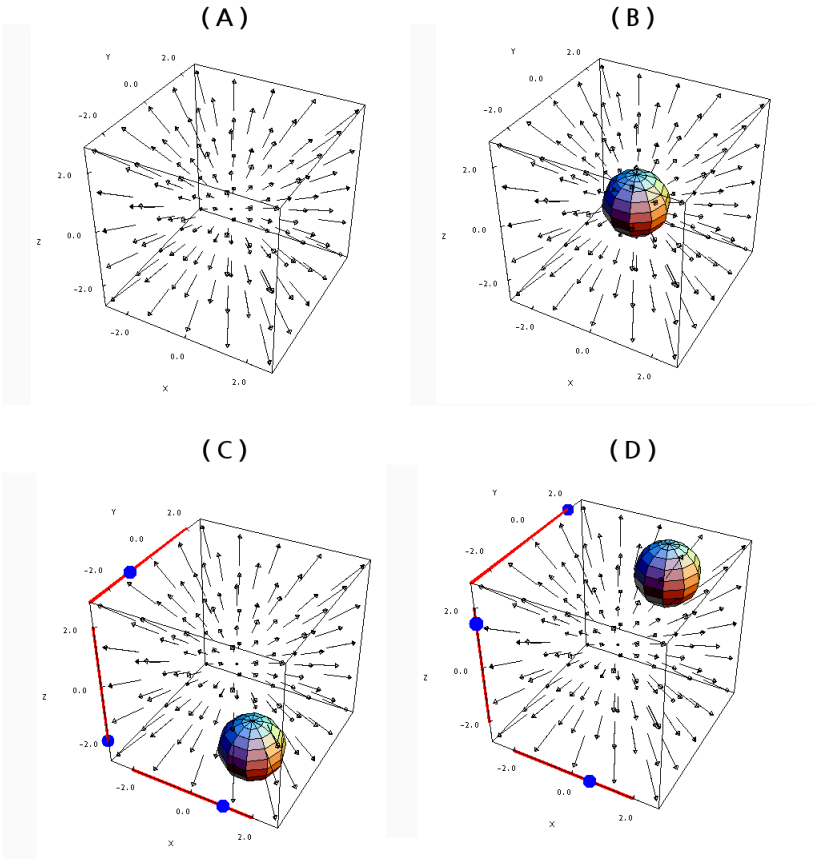


Figure 4.1: Illustration of divergence on a simple three-dimensional vector field. (A) shows a vector field which flows out from the origin. The magnitude increases at it moves outward from the center. (B) shows the measurement of divergence at the origin. Given the simplicity of the vector field, a simple visual inspection shows that the divergence at the origin is a positive value since its net flow is an outward flow or expansion. Both (C) and (D) also result in a net positive value for the divergence at different points. This is because the net outflow from the sphere is always greater than the net inflow into it. (Note: the lengths of the arrows in the vector field denote the magnitude) [70]

then the action of the Laplacian remains null, $\triangle u = 0$, whenever S forms part of the Euclidean plane.

Maximum Principle : *Harmonic functions*—functions which remain null (*i.e.*, $\triangle u = 0$) in the interior of the smooth surface S—have no *local extrema* (local maxima or minima) at their interior points.

Positive Semi-definiteness : This property relates to the Dirichlet energy E_D, over a given domain S. This is given as,

$$E_D(u) = \int_S \|\nabla u\|^2 dA, \tag{4.5}$$

and is a non-negative real number *i.e.*,$E_D(u) \geq 0$, for every given function u. Given an appropriate sign for the Laplacian,

$$E_D(u) = (\triangle u, v)_{L^2} \geq 0, \tag{4.6}$$

whenever u is sufficiently smooth and vanishes along the surface's boundary.

Applications of the Laplacian require that the proposed scheme for a discrete Laplace-Beltrami operator preferably possess all the above properties of the continuous Laplacian. Unfortunately, this is not the case. Wardetzky *et al* [106] show that, no current presented discrete Laplacian is able to retain all the properties of the continuous operator, but the price of attaining some properties always comes at the cost of losing others—"*No free lunch*". Levy [56] emphasizes this point by reiterating that, all the proposed *discrete* Laplace-Beltrami operators are just that, and should not be mistaken as the *discretized* Laplacians of its continuous counterpart.

4.4 Proposed Discrete Laplacians

We present some works for several proposed schemes of the discrete Laplace-Beltrami operator.

Definitions Let M be a triangular mesh with n vertices, denoted by $M = (V, E, F)$. Let V be the set of vertices, with each vertex $i \in M$ denoted in absolute Cartesian coordinates as, $v_i = (x_i, y_i, z_i)$. Given the continuous Laplacian as defined prior in Equation 4.1 as, $\triangle f = \nabla^2 f = \text{div}\,(\text{grad}\, f)$. Then the *Laplacian eigenvalue problem* is given as [82],

$$\triangle f = \lambda f. \tag{4.7}$$

The discrete solution to this equation is often approximated by a piecewise linear function over the triangular mesh $f : M \longrightarrow R$. The function f linearly interpolates values of $f(v_i)$ over the vertices of M. The discrete Laplacians are often represented by [82],

$$\triangle f(v_i) = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij}[f(v_i) - f(v_j)], \tag{4.8}$$

where, $N(i)$ are the members of the immediate neighbourhood of vertex v_i *i.e.*, the degree or valence of vertex v_i, d_i is the associated mass assigned to vertex v_i, w_{ij} is the symmetric weight assigned to the corresponding edge between vertex v_i and v_j. A subsequent matrix representation of the above defines a vector of the function for all the vertices v_i to v_n,

$$\mathbf{f} = [f(v_i), \ldots, f(v_n)]^{\mathrm{T}}. \tag{4.9}$$

Likewise, a *weighted adjacency matrix* $W = (w_{ij})$ contains all the corresponding neighbour edge weights for all the vertices. This matrix is usually symmetric and sparse. A *volume matrix* $U = \text{diag}(u_1, \ldots, u_n)$, which is a diagonal matrix with elements on its

leading diagonal u_i, is defined as,

$$u_i = \sum_{j \in N(i)} w_{ij}. \tag{4.10}$$

Given the weighted adjacency matrix W and the volume matrix U, we define the *stiffness matrix* A as,

$$A = U - W, \tag{4.11}$$

and the *lumped mass matrix* D being, $D = \text{diag}(d_1, \ldots, d_n)$. Finally, the *Laplace matrix*, L is defined with respect to the stiffness and lumped mass matrices as,

$$L = D^{-1}A. \tag{4.12}$$

With these matrices defined, the Laplacian eigenvalue problem can be written as,

$$L\mathbf{f} = \lambda\mathbf{f}. \tag{4.13}$$

The above equation can be better expressed as a generalized symmetric eigen decomposition problem with respect to the stiffness and lumped mass matrices as,

$$A\mathbf{f} = \lambda D\mathbf{f}. \tag{4.14}$$

It should be noted that most of the variations in the different discrete Laplacians proposed are primarily based on the different techniques for calculating the weights and masses that constitutes the stiffness and mass matrices, and consequently the Laplacian.

4.4.0.1 Graph Laplacians

Graph Laplacians are a type of Laplacian which use a simplified edge weighting system where $w_{ij} = 1$ if only there exists an edge between vertex v_i and v_j, otherwise, a value of 0 is assigned. Unit masses are assigned with respect to the weighting scheme *i.e.*, $d_i = 1$. This scheme, also called a *combinatorial Laplacian*, is employed by Zhang [121].

A slight variation is adopted in the Tutte Laplacian. Here, each edge weight is defined by a ratio with respect to the neighbourhood of the vertex to which the edge is connected. That is,

$$w_{ij} = \frac{1}{N(i)}, \tag{4.15}$$

where, $N(i)$ is the valence of vertex i. Wardetzky *et al* [106] identify the graph laplacians as failing to possess the continuous Laplacian geometric property of *linear precision*.

4.4.0.2 Cotangent Laplacians

The popular *cotangent scheme*, originally presented by Pinkall and Polthier [78], employ constant masses $d_i = 1$, with edge weights defined by,

$$w_{ij} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2}, \tag{4.16}$$

where, α_{ij} and β_{ij} are the two opposite angles to the edge e_{ij}. Figure 4.2 illustrates the angles used in cotangent scheme.

Desbrun *et al* [23] address the issue in the pure cotangent scheme where the cotangent weights are solely dependent on the mesh sampling. This is so due to the lack of adequate mass weighting values for d_i. The authors resolve the sampling dependency issue by proposing a mass weighting value which takes into account the area of the surrounding

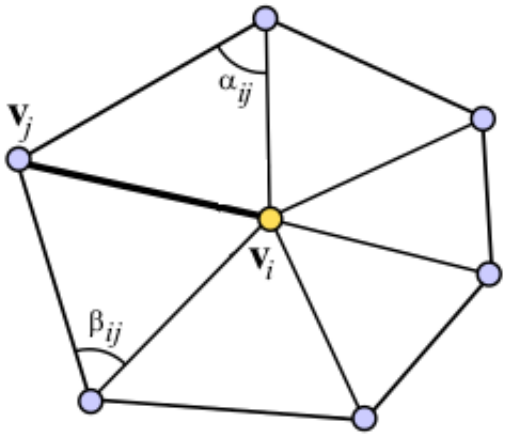


Figure 4.2: α_{ij} and β_{ij} angles for weighting edge e_{ij} (edge between vertices v_i and v_j) in the cotangent scheme for discrete Laplacians.

triangles of each vertex. That is,

$$d_i = \sum_{i \in N(i)} A(i)/3, \tag{4.17}$$

where A is the area of a given neighbour triangle of vertex i.

Meyer *et al* [65] likewise address the mass weighting issue of the original cotangent scheme by employing a mass value which uses the *Voronoi region* around a given vertex. The Voronoi region (also called the *dual area*) of a given vertex is formed by the region obtained from joining the *barycenters* or *circumcenters* of the surrounding triangles of the given vertex. Figure 4.3 illustrates the weighting scheme. The mass weighting is therefore defined as,

$$d_i = A^*(i), \tag{4.18}$$

where A^* is the area of vertex i's Voronoi region.

Reuter *et al* [83] present a variation of the cotangent scheme which uses the *Finite*

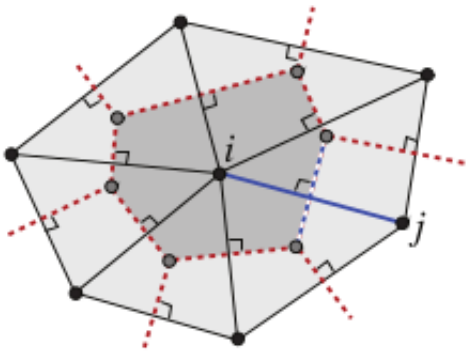


Figure 4.3: The light shaded region depicts the area of the triangles surrounding vertex i as used in [65]. The darker area is the Voronoi region obtained from connecting the corresponding barycenters of the surrounding triangles.

Element Method (FEM) approach and is based on the core principle of verifying the Laplacian eigenvalue problem in a weak sense, in that, $\triangle f = -\lambda f$ which results in the generalized eignenvalue problem given as,

$$A_{\text{cot}}\mathbf{f} = \lambda B\mathbf{f}, \tag{4.19}$$

where, $\mathbf{f} = [f(v_i), \ldots, f(v_n)]^{\text{T}}$, and A_{cot} (the stiffness matrix with cotangent weights) and B are respectively defined as,

$$A_{\text{cot}}(i,j) = \begin{cases} \frac{\cot(\alpha_{ij})+\cot(\beta_{ij})}{2} & \text{edge } e_{ij}, \\ -\sum_{k\in N(i)} A_{\text{cot}}(i,k) & i = j \end{cases}, \tag{4.20}$$

$$B(i,j) = \begin{cases} \frac{|t_1|+|t_2|}{12} & \text{edge } e_{ij}, \\ \frac{\sum_{k\in N(i)} |t_k|}{6} & i = j, \end{cases}. \tag{4.21}$$

where, $|t_i|$ is the area of the triangle t_i, t_1 and t_2 are the triangles that share the edge

e_{ij}. Reuter *et al* [83] state that, unlike the mass matrix D, the matrix B is a more robust and easily extends to higher polygonal dimensions for the underlying mesh. They further show results of faster eigen decomposition times as compared to the the regular cotangent schemes.

Due to the nature of the cotangent operator, Wardetzky *et al* [106] duly note that cotangent scheme violates the continuous Laplacian property of lacking positive edge weights for general meshes. Sorkine [95] further reiterates the problematic nature of handling negative weights in certain applications, although for adequately uniform meshes, this does not often pose a problem.

4.4.0.3 Cotangent Laplacians

Belkin *et al* [7], in an attempt to address the convergence problem inherent to the cotangent scheme, present a discrete Laplacian which samples the k-nearest neighbours for a given set of vertex points. In so doing, they consider not only the immediate connections of a given vertex, but a slightly larger neighbourhood. Such that, a defined *mesh Laplace operator* L_h on a function f for any given vertex v in V is given as,

$$L_h f(v) = \frac{1}{4\pi h^2} \sum_{t \in M} \frac{A(t)}{t_v} \sum_{p \in V(t)} e^{-\frac{\|p-v\|^2}{4h}} (f(p) - f(v)), \tag{4.22}$$

where, M is the underlying mesh with vertex set V, $A(t)$ corresponds to the area of a given triangle t, and t_v is the number of vertices in t, the input parameter h is a positive integer which is the number of neighbour points to consider per a given vertex. Results from their approach show convergence in L_∞ (and implicitly L_2) and is robust in terms of the triangulation of the mesh surface. However, a Laplacian can only be obtained this way on compact closed surfaces, given the approach is unclear as to how boundary conditions (*e.g.*, Dirichlet) are handled.

4.4.0.4 Mean-Value Laplacians

The *Mean-value Laplacians* employ convex edge weights in an attempt to emulate the cotangent scheme. Mean-value Laplacians may use one of the many variations of the mass weighting used in the aforementioned cotangent schemes, and use an edge weighting given by,

$$w_{ij} = \frac{tan\left(\frac{\theta_{ij}^1}{2}\right) + tan\left(\frac{\theta_{ij}^2}{2}\right)}{\|\mathbf{v}_i - \mathbf{v}_j\|}, \tag{4.23}$$

where, θ_{ij}^1 and θ_{ij}^2 are the adjacent angles to the edge between $\mathbf{v}_i$ and $\mathbf{v}_j$ as shown in Figure 4.4.

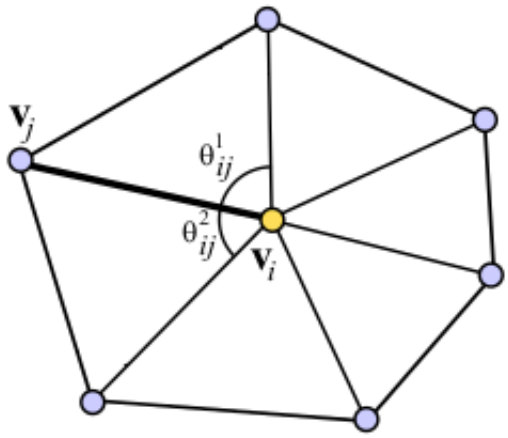


Figure 4.4: θ_{ij}^1 and θ_{ij}^2 are the angles used in the Mean-value Laplacian scheme.

The Graph Laplacians poorly approximate the continuous Laplacian and are therefore often inadequate in addressing problems which rely heavily on the surface geometry of the input object meshes. The Cotangent Laplacians, although not fully convergent, retains most of the continuous properties and is also able handle meshes with or without boundaries. This makes it suitable to a wider array of applications where models are

not always compact manifolds, but may be mesh surfaces with defined edges. In summary, this section considered discrete Laplacians that are useful for modelling 3D object surfaces.

4.5 Summary

The Laplace-Beltrami operator is a mathematical operator with several useful properties for modelling and manipulating object surfaces represented as meshes. To this effect, several discrete Laplacians have been proposed, all of which aim to preserve the properties of the continuous version. However, no single discrete Laplacian is able to retain all the characteristics of the continuous Laplacian. The popularity of the cotangent scheme is justified by the fact that it has been shown to approximate the continuous Laplacian well. In this chapter, we discussed the Laplace-Beltrami operator, its properties and presented the various schemes for its discrete construction.

The following Chapter 5 discusses three-dimensional object mesh descriptors, and in particular the *Heat Kernel Descriptor (HKS)* which serves as a very informative 3D shape descriptor.

Chapter 5

Three-dimensional Shape Descriptors

Several search and retrieval techniques and applications exist and are being advanced for many electronic content types such as, text documents, audio files, and video files. Resulting from the drastic increase in the availability of three-dimensional object models in both the industry and public domains, the recent decade has brought an increase in research into several approaches that best address the problem of 3D object retrieval. For such applications, one of the major steps in the retrieval process is being able to give a "*meaningful description*" of the 3D objects in its database. This description allows for obtaining adequate comparisons for object matching, which consequently returns useful results for any given search query.

In this Chapter, we present an overview of the different techniques employed in 3D object retrieval, with specific interest in *Content-based* approaches and the object descriptors employed for such purposes. We then give a detailed discussion on one such descriptor technique called the *Heat Kernel Signature* (also known as the *Heat Kernel Descriptor*), which is based on the mode of heat transfer or diffusion over an objects

surface.

5.1 3D Object Retrieval

Advances in technology and its ubiquity has resulted in a rapid increase in the amount of digital data that are continually captured and stored. Such data include, *textual data* (which is the most occurring form of recorded data), *audio, videos, two-dimensional (2D) images* and *three-dimensional (3D) objects*. Though 3D object data are collected in relatively less amounts as compared to the other more traditional data forms, recent developments in modelling, visualizing and digitizing techniques for 3D objects has drastically increased the availability of such data both in the public domain (primarily on the Internet), and also in the industrial domain [16, 99]. For the purposes of visualization, 3D objects are often stored and represented as polygonal meshes or polygon soups (an unorganized group of triangles/polygons with generally no existing relationship between them) formed from vertices connected through edges. Popular file formats for storing these 3D objects include, 3D Studio format (*.3DS*), Wavefront OBJ format (*.OBJ*), and Virtual Reality Modelling Language format (*.VRML*) [99, 116].

With any facility that allows the storage of data comes the logical resultant need to be able to retrieve that stored data, and particularly parts of the data, based on specific inputs or queries. Given the rapidly increasing corpus of 3D data, the need for 3D object retrieval from large databases is prevalent in several disciplines including, Computer-Aided Design and Manufacturing (CAD/CAM), game design and development, entertainment, computational and molecular biology, military applications, medical imaging, Virtual Reality (VR), and the new area of Augmented Reality (AR) [16].

The main approaches that seek to address the problem of 3D object retrieval may be put into two categories, namely *Semantic based* and *Content based* techniques.

Semantic based approaches do not rely directly on the actual geometry or appearance data of the 3D object, but rather on the accompanying textual descriptive data (*annotations*) associated with the object [32]. The data are often provided by the human creators of the model, or by domain experts who analyse the objects after their creation. The use of such annotation in searching for 3D objects ultimately reduces the 3D object retrieval problem to a text analysis problem.

Content based retrieval approaches, on the other hand, use the geometric and appearance information, usually stored as polygon and texture data, to define a basis for the notion of "similarity" between any given pair of 3D objects [16].

In some fields of application, there exist a sharp difference in the meaning of the low-level features as represented by the geometry, and its actual high-level semantic interpretation. In such instances, semantic based retrieval methods perform well. However, content based approaches have been shown to be generally more effective in a wider range of applications as compared to the semantic methods [66]. One of the main drawbacks of the semantic methods is the limitations and ambiguity that arise from the environmental conditions that influence the annotations provided by humans. This includes the language, specific field of application, gender, age, and culture, amongst others [66, 99].

5.2 Content-based 3D Object Retrieval

Recall that Content-based 3D object retrieval systems rely on the underlying geometric and appearance information of the object. The main problem posed in using the geometric data is to find an appropriate representation or *descriptor* which can be used as a basis for comparing 3D objects by providing an adequate notion of similarity. Unlike traditional text query/retrieval systems, multimedia retrieval systems (especially for 3D

objects) often have the goal of returning results which are similar to the query object, rather than returning exact matches. This is mainly due to the fact that, for many applications, there is a very low probability of having two objects which are an exact match without them, most likely, being some exact digital copies. Therefore, a distance measure is a consequence of the notion of a defined *similarity*. Hence, given an input query object, a typical multimedia system will return either, the *k nearest neighbours*—the k most similar objects to the query object—or all objects which are at a thresholded distance ε from the query object [16].

5.2.1 3D Object Descriptors

As noted prior, the success or failure of performing suitable similarity comparisons between objects is very much dependent on the 'competence' of the object descriptor employed. The distance that exists between any two pairs of descriptors dictates how similar (or *dissimilar*) they are, such that, small distances correspond to high similarity and large distances to low similarity. For the following discussion, we adopt the use *dissimilarity* (in the same vein as [99]) as a better semantic correlation to the notion of *distance measures* between objects. That is, a small distance corresponds to a small/low dissimilarity, likewise a large distance to a large/high dissimilarity. The formalization of the dissimilarity measure d on a set S can be given as a non-negative real valued function [99],

$$d : SxS \longrightarrow \mathbb{R}^{+} \cup 0. \tag{5.1}$$

Preferably all of the following five (5) properties of the function d should be possessed by any proposed dissimilarity measure [99]:

(i) Identity : $d(x, x) = 0, \forall\, x \in S$.

This simply states that, any given shape is exactly similar to itself, such that the

dissimilarity measure/distance when compared to itself is zero (0).

(ii) Positivity : $d(x, y) > 0, \forall\, x, y \in S : x \neq y$.

This property states that, any two different objects are not completely similar. Tangelder and Veltkamp [99] note that, this property is often very difficult to attain in high-level shape descriptors. However, it is often permissible to ignore strict adherence to this requirement if the failure is as a result of trivial details.

(iii) Symmetry : $d(x, y) = d(y, x), \forall\, x, y \in S$.

Symmetry is the equivalence of similarity of objects when compared to each other. However, as pointed out by Tangelder and Veltkamp [99] based on the observations of Tversky [103], this property is not always needed for human-oriented applications, because humans do not always perceive a given object x as being equally similar to another object y, as y is to x. Nevertheless, this property is required in applications where high precision comparisons are done, *e.g.*, in medicine and molecular biology applications.

(iv) Triangle Inequality : $d(x, z) \leq d(x, y) + (dy, z), \forall\, x, y, z \in S$.

The triangle inequality property preserves the close correlation in the sense of physical distances as used in gauging similarities of objects.

(v) Transformation Invariance : $d(g(x), g(y)) = d(x, y), \forall\, x, y \in S, g \in G$, where G is monotonic.

In applications where *orientation-* and *scale-invariant* object descriptors are desired, this property has to be met.

A *dissimilarity metric* is attained when all five (5) properties hold true for a given dissimilarity measure. When only the identity, symmetry and triangle inequality properties

are satisfied, the dissimilarity measure is considered a *pseudo-metric*, and a *semi-metric* when identity, positivity and symmetry hold true [99].

For the purpose of applications, certain other desirable properties that shape descriptors should possess have been proposed as follows.

Scope : Given any class of 3D object model, the object descriptor should be able to adequately describe the object [43]. In that, the descriptor should be able to handle manifold models, compact (or *watertight*) models, models with boundaries, spherical (curved) solids, amongst others.

Efficiency : This describes the computational and storage requirements needed for comparisons to be made. To enable fast searches and retrievals, the object descriptor should implicitly or explicitly allow for efficient indexing structures. This, in effect, often results in efficient search, comparison and retrieval within applications [99].

Effectiveness & Discriminative power : *Effectiveness* is the ability of a descriptor to properly match similar objects, such that, given a query object, similar objects returned possess '*true likeness*' based on the descriptors ability to properly discriminate. The notion of '*similarity*' is however quite subjective, and is therefore dependent on the domain of application [99]. Also, *precision* (the fraction of the returned objects that is significant with respect to a given query object) and *recall* (the fraction of all the significant objects which were returned) are two of the most popular effectiveness measures used [16].

Sensitivity : This property requires that a descriptor should be able to capture slight and subtle details and nuances of the object's shape [43].

Robustness (Stability) : A somewhat conflicting property to the *sensitivity* property, *robustness* requires that an object's descriptor is able to remain robust to changes.

This means that small changes in the object's shape should correspond to small changes in the object's descriptor [43], but should remain invariant to noise and negligible subtleties [99]. Iyer *et al* [43] indicate that, the conflicting properties of sensitivity and robustness can both be satisfied if the descriptor can separate the information that captures the general characteristics of the object, and that which represents more specific details of the object.

Multi-scale (Multi-resolution) support : This denotes descriptors that are able to hold information of an object's shape at different scales in a hierarchical structure, such that varying levels of detail are captured and stored for specific purposes [43].

Conciseness and Ease of indexing : This is mainly a property which addresses computational and storage costs of using any given object descriptor. Basically, a descriptor should preferably be of adequately low dimensionality, provide a means for indexing in order to facilitate accelerated searches in databases, and require minimal storage space [51].

We note that the other proposed properties are somewhat ultimately inherent within the aforementioned properties, *e.g.*, the need for *pose normalization*, and *partial matching* can both be appropriately sub-categorised as a consequence of the robustness property.

5.3 Proposed 3D Object Descriptors

We continue with a review of the recent techniques for three-dimensional shape descriptors that have been proposed in the literature. We present an abridged categorization based on the works of Tangelder and Veltkamp [99], Iyer *et al* [43], and Yubin *et al* [107] as, (i) *Feature based*, (ii) *Graph based*, (iii) *Appearance attribute based*, and (iv) *other techniques*.

It should be noted that the classification given is not strictly disjoint, and the techniques used in a particular class may overlap with those of another. This is maintained by all the aforementioned authors.

5.3.1 Feature based methods

Feature based methods employ the use of features of the 3D objects. Features may be represent the geometric and topological properties of the object. Feature based techniques can further to sub-categorized into (I) *global feature based*, (II) *global feature distribution based*, (III) *local feature based*, and (IV) *spatial maps based*.

Techniques (I),(II) and (IV) use a single d-dimensional vector as the resultant descriptor for a given object. Given the dimension of the descriptor vector is fixed at d for all objects, this allows for similarity measuring by computing the distance in Euclidean space between any two given descriptor vectors as points in a d-dimensional space [99]. The local feature based methods on the other hand, have surface point descriptors for a set number of points on the surface of the object, which describes the immediate 3D shape around that point.

5.3.1.1 Global Feature based methods

Global feature based methods utilize global features of the 3D object model. Such properties include the *statistical moments*, *Fourier descriptors* (of the volume or the object boundary), and *geometric ratios*. The moments m_{pqr} in three-dimensions of the order $p+q+r$ of a given piecewise continuous 3D object model $S(x_1, x_2, x_3)$ are defined as,

$$m_{pqr} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1^p x_2^q x_3^r \, S(x_1, x_2, x_3) dx_1, dx_2, dx_3. \tag{5.2}$$

Hence, the set of moments $mpqr$, for $p, q, r = 0, 1, 2, \ldots$ uniquely represents the 3D object model $S(x1, x2, x3)$. Certain moment invariants can defined which render the moments invariant to rotation, translation and scaling. Also, a small number of lower order moments can be employed as a feature vector for for 3D objects [43].

Chen and Chen [19] utilized moments addressing 3D protein structure group similarity searches. They reduce the comparison of these 3D structures to a spatial comparison of the sets of points that make up the structure by extracting the moments and aspect ratios of the points.

In addressing the shape description problem, Paquet *et al* [74] represent the coarse shape, scale and composition properties of both 2D and 3D objects using the bounding boxes, cords-based, moments-based and wavelets-based descriptors.

Based on other geometric features, Corney *et al* [21] present an descriptor based on the *convex-hull* of a 3D object model. Properties such as the *hull crumpliness*, *hull packing* and *hull compactness* serve as the basis for the proposed descriptor. Hull crumpliness is the ratio of the surface area of the object to the surface area covered by its convex hull. While the hull packing of a 3D object is the percentage of convex-hull volume which the object occupies. Finally, the hull compactness is the ratio obtained from the convex-hull cubed surface area to the convex-hull squared volume [99]. A related volumetric method, presented by Sanchez-Cruz and Bribiesca [87], uses the computed volumetric error that would arise from transforming an result object into the query object. This method however, tends to be far more computationally expensive for 3D objects relative to the 2D object domain from which it was adapted.

Using Fourier transform coefficients, object volume, total surface area and moments, Zhang and Chen [118] present a method for defining a descriptor computed directly from the object's 3D mesh representation without any transformation of the mesh into any

other representation, *e.g.*, to a volumetric representation. Features for each polygon (usually a triangle) on the mesh is computed and subsequently aggregated over the entire object to give a final global feature descriptor. They later show in [119] that, through an active learning process where additional information is acquired in the form of annotations by a domain expert, the discriminatory precision of their object matching retrieval systems improves sufficiently.

Elat *et al* [27] likewise apply a mixed approach of also factoring in an active learning process to better the retrieval results. They propose an interactive and iterative process where weights are assigned to a returned similarity distance measure as determined by a user's ranking of whether a query result was "*relevant*" or "*irrelevant*". A *Support Vector Machine (SVM)* based approach is used in iteratively adapting subsequent query results closer and closer to the user's "expected output".

5.3.1.2 Global Feature distribution based methods

Unlike global feature based methods, global feature distribution based methods do not define the descriptor immediately from the features of the object, but rather from the distributions of these features.

Osada *et al* [72] present an object descriptor where the distribution of the objects features, such as the *angle, distance* and *volume measurements*, between randomly sampled surface points form the basis of the descriptor. Similarity is evaluated by measuring the distance between the objects' distributions using a pseudo-metric. For the different distance measures used in evaluating their approach—D_2 shape distribution, L_1 norm, and mean normalization—the experimental results show that the D_2 shape distribution distance measure presented the best matches.

The D_2 shape distribution function, introduced by Osada *et al* [72], is later extended

by Ohbuchi *et al* [71] in presenting a pair of descriptors which are applicable to both manifold and non-manifold 3D object models, and robust to geometrical and topological errors and degeneracies. The proposed pair of shape descriptors are both obtained by first transforming a given object into an oriented point set model, whose joint 2D distance and orientation histogram between surface point pairs are then computed. One of the proposed descriptor pairs computes the *Absolute Distance histogram*, from a parametrized value which denotes the distance between a given pair of random points, while the *Absolute Angle histogram* is parametrized by the angle (solid angle) between the surfaces on which the random point pair lie. Experimental results from their research show an improved performance over the original D_2 shape distribution function, but at a higher computational cost.

Ohbuchi *et al* [71] improve on their earlier Absolute Angle-Distance descriptor by presenting a multi-resolution approach which computes the Absolute Angle-Distance descriptor on a number of prior computed multi-scaled alpha-shapes over the object surface. Even better retrieval performance is reported over their previous method. However, a relatively higher computational cost is yet again a trade-off in using this approach.

5.3.1.3 Local Feature based methods

Unlike their global counterparts, local feature based methods consider the surface featured within a neighbourhood of some set of points on the 3D object model.

Zaharia and Preteux [117] present the *3D Shape Spectrum Descriptor (3D SSD)* which is based on the shape index, and provides an intrinsic shape descriptor of an underlying 3D object mesh. This shape index, first introduced by Koenderink [50], is a local surface geometric property which is an expression of the angular coordinate of a polar representation of the principal curvature vector. The 3D SSD is defined as a distribu-

tion of the shape index values computed over the whole object mesh. The shape index value is computed for each mesh face by fitting a quadratic surface to the corresponding centroids of each face. The authors report fast computational times and small storage requirements resulting from a compact object descriptor as advantages to their approach. However, the required computationally expensive pre-processing stage for non-orientable and topologically incorrect meshes pose a drawback to their method.

Based on 2D concept of *Shape Contexts*, Kortgen *et al* [51] extend this concept into the three-dimensional vector space and utilize it as the basis of their proposed object descriptor. Given an object mesh and a set of randomly sampled center points of N faces, consider an $N-1$ set of vectors propagating from one sample point to all the remaining $N-1$ points on the surface of the object. For a chosen point p, the coarse histogram computed from the relative coordinates of the remaining $N-1$ points constitute the *Shape Contexts* of the reference point p. Kortgen *et al* [51] show experimental results that demonstrate the descriptors robustness to topological and geometrical noise or artifacts. The descriptor is however, susceptible to rotations and therefore requires a pre-alignment processing stage.

A relatively more recent local feature based descriptor, known as the *Generalized Shape Distributions (GSD)*, is presented by Liu *et al* [60]. Although classified here under local feature based methods, GSD indeed utilizes both local and global shape signatures for effective shape matching and analysis. Local descriptors for each point are computed by first creating spin images at each of these points. The distribution of the Euclidean distance of pairs of local shape clusters, formed from a quantization by K-Means clustering of the local descriptors, constitutes a given object's descriptor. Liu *et al* [60] state the two major benefits of their proposed descriptors as, robustness to geometric noise such as shape occlusions and deformations, and high discriminative power resulting

from the inherent consideration for the spatial locations of the local descriptors factored into the subsequent global descriptor.

Recent work by Sun *et al* [96] propose a isometry-invariant and topologically robust shape descriptor based on heat diffusion over an object surface. The descriptor is obtained by restricting the *heat kernel* of the Laplace-Beltrami operator of each point on a shape to the temporal domain alone. Though a local feature based method, the Heat Kernel Signature can be employed as global feature descriptor by using methods such as, vector quantization and Bags of Features. Further discussion of this descriptor is given in Section 5.4.

5.3.1.4 Spatial Maps based methods

Spatial maps represent an object by capturing the physical locations of sections of the object, in that, an arrangement of the map's entries within an object's spatial map preserves the relative positions of the different captured features of the given object [99].

A spatial spherical harmonics based descriptor is presented by Vranic *et al* [105]. For each ray propagating from the origin of a given object, a descriptor is computed for the objects surface by a value with corresponds to the distance from the ray origin and the point of intersection with the object model. Their proposed method is only rotation-invariant once a pose normalization has been carried out on the object models. Vranic *et al* [105] use the *Continuous Principal Component Analysis (PCA)* algorithm to achieve the pose normalization. They suggest that the continuous version performs better than the conventional PCA algorithm.

Kazhdan *et al* [49] address the rotation invariance for descriptors by employing spherical harmonics to make rotation dependent descriptors rotation invariant. Object shapes defined in *voxel* (volumetric pixels) grids or as a collection of spherical functions can be

described by this method. A final 2D histogram descriptor is obtained by first decomposing the collection of spherical functions into its corresponding spherical harmonics, and then computing the sum of the harmonics after which, the L_2-norm of each frequency component is also calculated.

5.3.2 Graph based methods

Graph based models, unlike the prior discussed feature based methods, do not obtain their descriptors directly from an objects 3D mesh, but attempt to capture a correlation between the object's underlying geometric information and a corresponding computed graph structure which preserves the relationships between the object's components.

A sub-classification of graph based methods uses *skeleton graph* for deriving its object descriptor (*skeletal graph based methods*). The skeleton of an object is considered the "*central-spine*" or "*stick figure*" representation of the object and is centered within it. The skeleton has close correlations with the *medial-axis* and *medial surface* for 2D and 3D objects, respectively [10]. Figure 5.1 shows an illustration of a few sample 3D object models and their resultant skeletons.

Sundar *et al* [97] propose a descriptor which encodes both the topological and geometrical information in the form of a skeletal graph for a given object model. Arriving at this descriptor first involves, obtaining the volume of the object, and then computing a set of skeletal nodes and forming an undirected acyclic graph from connecting these skeletal nodes using the Minimum Spanning Tree algorithm. Segments of the skeleton correspond to a node on the graph structure. Object comparisons are made by approximate matching of their skeletal graphs. A fast database indexing is achieved by assigning to each non-terminal node, an eigenvector representation of the subgraph adjacency matrix rooted at a given node. The authors suggest several useful benefits arising from

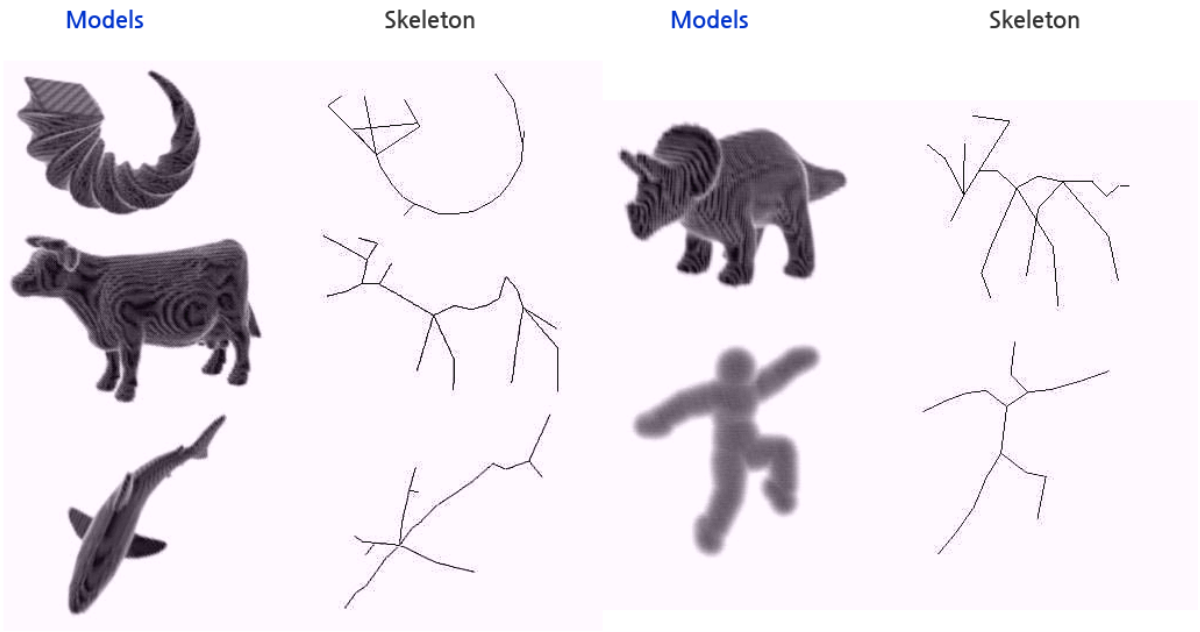


Figure 5.1: Illustration of sample object models and their corresponding skeleton graphs [97].

their proposed method. These include using their method for part or component matching, visualization, providing an intuitive representation of object models, and articulated object matching.

Another sub-category of the graph based methods is the *Reeb graph based* approaches. The Reeb graph is defined as the quotient space of an object model S and a quotient function f [99]. Bespalov *et al* [10] research the use of Reeb graphs for solid models. Their work is based of previous research carried out by Hilaga *et al* [38] who use quotient function based *Multi-Resolution Reeb Graphs (MRG)* which is constructed by using the resulting values of a predefined function that reflects the features of a given object model. Bespalov *et al* [10] employ the MRG descriptor and evaluate its applicability on several CAD models including primitive objects, and relatively more complicated mechanical parts such as articulated object models. From their experimental results, that point

out that a drawback to using the Reeb graphs method arises when handling complex structured objects, such that, the MRG descriptor becomes less sensitive to topology and more dependent on the geometry of the object.

5.3.3 Other methods

Several other shape matching methods have been researched that employ diverse approaches. ***Appearance attribute based*** *methods* rely on the appearance attributes like colour, texture, materials, specularity of the object's surface, and others, to describe the given object. Naturally, retrieval methods based on such a descriptor alone consequently presents several limitations, given it requires continuous user input, and results are also often very subjective and biased by the user's preferences [116]. Appearance attributes methods are best used as a complementary descriptor to other more comprehensive geometric descriptors.

Volumetric error based *methods* attempt to provide a shape descriptor based on the volumetric error obtained from comparing the query object to a sequence of *offset hulls* of other objects. The offset hulls of two objects A and B finds how much of the volume of object A is inside that of object B. Closest matching objects are the ones which minimize the error. Novotni and Klein [69] propose such a descriptor for 3D shape matching.

Investigations into ***View based*** *methods* for 3D object matching and retrieval have been carried out by several researchers including, Mahmoudi and Daoudi [62], Funkhouser [31], Chen *et al* [18]. View based methods operate on the notion of how humans discriminate between perceived objects. In identifying an object, humans first construct a representation of a perceived object from the retinal image. This representation is then compared with a remembered prior description. A match is returned if the representation

and the remembered description look similar from all viewing angles [100]. Funkhouser [31] reduce the 3D object matching problem to a 2D image comparison problem by implementing a 2D sketch query interface. Here, a set of thirteen (13) binary thumbnail images of boundary contours of the object, as projected from thirteen orthogonal viewpoints, define the descriptor for the 3D object models. A similarity is found by performing image matching comparisons on the image sets of corresponding query objects. Chen *et al* [18] show better retrieval results than [31] by an extension of their work which uses ten (10) 2D image silhouettes encoded by their Zernike moments and Fourier descriptors, which are obtained from ten (10) uniformly sampled viewpoints over the 3D objects viewing sphere.

In summary, for the purposes of partial shape matching, global feature based methods are inadequate or would require major modifications in order to be able to address such problems. Graph based methods, although topologically robust, likewise suffer from their inability to capture surface geometric properties. Therefore, in order to be able to appropriately compare surfaces, a local feature based method is needed. One such potent local feature based descriptor is the Heat Kernel Signature, which we employ in our research.

5.4 Heat Kernel Signature (HKS)

We present a detailed discussion on the feature based object descriptor as recently proposed by Sun *et al* [96] and extended by Ovsjanikov *et al* [73] and Sharma *et al* [90].

The underlying principle that serves as the basis for the *Heat Kernel Signature (HKS)* is the propagation or diffusion of heat over 3D object surfaces, which is completely described by the *heat kernel* associated with the object's Laplace-Beltrami operator. As

first introduced by Sun *et al* [96], the HKS serves as a highly informative pointwise descriptor obtained by restricting the heat kernel to the temporal domain over the object. The Heat Kernel Signature is shown, both theoretically and practically, to posses the following properties [96]: Firstly, it is an efficient multi-scale organization of intrinsic geometric information of a given object or shape. Secondly, it is concise and commensurable. Thirdly, it is stable and robust to shape perturbations. Finally, it can be efficiently estimated.

The HKS is also able to capture information about the neighbourhood of points on the object, making it very well suited for partial shape matching. Given a source point on an object's surface, heat diffuses to wider areas from that source. Adjusting the time parameter at which information is captured provides a logical sense of scale in describing the shape surrounding the given point. This subsequently facilitates multi-scale matching between points by comparing the obtained point Heat Kernel Signatures at varying time intervals [96].

5.4.1 Heat Kernel

We provide the definition of the heat kernel as given by [96]. The heat propagation (or diffusion) of a compact Riemannian manifold M, possibly with boundary be governed by the heat equation,

$$\triangle_M u(x,t) = -\frac{\partial u(x,t)}{\partial t}, \tag{5.3}$$

where, $\triangle_M$ is the Laplace-Beltrami operator (as discussed in Chapter 4) of the manifold M, u is a continuous smooth function. The Dirichlet boundary condition $u(x,t) = 0\ \forall x \in \partial M$, $\forall t$, will have to be satisfied for M with boundaries.

There exists a function $k_t(x, y) : R^+xMxM \longrightarrow R$ for any M, such that,

$$H_t f(s) = \int_M k_t(x, y) f(y) dy, \tag{5.4}$$

where, f is an initial heat distribution $f : M \longrightarrow R$, $H_t(f)$ denotes the heat distribution at time t, the minimum function $k_t(x, y)$ that satisfies Equation 5.4 is known as the *heat kernel*. Simply put, the heat kernel can be described as the amount of heat which diffuses from x to y in the time t when a unit heat source is applied at x. The restrictions of the heat diffusion on the boundary elements on a given manifold presents a close correlation to the Green's Function, which stipulates specified initial boundary conditions over a defined domain.

It is observed that the heat kernel possesses the following properties, (i) of being *symmetric* : $k_t(x, y) = k_t(y, x)$, (ii) satisfying the semigroup identity : $k_{t+s}(x, y) = \int_M k_t(x, z)\, k_s(y, z)\, dz$, (iii) of being isometry-invariant, (iv) informative, (v) multi-scale, and (vi) stable.

5.4.2 Heat Kernel Signatures

The Heat Kernel Signature is computed by eliminating the excess information that is captured in the heat kernel. This is achieved by considering only the heat propagation of reference points along the diagonal of the heat kernel matrix obtained from a manifold. Hence, for a given manifold M and a point x on M, the Heat Kernel Signature of $x, HKS(x) : R^+ \longrightarrow R$ is a function over the temporal domain given by [96],

$$HKS(x, t) = k_t(x, x). \tag{5.5}$$

The Laplace-Beltrami operator of the compact manifold M, has a discrete eigen decomposition of the form,

$$\triangle_x \varphi_i = \lambda_i \varphi_i, \tag{5.6}$$

where, λ_i and φ_i for $i = 0, 1, 2, \ldots$ are the eigenvalues and eigenfunctions respectively. The heat kernel can then be written as,

$$k_t(x, y) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \varphi_i(x) \varphi_i(y). \tag{5.7}$$

When the heat diffusion on a surface is considered from a point x to itself, the above Equation 5.8 may then be written as,

$$k_t(x, x) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \varphi_i(x)^2. \tag{5.8}$$

The subsequent Heat Kernel Signature for a given point x can then be given as a compact n-dimensional descriptor vector $\mathbf{p}(x) = (p_1(x), \ldots, p_n(x))^{\mathrm{T}}$, which contain the elements,

$$p_i(x) = c(x)\, k_t(x, x), \tag{5.9}$$

where, $c(x)$ is a normalization constant such that $\|\mathbf{p}(x)\|_2 = 1$. Sun *et al* [96] maintain that the HKS retains all the salient properties found in the heat kernel. Figure 5.2 demonstrates the propagation of heat for given point on a topologically modified object at different times.

Based on the above defined descriptor, Ovsjanikov *et al* [73] implement a topologically robust object matching application. The authors compute the final object descriptor using vector quantization, where a chosen vocabulary (which constitutes the *medoids* of the clusters) of size V of "*geometric words*" is obtained by clustering over all n pointwise

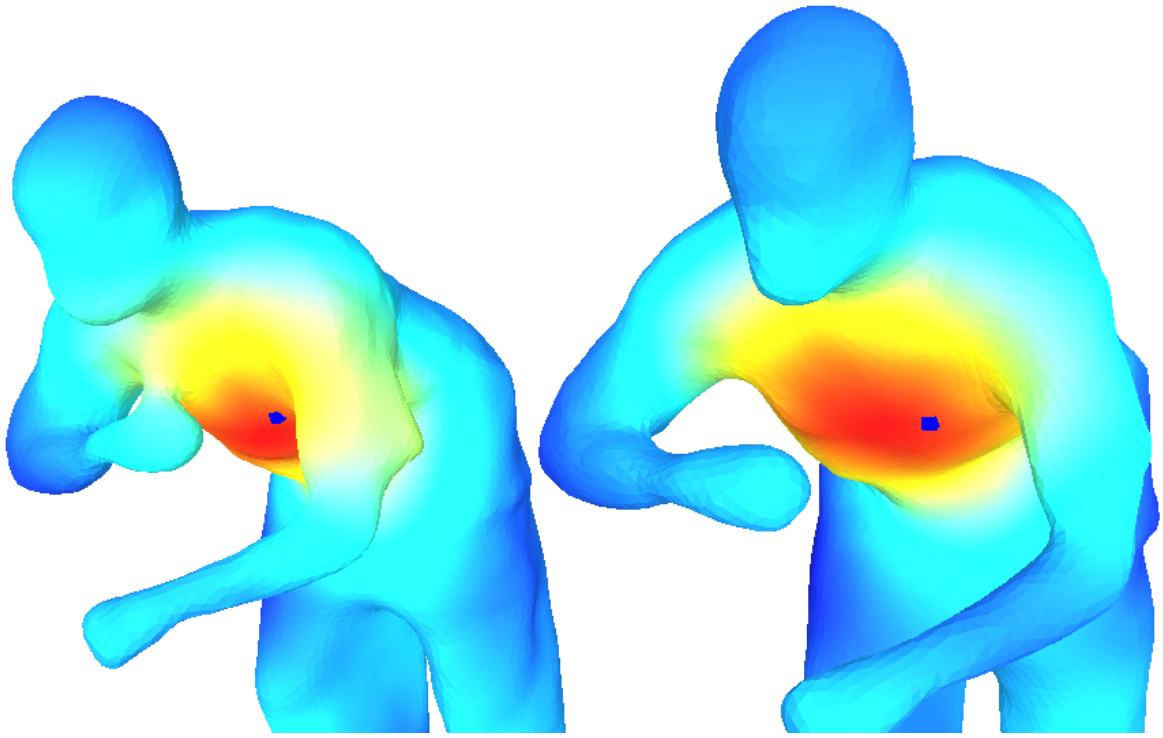


Figure 5.2: Illustration of heat propagation/diffusion from a single point over a 3D object and its topologically modified version, at two (2) time intervals [96].

signatures on the object using the K-Means algorithm, and then finding the feature distribution of each of the representative vectors (medoids) of the clusters with respect to all points over the object surface. This is a widely used technique in Computer Graphics called *Bag of Features (BoF)*. The authors present an alternate discriminative descriptor called *Spatially-Sensitive Bags of Features (SS-BoF)* which, unlike bag of features, considers both the distribution and the spatial location or relations of the words. Their experimental results show better retrieval performances of the spatially-sensitive bag of features over the the conventional bag of features for compact manifolds.

Sharma *et al* [90] employ the Heat Kernel Signature in proposing a topologically robust shape matching method which is ideal for finding dense correspondences between 3D objects. The method grows from an initial set of sparse points, called *seeds*, and gets

denser by increasing the number of points to be matched between a given pair of 3D shapes. Their approach, therefore, enables matching the correlation between parts of an articulated object.

5.5 Summary

Appropriate 3D object descriptors allow for the comparison, matching and fast retrieval of objects from a database. In this chapter we have reviewed the ideal properties most, and preferably all, of which any object descriptor should posses. We then provide a classification of the different types of three-dimensional object descriptors which have been proposed in the literature, namely; feature based methods, graph based methods and other methods. We further give a discussion of some of the methods found under of each of these categories. We finally expound on an isometry, topology and possibly scale-invariant feature based object descriptor based on heat diffusion on shapes, called the Heat Kernel Signature (HKS).

Chapter 6 discusses our approach for addressing the protein docking problem.

Part III

ADDRESSING THE DOCKING PROBLEM

Chapter 6

Addressing the Protein Docking Problem

We recall that two main approaches are employed in the several techniques that have been used in the literature in addressing the protein docking problem. Our goal is to find the appropriate match of a protein receptor and a corresponding protein ligand, in binding to form a stable complex. This is either addressed as a free energy optimization problem where combinatorial optimization algorithms are used in trying to match a protein and a ligand, or as a geometry-based shape complementarity matching problem, addressing possible structural deformation due to the flexibility of the active sites of the protein structures.

In this chapter, we present a geometry-based shape matching technique which utilizes an isometry-invariant and topologically robust shape descriptor based on the Heat Kernel Signature (HKS). The Heat Kernel Signature is also suggested to be well suited for partial shape matching. We therefore present our main *ProtoDock algorithm* and a secondary alternate algorithm for addressing the protein docking problem. The first method segments the complex three-dimensional structures of the protein objects into

smaller segments (or patches). This is done in order to reduce the overall computational complexity of the algorithm, by comparing the relatively smaller patches for possible geometric complementarity matches. The second method for addressing the docking problem considers the entire three-dimensional structures as a whole. However, it performs a partial point-wise match by comparing single points on the objects' surfaces. These points are representative of the immediate region to which they belong, and therefore serve as viable basis for comparing those regions.

6.1 Overview of the ProtoDock Algorithms

We present our *ProtoDock algorithm* which is a geometric feature based isometry-invariant and topologically robust shape matching algorithm. Our algorithm can be used for both whole and partial shape matching. The descriptor employed for its shape comparison is based on the Heat Kernel Signature, as originally proposed by Sun *et al* [96] and introduced in Chapter 5. The salient isometry-invariant property of the HKS particularly helps circumvent the issue of the combinatorially many alignment possibilities of a given pair of docking objects. It further helps address possible structural deformations that may be present when pairing the protein objects.

For a target three-dimensional mesh object, our algorithm begins by first constructing an appropriate discrete Laplace-Beltrami operator. As discussed in Chapter 4, there are several discrete Laplace-Beltrami operators which have been proposed in the literature, and which all attempt to preserve as many properties as possible inherent in the continuous version. The spectra of the Laplacian is the computed to obtain the eigenvalues and eigenvectors. This serves as the key underlying component for the remainder of our ProtoDock algorithm, by finding segments of the mesh object obtained from clustering the eigenvectors.

After segmentation, we again find the spectra of the Laplacian for each partition. Taking each segment as a complete three-dimensional structure, we then compute the Heat Kernel Signature for each vector point on the structure. Clustering the Heat Kernel Signatures to obtain the representative vectors (medoids) of each cluster serves as the basis for deriving our final compact descriptor vectors of the three-dimensional structure. A possible docking pair is found by calculating the distance between any two (2) target three-dimensional protein segments using an appropriate distance measure (usually Euclidean distance).

We note that since the Heat Kernel Signature is an intrinsic descriptor obtained solely from the mode of heat propagation/diffusion over a given surface. It therefore does not require an external reference frame, but only describes an object solely based on its actual geometric structure. Hence, our docking problem is duly addressed as a deformable (or non-rigid) shape complementarity matching problem.

6.2 Overview of the ProtoDock-1 Algorithm

Our ProtoDock-1 Algorithm performs a partial shape matching in four (4) main phases. Firstly, we construct the Laplacian matrix and compute its eigenvalues and eigenvectors. Secondly, we perform a segmentation of the 3D protein object mesh. Thirdly, we compute a segment description using the HKS descriptor. Finally, we obtain the compact descriptor vector for each segment and perform a comparison with other protein segments to find a match.

We discuss each of the four (4) phases of the ProtoDock algorithm in the following sections.

6.2.1 Laplace Eigen Decomposition

The first phase of our ProtoDock algorithm requires the construction of an Laplace-Beltrami operator and a subsequent eigen decomposition (*i.e.*, obtaining the eigenvalues and eigenvectors) of this operator associated with the three-dimensional mesh representation of a target object.

Given an object's triangular mesh $M = (V, E)$, let V be the set of n vertices, with each vertex $i \in M$ denoted in absolute Cartesian coordinates as $v_i = (x_i, y_i, z_i)$, and let E be the set of edges where e_{ij} is the edge which connects vertex v_i to vertex v_j. We approximate the piecewise function $f : M \longrightarrow R$ over the triangular mesh by a discrete Laplacian usually represented as,

$$\triangle f(v_i) = \frac{1}{d_i} \sum_{j \in N(i)} w_{ij}[f(v_i) - f(v_j)], \tag{6.1}$$

where, $N(i)$ are the members of the immediate neighbourhood of vertex v_i *i.e.*, the degree or valence of vertex v_i, d_i is the associated mass assigned to vertex v_i, w_{ij} is the symmetric weight assigned to the corresponding edge between vertex v_i and v_j. A matrix representation of the above Equation 6.1 consequently defines a vector of the function f for all the vertices $v_i, \ldots, v_n \in M$ as,

$$\mathbf{f} = [f(v_i), \ldots, f(v_n)]^{\mathrm{T}}. \tag{6.2}$$

We proceed to construct the *weighted adjacency matrix* $W = (w_{ij})$ which contains all the corresponding neighbour edge weights for all the vertices on the mesh. The *volume matrix* $U = \mathrm{diag}(u_1, \ldots, u_n)$ is then formed. This is a diagonal matrix with elements u_i

on its leading diagonal. Each of these elements is defined as,

$$u_i = \sum_{j \in N(i)} w_{ij}. \tag{6.3}$$

For the majority of our experiments, we employ the use of the cotangent weights adopted from the cotangent scheme. As discussed in Section 4.4, the cotangent Laplacians have been shown to posses most of the properties of the continuous Laplace-Beltrami operator, and therefore sufficiently approximate the continuous version. The cotangent Laplacians are also convergent on uniform triangular meshes. The weights w_{ij} for each edge e_{ij} and therefore obtained by computing,

$$w_{i,j} = \frac{\cot(\alpha_{ij}) + \cot(\beta_{ij})}{2} \ , \tag{6.4}$$

where α_{ij} and β_{ij} are the two opposite angles to the edge e_{ij}.

Given the weighted adjacency matrix W and the volume matrix U, we construct the *stiffness matrix* A as,

$$A = U - W, \tag{6.5}$$

and the subsequent *lumped mass matrix* D is given as $D = \text{diag}(d_1, \ldots, d_n)$.

Finally, the *Laplace matrix*, L is defined with respect to the stiffness and lumped mass matrices as,

$$L = D^{-1}A. \tag{6.6}$$

With all the requisite matrices defined, the Laplacian eigenvalue problem is evaluated as a generalized symmetric eigen decomposition problem with respect to the stiffness and lumped mass matrices as,

$$A\mathbf{f} = \lambda D\mathbf{f}. \tag{6.7}$$

6.2.2 Mesh Segmentation

With the Laplace matrix constructed for the mesh M, and the its associated eigensystem (*i.e.*, eigenvalues and their corresponding eigenvectors) obtained, we proceed to perform a segmentation on the mesh into a set number of disjoint connected components. We adopt a spectral segmentation approach (Section 3.3.3) which finds a minimum cut on the triangular mesh represented as a graph G. We accomplish this by reducing the combinatorial graph partitioning problem to a geometric space-partitioning problem by means of embedding the graph G from the mesh M into the space R_k using the first k eigenvectors of the Laplace matrix. This is arrived at by clustering the eigenvectors of the Lapalce matrix in k clusters using the K-Means algorithm.

That is, given the first n_{eig} smallest eigenvalues with its eigenvectors of the Laplacian, we represent the eigenvectors as a column-major eigenvector matrix E_{vec} by stacking the eigenvectors into an $n \times n_{eig}$ matrix with each row normalized. We derive a segmentation on mesh M by using the K-Means clustering algorithm to obtain k disjoint clusters of vertex points v from E_{vec} , where each row of E_{vec} is treated as a point in n_{eig}-dimensional space. We then map the cluster indices to each vertex v in the mesh to its specific cluster. In so doing, each of the k sets of vertices, each with elements taken from the connected vertices v assigned to a particular cluster, forms a corresponding object segment.

6.2.3 Segment Heat Kernel Signatures

Having extracted the segments from the object mesh M, we proceed with the partial matching by considering each individual segment as a target object mesh M_i as a complete three-dimensional mesh object on its own. We subsequently compute the Heat Kernel Signature for each mesh segment M_i by first constructing a Laplace matrix for the segment (as was done in the prior Phase 1 (in Section 6.2.1) of our ProtoDock algo-

rithm). The eigensystem of the obtained Laplace matrix is then computed.

Recall from Section 5.4 that, the underlying concept that serves as the basis for the Heat Kernel Signature is the principle of heat diffusion over the surface of three-dimensional objects. This is entirely described by the *heat kernel* associated with the object's Laplace-Beltrami Operator. The HKS serves as a highly informative pointwise descriptor which considers only the heat propagation of the points along the diagonal of the Heat Kernel matrix. Recall that the HKS has been shown to posses several useful properties including, providing an efficient multi-scale organization of intrinsic geometric information of a given object or shape, being concise and commensurable, and being stable and robust to shape perturbations.

The heat propagation on a compact segment manifold M_i with an associated Laplacian $\triangle_{M_i}$, is governed by the heat equation $\triangle_{M_i} u(x,t) = -\frac{\partial u(x,t)}{\partial t}$. For our algorithm, the Dirichlet boundary condition $u(x,t) = 0 \; \forall x \in \partial M, \forall t$ is maintained for each segment mesh M_i. By considering only the diagonal elements of the Heat Kernel as a given point x on the mesh M_i, and with the Laplace-Beltrami operator of the manifold with a discrete eigen decomposition of the form, $\Delta_{M_i}\varphi_i = \lambda_i\varphi_i$, where λ_i and φ_i for $i = 1, 2, \ldots, n_{eig}$ are the first n_{eig} eigenvalues and eigenfunctions respectively, then the heat kernel (the fundamental solution of the heat equation) is derived by evaluating,

$$k_t(x,y) = \sum_{i=1}^{n_{eig}} e^{-\lambda t}\varphi_i(x)\varphi_i(y) \; . \tag{6.8}$$

The subsequent Heat Kernel Signature for each given point x on the segment mesh M_i, given a *heating times* list $\{t\}$ of time intervals t_{int} to a set t_{max} is computed as a compact t_{max}-dimensional descriptor $\mathbf{p}(x) = (p_i(x), ..., p_{t_{max}}(x))^{\mathrm{T}}$, which contains the elements,

$$p_i(x) = c(x)K_i(x,x) \; , \tag{6.9}$$

where, $c(x)$ is a normalization constant such that $\|\mathbf{p}(x)\|_2 = 1$.

6.2.4 Descriptor Vector Computation and Segment Comparison

The Heat Kernel Signature for each mesh embeds a considerable amount of pointwise geometric information, since each point on the mesh is characterised by its own Heat Kernel Signature. In order to be able to achieve sufficiently fast geometric comparisons of segment meshes, a more compact descriptor has to be obtained from the larger Heat Kernel Signature data.

We proceed to attain this compact descriptor vector by first clustering each Heat Kernel Signatures. Each cluster is characterised by its *medoid* (representative vector), which is the HKS associated with the point that is closest to the cluster centre as resulting from the application of the K-Means clustering algorithm. The set of the Heat Kernel Signatures associated with the medoids $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ thus forms the basis for obtaining our compact shape descriptor vectors. We present three different methods for constructing the compact descriptor vectors using these medoids.

The first method, the ***Bag of Features (BoF)*** [73], is an accepted method in Computer Vision. It performs a soft vector quantization between the medoids and all other points over the entire segment mesh. The remaining two methods, namely ***Closest Medoid Set (CMS)*** and the ***Medoid Set Average (MSA)***, are completely novel methods which we propose. They both use just the medoids, which capture adequate information about their neighbourhoods, without any relative quantization of the other points on the mesh. We present the details of our descriptor vectors methods below.

(I) Bag of Feartures (BoF)

For the Bag of Features (BoF) method, we consider the medoid set of size l (also popularly called a vocabulary set of "*geometric words*"). For each point x on the segment mesh M_i, with its corresponding Heat Kernel Signature $\mathbf{p}(x)$, we compute the *feature distribution* $\mathbf{\Theta}(x) = (\theta_1(x), \ldots, (\theta_l(x))^{\mathrm{T}}$, an $l \times 1$ vector with its elements $\theta(x)$ defined as,

$$\theta(x) = c(x)e^{-\frac{\|\mathbf{p}(x)-\mathbf{p}_i\|_2}{2\sigma^2}} , \tag{6.10}$$

where, $c(x)$ is a normalization constant such that $\|\theta(x)\|_2 = 1$, and σ is set to the median of the medoids or geometric words. The clustering is carried out on the HKS of a each segment mesh at a time, given that the HKS is able to capture geometric information of the surfaces, and hence sufficiently provides a description of the surface.

A final $1 \times l$ feature descriptor vector $\mathbf{J}(M_i)$ for a given mesh M_i, is then obtained by integrating over the entire segment mesh M_i as $\mathbf{J} = \int_{M_i} \theta(x)da(x)$. We accomplish this for each segment mesh by first stacking our computed feature distributions $\mathbf{\Theta}(x)$ in row-major order and summing up all the columns of the matrix.

We proceed to obtain possible docking sites by performing a segment matching from different segments of other protein structures by calculating the Euclidean distance between the descriptor vectors. For example, given the descriptor vectors, $\mathbf{J}(M)$ and $\mathbf{J}(N)$ for two segments M and N, the comparative distance d_{BoF} between the two segments is evaluated by,

$$d_{\mathrm{BoF}}(M, N) = \|\mathbf{J}(M) - \mathbf{J}(N)\|_2 . \tag{6.11}$$

(II) Closest Medoid Set (CMS)

We introduce the first of our two novel descriptor methods which we refer to as *Closest Medoid Set (CMS)*. Unlike the Bag of Features method, the Closest Medoid Set method

employs the $l \times t_{int}$ normalized medoid set of a given segment as its descriptor, where t_{int} is the number of time steps/intervals. The motivation behind utilizing only the medoid set arises from the observation that, the medoids are adequately representative of the entire mesh to which they belong, given a sufficient number of time intervals and medoids. To this end, we proceed to find the sum of the smallest distances between each pair of the medoids between two given medoid sets associated with their respective segments. Here, the similarity between two segments M and N, each with their respective medoid sets $P_M = \{\mathbf{p}_{M1}, \ldots, \mathbf{p}_{Ml}\}$ and $P_N = \{\mathbf{p}_{N1}, \ldots, \mathbf{p}_{Nl}\}$ is defined by,

$$d_{\text{CMS}}(M, N) = \sum_{i}^{l} \frac{\mathbf{min}_{j \in [1,l]} \|\mathbf{p}_{Mi} - \mathbf{p}_{Nj}\|_2}{l^2}. \tag{6.12}$$

(III) Medoid Set Average (MSA)

The descriptor vector obtained by the second of our novel methods, called the *Medoid Set Average (MSA)* method, is closely related to that of the CMS method discussed above. That is, it also employs the medoid set solely for its descriptor vector. However, the Medoid Set Average method also adopts the column-wise summation used in the Bag of Features method. Here, after normalization, we form our final $1 \times t_{max}$ descriptor vector $\mathbf{J}$ from the medoid sets of a given segment mesh by finding the column-wise average of the $l \times t_{int}$ medoid set matrix. The similarity measure d_{MSA} between two segments M and N is calculated by the Euclidean distance between the two descriptors, as in,

$$d_{\text{MSA}}(M, N) = \|\mathbf{J}(M) - \mathbf{J}(N)\|_2 \, . \tag{6.13}$$

6.2.5 ProtoDock-1 Algorithm

We present our ProtoDock-1 Algorithm for deformable shape complementarity matching for addressing the protein docking problem.

Input :

- Triangulated three-dimensional object mesh M with n vertices.
- k number of segments

Procedure :

I *Phase 1 - Laplace Eigen Decomposition*

1. Construct the $n \times n$ symmetric adjacency matrix $W = (w_{ij})$, where w_{ij} are the symmetric edge weights of the mesh M.
2. Construct the $n \times n$ volume matrix V, which is a diagonal matrix where v_i is the sum of all edge weights w_ij directly connected to vertex v_i.
3. Form the stiffness matrix $A = V - W$.
4. Construct the mass matrix D, with elements d_i computed using cotangent scheme (Other weighting systems may be used here).
5. Form the Laplacian $L = D^{-1}A$ (Like the prior step, choice of Laplacian type may be varied).
6. Solve the eigen decompostion problem $A\mathbf{f} = \lambda D\mathbf{f}$ the first n_{eig} smallest (or largest depending on the type of Laplacian) eigenvalues and corresponding eigenvectors of the Laplacian.

7. Construct a normalized column-major eigenvector matrix E_{vec} by stacking the eigenvectors into an $n \times n_{eig}$ matrix and normalizing each row.

II *Phase 2 - Mesh Segmentation*

8. Obtain a segmentation on M by using K-Means clustering to retrieve k clusters from E_{vec} where each row of E_{vec} is treated as a point in n_{eig}-dimensional space.
9. Map the assigned cluster indices back to each vertex v in mesh M.
10. The set of vertices V_i with elements selected from the connected vertices v assigned to cluster i form a corresponding object segment.

III *Phase 3 - Segment Heat Kernel Signatures*

8. For each segment considered as an independent mesh M_i, repeat **Steps 1** to **7** to obtain their associated eigensystem.
9. Compute the heating times list $\{t\}$ of time t_{int} intervals starting from minimum time to a set t_{max}.
10. Given the eigenvalues $\{\lambda_i\}$, their corresponding eigenvectors $\{\varphi_i\}$, and the heating times list $\{t\}$, for each vertex point v in mesh M_i, compute the t_{max}-dimensional Heat Kernel Signatures $\mathbf{p}(v) = (p_i(v), \ldots, p_{t_{max}}(v))^{\mathrm{T}}$, where boundary element have a heat kernel of 0. Normalize each row vector.
11. Using the K-Means algorithm, partition the Heat Kernel Signatures into l disjoint clusters to arrive at the medoid set $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ with elements $\mathbf{p}_i$ which are the Heat Kernel Signatures associated with the medoids of each cluster.

IV *Phase 4 (A) - Descriptor Vector Computation* - **Bag of Features (BoF)**

12. Given the medoid (or vocabulary) set $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ of l "*geometric words*", for each vector point v in segment mesh M_i with HKS $(p)(v)$, compute the $l \times 1$ feature distribution vector $\Theta(v)$, and normalize each vector $\Theta(v)$.

13. For each vector point v in M_i, form an $n \times l$ feature distribution matrix Θ by stacking each feature distribution vector $\Theta(v)$ in row-major order.

14. Obtain the final $1 \times l$ compact descriptor vector $\mathbf{J}(M_i)$ for the mesh M_i, by summing all columns of the feature distribution matrix Θ.

15. For any given pair of mesh segments with the BoF descriptor vectors $\mathbf{J}(M_i)$ and $\mathbf{J}(N_i)$, compare the segments by computing the Euclidean distance between the two vectors in l-dimensional space.

V *Phase 4 (B) - Descriptor Vector Computation* - **Closest Medoid Set (CMS)**

16. Given the medoid set $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ of l medoids, compute the final mesh descriptor by normalizing each medoid vector $\mathbf{p}$ in the medoid set.

17. A comparison between two mesh segment descriptors $\mathbf{J}(M_i)$ and $\mathbf{J}(N_i)$ is computed by finding the sum of the squared average of the smallest distances between each pair of medoids in the segment descriptors.

VI *Phase 4 (C) - Descriptor Vector Computation* - **Medoid Set Average (MSA)**

18. Given the medoid set $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_l\}$ of l medoids, normalize each medoid vector $\mathbf{p}$.

19. Stack the medoid set into a column-wise $l \times t_{int}$ medoid set matrix.

20. Obtain the final $1 \times t_{int}$ compact descriptor vector $\mathbf{J}$ by finding the column-wise average of the medoid set matrix.

21. For a given pair of segments M_i and N_i with descriptors $\mathbf{J}(M_i)$ and $\mathbf{J}(N_i)$ respectively, a comparison is made by finding the Euclidean distance between the two descriptor vectors.

6.3 Review of the ProtoDock-2 Algorithm

The alternate version of our ProtoDock algorithm that aims to address the partial shape complementarity matching employs the suggestion made by Sun *et al* [96] in using the Heat Kernel Signatures for partial shape matching. The authors posit that, given an adequate time, the HKS for each point on the surface describes its neighbourhood sufficiently. We therefore present the following ProtoDock-2 algorithm as a partial shape matching method for addressing the docking problem. The main difference between the ProtoDock-1 and ProtoDock-2 algorithm is that, ProtoDock-2 does not explicitly segment the protein structure, but simply selects points on the surface of the object to be representative of the region to which they belong.

The ProtoDock-2 algorithm goes through these three (3) major phases. Firstly, we construct the Laplacian matrix and compute its eigenvalues and eigenvectors. Secondly, we obtain the Heat Kernel Signature for each point on the protein object. Finally, we select representative Heat Kernel Signatures and compare with other protein Heat Kernel Signatures for a match. A discussion of the three (3) phases of the ProtoDock-2 algorithm is given below.

6.3.1 Laplace Eigen Decomposition & HKS Description

The first two phases of the ProtoDock-2 algorithm follows similar steps as is found in the ProtoDock-1 algorithm (Section 6.2.1 and Section 6.2.2), where we begin by constructing

the Laplace matrix from a given object's graph G obtained from the underlying mesh M. However, unlike the ProtoDock-1 algorithm, no mesh segmentation is performed on the base mesh M. We therefore proceed to compute the Heat Kernel Signature for each vertex point v on the mesh M similar to Phase-3 (in Section 6.2.3) of the first ProtoDock algorithm. However, given the object mesh M is unsegmented and therefore remains a compact manifold, no (Dirichlet) boundary conditions have to be accounted for. We consequently have a set of Heat Kernel Signatures for each vertex point v a $\mathbf{p}(v) = (p_i(v), \ldots, p_{t_{max}}(v))^{\mathrm{T}}$ over the list of heating times $\{t\}$ from 1 to t_{max}.

6.3.2 Pointwise HKS sampling for Partial Comparison

With our pointwise Heat Kernel Signatures computed for each point v on the mesh M, our algorithm proceeds to perform a partial matching by sampling the Heat Kernel Signatures of the mesh M and comparing it with that from another target mesh N.

The selection of the sample points is motivated by the fact that the Heat Kernel Signature at a given point will sufficiently describe the region (neighbourhood) of which the point belongs to, thereby enabling a partial shape matching by comparing the HKS of points on different objects. Therefore, the setting of adequate time parameters and the selection of appropriate samples are key to the performance of the algorithm. To this effect, the selected sample points should sufficiently span the entire object in order to be representative of each part of the object as possible. We therefore consider three (3) sampling methods for selecting these 'representative' points on the object surface for the partial shape matching process, namely (I) uniform covering sampling, (II) random sampling, and (III) segment-based sampling.

The *uniform covering sampling*, as the name suggests, simply selects the points at a set uniform distance over the entire object surface. We achieve this by finding the

quotient of the total number of vertex points n and the given sample size s. An integer equivalence of this quotient is then obtained by finding the floor of the quotient. This value is finally used as the sampling interval over the entire mesh. We therefore define this *sample interval step* as,

$$s_{int} = \left\lfloor \frac{n}{s} \right\rfloor . \tag{6.14}$$

Random sampling performs a random selection of points using a random number generator. The selection of points here assumes no knowledge of the distribution of the vertex points over the mesh nor the uniformity of the mesh sizes. This, however, also presents a disadvantage of not being able to properly accommodate non-regular meshes of high surface curvature.

The *segment-based sampling* aims to utilize the advantages of being both random yet covering, such that, the object mesh is first partitioned into covering segments and then a random sampling is the carried out in each segment of the mesh. In so doing, we are able to adequately cover the entire object mesh and still refrain from assumptions of the distribution of points over the mesh. We note that this segment-based sampling doesn't present a major advantage when the underlying object mesh is adequately uniform.

We further point out that, for this ProtoDock-2 algorithm, improved computational times may be achieved if the computation of the Heat Kernel Signatures is done after the vertex points have been sampled, thereby removing the need to obtain the Heat Kernel Signatures for unused vertex points. Finally, for a given pair of sampled points on two meshes M and N, a complementary partial match is found by finding the Euclidean distance between the pairing Heat Kernel Signatures.

6.3.3 ProtoDock-2 Algorithm

We present our ProtoDock-1 Algorithm for deformable shape complementarity matching for addressing the protein docking problem.

Input :

- Triangulated three-dimensional object mesh M with n vertices.
- k number of segments

Procedure :

I *Phase 1 - Laplace Eigen Decomposition*

1. Construct the $n \times n$ symmetric adjacency matrix $W = (w_{ij})$, where w_{ij} are the symmetric edge weights of the mesh M.
2. Construct the $n \times n$ volume matrix V, which is a diagonal matrix where v_i is the sum of all edge weights w_ij directly connected to vertex v_i.
3. Form the stiffness matrix $A = V - W$.
4. Construct the mass matrix D, with elements d_i obtained using cotangent scheme (Other weighting systems may be used here).
5. Form the Laplacian $L = D^{-1}A$ (Like the prior step, choice of Laplacian type may be varied).
6. Solve the eigen decompostion problem $A\mathbf{f} = \lambda D\mathbf{f}$ the first n_{eig} smallest (or largest depending on the type of Laplacian) eigenvalues and corresponding eigenvectors of the Laplacian.

7. Obtain a normalized column-major eigenvector matrix E_{vec} by stacking the eigenvectors into an $n \times n_{eig}$ matrix and normalizing each row.

II *Phase 2 - Heat Kernel Signatures Sampling*

8. Compute the heating times list $\{t\}$ of time t_{int} intervals starting from a minimum time to a set t_{max}.

9. Obtain a sample set $S = v_i, \ldots, v_s$ of s vertices from the mesh M obtained by an appropriate sampling method.

10. Given the obtained eigenvalues λ_i, their corresponding eigenvectors φ_i, and the heating times list $\{t\}$, for each vertex point v in set S, compute the t_{max}-dimensional Heat Kernel Signatures $\mathbf{p}(v) = (p_i(v), \ldots, p_{t_{max}}(v))^{\mathrm{T}}$. Normalize each row vector.

11. Form an $s \times t_{max}$ HKS descriptor matrix $P(M)$ for the given mesh M by stacking each Heat Kernel Signature vector $\mathbf{p}(v)$ in row-major order.

III *Phase 3 - Pointwise Comparison*

12. For any given pair of object meshes with the HKS descriptor matrices $P(M)$ and $P(N)$, obtain a partial closeness measure by computing the Euclidean distance between each pairs of rows of each descriptor set considered as vectors in t_{max}-dimensional space. Smaller distance values denotes possible segment matching.

6.4 Summary

In this Chapter, we presented our algorithms for addressing the protein docking problem. The ProtoDock algorithm comes in two variations. The first and main version seeks to perform a partial shape complementarity matching by initially segmenting the underlying

protein object mesh using a spectral mesh segmentations approach. The Heat Kernel Signature (HKS) is subsequently computed for the obtained segments. A final descriptor vector using one of three different methods—Bag of Features (BoF), Closest Medoid Set (CMS), and Medoid Set Average (MSA)—is obtained from the Heat Kernel Signatures and used as the basis for the segment matching.

The second variation of our approach—ProtoDock 2—attempts to address the docking problem by a pointwise HKS descriptor matching. This is mainly motivated by the suggestion that, at adequate times, the Heat Kernel Signature of a point on a surface sufficiently describes its neighbourhood. Hence, the HKS of a point may serve as the representative descriptor of a given region of which it forms a part. For this purpose, we discuss three (3) sampling methods for selecting these points for the partial matching, each of which presents different advantages depending on the characteristics of the underlying object mesh.

In the next Chapter, we discuss the set-up and implementation of our proposed approaches for addressing the protein docking problem. We first give a description of the sources and formats of the data sets employed. We continue with the tools and platforms used in the development and implementation of our methods, and the necessary configurations required.

Chapter 7

Experimental Design

In Chapter 6, we presented our approach for addressing the protein docking problem. We discussed the two (2) versions of our proposed algorithms and the different phases and corresponding variations in parameters used in these phases. In the initial sections of this chapter, we discuss the two main experimental data sets employed for verification of our algorithms. The first preliminary data set serves as a benchmark for testing the quality and robustness of three-dimensional mesh shape descriptor and segmentation algorithms and their associated input parameters. The second is the data set used in addressing the protein docking problem. We further discuss, in the subsequent sections, the development platform and hardware employed for the implementation process.

7.1 The Datasets

There are two main datasets used in our implementation. The primary purpose of the first dataset, the *TOSCA dataset*, is to provide a test of quality and robustness of our shape descriptor and segmentation algorithm. The second dataset, the *Protein Solvent Accessible Surface (SAS) mesh dataset*, is the main data used to verify the viability of

our proposed approach for addressing the protein docking problem.

7.1.1 TOSCA Datasets

The *TOSCA (TOols for non-rigid Shape Comparison and Analysis) Project* [14] seeks to provide several resources and tools for the analysis and comparative study of three-dimensional mesh objects. The resources provided include software extensions, literature, and 3D object datasets. Two (2) main 3D mesh object datasets are provided for computational analysis, namely, the *SHREC'10 datasets* and the *TOSCA datasets*. The SHREC'10 (*Shape Retrieval Contest*) dataset includes three sub-classes of datasets which are used for the prime purpose of verifying the quality of object similarity and retrieval methods. The three sub-datasets provide a variation in the datasets for evaluating robustness, correspondence finding, and feature detection and description.

The TOSCA dataset, on the other hand, contains a set of different classes of non-rigid shapes in a variety of poses. The main usage of the dataset is for non-rigid shape similarity and correspondence analysis. The dataset further contains two subsets, namely, *TOSCA High-resolution* and *TOSCA Non-rigid World*. Although very similar, the main difference between these two subsets is in the vertex and face count of the three-dimensional object meshes. Also, with equal triangulation for object meshes in a given class, the TOSCA High-resolution dataset is very suitable for correspondence analysis, and has objects with average vertex counts of about 50,000. Conversely, the TOSCA Non-rigid World has object meshes averaging about 3,000 vertices.

The classes found within the TOSCA Non-rigid dataset are based on twelve (12) different objects, namely, *cat, dog, centaur, wolf, horse, seahorse, lion, gorilla, shark, female figure* ("*victoria*") and two different *male figures* (*i.e.*,"*david*" and "*michael*"). Each class contains several variations in the pose of the main class object. Figure 7.1

shows a few examples of the cat, dog and centaur classes with their first three corresponding pose variations found in each class.

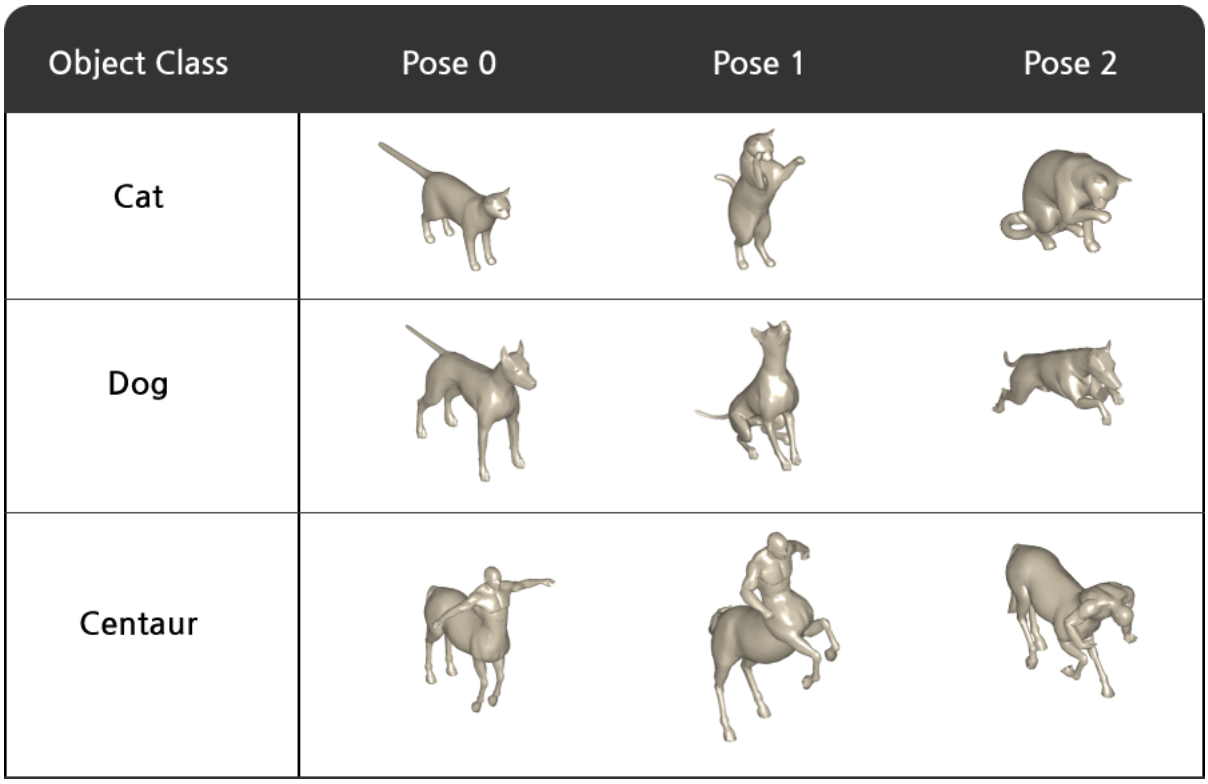


Figure 7.1: An example of the three classes from the TOSCA Non-rigid World dataset showing a few pose variations from the cat, dog, and centaur classes.

The shapes in each class are named in a *ClassIndex* format, where *class* is the class name of a given object, and *index* is an integer value (starting from 0 (zero)) assigned to a pose variation in that given class. For example, *centaur2* denotes a particular pose variation of the centaur class. Class objects with the index number 0 (zero) are ones with a symmetric pose and form the base objects from which the other pose variations are obtained. This is seen in Figure 7.1 where the first objects of each class, that is, *cat0, dog0* and *centaur0* are all symmetric in pose.

There are a total of 148 objects from all the classes in the TOSCA Non-rigid World

Table 7.1: Details of the TOSCA Non-rigid World classes and the number of objects in each class.

Object Class	**Number of object poses**
Cat	9
Dog	11
Horse	17
Wolf	3
Lion	15
Gorilla	21
Shark	1
Victoria (female figure)	24
David (male figure)	15
Michael (male figure)	20

dataset. Table 7.1 gives the details of the different classes and the subsequent number of different objects in each class.

The file types provided for the meshes in the TOSCA Non-rigid World dataset are in MATLAB (.mat) file format and in ASCII text files. The MATLAB file format provided is specifically suited for the MATLAB numerical computation software application. The more generic ASCII text files represents the object meshes in two text files, a .tri and a .vert text files. The .tri text file contains a 1-based tab-separated list of triangular faces, such that the first line corresponds to the triangle face with index number 1 in the objects mesh. This line has three integer values which each represent the vertex indexes which form the resultant triangular face. The .vert text file, likewise, contains a tab-separated 1-based list of vertex XYZ coordinates, such that the first line corresponds to the vertex point indexed at 1. The first, second and third real values on each row, respectively, corresponds to the the x, y, and z coordinates of the associated vertex point indexed at

that row/line number.

For the purposes of our implementation and experimentation, we select a subset of the TOSCA Non-rigid World objects. The selection is primarily motivated by picking a class of objects with at least five (5) objects or pose variations in its class for an adequately comprehensive and uniform comparison test. This results in the selection of the cat, dog, centaur, horse, gorilla, victoria, david and michael object classes.

7.1.2 Protein Datasets

As discussed in Section 2.2, there are several representations of protein objects used for different purposes in proteomics. These representation include, van de Waal atoms, ball and stick, Solvent-Accessible Surface (SAS), Solvent-Excluded Surface (SES), amongst others. For the purposes of addressing the protein-docking problem, the *Solvent-Accessible Surface (SAS)* and *Solvent-Excluded Surface (SES)* are often used. These representations are both based on surfaces obtained from the accessibility of a solvent with respect to the van de Waal surfaces of a protein object or generally, any given biomolecular structure. SAS measures the area accessible to a solvent while SES measures the converse. The SES are often computed by using variations of the "*rolling ball*" algorithm which employs the use of a sphere of a certain radius to probe along the surface of van de Waal atoms of a given protein structure [44].

The SAS is obtained from the trace formed from the center of the probe as it rolls along the van de Waal spheres. This can also be said to correspond to the surface obtained from merging extended sphere surfaces around each van de Waal atom, which have a radius equal to the radius of the van de Waal atom plus that of the probe. Hence, the SAS can be thought of as an expanded van de Waals surface.

On the other hand, the Solvent-Excluded Surface (SES), also known as the *molecular*

surface, is likewise obtained from the same probing method and is described as the closest point of the solvent probe when rolling along the surfaces of the van de Waal atoms. That is, the surface obtained from tracing the points of contact between the inward-facing surface of the probing solvent ball and the van de Waal atoms. The SES is composed of this contacting surface between the probe and the van de Waal atoms, and the reentrant surface formed by the probe when it is in contact with more than one atom. The surface obtained by SES is consequently smaller than that of the SES [44]. Figure 7.2 shows an illustration of both the SAS and SES obtained by probing the van de Waal atoms of a biomolecule.

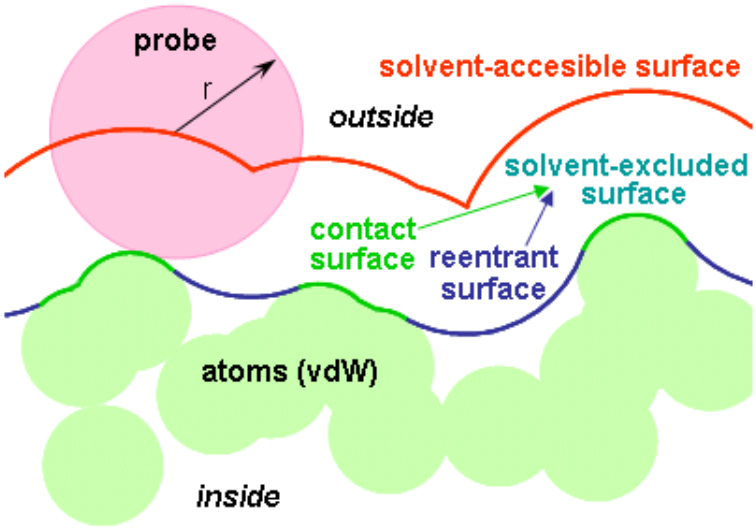


Figure 7.2: Illustration of the Surface-Accessible Area (SAS) and the Surface-Excluded Surface using the "rolling ball" method over the van der Waal atoms of an arbitrary biomolecule [44].

For our experimentation, and typically for the purposes of addressing the protein docking problem, the Solvent-Excluded Surface (SES) (molecular surface) representation is used for finding the active docking sites between corresponding receptor and ligand pairs.

The protein structures considered for our experiments are obtained from a benchmark of known protein docking pairs which have been obtained by laboratory experimentation. The benchmark is based the works of Hwang *et al* [42]— Protein-protein Docking Benchmark. The complexes in the benchmark have been curated over several years and is currently in Version 4.0, which contains a total of 176 receptor-ligand pairs. This is a 52 pair increment from the 124 pairs present in the prior Version 3.0 of the benchmark. The complexes considered for inclusion in this benchmark are high-resolution complex structures that are nonredundant at the family-family pair level. Two main criteria are set for the selection of the structures, a cutoff resolution of 3.25 Ångstrom (Å, a unit length 0.1 nanometers (nm) or 10^{-10} meters (m)), equivalent to that often used with X-ray structures, and minimum chain lengths of 30 residues.

Hwang *et al* [42], as with the previous version of the benchmark, further provide a classification of the newly added 52 complexes into three groups based on the measure of '*difficulty*' of predicting the pairings for docking algorithms. This level of difficulty is based on whether, and by how much, the active sites of the pairing proteins for docking undergo severe to no flexible deformations after pairing with each other. That is, the difficulty is measured by the structural difference between the bound and the unbound forms of the binding pairs. Rigid body conformations (i.e. the least difficult) are presented as having an I-RMSD $\leq$ 1.5Å, with Medium difficulty conformations having 1.5 Å $<$ I-RMSD $\leq$ 2.2Å, and finally difficult ones with I-RMSD $>$ 2.2Å.

I-RMSD is defined as the RMSD (Root Mean Square Distance/Deviation) between superimposed bound and unbound structures, calculated using the interface residue $C\alpha$ atoms of both binding proteins. Of the 52 newly added proteins, 33 cases present a least difficult scenario with a rigid body shape matching, 11 cases are of medium difficulty, and the final 8 are ranked as difficult cases. This brings the total of rigid body, medium

and difficult conformations for version 4 of the benchmark to 121, 30 and 25 respectively [42]. Also, the format of the names and structures provided in the benchmark are in line with that used in the Protein Data Bank (PDB) [81].

For our experimentation, we chose a set of *3 rigid body, 3 medium* and *4 difficult* conformations from the benchmark, as shown in Table 7.2.

Table 7.2: Protein-protein pairing data for protein docking experiments.

DIFFICULTY	PROTEIN PAIRING
Rigid Body	1K2I-1PMC
	3GMU-1ZG4
	3I1U-1ZFI
Medium	1IAM-1MQ9
	1QGV-1L2Z
	1R6C-2W9R
Difficult	1CL0-2TIR
	1J54-1SE7
	1ZM8-1J57
	1HUR-1R8M

7.1.2.1 Protein Data Sources

For each of the set of the chosen protein receptor and ligand pair from the Protein-protein Benchmark Version 4.0 to be used for our experimentation and evaluation, we obtain the source .pdb files from the Protein Data Bank repository [81] using their corresponding PDB Identifiers (ID). We consequently generate both the SAS and SES three-dimensional structural representations using a molecular solver or graphics software application. Examples of such molecular graphics applications include Avogadro, BALLView, Ascalaph Graphics, JMol, and VMD (Visual Molecular Dynamics). An expanded list of these computer applications can be found in [109].

We use BALLView version 1.4.1 [67] for generating both the SAS, SES and all other needed molecular structures employed in our implementation and experimentations. BALLView is a visualization application extension of the Biochemical Algorithms Library (BALL) Project, which provides an extensive collection of data structures, molecular mechanics, advanced solvation methods and molecular comparison and analysis among other capabilities. BALLView has several features for manipulating biomolecules, and uses OpenGL and the real-time ray tracer RTFact as render backends which is capable of offering both stereoscopic visualization in several different modes. It is computationally very effective and produces very stable three-dimensional structural representations. It is also cross-platform and is therefore available on all the major computer Operating Systems (OSs).

For the SAS and SES representations, the probe radius used for their construction is 1.4Å. This value is in line with popular practice in the literature, which suggests using a radius equal to that of a molecule of water [67]. Also, given only the vertex and polygon data forming the SAS and SES structures are needed, the choice of the three-dimensional file format for storing the SAS and SES structures is trivial, since no extra information, such as textures and light sources information, are needed. The generic and popular .OBJ (Wavefront Object) and .STL (STereoLithography) three-dimensional file formats are used. BALLView provides a numeric "*Resolution*" parameter when generating three-dimensional structures. This parameter generally determines the smoothness of the resultant structures, such that smoother representations consequently contain more vertices and faces than less smoother representations. Thus, for the SAS and SES models, smoother representations present a closer match to the prospective physical form, though computations performed on high resolution/smoother representations are more costly than their low resolution/coarser counterparts.

For our experimentation, we select two parameters for the Resolution, namely 3.5 and 0.5. The 3.5 value presents a default and adequately smooth representation of the structures. Our other chosen 0.5 value constructs the most coarse representation, which thus attempts to use the least possible number of mesh triangles. Figure 7.3 shows both the wireframe and solid models of the 3SSI SES representation at 3.5 and 0.5 resolution values. One of our main motivations for considering the coarse (0.5) representation is to experimentally determine whether such a coarse representation still provides adequate results (or possibly equivalent results) as compared to the default smooth representations. Competitive results for the coarse representation will prove very beneficial, since the computational costs associated with its minimal triangle/face count is a key advantage.

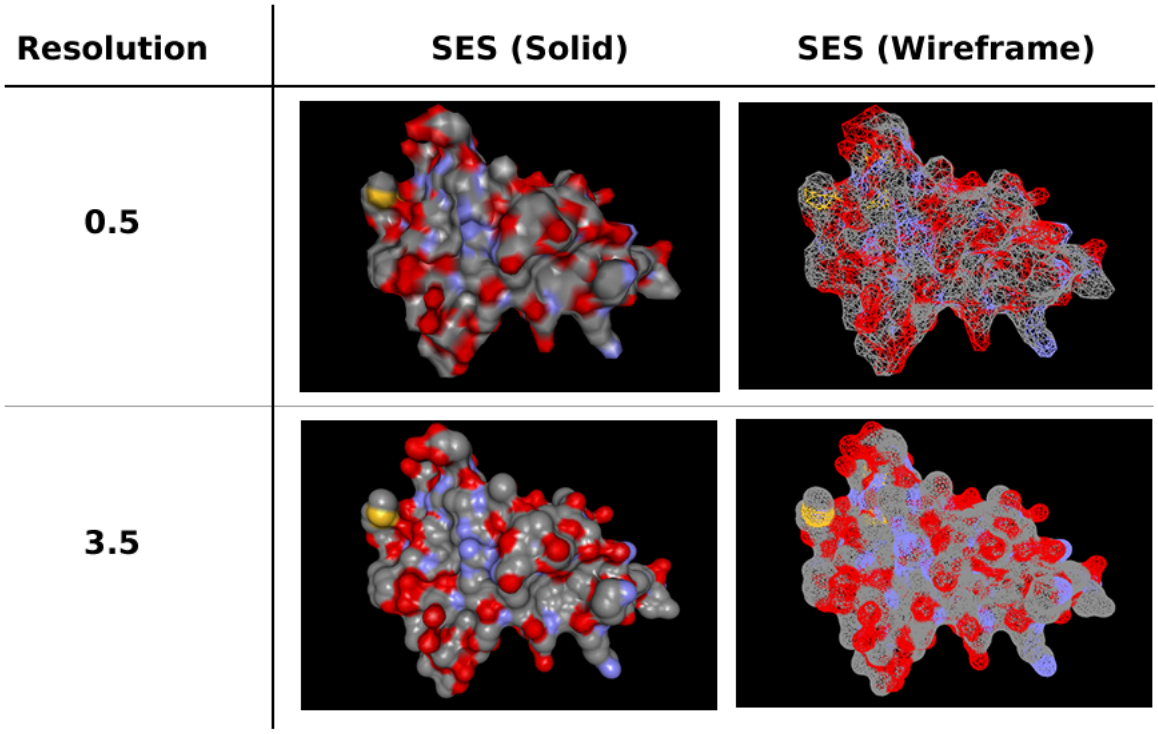


Figure 7.3: Shows the wireframe and solid models of the SES representation of the 3SSI protein with the BALLView resolution value set to 3.5 and 0.5.

7.2 Implementation Platform

For the implementation of both our ProtoDock algorithms, we use *Mathematica version 8.0* [112]. Mathematica is a computational software application that is used for engineering, scientific, mathematical and technical computing tasks. It provides several classes and packages for effectively manipulating mathematical objects such as matrices. It also integrates both input and display functionality into a single interface platform, thereby offering the capability of visualizing results without the need for third party data visualization applications.

7.2.1 Implementation Software

In line with the goals of our research, we consider developing a stand-alone application with a modular approach, such that, various aspects of the algorithm can be developed and later modified without much interference with the other sections of the entire application. Our application consists of two main parts—*script* files and *package* files. The script files provide a user with an interface for interacting with the application by running the application with a set of given parameters. The script files also provide various visualization forms from results obtained after running the application.

The package files, on the other hand, can be considered the back-end files. They contain the core and utility functions (or methods) which are called and executed from within the script files. The modularity of our application is mainly made manifest in the package files. In that, several parts of the ProtoDock algorithm are separated into various functions which can be sufficiently modified without too much interference to the entire application.

There are two main package files. The first, DescriptorIO.m, handles all inputs and outputs, such as, reading the three-dimensional .OBJ files of a given proteins SES rep-

resentation, or writing the descriptor values of a three-dimensional shape to an output file or database. The other package file, namely DescriptorHKS.m, caters to most of the functions needed for executing the ProtoDock algorithm itself. It contains functions for computing the required stiffness and mass matrices from a loaded triangulated mesh, computing the Laplacian and its corresponding eigensystem, and obtaining the medoids from a clustering process, amongst others. The DescriptorIO.m package file contains about *610* lines of code, while the DescriptorHKS.m has *1190* lines. Combined with about 32 script files which are responsible for various tasks of taking, computing and generating visualization for results, and each with an average of *200* lines of code, a total of about *8,840* lines of code make up our entire software package. Appendix **??** contains the list of functions and their descriptions found within the DescriptorIO.m and DescriptorHKS.m package files.

Also, for the purposes of portability of the application, we implement two methods for data storage of our shape matching system. The first uses a relational database management system (RDBMS) for storing data computed during the application run. The RDBMS employed here is MySQL—an open source cross-platform RDBMS that runs as a server providing multi-user access to multiple databases. The use of the approach requires some amount of configuration to be done on the implementation computer.

The second and more portable data storage method uses (text) files. This provides a very portable approach since there isnt any need for setting up and configuring an RDBMS. Text files containing data in a Mathematica readable format are generated on the fly during the application run, and are later read for further computations or to obtain final visualization results.

In order for the same script files and functions to be easily read, both from an RDBMS and text data files, we employ the use of what we call a '*name sequence*'. A name sequence

is simply a unique name generated for a particular instance of computing descriptor for a given target object. It is formed from concatenating the input parameters that were used in generating that particular descriptor. For example, Figure 7.4 shows the different parameters, including the number of eigenvectors, the type of Laplace-Beltrami operator and the size of the medoid set, that were used on constructing a descriptor given those parameters. The name sequence can be considered an *ID* for a given descriptor. In the file-based storage system, it is used to immediately identify a data text file. While in the RDBMS-based system, parts of the name sequence are used in database queries for retrieving specific objects. The hyphens (-) in the name sequences are used as delimiters.

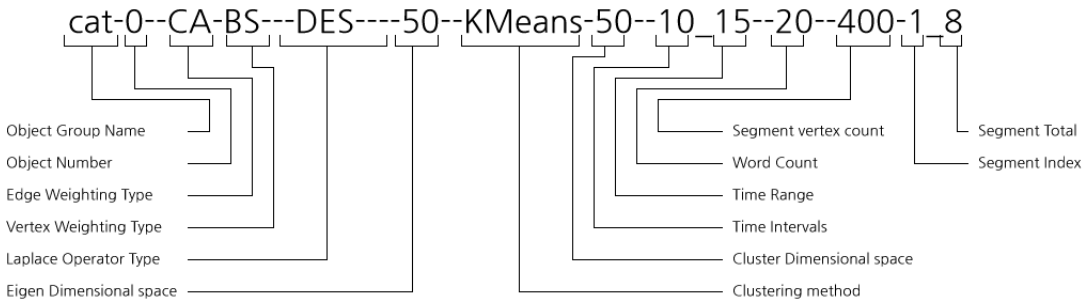


Figure 7.4: Shows the different parameters that are concatenated to make up an object's name sequence from the TOSCA dataset.

Figure 7.5 shows the database schema that we use in the RDMS-based storage version of our application software. The object table holds the base information of a target object mesh, such as the number of vertex points and triangles, and all the coordinate data associated with all the vertices of the mesh. The descriptor table contains the name sequences and their corresponding computed descriptor vectors. A descriptor ID, did,

serves as a cascading foreign key to the other tables, namely, bow, segment, laplace_eigen and hks which contain data values from computing the Bag of Features, mesh segmentation, Laplace eigen decomposition and the Heat Kernel Signatures, respectively.

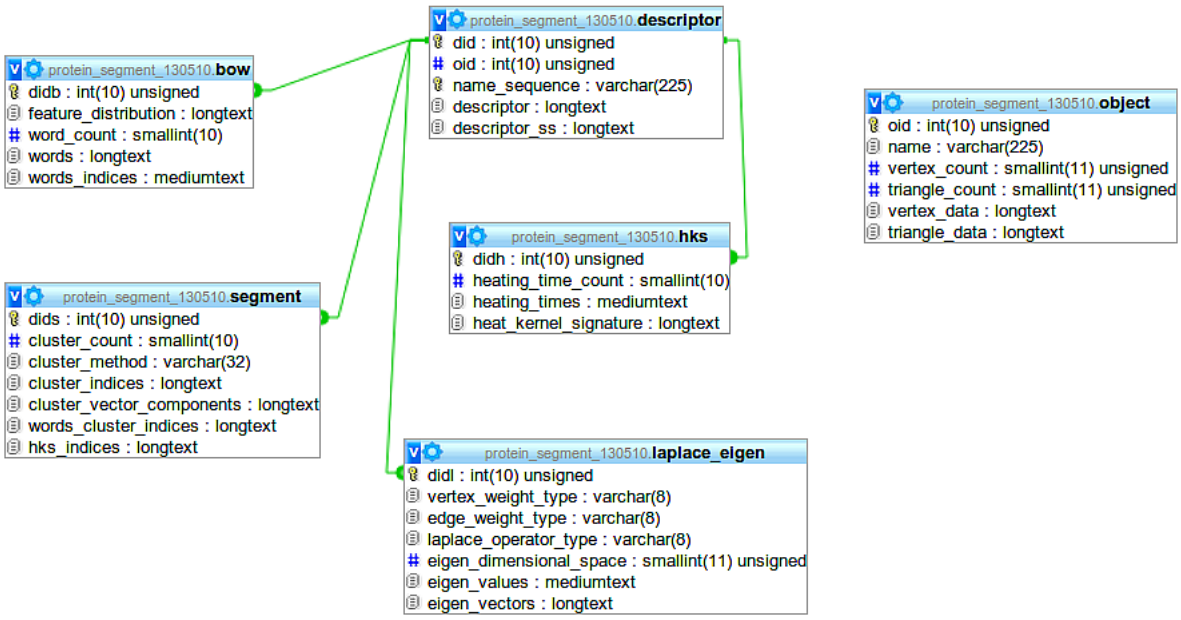


Figure 7.5: Shows the schema of the database used in the RDMS-based storage system.

7.2.2 Implementation Hardware

One of the motivating goals of the research is to present an adequately computationally inexpensive method for addressing the docking problem which can be run by inexpensive workstations or high-end desktops/laptops with reasonable processing resources, and not by extremely expensive highly powered workstations or servers. To this effect, we use an Intel Core i3 powered computer with 8GB RAM running a Ubuntu Linux Operating System.

7.3 Summary

In this Chapter, we presented the experimental setup and the details of the implementation procedures. We first described the experimental datasets used for both the preliminary evaluation of the ProtoDock algorithms and then for the main problem of protein-protein docking. We presented the sources of the protein data, the docking benchmark and the necessary parameters that were used in generating the appropriate protein structures. We finally discuss both the software and hardware platforms employed and reviewed both the RDBMS and file-based storage systems employed for the implementation of our shape matching application software system.

In the next Chapter, we present the evaluation methods and the subsequent results obtained from our experimentation. We first outline the employed evaluation criteria, which is mainly motivated by the prior discussed protein-protein docking benchmark. The remaining section presents the results of the experiments and an analysis of the observed results.

Chapter 8

Evaluation and Experimental Results

In Chapter 6, we presented several variations to our ProtoDock algorithm which presents a geometry-based deformable partial and whole shape matching technique for addressing the protein docking problem. Recall that, the core of the technique is based on the Heat Kernel Signature, which presents a multi-resolution description of an objects surface given by the rate of heat diffusion over that surface at multiple time intervals. Chapter 7 discussed the datasets and their sources along with the experimental setup and implementation of the ProtoDock algorithms.

In this Chapter, we provide the results of our experimentation of the ProtoDock algorithms as tested against receptor-ligand pairings in the Protein-Protein Docking Benchmark. We build up to the final protein docking results by first showing results of the different phases of the ProtoDock algorithm tested against the TOSCA deformable shape dataset. We begin with results from the segmentation phase as induced by our spectral minimum-cut approach. We then present results of our novel ProtoDock algorithm when used as both a deformable shape descriptor and a partial shape descriptor. For each

of these phases, several parameters are varied in order to analyse its effectiveness and efficiency at returning appropriate matches.

8.1 Evaluation Criteria

The main area of evaluation of the ProtoDock algorithms is in its phase of shape comparison and retrieval. Recall that, shape matching and retrieval presents the problem of being able to obtain a set of closest matching objects given a query object *i.e.*, an initial object of reference. Although the notion of "closeness" is rather subjective depending on the domain of application, given a well constructed dataset (such as the TOSCA benchmark dataset), the efficiency of retrieval of a given algorithm can be sufficiently measured. Hence, we discuss the following accepted performance measures that are often used for such purposes, particularly, precision and recall.

8.1.1 Performance Measures

Confusion Matrices are widely used in information retrieval systems. It aids in visualizing the output of a retrieval system, classifier or predictor. For a given prediction or classification experiment, the outcomes of the experiment can be constructed in a confusion matrix as shown in Figure 8.1,

In the context of shape retrieval, True Positive (TP) refers to an object or shape which is retrieved or classified as similar to the query object and is truly so. Likewise, True Negatives (TN) are non-matching objects which are predicted as not similar. On the other, False Positives (FP) are those objects retrieved as positive while not being so. And False Negatives are when matching objects are predicted as not similar to a query object.

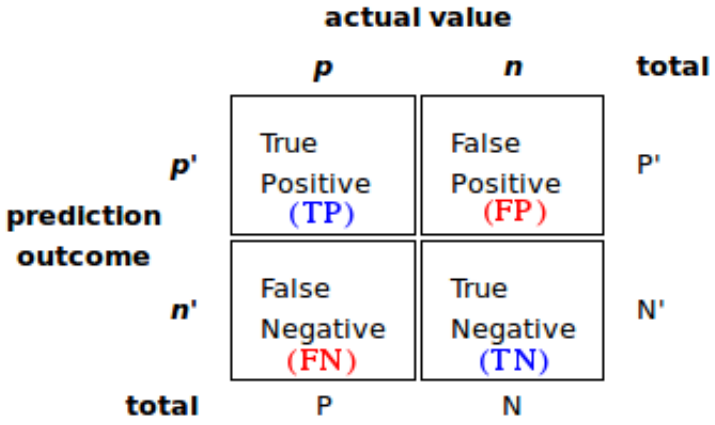


Figure 8.1: A confusion matrix of a generic prediction experiment [67].

Recall or *True Positive Rate (TPR)* or *Sensitivity* is one of the popular metrics for measuring the quality of a retrieval system. In the context of shape retrieval, recall is the fraction of objects retrieved that are truly relevant to the query object. This is given as,

$$Recall = \frac{TP}{TP + FN} \, . \tag{8.1}$$

Precision is defined as the fraction of all objects returned as similar that are indeed a close match to the query object, and is defined as,

$$Precision = \frac{TP}{TP + FP} \, . \tag{8.2}$$

Fall-out

or *False Positive Rate (FPR)*is the fraction of non-similar objects that were retrieved out of all the non-similar objects. This is defined by the formula,

$$Fallout = \frac{FP}{FP + TN} \, . \tag{8.3}$$

8.2 Results Evaluation

As a proof of concept we first test the different phases of our methods on the well formatted TOSCA datasets. We experiment with the two main phases, namely, mesh segmentation and deformable whole and partial shape matching.

8.2.1 Mesh Segmentation Results

The segmentation approach adopted in the ProtoDock algorithms uses a spectral-based method, where a minimum-cut of the underlying object mesh is obtained. This is achieved by first computing the eigensystem of the mesh's Laplace-Beltrami operator, and then clustering the eigenvectors using the K-Means algorithm. The subsequent clusters of mesh vertices obtained after the partitioning process corresponds to individual segments. Recollect from Chapter 4 that the construction of the Laplacian is subject to several parameter variations, mainly arising from the choice of edge and vertex weighting employed in computing the stiffness and mass matrices. Another parameter that contributes to the quality of segmentations obtained is the number of eigenvectors used. Finally, the k number of clusters also plays a key determinant in the "quality" of segmentation, or by how much the segmentation respects the notion of parts as perceived by humans (refer to the minima rule introduced in Chapter 3).

Figure 8.2 below shows results arising from variations in the influencing parameters for spectral mesh segmentation.

8.2.1.1 Discussion

From the segmentation results in Figure 8.2, we observe that spectral segmentation adequately partitions the objects along lines of concavity as perceived by humans. The segments for the *cat-pose-0* show all four legs appropriately segmented from the main

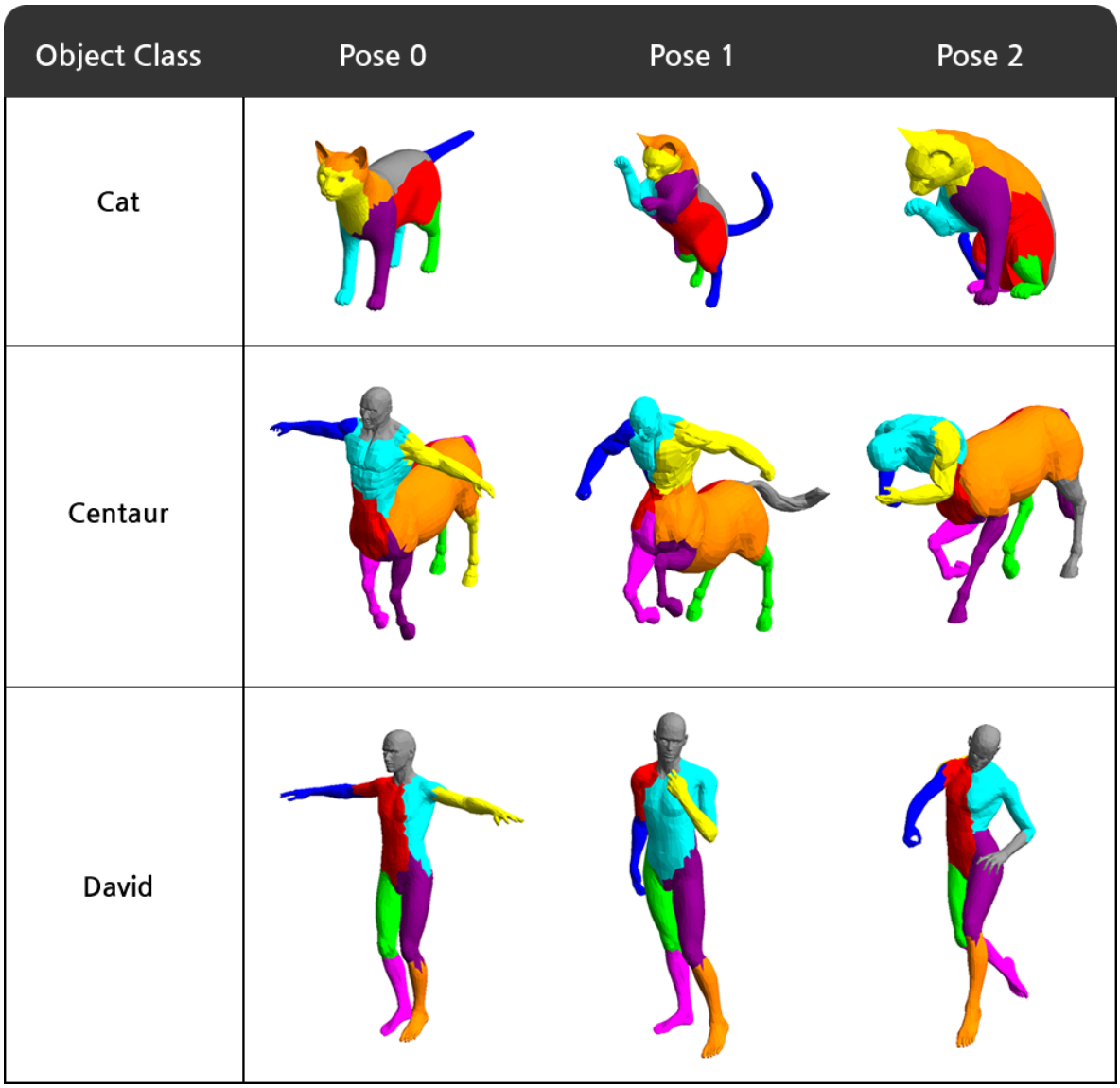


Figure 8.2: Spectral segmentation on Desbrun Laplace-Beltrami Operator, with the first 10 eigenvectors, and cluster number (k) corresponding to 9 segments, for the first three (3) poses of the a the cat, centaur and david TOSCA objects.

trunk of the body. Likewise, the protruding tail is also returned as a complete segment. However, a relatively unintuitive segmentation is produced around the head region, such that the front and rear parts of the head do not form a single segment. We notice that this slight anomaly is corrected in *cat-pose-2*, where the entire head is returned as

a complete segment. This trend of poor head segmentation is further observed in the *centaur* objects as well. However, the *david* poses consistently and correctly identifies each head as a single segment.

Another key point of interest is the segmentations' ability to remain sufficiently robust to changes in pose. Here, we observe near equivalent segmentations in the poses of all three *cat*, *centaur* and *david* objects. For example, for all three (3) poses of the *david* objects, the legs remain consistently segmented. This is also evident in two of the three poses for both the *cat* and the *centaur* objects.

Figure 8.3 shows the segmentations induced on the *cat* object with all other parameters kept constant, except the number of eigenvectors. From the segmentation results, it can clearly be seen that the eigenvectors play a key role in determining the quality of segmentation. For any given mesh M with n vertices, a maximum of n eigenvectors (with n corresponding eigenvalues) can be computed for its Laplacian. The vertex count for the *cat* objects are 3,400. However, we realize that far smaller eigenvector numbers, of less than 20, return much higher quality segmentations than the larger numbers. Thus, an increase in the number of eigenvectors drastically diminishes the quality.

Figure 8.4 shows the segmentations obtained on the symmetric pose of the *cat* object where the first 5, 10, 20, 50 and 1000 eigenvectors are chosen. A visual inspection of the plot of the eigenvalues associated with these eigenvectors does not immediately show any drastic change in the eigenvalues. We suggest that the contributing factor to the abrupt deterioration of the segmentation quality, with increasing eigenvectors, may well be as a result of factors that would require more analysis and experimentation.

Variation in Laplacian. Figure 8.5 presents the results of segmentations induced by four (4) other variations of the Laplacian 4, namely, the Graph Laplacian (GL), Ng, Jordan and Weiss (NJW), and Linear Finite Element Method (LFEM). We observe that,

the produced segments shows similar quality across all the types of the Laplacian. With GL and NJW being variations of the graph Laplacians, they both show near identical

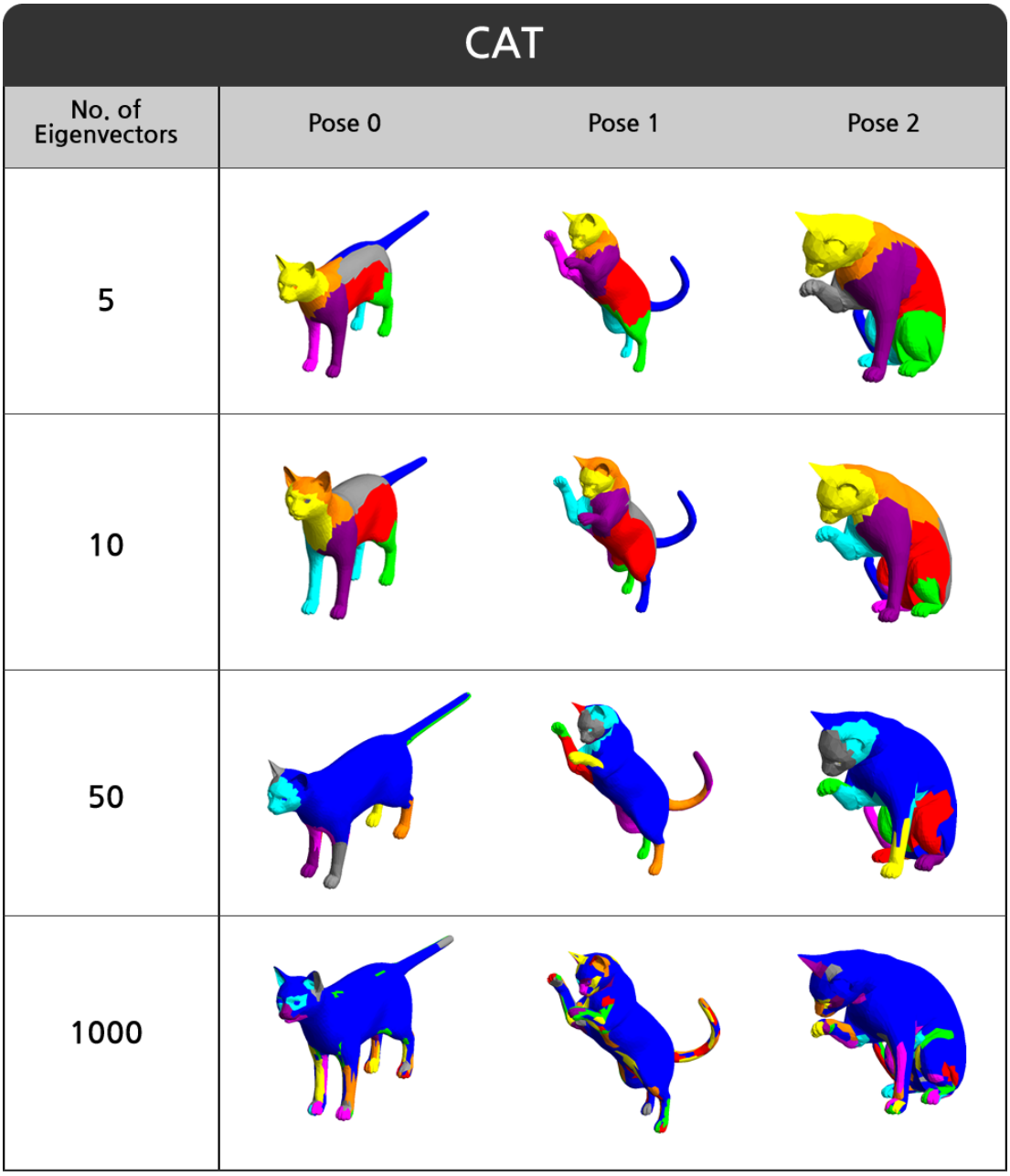


Figure 8.3: Induced segmentation on the cat object with an increasing number of eigenvectors from, 5, 10, 50 to 1000. The quality of segmentation depreciates with the rising eigenvectors.

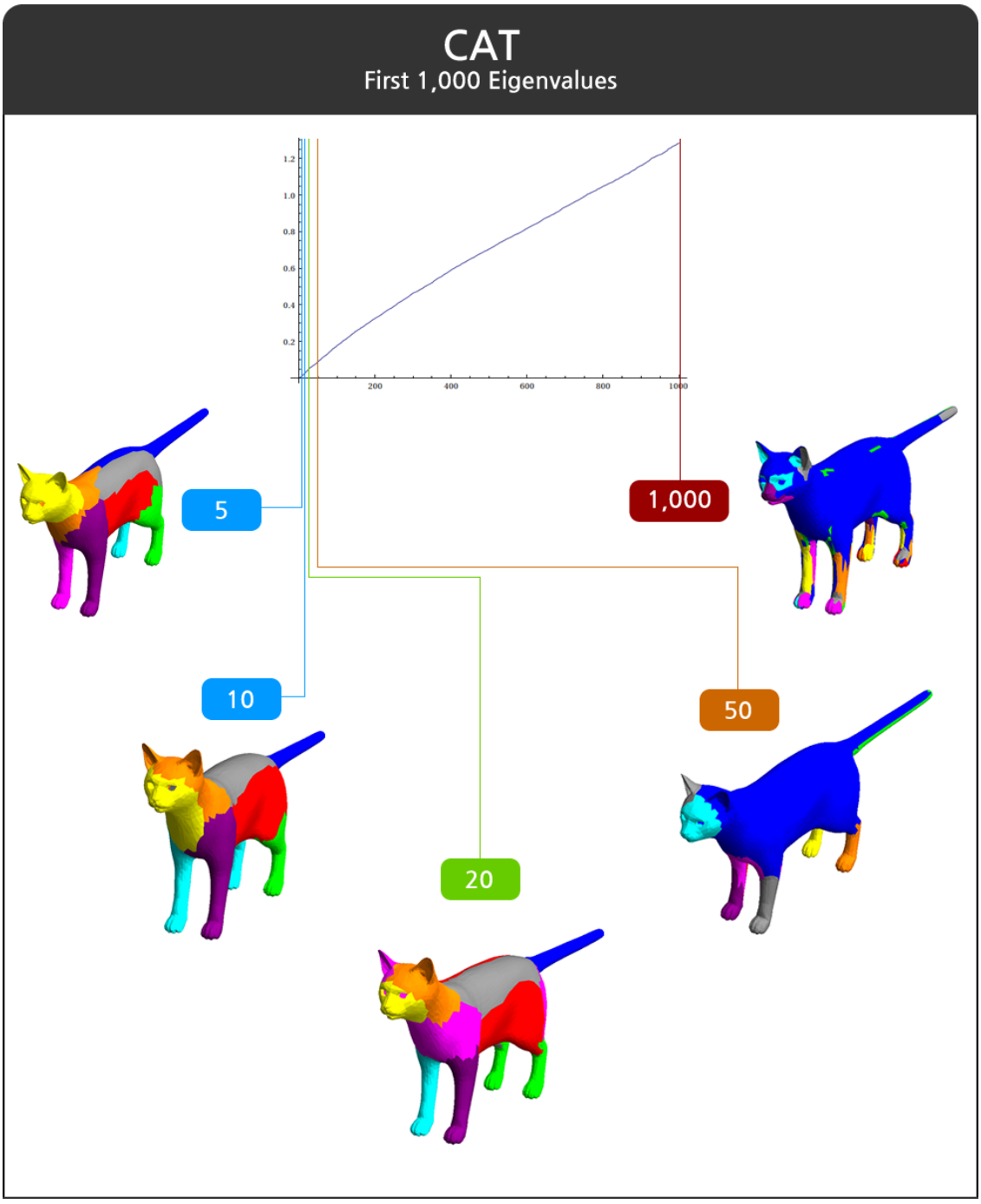


Figure 8.4: Induced segmentation on the cat object with an increasing number of eigenvectors from, 5, 10, 20, 50 to 1000 and an corresponding plot of its associated eigenvalues.

partitions using the 5th and 8th eigenvectors. Additionally, we recognize the persistent trend of poorer segmentations when the number of eigenvectors increases (although the NJW Laplacian, unlike the others, uses the largest eigenvectors and not the first smallest eigenvectors). This suggests that, the quality of segmentation is not greatly affected by the choice of Laplacian, but rather primarily by the number of eigenvectors chosen.

Nodal Set Segmentation. We present preliminary results which suggest that the spectral segmentations induced by nodal sets isn't suitable for our current application for object mesh partitioning. Figure 8.6 displays the segments produced on the *cat* object when the 2nd, 3rd, 9th and 14th column of the *cat*'s Laplacian eigenvector matrix (obtained by stacking the eigenvectors in row-major order). The first and main drawback of nodal set segmentation arises from the fact that, although the number of segments returned is only guaranteed to be always equal to or less than the nth column of the eigenvectors matrix, it is usually far less for large n's. That is, both the 2nd and 3rd eigenvector columns appropriately return 2 and 3 segments, respectively. However, using the 9th and 14th columns of the eigenvectors matrix induce only 4 and 5 segments, respectively. A second observed drawback is its inability to adequately partition along perceived lines of concavity, when applied to such low resolution object meshes.

8.2.2 TOSCA Shape Retrieval Results

Recall that the next key phase of our proposed ProtoDock algorithm requires that we describe the segments obtained using an approach based on the Heat Kernel Signature. As with the segmentation phase, several parameters affect the performance of the final shape descriptor when used in a retrieval system, namely, the choice of Laplace-Beltrami Operator and its corresponding number of the eigensystem employed, the choice of the maximum time parameter t_{max}, and the vocabulary size (number of medoids) when

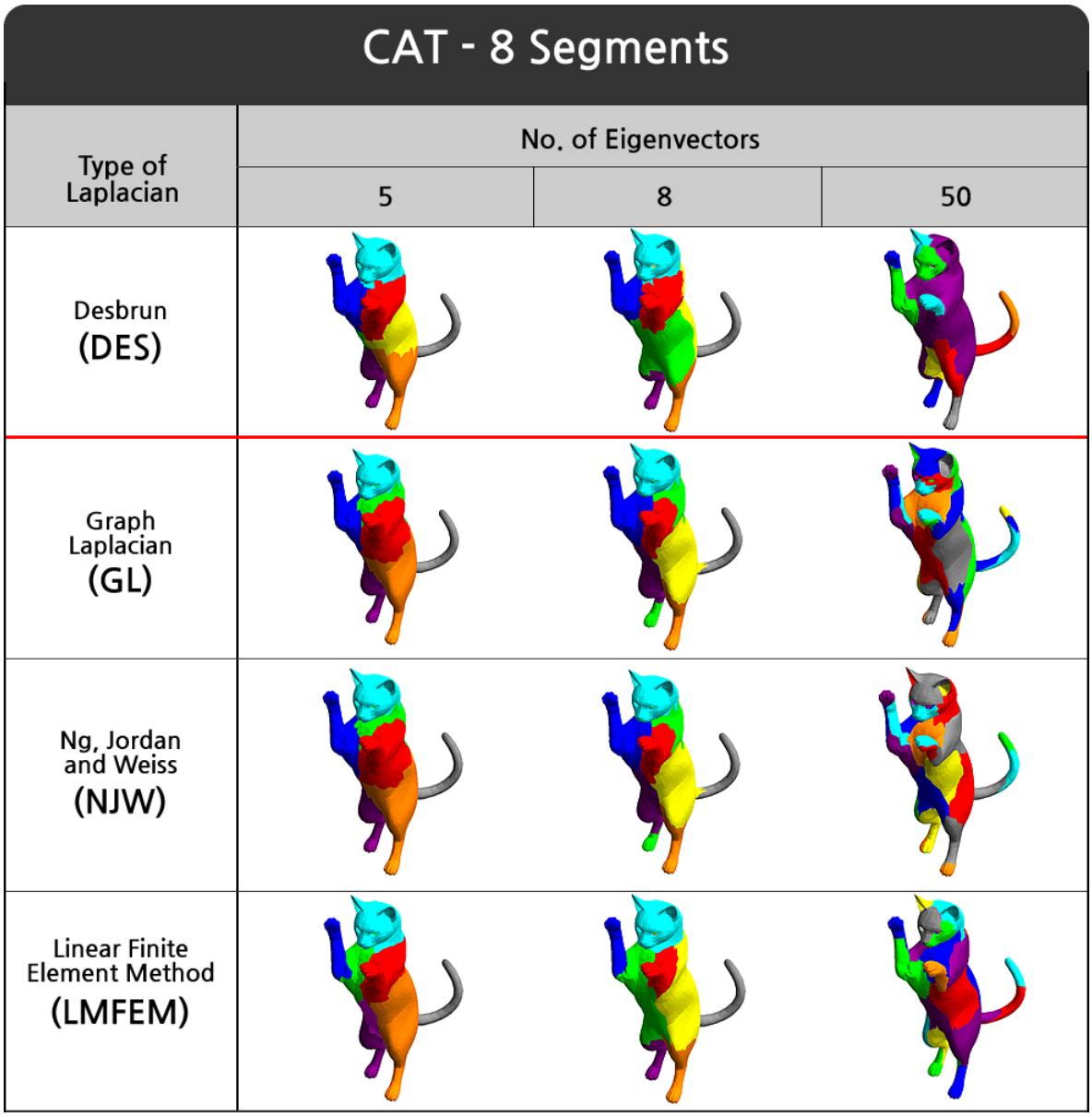


Figure 8.5: Similar segmentation produced by different Laplace-Beltrami operators—Desbrun (DES), Graph Laplacian(GL), Ng, Jordan and Weiss(NJW), and Linear Finite Element Method (LFEM). The number of eigenvectors were varied between 5, 8 and 50 in partitioning into 8 segments.

clustering the HKS. We present some of the better retrieval results obtained for each of the three descriptor methods, namely, the Bag of Features (BoF), our two methods,

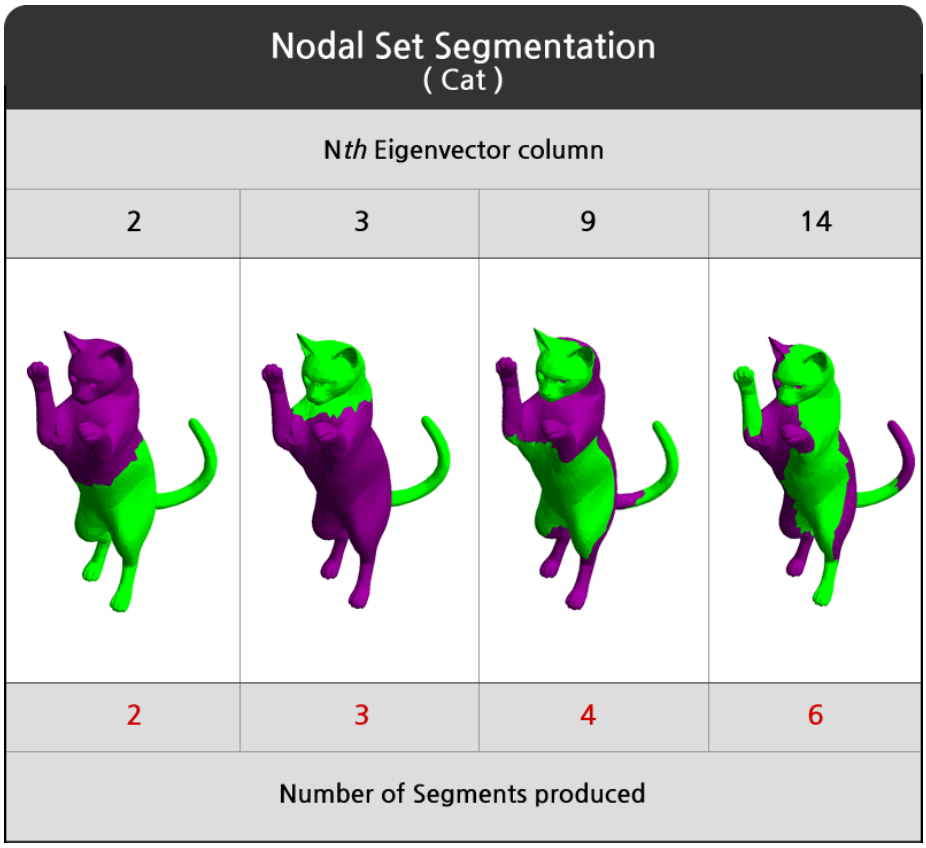
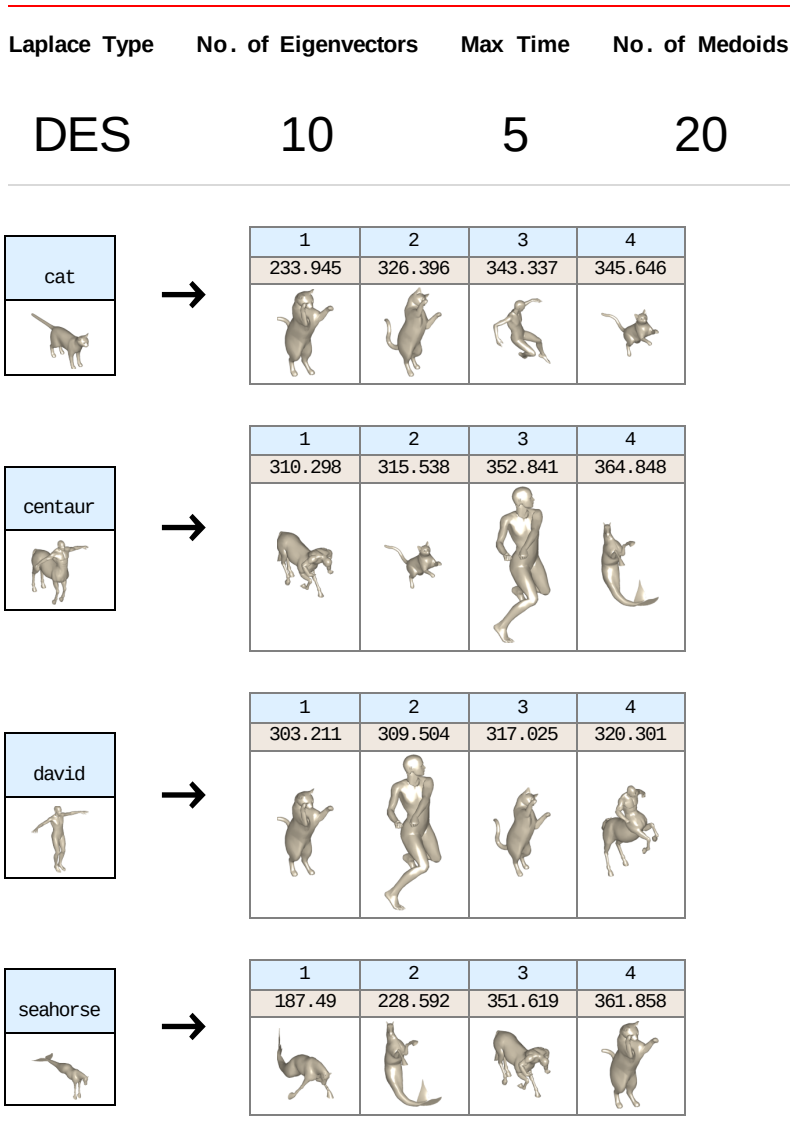


Figure 8.6: Nodal set segmentation of the cat object using the 2nd, 3rd, 9th and 14th eigenvector column.

Medoid Set Average (MSA) and Closest Medoid Set (CMS).

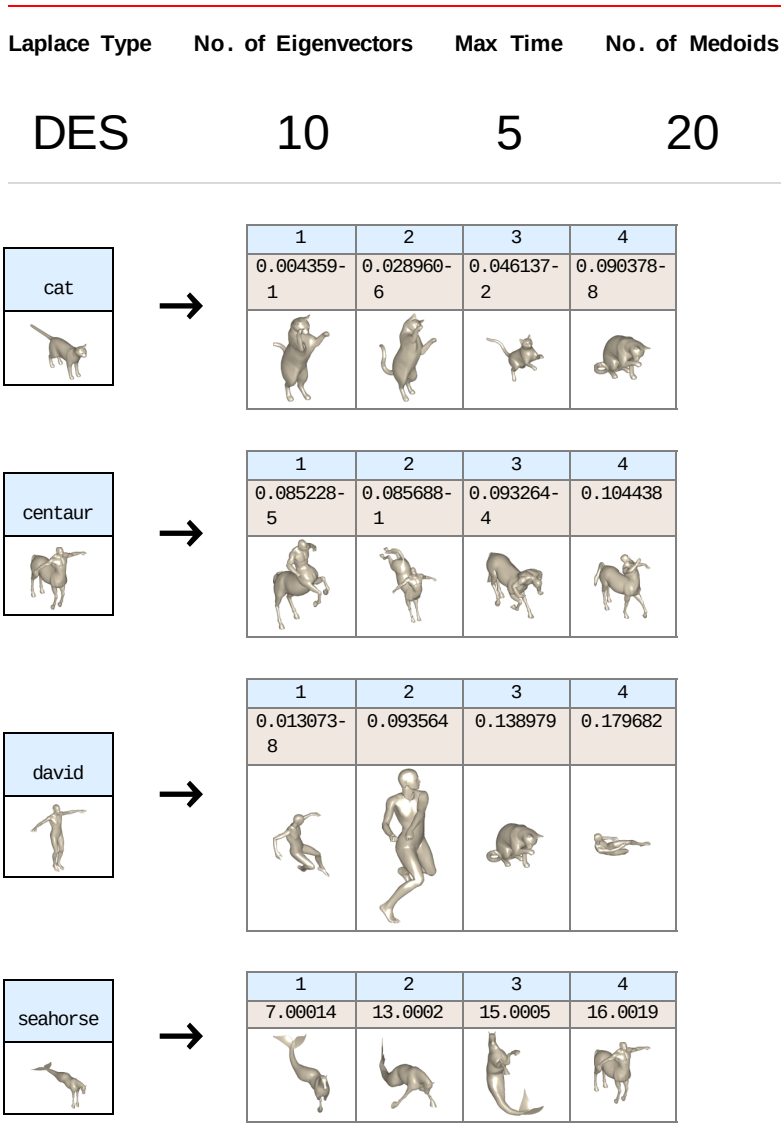
We first show three outputs (for each of the three descriptor methods BoF, MSA, CMS) obtained from a run of our shape matching application software. We display the input parameters that are provided by the user at runtime, a visualization of each base object, and a ranked list of the closest matching targets along with the computed distance measures. Figures 8.7, 8.8 and 8.9 present the outputs for the BoF, MSA, and CMS methods, respectively.

We present a line graph of the cumulative accuracy obtained by running the deformable shape matching algorithm with the following parameters: Maximum time at



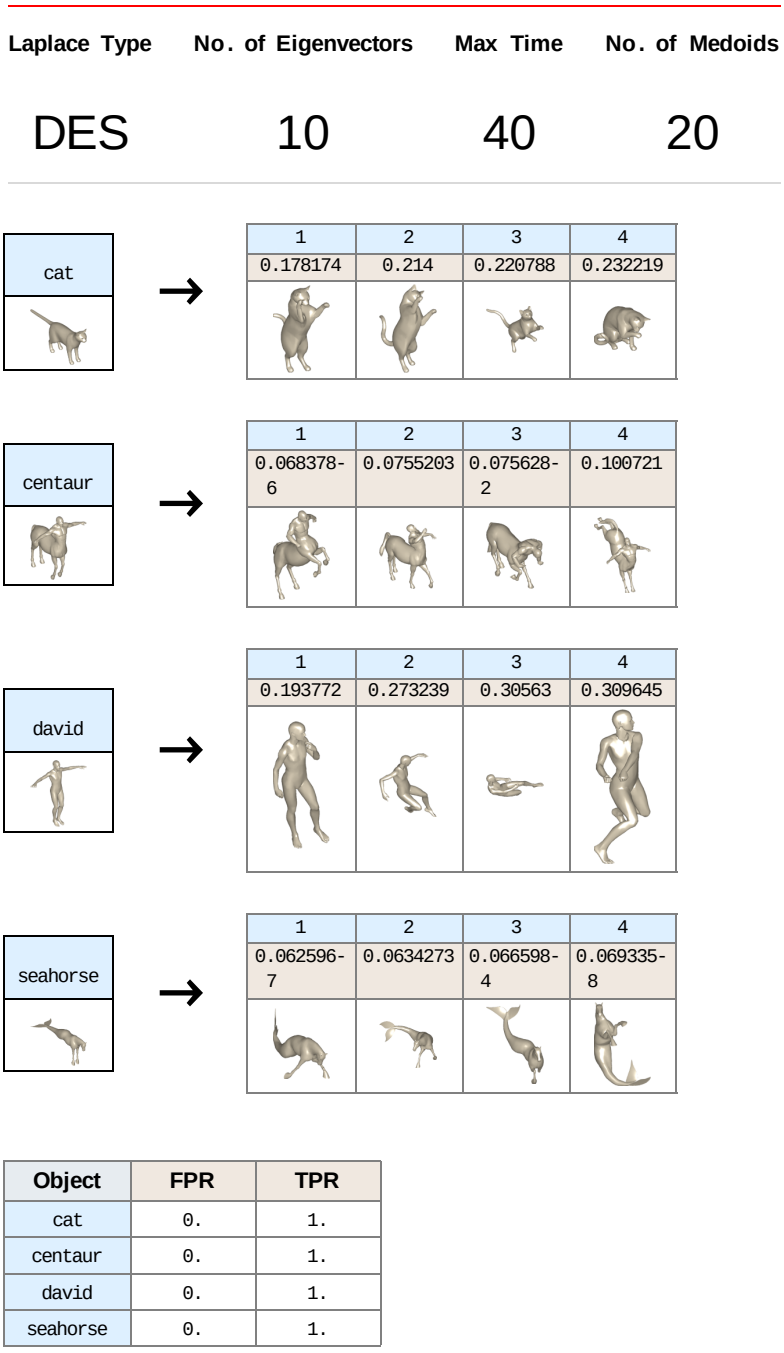
Object	FPR	TPR
cat	0.0666667	0.75
centaur	0.2	0.25
david	0.2	0.25
seahorse	0.133333	0.5

Figure 8.7: Sample visualization output for a deformable shape matching experimental run for the **Bag of Features (BoF)** method.



Object	FPR	TPR
cat	0.	1.
centaur	0.	1.
david	0.0666667	0.75
seahorse	0.0666667	0.75

Figure 8.8: Sample visualization output for a deformable shape matching experimental run for the **Medoid Set Average (MSA)** method.



Object	FPR	TPR
cat	0.	1.
centaur	0.	1.
david	0.	1.
seahorse	0.	1.

Figure 8.9: Sample visualization output for a deformable shape matching experimental run for the **Closest Medoid Set (CMS)** method.

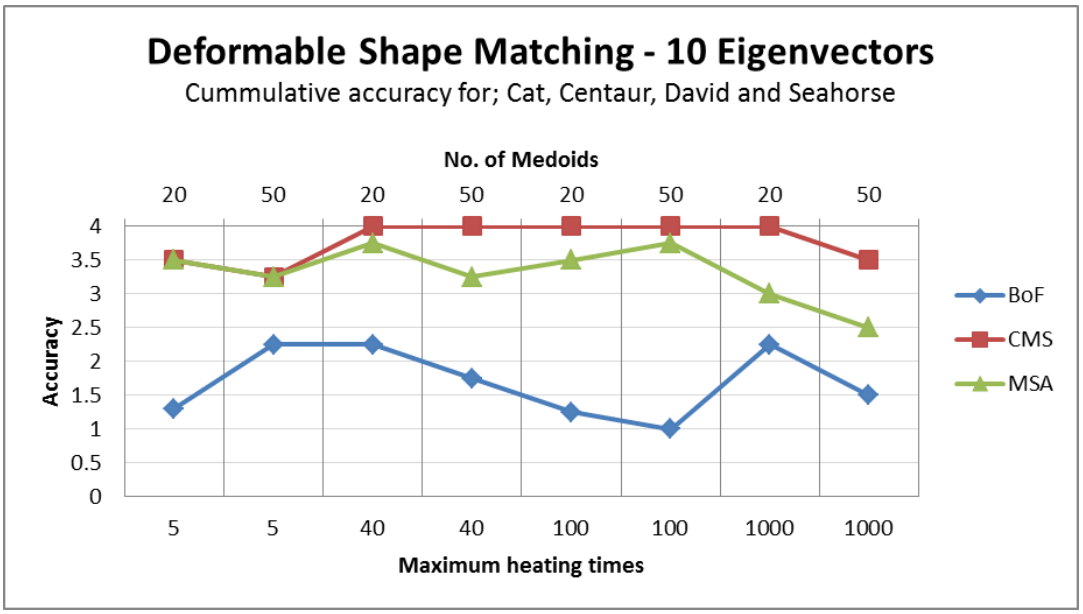


Figure 8.10: Shows a point line graph of the cumulative accuracy of the three descriptor vectors BoF, MSA, and CMS when used in our deformable shape retrieval system. All four object groups of cat, centaur, david and seahorse, each with 5 poses use the first 10 eigenvectors for the construction of the Laplacian, and vary the Maximum heat time at 5, 40, 100 and 1000.

5, 40, 100, and 1000, each with a medoid set size (or number of medoids) of 20 and 50. These values were set by experimentation. An accuracy of 4 denotes a perfect match for each object class. Figure 8.11 shows the results when five (5) poses, each from the *cat, centaur, david* and *seahorse*, are used as input objects. This gives a total of 19 target objects to which a given base query object, from each object class, will be compared. Furthermore, the first 10 eigenvectors of the Laplacian were used when computing the results found in Figure 8.10, while Figure 8.11 used the first 100 eigenvectors.

8.2.2.1 Discussion

From the results in Figure 8.10 and Figure 8.11, the Closest Medoid Set (CMS) descriptor method generally outperforms both the Bag of Features (BoF) and Medoid Set Average

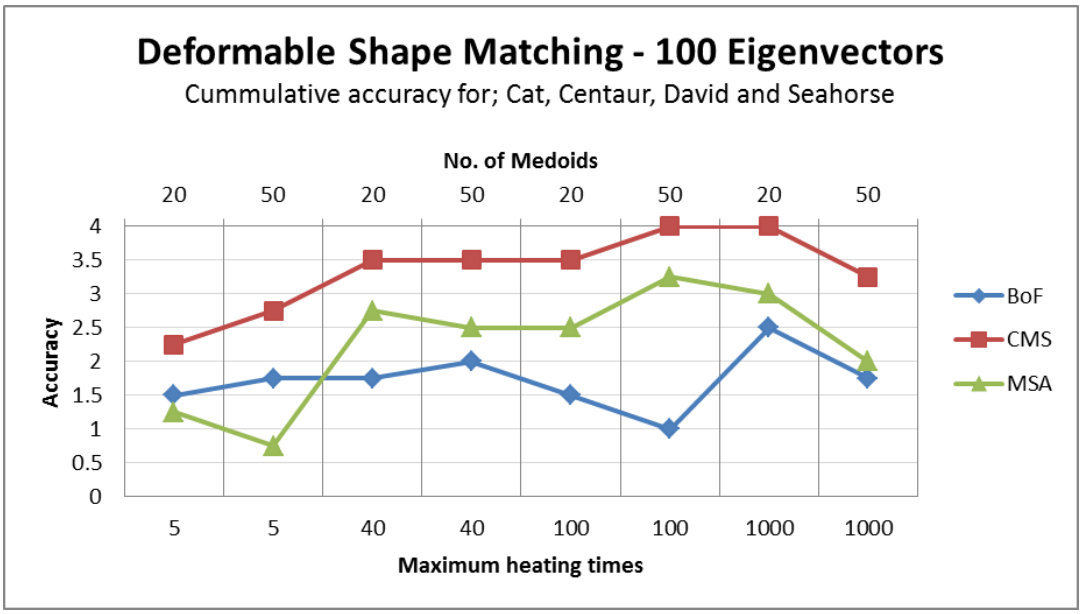


Figure 8.11: Shows a point line graph of the cumulative accuracy of the three descriptor vectors BoF, MSA, and CMS when used in our deformable shape retrieval system. All four object groups of cat, centaur, david and seahorse, each with 5 poses use the first 100 eigenvectors for the construction of the Laplacian, and vary the Maximum heat time at 5, 40, 100 and 1000.

(MSA) approaches. From Figure 8.10, we observe the CMS method indeed is perfectly accurate in matching the other 4 poses of each of the object classes at maximum heat time of 40 and 100 (except for the last comparison which only falsely classifies one wrong object in one of the object classes). Also, the Bag of Features method performs the poorest of the three. For the smaller maximum heat times of 5 and 40, the BoF returns relatively better matches with bigger medoid set sizes, but returns poorer accuracy at the larger times of 100 and 1000.

Figure 8.11 shows relatively poorer matching results when the Laplacian is constructed with the first 100 eigenvectors, instead of the first 10 (as is the case in Figure 8.10). This general trend may be as a result of the more global details that is captured, and its increased sensitivity to noise at larger eigenvectors. Furthermore, the

trend of larger medoid sets returning relatively poorer matches continues here for the BoF and MSA.

We further observe that, for both sets of results, all the descriptor vectors drop in accuracy at maximum time of 1000, with medoid set size of 50. This may be because of the HKS loosing local information due to the considerably high heating time, which allows the HKS to cover too large an area of the object mesh. This is further worsened by an even larger number of representative vectors, such that the HKS of these medoids unduly overlap.

Medoid Distribution. Recall that all three descriptor methods employ the medoids (representative vectors) of the clustered Heat Kernel Signatures of a mesh's vector points as the basis for deriving their final compact descriptor vectors. Figure 8.12 illustrates the positions of the twenty (20) medoids on the *centaur-1* object computed using the first 100 eigenvectors at a maximum time of 100. We observe an appropriate distribution of the medoids over the entire object mesh. The unique regions, particularly the head and tail, also have medoids present which are adequately representative of those regions. Likewise, both arms also have medoids present which serve to describe those areas of the *centaur* structure. Such induced distributions of the medoids is one of the key factors that enables using the medoids as the grounds for three-dimensional shape matching.

8.2.3 TOSCA Partial Shape Matching Results

Using the TOSCA dataset as a test case, Table 8.1 presents the experimental results of our ProtoDock-2 algorithm. Recollect that this variant of the ProtoDock algorithm aims to perform a partial shape matching by comparing the Heat Kernel Signatures of samples vector points on target object meshes. Recall that we presented three (3) sampling options, namely, Uniform Sampling (US), Random Sampling (RS), and our proposed

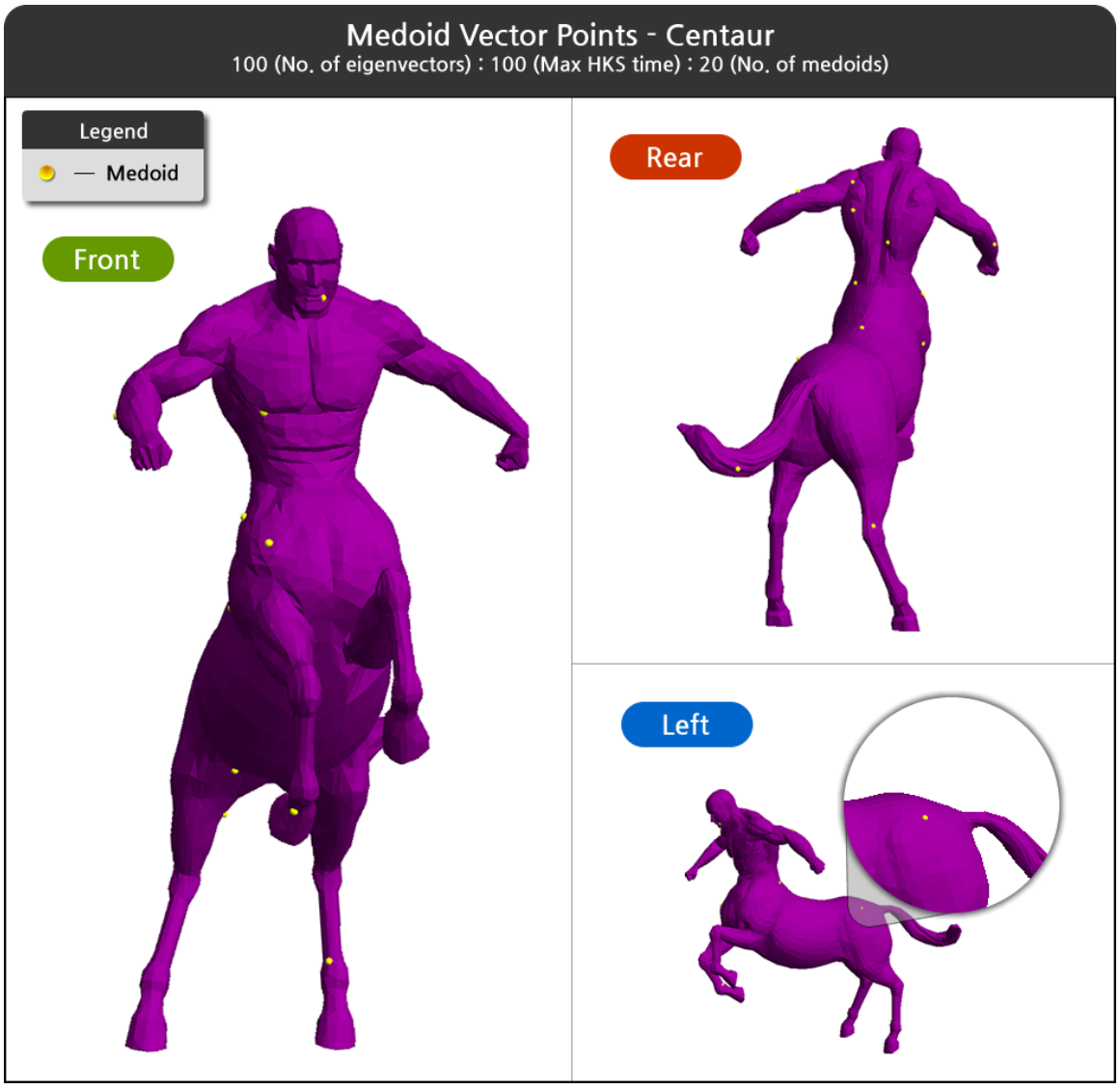


Figure 8.12: Illustration of the adequately distributed medoids of the *centaur-1* object over the entire mesh. Observe the appropriate distribution of the medoids (representative vectors) over the entire object mesh.

Segment-based Random Sampling (SRS). The Uniform Sampling method simply selects the samples uniformly from the mesh vertex indices. The Random Sampling approach uses a random number generator for selecting the sample points. The Segment-based Random Sampling technique aims to achieve random selections from the vector indices after they have been partitioned by a (spectral) segmentation algorithm.

Table 8.1: Estimated Accuracy Results for Partial Matching using Proto-Dock-2 on TOSCA Dataset

NO. OF SAMPLES	MAXIMUM TIME	US	RS	SRS
20	5	1.7	2.42	2
200	5	2.7	2.98	2.8
20	40	1.3	2.52	2.3
200	40	2.4	3.32	2.8
20	100	1.5	2.92	2.4
200	100	2.1	3.36	2.9
20	1000	2.2	2.9	3.1
200	1000	3	3.12	3.3

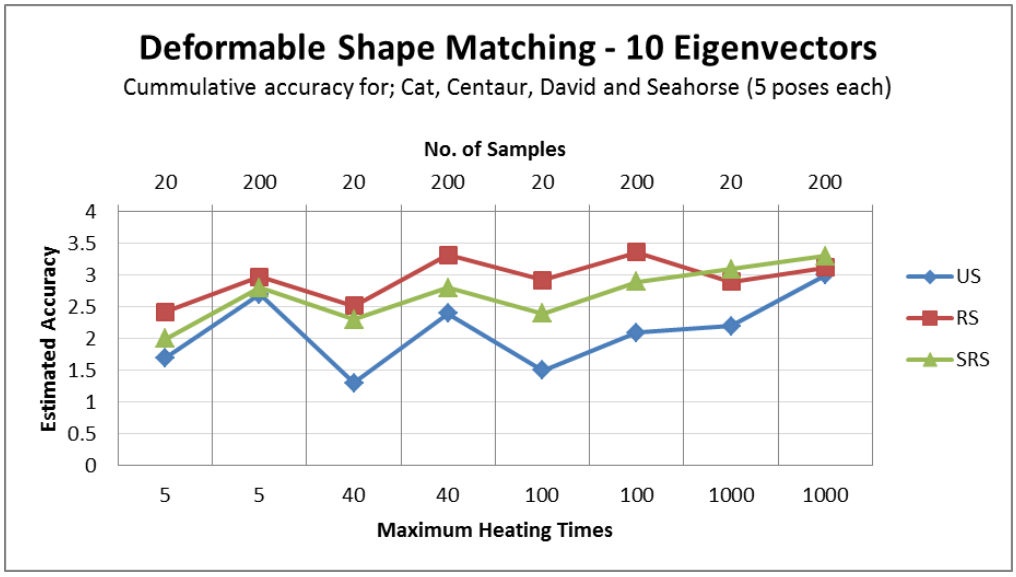


Figure 8.13: A point line graph showing the estimated accuracy of the ProtoDock-2 partial shape matching algorithm on the first five (5) poses of the TOSCA *cat, centaur, david* and *seahorse* objects, evaluated using the Uniform Sampling (US), Random Sampling (RS) and our proposed Segment-based Random Sampling (SRS) methods. The first 10 eigenvectors were used in computing the HKS.

Table 8.1 (with its corresponding point line graph as Figure 8.13) presents the results of ProtoDock-2 algorithm using the *cat, horse, david* and *seahorse* TOSCA objects as its input data. A total of five (5) poses were considered from each object class. For each of the varying poses, *maximum heating times* of 5, 40, 100, and 1000 were chosen, while

the sample sizes were set to 20 and 200. Hence, for the sample sizes of 20 and 200, a total of 7,600 and 760,000 pointwise HKS comparisons were made, respectively. For both the RS and SRS methods, the average of five (5) runs was recorded. These values were selected by experimentation.

An accuracy of 4 denotes a perfect match for each object class. However, the '*accuracy*' measure used for the evaluation of the three sampling methods is not a completely sufficient technique for assessing the methods, as a result of the dataset used. That is, objects in the different object classes of the TOSCA dataset may have regions of close similarity, and hence, points lying within those regions could have similar Heat Kernel Signatures. The *accuracy* measure may therefore unduly penalise the matching results in some cases. Figure 8.14 provides an illustration of two regions (trunk and feet) of the *centaur* and the *seahorse* objects which are most likely to return very close matches. We observe that the trunks of both *centaur* and *seahorse* have very similar surface geometry and will both contain vector points of very identical Heat Kernel Signatures. We reserve further exploration of the partial matching methods on more specialized datasets for future work.

8.2.4 Protein-Protein matching results

Having evaluated and discussed the underlying concepts of the phases of our ProtoDock algorithm, we proceed to the results obtained from applying it to the protein-protein docking problem. With the Protein-Protein Docking Benchmark 4.0, and similar to other geometric-matching based methods, we return the closest matching pairs between the segments of a given base receptor protein and the segments from other ligands. Also, as discussed before, we consider complexes from all the three (3) difficulty levels from the benchmark. We select the first 100 eigenvalues and their associate eigenvectors. A

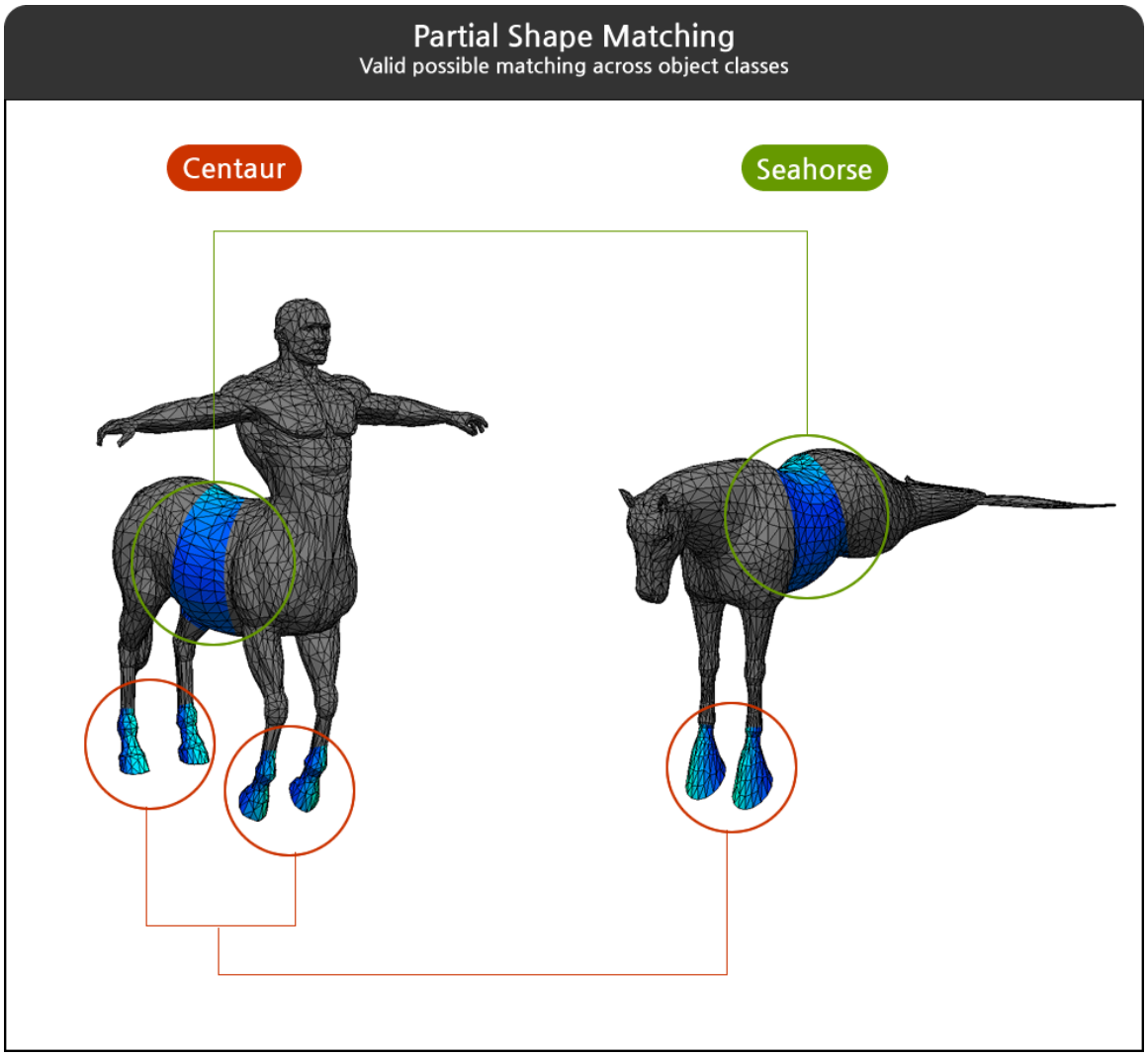


Figure 8.14: Illustrates two (2) regions of the Centaur and Seahorse objects which may be returned as valid possible partial matches.

time interval of 5 steps with a maximum time of 100 is chosen in computing the Heat Kernel Signatures. Finally, a medoid set size of 50 and 80 are chosen for the segment mesh size of 3,500 and 5,000 vertices, respectively, for the object segmentation. The above-mentioned parameters were set by experimentation.

As a visual aid, we again show an output a run of our application software for the protein-protein matching problem. We render each matching segment unto their cor-

responding entire protein structure. We point out that this rendered output allows an end-user to interact with the models by performing general 3D actions such as rotating, translating, and so on. Figure 8.15, 8.17 and 8.18 show the rendered output obtained from the correct pairings of 1IAM-1MQ9, 3GMU-1ZG4, 1ZM8-2TIR and 1QGV-1L2Z, respectively.

Tables 8.2, 8.3 and 8.4 show the ranking of the closest matching pairs obtained from our experimentation. The segments sizes compared here have an average vertex count of 3,500, 4,000 and 5,000, respectively. Also, the descriptor methods (BoF, CMS and MSA) are employed here in evaluating the matching pairs. For each ranking result of attempting to pair receptor-ligands segments, we further show the number of segment comparisons that were performed. The number of segment comparisons ranges from 512 to 1026. Recall from Section 6.2.4 that, the distance measure used for computing the similarity is the Euclidean distance. Figures 8.19, 8.20 and 8.21 show the corresponding

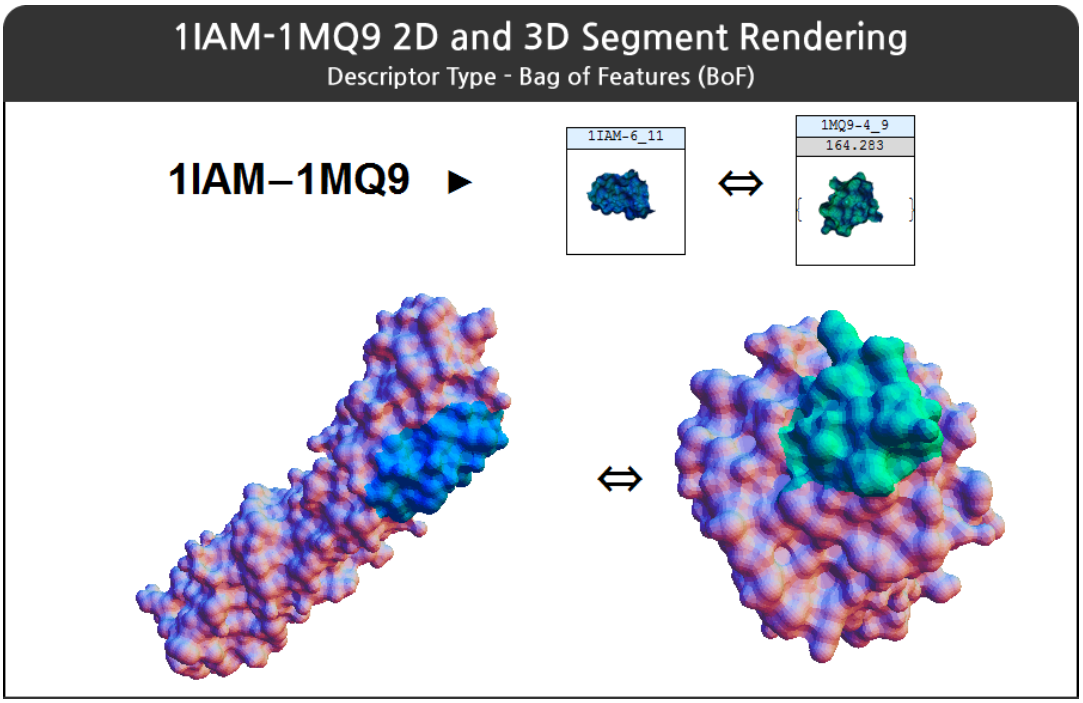


Figure 8.15: Closest matching segments for the 1IAM and 1MQ9 pairing proteins.

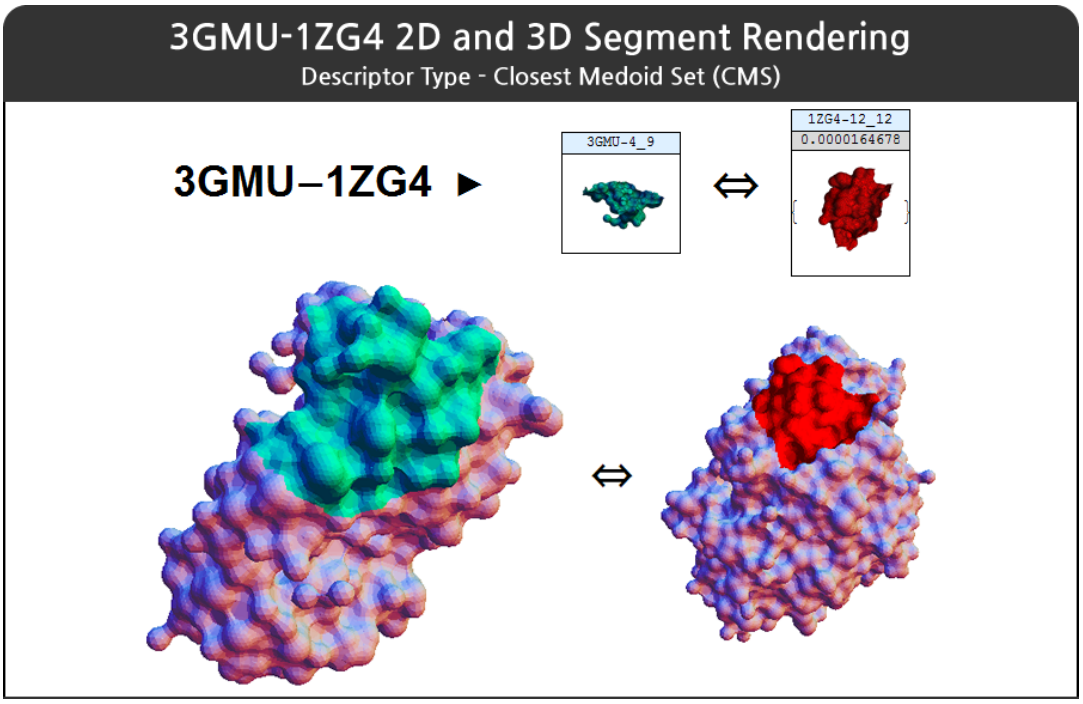


Figure 8.16: Closest matching segments for the 3GMU and 1ZG4 pairing proteins.

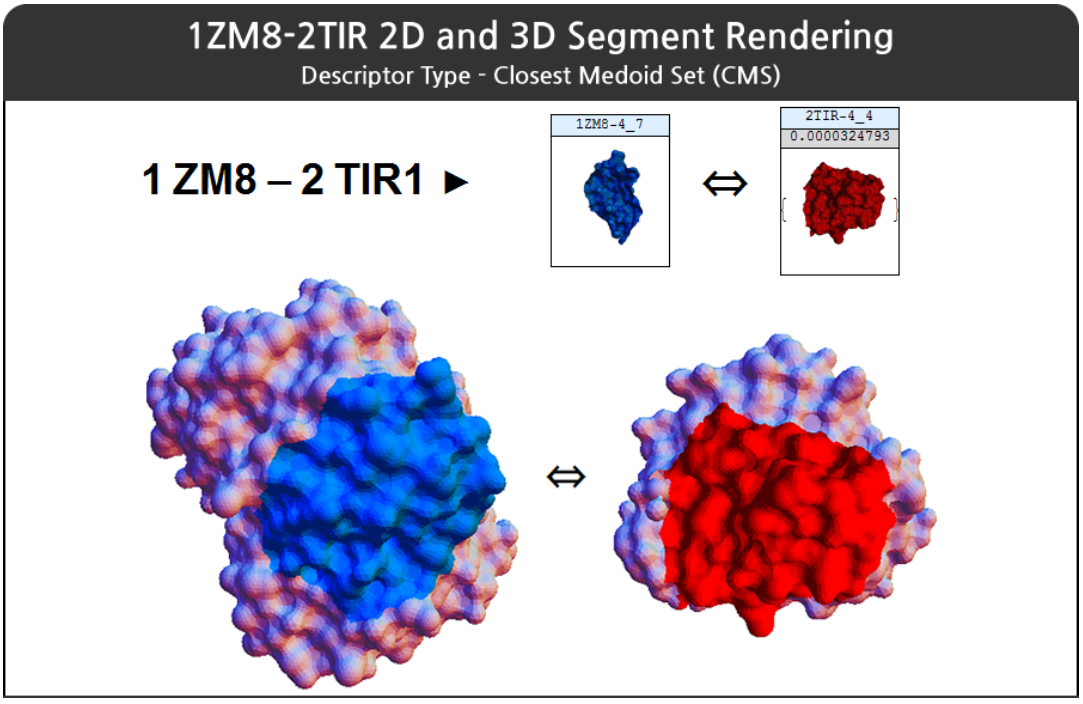


Figure 8.17: Closest matching segments for the 1ZM8 and 2TIR pairing proteins.

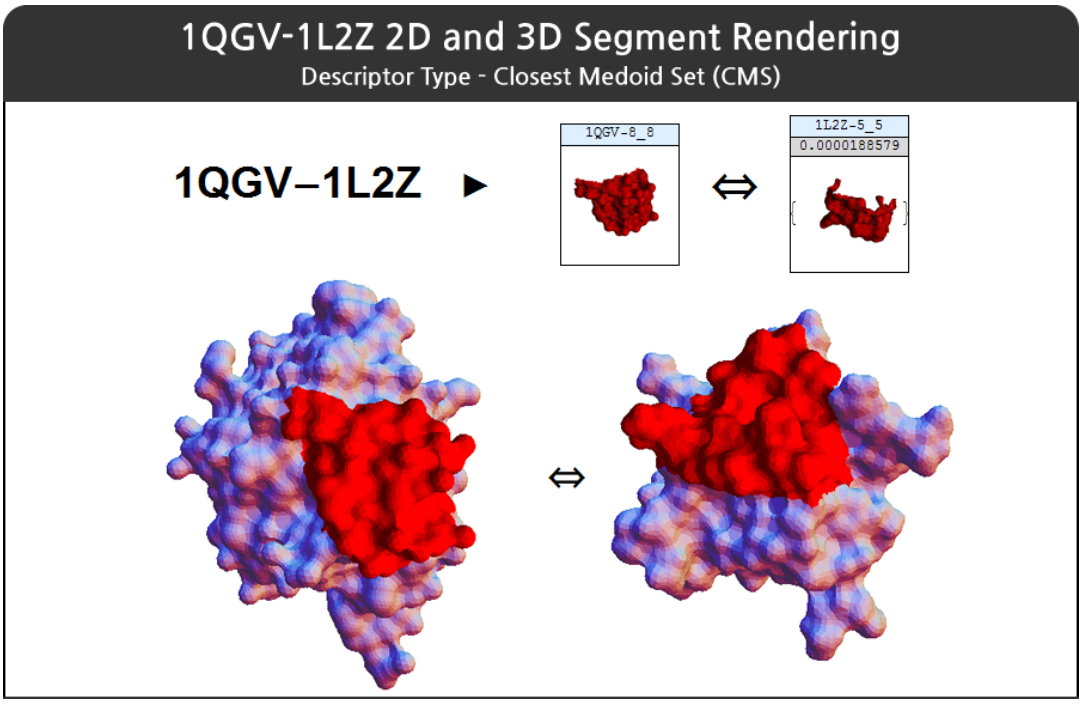


Figure 8.18: Closest matching segments for the 1QGV and 1L2Z pairing proteins.

point-line graphs for Tables 8.2, 8.3 and 8.4.

From Table 8.2, the results show that all three descriptor methods are able to retrieve the complementary ligand segment given a query receptor protein. Specifically, for the matching of the six pairs that are considered difficult or moderately difficult, the Bag of Features (BoF) and Medoid Set Average (MSA) methods correctly rank the matches within the first 12 closest pairings. For example, for the 1QGV-1L2Z pairing, there were 512 segment comparisons performed, and both the BoF and CMS methods correctly rank the segments from the receptor and the known ligand as the closest match. The MSA technique also performs well, ranking the correct pair for 1IAM-1MQ9 first along with the BoF method. It also shows resilience in performing relatively better when matching segments from the difficult complexes.

Results from Table 8.3 (Figure 8.20) depicts pairing performance nearly equal to segments of member counts of 3,500. As with the prior case, segments from the 3GMU

and 1ZG4 proteins structures are paired appropriately as the closest match by both CMS and MSA methods. Again, both BoF and CMS rightfully return segments of 1IAM-1MQ9 as the right matching sites. Finding the matching segments for 1CLO-2TIR pairing returns relatively poorer ranks as compared to the 3,500 member count case. Furthermore, we observe even poorer rankings when the segment member count is increased to 5,000, as found in Table 8.4. Although the CMS descriptor method is the only one that correctly returns the closest match for the *difficult* class of pairings for the 4,000 member count case, we notice that the BoF relatively outperforms both the CMS and MSA methods by returning very close ranks for the 1J54-1SE7, 1ZM8-1J57 and 1HUR-1R8M protein pairings.

Table 8.2: Protein-protein Segment Matching at Average Mesh Vertex Count of 3,500

DIFFICULTY	PAIRING	SEG. COMPARISONS	BoF	CMS	MSA
Rigid Body	1K2I-1PMC	704	9	20	22
	3GMU-1ZG4	576	2	1	4
	3I1U-1ZFI	832	18	3	39
Medium	1IAM-1MQ9	704	1	16	1
	1QGV-1L2Z	512	1	1	2
	1R6C-2W9R	576	11	5	7
Difficult	1CL0-2TIR	640	10	12	8
	1J54-1SE7	576	7	17	8
	1ZM8-1J57	640	8	3	6
	1HUR-1R8M	1026	11	9	24

From Table 8.4, we report relatively poorer retrieval results when the average vertex count is increased to 5,000 with a subsequent surface area per segment of about 6.25×10^8 Å^2. However, given the variations in docking sites for certain pairing proteins, we observe that the retrieval for e.g. the 3I1U-1ZFI pairing, returns the closest match of 1 for this larger segment size. This suggests that the resultant area, of between about 4.37×10^8 Å^2

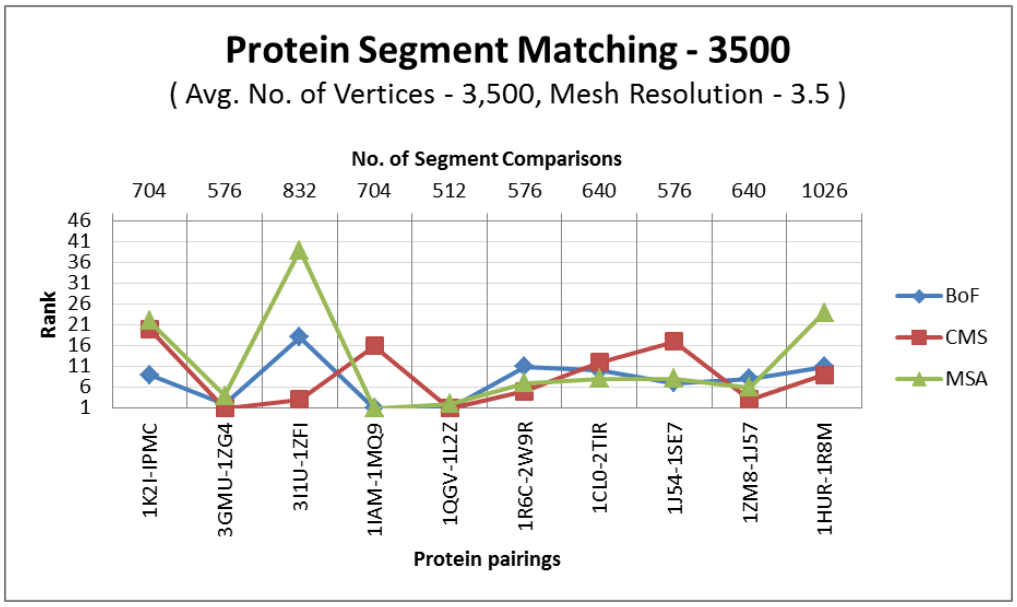


Figure 8.19: A graph showing the rank of the closest matching segments (and the number of segment comparisons) for each known protein pairing with average vertex count of 3,500 for BoF, CMS and MSA descriptor methods.

and 5.18×10^8 Å^2 obtained from the 3,500 and 4,000 member count, forms an adequate segment size for comparing the segments at this resolution. We further posit that an

Table 8.3: Protein-protein Segment Matching at Average Mesh Vertex Count of 4,000

DIFFICULTY	PAIRING	SEG. COMPARISONS	BoF	CMS	MSA
Rigid Body	1K2I-1PMC	560	4	14	19
	3GMU-1ZG4	448	3	1	1
	3I1U-1ZFI	616	78	3	14
Medium	1IAM-1MQ9	560	1	1	3
	1QGV-1L2Z	392	6	12	10
	1R6C-2W9R	392	40	15	11
Difficult	1CL0-2TIR	448	16	14	13
	1J54-1SE7	448	3	17	2
	1ZM8-1J57	504	2	20	19
	1HUR-1R8M	896	2	1	9

Table 8.4: Protein-protein Segment Matching at Average Mesh Vertex Count of 5,000

DIFFICULTY	PAIRING	SEG. COMPARISONS	BoF	CMS	MSA
Rigid Body	1K2I-1PMC	308	8	9	12
	3GMU-1ZG4	264	5	1	1
	3I1U-1ZFI	396	1	34	22
Medium	1IAM-1MQ9	352	11	12	22
	1QGV-1L2Z	220	1	4	24
	1R6C-2W9R	264	11	6	4
Difficult	1CL0-2TIR	770	36	8	16
	1J54-1SE7	308	4	20	13
	1ZM8-1J57	308	10	4	6
	1HUR-1R8M	416	21	18	26

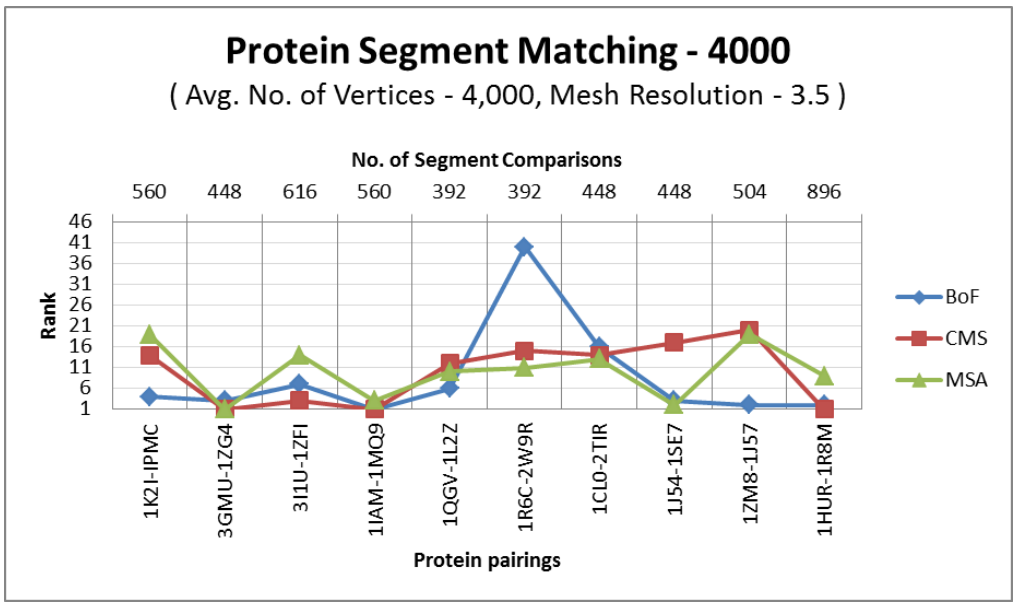


Figure 8.20: A graph showing the rank of the closest matching segments (and the number of segment comparisons) for each known protein pairing with average vertex count of 4,000 for BoF, CMS and MSA descriptor methods.

increase in the number of varied segment sizes will likely produce more insight into the possible protein matching segments, given that it will sufficiently cover the variations in the sizes of the docking sites for different protein complexes.

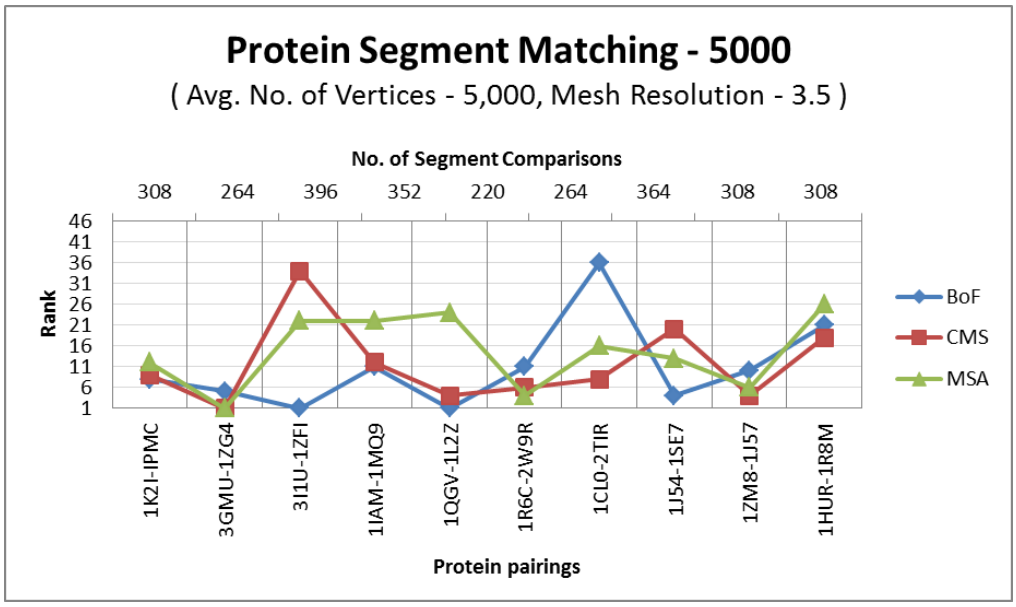


Figure 8.21: A graph showing the rank of the closest matching segments (and the number of segment comparisons) for each known protein pairing with average vertex count of 5,000 for BoF, CMS and MSA descriptor methods.

Low Resolution Protein Matching. Recall that the three-dimensional molecular representations of the protein structures were generated at the default mesh resolution of 3.5 and the low resolution of 0.5. As discussed prior, the 0.5 resolution presents a very coarse representation of the protein structures by using very minimum mesh triangulations.

Table 8.5 presents results of our experimentation using the 0.5 mesh resolution protein structures. We set a segment member count of 820, using the first 100 eigenvectors at a maximum heating time of 100, and a medoid set of 30. These parameters were chosen by estimating the proportions obtained from the 3.5 mesh resolution structures, which had segments of 4,000 vertices. As with the prior instances, we first provide the 2D and 3D renderings of the obtained closest rankings as a visual aid. Figure 8.22 shows the closest ranking segments correctly retrieved for the 3GMU and 1ZG4 protein complex.

From the results in Table 8.5 (with its associated point-line graph in Figure 8.23),

we observe relatively similar segment matching results here as compared to the high resolutions ranking for 3,500 (Table 8.2) and 4,000 (Table 8.3) member count cases. The pairings of 3GMU-1ZG4, 1IAM-1MQ9, and 1CLO-2TIR are correctly ranked with the closest segments. This ranking is in close correlation with the rankings obtained in the high resolution context. Furthermore, we observe comparatively better rankings for the *Rigid Body* and *Medium* classes of difficulty with respect to the *difficult* complexes. This suggests that, the low resolution mesh structures loose some information that is required to correctly identify deformable (or flexible) structures, once some deformation has occurred.

Given the computational efficiency, the low resolution meshes can be duly employed as a preliminary filtering process to aid the more thorough and computationally expensive matching process using the higher resolution mesh structures. Additionally, the use of

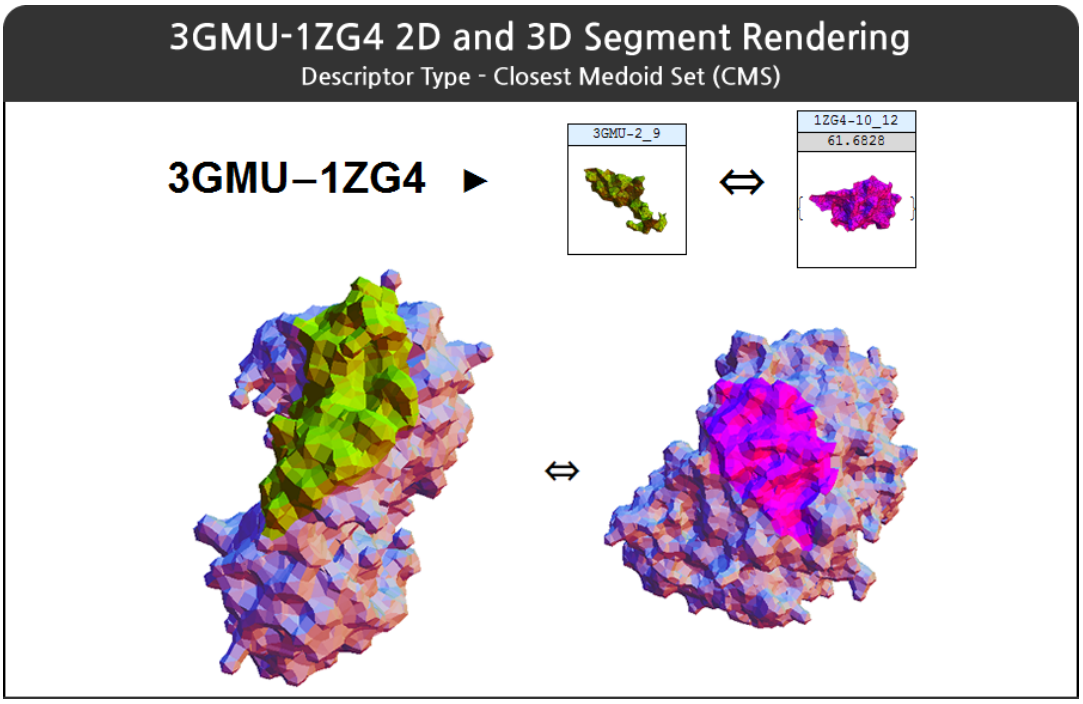


Figure 8.22: Closest matching segments for the 3GMU and 1ZG4 pairing proteins at 0.5 mesh resolution.

the low resolution meshes may be used to fully address the segment matching problem, but only limited to *Rigid body*, and to a lesser extent *medium* class protein complexes.

Table 8.5: Protein-protein Segment Matching at Average Mesh Vertex Count of 820 at 0.5 mesh resolution

DIFFICULTY	PAIRING	SEG. COMPARISONS	BoF	CMS	MSA
Rigid Body	1K2I-1PMC	720	15	12	18
	3GMU-1ZG4	648	1	2	1
	3I1U-1ZFI	936	2	8	2
Medium	1IAM-1MQ9	864	3	1	2
	1QGV-1L2Z	576	2	16	12
	1R6C-2W9R	648	41	20	2
Difficult	1CL0-2TIR	1152	14	1	14
	1J54-1SE7	648	26	40	32
	1ZM8-1J57	720	7	19	20
	1HUR-1R8M	1349	4	4	12

8.2.4.1 Per-segment Ranking

We finally present a brief discussion on an alternative segment matching ranking system. Here, for each segment of a target receptor protein, the n closest segments are retrieved. Figure 8.24 and Figure 8.25 show the *4th* closest match for each segment of the target 3GMU target receptor protein, when compared to the segments of the other ligands, at 3.5 and 0.5 mesh resolutions respectively.

We observe that, for both per-segment rankings of 3.5 and 0.5 mesh resolutions, segments from the known docking ligand 1ZG4 occur more frequently than any other ligand. For example, for the 3.5 high resolution mesh comparisons, out of a total of 36 instances, segments from 1ZG4 are ranked within the first 4 for each segment of 3GMU, and appear 13 times. Furthermore, 1ZG4 was ranked the closest matching for segments

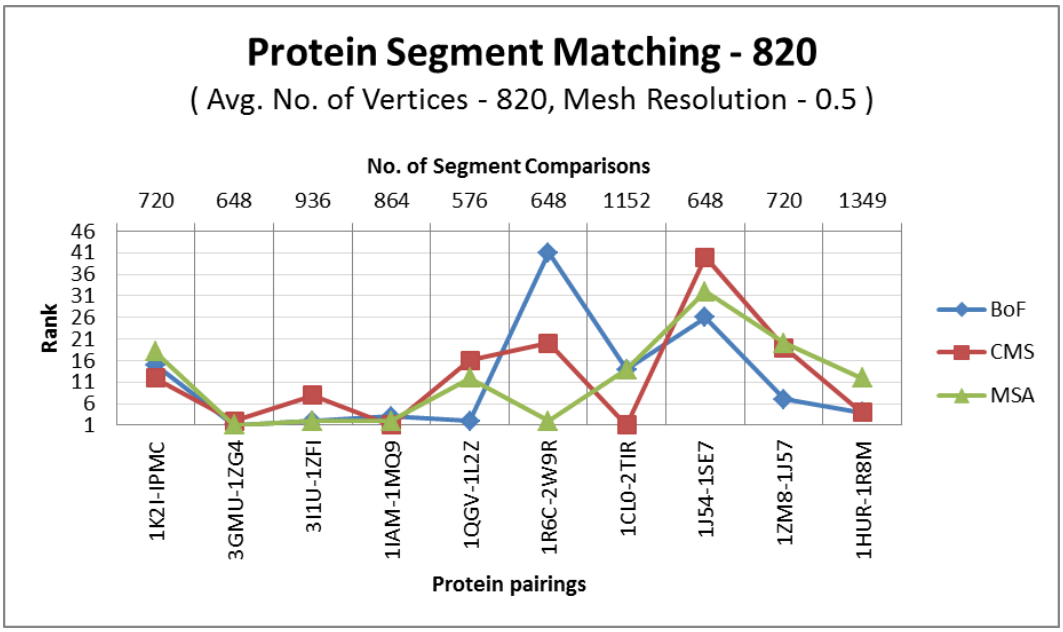


Figure 8.23: A graph showing the rank of the closest matching segments (and the number of segment comparisons) for each known protein pairing with average segment vertex count of 820 for BoF, CMS and MSA descriptor methods, on mesh resolution of 0.5.

3,4 and 6 of 3GMU. A similar trend is exhibited for the 0.5 low-res mesh resolutions. From a similar total of 36 occurrences, 13 segments of 1ZG4 are present within all the rankings, with 2 of the closest matching segments are being from the 3GMU ligand.

We suggest that, this per-segment ranking presents another avenue for a filtering process (and possibly for finding ligands of high docking probability). The filtering technique will select the k most likely ligands by considering not only the frequency of occurrence of their segments, but also the closeness measure of each segment as a factor of the closest and farthest matching segment. We further propose the use of this ranking method as an alternate (or complementary) pre-filtering phase for more comprehensive pairing techniques.

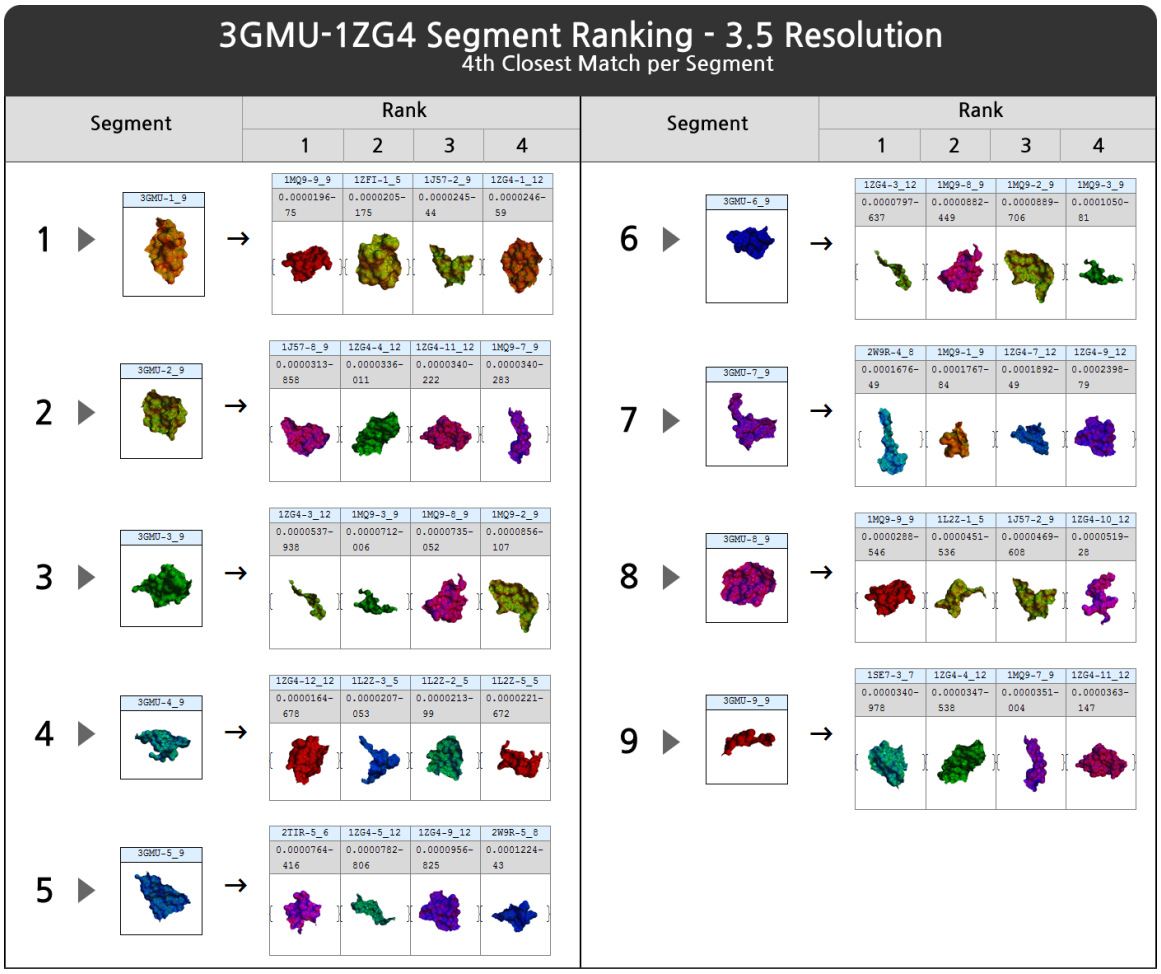


Figure 8.24: Per-segment closest matching ranking for the 3GMU and 1ZG4 pairing proteins at 3.5 mesh resolution.

8.3 Conclusion

In this Chapter, we first presented the results of our experiments arising from our preliminary evaluation of the spectral segmentation and the influence of the variation in the input parameters to the quality of segmentations produced. From the results, the main contributing factor to the correctness of the segmentation is the number of eigenvectors used, such that, smaller numbers of eigenvectors obtains the best and most intuitive segmentations. We then reviewed the results obtained from the evaluation of our two de-

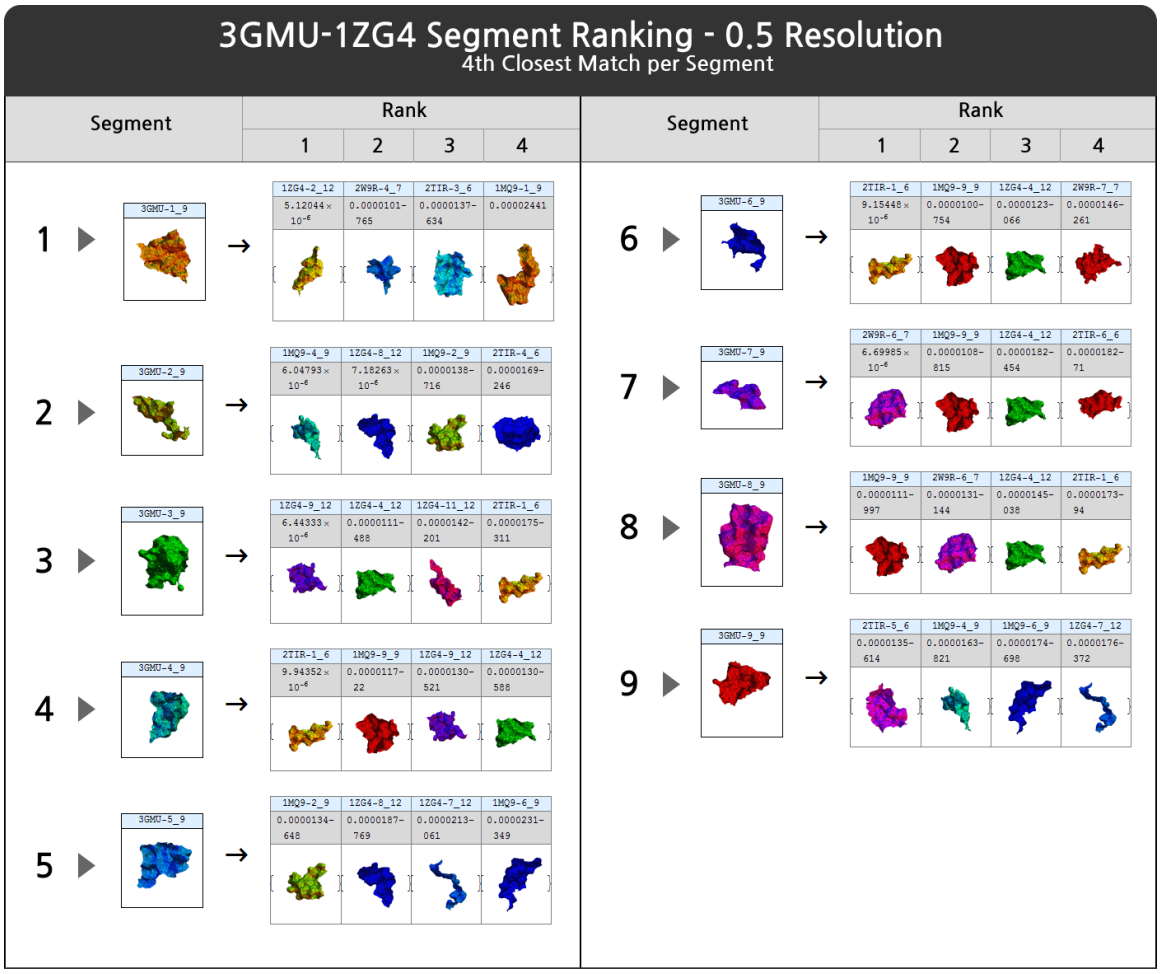


Figure 8.25: Per-segment closest matching ranking for the 3GMU and 1ZG4 pairing proteins at 0.5 mesh resolution.

scriptor vector methods, Closest Medoid Set (CMS) and the Medoid Set Average (MSA) when compared to the Bag of Features (BoF) method. The CMS method consistently outperformed the rest, returning perfect precision when small eigenvectors are used in the construction of the descriptor's Laplacian. Furthermore, both the CMS and the MSA, generally, returned better matches than the BoF method.

For the protein docking pair prediction, we report results which indicate that our proposed ProtoDock algorithm is a very suitable method for finding the matching sites on prospective protein structures. For most of the segment sizes containing averagely

3,500 and 4,000 vertices, the correct pairings were ranked first for three (3) complexes, with the correct possible pairing proteins consistently found within the first 12 closest ranked segments. Although a further increase in the segment size to 5,000 showed less accurate rankings for most of the complexes, it suggests that more insight can be obtained by sufficiently varying the segment sizes in order to account for different sizes of docking sites. Additional experimentation with lower resolution protein meshes also returned encouraging results for rigid body and medium difficulty pairings, suggesting the use of the low resolution meshes as an even less expensive pruning approach.

In the following final Chapter 9, we present the concluding remarks, discuss our contributions, and then give a brief overview of our future work.

Chapter 9

Conclusion and Future Work

The protein docking problem refers to the computational method of predicting the appropriate matching segments of a given receptor to that of another ligand in the process of binding to form a stable molecular complex. The solution to this problem often proves highly combinatorial where either optimization methods attempt a minimum energy estimation, or the relatively less computationally expensive geometric methods try to match the segments using the geometry of the binding surfaces.

In this thesis, we presented a geometric deformable shape matching algorithm which is able to appropriate match pairing segments between receptor and ligand protein structures. The algorithm employs the highly informative multi-resolution Heat Kernel Signature as its core method of describing a given surface. Our algorithm finally utilizes our novel compact descriptor methods for shape comparisons.

9.1 Thesis Contributions

We give a summary of our thesis contributions arising from the presented algorithm in Chapter 6 for addressing the protein docking problem, and also from the explorative

experimentations of the different phases of our algorithm.

ProtoDock-1 Algorithm. We presented out ProtoDock-1 algorithm, which is an isometry-invariant, topologically robust deformable partial and whole shape matching algorithm. Our ProtoDock-1 algorithm utilizes the highly informative Heat Kernel Signature which is based on the diffusion of heat on a surface restricted to the temporal domain. Given its invariance to isometry and its property of being intrinsic, our ProtoDock algorithm addresses the shape matching problem by requiring only basic mesh information, and without the need for any preprocessing activities, such as pose normalization. Our ProtoDock-1 algorithm has experimentally shown its viability for finding matching segments. Also, given that it uses a very compact descriptor vector in representing a segment, our approach therefore provides a fast method of performing future comparisons at very little computational costs.

ProtoDock-2 Algorithm. We further presented an alternate ProtoDock-2 algorithm which also presents a method for finding partial matches on deformable shapes. Like the ProtoDock-1 algorithm, this approach also utilizes the Heat Kernel Signature in describing segments of an object. However, the partial shape matching is achieved by comparing the Heart Kernel Descriptor of points sampled on the surface of an object. We proposed 3 sampling methods which attempt to be as covering as possible. The ProtoDock-2 algorithm has the benefit not requiring the complete computation of all the Heat Kernel Signatures over a given query object.

Closest Medoid Set (CMS) descriptor method. We proposed a novel descriptor method, called the Closest Medoid Set, for representing three-dimension surfaces. The CMS descriptor uses only the set of representative vectors obtained by applying the K-

Means clustering method to the set of Heat Kernel Signatures associated with the points over an object's surface. From preliminary experimentations, the Closest Medoid Set has shown to be more robust and sufficiently superior at matching regular deformable/non-rigid shapes, when compared to the popular Bag of Features methods.

Medoid Set Average (MSA) descriptor method. We further proposed another novel descriptor method, called the Medoid Set Average (MSA). Similar to the CMS descriptor method, the MSA also utilizes just the representative Heat Kernel Signatures of a shape in describing it. The Medoid Set Average forms an even more compact descriptor than the CMS approach, by finding the column-wise average of the set of medoids of a given shape. Although not as accurate as the Closest Medoid Set method at matching regular deformable shapes, the MSA method still generally outperforms the Bag of Features method in this regard. It also shows comparably effective complementary matching as the Bag of Features when used in addressing the protein docking problem.

Sampled Laplacians for HKS Description. We investigated the viability of Laplacians constructed from a sampling on the original mesh for both whole and partial shape matching. The modified Laplacian is constructed by selecting the k closest neighbours of a set of sample points on the object mesh. We show the competence of this approach for performing shape matching.

Per-segment Sampling. We examined the use results of the segmentations as induced by spectral analysis and proposed a consistent and topologically robust per-segment sampling technique, such that identified regions are persistently segmented irrespective of appreciable flexible deformations of the mesh graph. The sampling method first performs a partition on the matrix Laplacian of the underlying graph to obtain the required

segments. Such sampling techniques may be useful in other application domains such as in social network analysis. We also suggested that this technique isn't limited to connected graph objects alone, but can be extend to point clouds given appropriate Laplacians.

9.2 Future Work

One main direction of advancing this work is in the consideration of several descriptors as an ensemble of collaborative 'classifiers', such that, a matching between two shapes is performed by several descriptor methods, and a final measure is arrived at by a form of majority weighted voting. In this way, the cumulative ranking of a pair of possible matches is skewed by the confidence assigned to each descriptor method. The use of such an ensemble technique brings the advantage of have accruing the several beneficial properties inherent in the various descriptors. Also, the confidence assigned to each descriptor can be varied depending on the characteristics of the shape being assessed *e.g.*,. isometry invariant descriptors will be assigned a higher confidence when dealing with deformable shapes.

Another area of concentration for future work lies in considering the chemical properties of the protein structures under consideration, particularly hydrophobicity, since this is one of the key determinants as to whether a receptor and a ligand will bind.

Another area of interest lies in finding intelligent computational and possibly unsupervised methods for selecting the different input parameters when computing descriptors for shape matching using the Heat Kernel Signature. Such parameters as the maximum heating time, the size of the medoid set size, and the number of eigenvectors for constructing the Laplace-Beltrami operator, play a key role in determining the quality and consequent performance of the final descriptor. Creating an unsupervised method, which

utilizes *e.g.*,the mesh information in determining the needed input parameters, will be highly beneficial.

We will also consider evaluating our algorithm against other geometric and possibly low-energy optimization methods. This will involve using a common dataset of protein structures validated using both a benchmark and verified results from well established docking applications.

A final avenue of importance is in exploring the correctness of our partial shape matching and complementary shape algorithm. As of this writing, there exist no established dataset for assessing shape complementarity algorithms. Such a dataset should provide three-dimensional mesh objects with unique regions of known complementary shape matching. To this end, work will also be dedicated to creating such a benchmark dataset.

Appendix A

Glossary of Terms

BoF Bag of Features.

CMS Closest Medoid Set.

HK Heat Kernel.

HKS Heat Kernel Signature (also known as Heat Kernel Descriptor).

LBO Laplace-Beltrami Operator.

MSA Medoid Set Average.

RS Random Sampling.

SAS Solvent-Accessible Surface.

SES Solvent-Excluded Surface (also known as Molecular Surface).

SRS Segment-based Random Sampling.

US Uniform Sampling.

Bibliography

[1] Agathos, A., Pratikakis, I., Perantonis, S., Sapidis, N., and Azariadis, P. 3D Mesh Segmentation Methodologies for CAD Applications. *Computer-Aided Design & Applications 4*, 6 (2007), 827–841.

[2] Akbar, S., Kung, J., and Wagner, R. Exploiting Geometrical Properties on Protein Similarity Search. In *DEXA Workshops* (2006), IEEE Computer Society, pp. 228–234.

[3] Atilgan, E., and Hu, J. Efficient Protein-ligand Docking using Sustainable Evolutionary Algorithm. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2010), GECCO '10, ACM, pp. 211–212.

[4] Attene, M., Katz, S., Mortara, M., Patané, G., Spagnuolo, M., and Tal, A. Mesh Segmentation - A Comparative Study. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on* (2006), IEEE, pp. 7–7.

[5] Axenopoulos, A., Daras, P., Papadopoulos, G., and Houstis, E. 3D Protein-Protein Docking using Shape Complementarity and Fast Alignment. In

Image Processing (ICIP), 2011 18th IEEE International Conference on (September 2011), pp. 1569–1572.

[6] Belkin, M., and Niyogi, P. Convergence of Laplacian Eigenmaps. *Advances in Neural Information Processing Systems 19* (2007), 129.

[7] Belkin, M., Sun, J., and Wang, Y. Discrete Laplace Operator on Meshed Surfaces. In *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry* (New York, NY, USA, 2008), SCG '08, ACM, pp. 278–287.

[8] Benhabiles, H., Vandeborre, J.-P., Lavou, G., and Daoudi, M. A Comparative Study of Existing Metrics for 3D Mesh Segmentation Evaluation. *The Visual Computer 26*, 12 (2010), 1451–1466.

[9] Besl, P. J., and Jain, R. C. Segmentation through Variable-Order Surface Fitting. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 2 (Mar. 1988), 167–192.

[10] Bespalov, D., Regli, W. C., and Shokoufandeh, A. Reeb Graph Based Shape Retrieval for CAD. - (2003).

[11] Biasotti, S., Marini, S., Mortara, M., and Patané, G. An Overview on Properties and Efficacy of Iopological Skeletons in Shape Modeling. In *Shape Modeling International, 2003* (2003), IEEE, pp. 245–254.

[12] Braunstein, M. L., Hoffman, D. D., Saidpour, A., et al. Parts of Visual Objects: An Experimental Test of the Minima Rule. *Perception 18*, 6 (1989), 817–826.

[13] Bronstein, A. M., Bronstein, M. M., Kimmel, R., Mahmoudi, M., and Sapiro, G. A Gromov-Hausdorff Framework with Diffusion Geometry for

Topologically-Robust Non-rigid Shape Matching. *Int. J. Comput. Vision 89*, 2-3 (Sept. 2010), 266–286.

[14] Bronstein, Alexander and Bronstein, Michael and Kimmel, Ron . TOSCA Project– http://tosca.cs.technion.ac.il/.

[15] Burkhard Rost. Protein Structure Prediction in 1D, 2D, and 3D. - *3* (1998), 2242–2255.

[16] Bustos, B., Keim, D., Saupe, D., and Schreck, T. Content-Based 3D Object Retrieval. *Computer Graphics and Applications, IEEE 27*, 4 (2007), 22–27.

[17] Chazelle, B., Dobkin, D. P., Shouraboura, N., and Tal, A. Strategies for Polyhedral Surface Decomposition: An Experimental Study. *Comput. Geom. Theory Appl. 7*, 5-6 (Apr. 1997), 327–342.

[18] Chen, D.-Y., Tian, X.-P., Shen, Y.-T., and Ouhyoung, M. On Visual Similarity Based 3D Model Retrieval. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 223–232.

[19] Chen, S.-C., and Chen, T. Retrieval of 3D Protein Structures. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (2002), vol. 3, IEEE, pp. 933–936.

[20] Chen, X., Golovinskiy, A., and Funkhouser, T. A Benchmark for 3D Mesh Segmentation. In *ACM SIGGRAPH 2009 papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 73:1–73:12.

[21] Corney, J., Rea, H., Clark, D., Pritchard, J., Breaks, M., and Macleod, R. Coarse Filters for Shape Matching. *IEEE Comput. Graph. Appl. 22*, 3 (May 2002), 65–74.

[22] Cui, C., and Shi, J. Automatic Retrieval of 3D Protein Structures Based on Shape Similarity. In *Storage and Retrieval Methods and Applications for Multimedia* (2004), pp. 543–549.

[23] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 317–324.

[24] Dong, S., Bremer, P.-T., Garland, M., Pascucci, V., and Hart, J. C. Spectral Surface Quadrangulation. In *ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 1057–1066.

[25] Duda, R. O., Hart, P. E., and Stork, D. G. *Pattern Classification*. John Wiley & Sons, 2012.

[26] Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. Multiresolution Analysis of Arbitrary Meshes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 1995), SIGGRAPH '95, ACM, pp. 173–182.

[27] Elad, M., Tal, A., and Ar, S. Content Based Retrieval of VRML objects: An Iterative and Interactive Approach. In *Proceedings of the Sixth Eurographics Workshop on Multimedia 2001* (New York, NY, USA, 2002), Springer-Verlag New York, Inc., pp. 107–118.

[28] ExPasy. ExPasy Bioinformatics Resource—http://www.expasy.org.

[29] ExPasy. ExPasy Bioinformatics Resource on Sequence Entries– http://web.expasy.org/docs/relnotes/relstat.html, March 2013.

[30] Fowlkes, C., Belongie, S., Chung, F., and Malik, J. Spectral Grouping using the Nystrom Method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*, 2 (2004), 214–225.

[31] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D. A Search Engine for 3D Models. *ACM Trans. Graph. 22*, 1 (Jan. 2003), 83–105.

[32] Gao, B., Zheng, H., and Zhang, S. An Overview of Semantics Processing in Content-Based 3D Model Retrieval. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on* (2009), vol. 2, IEEE, pp. 54–59.

[33] Garland, M., Willmott, A., and Heckbert, P. S. Hierarchical Face Clustering on Polygonal Surfaces. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics* (New York, NY, USA, 2001), I3D '01, ACM, pp. 49–58.

[34] Gelfand, N., and Guibas, L. J. Shape Segmentation using Local Slippage Analysis. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (New York, NY, USA, 2004), SGP '04, ACM, pp. 214–223.

[35] Gregory, A., State, A., Lin, M. C., Manocha, D., and Livingston, M. A. Interactive Surface Decomposition for Polyhedral Morphing. *The Visual Computer 15*, 9 (1999), 453–470.

[36] hao Chi, P., Scott, G., and ren Shyu, C. A Fast Protein Structure Retrieval System using Image-based Distance Matrices and Multidimensional Index.

International Journal of Software Engineering and Knowledge Engineering, Special Issue on Software and Knowledge Engineering Support in Bioinformatics 2005 (2004), 522–532.

[37] Hashmi, I., and Shehu, A. A Basin Hopping Algorithm for Protein-Protein Docking. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2012), IEEE Computer Society, pp. 1–4.

[38] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 203–212.

[39] Hoffman, D. D., and Richards, W. A. Parts of Recognition. *Cognition 18*, 1 (1984), 65–96.

[40] Hu, J., Goodman, E. D., Seo, K., Fan, Z., and Rosenberg, R. The Hierarchical Fair Competition (HFC) Framework for Sustainable Evolutionary Algorithms. vol. 13, pp. 241–277.

[41] Huang, Z., Zhou, X., Shen, H. T., and Song, D. 3D Protein Structure Matching by Patch Signatures. In *Proceedings of the 17th International Conference on Database and Expert Systems Applications* (Berlin, Heidelberg, 2006), DEXA'06, Springer-Verlag, pp. 528–537.

[42] Hwang, H., Vreven, T., Janin, J., and Weng, Z. Protein-Protein Docking Benchmark Version 4.0. *Proteins: Structure, Function, and Bioinformatics, 78*, 15 (2010), 3111–3114.

[43] Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., and Ramani, K. Three-dimensional Shape Searching: State-of-the-Art Review and Future Trends. *Comput. Aided Des. 37*, 5 (Apr. 2005), 509–530.

[44] JMol. JMol– http://jmol.sourceforge.net/docs/surface/.

[45] Kahraman, A., Morris, R., Laskowski, R., and Thornton, J. Shape Variation in Protein Binding Pockets and their Ligands. vol. 368, pp. 283–301.

[46] Kalogerakis, E., Hertzmann, A., and Singh, K. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions in Graphics 29*, 4 (July 2010), 102:1–102:12.

[47] Kalvin, A., and Taylor, R. Superfaces: Polygonal Mesh Simplification with Bounded Error. *Computer Graphics and Applications, IEEE 16*, 3 (1996), 64–77.

[48] Katz, S., and Tal, A. Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts. In *ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), SIGGRAPH '03, ACM, pp. 954–961.

[49] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), SGP '03, Eurographics Association, pp. 156–164.

[50] Koenderink, J. J. *Solid Shape*. MIT Press, Cambridge, MA, USA, 1990.

[51] Körtgen, M., Park, G.-J., Novotni, M., and Klein, R. 3D Shape Matching with 3D Shape Contexts. In *The 7th Central European Seminar on Computer Graphics* (2003), vol. 3, pp. 5–17.

[52] Koschan, A. Perception-based 3D Triangle Mesh Segmentation using Fast Marching Watersheds. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 2, IEEE, pp. II–27.

[53] Kryshtafovych A., Venclovas C., Fidelis K., Moult J. Progress Over the First Decade of CASP Experiments. *Proteins 61 Suppl 7* (2005).

[54] Laurence A. Moran, Robert A. Horton, G. S. M. P. *Principles of Biochemistry*, 5th ed. Prentice Hall, September 2011.

[55] Lavoué, G., Dupont, F., and Baskurt, A. A New CAD Mesh Segmentation Method based on Curvature Tensor Analysis. *Comput. Aided Des. 37*, 10 (Sept. 2005), 975–987.

[56] Levy, B. Laplace-Beltrami Eigenfunctions: Towards an Algorithm That "Understands" Geometry. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006* (Washington, DC, USA, 2006), SMI '06, IEEE Computer Society, pp. 13–.

[57] Lévy, B., Petitjean, S., Ray, N., and Maillot, J. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Transactions in Graphics 21*, 3 (July 2002), 362–371.

[58] Liu, R., and Zhang, H. Segmentation of 3D Meshes through Spectral Clustering. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference* (Washington, DC, USA, 2004), PG '04, IEEE Computer Society, pp. 298–305.

[59] Liu, R., and Zhang, H. Mesh Segmentation via Spectral Embedding and Contour Analysis. In *Computer Graphics Forum* (2007), vol. 26, Wiley Online Library, pp. 385–394.

[60] Liu, Y., Zha, H., and Qin, H. The Generalized Shape Distributions for Shape Matching and Analysis. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006* (Washington, DC, USA, 2006), SMI '06, IEEE Computer Society, pp. 16–.

[61] Lo, Y.-T., Tsai, Y.-L., Wang, H.-W., Hsu, Y.-P., and Pai, T.-W. Using Solid Angles to Detect Protein Docking Regions by CUDA Parallel Algorithms. In *Proceedings of the International Symposium on Parallel and Distributed Processing with Applications* (Washington, DC, USA, 2010), ISPA '10, IEEE Computer Society, pp. 536–541.

[62] Mahmoudi, S., and Daoudi, M. 3D Models Retrieval by using Characteristic Views. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on* (2002), vol. 2, IEEE, pp. 457–460.

[63] Mangan, A. P., and Whitaker, R. T. Partitioning 3D Surface Meshes using Watershed Segmentation. *Visualization and Computer Graphics, IEEE Transactions on 5*, 4 (1999), 308–321.

[64] Martin, D., Fowlkes, C., Tal, D., and Malik, J. 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* (2001), vol. 2, pp. 416–423 vol.2.

[65] Meyer, M., Desbrun, M., Schröder, P., Barr, A. H., et al. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. *Visualization and Mathematics 3*, 2 (2002), 52–58.

[66] Min, P., Kazhdan, M., and Funkhouser, T. A Comparison of Text and Shape Matching for Retrieval of Online 3D Models. In *Research and Advanced Technology for Digital Libraries*. Springer, 2004, pp. 209–220.

[67] Moll, A., Hildebrandt, A., Lenhof, H.-P., and Kohlbacher, O. BALLView: An Object-Oriented Molecular Visualization and Modeling Framework. *Journal of Computer-aided Molecular Design 19*, 11 (2005), 791–800.

[68] Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. Automated Docking using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function. vol. 19, pp. 1639–1662.

[69] Novotni, M., and Klein, R. A Geometric Approach to 3D Object Comparison. In *Proceedings of the International Conference on Shape Modeling & Applications* (Washington, DC, USA, 2001), SMI '01, IEEE Computer Society, pp. 167–.

[70] Nykamp, D. Q. The Idea of the Divergence of a Vector Field- Math Insight– http://mathinsight.org/divergence_idea.

[71] Ohbuchi, R., Minamitani, T., and Takei, T. Shape-Similarity Search of 3D Models by using Enhanced Shape Functions. In *Proceedings of the Theory and Practice of Computer Graphics 2003* (Washington, DC, USA, 2003), TPCG '03, IEEE Computer Society, pp. 97–.

[72] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. Shape Distributions. *ACM Trans. Graph. 21*, 4 (Oct. 2002), 807–832.

[73] Ovsjanikov, M., Bronstein, A. M., Bronstein, M. M., and Guibas, L. J. Shape Google: A Computer Vision Approach to Isometry Invariant Shape

Retrieval. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 320–327.

[74] Paquet, E., Rioux, M., Murching, A., Naveen, T., and Tabatabai, A. Description of Shape Information for 2-D and 3-D Objects. *Signal Processing: Image Communication 16*, 1 (2000), 103–122.

[75] Paquet, E., and Viktor, H. L. Finding Protein Family Similarities in Real Time through Multiple 3D and 2D Representations, Indexing and Exhaustive Searching. In *KDIR* (2009), pp. 127–133.

[76] Park, H., Park, J., and Park, H. A Protein Structure Retrieval System Using 3D Edge Histogram. In *Key Engineering Materials* (2005), vol. 277-279, pp. 324–330.

[77] Paschalidis, I. C. C. h., Shen, Y., Vakili, P., and Vajda, S. Protein-Protein Docking with Reduced Potentials by Exploiting Multi-dimensional Energy Funnels. vol. 1, pp. 5330–5333.

[78] Pinkall, U., and Polthier, K. Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics 2*, 1 (1993), 15–36.

[79] Pulla, S., Razdan, A., and Farin, G. Improved Curvature Estimation for Watershed Segmentation of 3-Dimensional Meshes. *IEEE Transactions on Visualization and Computer Graphics* (2001).

[80] Rand, W. M. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association 66*, 336 (1971), 846–850.

[81] RCSB PDB. RCSB Protein Data Bank—http://www.rcsb.org/.

[82] Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., and Spagnuolo, M. Discrete Laplace–Beltrami Operators for Shape Analysis and Segmentation. *Computers & Graphics 33*, 3 (2009), 381–390.

[83] Reuter, M., Wolter, F.-E., and Peinecke, N. Laplace-Beltrami Spectra as 'Shape-DNA' of Surfaces and Solids. *Comput. Aided Des. 38*, 4 (Apr. 2006), 342–366.

[84] Rohl, Carol A., Strauss, Charlie E., Misura, Kira M., Baker, David. Protein Structure Prediction using Rosetta. *Methods in Enzymology 383* (2004), 66–93.

[85] Rosenfeld, R., Vajda, S., and DeLisi, C. Flexible Docking and Design. *Annual Review of Biophysics Biomolecular Structures 24* (1995).

[86] Royer, C. Protein-Protein Interactions. *Biophysics* (October 2009).

[87] Sánchez-Cruz, H., and Bribiesca, E. A Method of Optimum Transformation of 3D Objects used as a Measure of Shape Dissimilarity. *Image and Vision Computing 21*, 12 (2003), 1027–1036.

[88] Schulz, R. Protein Structure Prediction. *Proteins* (2007).

[89] Shamir, A. A Survey on Mesh Segmentation Techniques. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1539–1556.

[90] Sharma, A., Horaud, R., Cech, J., and Boyer, E. Topologically-robust 3D Shape Matching based on Diffusion Geometry and Seed Growing. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2011), CVPR '11, IEEE Computer Society, pp. 2481–2488.

[91] Shenoy, S. R., and Jayaram, B. Proteins: Sequence to Structure and Function–Current Status. *Current Protein and Peptide Science 11*, 7 (November 2010), 498–514.

[92] Shi, J., and Malik, J. Normalized Cuts and Image Segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 8 (2000), 888–905.

[93] Shlafman, S., Tal, A., and Katz, S. Metamorphosis of Polyhedral Surfaces using Decomposition. In *Computer Graphics Forum* (2002), vol. 21, Wiley Online Library, pp. 219–228.

[94] Singh, M., and Hoffman, D. D. Part-based Representations of Visual Shape and Implications for Visual Cognition. *Advances in Psychology 130* (2001), 401–459.

[95] Sorkine, O. *Laplacian Mesh Processing*. PhD thesis, Tel Aviv University, 2006.

[96] Sun, J., Ovsjanikov, M., and Guibas, L. A Concise and Provably Informative Multi-scale Signature Based on Heat Diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1383–1392.

[97] Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. Skeleton Based Shape Matching and Retrieval. In *Proceedings of the Shape Modeling International 2003* (Washington, DC, USA, 2003), SMI '03, IEEE Computer Society, pp. 290–.

[98] Suzuki, T. Structural Modulation of the Protein Proteinase Inhibitor SSI (Streptomyces Subtilisin Inhibitor).

[99] Tangelder, J. W., and Veltkamp, R. C. A Survey of Content Based 3D Shape Retrieval Methods. *Multimedia Tools Appl. 39*, 3 (Sept. 2008), 441–471.

[100] Tarr, M. J., Williams, P., Hayward, W. G., and Gauthier, I. Three-Dimensional Object Recognition is Viewpoint Dependent. *Nature Neuroscience 1*, 4 (1998), 275–277.

[101] Thomas Lengauer, M. R. Computational Methods for Biomolecular Docking. *Current Opinion in Structural Biology* (June 1996), 402–406.

[102] Tong, W., and Weng, Z. Clustering Protein-Protein Docking Predictions. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE* (September 2004), vol. 2, pp. 2999–3002.

[103] Tversky, A. Features of Similarity. *Psychological Review 84*, 4 (1977), 327.

[104] Vieira, M., and Shimada, K. Surface Mesh Segmentation and Smooth Surface Extraction through Region Growing. *Comput. Aided Geom. Des. 22*, 8 (Nov. 2005), 771–792.

[105] Vranic, D. V., Saupe, D., and Richter, J. Tools for 3D-Object Retrieval: Karhunen-Loeve Transform and Spherical Harmonics. In *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on* (2001), IEEE, pp. 293–298.

[106] Wardetzky, M., Mathur, S., Kälberer, F., and Grinspun, E. Discrete Laplace Operators: No Free Lunch. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 33–37.

[107] Wikipedia. Gradient– http://en.wikipedia.org/wiki/Gradient.

[108] Wikipedia. Laplace-Beltrami Operator–http://en.wikipedia.org/wiki/Laplace-Beltrami_operator.

[109] Wikipedia. List of Molecular Graphics Systems– http://en.wikipedia.org/wiki/List_of_molecular_graphics_systems.

[110] Wikipedia. Molecular Docking—http://en.wikipedia.org/wiki/Docking_(molecular).

[111] Wikipedia. Proteomics—http://en.wikipedia.org/wiki/Proteomics.

[112] Wolfram Research, Mathematica. Mathematica– http://www.wolfram.com/mathematica/.

[113] Wu, K., and Levine, M. D. 3D Part Segmentation Using Simulated Electrical Charge Distributions. *IEEE Trans. Pattern Anal. Mach. Intell. 19*, 11 (Nov. 1997), 1223–1235.

[114] Xu, G. Discrete Laplace-Beltrami Operators and their Convergence. *Comput. Aided Geom. Des. 21*, 8 (Oct. 2004), 767–784.

[115] Yeh, J.-S., Chen, D.-Y., Chen, B.-Y., and Ouhyoung, M. A Web-based Three-dimensional Protein Retrieval System by Matching Visual Similarity. *Bioinformatics 21*, 13 (July 2005), 3056–3057.

[116] Yubin, Y., Hui, L., and Yao, Z. Content-Based 3D Model Retrieval: A Survey. *Trans. Sys. Man Cyber Part C 37*, 6 (Nov. 2007), 1081–1098.

[117] Zaharia, T., and Preteux, F. J. 3D-Shape-Based Retrieval within the MPEG-7 Framework. In *Photonics West 2001-Electronic Imaging* (2001), International Society for Optics and Photonics, pp. 133–145.

[118] Zhang, C., and Chen, T. Efficient Feature Extraction for 2D/3D Objects in Mesh Representation. In *Image Processing, 2001. Proceedings. 2001 International Conference on* (2001), vol. 3, IEEE, pp. 935–938.

[119] Zhang, C., and Chen, T. Indexing and Retrieval of 3D Models Aided by Active Learning. In *Proceedings of the Ninth ACM international conference on Multimedia* (New York, NY, USA, 2001), MULTIMEDIA '01, ACM, pp. 615–616.

[120] Zhang, H., and Liu, R. Mesh segmentation via Recursive and Visually Salient Spectral Cuts. In *Proceedings of Vision, Modeling, and Visualization* (2005), Citeseer, pp. 429–436.

[121] Zhang Y., Skolnick J. Tertiary Structure Predictions on a Comprehensive Benchmark of Medium to Large Size Proteins. *Proteins* (October 2004).

[122] Zhou, H., Pandit, S. B., Lee, S. Y., Borreguero, J., Chen, H., Wroblewska, L., and Skolnick, J. Analysis of TASSER-based CASP7 Protein Structure Prediction Results. *Proteins 69 Suppl 8* (2007).

[123] Zhou, Y., and Huang, Z. Decomposing Polygon Meshes by Means of Critical Points. In *Proceedings of the 10th International Multimedia Modelling Conference* (Washington, DC, USA, 2004), MMM '04, IEEE Computer Society, pp. 187–.

[124] Zöckler, M., Stalling, D., and Hege, H.-C. Fast and Intuitive Generation of Geometric Shape Transitions. *The Visual Computer 16*, 5 (2000), 241–253.

[125] Zuckerberger, E., Tal, A., and Shlafman, S. Polyhedral Surface Decomposition with Applications. *Computers & Graphics 26*, 5 (2002), 733–743.