

Morse-Smale Complexes: Computation and Applications

A THESIS
SUBMITTED FOR THE DEGREE OF
Doctor of Philosophy
IN THE FACULTY OF ENGINEERING

by

Nithin Shivashankar



Computer Science and Automation
Indian Institute of Science
BANGALORE – 560 012

September 2014

©Nithin Shivashankar

September 2014

All rights reserved

To

Appa and Akka

Acknowledgements

Foremost, I would like to acknowledge my parents and sister whose constant and unwavering support in my endeavors has been the source of inspiration and encouragement through the years of graduate study.

I am deeply indebted to my thesis advisor, Dr. Vijay Natarajan for his guidance and council. Learning is an essential life process that is often accidental and unforeseen. I firmly believe that I have learned more unquantifiably and intangibly from him than intentionally and directly. Through the course of many projects and research papers, his emphasis on critical thinking and academic rigor have deeply influenced my approach to both, academic and non-academic pursuits. During the course of my research, I have had the opportunity to work with many wonderful researchers. In the application of my work to cosmology, I had the opportunity to interact with Dr. Rien van de Weygeart and Pratyush Pranav, to whom I'm grateful for their support and advice. In the application of my work to structural Biology, I would like to thank Dr. Nagasuma Chandra, Sonali Patil, and Amrisha Bhosle for their support and advice. Finally, I had the opportunity to work with Dr. Patrice Koehl, to whom I thankful for his patience during my initial days in research.

IISc today preserves a small piece of Bengaluru of yesteryear, in which I have felt privileged to find calm and clarity during my graduate work. Its people and

activities have played an integral role during my time here. Labmates and other-lab-mates have been a constant source of coffee and conversation, both academic and not-so-academic. The offices of department of Computer Science and Automation have been ably manned by its staff to whom I'm indebted for their help and support. Music was another accidental and unforeseen learning whose privilege I have had by being part of Rhythmica, the music team of IISc.

During my research, I was supported via a stipend by the Ministry of Human Resource Development and The Robert-Bosch center for Cyber Physical systems in IISc. Travels to various conferences were supported by Microsoft Research India, IBM India Research Labs, and Indian Academy for Research in Computer Science (IARCS). A part of my research was funded by Intel and Department of Science and Technology, India via grant SR/S3/EECE/048/2007. I am grateful to these agencies/institutions for their monetary support of my work.

As I reflect over my time here, I find an inescapable connection with my past and the institutions that lead upto here. My high school years were spent at the Lawrence School, Lovedale, nestled in the hills of the Nilgiris, which is also my home town. My undergraduate years were spent at the Manipal Institute of Technology. Both these institutions have proud legacies whose traditions have imparted their indelible imprints upon me. Friendships that were forged there an eon ago have been a source of camaraderie through my graduate years and will continue to be so. I would like to conclude by quoting the Lawrencian credo: "And with strong determination, play the game and **'never give in'**".

Publications based on this Thesis

Conference and Journal publications

1. Nithin Shivashankar, Senthilnathan Maadaswamy, and Vijay Natarajan, “Parallel computation of 2D Morse-Smale complexes”, *IEEE Transactions on Visualization and Graphics*, vol. 18, no. 10, pgs. 1757-1770, 2012.
2. Nithin Shivashankar and Vijay Natarajan, “Parallel computation of 3D Morse-Smale complexes”, *Computer Graphics Forum*, vol. 31, no. 3.1, pgs. 965-974, 2012.
3. Nithin Shivashankar, Pratyush Pranav, Vijay Natarajan, Rien van de Weygaert, E. G. Patrick Bos, and Steven Rieder, “Felix: A Topology based Framework for Visual Exploration of Cosmic Filaments”, *IEEE Transactions on Visualization and Graphics*, Under Submission.
4. Nithin Shivashankar, Sonali Patil, Amrisha Bhosle, Nagasuma Chandra, and Vijay Natarajan, “MS3ALIGN: An efficient molecular surface aligner using the topology of surface curvature”, *BMC Bioinformatics*, Under Submission.
5. Nithin Shivashankar and Vijay Natarajan, “Efficient software for programmable visual analysis using the Morse-Smale complex”, *Workshop on Topology-Based methods in Visualization, 2015*, To Appear.

Abstract

In recent decades, scientific data has become available in increasing sizes and precision. Therefore techniques to analyze and summarize the ever increasing datasets are of vital importance. A common form of scientific data, resulting from simulations as well as observational sciences, is in the form of scalar-valued function on domains of interest. The Morse-Smale complex is a topological data-structure used to analyze and summarize the gradient behavior of such scalar functions. This thesis deals with efficient parallel algorithms to compute the Morse-Smale complex as well as its application to datasets arising from cosmological sciences as well as structural biology.

The first part of the thesis discusses the contributions towards efficient computation of the Morse-Smale complex of scalar functions defined on two and three dimensional datasets. In two dimensions, parallel computation is made possible via a parallelizable discrete gradient computation algorithm. This algorithm is extended to work efficiently in three dimensions also. We also describe efficient algorithms that synergistically leverage modern GPUs and multi-core CPUs to traverse the gradient field needed for determining the structure and geometry of the Morse-Smale complex. We conclude this part with theoretical contributions pertaining to Morse-Smale complex simplification.

The second part of the thesis explores two applications of the Morse-Smale

complex. The first is an application of the 3-dimensional hierarchical Morse-Smale complex to interactively explore the filamentary structure of the cosmic web. The second is an application of the Morse-Smale complex for analysis of shapes of molecular surfaces. Here, we employ the Morse-Smale complex to determine alignments between the surfaces of molecules having similar surface architecture.

Contents

Acknowledgements	i
Publications based on this Thesis	iii
Abstract	iv
1 Introduction	1
1.1 The Morse-Smale Complex	2
1.2 Contributions	4
1.2.1 Efficient Computation	5
1.2.2 Applications of the Morse-Smale complex	7
2 Background	11
2.1 Morse Theory and the Morse-Smale complex	11
2.2 Simplification	14
2.3 The hierarchical Morse-Smale complex	16
2.4 Discrete Morse theory	17
2.5 Piecewise Linear(PL) Functions	20
2.6 Simulation of simplicity	22
3 Related Work	23
3.1 Computation	23
3.2 Applications	28
4 MS Complex Algorithm	33
4.1 In-core Algorithm	33
4.1.1 Discrete function	34
4.1.2 Weighted discrete function	35
4.1.3 Computing gradient pairs	40
4.1.4 Computing the MS complex	48
4.1.5 Analysis	52
4.2 Out-of-core Algorithm	57

4.2.1	Gradient and MS Complex on sub-domains	57
4.2.2	Merging sub-domain MS complexes	59
4.2.3	History Tree	60
4.2.4	Geometry extraction	61
4.3	Discussion	62
4.4	2D Implementation and Evaluation	64
4.4.1	Implementation	64
4.4.2	Experimental Results	67
4.5	3D Implementation and Evaluation	70
4.5.1	Implementation	71
4.5.2	Experiments	73
4.6	Persistence and Simplification in 3D	76
4.7	Conclusions	83
5	Exploring Cosmic Filaments	84
5.1	Introduction	85
5.2	Related Work	87
5.3	Contributions	91
5.4	Methodology	93
5.4.1	Density estimation and filament modeling	93
5.4.2	Density range based filament selection	96
5.4.3	Volume Rendering and Enhancement	100
5.5	Model Description	102
5.5.1	Voronoi evolution models	102
5.5.2	Λ CDM cosmological simulations	104
5.6	Experiments	106
5.6.1	Filaments in the Voronoi model: a comparison	108
5.6.2	Filament Exploration	115
5.6.3	Volume enhancement	118
5.7	Conclusions	121
6	Molecular Surface Alignments	122
6.1	Related Work	123
6.2	Contributions	124
6.3	Background	125
6.4	Surface Alignment Pipeline	127
6.4.1	Curvature Computation	127
6.4.2	Landmark extraction	128
6.4.3	Landmark correspondences	129
6.4.4	Maximal correspondence sets	129
6.4.5	Surface Alignment	130

6.5	Experiments	132
6.5.1	Molecular surfaces	132
6.5.2	Performance analysis	135
6.5.3	A comparison with SURFCOMP	137
6.5.4	A validation using POCKETMATCH and PYMOL	145
6.5.5	Aligning Electron Microscopy Iso-Surfaces	148
6.6	Conclusions	151
7	Conclusions	152
	Bibliography	155

List of Tables

3.1	Comparison of prior Morse-Smale complex algorithms	25
3.2	Contributions for efficient Morse-Smale complex computation	27
3.3	A comparison of some Morse-Smale complex applications	31
4.1	Running time comparison with other MS complex algorithms	75
5.1	Relative abundance of particles in each structural element in the Voronoi Kinematic datasets	104
5.2	Comparing classification recovery rates of Felix with Nexus/MMF, SpineWeb, and DisPerSE	114
6.1	Comparison of the chemically relevant alignments of SURFCOMP and MS3ALIGN	146
6.2	Electron Microscopy dataset details	150
6.3	RMSD after aligning Cryo-EM iso-surfaces	150

List of Figures

1.1	Scalar function examples	2
1.2	Morse-Smale complex example	3
2.1	Morse theory concepts	12
2.2	Topological simplification	15
2.3	Strangulation example	16
2.4	Hierarchical Morse-Smale complex	17
2.5	Cell complex examples	18
2.6	Discrete Morse function	19
2.7	Betti Numbers of a Torus	21
4.1	Examples of G_0 and G_1	36
4.2	ϵLST and ϵLLK of a cell.	40
4.3	Extending a scalar function f and gradient pairs.	41
4.4	The need for Algorithm ASSIGNGRADIENT2	43
4.5	Proof of extended Lemma 4.1.3	45
4.6	MS complex from gradient field.	48
4.7	Complexity of 1-saddle-2-saddle substructure.	51
4.8	Homology preservation of Algorithms 4.3 and 4.4	55
4.9	Quality of gradient of Algorithms 4.3 and 4.4	56
4.10	Computing MS Complex Out-of-core.	58
4.11	Scaling studies using the 2D <code>wgauss</code> dataset	67
4.12	Morse decompositions of the 2D <code>wgauss</code> dataset	68
4.13	Timing study for the Isabel dataset	69
4.14	Extracting the eye of the hurricane in the Isabel dataset	69
4.15	Scaling study using the 3D <code>wgauss</code> and <code>wgauss_multi</code> datasets	75
4.16	Feature extracted from common 3D datasets	77
5.1	Ascending and descending manifolds of a 2-saddle	94
5.2	Persistence plots of cosmic web datasets	94

5.3	Modeling filaments as ascending paths of 2-saddles connecting two extrema	98
5.4	Density rendering of the snapshots in the Voronoi evolution time-series	103
5.5	Filamentary structures extracted using Felix	110
5.6	Filamentary structures extracted using DisPerSE	110
5.7	Comparing classification recovery rates of Felix and DisPerSE . . .	112
5.8	Two classes of filaments in the <i>CosmogridA</i> dataset	116
5.9	Three classes of filaments in the <i>CosmogridB</i> dataset	119
5.10	Using Felix for volume rendering enhancement	120
6.1	Differential geometry basics	126
6.2	Surface alignment pipeline	127
6.3	Partially overlapping surfaces	134
6.4	RMSD graphs for performance analysis	136
6.5	Study of runtime breakup of MS3ALIGN	137
6.6	Thermolysin inhibitor ligand structures	139
6.7	Alignments found by MS3ALIGN and not SURFCOMP (Thermolysin dataset)	140
6.8	Alignment of the thermolysin inhibitor ligands in 1THL and 1TMN	141
6.9	Other relevant alignments of ligands in 4TMN, 5TMN, and 6TMN .	142
6.10	Dihydrofolate reductase ligands	143
6.11	Alignments found by MS3ALIGN and not SURFCOMP(DHFR dataset)	144
6.12	RMSD of alignments found by SURFCOMP and MS3ALIGN	145
6.13	Visual verification of alignments	147
6.14	Validation dataset RMSD	148
6.15	Aligning Cryo-Electron Microscopy datasets	149

List of Algorithms

4.1	COMPAREDCELLS	38
4.2	COMPARECELLS	39
4.3	ASSIGNGRADIENT	41
4.4	ASSIGNGRADIENT2	43
4.5	COMPARECELLSPOSTPAIR	46
4.6	CONNECTSADDLES	53
5.1	SELECT2SADDLES	98

Chapter 1

Introduction

A common form of scientific data is the scalar field: a numerical measure of an interesting quantity over a domain of study. Figure 1.1 shows two examples of scalar fields. Figure 1.1(a) shows the wind speed computed from a simulation [1] of the hurricane Isabel that struck the coast of Florida, USA, in September of 2003. Figure 1.1(b) shows the electrostatic potential over the surface of the dihydrofolate reductase enzyme obtained from the Protein Data Bank [2] (Protein Data Bank ID=1AOE) computed and visualized using PyMol [3]. Often, these scalar fields contain features of interest that need to be identified and extracted, such as the region corresponding to the eye of the hurricane shown in Figure 1.1(a). A common approach for direct comprehension of such features is by *visualization*: the process of transforming data into an image, so as to convey the key features present in the data graphically. In Figures 1.1(a) and 1.1(b), smaller values of respective scalar fields are mapped to blue shades, intermediate values are mapped to white shades, and higher values are mapped to red shades. As is seen in both figures, the key features of the data set are easily visually communicated by the color mapping. With easy access to modern computing resources, datasets of increasing size and complexity are becoming commonplace. Such developments allow for

analysis of datasets that are rich in topological features. An example of this is the study of mixing of fluids by Laney *et al.* [4], where the rich set of topological features are correlated to the formation of bubbles during mixing. In such feature rich datasets, direct visualizations become insufficient for conveying and studying features. Thus, tools and techniques for analysis and summarization become crucial for comprehending features being studied in the data. This thesis explores the computation and applications of the *Morse-Smale complex*: a topological data structure for the analysis of features in scalar fields.

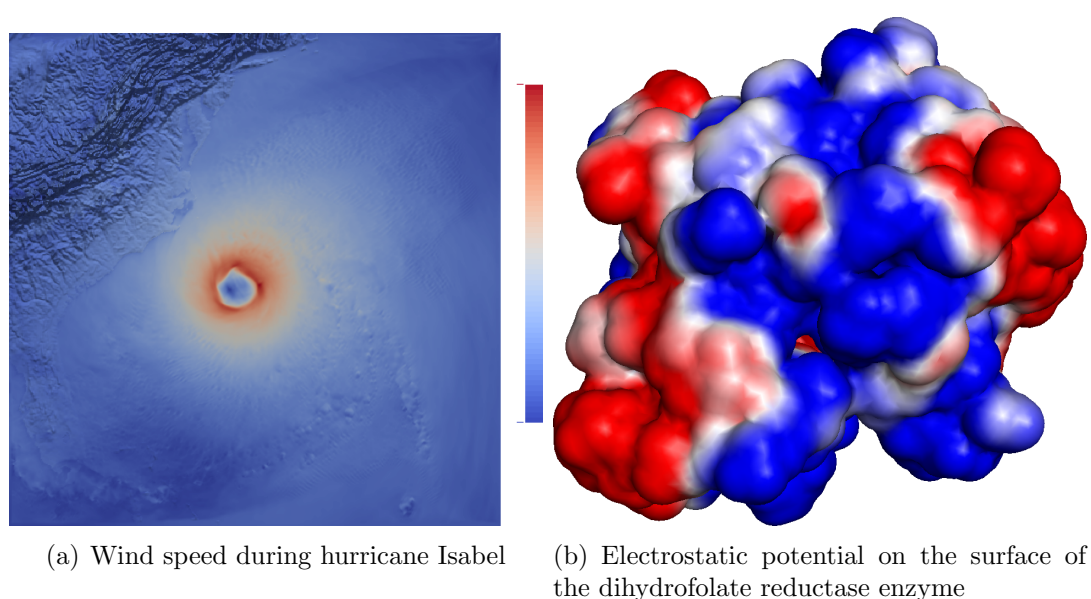


Figure 1.1

1.1 The Morse-Smale Complex

The Morse-Smale complex is a topological data structure that is constructed based on the *gradient* of a given scalar field. Informally, the gradient at any point of the domain captures the direction along which the change of the scalar field is maximized. Figure 1.2(a) shows an example of gradients for the scalar field

computed by the analytic function $\sin(x) + \sin(y)$. *Integral lines* passing through a given point are constructed by integrating along and against the gradient. Integral lines originate and terminate at *critical points* where the gradients vanish. The Morse-Smale complex is defined as the partition of the domain into *cells* whose integral lines share a common origin and terminal. Figure 1.2(b) shows an example of a Morse-Smale cell of the function shown in Figure 1.2(a) formed by a set of integral lines that share a common origin and terminal.

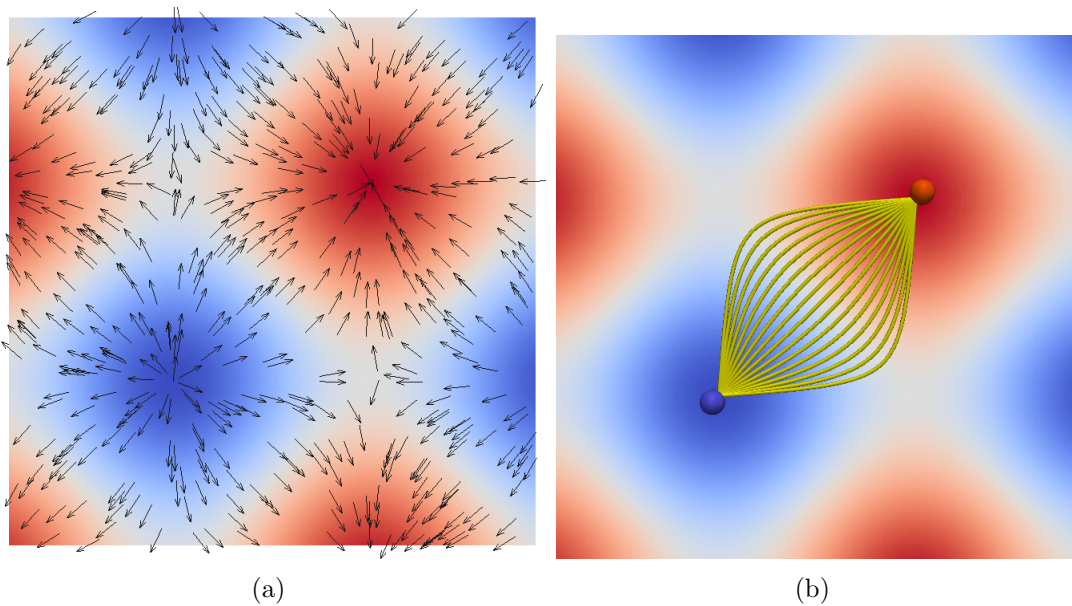


Figure 1.2: (a) An analytic scalar function given by $\sin(x) + \sin(y)$ and its gradient at a few randomly selected points (black arrows). (b) A Morse-Smale cell whose integral lines (yellow lines) share a common source (blue sphere) and destination (red sphere).

In recent years, the Morse-Smale complex has been used in a variety of scientific domains and applications, such as material sciences [5], fluid dynamics [4], shape analysis [6, 7], terrain modeling [8], cosmology [9, 10], and molecular bonding analysis [11]. In Section 3.2, we briefly discuss some exemplary applications to illustrate how the Morse-Smale complex was used. In particular, Laney *et al.* [4]

described an application of the Morse-Smale complex to analyze mixing of fluids using hydrodynamic simulations. They analyzed statistics of topological features, which are manifestations of bubbles formed during the mixing process. Due to the richness of topological features in this dataset, and the unavailability of efficient methods for 3D Morse-Smale complexes at that time, they were forced to restrict their analysis using 2D Morse-Smale complexes. Subsequent developments focused on efficient techniques [12, 13, 14] for computing Morse-Smale complexes. These developments have enabled several applications, some of which are cited above. Also, the topological features derived from the Morse-Smale complex are often not directly useful. They need to be filtered and/or simplified, which often results in analysis procedures that are refined over several iterations. Thus, efficiency in technology for 3D Morse-Smale complexes is vital for fueling such analysis. In recent years, with the advent of GPU systems and Multi-Core CPU architectures, algorithms that accommodate parallelization and out-of-core processing become increasingly important when scaling up analysis to large datasets. In the following section, we outline our contribution for efficient contribution of Morse-Smale complexes. We also briefly outline our contributions towards their simplification. Finally, we discuss our contributions towards application of Morse-Smale complexes for analysis of features in two applications.

1.2 Contributions

This thesis focuses on the following aspects of the Morse-Smale complex.

1. Efficient computation of the Morse-Smale complex.
 - (a) Algorithm for 2D in-memory and out-of-memory computation [15].
 - (b) Algorithm for 3D domains [16].

- (c) Open-Source software implementations of the above algorithms [17]
 - (d) Theoretical contributions for simplification.
2. Applications of the Morse-Smale complex.
- (a) Topology based visual exploration Framework for cosmic filaments [18].
 - (b) Aligning molecular surfaces using Morse-Smale complexes [19].

1.2.1 Efficient Computation

We design efficient algorithms for computing the Morse-Smale complex in two and three dimensions. Following the approach of earlier algorithms [7, 12, 20] for direct parallelization poses significant problems in terms of the consistency of the computed complex (see Section 3.1). In our approach, we develop an alternate characterization of the Morse-Smale complex, which is amenable for parallel computation. We also develop upon existing theory on simplification of Morse-Smale complexes.

2D algorithm

We develop a parallel algorithm for computing the Morse-Smale complex for two-dimensional domains. We prove two key lemmas that allow us to process data using desktop parallel processors (GPUs and multi-core CPUs) and develop a divide and conquer framework for data that does not fit in memory. The first lemma allows the computation of the discrete gradient in parallel. The second lemma allows us to split datasets into sub-domains, compute the Morse-Smale complex of individual sub-domains independently, and subsequently merge them to accurately form the Morse-Smale complex of the entire domain. We evaluate the algorithm's performance using simulated and real world datasets. We observe

a near linear scaling of running times with respect to data size and number of processor cores.

3D algorithm

We extend the algorithm for two-dimensional domains to work with scalar functions defined on three-dimensional domains. The extension is non trivial because of the additional structure present due to the two type of saddle critical points in three dimensions. The key problem that arises in 3D and not 2D is that the gradient structure, defined by the discrete Morse-theoretic approach [21], possesses the structure of a directed acyclic graph, as compared to a simpler tree structure in 2D. Efficiently traversing this graph to reveal the structure of the Morse-Smale complex is addressed by a priority queue based algorithm that allows for a hybrid computation of the Morse-Smale complex using both the CPU and the GPU. Again, we evaluate the performance of both using simulated and real world datasets. We observe a near linear scaling in running time with respect to data size. We also observe that the algorithm outperforms the existing methods of Gyulassy *et al.* [12] and Günther *et al.* [14].

Open Source Implementation

The above algorithms are implemented in C++ using the OpenCL framework for deployment on massively parallel systems. Furthermore, the software is written with a Python [22] interface to allow for interactive data analysis and visualization. The source code is released under the GNU Lesser General Public License [23] and is available at <http://vgl.serc.iisc.ernet.in/mscomplex/>. The website also contains a video that demonstrates the usage of the software, along with some simple code samples to illustrate its usage.

Simplification guarantees

We present theoretical results relating to the simplification of the Morse-Smale complex. Simplification is the process of systematically eliminating pairs of critical points from the Morse-Smale complex, so that it contains fewer critical points. The theory of persistence [24] provides a topological foundation for this process for 2D Morse-Smale complexes. We show that it is possible to eliminate extremum critical points in the 3D Morse-Smale complex consistent with persistence.

1.2.2 Applications of the Morse-Smale complex

We employ the Morse-Smale complex to develop two novel applications. The first application is an exploration tool for cosmic filaments in cosmological density distributions. In the second application, the Morse-Smale complex is used to develop an efficient molecular surface alignment tool, MS3ALIGN.

Topology based Visual Exploration Framework for cosmic filaments

The universe has a web-like appearance, commonly known as the *cosmic web* [25], at very large scales spanning distance from a few megaparsecs to a few hundred megaparsecs¹. The cosmic web is composed of huge blobs called *clusters*, which contain thousands of galaxies. From these clusters, gigantic tentacles containing thousands of galaxies spread out. These tentacles, known as *cosmic filaments*, serve as transport channels for mass flow in the universe. The cosmic web also consists of huge sheet-like collections of galaxies commonly known as *walls*. The clusters, filaments, and walls of the web are intertwined around vast near-empty *cosmic voids*. Cosmic filaments form unique structural elements, in that they may

¹A parsec is the standard unit of measurement of distances in the cosmos. A parsec is 3.26 times the *light-year*, the distance light covers in a year. A megaparsec is a million parsecs.

be found in a wide range of density and spatial scales. Indeed, it is this aspect that makes their study both interesting and challenging.

We develop Felix: a topology based framework for visual exploration of cosmic filaments. Here, the filaments are modeled as geometric elements associated with the Morse-Smale complex. In particular, the framework is designed to accommodate a visual inquiry into the Cosmic filaments in multiple density ranges and spatial scales. The query framework allows for user-specified density range parameters to be adjusted to visually correlate the extracted set of filaments. Specifically, this framework uses a hierarchy of Morse-Smale complexes that is generated by incrementally simplifying the Morse-Smale complex of the density field. The extracted set of filaments may be obtained simultaneously from any Morse-Smale complex within this hierarchy. A key challenge that is overcome in this application is in terms of providing for an efficient query framework to support such a query.

We demonstrate the usefulness and the applicability of Felix via two primary case studies. First, we develop a rudimentary structure classifier that identifies galaxies as belonging to cosmic filaments and clusters or not. This is validated against the heuristic Voronoi-Kinematic simulation datasets (see Section 5.5), for which such a-priori classifications are already available. We show that we can visually isolate and extract a qualitatively relevant set of filaments. These classifications are quantitatively shown to be similar and in many cases better than those extracted using other structure classifiers. A video demonstrating the visual exploration process in this experiment is available at <https://youtu.be/8TRtX937Xjk>.

In the second case study, we demonstrate the visual extraction of different classes of cosmic filaments within the Cosmogrid simulation dataset (see Section 5.5). Using a structured grid realization of the simulation, we investigate two classes of filaments within this dataset. First, we extract those within the

high density regions, which form the “spine” of the cosmic web. Next, we identify significant cosmic filaments within the low density regions, which formed the spine in earlier cosmic epochs. This highlights a key advantage of Felix, which helps to extract and isolate such filaments that is not easily possible using earlier methods. Using another realization of this simulation with tetrahedral meshes, we demonstrate its usefulness to zoom into a specific region of the simulation.

The Multi-Scale Morse-Smale Molecular Surface Aligner

In structural biology, aligning and registering similar structures is often a first step to analyze structurally similar molecules. With the wealth of 3D scanning techniques such as X-Ray crystallography and cryo Electron-Microscopy, it has been possible to create vast freely available repositories of molecular structures such as the Protein Database [2] and the Electron-Microscopy Database [26]. Often, such datasets, geometric entities such as the molecular surfaces are readily available representations of the underlying molecular surface. Thus, direct alignment of such geometric representations provides for a data-minimal approach, which allows for two advantages. First, the required representation is easily computable from the underlying scanned data. Second, it is possible to align across representations from different scanning methodologies, where the underlying geometric structures are preserved.

In this application, we develop MS3ALIGN, a method (and the implementation of it) to align molecular surfaces using the Morse-Smale complex. The tool leverages the fact that, molecular surfaces often contain several significant protrusions that are manifestations of groups of atoms, such as aromatic rings and amino groups, close to the exterior surface. Given the input of two molecular surfaces, the mean surface curvature of the surfaces are computed, and the Morse-Smale complex is employed to identify and segment significant protrusions. Correspondences across

a pair of surfaces are established based on the similarity of mean curvature values averaged over a range of spatial scales. A product graph is constructed based on the pairwise geometric consistency of correspondences. Cliques within this graph are used to compute an alignments. The alignments are evaluated an RMSD based measure, and the best few are reported.

We evaluate MS3ALIGN in three settings. First, we establish performance using known alignments with varying overlap and noise values. Second, we compare the method with SURFCOMP [27], an existing surface alignment method. Third, we validate the ability of MS3ALIGN to determine alignments in the case of structurally dissimilar binding sites. Fourth, we demonstrate the ability of MS3ALIGN to align iso-surfaces derived from cryo-electron microscopy scans.

An open source implementation of the MS3ALIGN tool and a webserver that uses this implementation is available at <http://vgl.serc.iisc.ernet.in/ms3align/>. The source code is released under the GNU Lesser General Public License [23].

Chapter 2

Background

This chapter reviews the necessary background on Morse theory and discrete Morse theory required for the remainder of the thesis. For a more detailed exposition on the concepts of Morse theory, the reader is redirected to the book by Matsumoto [28]. For the concepts on discrete Morse theory, more details may be obtained from the publication by Forman [21]. On the definition of simplicial complexes and their Betti numbers, the reader is redirected to the book by Munkres [29] for more details.

2.1 Morse Theory and the Morse-Smale complex

Consider a smooth scalar function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. The *gradient* of f at a point x is defined as the vector of first order partial derivatives,

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}(x), \frac{\partial f}{\partial x_2}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right)$$

A point $p \in \mathbb{R}^n$ is called a *critical point* with respect to f if the *gradient* of f , is identically zero at p . The *Hessian* of f at a point x is defined as the matrix of

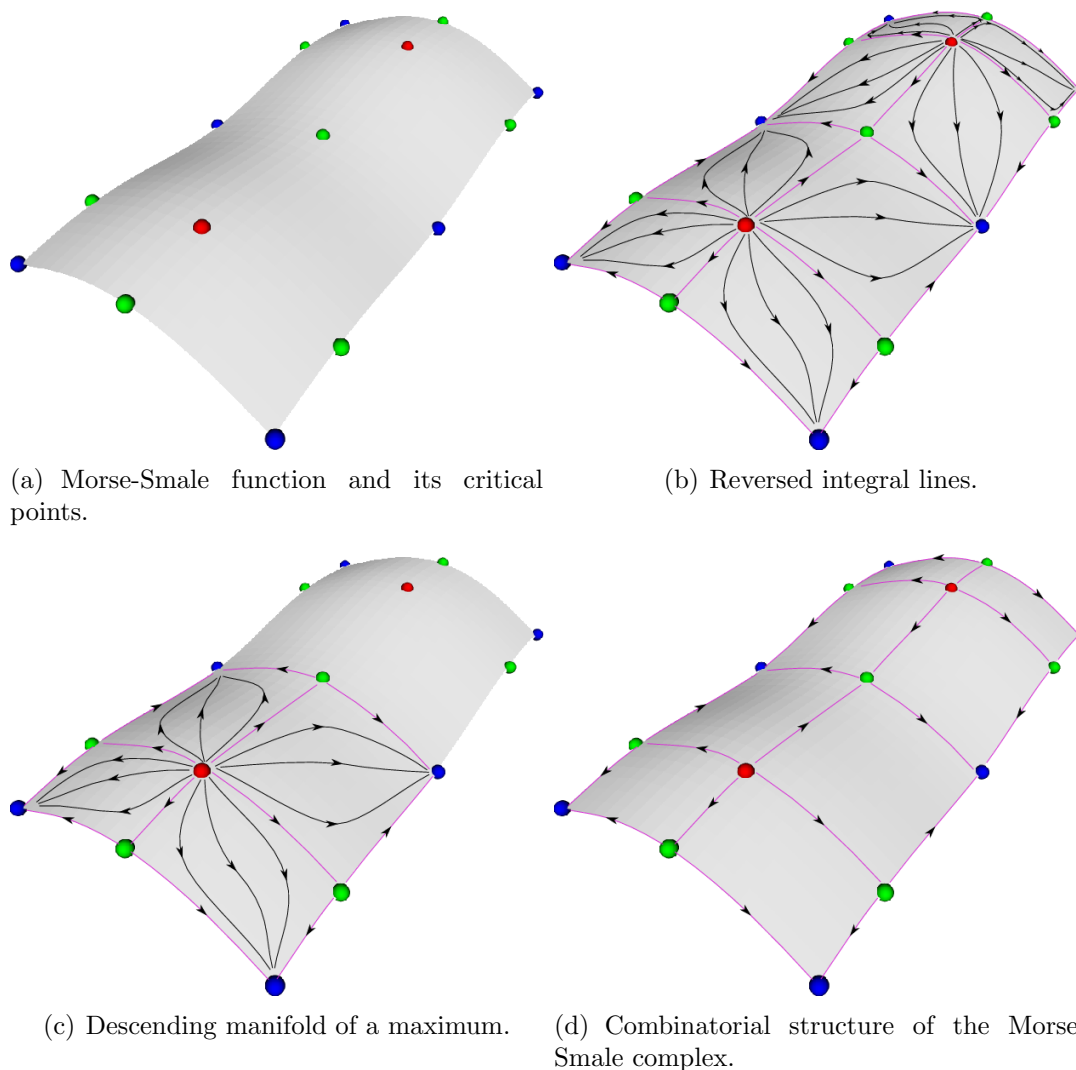


Figure 2.1: (a) A Morse function shown as a height field over a two dimensional domain. Critical points are shown in red, green, and blue corresponding to maxima, saddle, and minima, respectively. (b) The reversed integral lines of the function over the surface. (c) The descending manifold of a maximum shown as the closure of the set of reversed integral lines that originate from the critical point. (d) Combinatorial structure of the Morse Smale complex where nodes are critical points and connecting integral lines are arcs.

second order partial derivatives and is given by

$$\mathbb{H}_f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \frac{\partial^2 f}{\partial x_1 \partial x_2}(x) & \frac{\partial^2 f}{\partial x_2^2}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_2}(x) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \frac{\partial^2 f}{\partial x_2 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n^2}(x) \end{bmatrix}$$

A critical point is non-degenerate if the Hessian of f is non-singular. The function f is a *Morse function* if all of its critical points are non-degenerate (see Figure 2.1(a)).

The *index* of a critical point is the number of negative eigenvalues of the Hessian matrix. An *integral line* passing through a point p is a one-dimensional curve $l : \mathbb{R} \rightarrow \mathbb{R}^n$, where $\frac{\partial}{\partial t}l(t) = \nabla f(l(t))$, $\forall t \in \mathbb{R}$ and $l(0) = p$. In other words, it is a maximal curve in \mathbb{R}^n whose tangent at every point equals the gradient of f at that point (see Figure 2.1(b)). The function f increases along the integral line. The limit points of integral lines, $t \rightarrow \pm\infty$, are the critical points of f .

The set of all integral lines that share a common source $p = \lim_{t \rightarrow -\infty} l(t)$, together with the point p , is called the *ascending manifold* of p and the set of all integral lines that share a common destination $p = \lim_{t \rightarrow \infty} l(t)$, together with the point p , is called the *descending manifold* of p (see Figure 2.1(c)). The ascending manifolds of all critical points partition the domain. Similarly the descending manifolds of all critical points also partition \mathbb{R}^n . The *Morse-Smale complex* is a partition of \mathbb{R}^n into cells formed by the collection of integral lines that share a common source and a common destination.

The ascending manifold of a critical point of index d is a $(n - d)$ -dimensional manifold, where as its descending manifold is an n -dimensional manifold. A Morse function f is called a *Morse-Smale function* if all ascending and descending manifolds of two critical points intersect transversally. For example, if the index of two critical points differ by one, then their ascending / descending manifolds either do

not intersect or intersect along a 1-dimensional manifold connecting the critical points. In this case, the 1-manifold that connects the two critical points is referred to as an *arc*. The *combinatorial structure* or *1-skeleton* of the Morse-Smale complex is a graph whose *nodes* are the critical points and edges exist between the nodes if the corresponding critical points are connected by an arc (see Figure 2.1(d)).

2.2 Simplification

A Morse-Smale function f can be simplified to a smoother function by repeated application of a cancellation operation that removes a pair of critical points connected by an arc in the Morse-Smale complex. This cancellation corresponds to the removal of the feature represented by the critical point pair. Features are ordered based on the notion of *persistence*, equal to the absolute difference in function value between the two critical points. Persistence measures the importance of a critical point pair [24]. Edelsbrunner *et al.* [30] show that the least persistent critical point pair is always connected by an arc in the 2D Morse-Smale complex.

Simplification of a pair of critical points can be achieved by a local smoothing of the function in the neighborhood of the two critical points, more precisely within the ascending / descending manifolds containing the critical points. The cancellation is realized by updating the 1-skeleton of the Morse-Smale complex. For example, consider the case of a two-dimensional Morse-Smale function after a maximum-saddle cancellation. The 1-skeleton is updated by deleting the two nodes, deleting the arcs incident on the saddle, and re-routing the arcs incident on the maximum to the surviving maximum adjacent to the saddle (see Figure 2.2). The embedding of a new arc is obtained by extending the old arc along the arc between the maximum and saddle. We allow only those cancellations that can be realized by a local smoothing of the function. This is feasible if the pair of critical

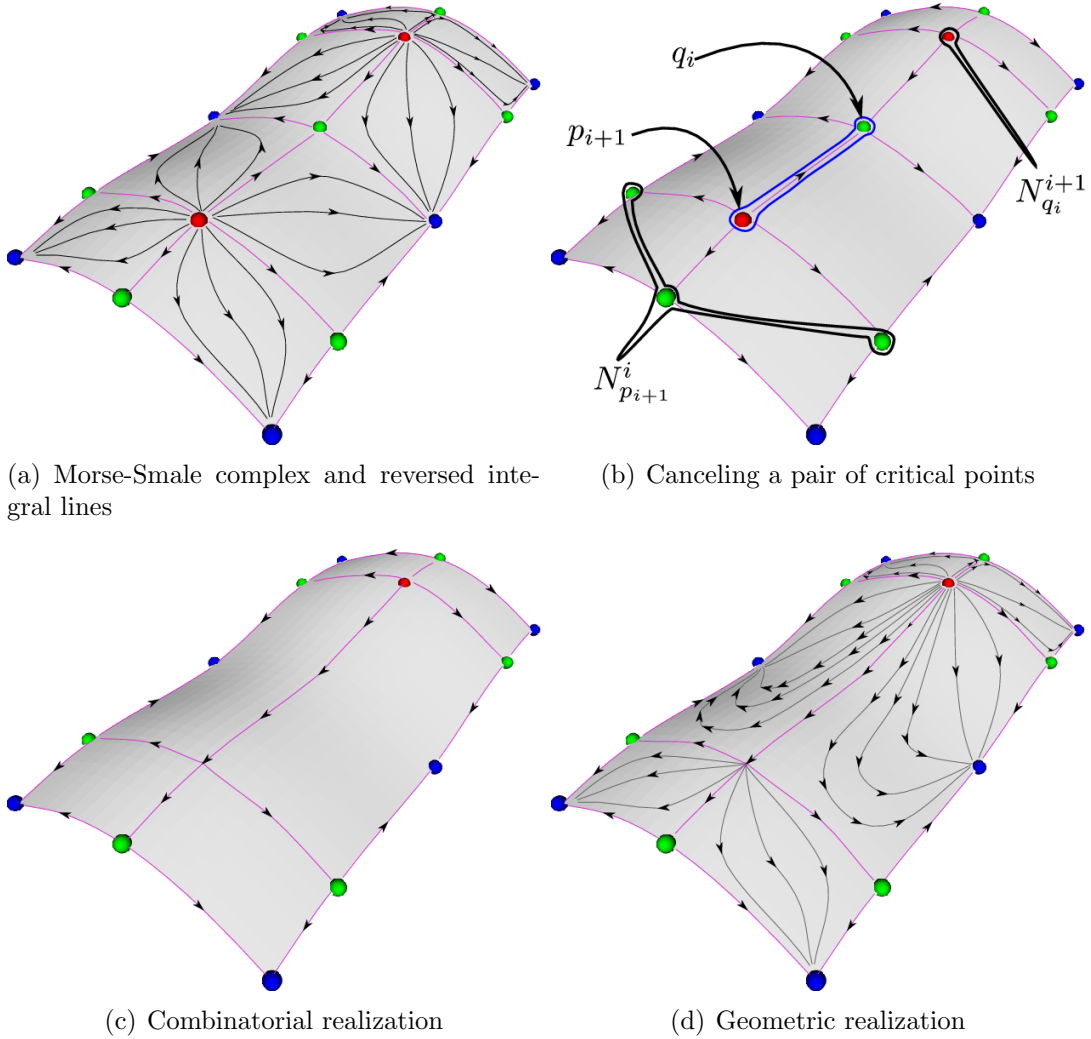


Figure 2.2: (a) Morse-Smale complex for a simple height function. (b) Canceling a pair of critical points, q_i, p_{i+1} , of index $i, i + 1$ that are connected by a single 1-manifold. (c) Combinatorial realization: connect all index i critical points ($N_{p_{i+1}}^i$) that are connected to p_{i+1} except q_i , to index $i + 1$ critical points ($N_{q_i}^{i+1}$) that are connected to q_i except p_{i+1} . (d) Geometric realization: compute the union of the descending manifold of p_{i+1} with the descending manifolds of all index $i + 1$ critical points connected to q_i . Compute the union the ascending manifold of q_i with the ascending manifolds of all index i critical points connected to p_{i+1} .

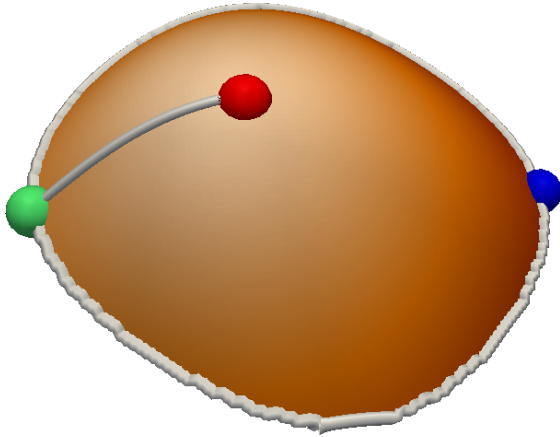


Figure 2.3: An example of a strangulation. The function is shown as a color mapped height field. Here, the saddle (green sphere) is connected to the minimum (blue sphere) via two arcs.

points is connected by a single arc. A pair of critical points that are connected by two distinct arcs in the Morse-Smale complex results in a *strangulation*. Such a pair cannot be canceled as the resulting function cannot be realized by a local smoothing. Figure 2.3 shows an example of a strangulation.

2.3 The hierarchical Morse-Smale complex

A sequence of cancellations results in a hierarchical sequence of Morse-Smale complexes $MSC_0, MSC_1, \dots, MSC_n$, where each Morse-Smale complex is a simpler version of the preceding Morse-Smale complex containing fewer critical points. Morse-Smale complex MSC_i is said to be coarser than MSC_j if $i > j$ and finer if $i < j$. The version index i enumerates the Morse-Smale complexes in the hierarchy. Each non-zero version of the Morse-Smale complex, MSC_i , is associated with the absolute difference in function value, t_i , of the pair of critical points canceled in the preceding version, MSC_{i-1} . As each iteration selects the pair of critical points with the least absolute difference in function value, the sequence of t_i 's is monotonically increasing i.e. $(t_0 = 0) \leq t_1 \leq t_2 \dots \leq t_n$. For completeness of the sequence, t_0 is set to zero. Figure 2.4 illustrates a hierarchy of Morse-Smale complexes.

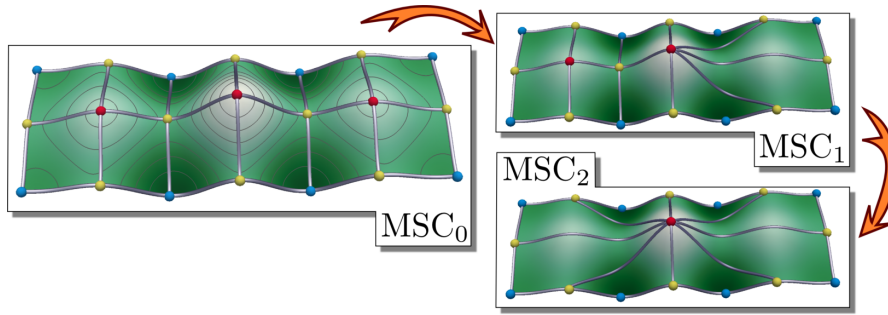


Figure 2.4: A hierarchical Morse-Smale complex. (*left*) The combinatorial structure of the Morse-Smale complex, MSC_0 , of a function with three maxima. The critical points are shown as spheres, blue for minima, yellow for saddle, and red for maxima. The arcs are shown as purple tubes. (*right*) Two cancellation operations applied to the Morse-Smale complex eliminates two maximum saddle pairs connected to the central maximum. Each cancellation operations re-routs arcs from the maxima connected to the canceled saddle to the saddles connected to the canceled maximum. The two cancellations result in two successive versions of the Morse-Smale, MSC_1 and MSC_2 respectively. This sequence of versions referred to as the hierarchical Morse-Smale complex.

It is not necessary to explicitly store all versions of the hierarchy of Morse-Smale complexes. Instead, the combinatorial representation of only MSC_0 is computed initially. Subsequently, MSC_i can be obtained from a finer MSC_{i-1} by performing topological cancellation (see Section 2.2). Analogously, MSC_i is obtained from a coarser MSC_{i+1} by applying the inverse operation of a topological cancellation.

2.4 Discrete Morse theory

Discrete Morse theory was developed by Forman [21] to study the topology of cell complexes. A d -cell α^d is a topological space homeomorphic to a d -ball $B^d = \{x \in \mathbb{E}^d : |x| \leq 1\}$. For example, a vertex is a 0-cell, an edge between two vertices is a 1-cell, a polygon is a 2-cell, and in general a d -dimensional polytope is a d -cell. We will restrict our attention to cells of the above kind, which can be represented by a set of vertices. A cell α is a *face* of β , denoted $\alpha < \beta$, if α is represented by a

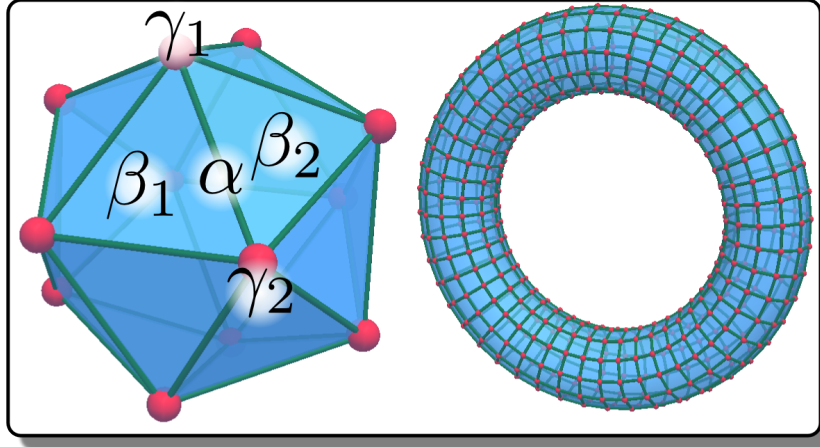


Figure 2.5: Examples of 2D cell complexes. (*left*) Icosahedron formed from 12 vertices (red spheres), 30 edges (green), and 20 triangles (blue). The edge marked α has two cofaces, β_1 , β_2 , and two faces, γ_1 , γ_2 . For a d cell, its cofaces with dimension $d + 1$ are called cofacets, and faces having dimension $d - 1$ are called facets. Here, α has two cofacets, β_1 , β_2 , and two facets, γ_1 , γ_2 . Also, β_1 and β_2 are cofaces of γ_1 , but not cofacets of γ_1 . (*right*) a torus formed by a set of vertices, edges, and quad 2D elements.

subset of vertices of β . The cell β is called a *coface* of α . A face α is called a *facet* of β if $\alpha < \beta$ and $\dim(\alpha) + 1 = \dim(\beta)$. In this case β is a *cofacet* of α denoted by $\alpha \triangleleft \beta$. The set of zero-dimensional faces of a cell α is called the vertex set of α denoted by V_α .

A *cell complex* K is a collection of cells that satisfies two properties: (a) If α belongs to K then so do all faces of α , and (b) If α_1 and α_2 are two cells in K then either they are disjoint or they intersect along a common face. A *regular cell complex* is a cell complex in which, given two incident cells, β^{d+1} and γ^{d-1} , there are exactly two cells α_1^d, α_2^d such that $\gamma^{d-1} < \alpha_1^d, \alpha_2^d < \beta$. We consider only finite regular cell complexes. Figure 2.5 shows two examples of 2D cell complexes, with examples of an edge cell and its faces and cofaces marked. A *filtration* of a cell complex K is a sequence of nested cell complexes K_0, K_1, \dots, K_n , such that K_0 is the empty cell complex, K_n is the cell complex K , and K_i is obtained by attaching

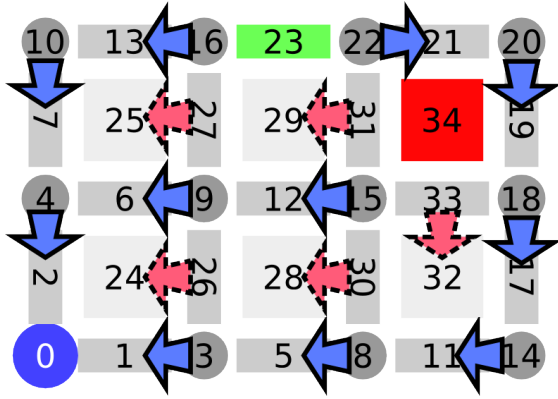


Figure 2.6: A discrete Morse function defined over a 2D cell complex. The gradient pairs are shown as arrows oriented towards the higher dimensional cell. Dashed red arrows denote edge-quad pairs and solid blue arrows denote vertex-edge pairs. Critical cells are shown in red (maxima), green (saddle), and blue (minima).

one or more cells to K_{i-1} for $i = 1..n$.

Note that a *simplex* is a d -cell which has exactly $d + 1$ vertices in its vertex set. A *simplicial cell complex* is a cell complex whose cells are d -dimensional simplices such as vertices, edges, triangles, tetrahedra and so on. A simplicial complex is also a regular cell complex.

Given a regular cell complex K representing the domain, a function $f : K \rightarrow \mathbb{R}$ is said to be a *discrete Morse function* if for all d -cells $\alpha^d \in K$,

$$|\{\beta^{d+1} \mid \alpha^d < \beta^{d+1} \text{ and } f(\beta) \leq f(\alpha)\}| \leq 1 \text{ and}$$

$$|\{\gamma^{d-1} \mid \gamma^{d-1} < \alpha \text{ and } f(\gamma) \geq f(\alpha)\}| \leq 1.$$

A cell α^d is critical if

$$|\{\beta^{d+1} \mid \alpha^d < \beta^{d+1} \text{ and } f(\beta) \leq f(\alpha)\}| = 0 \text{ and}$$

$$|\{\gamma^{d-1} \mid \gamma^{d-1} < \alpha \text{ and } f(\gamma) \geq f(\alpha)\}| = 0$$

A discrete *vector* is a pairing between two incident cells that differ in dimension by one. A *discrete vector field* on K is a set of discrete vectors such that every cell in K is represented in at most one pair of the field. A *V-path* is a sequence of cells

$$\alpha_0^d, \beta_0^{d+1}, \alpha_1^d, \beta_1^{d+1}, \dots, \alpha_r^d, \beta_r^{d+1}, \alpha_{r+1}^d$$

such that α_i^d and α_{i+1}^d are facets of β_i^{d+1} and $(\alpha_i^d, \beta_i^{d+1})$ is a vector, $i = 1..r$. A *V-path* is called a *gradient path* if it contains no cycles (see Figure 2.6). A *discrete*

gradient field is a discrete vector field that contains no non-trivial closed V -paths. We refer to discrete vectors in a discrete gradient path as *gradient pairs*.

Maximal gradient paths of the discrete Morse function correspond to the notion of integral lines of Morse functions. Ascending / descending manifolds are similarly defined for discrete Morse functions.

2.5 Piecewise Linear(PL) Functions

Earlier approaches to compute Morse-Smale complexes were based on PL extensions of functions sampled at vertices of simplicial complexes [31, 30]. Though we adopt the discrete formulation of Morse-Smale complexes for our computations, we introduce here some notions of PL function so that we may establish the closeness of our approach to the PL approach. For further reading on the basic notions of algebraic topology, we refer the reader to the classic text books by Munkres [29] and Hatcher [32].

A function f sampled at vertices of a simplicial complex may be extended to form a continuous function that is linear on every cell. The *star* of a vertex v is the set of simplices incident on v . The *link* of a vertex v is the set of faces of cells in the star of v , that are not incident on v . The *lower star* of vertex v is the set of cells in the star where the PL extension assumes values lower than $f(v)$. The *lower link* of a vertex v is the set of faces of cells in the lower star of v , that are not incident on v .

The *Betti numbers* of a cell complex, K , are a useful characterization of the underlying space of a cell complex. They are defined for each $k = 0, 1, \dots, \dim(K)$. Intuitively, the 0^{th} Betti number counts the number of components of K , the 1^{st} Betti number counts the number of tunnels in K , and the 2^{nd} Betti number counts the number of voids of K . Figure 2.7 shows an example of the Betti numbers of a

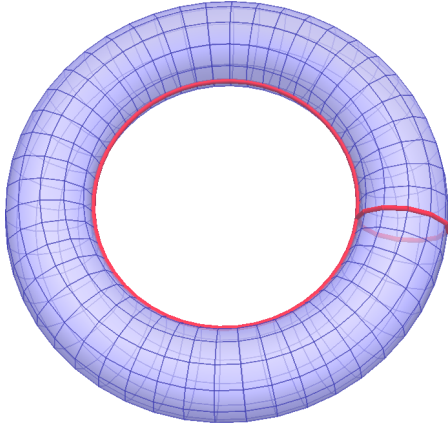


Figure 2.7: The Betti numbers of a torus. The torus has a single component. There are two tunnels that depicted by the red circles. The surface encloses a single void. Thus, the 0^{th} Betti number is 1, the 1^{st} Betti number is 2, and the 2^{nd} Betti number is 1.

simple torus. The *reduced Betti number* is defined for $k = -1, 0, 1, \dots, \dim(K)$. The k^{th} reduced Betti number is equal to the k^{th} Betti number, for all $k = 1, \dots, \dim(K)$. The 0^{th} reduced Betti number is equal to 0^{th} Betti number reduced by one. The -1^{th} reduced Betti number is equal to 1 if the 0^{th} Betti number is equal to 0, and 0 otherwise.

Reduced Betti numbers of the lower link can be used to classify a vertex of a simplicial complex as non-critical (regular) or critical and to further classify critical vertices. A vertex in a 2D simplicial complex is said to be regular if all the reduced Betti numbers of the lower link are 0, minimum if the -1^{th} reduced Betti number is 1 and the others are 0, simple saddle if the 0^{th} reduced Betti number is 1 and the others are 0, and maximum if the 1^{st} reduced Betti number is 1 and the others are 0. It is possible for saddles to have the 0^{th} Betti number greater than 1, and such saddles are called multi-saddles.

The weak Morse inequality is a classic result of Morse theory which states that, given a Morse function f defined on a manifold, the number of index k critical points of f is greater than or equal to the k^{th} Betti number [29]. Forman established the analogous result for discrete Morse functions [21].

2.6 Simulation of simplicity

Simulation of simplicity (SoS) is a programming technique that allows us to cope with degenerate data for many geometric algorithms [33]. In the context of Morse-Smale complexes we require that the function be nowhere flat to ensure non-degeneracy. Implementing SoS is simple in this case. We simulate a non-degenerate function using the available order of vertices in the storage device and hence consistently resolve comparisons when function values at two vertices are equal. Hence, we assume that $f(x) \neq f(y)$ for all vertices $x \neq y$.

Chapter 3

Related Work

This chapter discusses related work with respect to computation and selected applications of the Morse-Smale complex. Section 3.1 discusses prior work related to the computation of Morse-Smale complexes and contrasts the works with the contribution of this thesis. Section 3.2 discusses selected applications that apply Morse-Smale complexes for various tasks. In particular, the discussed applications are contrasted against the applications discussed in this thesis to provide a perspective on the novel usage of Morse-Smale complex introduced in this thesis. The arguments for the usage of Morse-Smale complex for each application are discussed in the respective chapters.

3.1 Computation

Morse-Smale complexes were initially introduced in the context of dynamical systems [34, 35]. Several algorithms have been proposed to compute the Morse-Smale complex of scalar functions defined on cell complexes. Broadly, they fall into two classes. The first is based on the notion of quasi Morse-Smale complexes, introduced by Edelsbrunner *et al.* [30]. The second is based on a discrete formulation

of Morse theory introduced by Forman [21]. Table 3.1 provides a brief comparison of the prior algorithms for the computation of the Morse-Smale complex. In particular, we remark on the salient shortcomings of these approaches.

Edelsbrunner *et al.* [30] introduced the notion of quasi Morse-Smale complex for two-dimensional piecewise linear functions. They computed the cells of the Morse-Smale complex while restricting the bounding arcs of the complex to edges of the input mesh. This approach was extended to construct Morse-Smale complexes of three-dimensional functions [31]. These early approaches traced the gradient paths from saddle critical points and produced a boundary representation of cells in the Morse-Smale complex. Bremer *et al.* [36] also followed this approach and developed a multi-resolution representation of 2D scalar functions via controlled topological simplification, and demonstrated the application of the Morse-Smale complex to feature identification, noise removal, and view-dependent simplification.

Gyulassy *et al.* [5, 37] focused on three-dimensional functions and employed an approach based on repeated cancellations of critical point pairs applied on an artificial complex created from the input mesh by including dummy critical points. The cancellations were appropriately scheduled in order to remove the dummy critical points leaving behind the true critical points and cells of the Morse-Smale complex. The order of cancellation determines the quality of the resulting Morse-Smale complex and the algorithm efficiency.

Another approach to compute the Morse-Smale complex is based on a discrete analog of Morse theory [38] introduced by Forman [21] to study discrete functions defined on cells of a cell complex. Reininghaus *et al.* [39, 40] discussed an application of discrete Morse theory to analyze vector fields. Bauer *et al.* [41] computed simplified two-dimensional scalar functions while ensuring that the input function is modified by no more than a threshold δ and all surviving critical point pairs have persistence greater than 2δ .

Method	Input	Approach	Remarks
Edelsbrunner <i>et al.</i> , 2003, [30]	2D Triangle complexes	Quasi MS Complex	a) Mesh specific interpretation of Separatrix lines. b) Sequential execution.
Edelsbrunner <i>et al.</i> 2003, [31]	3D Triangle complexes	Quasi MS Complex	a) Mesh specific interpretation of Separatrix lines/sheets. b) Sequential execution.
Gyulassy <i>et al.</i> 2006, [37]	3D Cell complexes	Repeated application of topological cancellations.	a) Cancellation order dependent. b) Sequential execution.
Cazals <i>et al.</i> 2003, [7]	2D Triangle complexes	Discrete Morse Theory	a) Sequential discrete gradient algorithm. b) Ambiguous definition of discrete Morse function.
Gyulassy <i>et al.</i> 2008, [12]	3D Cell complexes	Discrete Morse Theory. Split & merge strategy for large data.	a) Sequential discrete gradient algorithm. b) Ambiguous definition of discrete Morse Function. c) Expensive gradient field traversal step.
Robins <i>et al.</i> 2011,[13]	3D Cell complexes	Discrete Morse Theory.	a) CPU Parallel Discrete Gradient Algorithm. b) Ambiguous definition of discrete Morse Function. c) Expensive gradient field traversal step.
Günther <i>et al.</i> 2011,[14]	3D Cell complexes	Discrete Morse Theory. Builds on approach of Robins <i>et al.</i> to improve gradient field traversal.	a) CPU Parallel Discrete Gradient Algorithm. b) Ambiguous definition of discrete Morse Function. c) Requires data sized flag buffers for each traversal

Table 3.1: A comparison of prior algorithms to compute the Morse-Smale complex.

Cazals *et al.* [7] and Lewiner *et al.* [42] successfully employed Forman’s discrete Morse theory to compute Morse-Smale complexes of piecewise-linear functions and demonstrated applications to segmentation, visualization, and mesh compression. Gyulassy *et al.* [12] also used a discrete Morse theory based formulation to develop an efficient algorithm for computing Morse-Smale complexes of large 3D data that do not fit in main memory. They partition the data into blocks called “parcels” that fit in memory, compute gradient flows on the boundary of the parcels, propagate the flows to the interior and compute the Morse-Smale complex restricted to a parcel. Critical cells that are created on the boundary are canceled during a subsequent merge step resulting in the Morse-Smale complex of the union of the parcels. This serial method scales well for large data but the geometry of the Morse-Smale complex is sensitive to the order of cancellations chosen during the merge step. Furthermore, both methods do not use a robust definition of the discrete Morse function. Thus, the gradient pairs are also potentially ambiguous and therefore inconsistent.

Robins *et al.* [13] proposed an algorithm to compute the Morse complex of 2D and 3D gray-scale digital images modeled as discrete functions on cubical complexes. The algorithm computes the Morse complex with provable guarantees on its correctness with respect to the critical cells. The algorithm focuses on determining discrete gradient pairs so that the set of critical cells that remain unpaired correspond to the set of piecewise linear critical vertices. However, this leads to pairs that do not necessarily determine the direction of steepest descent and hence may result in gradient pair directions that are inconsistent with the numerical notion of gradients. Also, the pairing algorithm is not easily implementable on current generation GPUs because of the usage of dynamically allocated data structures such as priority queues. Furthermore, the algorithm does not guarantee polynomial time execution since they use a modified breadth first search algorithm

Method	Input	Approach	Remarks
Shivashankar <i>et al.</i> 2012,[15]	2D Cell complexes	Discrete Morse Theory.	a) GPU parallel discrete gradient algorithm. b) Defines a totally ordering discrete Morse function based on the steepest descent notion.
Shivashankar <i>et al.</i> 2012,[16]	3D Cell complexes	Discrete Morse Theory.	a) Extended GPU parallel discrete gradient algorithm. b) Memory efficient traversal procedure that only requires the frontier in memory

Table 3.2: The contributions of this thesis towards efficient Morse-Smale complex computation.

that traverses all possible paths between two nodes, which can be exponential in the number of nodes.

Günther *et al.* [14] described a memory efficient algorithm using the gradient algorithm of Robins *et al.* [13]. In their improvement, they ensure that the computational complexity of the traversal is quadratic instead of exponential. They do this by maintaining the set of visited nodes and ensure that each node is processed only after all possible paths have arrived to it. This necessitates the maintenance of a container that potentially grows the size of the entire descending / ascending manifold of a critical point. Hence, it becomes memory inefficient when deployed in parallel.

More recently, Peterka *et al.* and Gyulassy *et al.* [43, 44] introduced a set of building blocks for implementing parallel algorithms, which leverage high performance computing clusters. In particular, they discuss a parallel implementation of the discrete Morse theory based algorithm proposed by Gyulassy *et al.* [12] using their framework. They deploy the method on large supercomputing clusters and observe the near linear scaling behavior with data-size (discounting IO).

Table 3.2 provides a brief summary of the contributions of this thesis in comparison to the discrete Morse-Smale complex algorithms discussed in Table 3.1. A more detailed discussion of the differences with respect to the approach of earlier algorithms is discussed in Section 4.3.

3.2 Applications

This section briefly discusses a few selected applications of Morse-Smale complexes in various domain sciences. In particular, the specific usage of the Morse-Smale complex is highlighted. This serves to highlight some of the key ways in which Morse-Smale complexes are useful for scientific data analysis. Also, some of the novel usages of the Morse-Smale complex in the applications discussed in this thesis are highlighted here. Table 3.3 presents a brief comparison of the essential structures used from the Morse-Smale complex for each application, as well as a comparison of the simplification methodology adopted.

Laney *et al.* [4] discuss an application of 2D Morse-Smale complexes for the analysis of hydrodynamic simulations. In particular, they analyze the turbulent mixing layer between two liquids with different densities upon the application of pressure. The mixing layer is extracted as a 2D slice from the density field and the Morse-Smale complex of the resulting field is computed. The statistics of minima and maxima are correlated with known phases in the mixing of such fluids. Furthermore, the segmentation of the mixing layer is used for a qualitative analysis of the creation and destruction of bubbles across the simulation time span.

Gyulassy *et al.* [5] discuss an application of the 3D Morse-Smale complex to compute the core structure inside a porous medium. Here, the core structure is represented as ascending arcs of 2-saddles in the Morse-Smale complex of a signed distance field that captures the structure of the porous medium. The Morse-Smale

complex is simplified so that 2-saddle maximum pairs that do not entirely cross a range of function values are simplified away. The resulting sub-complex formed by the 2-saddle-maxima connections in the Morse-Smale complex, along with their geometric embedding form the “core” of the porous media. In a simulation of the impact of a foreign body on the porous medium, they present statistics pertaining to the number of loops that survive the impact.

Sousbie *et al.* [9, 10] describe an application of the Morse-Smale complex to analyze the large scale structure of the Universe. The large scale Universe consists of collections of billions of galaxies that organize themselves into *cosmic clusters*, gigantic tentacle like structures known as *cosmic filaments*, and flattened wall like structures, referred to as *cosmic walls*. They surround vast regions of relatively empty regions of space known as *cosmic voids*. Sousbie *et al.* compute the Morse-Smale complex of the logarithm of the density field computed from the galaxy distribution. In it, the clusters are represented by maxima. The filaments and walls are modeled by the ascending geometry of 2-saddles and 1-saddles respectively. Voids are modeled by the ascending geometry of minima. In cosmic density distributions, one expects such features to be found in multiple density and spatial scales. Therefore, they employ a variation on the persistence simplification mechanism, where they use separate thresholds for minimum-1-saddle, 1-saddle-2-saddle, and 2-saddle-maximum pairs. The method is shown to be effective in isolating such large scale structures in several cosmological simulations and cosmological survey datasets.

Günther *et al.* [11] present an application of the Morse-Smale complex to analyze molecular bonding in chemical systems using the electron density field. In particular, they use topological analysis to classify interactions as *covalent* and *non-covalent*. Their method computes the Morse-Smale complex of a weighted version of the electron density field and analyzes the 1-saddle minimum arcs in the

complex. They simplify the Morse-Smale complex to eliminate 1-saddle-minimum pairs in three steps. First, they eliminate low persistence pairs. Next, they remove minimum-1-saddle pairs so that only minima within a prescribed density range are retained. Finally, they eliminate 1-saddles that are not within “interaction” regions, which are obtained using a pre-processing step. Using the retained 1-saddles and the embedding of their ascending geometry, they study the interactions of various chemically relevant quantities along this bonding path.

Gerber *et al.* [45] describe the application of Morse-Smale complexes for estimating regression models. In particular, given points in high dimensional space, they compute a neighborhood representation using a k -nearest neighborhood graph. They partition the data based on the source and sink of steepest monotonic paths in this graph, where steepness is measured as the absolute difference of values of the response variable at the end points of the edges. For each Morse-Smale cell, they compute a linear regression surface. Next, they design a classifier to estimate the probabilities of a given input point belonging to the Morse-Smale cells. Then, the response value of each Morse-Smale regression surface is weighted by these probabilities, to produce the final response value for the given input.

In this this thesis, we explore two applications of the Morse-Smale complex. In the first application (See Chapter 5), we develop upon the methodology of Sousbie *et al.* [9], to create a Topology based Visual Exploration Framework [18], to explore cosmic filaments. A particular highlight of this approach is that we do not directly simplify any features. Instead, features are extracted from across a hierarchy of Morse-Smale complexes based on a density scale of interest, which is different from the approach taken by the above applications for feature extraction from the Morse-Smale complex. This is in analogy with the multi-scale nature of cosmological data, where one wishes to explore features with particular density characteristics. In the second application (See Chapter 6), the Morse-Smale

Application	Features	Domain	Dim	Remarks
Laney <i>et al.</i> , [4],2006	a) Extrema Statistics b) Extrema Morse cells for feature tracking	Fluid Dynamics	2D	Persistence based simplification
Gyulassy <i>et al.</i> [5],2007	2-saddle ascending geometry	Material Sciences	3D	Persistence and function range based simplification
Sousbie <i>et al.</i> [9, 10],2011	Maxima positions, ascending geometry of 2-saddles, 1-saddles, and minima	Cosmology	3D	Persistence based simplification
Günther <i>et al.</i> [11],2014	Descending geometry of 1-saddles	Physical Chemistry	3D	Persistence + function range based + region classification based simplification
Gerber <i>et al.</i> [45],2012	Extrema Morse-Smale cells	Statistical Data Analysis	nD	Persistence based simplification
Shivashankar <i>et al.</i> ,[18],2015 [This Thesis]	2-saddle ascending geometry	Cosmology	3D	Features are extracted from a persistence hierarchy of Morse-Smale complexes
Shivashankar <i>et al.</i> ,[19],2015 [This Thesis]	Maxima descending geometry	Structural biology	2D	Persistence based Simplification.

Table 3.3: A comparison of the features extracted from the Morse-Smale complex in different applications

complex is employed to compute the segmentation of molecular surfaces to identify significant protrusions [19]. Here, the Morse-Smale complex of mean surface curvature is simplified using topological persistence.

Chapter 4

MS Complex Algorithm

In this chapter, we discuss our algorithm to efficiently compute the Morse-Smale complex, abbreviated as MS complex henceforth. Section 4.1 discusses our efficient Morse-Smale complex computation algorithm applicable to 2D and 3D domains with the assumption that datasets fit in memory. In Section 4.2, we discuss the extension of our algorithm to datasets that do not fit entirely in memory. Section 4.3 compares and highlights differences and advantages of our algorithm over other Morse-Smale complex algorithms. Sections 4.4 and 4.5 discuss the implementation and evaluation of our algorithm for 2D and 3D structured grids. In Section 4.6, we discuss our contributions towards simplification of the Morse-Smale complex. We conclude this chapter with Section 4.7.

4.1 In-core Algorithm

We now describe our algorithm to compute the MS complex. In this section we assume that the dataset fits entirely in memory. Section 4.2 discusses the extension of the method to datasets that do not fit in memory. We first describe a canonical extension of scalar functions sampled at vertices to discrete Morse

functions and demonstrate why it is not a suitable extension for computing the MS complex. Next, we introduce a weighted discrete Morse function which satisfies a key property leading to an algorithm that computes gradient pairs in parallel. The algorithms to compute the gradient field applicable for general n-Dimensional regular cell complexes. Thus the arguments and proofs are constructed to ensure that they work for those cases. Next, we discuss efficient algorithms to traverse the gradient field to extract the MS complex. Since the structure of the gradient field in 2D is simpler than 3D gradient fields, their traversal may be performed on the GPU. Efficient algorithms to traverse the 2D gradient field in CPUs and GPUs are discussed in Section 4.1.4. These algorithms may be directly adapted for traversing the gradient paths originating/terminating at the extrema 3D gradient fields. However this is not true for the additional 1-saddle-2-saddle sub-structure present in 3D gradient fields. In Section 4.1.4, we discuss efficient algorithms to traverse this additional sub-structure. Finally, we analyze the computed gradient field and argue for its correctness.

4.1.1 Discrete function

Given a regular cell complex K with vertex set V and a scalar function $f : V \rightarrow \mathbb{R}$, a canonical extension of f to a discrete Morse function, $F_d : K \rightarrow \mathbb{R}$, is defined recursively on a cell α as $F_d(\alpha) = \max_{\sigma < \alpha} F_d(\sigma) + \epsilon$, where $\epsilon > 0$ is an infinitesimally small real value [12]. Extending the function f in this manner results in all cells becoming critical with respect to the discrete Morse function F_d . This implies that each cell in the input is essentially a cell of the MS complex. Further, newly introduced critical cells that are incident on each other can be canceled using an infinitesimally small persistence threshold ϵ to create an ϵ -persistent MS complex. The motivation for extending the function f to F_d is that the MS complex can be computed via repeated cancellations of ϵ -persistent pairs. The collection of

ϵ -persistent critical point pairs are viewed as a pairing of incident cells or discrete gradient pairs. The pairs are represented by arrows from the lower dimensional cell to the higher dimensional cell indicating descent. These pairings constitute a discrete gradient field. Gyulassy *et al.* [12, 37] compute the MS complex via a sequence of cancellations of the ϵ -persistent critical point pairs. However, this approach does not necessarily compute paths of steepest descent. Consider the case when two cells, β_1 and β_2 , share a common facet α such that $F_d(\beta_1)$ and $F_d(\beta_2)$ are written as

$$\begin{aligned} F_d(\beta_1) &= F_d(\alpha) + \epsilon \\ F_d(\beta_2) &= F_d(\alpha) + \epsilon. \end{aligned}$$

Either one of β_1 or β_2 can be paired with α . For both pairs, the difference in value of F_d is equal to ϵ . The tie is broken arbitrarily in this case.

4.1.2 Weighted discrete function

We now describe a method to extend a given real valued function (f) on the vertex set (V) of a given mesh (K) to a function (F_w) that is defined on all cells of the mesh. We show that this function is a discrete Morse function and that it imposes a total order on the cells. Since the algorithm for computing the MS complex requires only the order between cells, we describe a symbolic comparator that does not explicitly compute the function value. We assume that the input vertices are totally ordered based on the input function specified at the vertices.

We define a weighted discrete function F_w on a d -dimensional cell α^d recursively as

$$F_w(\alpha^d) = F_w(G_0(\alpha^d)) + \epsilon^d \times F_w(G_1(\alpha^d)),$$

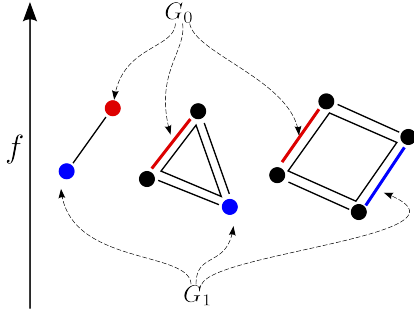


Figure 4.1: The weighted discrete function is defined recursively as a weighted sum of the function value at faces G_0 and G_1 . Faces G_0 (in red) and G_1 (in blue) for an edge, triangle and quad cells are shown. The function value at vertices increases along the vertical axis.

where ϵ is an infinitesimally small positive real number,

$$G_0(\alpha^d) = \arg \max_{\gamma < \alpha^d} F_w(\gamma), \text{ and}$$

$$G_1(\alpha^d) = \arg \max_{\gamma < \alpha^d, V_\gamma \cap V_{G_0(\alpha^d)} = \emptyset} F_w(\gamma).$$

$V_{G_0(\alpha)}$ is the vertex set of $G_0(\alpha^d)$, and $\arg \max$ denotes the value of the argument γ that maximizes the function. Similar to F_d , F_w is also equal to f at the vertices. The weighted version of the discrete function ensures that when two cells share a common face whose function value is the maximum among both face sets, then the tie is broken using the second maximum face whose vertex sets are disjoint from the above common face. See Figure 4.1 for the definition of the weighted discrete function for some common cell types. Figure 4.3(a) shows the expansion of F_w for a function sampled on a 2D grid.

$G_0(\alpha^d)$ is necessarily a $d - 1$ cell. This is because F_w of any $d - 1$ face of α^d is greater than all faces incident on the $d - 1$ cell. Thus, the $d - 1$ cell that maximizes F_w will have higher function value than all faces of α^d . Also $G_1(\alpha^d)$ must exist for all cells with $d > 0$. Theoretically, we require $F_w(G_1(\alpha^d))$ to be strictly positive to ensure that its value at cofacets is greater than at the facet. This assumption is valid if we rescale the range of f to $[0 + \delta, 1]$, $\delta \in (0, 1)$. In practice we obtain the order on the cells via a symbolic comparison and do not need to explicitly compute

F_w .

F_w is well defined and totally ordering

For F_w to be well defined we require G_0, G_1 to be unique. The cells $G_0(\alpha^d)$ and $G_1(\alpha^d)$ for a given α^d are unique if F_w induces a total order on all cells of dimension less than d . This suggests an inductive proof, which we outline below.

Since $f(x) \neq f(y)$ for all $x \neq y$, F_w is well defined and induces a total order on all zero-dimensional cells. Now, assume that F_w induces a total order on all cells of dimension less than d . We will show that we can order two cells α_1^d, α_2^d or $\alpha_1^{d'}, \alpha_2^{d'}$ where $d' < d$ and $\alpha_1 \neq \alpha_2$.

Let $G_0(\alpha_1^d) = \gamma_1$ and $G_1(\alpha_2^d) = \gamma_2$. We have $F_w(\alpha_1^d) = F_w(\gamma_1) + \epsilon^d \times F_w(G_1(\alpha_1^d))$ and $F_w(\alpha_2^d) = F_w(\gamma_2) + \epsilon^d \times F_w(G_1(\alpha_2^d))$. Assume that $\gamma_1 \neq \gamma_2$. Cells γ_1 and γ_2 have dimension less than d and can therefore be ordered. We may choose ϵ to be arbitrarily small, so that the comparison of $F_w(\alpha_1^d)$ and $F_w(\alpha_2^d)$ is dominated by the comparison of $F_w(\gamma_1)$ and $F_w(\gamma_2)$. i.e. $F_w(\gamma_1) < F_w(\gamma_2) \Rightarrow F_w(\alpha_1) < F_w(\alpha_2)$ and $F_w(\gamma_1) > F_w(\gamma_2) \Rightarrow F_w(\alpha_1) > F_w(\alpha_2)$.

If $\gamma_1 = \gamma_2$ then the second term induces an order on α_1 and α_2 . Note that if $\alpha_1^d \neq \alpha_2^d$ then $G_1(\alpha_1^d) \neq G_1(\alpha_2^d)$ because K is a cell complex. This is because if two d -dimensional cells intersect they do so along a single common cell whose dimension is less than d . The cells $\alpha_1^{d'}$ and $\alpha_2^{d'}$ can be ordered using a similar argument. Thus, the weighted discrete function F_w is well defined and induces a total order.

Implementing F_w 's total order

Subsequent algorithms require only an ordering of cells in K and not the explicit values of F_w . We describe Algorithm `COMPARECELLS` which establishes this order on all cells using only comparisons of the function f on the vertices of K . We

begin with Algorithm `COMPAREDCELLS` that orders two d -cells. Both algorithms describe a comparator that accepts two cells as input and returns true if the first cell is lesser than the other and false otherwise.

Algorithm 4.1 `COMPAREDCELLS`

```

1: procedure COMPAREDCELLS( $d, \alpha, \alpha'$ )
2:   if  $d = 0$  then
3:     return  $f(\alpha) < f(\alpha')$ 
4:    $\gamma \leftarrow G_0(\alpha), \gamma' \leftarrow G_0(\alpha')$  ▷ Both  $\gamma$  and  $\gamma'$  are  $(d - 1)$ -cells.
5:
6:   if  $\gamma \neq \gamma'$  then
7:     return COMPAREDCELLS( $d - 1, \gamma, \gamma'$ )
8:   else
9:      $\psi \leftarrow G_1(\alpha), \psi' \leftarrow G_1(\alpha')$  ▷ Both  $\psi$  and  $\psi'$  are  $d_\psi$ -cells1,
10:     $d_\psi \leftarrow \text{DIM}(\psi)$  ▷ where  $0 \leq d_\psi < d$ .
11:    return COMPAREDCELLS( $d_\psi, \psi, \psi'$ )

```

Algorithm `COMPAREDCELLS` is a recursive procedure based on the recursive definition of F_w . The base case of the recursion is handled by comparison of the scalar function value of f at the vertices of K (Line 3), where it is assumed that it is always possible to consistently order the given input scalar function [33]. Line number 11 of Algorithm `COMPAREDCELLS` assumes that ψ and ψ' have the same dimension ¹.

Algorithm `COMPARECELLS` extends Algorithm `COMPAREDCELLS` so that cells of any dimension may be compared. Algorithm `COMPARECELLS` begins by first ensuring that the two cells are distinct. If not, as a cell cannot be less than itself, false is returned. In case α is the higher dimensional cell, then Algorithm `COMPARECELLS` is invoked again with the arguments reversed and the result is then negated. Algorithm `COMPARECELLS` invokes Algorithm `COMPAREDCELLS` to compare the

¹This is only true for cell complexes which contain maximal cells of one type such as triangle/tetrahedral meshes and structured grids. For more general cell complexes, this need not be true. For such cases, line number 11 of Algorithm `COMPAREDCELLS` needs to be replaced with `COMPARECELLS`(ψ, ψ').

d -cell α with the highest d -face of β α' . However, if α and α' are the same cell, because $\alpha \neq \beta$, it must be that α' is a proper face of β . In that case, since a cell is higher than all its face by the ordering of F_w , the procedure returns true.

Algorithm 4.2 COMPARECELLS

```

1: procedure COMPARECELLS( $\alpha, \beta$ )
2:   if  $\alpha = \beta$  then                                     ▷ If both cells are the same,
3:     return False                                         ▷ a cell is not less than itself.
4:
5:    $d \leftarrow \text{DIM}(\alpha)$ 
6:   if  $d > \text{DIM}(\beta)$  then                                 ▷ If  $\alpha$  has higher DIM, re-invoke with
7:     return  $\sim$ COMPARECELLS( $\beta, \alpha$ )                       ▷ args reversed and negate.
8:
9:    $\alpha' \leftarrow \beta$                                      ▷ Get  $\beta$ 's highest  $d$ -face  $\alpha'$ .
10:  while  $d < \text{DIM}(\alpha')$  do
11:     $\alpha' \leftarrow G_0(\alpha')$ 
12:
13:  if  $\alpha = \alpha'$  then
14:    return True                                           ▷  $\alpha$  is a proper face of  $\beta$ 
15:
16:  return COMPAREDCCELLS( $d, \alpha, \alpha'$ )

```

Additional terminology : G_0^i , ϵLST , and ϵLLK

Here, we introduce the terms G_0^i , ϵ -lower star (ϵLST), and ϵ -lower link (ϵLLK) to ease correctness and analysis arguments in ensuing sections.

We introduce the notation G_0^i to help unravel the recursive definition of $F_w(\alpha^d)$. Let $G_0^i(\alpha^d)$ denote the G_0 function applied i (≥ 0) times on a cell α^d . For example, $G_0^2(\alpha^d) = G_0(G_0(\alpha^d))$, $G_0^0(\alpha^d) = \alpha^d$.

We define ϵ -lower star of the vertex v as the set of cells σ^d such that $v = G_0^d(\sigma)$:

$$\epsilon LST(v) = \{\sigma^d \in K \mid v = G_0^d(\sigma^d)\}$$

We note that if K is a simplicial complex, v is a vertex and the function is a PL extension of samples at the vertices, then the ϵ -lower star of v is exactly the lower

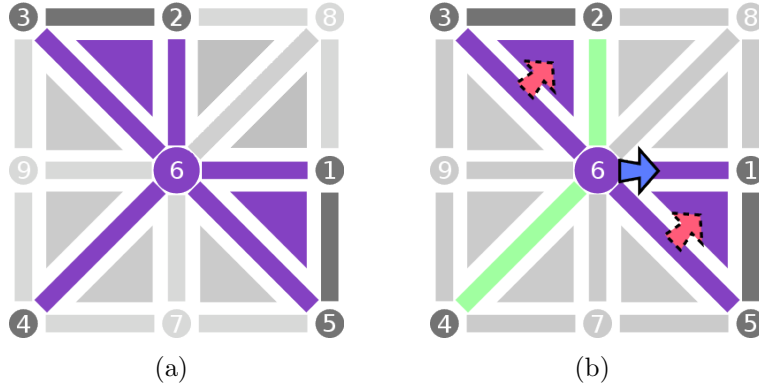


Figure 4.2: (a) ϵLST (cells in purple) and ϵLLK (cells in dark gray) of a vertex, for a function sampled at the vertices. Other cells are shown in light gray. (b) Gradient pairs and critical cells determined by algorithms [ASSIGNGRADIENT](#) and [ASSIGNGRADIENT2](#).

star of v (See Figure 4.2(a)). Similarly, we define ϵ -lower link (ϵLLK) of a vertex v to be the set of faces of cells in $\epsilon LST(v)$ that are not incident on v .

We claim that the cells in $\epsilon LST(v)$ are ordered contiguously by F_w . Consider any $\gamma^d \notin \epsilon LST(v)$. We can express $F_w(\gamma^d)$ as

$$F_w(\gamma^d) = F_w(G_0^d(\gamma^d)) + \sum_{i=1}^d \epsilon^i \times F_w(G_1(G_0^d(\gamma^d)))$$

by successively rewriting the leading term. Since $\gamma^d \notin \epsilon LST(v)$ we have that $G_0^d(\gamma^d) \neq v$. By writing the expression for F_w for all cells $\sigma^{d'} \in \epsilon LST(v)$ in the above form, the comparison of γ^d and $\sigma^{d'}$ will be dominated by the comparison of cells $G_0^d(\gamma^d)$ and $G_0^{d'}(\sigma^{d'})$. Hence γ^d would precede or succeed all cells of $\epsilon LST(v)$.

4.1.3 Computing gradient pairs

We now outline algorithm [ASSIGNGRADIENT](#) and its extension [ASSIGNGRADIENT2](#) to compute discrete gradient pairs using the Comparator [COMPARECELLS](#). We prove that the pairs found are unique and independent of the order in which

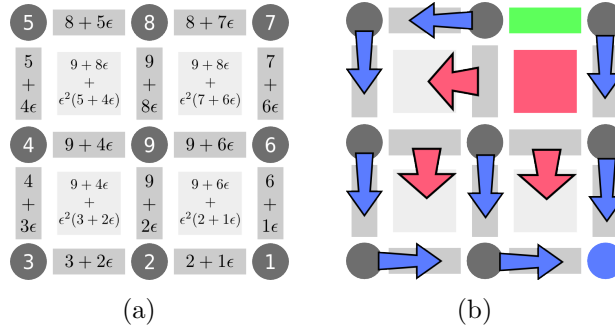


Figure 4.3: (a) A scalar function, f , defined on the vertices is recursively extended to a discrete Morse function F_w . The value of F_w is shown for each cell. (b) Gradient pairs determined by [ASSIGNGRADIENT](#).

the cells are considered, thus providing scope for parallelizing the algorithms. We then prove the validity of the gradient field. In doing so, we develop the [COMPARECELLSPOSTPAIR](#) comparator that generates a total order which respects the partial order induced by the gradient field.

Gradient Algorithm

Algorithm 4.3 [ASSIGNGRADIENT](#)

- 1: **procedure** [ASSIGNGRADIENT](#)()
 - 2: **for all** $\alpha \in K$ **do**
 - 3: $P_\alpha = \{\beta \mid \alpha = G_0(\beta)\}$
 - 4: **if** $P_\alpha \neq \phi$ **then**
 - 5: $\beta = \text{Min}(P_\alpha)$ \triangleright Lowest in P_α ordered by [COMPARECELLS](#).
 - 6: pair_cells (α, β)
-

Algorithm [ASSIGNGRADIENT](#) list the gradient pairing method, where α denotes a cell in the complex K . The set P_α is the set of α 's cofacets whose maximal facet is α itself. Figure 4.3(b) shows the gradient field determined by the algorithm [ASSIGNGRADIENT](#) for the function in Figure 4.3(a). We now prove the uniqueness of pairs found by Algorithm [ASSIGNGRADIENT](#).

Lemma 4.1: ORDER INDEPENDENT PAIRING LEMMA. *The pairing determined by the algorithm `ASSIGNGRADIENT` is independent of the order in which cells are processed. In particular, if a cell α pairs with its cofacet β then β will not pair with any of its cofacets.*

Proof. Inconsistencies occur if the algorithm determines two or more pairs for the same cell. A cell present in two pairings can be of the nature $(\alpha, \beta), (\alpha, \beta')$ or $(\alpha', \beta), (\alpha, \beta)$ or $(\gamma, \alpha), (\alpha, \beta)$ where $\gamma \triangleleft \alpha \triangleleft \beta$.

This first conflict is trivially not possible because for a cell α we determine a unique pair from a set of candidate facets. In the second case, if β were to be paired with two different facets, α and α' , then $\beta \in P_\alpha, P_{\alpha'}$. But, from the definition of P_α we know that $G_0(\beta)$ is unique and equal to either α or α' . Therefore, β must either belong to P_α or to $P_{\alpha'}$ but not both. So, β is paired either with α or with α' .

To prove that the third conflict does not arise, we show that if α pairs with one of its cofacets β , then α is not the lowest pairable cofacet of any of its facets *i.e.* $\beta = \text{Min}_F(P_\alpha)$ implies $\alpha \neq \text{Min}_F(P_\gamma)$ for all $\gamma \triangleleft \alpha$. This will imply that if α paired with β , then it is not paired with any other cell γ . Consider a facet γ of α , $\gamma \triangleleft \alpha$. If $\alpha \notin P_\gamma$ then there is nothing to prove because the algorithm will not pair γ with α . Now assume $\alpha \in P_\gamma$. For a regular cell complex, if γ is a face of a cell β such that $\dim(\gamma) = \dim(\beta) - 2$, then there exists exactly two cells σ_1, σ_2 such that $\gamma \triangleleft \sigma_1 \triangleleft \beta$ and $\gamma \triangleleft \sigma_2 \triangleleft \beta$. Without loss of generality, we relabel σ_1, σ_2 as α, α' . Since (α, β) form a pair and not (α', β) , we have $F_w(\alpha') < F_w(\alpha)$. Hence, it is sufficient to show that $\alpha' \in P_\gamma$.

Assume that $\alpha' \notin P_\gamma$. There exists $\gamma' \neq \gamma \in K$ such that $\gamma' \triangleleft \alpha'$ and $\alpha' \in P_{\gamma'}$. This implies $F_w(\gamma') > F_w(\gamma)$. Since $F_w(\alpha) = F_w(\gamma) + \epsilon$ and $F_w(\alpha') = F_w(\gamma') + \epsilon$ we have $F_w(\alpha) < F_w(\alpha')$. This is a contradiction. Hence, we have $\alpha' \in P_\gamma$ and $F_w(\alpha') < F_w(\alpha)$. So, if (α, β) is a pair then $\alpha \neq \text{Min}_F(P_\gamma)$ for any $\gamma < \alpha$, which

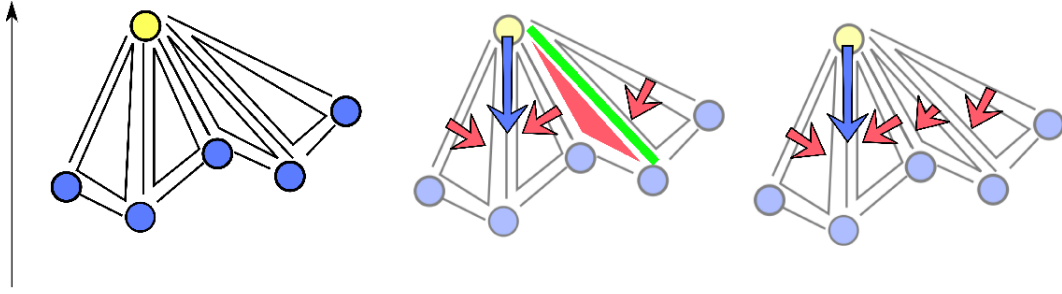


Figure 4.4: (left) The ϵ -lower-star and the lower link of a vertex with the height function defined. (center) Algorithm `ASSIGNGRADIENT` declares the green edge and the red triangle as critical because the edge is not the highest facet of any of its cofacets. (right) Algorithm `ASSIGNGRADIENT2` pairs these cells because the edge is the second highest facet of the unpaired triangle.

implies that there is no such pair (γ, α) . \square

Extended Gradient Algorithm

Algorithm `ASSIGNGRADIENT` is conservative and may leave multiple cells unpaired thereby declaring them critical (See Figure 4.4). We introduce an additional pairing procedure that avoids the creation of such ϵ -persistent pairs within the ϵ -lower star. This procedure executes during a second pass over the cells and essentially seeks to pair cells with their second highest facet consistently when their highest facet is paired with another cell.

Algorithm 4.4 `ASSIGNGRADIENT2`

```

1: procedure ASSIGNGRADIENT2()
2:   for all  $\alpha \in K \setminus K^{(0)}$  do
3:     if  $\alpha$  not paired then
4:        $P_\alpha = \{\beta \mid \alpha \text{ is second highest facet of } \beta\}$ 
5:       if  $P_\alpha \neq \phi$  then
6:          $\beta = \text{Min}(P_\alpha)$   $\triangleright$  Lowest in  $P_\alpha$  ordered by COMPARECELLS.
7:         if  $\beta$  not paired then
8:           pair_cells  $(\alpha, \beta)$ 

```

We now state and prove a lemma to show that the pairs determined by Algorithm [ASSIGNGRADIENT2](#) are unique and hence the algorithm can be parallelized. The following Lemma states that if a cell is paired by Algorithm [ASSIGNGRADIENT2](#), then the pairing is unique. In other words, the cell is either paired with one of its cofacets or with its second highest facet, independent of the order in which cells are paired.

Lemma 4.2: EXTENDED ORDER INDEPENDENT PAIRING LEMMA. *If Algorithm [ASSIGNGRADIENT2](#) pairs a cell β with its second-highest facet α then it will not pair α with its second-highest facet γ .*

Proof. Consider the incidence relationships shown in Figure [4.5\(a\)](#) between a d -cell β , its highest facet α' , second highest facet α , highest $(d - 2)$ face γ' , highest $(d - 3)$ face ψ' , and α 's second highest facet γ . Since the input cell complex K is regular, there exists exactly two facets of β , say α_1 and α_2 , incident on γ' . Further γ' is the highest facet of α_1 and α_2 , which in turn implies that any third facet of β does not contain γ' and is hence lower than both α_1 and α_2 . So, α_1 and α_2 are the highest and second highest facets of β , namely α' and α . A similar argument on α and its highest $(d - 3)$ face ψ' shows that γ and γ' are incident on ψ' and ψ' is their highest facet.

We will prove the existence of a facet, α'' , of β that contains γ as its highest or second highest facet. In either case, γ will be paired with a cell different from α . If γ is the highest facet of α'' then Algorithm [ASSIGNGRADIENT](#) would have paired it with a cell different from α because α remains unpaired until it is processed by Algorithm [ASSIGNGRADIENT2](#). If γ is the second highest facet of α'' , then Algorithm [ASSIGNGRADIENT2](#) will seek to pair it with the lowest cofacet in P_γ . The cell α'' is lower than α and belongs to P_γ . So α will not be paired with γ .

We now show the existence of the cell α'' . Consider the $(d - 2)$ cell γ as a

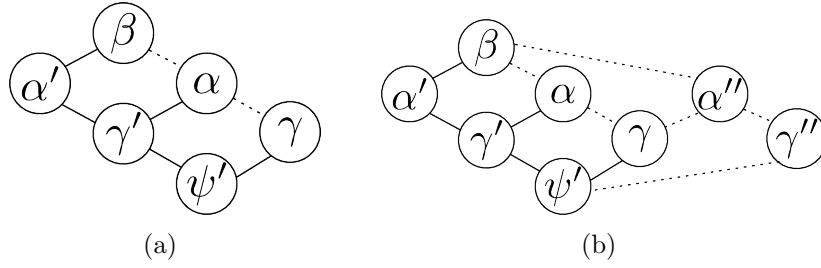


Figure 4.5: (a) α is the second highest facet of β and γ is the second highest facet of α . Respective maximal facets are shown (α', γ'). Solid lines represent maximal facet relation. Dotted lines represent incidence relation. (b) The regularity of K implies the existence of faces α'' and γ'' .

face of β . Since the input is a regular cell complex, there exists exactly two facets of β incident on γ . The cell α is one such facet of β . Let α'' be the other. The regularity of the input cell complex also implies the existence of exactly two facets, γ and γ'' , that are incident on ψ' . Further, ψ' is the highest $(d - 3)$ face of α'' . It follows that γ is either the highest facet or second highest of α'' using the same argument as above to show that α and α' are incident on γ' . \square

Validity of the gradient field

We now prove that Algorithms [ASSIGNGRADIENT](#) and [ASSIGNGRADIENT2](#) produce a valid discrete gradient field. To do this, we first construct a second total order, induced by the gradient pairs, by modifying the order of Algorithm [COMPARECELLS](#). Then, we prove a lemma to establish the validity of the gradient field.

The order induced by Algorithm [COMPARECELLS](#) is modified for each gradient pair (α, β) , where α is a facet of β . In particular, α is promoted to be infinitesimally higher than its cofacet and pair β . Thus, α immediately succeeds β in the new total order. Since gradient pairs are unique, the resulting total order is well defined. Algorithm [COMPARECELLSPOSTPAIR](#) lists the comparator procedure that induces

Algorithm 4.5 COMPARECELLSPOSTPAIR

```

1: procedure COMPARECELLSPOSTPAIR( $\alpha, \beta$ )
2:   if  $\alpha = \beta$  then
3:     return False
4:
5:   if DIM( $\alpha$ ) > DIM( $\beta$ ) then
6:     return  $\sim$ COMPARECELLSPOSTPAIR( $\beta, \alpha$ )
7:
8:   if ISPAIR( $\alpha, \beta$ ) then
9:     return False            $\triangleright$  Consider lower dim cell of a pair as Higher
10:
11:  if ISPAIRED( $\alpha$ ) and DIM( $\alpha$ ) < DIM(GETPAIR( $\alpha$ )) then
12:     $\alpha \leftarrow$  GETPAIR( $\alpha$ )    $\triangleright$  If  $\alpha$  is a facet of its pair, replace with its pair.
13:
14:  if ISPAIRED( $\beta$ ) and DIM( $\beta$ ) < DIM(GETPAIR( $\beta$ )) then
15:     $\beta \leftarrow$  GETPAIR( $\beta$ )    $\triangleright$  If  $\beta$  is a facet of its pair, replace with its pair.
16:
17:  return COMPARECELLS( $\alpha, \beta$ )

```

this total order. Lines 2 to 7 of Algorithm COMPARECELLSPOSTPAIR ensure that the input cells α and β are distinct and α has dimension lesser than or equal to β (similar to analogous lines of Algorithm COMPARECELLS). If α is the lower dimensional cell of a pair, it is considered to be infinitesimally higher than its pair. Therefore, in this case, lines 11 to 13 of Algorithm COMPARECELLSPOSTPAIR reorders α to be immediately higher than its pair by replacing α with its pair for comparison. A similar reordering of β is performed if necessary. Finally, the possibly reordered/replaced cells are compared using Algorithm COMPARECELLS. A corner case arises if both α and β themselves form a gradient pair which is dealt with in line 9.

Next, we prove the following lemma concerning the order of COMPARECELLS and COMPARECELLSPOSTPAIR.

Lemma 4.3: PAIRING VALIDITY LEMMA. *Given a gradient pair (α, β) determined by Algorithms `ASSIGNGRADIENT` or `ASSIGNGRADIENT2`, where α is a facet of β , every other cofacet $\tilde{\beta}$ of α is higher than β in the total order of Algorithms `COMPARECELLS` and `COMPARECELLSPOSTPAIR`.*

Proof. We begin by showing that the proposition is true for Algorithm `COMPARECELLS`. If (α, β) is a pair found by Algorithm `ASSIGNGRADIENT`, then β is the lowest cofacet of α . Therefore, $\tilde{\beta}$ is higher than β . If (α, β) is a pair found by Algorithm `ASSIGNGRADIENT2`, then α is not the highest facet of $\tilde{\beta}$ since it remained unpaired after being processed by Algorithm `ASSIGNGRADIENT`. If α is the second highest facet of $\tilde{\beta}$, then $\tilde{\beta}$ is higher than β because it was not selected by Algorithm `ASSIGNGRADIENT2`. If α is neither the highest nor the second highest facet of $\tilde{\beta}$, then the highest facet of α , say γ' , is not the highest $(d - 2)$ face of $\tilde{\beta}$. This follows from the converse of the argument used in the proof of Lemma 4.1.3 to show that α' and α , the highest and second highest facets of β , are incident on γ' . Cells in the ϵ -lower-star of γ' appear contiguously in the ordering induced by Algorithm `COMPARECELLS` (see Section 4.1.2). Since $\tilde{\beta}$ does not lie in the ϵ -lower-star of γ' and β does, it follows that $\tilde{\beta}$ is not lower than β .

We now show that the proposition is true for Algorithm `COMPARECELLSPOSTPAIR`. We observe that Algorithm `COMPARECELLSPOSTPAIR` only reorders α and not β . Furthermore, α is reordered to be immediately above β . Since every other cofacet $\tilde{\beta}$ of α is already higher than β in the order of Algorithm `COMPARECELLS`, α is not reordered to be higher than $\tilde{\beta}$. On the other hand, if $\tilde{\beta}$ is reordered, then it may only move higher and thus α still remains lower than it. \square

We now prove the validity of the gradient field determined by the order of Algorithm `COMPARECELLSPOSTPAIR`. A direct consequence of the above lemma is that the number of cofacets that are lower than any cell is at most one. Conversely, we claim that the number of facets higher than a cell is also at most one. If this

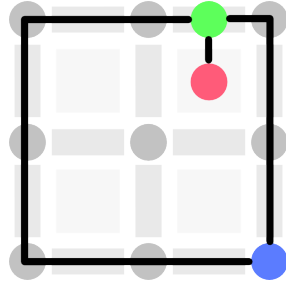


Figure 4.6: The combinatorial MS complex computed using a BFS traversal on the gradient field shown in Figure 4.3(b).

were not true, then there exists a cell β whose two facets α and α' are higher than it in the order of `COMPARECELLSPOSTPAIR`. This would be only possible when α and α' were higher than their respective paired cofacets, say $\tilde{\beta}$ and $\tilde{\beta}'$. By the uniqueness of the pairing algorithms, we must have that $\tilde{\beta}$ and $\tilde{\beta}'$ are distinct cells, one of which is distinct from β . Without loss of generality, assume this cell to be $\tilde{\beta}$ that is paired with α . However, this implies that two of α 's cofacets, $\tilde{\beta}$ and β are higher than it. This violates the above lemma, and thus establishes the validity of the gradient field.

4.1.4 Computing the MS complex

Once the discrete gradient field is computed, the descending / ascending manifolds and the combinatorial MS complex are extracted as a collection of gradient paths. From the definition, the critical points are cells in K that remain unpaired after all gradient pairs have been computed. The descending manifold of a critical point is equal to the closure of all gradient paths that originate from that critical point. This is computed using a breadth first traversal of gradient pairs beginning from the critical point. The ascending manifold is the closure of the set of gradient paths that terminate at a given critical point. This is computed using a breadth first traversal of reversed gradient pairs beginning from the critical point. A combinatorial connection between any two critical cells is established if there is a gradient path that connects them. Figure 4.6 shows the combinatorial MS

complex extracted from the gradient field shown in Figure 4.3(b). In the following sections, we discuss the challenges and algorithms to efficiently perform BFS traversals for computing the Morse-Smale complex using multicore and massively parallel environments. Due to differences in the structure of gradient paths originating/terminating at extrema and saddles, we discuss their traversal separately.

Gradient Paths from/to Extrema

In this section, we discuss the BFS traversal of gradient paths that originate/terminate at extrema. These algorithms are directly applicable to both, 2D and 3D, gradient fields. A key aspect of gradient paths that originate at maxima is that they split but do not merge [7]. Analogously, discrete gradient paths that terminate at minima merge but do not split. Thus, the structure of the gradient field originating (terminating) at a maxima (minima) is a (reversed) tree.

For multicore environments, BFS traversals from each critical point may be computed independently. The number of parallel BFS traversals launched in parallel usually depends on the number of cores. We adopt a different strategy in the case of massively parallel environments. Due to the tree structure of the gradient paths from/to extrema, every gradient pair on a path from a maximum (minimum) is immediately preceded by a unique source (destination) which is either another gradient pair or the maximum (minimum). The traversal is now posed as an iterative search for the source/destination extremum of every gradient pair. For completeness, maxima are their own source and minima are their own destination. Each work item (thread) is mapped to iteratively determine the eventual source/destination of a gradient pair. At every iteration the source of gradient pairs that are on gradient paths originating from a unique maximum is updated to the source of its source. Similarly, the destination of gradient pairs that are on gradient paths terminating at a unique minimum is updated to the destination of

its destination. The iterations stop when all pairs find their unique source or destination. For a path of length n , the first iteration updates each node's source to the gradient pair at a distance two. The next iteration updates it to the gradient pair at a distance four. Thus, the process terminates in $\log_2(n)$ steps. Though the worst case asymptotic complexity of this traversal is $n \log_2(n)$, in practice we observe that traversal requires $\log_2(n)$ time due to the parallelization. The combinatorial MS complex is computed by querying the source/destination of gradient paths that originate/terminate at facets/cofacets of saddles. The geometry of extrema is available as a disjoint set of trees rooted at them.

Gradient Paths from/to Saddles

In this section, we discuss the BFS traversal of gradient paths that originate/terminate at saddles. In particular, we discuss the traversal of paths that originate at 1-saddles and terminate at 2-saddles in the case of 3D domains. The same methods are applicable in the traversal of paths that originate at saddles and terminate at minima (or reversed paths that terminate at maxima).

The sub-structure of the gradient field consisting of 1-saddles, 2-saddles and the (1,2) gradient pairs between them can be very intricate, where (1,2) denotes the dimension of the cells of gradient pairs. This is because 1-saddle-2-saddle gradient paths in a three-dimensional domain may both split and merge. Figure 4.7 depicts the sub-structure of a gradient field that originates from a 2-saddle, splits and merges twice, before reaching a 1-saddle.

We trace the (1,2) gradient paths by interpreting the sub-structure as a directed acyclic graph (DAG) induced by them. The number of paths between a 2-saddle and 1-saddle may be counted as the number of paths between 2-saddles and 1-saddles nodes in this DAG. We do not employ the standard breadth first search algorithm to traverse the graph because this would necessitate the use of an array

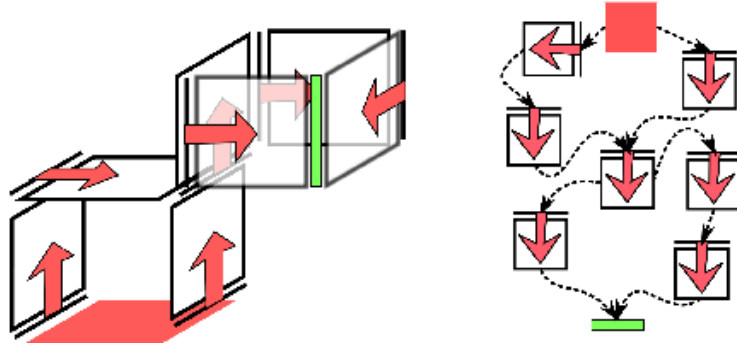


Figure 4.7: (*left*) The sub-structure of a possible gradient field between the red 2-saddle and the green 1-saddle. (*right*) The gradient field interpreted as a directed acyclic graph. The nodes are 2-saddles, the gradient pairs, and 1-saddles. Dashed curves show directed edges from 2-saddles or from the 2-cells of (1,2) gradient pairs to incident 1-cells of distinct (1,2) pairs or to 1-saddles. The gradient paths from the 2-saddle split and merge twice before they reach the 1-saddle resulting in four possible paths between them. Repetition of this configuration causes an exponential growth in the number of paths connecting the 2-saddle to the 1-saddle.

of flags to maintain if every cell is visited or not. Parallelizing the traversal will require a buffer, whose size equals that of the input, for each thread. This approach is clearly not scalable. We note that the number of critical cells reachable from a given cell tends to be small. Algorithm `CONNECTSADDLES` describes a priority-queue based traversal method to determine the paths between 2-saddles and 1-saddles. The algorithm computes the number of gradient paths from a given 2-saddle to all (1,2) gradient pairs and 1-saddles that are reachable from it.

The algorithm begins by first initializing a priority queue Q that can contain 2-cells and 1-cells of K . The priority queue is ordered using the comparator `COMPARECELLSPOSTPAIR`. The number of paths that arrive from the starting cell σ to each element α in Q is associated with α . Q is initialized with $(\sigma, 1)$. The algorithm pops the first cell α from Q . It is possible for copies of the same cell to be entered into Q . Since all these cells have the same priority, Q is repeatedly popped until all copies of α are removed and the number of paths n that reach α are summed over all copies. If α is a critical 1-cell, then an arc with n multiplicity

is inserted between σ and α . If α is a 2-cell, then all the 1-saddles and 2-cells of (1,2) pairs incident on the boundary of α (other than itself) are inserted into Q . The newly inserted pairs/saddles lie on n number of paths from σ through α . Newly inserted cells are lower than α . So, α never re-enters Q . A cell is inserted into Q when processing one of its neighboring cells. So, the number of copies of the cell in Q is upper bounded by the number of its neighbors.

Each cell enters Q only a constant number of times. So, the complexity of the algorithm is $n \log(n)$, where n is the number of 2-cell-1-cell pairs and 1-saddles. The entries in Q represent the frontier of the traversal. This set, in practice, is much smaller than the size of the input n . Descending manifolds of saddles are computed by modifying Algorithm [CONNECTSADDLES](#) to save the cells popped out of the priority queue at each iteration of the main loop. Ascending manifolds of 1-saddles are computed by employing the same procedure after reversing the priority, and reversing the role of 1-cells and 2-cells.

4.1.5 Analysis

In this section, we argue for the correctness of the MS complex computed by our algorithm. Specifically we show that the computed critical points and gradient pairs are close to those of the PL function.

Closeness of critical cells to PL critical points

We now show that every index- i PL critical vertex has a critical cell of dimension i incident upon it. Furthermore, we also show multi-saddles are also approximated with the appropriate number of critical cells. This is done by showing that the number of index k critical cells retained by any discrete gradient pairing algorithm that pairs cells within $\epsilon LST(v)$ must be greater than or equal to the k^{th} reduced Betti number of $\epsilon LLK(v)$.

Algorithm 4.6 CONNECTSADDLES

```

1: procedure CONNECTSADDLES( $\sigma$ )
2:    $Q \leftarrow$  CREATEPRIORITYQUEUE()
3:    $Q.PUSH(\sigma, 1)$ 
4:
5:   while  $Q \neq \phi$  do
6:      $\alpha, n \leftarrow Q.POP()$ 
7:     while  $Q \neq \phi$  do
8:        $\alpha', n' \leftarrow Q.POP()$ 
9:       if  $\alpha' = \alpha$  then
10:         $n \leftarrow n + n'$ 
11:       else
12:         $Q.PUSH(\alpha', n')$ 
13:        break
14:
15:     if  $\text{DIM}(\alpha) = 1$  then
16:       CONNECT( $\sigma, \alpha, n$ )
17:
18:     if  $\text{DIM}(\alpha) = 2$  then
19:       for all  $\gamma \lessdot \alpha$  do
20:         if ISCRITICAL( $\gamma$ ) then
21:            $Q.PUSH(\gamma, n)$ 
22:         else
23:           if  $\text{DIM}(\text{PAIR}(\gamma)) = 2$  and  $\text{PAIR}(\gamma) \neq \alpha$  then
24:              $Q.PUSH(\text{PAIR}(\gamma), n)$ 

```

We note that the family of cell complexes formed by attaching cells in increasing order of F_w is a valid filtration. This is because the faces of a cell always have function value lower than that of the cell (by definition of F_w) and therefore appear before the cell in the ordering. Next, we observe that algorithms [ASSIGN-GRADIENT](#) and [ASSIGNGRADIENT2](#) pair cells within the ϵLST of a vertex v . In other words, if (α^d, σ^{d+1}) is a pair, then both α^d and σ^{d+1} belong to the ϵ -lower star of some vertex v . Thus the same pairs are determined for a given ϵ -lower star attached to a given ϵ -lower link regardless of other cells in the cell complex.

Consider the hypothetical situation where a vertex v' precedes v in the filtration such that $\epsilon LST(v')$ is a duplicate of the $\epsilon LST(v)$ attached to $\epsilon LLK(v)$. We will relate the reduced Betti numbers of $\epsilon LLK(v)$ to the increase in the Betti numbers of the complex after attaching v and its ϵ -lower star (See Figure 4.8). Let $K_{v'}$ denote the cell complex obtained after attaching v' and its ϵ -lower star. Since the gradient pairs are determined within the $\epsilon LST(v)$, they are not affected by gradient pairing in the rest of the complex. Assume that the gradient field is optimal in the sense that the number of critical points of index k (n_k) is exactly the same as the k^{th} Betti number. Here, attaching $\epsilon LST(v)$ causes the k^{th} Betti number of K increases by the value of the $k - 1^{th}$ reduced Betti number of $\epsilon LLK(v)$, for $k = 0, 1, \text{ and } 2$. For example, if the -1^{th} reduced Betti number of $\epsilon LLK(v)$ is 1, then attaching $\epsilon LST(v)$ would create a new component. In other words, it increases the 0^{th} Betti number of $K_{v'}$ by one. Similarly, if the 0^{th} reduced Betti number of $\epsilon LLK(v)$ is c , then attaching $\epsilon LST(v)$ would increase the 1^{st} Betti number of $K_{v'}$ by c . Again, if the 1^{st} reduced Betti number of $\epsilon LLK(v)$ is 1, then attaching $\epsilon LST(v)$ would increase the 2^{nd} Betti number of $K_{v'}$ by 1. Thus, by attaching $\epsilon LST(v)$ to $K_{v'}$, the k^{th} Betti number increases by the $k - 1^{th}$ reduced Betti number of $\epsilon LLK(v)$. Since the gradient field was optimal before $\epsilon LST(v)$ was attached, the number of index k critical cells should correspondingly increase

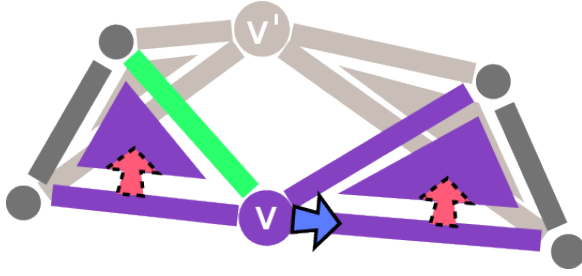


Figure 4.8: A cell complex where a duplicate of $\epsilon LST(v)$ precedes v in the filtration and is attached to $\epsilon LLK(v)$.

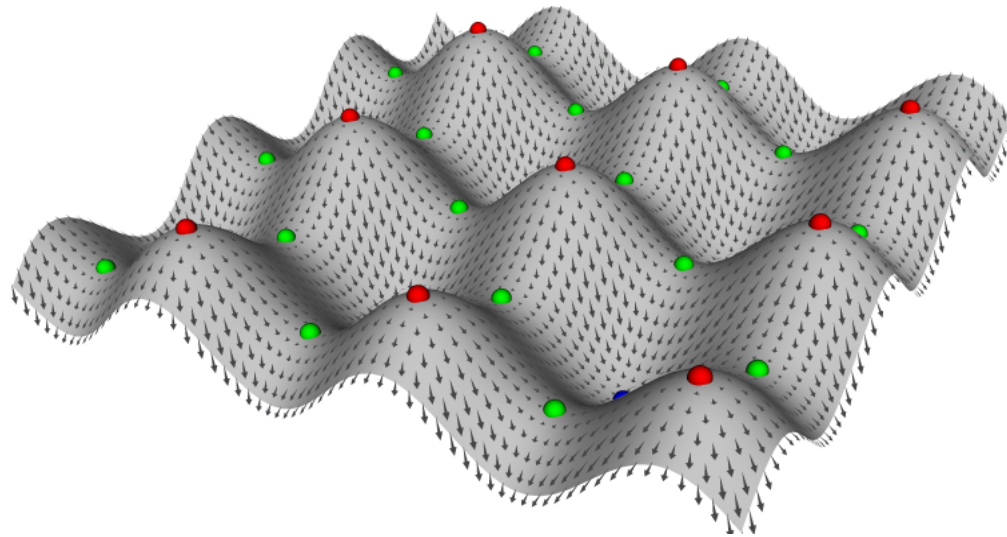
by at least the $k - 1^{th}$ reduced Betti number of $\epsilon LLK(v)$ to satisfy the weak Morse inequality. Since the only new cells were that of $\epsilon LST(v)$, the new critical points must be present within the $\epsilon LST(v)$.

This result shows that PL critical points are approximated by a critical cell incident on the PL critical vertex. Furthermore multi-saddles are also approximated with the appropriate number of critical cells.

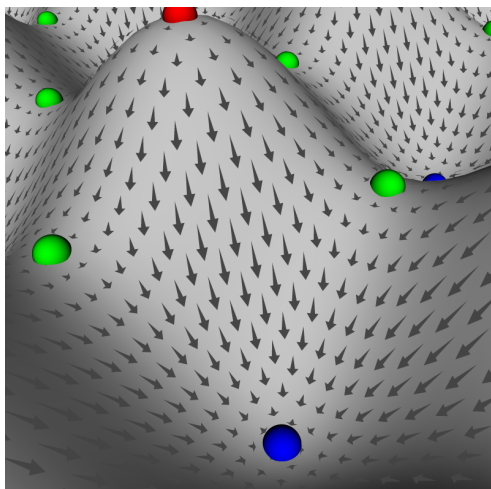
Steepest descent

Consider a PL function defined on a simplicial complex whose function value at vertices is known. The gradient pairing algorithm will attempt to pair a cell α^d with a cell σ^{d+1} , where σ^{d+1} is a simplex formed by adding a vertex to V_α and the new vertex has function value lesser than all vertices in V_α . For every point on α^d , the gradient of the PL interpolant is oriented towards the new vertex. Hence the gradient lines originating from the interior of α^d , are oriented towards the interior of σ^{d+1} . Because of the discontinuity of gradients of PL interpolants on cells that are shared, the gradient algorithm will pair the $d + 1$ -cell attached to α^d with minimum function value. This will be the $d + 1$ -cell attached to α^d with minimum function value on the vertex not present in V_α , therefore maximizing the magnitude of the gradient. Hence the gradient vector pairing agrees with the maximal PL gradient on a simplicial complex.

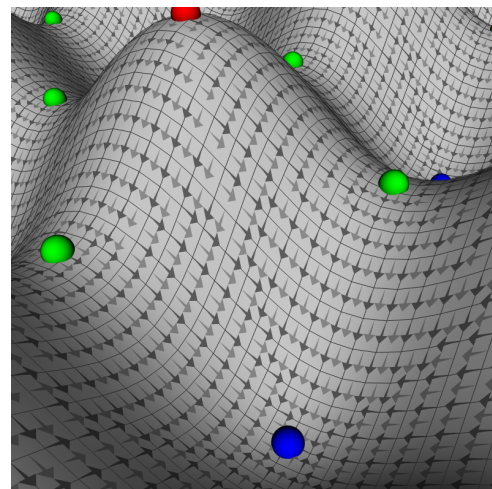
In the case of two dimensional rectilinear grids using a bilinear interpolant it is seen that the same argument applies except for the case when the quad contains



(a)



(b)



(c)

Figure 4.9: (a) Gradient field of the function $\sin(x) + \sin(y)$ evaluated at mesh vertices. (b) Close up view of the gradient field. (c) Discrete gradient vectors for function sampled at vertices.

a face saddle. In this case we see that gradient at the mid point of the maximal edge has steepest descent gradient towards the quad element.

Figure 4.9 shows the comparison of the continuous gradient of the analytic function $\sin(x) + \sin(y)$ evaluated at the vertices of the two dimensional rectilinear grid, with the discrete gradient computed on the grid using the gradient algorithm. The discrete gradient pair arrow are aligned along edges for vertex-edge pairs and orthogonal to edges for edge-quad pairs. In both cases, they agree with the gradients computed for the analytic function at mesh vertices.

4.2 Out-of-core Algorithm

We now discuss the computation of the MS complex with a focus on large datasets that do not fit entirely in memory. The computation is done in five stages (see Figure 4.10). The data is first hierarchically partitioned into sub-domains blocks. The partitioning stops when the sub-domains are small enough to fit in memory.

4.2.1 Gradient and MS Complex on sub-domains

The computation of the gradient proceeds as outlined in the previous section. To obtain a equivalent gradient field on a sub-domain, the gradient algorithm needs only a cell's cofacets and their facets in the domain. The cell complex of the sub-domain is extended to include the set of cells that are incident on the shared boundary of sub-domains and gradient is computed only on the initial sub-domain cell complex (see Figure 4.10(a)). Thus, we obtain identical pairings for cells along the shared boundary when we process all sub-domains that share the boundary cell.

To facilitate merging we mark all gradient pairs that cross a shared boundary as critical (see Figure 4.10(a)). We establish the validity of this step in the following

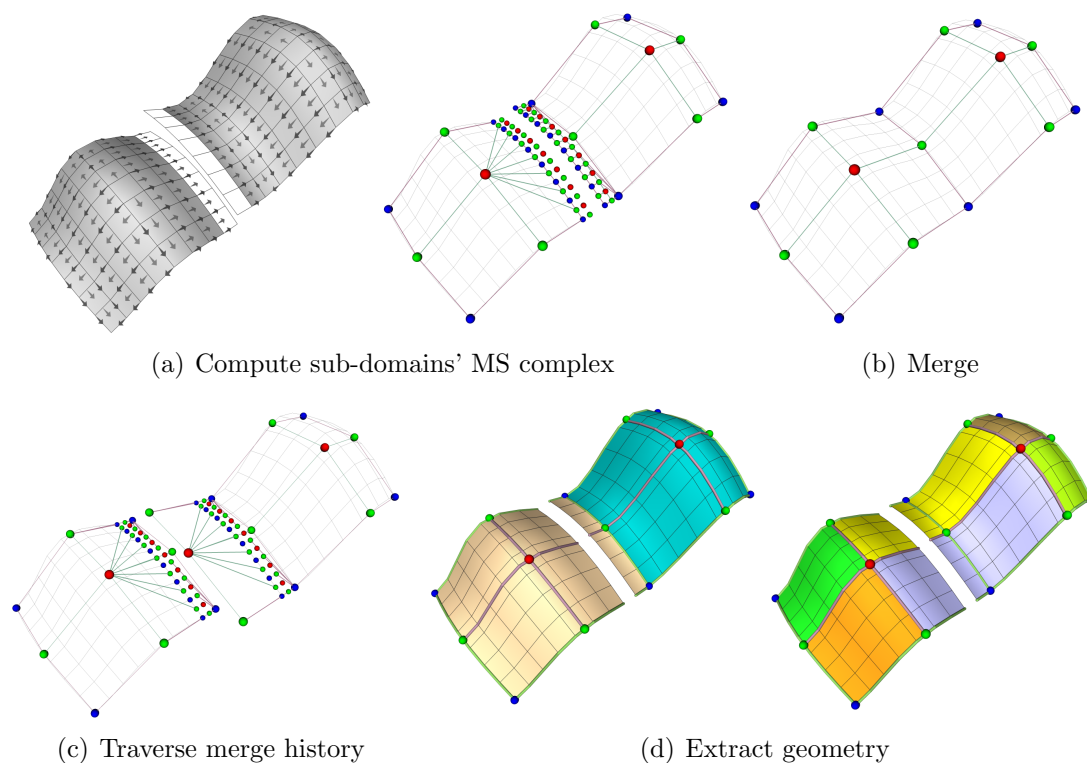


Figure 4.10: MS complex for large domains is computed in five stages. Data is first split into sub-domains. (a) Gradient is computed on sub-domains. Unpaired cells and gradient pairs incident on shared boundary are marked critical. Combinatorial MS complex on each sub-domain is computed. (b) The combinatorial MS complex of the domain is computed by identifying and canceling gradient pairs incident on the shared boundary. (c) The history of merge cancellations is traversed to reveal the incidence of critical cells across sub-domains. This information is used to trace the geometry of the cells of the MS complex. (d) For each sub-domain, the geometry of the descending and ascending manifold of an incident critical cell restricted to the sub-domain is extracted.

section.

4.2.2 Merging sub-domain MS complexes

Next, we merge the sub-domains in a bottom up fashion by identifying boundary critical point pairs and canceling them when they enter the interior of the union. The cancellation repeatedly merges the MS complex across the sub-domains till we obtain the MS complex of the input function.

We first establish the equivalence of gradient paths and the paths computed by a sequence of cancellations. A consequence of this result is that we can process the sub-domains in parallel and later merge them to obtain the MS complex while ensuring combinatorial and geometric equivalence.

Lemma 4.4: ORDER INDEPENDENT CANCELLATION LEMMA. *Let $p, \alpha_0, \sigma_0, \dots, \alpha_i, \sigma_i, \dots, \alpha_k, \sigma_k, q$ denote a gradient path between two critical points p and q . This gradient path is faithfully traced independent of the scheduled order of boundary critical point pair cancellations.*

Proof. In the above gradient path, canceling pair α_i, σ_i results in establishing the connectivity between $\sigma_{i-1}, \alpha_{i+1}$. Iterating forward, we see that cancellation of any pair along the gradient path successively establishes connectivity between the preceding and succeeding surviving critical point. Eventually the critical points p, q are connected by an arc. Thus combinatorially, this is equivalent to the MS complex obtained without by tracing a path directly from p or q without any intermediate step of creating boundary critical points. The same argument extends to prove the resulting geometric equivalence. \square

As a consequence of the above lemma, we can schedule cancellations of boundary critical point pairs in any order. Gyulassy *et al.* [12] also employ a divide and conquer approach to compute the MS complex. However, they partition the

domain into “parcels” that do not share common boundary. The merge step, therefore, has to process new cells and may introduce new critical points. Hence, they are not able to ensure the geometric equivalence of the MS complex. Our partitioning scheme is the central reason for the Order Independent Cancellation Lemma to be true.

4.2.3 History Tree

One of the implications of declaring all boundary cells and their outgoing / incoming pairs as critical is the creation of a large number of critical cells. Since the merge operation involves cancellation of critical points, the ascending and descending manifolds need to be computed and merged. However the number of cells that are present in the ascending / descending manifold of a critical point is $O(n)$, where n is the number of cells in the cell complex. This leads to a large memory foot print of intermediate complexes.

The artificial critical points represent regions through which flow enters / leaves a sub-domain. Therefore, recording the combinatorial connectivity to a surviving critical point at the boundary is sufficient to compute the ascending/descending manifold restricted to the sub-domain. We record this information during the merge step and are therefore able to compute the 1-skeleton of the MS complex with a small memory footprint. The recorded combinatorial connectivity between boundary critical points is used later to extract the geometry of the gradient paths. We now describe how we traverse the history of cancellations to compute the geometry of the arcs.

Consider a series of k cancellations to determine the combinatorial connection between two critical points p^i and q^{i-1} . The series of canceled critical point pairs

is equal to the gradient path connecting the two critical points:

$$p, \dots, \alpha_{k-1}^{i-1}, \sigma_{k-1}^i, \dots, \alpha_k^{i-1}, \sigma_k^i, \dots, \alpha_{k-2}^{i-1}, \sigma_{k-2}^i, \dots, q$$

Consider the final cancellation that determines the connection between p and q . Before cancellation, p is contained in ascending connections of α_k and q is contained in the descending connections of σ_k . Before the cancellation of the $(k-1)^{th}$ pair, α_k^{i-1} is connected to σ_{k-1}^i . By retaining this information, after the k^{th} cancellation we can infer that σ_{k-1}^i is connected to all surviving critical points in the descending connections of α_k 's pair. Extending this further to previous cancellations, we see that if we traverse the critical point pairs in reverse order of their cancellations, we can infer the entire geometry of the gradient path. This is accomplished by traversing the history tree, which records all merges, in a top-down manner. At the leaf of the history tree, we obtain the combinatorial connections from the BFS traversal within the sub-domain.

4.2.4 Geometry extraction

The history tree traversal returns the points of entry and exit of all critical cells that have gradient entering or leaving the sub-domain. Thus the geometry of the descending/ascending manifold of a critical cell restricted to the sub-domain can be computed by tracking the gradient from the cells of entry/exit that are on shared boundaries. If the critical cell is contained in the sub-domain then the geometry is computed as indicated in the first stage.

4.3 Discussion

In this section, we highlight key differences and advantages of our algorithm with other discrete Morse theory based algorithms to compute the Morse-Smale complex.

In comparison with the algorithms of Cazals *et al.* [7], Lewiner *et al.* [42], and Gyulassy *et al.* [12], our formulation using the weighted discrete function provides for a consistent total order that leads to a robust and unambiguous gradient field. Also, the gradient pairing algorithms `ASSIGNGRADIENT` and `ASSIGNGRADIENT2` allow for a simple massively parallel implementation on GPUs. Furthermore, our traversal methods allow for parallel deployment on GPUs for extrema and simultaneously on multi-core CPUs. In comparison with the method by Peterka *et al.* and Gyulassy *et al.* [43, 44], being a supercomputer/compute-cluster deployment of the algorithm of Gyulassy *et al.* [12], our algorithm is targeted towards commodity desktop hardware. A key similarity of our algorithm with the aforementioned methods is the adherence to the notion of steepest descent.

With respect to the algorithm by Robins *et al.* [13], a key difference is in the gradient pairing algorithm. Robins *et al.* focus on the topological analogy with piecewise linear critical points to track level set topology changes. Our algorithms focus on the notion of steepest descent, choosing to over-approximate with additional, possibly spurious critical points, in cases when the steepest gradient is not apparent. Furthermore, the exponential traversal step of their algorithm results in a significant loss in performance. Günther *et al.* [14] amend this aspect of Robins' algorithm by describing a memory efficient traversal algorithm having an overall quadratic complexity. However, their algorithm needs to maintain the set of visited nodes and ensure that nodes are processed in the correct order. This set can potentially grow the entire ascending / descending manifold of the critical point. Also, a second set representing the frontier of the traversal needs to be maintained.

These sets may be either maintained as data-sized flag arrays or hash-table based containers. Hash tables typically incur an asymptotic complexity of $O(\log n)$ for each access. As the algorithm is claimed to be linear for each traversal, *i.e.*, constant for each access to the container, we presume that these sets are maintained as data sized flag arrays. Even if the algorithm were amended to use hash tables, a hash table representing the descending manifold of each critical point needs to be maintained for each traversal. Thus, memory becomes potentially limiting when many traversals are deployed in parallel. In contrast, we leverage the fact that discrete gradient field itself imposes a partial order on the cells. We extend this to a total order that allows us to implicitly process nodes in the correct order (see Algorithm [COMPARECELLSPOSTPAIR](#)). We only need to maintain the frontier during the traversal, which allows us to run many traversals in parallel with lesser memory. Furthermore, when determining the 2-saddle-1-saddle combinatorial connectivity, we need to only trace descending paths from 2-saddles that intersect with ascending paths from 1-saddles. The frontier set for such paths often tends to be close to constant, resulting in further memory optimization.

We now discuss a few limitations of the algorithms. The above discussed algorithms have been designed and implemented to use GPU systems. In particular, the 3D implementations can leverage the efficient hardware capabilities of current generation GPU hardware, which provide efficient caching mechanisms for 3D structured grid data. Such mechanisms are currently unavailable for higher dimensional structured grid and unstructured grid data. Another limitation is that the algorithms have been designed to respect the notion of steepest descent. In some applications, it is desirable to design other kind of gradient fields, for example, one that exactly respects the level set topology changes as is done by Robins *et al.* [13].

4.4 2D Implementation and Evaluation

In this section, we discuss the implementation aspects of the efficient Morse-Smale complex algorithm for two dimensional structured grids. Then, we discuss experiments to study the performance of our algorithm using simulated datasets and a real world application to track the eye of a hurricane in a weather simulation dataset.

4.4.1 Implementation

We implement the Morse-Smale complex computation algorithm to leverage both, multi-core architectures (multi-core CPUs) and massively parallel architectures (GPUs). We use the OpenCL framework for programming the GPU.

Data representation

The cells of the domain, being a structured grid, are implicitly represented using the Cartesian coordinates of their centroids as identifiers. In general, integral values of coordinates are used to represent successive vertices along each axis. We scale the coordinates by two so that the interleaving cells, namely edges and faces, also obtain integral coordinate values at their centroids. Queries for facets / co-facets are therefore implicitly computed taking into consideration the boundary conditions imposed by the grid size. The scalar function values is maintained in a two-dimensional buffer whose size corresponds to the domain size. A two-dimensional single byte buffer that is twice as large as the domain on each axis is used to maintain the discrete gradient information. The first two bits of each byte represent the direction of each cell's maximal facet. The next two bits represent the direction of the cell's pair. A single bit is used to indicate if the cell is critical or not.

Gradient Pairs and MS complex on sub-domains

The algorithm begins by computing the discrete gradient using either the GPU, if available, or the CPU. While working with the GPU for the gradient assignment, we need to mirror the two data buffers in both the CPU and GPU.

For computing the set of critical points in the CPU, a simple linear counting and collection step is used. The counting and collection of critical points from the data buffers in the GPU is posed as the parallel prefix sum problem [46, 47]. The prefix scan implementations have asymptotic complexity of $O(n \log_2(n))$ but in practice we observe that traversal requires $\log_2(n)$ time due to the parallelization.

The BFS traversals required for the MS complex algorithm is implemented in the CPU using the standard BFS algorithm. The analogous implementation for the GPU is done using the iterative source/destination algorithm detailed in Section 4.1.4. Here, two data sized buffers are used to represent each 2D cell's source maximum. Each iteration reads the source information from one buffer and updates it to the second buffer. In the next iteration the roles of the buffers are swapped. A global boolean is initialized to *false* and set to *true* if a cell updates its buffer. An analogous pair of buffers is used for each 0D cell's destination minimum.

Merging

To enable stream processing of sub-domains we recursively divide the domain along a single axis. The desired level of subdivision is adjusted to accommodate the largest possible sub-domain within memory (GPU memory in the case of the GPU implementation). The recursive subdivision leads to a hierarchical structure with 2^d sub-domains, where d is the depth of the recursion, and $2^d - 1$ intermediate nodes that represent the hierarchy. Merging of intermediate nodes in each level can be done in parallel.

Simplification

We perform a persistence based simplification of the final MS complex. The simplification affects the MS complex computed for each sub-domain. The MS complex of a sub-domain is updated by identifying surviving critical points, deactivating them, and introducing new critical points that may have become incident on the sub-domain. Since simplification by persistence does not require any geometry computation, we simplify before we traverse the history tree and push the results down the tree.

History tree

The history tree that records the merges is traversed to compute the incidence of surviving critical points on sub-domain boundary. Because of the hierarchical decomposition, the traversal can be done in parallel for all nodes within a level.

Fast Geometry Queries

Once we know the combinatorial structure of the MS complex at the boundary of a sub-domain, the computation of descending and ascending manifolds is essentially a traversal of gradient paths from these entry and exit points along with the paths that originate from or terminate at the critical point. In our implementation, we track only the surviving saddle points, because maxima partition the diverging gradient flows and minima partition the converging gradient flows. In our experiments we recompute the gradients because we found that the disk latency involved in storing the gradient and retrieving them later is costlier. This is because, recomputing the gradient requires only a single read of the function values at the vertices.

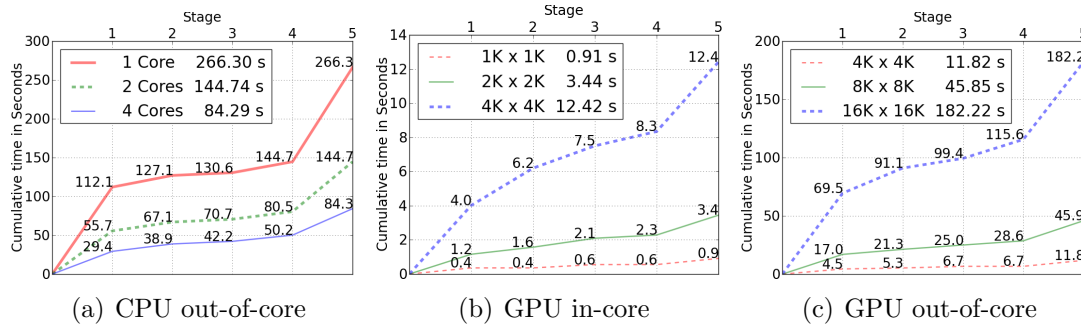


Figure 4.11: Time required to compute the MS complex for the `wgauss` dataset cumulated over the five stages of the algorithm. (a) The 8192×8192 data does not fit in CPU memory. (b) Data fits in CPU but not GPU memory. (c) Data fits neither in CPU nor GPU memory.

4.4.2 Experimental Results

We now present results of our experiments on both synthetic and the hurricane Isabel dataset [1]. All experiments were performed on a workstation with two Intel Xeon quad core processors, 8GB RAM, and nVidia GeForce 260 GTX graphics card which has 196 cores and 896MB RAM. The first synthetic data set `sine` is a sinusoidal function sampled over a rectilinear grid. The second synthetic data set `wgauss` is a 2D Gaussian distribution centered at the origin and weighted by a radially decreasing sinusoidal curve. The `wgauss` dataset contains large number of critical points and degenerate regions which help to stress test our algorithm. We study the performance and scalability of our algorithm using these two synthetic data sets.

Figure 4.11(a) shows the speed up obtained for `wgauss` sampled on an 8192×8192 grid for varying number of processors using the CPU implementation. Time is cumulated over the five stages of the algorithm. The data is processed out-of-CPU-core (not all data is present in CPU memory) to conserve memory. The graphs indicate near linear scaling with the number of cores. We observed a similar execution profile for the sinusoidal dataset with 16384×16384 data points. The

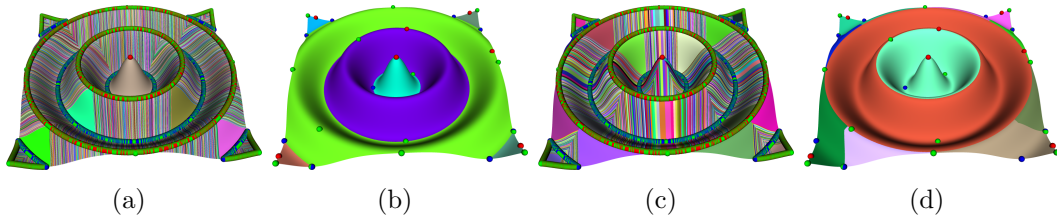


Figure 4.12: (a), (c) The full resolution descending and ascending Morse complex for the `wgauss` dataset for a grid size of 1024×1024 . (b), (d) The simplified descending and ascending Morse complex simplified upto 10%. As expected the descending manifolds partition the domain into regions that correspond to peaks and the ascending manifold partition the domain to regions that correspond to valleys.

MS complex was computed in 3 minutes and 6 seconds.

To study the scalability of the algorithm with input sizes we conducted experiments with the `wgauss` dataset computed on various grid sizes. Figure 4.11(b) shows results from the GPU execution for the `wgauss` for varying grid sizes and the corresponding speed up. Here the data is resident in the CPU memory.

Figure 4.11(c) shows results from an out-of-CPU-core execution on `wgauss` using the GPU for varying domain sizes. The size of the sub-domains is restricted to contain 1 million points. Figure 4.12 shows the full resolution and simplified descending and ascending Morse complex of the `wgauss` dataset with a grid size of 1024×1024 .

We consider a simulation of the hurricane Isabel [1] that struck the west Atlantic region in September 2003. The domain is a 3D rectilinear grid of size $500 \times 500 \times 100$ available over 48 time steps. We extract a 500×500 grid representing the land/sea surface to study the pressure (Pf), temperature (TCf) and magnitude of wind velocity fields over time. We compute the MS complex for all three scalar fields in each time step using our parallel algorithm and track significant features in the data. Figure 4.13 shows the execution profile, along with the stage wise breakup of time, for the pressure and magnitude of wind velocity fields,

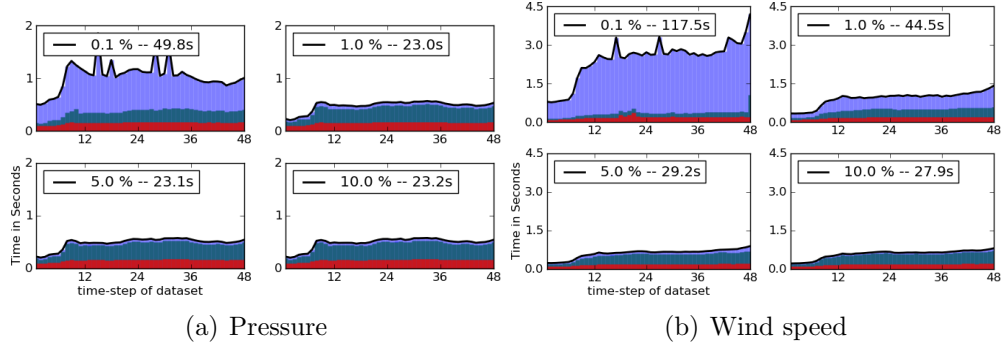


Figure 4.13: Time taken for computing the MS complex for all time-steps of the Isabel dataset for (a) Pressure and (b) Magnitude of wind velocity fields using simplification thresholds of 0.1%, 1%, 5% and 10%. Time taken for stages one, three and five are shown in the breakup along y-axis. Stages two and four are not present since the data for each field of each time-step fits in GPU memory. Time taken for geometry extraction in stage five reduces drastically if the MS complex is simplified.

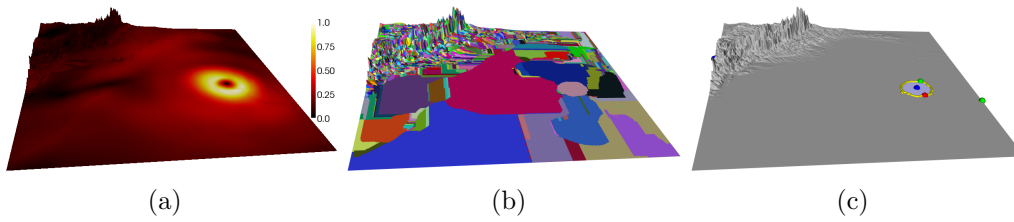


Figure 4.14: (a) The wind speed field of the 1st time step over the surface (function normalized to $[0, 1]$). (b) The full resolution ascending Morse complex. (c) The simplified MS complex retains significant critical points. The most persistent minimum corresponds to the eye of the hurricane.

for various simplification thresholds. Since the data in each field of each time-step fits in GPU memory, the merge and history tree traversal stages are not present. We observed that the time required for computation of the MS complex for most time steps was below 0.5 seconds. Without simplification, the time required to compute the MS complex increased up to 6 seconds. However, it dropped below 0.5 seconds for several time steps once we simplified critical pairs below a 0.1% persistence threshold.

Figure 4.14 shows the decomposition of the domain into ascending manifolds of the critical points of wind speed. Our implementation supports the interactive extraction of these manifolds using a parallel algorithm. We simplify the wind speed field within each time step to identify significant features after removing all the small features. Figure 4.13(a) shows the result of this experiment using the wind speed, where we track the ascending manifold of the most persistent minimum corresponding to the eye of the hurricane. Currently, we are able to process each time-step of the speed within 0.5 seconds for simplification threshold of above 5%, thereby supporting interactive analysis of the data. With additional optimizations we hope to be able to further reduce the processing time and hence enable real-time analysis and feature tracking on larger time-varying data.

4.5 3D Implementation and Evaluation

In this section, we discuss the implementation aspects of the efficient Morse-Smale complex algorithm for three dimensional structured grids. Then, we discuss experiments to study the performance of our algorithm using simulated datasets and real world datasets to compare with existing algorithms.

4.5.1 Implementation

We implemented the Morse-Smale complex algorithm to leverage both GPU computing, and multi-core CPU architectures. We used the OpenCL framework to implement Algorithms [ASSIGNGRADIENT](#) and [ASSIGNGRADIENT2](#) on the GPU. We implemented the Algorithm [CONNECTSADDLES](#) to process individual 2-saddles in parallel on the CPU. We use the Boost threading [48] library to manage multiple threads. We now discuss various implementation and optimization issues with respect to the individual stages.

Data Representation

The data structures used for representing the domain and the gradient are analogous to the data structures described in Section 4.4.1. The cells of the domain are represented using the three-dimensional analogue of the implicit representation used for the two-dimensional domain. A float sized three-dimensional data sized buffer is used to store the function value at the vertices. A byte sized three-dimensional buffer that is double the size of the domain on each axis is used to represent the gradient field information of each cell. Here, the first three bits are used to represent the direction of the maximal facet and the next three are used to represent the direction of the pair. A single bit is used to indicate if a cell is critical.

Gradient pairing and MS complex

The gradient pairs and MS-complex are computed within each sub-domain. The gradient pairs are computed in two passes using Algorithm [ASSIGNGRADIENT](#) and Algorithm [ASSIGNGRADIENT2](#). Next, the decomposition of the domain into descending/ascending manifolds of extrema are computed on the GPU. This is done

using the GPU extrema traversal method described in Section 4.1.4. Simultaneously, the CPU executes Algorithm `CONNECTSADDLES` to determine connections between 2-saddles and 1-saddles. This step is optimized to traverse only those paths that reach 1-saddles by executing a single breadth first search traversal that begins from all 1-saddles and marking all (1,2) pairs that are reached when traversing the gradient field upwards. While processing paths that descend from the 2-saddles, only pairs that are marked reachable from a 1-saddle are inserted into the priority queue Q . One bit of the 3D array, which records per cell information, is used to represent visited or not visited state of a cell.

The Morse-Smale complex is represented as a graph with nodes as critical cells. Adjacencies are represented as list of associative arrays, one for each critical point and the multiplicity of paths associated with each adjacency. Hence the complexity to access a particular adjacency of critical cell is $\log(n)$, n being the maximum number of adjacent critical cells.

Merging proceeds by cancellation of gradient pairs that cross a shared boundary. So, the descending connections of lower index critical cell of the pair and the ascending connections of the higher index critical cell are discarded by the cancellation. We further optimize by not recording such connections in the MS complex and not launching Algorithm `CONNECTSADDLES` from 2-saddles that are paired with maxima.

Merging sub-domains

The merging procedure proceeds in the reverse order of subdivision of the domain. Two sub-domains are merged into a single MS complex while ensuring that gradient pairs that cross the shared boundary are identified. Then these pairs are canceled out. This procedure ensures that there is no duplication of critical cells (and their combinatorial connections) that lie on the shared boundary. Similar to

the previous step, a persistence based simplification is performed on the combined MS complex after each merge.

Traversing merge history and extracting geometry

The traversal of merge histories computes paths from critical cells that enter a sub-domain through a shared boundary. Since we simultaneously perform simplification during merging, we consider critical cells within a sub-domain that are paired with other critical cells, possibly outside the sub domain, as entry points of gradient flow from surviving critical cells.

For extracting geometry, we require the original gradient field information. To obtain this, we reload the original function and recompute the gradient field, since the time taken to store and load this information is significantly higher than recomputing it. The partition of the domain based on gradient paths from extrema is computed on the GPU and the ascending/descending manifolds of saddles are computed using the modified version of Algorithm [CONNECTSADDLES](#).

4.5.2 Experiments

We performed experiments on two different classes of datasets. First, we evaluated our algorithm with synthetic datasets to analyze its efficiency and scaling behavior with varying parameters such as regions of near flat-gradient and large numbers or gradients crossing shared boundaries. Second, we evaluated our algorithm's performance on various volume datasets available from <http://www.volvis.org> and a dataset obtained from the simulation of a 3D Taylor-Green vortex flow on a Cartesian grid. All experiments were performed on an Intel-Xeon 2 GHz CPU with 4 cores and 16 GB of RAM and an NVidia GeForce GTX 460 GPU with 336 cores and 1GB of memory. Data was split into $256 \times 256 \times 256$ sized sub-domains that fit in memory.

Synthetic data

We use a synthetic dataset WGAUSS to stress test the algorithm. The function is defined as the product of a cosine wave with the 3D Gaussian *i.e.* $f(x, y, z) = \cos(2\pi\nu d_c) \times G_{c,\sigma}(x, y, z)$, where ν is the frequency of the cosine wave, d_c is the distance of the point from the center, $G_{c,\sigma}$ is the 3D Gaussian centered at c with variance σ , and the domain is the unit cube. The function is sampled at various grid resolutions to study scalability. This dataset is challenging since it contains multiple flat-regions at concentric spheres, where the cosine wave achieves its maximum or minimum.

We use two variants of this function to study our algorithm. First, we place the Gaussian at the center of the domain with variance 0.5 in all directions and set $\nu = 5$. The function contains concentric flat regions distributed across sub-domains resulting in several insignificant critical points. Experiments with this dataset helps study the scalability of the algorithm in the presence of noise and flat regions. Second, we distribute eight Gaussians, such that each one is centered in each octant of the domain. This variant does not possess as many flat regions because the multiple cosine waves superpose and break up the flat regions. This causes several gradients to cross common boundaries thus stressing the scalability of the merge and merge history traversal.

Figure 4.15 shows the computation time for large data sizes. As can be seen from the figures, our algorithm performs better on the WGAUSS_MULTI dataset which contains fewer flat regions. The time to cancel the gradient pairs that cross shared boundaries is lesser than the time taken to perform a persistence based simplification of the MS complex. This is reflected in the time taken to merge the sub-domains of the WGAUSS dataset. Figure 4.15 also plots running time for increasing data sizes of the WGAUSS dataset. The scaling results are similar for various values of persistence threshold. We note that the curves deviate away

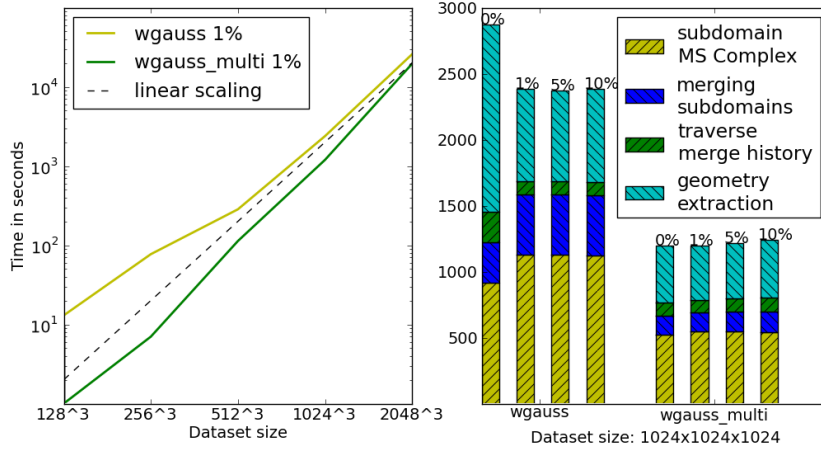


Figure 4.15: (left) Scaling behavior with varying sizes of the WGAUSS dataset simplified using a threshold of 1%. (right) Computation times for stages 2-5 for the 1024³ datasets. The persistence threshold for each run is shown above the bar-plots. Time taken to split the data into sub-domains is approximately 3 minutes.

Dataset	size	#crits.	time	(a)	(b)
Silicium	98 × 34 ²	1375	0.1s	3s	-
Fuel	64 ³	773	0.2s	5s	-
Neghip	64 ³	5663	0.3s	16s	7s
Hydrogen	128 ³	26725	1.5s	69s	47s
Anuerism	256 ³	95865	15s	118m	5m
WG	1024 ³	13531699	42m	-	-
WG_M	1024 ³	4599	20m	-	-
VORTEX- μ	1024 ³	1266976	32m	-	-
WG	2048 ³	54141119	464m	-	-
WG_M	2048 ³	4575	370m	-	-

Table 4.1: Timings for datasets available from <http://volvis.org/> compared with timings to compute the MS complex as reported in (a) [14] and (b) [12].

from linear scaling with increasing data sizes. On detailed analysis, we observed that stages two and five scale linearly whereas stages three and four did not. This is because of the representation of the combinatorial MS complex as a list of associative arrays.

Performance

To verify the benefits of parallelization, we compare our algorithm against existing methods [12, 14] on datasets available from <http://volvis.org/>, and a vortex flow data set, see Table 4.1. The experimental results indicate orders of magnitude improvement in the running time. Further, the memory required by our algorithm is less than 2GB even for the larger WGAUSS dataset. Gyulassy *et al.* and Günther *et al.* [12, 14] report 23h and 5h, respectively, to process data sizes close to 1024^3 . Figure 4.16 shows the critical 2- and 3-cells along with the ascending manifolds of 2-saddles for various datasets as well as slice visualizations of the WGAUSS dataset along with the decomposition of one sub-domain into descending / ascending manifolds of maxima / minima.

4.6 Persistence and Simplification in 3D

Simplification of the Morse-Smale complex refers to the systematic cancellation of pairs of critical points to obtain a simpler Morse-Smale complex with fewer critical points (see Section 2.2). Topological persistence [24] lays a theoretical framework that may be adapted for simplification of the Morse-Smale complex. As one sweeps across the range of the function, topological persistence tracks the evolution of the topology of sub-level sets of the input scalar function. During the sweep, it records pairs of critical points that correspond to the creation and

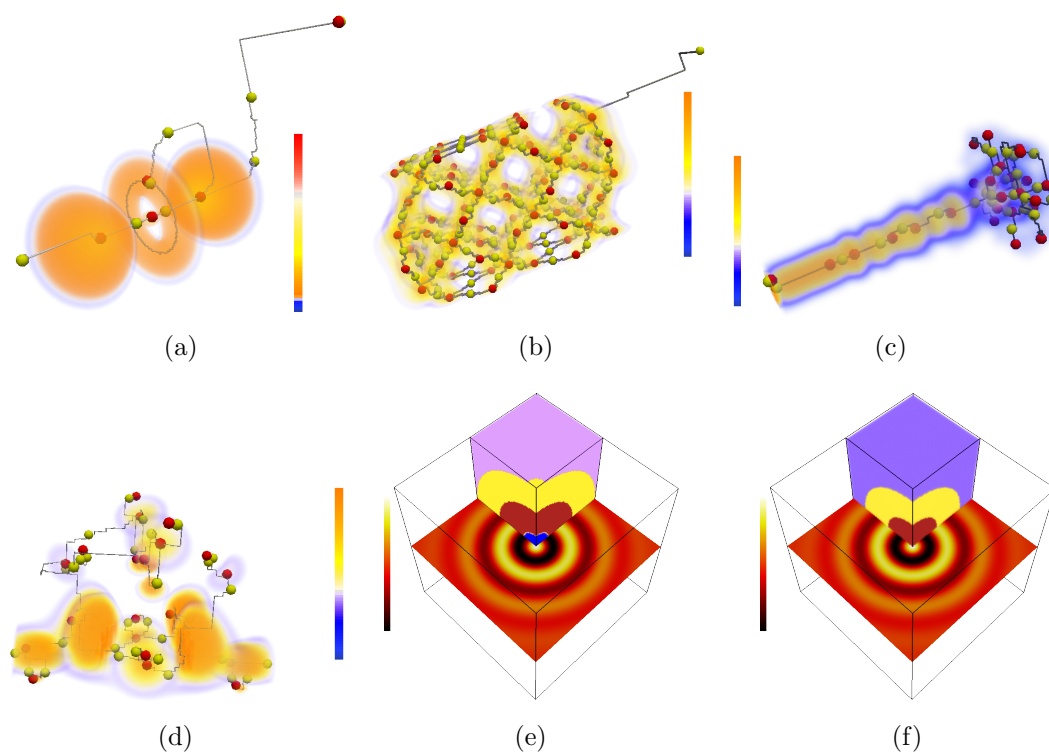


Figure 4.16: Ascending arcs between 2-saddles and maxima shown with the volume rendered image for (a) Hydrogen, (b) Silicium, (c) Fuel and (d) Neghip Datasets. Segmentation of the WGAUSS dataset into (e) descending manifolds (f) ascending manifolds.

destruction of topological features in the sub-level set. In 3D, the topological features are connected components, loops, and voids, whose creation and destruction is tracked by minimum-1-saddle, 1-saddle-2-saddle, and 2-saddle-maximum critical point pairs respectively. Such pairs of critical points are referred to as *persistent* pairs. The relative importance of a feature is measured by its life-time while sweeping across the function range, *i.e.*, the absolute difference in the function values of the creator and destroyer critical points. This life-time is said to be the *persistence* of the topological feature that is represented by the pair of critical points. Thus, features whose life time is smaller are deemed to be less relevant to those that “persist” for a larger life time. Edelsbrunner *et al.* [30] show that the least persistent critical point pair is always connected by an arc in the 2D Morse-Smale complex. Thus, one may simplify arcs of a 2D Morse-Smale complex so that the critical points at the end of the arcs correspond exactly to the persistent critical points. The simplification is iteratively carried out by canceling the arc with least absolute difference in function value of its two critical points. This method, being equivalent to persistence in 2D, is referred to henceforth as the *least absolute difference in function value measure*, abbreviated as the LADF measure.

For the 3D Morse-Smale complex, it is not theoretically known if the LADF measure is equivalent to persistence. In this section, we prove two lemmas to show that simplification of extremum-saddle pairs (maximum-2-saddle and minimum-1-saddle pairs) using the LADF measure leads to pairs that are consistent with topological persistence. The first lemma, the strong adjacency lemma, is an extension of the adjacency lemma proved by Edelsbrunner *et al.* [30]. The underlined terms indicate the difference between the adjacency lemma and the strong adjacency lemma.

Lemma 4.5: STRONG ADJACENCY LEMMA. *For every positive i , the i -th pair of minimum-1-saddle critical points ordered by persistence forms a unique arc in*

the complex obtained by canceling the first $i - 1$ minimum-1-saddle pairs.

Proof. The existence of an arc between the i -th minimum-1-saddle pair after canceling the preceding $i - 1$ minimum-1-saddle pairs follows from the same argument in the proof of the Adjacency Lemma. This is because the path that connects the i -th minimum-1-saddle may form after canceling other minimum-1-saddle pairs, not after canceling 1-saddle-2-saddle pairs or 2-saddle-maximum pairs.

Thus it remains only to prove the uniqueness. Let the i -th minimum-1-saddle pair ordered by persistence be m, s . Let $j + 1$ be the time at which s appears, and let K_1^j and K_2^j be the non intersecting connected components joined by s , where $m \in K_1^j$. Thus, the other descending arc from the saddle s connects to a minimum $m' \neq m$ contained in K_2^j .

If m' is not paired, there is nothing to prove. Assuming otherwise, let s' be the saddle paired with m' . If (m', s') are sequenced after (m, s) , there is nothing to prove. Let us assume (m', s') is sequenced before (m, s) . Let $j' + 1$ be the time at which s' appears and let $K_1^{j'}$ and $K_2^{j'}$ be the non intersecting connected components joined by s' , where $m' \in K_1^{j'}$.

It is sufficient to show that $K_2^{j'}$ and K_1^j do not intersect, so that the other arc from s extends to another minimum $m'' \in K_2^{j'}$ that is not m . If we show this, then we may reiterate the above argument with $m'' \neq m$ replacing m' . If $j' < j$, then $K_1^{j'}, K_2^{j'} \subset K_2^j$, and because K_1^j, K_2^j do not intersect, we have that $K_1^j, K_2^{j'}$ do not intersect. If $j' > j$, then $K_1^j, K_2^j \subset K_1^{j'}$, therefore we have that $K_1^j, K_2^{j'}$ do not intersect. As the number of minima is finite, the above argument may be reiterated to show that one arc from s terminates at some minimum that is not m . This establishes the uniqueness of the arc from s to m . \square

The above lemma ensures that a persistent minimum-1-saddle pair is connected by a unique arc in the Morse-Smale complex if the the preceding persistent

minimum-1-saddle pairs have been canceled. We now show that iteratively canceling minimum-1-saddle arcs in the Morse-Smale complex using the LADF measure results in persistent minimum-1-saddle pairs.

Lemma 4.6: PERSISTENCE-LADF EQUIVALENCE LEMMA. *Minimum-1-saddle pairs that result from iteratively canceling the singular minimum-1-saddle arcs with LADF measure are persistent minimum-1-saddle pairs.*

Proof. Assume that (m, s') is the first non persistent minimum-1-saddle arc we cancel as the Morse-Smale complex is iteratively simplified by canceling singularly connected minimum-1-saddle arcs using the LADF measure. Let $\delta = f(s') - f(m)$.

First, we claim that all persistent minimum-1-saddle pairs having persistence less than δ have been canceled. Since, this is the first non persistent minimum-saddle arc to be canceled, all preceding arcs were persistent minimum-1-saddle pairs. By the strong adjacency lemma we know that the least persistent minimum-1-saddle pair that is not yet canceled is connected by a unique arc in the Morse-Smale complex. If this arc had persistence less than δ it would have been canceled. Iterating this argument over the sequence of persistent minimum-1-saddle pairs, all persistent minimum-1-saddle pairs having persistence less than δ have been canceled.

The pair (m, s') may not be a persistent pair because, either m is the global minimum and therefore does not form a persistent pair at all or it does form a persistent pair with a saddle s . In the first case, consider the saddle s' . It has one descending arc that terminates at the global minimum m . Since this arc is singular, the other descending arc from s' terminates at another minimum m' . Since m is the global minimum, $f(m) < f(m')$. But that would make the arc (m', s') the singular arc with least absolute difference in function value. Thus, (m, s') could not have been the singular arc with the least absolute difference in function value, and hence we have a contradiction.

Now assume that (m, s) form a persistent pair. Since we already have canceled all persistent minimum-1-saddle pairs having persistence less than δ , we have $f(s) - f(m) > \delta = f(s') - f(m)$, *i.e.*, $f(s) > f(s')$. The other descending arc from s' terminates at some minimum m' contained in the connected component K^j , where $j + 1$ is the time at which s arrives. Since s destroys the component created by m , m is the lowest minimum in the component in K^j . If $m' = m$, then (m, s') is a non-singular arc and thus cannot be canceled. If not, then m' is paired by persistence with a saddle s'' that enabled it to union into the component K^j . Also, we have that $f(m') > f(m)$ and $f(s'') < f(s)$. Since we have canceled all persistent pairs with persistence less than δ , the arc $s'm'$ should now extend to lower minimum m'' in K^j . Reiterating the argument, the other arc must be connected to m . Thus (s', m) has to be multiply connected making the cancellation impossible. \square

By duality, the above two lemmas may be applied to maximum-2-saddle pairs. This is because, the cancellation of maximum-2-saddle never affects the combinatorial of connectivity minimum-1-saddles and vice-versa. Thus, we may simultaneously and iteratively cancel extremum-saddle arcs in the Morse-Smale complex using the LADF measure so that the resulting sequence of pairs are indeed persistent pairs.

We also infer from the above lemmas that the remaining 1-saddles and 2-saddles neither destroy 0-cycles nor create 2-cycles. Otherwise, they would have formed persistent extremum-saddle pairs. In the case when the domain is \mathbb{R}^3 or \mathbb{S}^3 , we know that these 1-saddles and 2-saddles necessarily pair with each other so that none remain unpaired. From a topological standpoint, each such 2-saddle necessarily destroys a 1-cycle created by some 1-saddle. However, possibly due to discretization of gradients, we find in many situations that these critical points are non-singularly connected. Another possibility is that the least persistent 2-saddle-1-saddle arc may not be connected by an arc in the Morse-Smale complex. Here,

it is seen that if the discrete Morse-Smale complex is constructed by gradient pairs such that they are persistent pairs for some filtration of the domain, then the least persistent 2-saddle-1-saddle pair is connected by an arc. This is because the process of gradient traversal via boundary expansion is analogous to the *cycle search* step of the persistence computation algorithm of Edelsbrunner *et al.* [30]. Thus, by the correctness arguments, we may infer that there exists at least one gradient path between the least persistent 2-saddle-1-saddle pair. However, the difficulty arises when we attempt to iterate the argument when persistent 2-saddle-1-saddle pairs are not eliminated due to non-singular connectivity. For the above reasons, in applications that prioritize topological considerations, we believe it is justified to eliminate these 1-saddle-2-saddle pairs despite the non-singular connectivity. Here, we conjecture that iteratively eliminating singular extremum-saddle arcs and then 2-saddle-1-saddle arcs with odd multiplicity, ordered using the LADF measure yields persistent 2-saddle-1-saddle pairs also.

The above simplification scheme requires simplification of extremum-saddles to be completely determined and then proceeds to eliminate 2-saddle-1-saddle pairs. In some applications, it is desirable for the sequence to be conjoined and not separate. In such cases, we may first determine the sequences of pairs and then merge the two sequences which are then ordered using the LADF measure. This is still a valid sequence of cancellation pairs. This is because the arcs of one type do not form parts of arcs of another type. For example, a 2-saddle-maximum arc will never be part of any 2-saddle-1-saddle arc, *i.e.*, the formation of a 2-saddle-1-saddle arc is only influenced by other 2-saddle-1-saddle arcs. Thus, even though the sequences are merged, the relevant arcs will form at the appropriate time because the relative order of each type of arc is unchanged.

4.7 Conclusions

In this chapter, we have discussed an efficient parallel algorithm to compute the Morse-Smale complex of scalar functions defined on large two-dimensional and three-dimensional domains. This is done by laying robust theoretical foundations for consistent and independent computation of the discrete gradient field. The validity and the quality of the gradient field is also proved using theoretical arguments. Efficient algorithms for traversal of the gradient fields in GPUs and multicore-CPU are discussed. These algorithms were implemented and evaluated for 2D and 3D structured grids. The evaluations show near linear scaling of the algorithms with data size and computing cores. We compare our synergistic CPU-GPU 3D implementation with other algorithms to demonstrate an order of magnitude improvement in run-times over earlier algorithms for common visualization datasets. In the last part of the chapter, we discussed our contributions towards the analysis of simplification of the Morse-Smale complex. In particular, we showed that it is possible to simplify extrema-saddle pairs in the 3D Morse-Smale complex in a manner that is consistent with topological persistence.

Chapter 5

Exploring Cosmic Filaments

In this chapter, we discuss the application of the Morse-Smale complex to visually explore the filamentary structure of the cosmic web¹. The Morse-Smale complex plays key roles in both identifying filamentary structures and filtering relevant filaments based on user parameters. In Section 5.1, we begin with a brief introduction to the cosmic-web and the challenges present in its study. Next, we highlight significant related work (Section 5.2) and then discuss the key contributions of our Morse-Smale complex based approach (Section 5.3). In Section 5.4, we discuss Felix’s methodology. Section 5.5 discusses the cosmological datasets that we use in our experiments in detail. In Section 5.6, we discuss experiments to validate our approach as well as showcase the advantages of Felix. We conclude this chapter with the possible future directions of our work in Section 5.7.

¹This work was carried out in collaboration with Pratyush Pranav, Rien van de Weygaert, E G Patrick Bos, and Steven Rieder who are with the Kapteyn Astronomical Institute.

5.1 Introduction

At scales from a megaparsec to a few hundred megaparsecs², the universe has a web-like appearance. In the *cosmic web* [25, 49], galaxies, intergalactic gas, and dark matter have aggregated in an intricate wispy spatial pattern marked by dense compact *clusters*, elongated *filaments* and sheetlike *walls*, and large near-empty void regions. The filaments, stretching out as giant tentacles from the dense cluster nodes, serve as transport channels along which mass flows towards the clusters. They surround the flattened walls, which are tenuous, membrane-like features in the cosmic mass distribution.

All structure and objects in the universe emerged out of primordial fluctuations that were generated during the *inflationary era*, moments after its birth, as the universe underwent a rapid phase of expansion [50, 51, 52]. The quantum fluctuations generated during this phase manifest themselves as fluctuations in the temperature of the cosmic microwave background [53, 54, 55]. The gravitational growth of these density and velocity perturbations has resulted in the wealth of structure that we see in the Universe. The web-like patterns mark the transition phase from the primordial Gaussian random field to highly nonlinear structures that have fully collapsed into halos and galaxies. As our insight into the complex structural pattern of the cosmic web has increased rapidly over the past years, it has become clear that the cosmic-web contains a wealth of information on a range of cosmological and astronomical aspects and processes.

An important illustration of the cosmological significance of the cosmic web concerns its dependence on the nature of dark energy and matter, the dominant but as yet unidentified forms of energy and matter in the Universe. One telling

²A parsec is the standard unit of measurement of distances in the cosmos. A parsec is 3.26 times the *light-year*, the distance light covers in a year. A megaparsec is a million parsecs, the typical scale of measurement of size of the large scale structures in the universe.

example of this is the recent realization that cosmic voids are sensitive and useful probes of the nature of dark energy and dark matter and testing grounds for modified gravity theories [56, 57, 58, 59, 60, 61, 62, 63]. As the cosmic web is first and foremost defined and shaped by the gravitationally dominant dark matter, it would be of considerable importance to be able to obtain detailed maps of dark matter distribution. In recent years, great strides have been made towards this goal as gravitational lensing of distant galaxies and objects by the dark matter have enabled an increasingly accurate view of its spatial distribution [64, 65]. Initial efforts concentrated on the detection and mapping of the deep potential wells of the nodes in the cosmic web, *i.e.*, of galaxy clusters. Recent results have opened the path towards the mapping of filaments via their lensing effect on background sources [66]. The identification of the structural components of the cosmic web is also important for our understanding of the relation between the formation, evolution, and properties of galaxies and the structural environment of the cosmic web. A direct manifestation of this is the generation of the angular momentum of galaxies. This is a product of the torqueing by the large-scale tidal force field [67, 68, 69]. While these are also the agent for the formation and shaping of filaments, we would expect that this results in the alignment of the spin axis of galaxies with respect to cosmic filaments [70, 71, 72].

The identification, description, and characterization of the elements of the cosmic web is a non-trivial problem. Several characteristics of the mass distribution in the cosmic web have made it an extremely challenging task to devise an appropriate recipe for identifying them:

a) The cosmic web is a complex spatial pattern of connected structures displaying a rich geometry with multiple morphologies and shapes.

b) There are no well-defined structural objects at a single spatial scale or within a specific density range. Instead, elements of cosmic web are found at a wide range

of densities and spatial scales. This is a consequence of the hierarchical evolution of structure formation in the universe, such that smaller high-density structures merge to form larger objects.

c) There is a clear anisotropy in the structures of the cosmic web, a consequence of gravitational instability. The structures in the cosmic web exhibit elongated and flattened characteristics.

The attempts to analyze the structure of the cosmic web has a long history. The absence of an objective and quantitatively accurate procedure for identifying and isolating the components of the cosmic web has been a major obstacle in describing it. In recent years, more elaborate and advanced techniques have been developed to analyze and describe the structural patterns in the cosmic web. Nonetheless, a consensus on the proper definition of filaments is yet to be achieved. In Section 5.2, we present a short account of the available techniques and the definitions on which they are based.

5.2 Related Work

Statistical measures such as the auto-correlation function [73] of the matter distribution in the web have been the mainstay of cosmological studies over many decades. However, while this second-order measure of clustering does not contain any phase information (one may e.g. always reproduce a distribution with the same 2nd order moments and random Fourier phases), the auto-correlation function is not sensitive to the existence of complex spatial patterns. Higher order correlation functions only contain a very limited amount of such structural information, while in practical observational circumstances it quickly becomes cumbersome to measure them. The magnitude of the error increases drastically with increasing order.

The first attempts towards characterizing complex geometric patterns in the galaxy distribution mainly involved heuristic measures. Early examples of techniques addressing the global connectivity of structure in the Universe are percolation analysis [74] and the minimum spanning tree of the spatial galaxy distribution [75, 76]. While these are useful global descriptions, they do not capture and describe local characteristics of the mass distribution.

More elaborate and advanced techniques have been developed in recent years. Several of these methods apply sophisticated mathematical and visualization techniques, involving geometric and topological properties of the cosmic mass distribution. There are a multitude of different methods for detecting filaments, based on a range of different techniques. We may recognize several categories of techniques.

One class of methods seeks to describe the local geometry on the basis of the Hessian of the density field [77, 78, 79, 80] or closely related quantities such as the tidal force field [81, 82] or the velocity shear field [83, 80]. The Hessian provides direct information on the local shape and dynamical impact of the corresponding field. The morphological elements of the cosmic web are identified by connecting the areas within which a specific range of anisotropies is registered.

These studies concentrate on a single scale by appropriately smoothing the field, and do not consider the multi-scale nature of the cosmic mass distribution. The Hessian based Nexus/MMF technique, introduced by Aragon-Calvo *et al.* [78] and perfected into a versatile and parameter-free method by Cautun *et al.* [80], implicitly takes into account the multi-scale nature of the web-related fields. It accomplishes this by a scale-space analysis of the fields. At each location the optimal morphological signal is extracted via the application of a sophisticated filter bank applied to the Hessian of the corresponding fields in scale space. The application of this machinery has enabled thorough studies of the hierarchical evolution and buildup of the cosmic web [84].

A promising and highly interesting recent development has opened up the path towards dynamical analysis of the evolving mass distribution in full six-dimensional phase-space (in which the position of each mass element is specified by its space coordinates and velocity/momentum). In the 6D phase space, the cosmic mass distributions defines a 3D sheet. Independently, three groups arrived at tessellation based formalisms that exploit the evolving structure and folding of the *phase space sheet* in phase space [85, 86, 87] (also see e.g. [88]). The number of folds of the phase space sheet at a given location indicates the number of local velocity streams, and forms a direct indication of the morphology of the local structure. Interestingly, the resulting characterization of the web-like distribution, the Origami formalism of Neyrinck [87] for example, appears to resemble that of the Nexus/MMF formalism [84].

An entirely different class of techniques is based on a thorough Bayesian statistical analysis of the Bisous model. Stoica *et al.* [89] model the filamentary galaxy distribution within the context of a stochastic geometric model involving a random configuration of interacting cylindrical segments. It has been developed into a versatile, statistically solid yet computationally challenging formalism for the identification of filaments in a spatial point distribution, such as N-body simulations and galaxy redshift surveys [90]. An additional example of a method involving statistical analysis of a geometric model is that of Genovese *et al.* [91], which seeks to describe the filamentary patterns of the cosmic web in a non-parametric way by recovering the medial axis [92] of the point-set of galaxies.

The fourth major class of methods, the one which we will also pursue, exploits the topological structure of the cosmic mass distribution. The fundamental basis of these methods is Morse theory [38]. The geometric structure of the *Morse-Smale complex* [30] naturally delineates the various morphological components on the basis of the connections between the critical points of the density fields and

the higher dimensional cells that are incident on the critical points. Various Morse theory based formalisms have been applied to the identification of components of the cosmic web. One of the first applications concerned the detection of voids in the cosmic density field. The Watershed Void Finder [93] identifies these with the watershed basins around the density minima. The SpineWeb procedure [94] extended the watershed transform towards the detection of the full array of structural components, filaments, walls and voids. These techniques use a user-defined filter to incorporate the multi-scale structure of the cosmic density field.

A natural topological means to address the multi-scale topological structure emanates from the concept of persistence [24]. It provides a natural recipe for detecting and quantifying the components of the cosmic web in a truly hierarchical fashion. Sousbie [9, 10] has exploited and framed this in an elegant and impressive framework, the DisPerSE formalism. Following the construction of the Morse-Smale complex, they proceed to simplify it. The simplification proceeds by canceling pairs of critical points iteratively, where each pair represents a structure in the cosmic web. Topological persistence is invoked to order the critical point pairs. However, this measure of importance is not unique, and one may consider alternatives, dependent on the specific interest and purpose.

In effect, to tackle similar issues in other visualization areas, a range of variations have been proposed in other studies [20, 95, 96]. Weinkauff *et al.* [96] describe the concept of separatrix persistence, where they compute the strength of separation of points on a separatrix curve (in 2D) connected to a saddle as the sum of the absolute differences of function values of the saddle and the extrema connected to it. This concept is extended to 3D separating sheets by Gunther *et al.* [20] in a slightly modified form. Reininghaus *et al.* [95] develop the concept of scale-space persistence where they accumulate the absolute difference in function value measure of critical points across a hierarchy of derived functions. The set

of derived functions are generated by smoothing the function using a Gaussian kernels of increasing variances. This is similar to the Multi-scale Morphology Filter Nexus/MMF [78, 80] described above. Both methods adopt the scale space formalism as the first step to detect features at multiple scales. However, scale space persistence and separatrix persistence, disregard specific density regimes of interest and are potentially inappropriate when small scale features with specific density characteristics are of interest.

5.3 Contributions

In the present study, we describe and introduce a technique for the identification of filaments based on the topological characteristics of the density field. A key aspect of the proposed technique is its interactive nature, involving a tunable density parameter. Specifically, we describe the following contributions:

a) We describe Felix³: a topology based framework for visual exploration of filaments in the cosmic web. In particular, we develop a query framework to extract filamentary structures from a hierarchy of Morse-Smale complexes of the density field. The filaments in Felix are parameterized by the density values of the maxima and the 2-saddle that define them.

b) Using Felix, we develop a semi automatic structure finder that classifies galaxies as cluster/filamentary or not. We demonstrate its efficiency through two tests. First, using the Voronoi Kinematic model as a benchmark, we demonstrate that we are able to recover the classification with high efficiency. Second, we show that the classifications are quantitatively comparable to, and in several cases better than, existing classifiers.

c) We investigate the nature of filaments in two realizations of the Λ CDM

³The name Felix is formed from an abbreviation of *F*ilament *e*xplorer.

simulations. In the first dataset, we investigate filaments concerning two density regimes. The first concerns filaments in the high density regions around compact dense clusters, which are known to function as the transport channels along which matter moves into the clusters. A second regime concerns the tenuous low-density filaments found in low-density void regions. A video demonstrating the visual exploration process in this experiment is available at <https://youtu.be/8TRtX937Xjk>. In the second dataset, we investigate the nature of three classes of filaments in a relatively cleaner region of a Λ CDM dataset. The first class consists of filaments within clusters, the second consists of filaments within void-like regions, and the third consists of those that extend across cluster like regions, which forms the spine of the cosmic web.

d) We describe an efficient structure based volume rendering enhancement routine that allows us to highlight the density distribution in regions that are close to the selected features.

The distinction between noise and significant structures is often ill-defined, and at occasions noise may be confused with genuine structures in the hierarchically evolved mass distribution (see Figure 5.2 and the caption thereof for an illustration). This problem is more pronounced when one studies the properties of tenuous filaments and walls in low density void-like regions. For the understanding of the formation and evolution of galaxies in such regions, we need to assess the possible dependence of galaxy and halo properties on the morphology and density of the local environment. This must be based on the successful extraction of filaments in low density regions and the correct identification of galaxies associated with them. In view of this, we include an interactive handle on the density regimes so that one can concentrate on and probe structures in specific density regimes.

5.4 Methodology

Exploring the filamentary patterns of the cosmic web is challenging because of the large range of the spatial scales and density range it exhibits. A proper characterization should also account for the hierarchical nature of structures, which adds considerable challenge to the task. Though there exist different notions of filaments, the primary evidence relied upon for extraction and analysis is most often visual. It is therefore not surprising that structure finding methods often visually verify results by superimposing the extracted structures upon visualizations of the density field or the particle distribution. However the visualization plays a role only after structure extraction process in these methods. We differ in this respect by providing the capability to interact with the structure finding procedure and extract structures that are visually relevant. To accomplish such a visual exploration framework, a succinct model of filament definition, an efficient representation of hierarchical structures, and an appropriate query mechanism that supports the extraction of these structures are paramount. The following exposition details our framework on these terms.

5.4.1 Density estimation and filament modeling

Cosmological simulations are N-body particle experiments that simulate structure formation and evolution by tracing positions of the particles under the influence of physical laws. In the observational reality, the information about structures in the cosmos comes through observing the galaxies. The galaxies can be treated as particles also for the purpose of analysis in the context of large scale structures.

The input to Felix is the logarithm of the density scalar field on the given 3D domain of interest. The domain could be 3D structured grids or tetrahedral meshes, with the density specified on the vertices of the grid/mesh. We find that

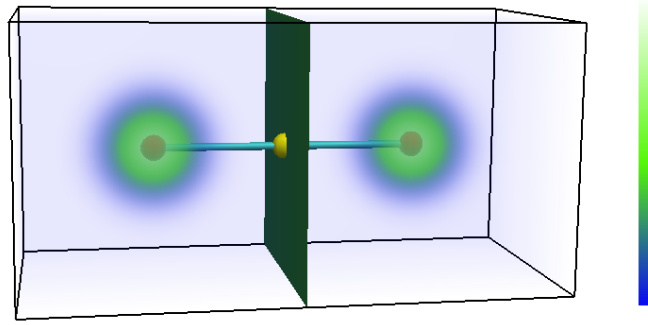


Figure 5.1: Ascending manifolds of a 2-saddle (yellow sphere). The scalar function is a sum of two 3D Gaussians centered on either side of the volume. The two arcs incident on the 2-saddle constitute the ascending manifold and terminate at the two maxima (red spheres) of the scalar function.

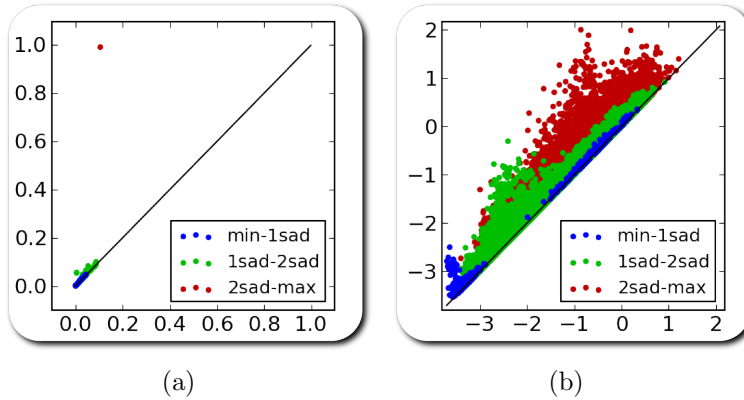


Figure 5.2: (a) A scatter plot of the function values of the canceled critical point pairs for the function shown in Figure 5.1. A 2-saddle-maximum pair is the only pair that is far removed from the diagonal. This corresponds to cancellation of the 2-saddle with a maximum that represents one of the Gaussians in Figure 5.1. Other pairs close to the diagonal represent insignificant features that manifest due to the added Gaussian noise as well as sampling noise. (b) A scatter plot of the function values of the canceled critical point pairs for the Voronoi-Kinematic dataset B (see Section 5.5.1). No discernible separation of points is seen, though there are many points that are far removed from the diagonal. Thus, no clear global simplification threshold may be used for filament extraction.

the logarithm of the density field, instead of the density field itself, resolves the structures with more visual clarity. This has also been independently established in an earlier work [80]. Additionally, the input may be specified as a distribution of particles within a 3D region of interest. This could be a snapshot from a cosmological simulation, or galaxies in real observational data. We use the Delaunay tessellation field estimator (DTFE) [97, 98] to estimate the density of the input particles in the 3D region of interest. This procedure begins by computing the periodic Delaunay triangulation [99] on the points (simulation particles or galaxies). Next, the density at each vertex of the triangulation is estimated by the inverse of the volume of the tetrahedra incident upon it. Finally, the density is linearly interpolated onto the edges, faces, and tetrahedra of the Delaunay triangulation to yield a piecewise linear density function on the domain. The periodic Delaunay triangulation, computed by the DTFE procedure, is used to represent the domain.

The Morse-Smale complex of the logarithm of the density field is computed. Filaments are modeled as the ascending manifolds of 2-saddles of the Morse-Smale complex. These arcs represent paths of steepest descent from the two maxima merging at the 2-saddle. This 2-saddle represents the lowest density point along the arcs connecting the two maxima. A schematic illustration of this is presented in Figure 5.1. There are many algorithms available in the literature to compute the 3D Morse-Smale complex. The algorithms are primarily based on either the quasi Morse-Smale complex formulation [100, 31] or Forman’s [21] discrete Morse theory [12, 13, 15, 16]. We use the parallel algorithm described in Chapter 4, resulting in fast computation even for large datasets.

The density field is rarely smooth and several local maxima obscure a view of the larger scale behavior of the density field. This is especially true if the density field is computed on the raw particle distribution, where the density field tends to be spiky and with a lot of fluctuations in the high density cluster-like regions.

The Morse-Smale complex is simplified by iteratively canceling pairs of singularly connected critical points with least absolute difference in function value to generate a hierarchy of Morse-Smale complexes.

In most applications, a specific version of the Morse-Smale complex from the hierarchy is chosen based on a perceptibly clear separation of noise and features. One way to choose such a threshold separating noise and feature is by using a scatter plot of the function values of canceled critical point pairs (see Figure 5.2) where the lower function value among the pair corresponds to the x-coordinate and the higher function value corresponds to the y-coordinate. In datasets where topological features are well separated (see Figure 5.2(a)), pairs representing significant features appear far away and isolated from the diagonal. In such cases, the coarsest Morse-Smale complex version wherein the insignificant pairs are removed is selected for feature analysis/extraction. However, this strategy is not easily applicable to cosmology datasets (see Figure 5.2(b)). A well defined separation is rarely discernible, though there are many scatter points that are far removed from the diagonal. Hence, we drop the assumption that we must work with a specific version. Instead, we query for features across all Morse-Smale complexes in the hierarchy, as discussed in the following sub-section.

5.4.2 Density range based filament selection

Cosmic filaments exhibit a large range of variation in their density characteristics. Indeed, one expects filaments to be present both in void like regions and between cluster like regions. While strong dense filaments in between clusters define the spine of the Cosmic Web, in the hierarchically evolving mass distribution we encounter a wide spectrum of ever more tenuous filaments on smaller mass scales. Small filaments define the directions of mass inflow into galaxies, and form a crucial component in the formation of galaxies [101]. Even more tenuous are the systems

of filaments stretching over the hollows of voids, often conspicuously aligned along the direction defined by neighbouring superstructures. The understanding of this network is tightly related to the issue of the “missing” dwarf galaxies in voids [102]. While this illustrates the complexity of the multiscale filigree of filaments in the Cosmic Web, we follow a strategy in which we focus our attention on specific aspects and details of the cosmic web. Dependent on the identity of objects and structures of interest, we wish to be able to zoom in on to the corresponding filamentary network. This is largely dependent on the mass scales of the objects involved, and the density values of the corresponding filament generating density peaks [103, 84].

Following this rationale, we translate this strategy into the use of queries that depend on the density properties of interest. Specifically, we query for filaments by specifying the density range $[M_b, M_e]$ of the clusters they connect (the maxima at the end points), as well as the density range $[S_b, S_e]$ of the lowest point along the connecting path (the density range of the 2-saddles). Figure 5.3 conceptually illustrates the characterization of filaments using density ranges, where density along filaments vary significantly necessitating simplification.

Algorithm [SELECT2SADDLES](#) lists the algorithm to process such a query. The algorithm accepts, together with the combinatorial Morse-Smale complex MSC , the density ranges $[S_b, S_e]$ and $[M_b, M_e]$ as input. The algorithm returns a list of 2-saddles that satisfy the above criteria together with the maximal Morse-Smale complex version in which they do so.

The algorithm begins by creating a list S of 2-saddles that have their function value in the given 2-saddle range $[S_b, S_e]$. Then, for each 2-saddle in S , a Morse-Smale complex version in which it possibly connects two maxima within $[M_b, M_e]$ is computed. The appropriate version is given by the minimum of three version indices v_a, v_b and v_c .

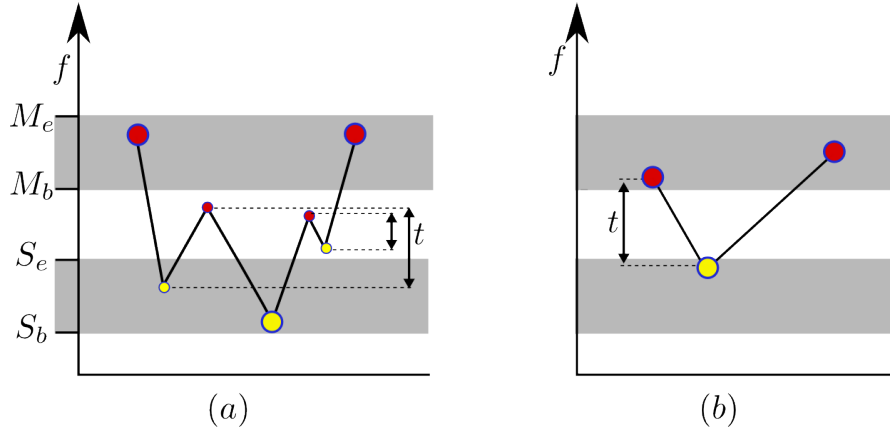


Figure 5.3: (a) Filaments are modeled as the ascending paths of 2-saddles connecting two extrema. The 2-saddles are filtered based on the range constraints $[M_b, M_e]$ and $[S_b, S_e]$ on the highest and lowest values respectively along the ascending paths. The highest values along the 2-saddle's ascending manifold are at extrema and the lowest value is at the 2-saddle. The function along the paths is rarely smooth and needs to be simplified. In the illustration, a simplification threshold of t reveals a filament with appropriate density characteristics. However, imposing such a threshold uniformly will cause another filament (b) having the required density characteristics to be destroyed. It is therefore necessary to extract filaments by querying all Morse-Smale complexes within a given hierarchy.

Algorithm 5.1 SELECT2SADDLES

```

1: procedure SELECT2SADDLES( $[S_b, S_e], [M_b, M_e]$ )
2:    $Sver \leftarrow$  Empty Map
3:    $S \leftarrow \{s \mid s \text{ is a 2-saddle}, S_b < f(s) < S_e\}$ 
4:   for all  $s \in S$  do
5:      $v_a \leftarrow \text{Max}\{i \mid s \text{ is not simplified in } MSC_i\}$ 
6:      $v_b \leftarrow \text{Max}\{i \mid t_i < M_b - f(s)\}$ 
7:      $v_c \leftarrow \text{Max}\{i \mid s \text{ connects distinct Maxima in } MSC_i\}$ 
8:      $Sver[s] \leftarrow \text{Min}(v_a, v_b, v_c)$ 
9:   Sort  $S$  by  $Sver$ 
10:   $Ssel =$  Empty Set
11:  for all  $s \in S$  do
12:     $i = Sver[s]$ 
13:     $m_a, m_b =$  Maxima connected to  $s$  in  $MSC_i$ 
14:    if  $M_b < f(m_a), f(m_b) < M_e$  then
15:      insert  $(s, Sver[s])$  in  $Ssel$ 
return  $Ssel$ 

```

The version index v_a is the finest Morse-Smale complex version in which the 2-saddle s survives. In other words, s is canceled in MSC_{v_a+1} but not in MSC_{v_a} . This is pre-computed by examining the cancellation sequence. The version index v_b corresponds to the last Morse-Smale complex version at which the 2-saddle s connects two maxima, both with function value less than M_e . This is possible because in successive versions of the Morse-Smale complex, the maxima connected to a 2-saddle via the same arc form an increasing sequence in terms of their function value. Thus, in the version where the absolute difference in function value of the last canceled pair is less than $M_b - f(s)$, the 2-saddle s still possibly connects two maxima with function value less than M_e . The version index v_c is the last Morse-Smale complex version at which the 2-saddle s separates distinct maxima. In other words, it is not a strangulation in MSC_{v_c} . As a consequence of the cancellation preconditions, once a strangulation is created by a 2-saddle, it may be destroyed only by canceling the 2-saddle with a 1-saddle. Thus there exists a maximal version index v_c after which the 2-saddle remains connected to a single maximum. The version index v_c is -1 when the 2-saddle is a strangulation in the initial Morse-Smale complex. In this case, the 2-saddle is not considered in further steps and is removed from S . Again, this is easily pre-computed for each 2-saddle by examining the cancellation sequence.

The 2-saddles in the set S are sorted based on their version indices. This is done to optimize switching between the required Morse-Smale complex versions. Next, each 2-saddle s in S is checked to see if it separates two maxima within the maxima density range $[M_b, M_e]$. The list of 2-saddles that fulfill all of the above criteria are returned together with their associated version numbers. This above list of 2-saddles is used to extract the filament geometry. Specifically, the ascending manifold of each 2-saddle is extracted from the corresponding version of the Morse-Smale complex. This may be done efficiently using the *cancellation*

merge DAG data structure discussed by Gyulassy *et al.* [104].

In some situations, it is desirable to perform some simplification to eliminate Poisson noise introduced due to meshing the domain. In these cases, a global simplification specifically for noise elimination, can be optionally introduced. Specifically, Algorithm `SELECT2SADDLES` returns only those 2-saddles that survive in hierarchical Morse-Smale complex versions above a specified threshold T_s , where T_s is specified as a normalized fraction of the range of log-density values (normalized to $[0, 1]$). T_s is set to 0.0 unless specifically mentioned. Similar to the inputs of Algorithm `SELECT2SADDLES`, T_s may be updated during run-time.

5.4.3 Volume Rendering and Enhancement

We use volume visualizations of the density field to aid selection of parameters for Algorithm `SELECT2SADDLES`. The geometry of the selected filaments using Algorithm `SELECT2SADDLES` is superimposed upon a volume rendering of the density field. Based on the visualization of the extracted filaments and the density volume rendering, the parameters may be adjusted so that the structures correspond with the density volume rendering. Figure 5.5 shows an example of the overlay of the volume visualization with the selected structures. For volume rendering, we use a standard implementation [105] of the Hardware Accelerated Projected Tetrahedra (HAPT) technique proposed by Maximo *et al.* [106], which in turn is an adaptation of the classic Projected Tetrahedra algorithm by Shirley and Tuchman [107].

Direct volume rendering of the density is often not effective for visualization because of the formation of clusters at multiple scales. Furthermore, these clusters are often spatially far removed from the features of interest. Figure 5.10(a) shows the visual clutter caused in a cosmic web density field.

We modify the HAPT algorithm to suppress the opacity of regions spatially far removed from filamentary features of interest. The input to our modified algorithm

is the set of selected features, in addition to the density field and a transfer function that translates density values to color and opacity. The HAPT algorithm projects tetrahedra in visibility order, while blending their raster pixels in a back-to-front order. The color and opacity of each raster pixel of a tetrahedron is given by the volume integral, which integrates color and opacity values along view rays as they enter and exit the tetrahedron. Moreland and Angel [108] propose a linearized form of the integral for efficient evaluation and acceptable quality. The integral resolves into a linear expression with one non-linear term obtained from a pre-integrated table. A key advantage of this method is that the input required for each view ray is the color and opacity value at the end points of each view ray along with the length of the view ray through the tetrahedron. In the HAPT algorithm, the density at the vertices of the tetrahedron is first linearly interpolated to the endpoints of the view-ray tetrahedron intersection and then translated to color and opacity values using the transfer function. In an analogous manner, we first compute the distance of the vertices from the selected set of features. This may be efficiently computed using a kd-tree [109] in $O(n \log n)$ time, where n is the number of vertices. Then, this distance is linearly interpolated to each end point of each view-ray tetrahedron intersection. The interpolated distance value at each end point of the view-ray tetrahedron is then used as an argument of a Gaussian to compute corresponding secondary opacity values. These secondary opacity values are multiplied by the opacity values obtained from the transfer function. The variance is used as a distance control parameter, where a lesser variance results in the suppression of the opacity of features that are spatially far removed from the selected set of features. The remainder of the HAPT algorithm is retained unchanged.

5.5 Model Description

In this section, we briefly describe the models used to test our filament detection routine. These are the Voronoi evolution models and Λ CDM cosmological simulations.

The Voronoi models provide us with vital quantitative information on the sensitivity of Felix to anisotropic filamentary patterns in the galaxy distribution. To this end it is of key importance that the Voronoi models have an a-priori known population fraction in different morphological elements: clusters, filaments, walls and voids. This makes them perfect test models for evaluating success and failure rates of the various identification methods.

Although they involve filaments with a broad distribution of densities, the Voronoi models do not incorporate the multi-scale web-like patterns we see in realistic cosmological scenarios. To assess this aspect of the cosmic mass distribution, we turn to simulations of structure formation in the standard Λ CDM cosmology. Implicitly these include all relevant physical and dynamical processes of the evolving cosmic dark matter distribution. However, as we have no control over all aspects of the emerging mass distribution in Λ CDM simulations, for testing purposes they are not as informative as Voronoi models ⁴

5.5.1 Voronoi evolution models

The Voronoi evolution models are a class of heuristic models for cellular distributions of galaxies that mimics the evolution of the Megaparsec universe towards a weblike pattern. They use Voronoi tessellations as a template for distribution of matter and related galaxy population [114, 115, 112], and its subsequent evolution.

⁴In the literature, several studies use simplistic models using Voronoi tessellations. The models we use here are considerably more sophisticated, and represent a rather realistic depiction of the cosmic web in void-dominated cosmologies, see e.g. [110, 111, 112, 113].

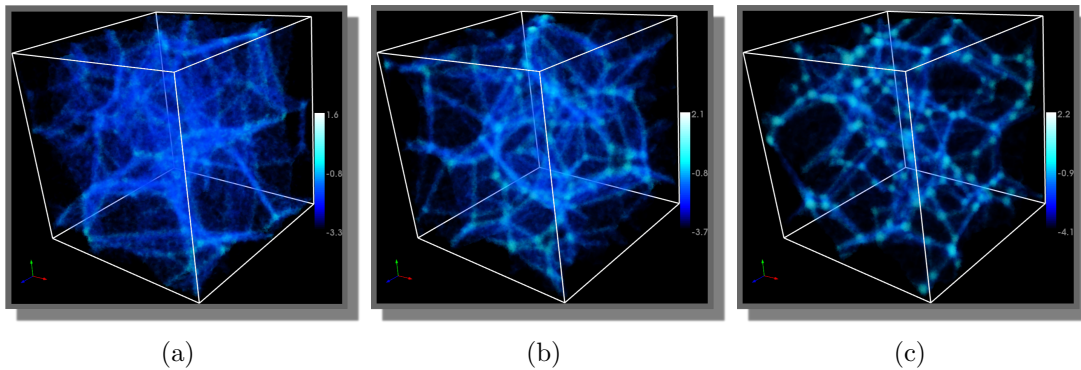


Figure 5.4: Density rendering of the snapshots in the Voronoi evolution time-series: (a) is the least evolved stage with almost 50 percent particles located in the Voronoi cells, (b) at a medium stage of evolution, while (c) is the most evolved stage of the model. We refer to Table 5.1 for a percentage distribution of particles in different morphological components.

In these models, one begins by fixing an underlying Voronoi skeleton, defined by a small set of randomly distributed nuclei in the simulation box. One then superposes a set of N randomly distributed particles on this skeleton. The resulting spatial distribution of particles in the model is obtained by projecting the initially random distribution of particles on to the faces, edges, and nodes of the Voronoi tessellation. This results in a pattern in which one can distinguish four structural components: *field* particles located in the interior of Voronoi cells, *wall* particles within and around the Voronoi faces, *filament* particles within and around the Voronoi edges and *cluster* particles within and around the Voronoi nodes.

One particular class of Voronoi clustering models are the Voronoi kinematic models, which seek to approximate the dynamical evolution of the large scale cosmic mass distribution. These models involve a continuous flow of galaxies towards the nearest wall, along a filament at the wall's edge, and subsequently towards the final destination, a vertex of the Voronoi tessellation. This motion is regulated by the increase of mean distance between the galaxies, an expression of void expansion and evacuation as a function of time.

	cell	wall	filament	cluster
A	29.88%	43.57%	22.20%	4.33%
B	9.82%	32.13%	38.62%	19.42%
C	3.5%	16.50%	28.70%	51.30%

Table 5.1: The relative abundance of particles in each structural element throughout the course of evolution.

The Voronoi models used in our experiments have 262,144 particles distributed along the vertices, edges, faces and cells of the Voronoi skeleton in a box of side-length $200h^{-1}\text{Mpc}$. The skeleton is generated by 32 randomly placed nuclei in the box. For the least evolved stage, most of the particles are in the cells, while for the most evolved stage, most of the particles are located in and around clusters. Table 5.1 presents the percentage distribution of particles in the various structural elements, as it changes with time. Stage 1 (dataset A) is the least evolved, while Stage 3 (dataset C) is the most evolved. The number density of particles in and around the nodes, edges, and walls are Gaussian distributed around these elements, characterized by a thickness scale R_f which is the standard deviation of the distribution. For our models, $R_f = 2h^{-1}\text{Mpc}$. Figure 5.4 presents volume renderings of the density distribution. From left to right, we depict progressively higher stages of evolution.

5.5.2 ΛCDM cosmological simulations

The ΛCDM simulations are fully physical models that trace the distribution and evolution of dark matter in the universe based on current understanding of real physical laws. Dark matter is the gravitationally dominant matter component in the Universe and constitutes the major fraction of matter. As it is known to only interact gravitationally, modeling the behavior of dark matter is computationally

fast and efficient. Such dark matter simulations form one of the principal tools towards understanding the evolution of the matter distribution in the Universe.

The cosmological simulations that we used follow the standard Λ CDM cosmology. In this model, the matter content of this Universe is dominated by collisionless *cold dark matter* (CDM) particles. The biggest contribution to the energy content of this Universe comes from dark energy, in the form of the cosmological constant Λ (see [116]), which drives its accelerate expansion at the current epoch.

We use two different Λ CDM simulations. They differ in the scale over which they simulate the Universe.

Cosmogrid datasets. To present the results of our visual exploration framework, we use the Cosmogrid simulation [117]. Cosmogrid is a suite of dark matter only high-resolution simulations that simulates the universe at a relatively small scale of $21h^{-1}\text{Mpc}$. Here, 512^3 particle realization was used, which achieves a mass resolution of $8.21 \times 10^6 M_{\odot}$. The initial conditions are set up at $z = 65$ using the Zel'dovich approximation [118].

Particularly characteristic in the evolving mass distribution of the Cosmogrid simulation is the large central under-density, surrounded by a range of smaller voids near its outer edge. In combination with its extremely high spatial resolution and state-of-the-art dynamic range, this renders the Cosmogrid simulation uniquely suited as a testbed for a case study of the internal structure of voids. It was precisely this circumstance that formed the rationale behind its exploitation in a previous study of the formation of dark halos along tenuous void filaments [119].

We use two realizations of this simulation. The first is defined on the entire domain, with a box size of $128 \times 128 \times 128$. This dataset, referred to as *CosmogridA*, captures the void features described in the paragraph above. The second dataset we use, referred to as *CosmogridB*, is a subset of the particle distribution of the simulation. The region chosen is characterized by a wide density range, multiple

structural morphologies and low visual confusion. The particle distribution is used as input for computing a tetrahedral domain using a periodic triangulation and the density field is computed using the DTFE methodology [97].

subhaloes dataset. For presenting the results of our volume enhancement routine, we use a large scale simulation in a box of $300h^{-1}\text{Mpc}$, that uses 512^3 particles [60, 120]. The particles have a mass resolution of $4.43 \times 10^9 M_\odot$, and the initial conditions are set up at $z = 60$. For this experiment, we use the dark matter haloes instead of the particles. The halos consist of dark matter particles that clump together to form gravitationally bound, often virialized objects. The mean halo mass at $z = 0$ is $1.1 \times 10^{12} M_\odot$, with a halo mass range of $4.9 \times 10^{11} - 7.9 \times 10^{14} M_\odot$.

The dark matter haloes fairly accurately trace the patterns of the underlying mass distribution. We detect the haloes using one of the standard halo finding algorithms SUBFIND [121]. The first basic step involves the friends-of-friends method for finding halos. Particles are assigned to a group when they are within some linking length of any other particle in the group. This step is augmented by a sophisticated sequence of criteria for identifying gravitationally bound subclumps within the halo, ultimately producing a sample of halos and their subhalos in the simulated dark matter distribution.

5.6 Experiments

In this section, we demonstrate and discuss the salient features and potential applications of Felix. First, we evaluate the filaments extracted using Felix and compare with those extracted using MMF, SpineWeb, and DisPerSE using the Voronoi kinematic datasets. Next, we present a visual exploration of the filaments in the Cosmogrid simulation. Lastly, we present results from applying the structure

based volume enhancement algorithm (see Section 5.4.3). The methods described above were implemented and tested on a computer system with an Intel Xeon(R) CPU, running at 2.0 GHz, with 8 GB of RAM. We begin by describing briefly the structure finders to which we compare Felix.

DisPerSE. DisPerSE [9, 10] is a closely related structure finder that also uses the Morse-Smale complex of the logarithm of the density field. It simplifies the Morse-Smale complex using Topological Persistence. The function values are normalized by the rms of the density field fluctuation with respect to the mean, and the significance level for simplification is quoted in this unit. Felix is closely related to DisPerSE as both use the Morse-Smale complex of the log-density field and involve feature extraction from it. DisPerSE defines significant features as only those that remain unsimplified using the user defined significance threshold. It ignores the density range characteristics of the extracted features. A significant consequence is that filaments within void-like regions and cluster like regions are ignored/simplified away. If they are retained, then the mixing of features causes visual clutter. Furthermore, the significance parameter selection is a fixed constant and visual interaction plays no role in its selection. In contrast, given the ubiquity of filaments in various density regimes, Felix allows for density ranged based probes into filaments, within clusters and voids. Furthermore, the visual interactive aspect allows for user engagement in parameter selection, which is crucial for the set of features identified. Another difference is that Felix uses simplification only for noise removal and not feature identification.

SpineWeb. Spineweb [94] is also a technique based on Morse theory and exploiting the singularity structure of the density field. It exploits the fact that the watershed transform naturally outlines the basins around the local minima of the density field and directly defines the topological structure of the field, and as such forms a direct and practical tool for its analysis in terms of Morse theory.

In this sense, SpineWeb is a direct development of the Watershed Void Finder WVF [93], which introduced the definition of voids in terms of watershed basins.

As a preprocessing step, SpineWeb uses DTFE [97, 98] to compute the density field on a regular grid from the given particle distribution. SpineWeb subsequently identifies the spine of the web as the regions that are excluded from the watershed basins. A given location in this spine is classified as belonging to a filament if it has three distinct void regions in its neighborhood. A region is classified as belonging to a wall if it has two and no more distinct void regions in its neighborhood.

Nexus/MMF. The Nexus/MMF Multiscale Morphology filter [78, 80] identifies the morphology of regions on the basis of the ratio of the eigenvalues of the Hessian of the density field (MMF, [78]) or of fields that have a physical relation to the evolution and formation of the weblike structure in the cosmic mass distribution. The exact conditions for the classification as cluster, filament, wall and void are give by Aragón-Calvo *et al.* [78] and Cautun *et al.* [80].

A key aspect of Nexus/MMF is its explicit multi-scale nature. The density field is translated into a four-dimensional scale space map by convolving the density field with a Gaussian filter over a large range of different scales. The morphological identity of a give location is determined on the basis of a set well-defined morphological filters that compare the corresponding Hessian signature over a stack of convolved images in scale space.

As a result, we obtain a “scale-free” morphological assessment, in the sense that each region gets a unique classification tag based on a criterion that determines the most dominating morphology across the chosen range of scales.

5.6.1 Filaments in the Voronoi model: a comparison

Here we present an analysis of the filamentary structures extracted using Felix, and compare the results with the techniques detailed above. The comparison study

concerns the analysis results obtained for the set of heuristic Voronoi evolution models described above. Since they are input parameters, in these models the classification of galaxies as void, wall, filament, and cluster are known a-priori. Following the application of one of the detection techniques we may then examine the validity and authenticity of the extracted structures by direct comparison with the true identity of a galaxy.

For the comparison study, we define two measures. One quantifies the true detection rate of a method, the other the false identifications. We classify all galaxies within a distance d from the extracted structures to be filament and cluster particles and the others to be void and wall galaxies. For a given set of structures extracted from a given dataset and a distance d , the true positive classification rate TP_d is defined as

$$TP_d = \frac{\# \text{ filament and cluster galaxies correctly classified}}{\# \text{ filament and cluster galaxies}}.$$

Similarly, the false positive classification rate FP_d is defined as

$$FP_d = \frac{\# \text{ filament and cluster galaxies incorrectly classified}}{\# \text{ filament and cluster galaxies}}.$$

A large separation between these two measures indicates good discriminatory power of the classifier, and thus the proximity of relevant galaxies to the extracted structures.

As we discuss in more detail below, the Felix's true and false detection rates are comparable, and in some situations better, than those obtained by DisPerSE, SpineWeb, and Nexus/MMF.

Felix Figure 5.5 shows the extracted filaments for the Voronoi kinematic datasets A , B , and C using Felix. The input density range parameters for Algorithm [SELECT2SADDLES](#) are selected interactively, using the visualization information of

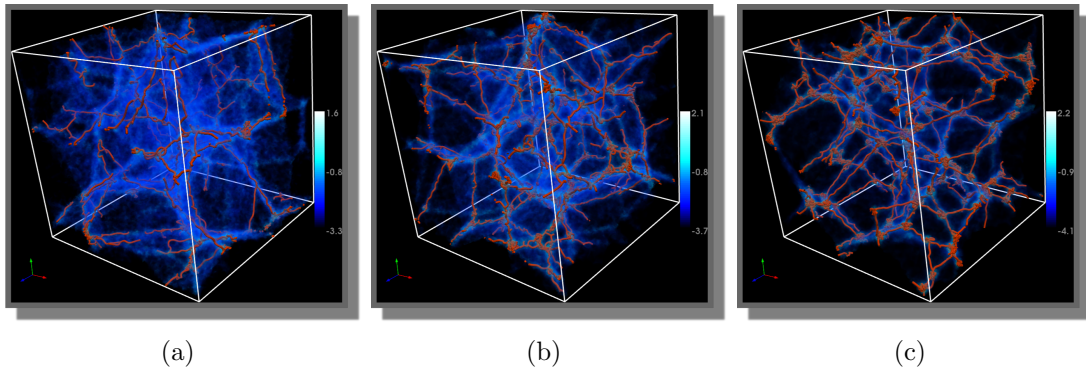


Figure 5.5: Filamentary structures extracted from datasets A, B, and C of the Voronoi evolution time-series using Felix with parameters for Algorithm `SELECT2SADDLES` set as follows: (a) $[S_b, S_e] = [10^{-1.6}, \infty]$ and $[M_b, M_e] = [10^{-0.1}, \infty]$, (b,c) $[S_b, S_e] = [10^{-1.6}, \infty]$ and $[M_b, M_e] = [10^0, \infty]$. Filaments are shown as orange tubes along with a volume rendering of the log-density field. The dense knot like structures show filaments within cluster-like regions.

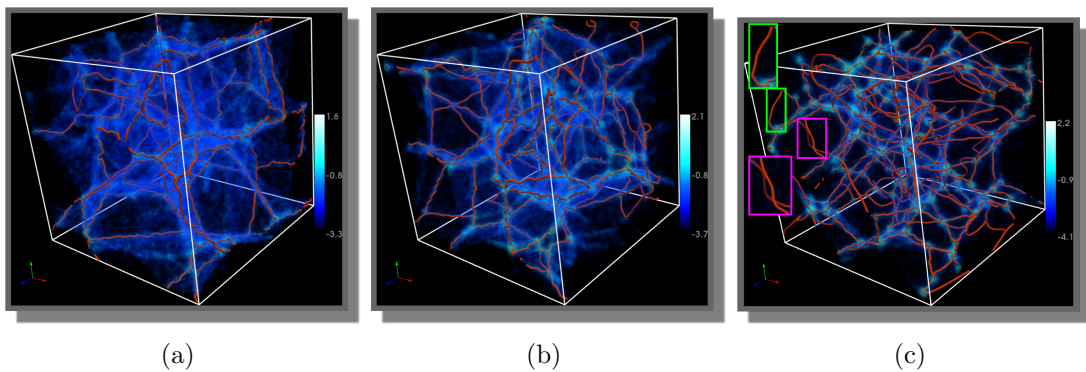


Figure 5.6: Filamentary structures extracted from datasets A, B and C of the Voronoi evolution time-series using DisPerSE with significance level of 5σ . Filaments are shown as orange tubes along with a volume rendering of the log-density field. The inset pictures show identified filaments that are within wall-like and void-like regions of the Voronoi kinematic datasets.

the procedure. Each update is accomplished within 2-3 seconds. This enables interactive visual feedback so that parameters may be adjusted further in subsequent iterative steps. For the given rendering and extraction, the parameter values are $[\ln(S_b), \ln(S_e)] = [-1.6, \infty]$ and $[\ln(M_b), \ln(M_e)] = [-0.1, \infty]$ for dataset *A* and $[\ln(S_b), \ln(S_e)] = [-1.6, \infty]$ and $[\ln(M_b), \ln(M_e)] = [1, \infty]$ for datasets *B* and *C*.

Comparison A: Felix and DisPerSE In this experiment, we demonstrate the consequences of not correlating the density characteristics for filament extraction. Specifically, we demonstrate that the filaments extracted using Felix are more spatially proximal to filament and cluster particles in the Voronoi Kinematic simulation. Furthermore, we show that tuning the significance parameter is not a sufficient mechanism to extract the desired filaments in this dataset. In the next experiment, we demonstrate the exploration of filaments within high-density cluster like regions and low-density void like regions. Such a delineation, coupled with the visual exploration process, is not possible using DisPerSE.

The recovery and failure rates for the Voronoi kinematic models *A*, *B*, and *C* are shown in Figure 5.7. The TP_d and FP_d of the Felix filaments are plotted in the top-left panel, the ones for DisPerSE can be found in the top right-hand panel (1σ significance threshold), bottom left-hand panel (3σ significance threshold) and bottom-right panel (5σ significance threshold).

Felix shows good recovery rates for all datasets, particularly around $d = 3h^{-1}\text{Mpc}$. For the least evolved configuration *A*, and for locations where structures are least distinct, Felix still obtains moderately good recovery rates. The results for DisPerSE with simplification thresholds 1σ and 3σ are comparable: at short distances the true detection rate is slightly lower than that of Felix, while at larger distances it performs marginally better. The situation is slightly different in the case of DisPerSE with a 5σ simplification threshold. In the case of

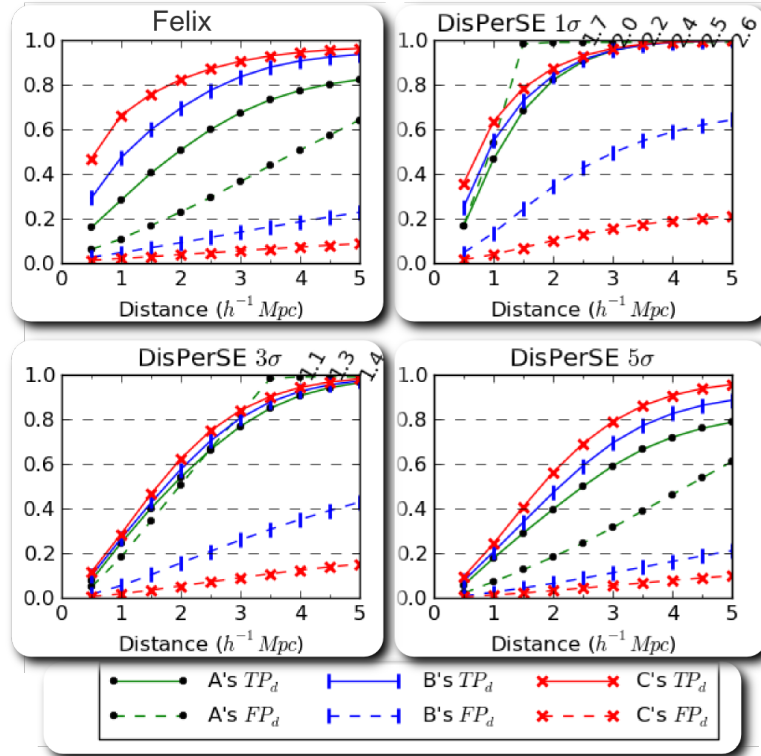


Figure 5.7: Classification recovery rates for Voronoi kinematic datasets A, B, and C. Top left: Felix; top right: DisPerSE, 1σ significance level; bottom left: DisPerSE, 3σ significance level, and (bottom right) DisPerSE with a 5σ significance level. False positive rates greater than 1.0 are clipped and respective values are shown.

the more strongly evolved B and C datasets, the DisPerSE and Felix results have similar false detection rates FP_d , while the true detection rates TP_d of Felix are consistently higher.

For the lesser evolved datasets A and B , the false detection rates FP_d for DisPerSE quickly increase to rather large values. For simplification thresholds of 1σ and 3σ it even surpasses values of unity. This may indicate that in certain circumstances an automatic detection of filaments from the Morse-Smale complex runs the risk of over-determining the population of filaments, even after considerable simplification. While the problem is not so acute in the most evolved stage C , where the morphologies are well separated, direct simplification strategies may not always succeed in properly classifying all filament, wall and void regions in the more moderately evolved stages A and B .

Figure 5.6 shows the filaments in the three Voronoi models detected by DisPerSE, with a simplification threshold of 5σ . In comparison with the structures in Figure 5.5, the knot like structures present in clusters are absent. This leads to the cluster particles being far away from the filament end points, and thus the reduced TP_d rates of DisPerSE. In contrast, Felix's ranged query allows us to retain only the filaments in cluster like regions and those that connect these cluster like regions, leading to better TP_d rates. Also, many filaments found by DisPerSE are within the wall-like and void-like regions of the Voronoi Kinematic simulation. Some examples are highlighted using insets in Figure 5.6. Again these are filtered out by Felix's ranged query, which is not directly possible in DisPerSE. The inclusion of such structures in DisPerSE leads to its higher FP_d rates.

These findings suggest that a structure identification strategy based on a direct simplification procedure of the Morse-Smale complex should be applied with care to the density regimes being studied. Specifically, the superior classification rate profiles confirm that, using Felix, we can easily extract filaments that are

Dataset		TP_d	FP_d
A	<i>Felix</i> ($d = 3h^{-1}\text{Mpc}$)	0.67	0.36
A	<i>DisPerSE</i> ($5\sigma, d = 3h^{-1}\text{Mpc}$)	0.59	0.31
B	<i>Felix</i> ($d = 3h^{-1}\text{Mpc}$)	0.84	0.13
B	<i>DisPerSE</i> ($5\sigma, d = 3h^{-1}\text{Mpc}$)	0.69	0.11
	<i>Nexus/MMF</i>	0.85	0.13
C	<i>Felix</i> ($d = 3h^{-1}\text{Mpc}$)	0.90	0.05
C	<i>DisPerSE</i> ($5\sigma, d = 3h^{-1}\text{Mpc}$)	0.78	0.05
	<i>SpineWeb</i>	0.87	0.10

Table 5.2: Recovery rates of galaxies within $3h^{-1}\text{Mpc}$ of structures extracted using Felix compared with Nexus/MMF, SpineWeb, and DisPerSE (5σ significance level).

spatially more proximal to the cluster and filament particles in the Voronoi Kinematic datasets. This issue is not easily addressed by the significance threshold of DisPerSE. In contrast, Felix provides an intuitive density based handle to extract the desired features. Also, the possibility of having FP_d value larger than 1 in extreme situations is indicative of over-detection of filaments. This is potentially cumbersome for the analysis of genuine cosmological simulations and observational surveys. In more complex realistic circumstances, cosmic structure involves features over a wide range of densities and scales and structural morphologies that are not as well separated as in the simpler Voronoi models.

Comparison B: Felix and SpineWeb For the comparison of the Felix and SpineWeb [103], we use a Voronoi Evolution model realization that is comparable to the advanced state of dataset *C*. We use the test result reported by Aragon-Calvo *et al.* [103] with respect to the model that had a similar percentage of particles in the four morphological features. In these model realizations, the clusters, filaments and walls had a Gaussian density profile with a scale of $R_g = 1h^{-1}\text{Mpc}$. The spine

has an effective width $d = 2R_g$ as the identified structures are thickened by 1 voxel, with a size of $R_g = 1h^{-1}\text{Mpc}$.

For this configuration, Aragon-Calvo *et al.* [103] report true and false detection rates of $TP_d = 0.87$ and $FP_d = 0.10$ (see Table 5.2). Felix attains the same recovery rate of $TP_d = 0.87$ at a smaller distance $d = 3h^{-1}\text{Mpc} = 1.5R_g$. For the same configuration, the failure rate parameter, $FP_d = 0.05$, is comparable to that reported for SpineWeb. By comparison, at $d = 2R_g$, for Felix the recovery rates are $TP_d = 0.93$ and $FP_d = 0.07$.

In summary, these results appear to suggest that Felix performs as well as SpineWeb.

Comparison C: Felix and Nexus/MMF Since Nexus/MMF concerns a sophisticated formalism based on a scale space analysis, the parameters of detection do not have a direct correspondence with topology based techniques like DisPerSE, SpineWeb, and Felix.

We use dataset B for a comparison with Nexus/MMF. This dataset is similar to the least evolved dataset used in the evaluation of the MMF [78], the original density field based Nexus/MMF implementation. For similar values of the detection rate TP_d , both Felix and MMF have identical failure rates FP_d . This indicates that both procedures have a comparable detection behavior.

5.6.2 Filament Exploration

In this section we discuss the application of Felix to explore different classes of filaments from cosmological simulations. The ability to filter filaments on a combination of morphological and density properties is helpful in situations where we wish to focus on, for example, the properties of galaxies residing in filaments in low-density void regions or in the high-density outskirts of clusters. This ability

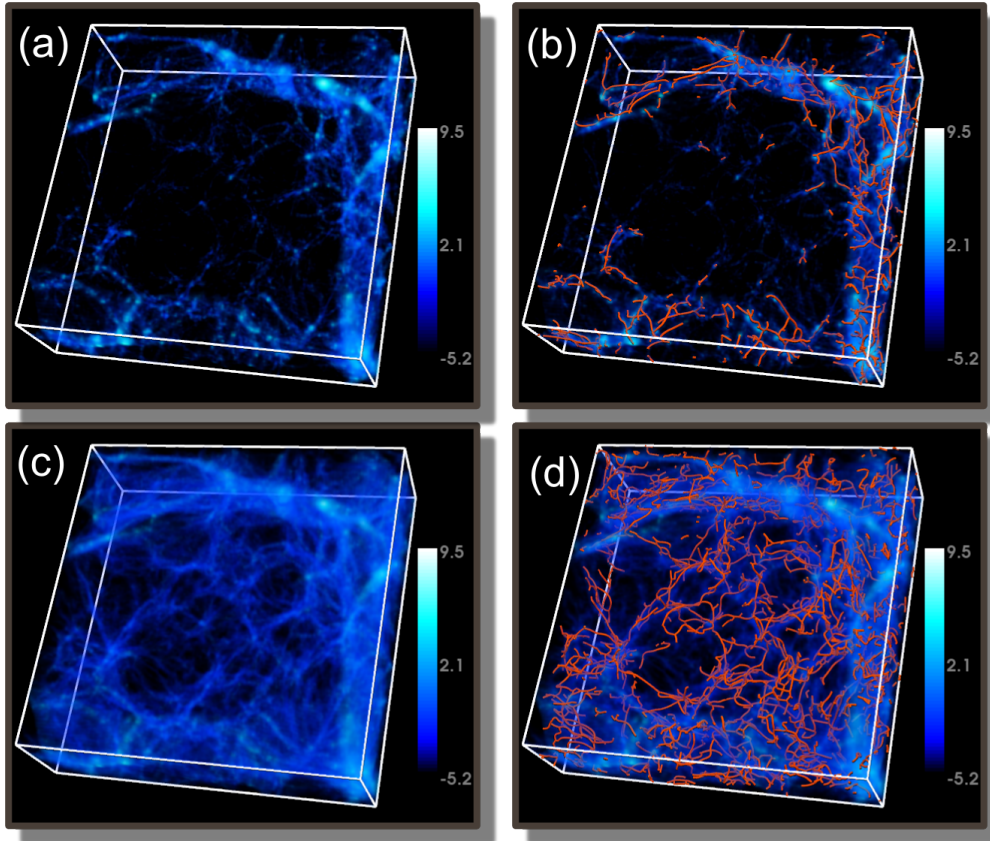


Figure 5.8: Exploring filaments in high-density cluster-like environments and low-density void like regions in the *CosmogridA* dataset. The selected region, comprising of a 3D region of z -slices from 69 through 105, contains filamentary structures in both types of environments. Volume rendering of the density field with opacity adjusted to highlight filaments in (a) high density cluster-like regions, and (c) low density void-like regions. (b) Filaments within high density cluster-like regions extracted with parameters $[S_b, S_e] = [10^0, 10^{9.6}]$ and $[M_b, M_e] = [10^{2.3}, 10^{9.6}]$. (d) Filaments within low density void-like regions extracted with parameters $[S_b, S_e] = [10^{-2.5}, 10^{0.5}]$ and $[M_b, M_e] = [10^{0.5}, 10^{3.5}]$. For both sets of filaments, the value of T_s was set to 0.05.

of Felix to identify a specified population of intravoids or cluster inflow channels provides us with a microscopic instrument that allows a detailed and systematic exploration of the fine structure in the hierarchy of cosmic structure. Felix is able to zoom in on such regions and delineate its detailed infrastructure.

As the criteria for the identification of filaments and other web-like features still differ substantially between the various available techniques, the visual interaction aspect of Felix is a major practical asset in obtaining a proper user-defined selection of filaments. To this end, we may also point out that available automatic detection techniques may produce significant spurious results, which may substantially influence the results of targeted studies as the one illustrated here. A telling example of this has been discussed in the previous section.

CosmogridA: Figures 5.8a and 5.8c present volume renderings of a 3d region of the *CosmogridA* dataset ranging from z-coordinates 69 to 105. The bounding box of the dataset is $128 \times 128 \times 128$. This region was selected as it contains a large void like region surrounded by a large number of high density region. The transfer function opacities have been adjusted to highlight the filament like structures in cluster-like and void-like regions respectively. The hierarchical Morse-Smale complex computation and the filament selection are executed on the entire dataset. The resulting filaments and volume rendering are clipped to the above mentioned region of interest. The filament selection parameters of Algorithm [SELECT2SADDLES](#) are adjusted interactively, via a visualization step. A video demonstrating the visual exploration process in this experiment is available at <https://youtu.be/8TRtX937Xjk>. Each selection query takes approximately 1 second to process (Intel Xeon processor, 2 GHz, 8 GB ram). The subsequent extraction of filament geometry depends on the number of selected 2-saddles. This takes approximately 4 seconds. Thus, the query framework may be used to interactively change parameters and visually correlate the set of extracted features

with the underlying density distribution.

The filament selections obtained following the application of the interactive procedure are illustrated Figures 5.8b and 5.8d. Each shows filaments in a different environmental density regime. Figure 5.8b shows filaments that exist near and within the high density cluster like regions. These are the filaments that form the spine of the cosmic web. Figure 5.8d shows filaments within void-like regions. The combination of density criteria and interactive visualization enables us to zoom in on this system of intravoids filaments. They are the faint residuals of the smaller-scale filaments that constituted the spine of the cosmic web at earlier cosmic epochs, and as such represent a direct manifestation of the hierarchical buildup of cosmic structure. At the current epoch, the intravoids filaments appear to define a different pattern than the prominent filamentary bridges between clusters of galaxies. As a result of the tidal influence of surrounding large-scale mass concentrations they are conspicuously aligned along a direction correlated with the main axis of the embedding void.

CosmogridB: The filament selections obtained following the application of the interactive procedure using the *CosmogridB* dataset are illustrated in three panels. Each shows filaments in a different environmental density regime. Figure 5.9(b) shows filaments that exist near and within the high density cluster like regions, Figure 5.9(c) shows filaments within void-like regions and Figure 5.9(d) shows filaments that stretch all the way from cluster to void like regions. The latter are the filaments that form the spine of the cosmic web.

5.6.3 Volume enhancement

In this section, we illustrate the use of the volumetric enhancement routine in presenting a three-dimensional impression of the web-like network. The routine suppresses the depiction of components far removed from filaments of interest

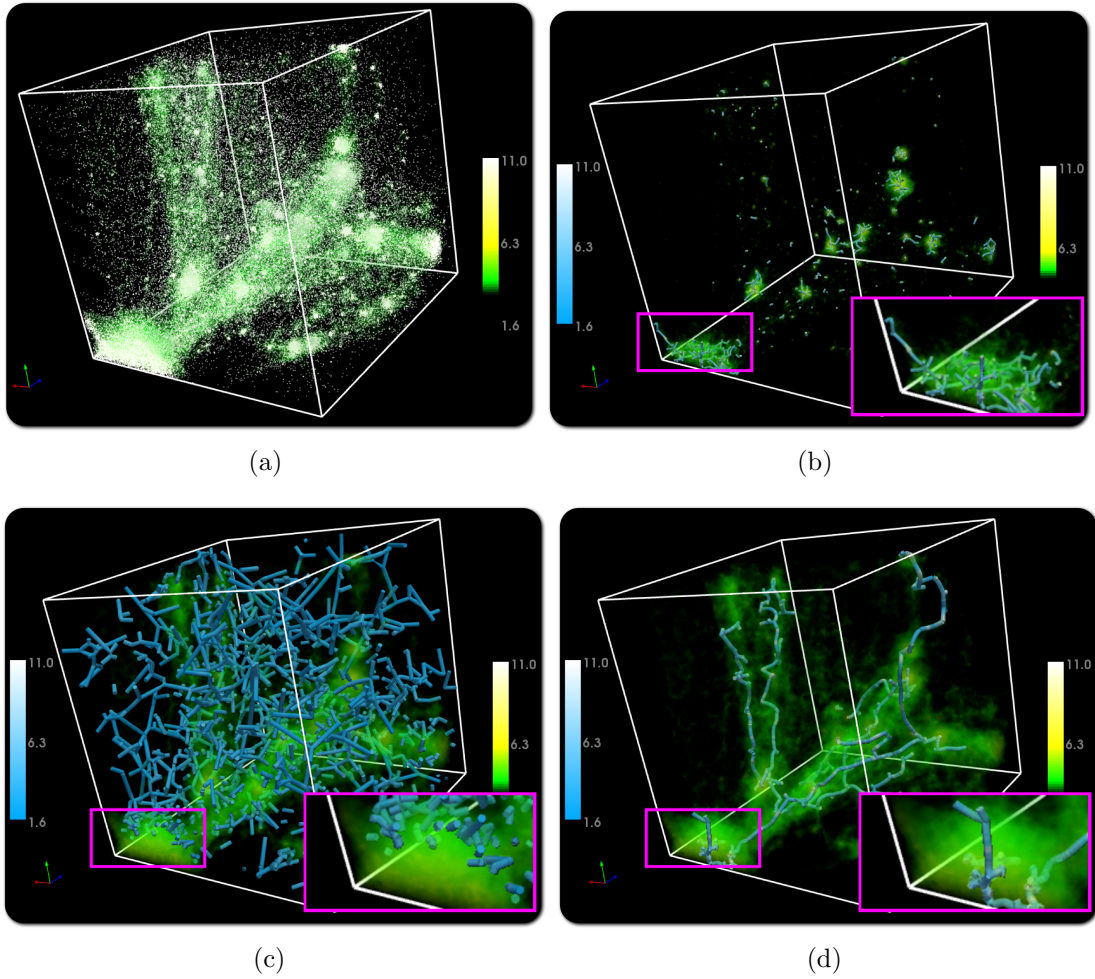


Figure 5.9: Three classes of filaments extracted from the *CosmogridB* dataset. (a) Particle distribution shown along with a volume rendering of the DTFE density. (b) Filaments within cluster like regions extracted with parameters $[S_b, S_e] = [10^5, 10^8]$ and $[M_b, M_e] = [10^8, 10^{12}]$. The highlighted region shows the retention of intricate topological structures within a large cluster region. (c) Filaments within void like regions extracted with parameters $[S_b, S_e] = [10^0, 10^5]$ and $[M_b, M_e] = [10^0, 10^5]$. Shown in the inset is a cluster like region within which filamentary structures are filtered away as desired by the query to the framework. (d) Filaments that stretch from cluster like regions all the way down to void like regions using parameters $[S_b, S_e] = [10^3, 10^{11}]$ and $[M_b, M_e] = [10^9, 10^{11}]$. The highlighted region shows a single filament passing through the large cluster like region with intricate topological details filtered out by the framework.

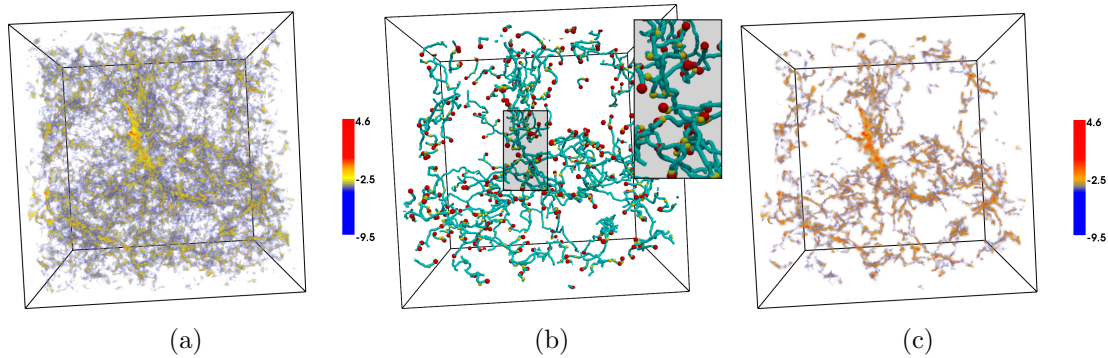


Figure 5.10: Extracting filamentary features from the $300h^{-1}\text{Mpc}$ LCDM dataset and enhancing the volume rendering. (a) Volume rendering of the density field. (b) Filaments selected by Algorithm `SELECT2SADDLES` with $[S_b, S_e] = [e^{-4.0}, \infty]$ and $[M_b, M_e] = [e^{1.0}, \infty]$ retains the filaments in high density regions. The intricate filamentary structure within a predominantly high density region is highlighted in the inset. (c) An enhancement produced by the volume rendering procedure applied to the selected filaments.

(see Section 5.6.3). One potential application of the enhancement algorithm is as a noise removal tool that is able to render the density field in the vicinity of structures of interest. This will help substantially towards elimination of the visual confusion and of possibly less interesting structures in the process.

For the experiment shown in Figure 5.10, we use the the *subhaloes* dataset described in Section 5.5.2. This large-scale simulation provides an appropriate setting to illustrate the ability of the volume enhancement method. The dataset contains a multitude of structures spanning a wide range of density values that usually evoke a significant visual confusion. This may be clearly appreciated from the lefthand Figure 5.10(a). The subsequent filament selection and volume enhancement are shown in Figure 5.10(b). It provides a better appreciation of the spine of the filamentary network.

5.7 Conclusions

We have presented a Topology based Framework, named Felix, to probe filament structure in the large scale universe. The framework is particularly designed to probe filamentary structures in different density regimes, and optimally preserve structural detail in regimes of interest. While other cosmic structure analysis tools do not include a facility to select web-like features according to tailor-made aspects and characteristics, this is precisely the mission of the Felix procedure. We directed Felix towards a case study of the the filamentary infrastructure and architecture of cosmic voids and demonstrated that it successfully extracts the network of tenuous filaments pervading their interior [110, 111, 112, 113] .

In an accompanying study, we plan to exploit the Felix facility to systematically study the physical characteristics of the extracted samples of intra-void filaments. This also involves their halo and subhalo population, their gas content, and the relation of these with the embedding voids and surrounding large-scale mass distribution. This will be of key importance towards understanding the formation and evolution of void galaxies [122, 123, 124] and specifically that of of the issue of the missing dwarf galaxies [102]. In addition, following the recognition that void architecture represents a potentially sensitive probe of dark energy and dark matter and a keen test of modified gravity theories [56, 57, 58, 59, 60, 61, 62, 63], the filament samples extracted by Felix will be subjected to a systematic study of their dependence on cosmological parameters.

As an immediate extension, we plan use Felix with other scalar fields such as the tidal force field. Another possible direction is the visualization and analysis of the hierarchy of voids, walls, and filaments in cosmological datasets. Interactive methods for the visualization of these intricate structural networks remains a challenging and largely unexplored problem of major significance.

Chapter 6

Molecular Surface Alignments

Aligning similar molecular structures is an important step in the process of biomolecular structure and function analysis. Often, as in the case of Electron Microscopy (EM) data, fully resolved molecular structure is not available. The data is however geometric in nature, in the form of electron density maps as well as simple derived forms such as isosurfaces. Molecular surfaces often contain several significant protrusions that are manifestations of groups of atoms, such as aromatic rings and amino groups, close to the exterior surface. Such significant protrusions may be leveraged to determine alignments between molecules, where surface patches corresponding to these protrusions are aligned. In this chapter, we present a method that aligns molecular surfaces based on such significant protrusions on the molecular surface¹. A key step of the method involves identification and segmentation of the molecular surface. For this step, the 2D Morse-Smale complex is employed to segment significant protrusions on the molecular surface using the mean surface curvature as a scalar field on the surface. We implement our method to develop MS3ALIGN: the multi-scale Morse-Smale molecular surface

¹This work was carried out in collaboration with Sonali Patil, who is with the College of Technology at Purdue University, Amrisha Bhosle and Nagasuma Chandra, who are with the Department of Biotechnology at the Indian Institute of Science.

aligner.

6.1 Related Work

Three dimensional solved crystal structures of proteins provide valuable insights regarding the function of the protein as the precise position of all functionally and structurally important residues is known. Since structure determines function, the function of an unknown protein may be determined by comparing its structure to structures of proteins whose functions are already known. Tools such as MUSTANG and DALI [125, 126], which are widely used to compare protein structures, use three-dimensional co-ordinates of atoms in the protein structures as inputs and report structural dissimilarities in terms of an RMS distance between their aligned coordinates.

Proteins that function as enzymes and transporters contain a pocket or the binding site in the structure that accommodates the substrate and cargo small molecules respectively. The arrangement of amino acid residues in the binding site often determines the specificity of a small molecule ligand towards a receptor protein. It is intuitive therefore that structurally similar ligands will bind to pockets that are structurally similar. Therefore, pocket and ligand alignments could potentially provide insights into protein function

There exists a large number of tools that determine alignments. Common approaches for determining alignments include aligning residues [125, 127, 128, 129], secondary structures [130, 131], or molecular surfaces [132, 27, 133, 134, 135]. Surface based methods offer advantages in the study of protein-ligand and protein-protein interactions as they determine alignments based on the molecular surfaces at the site of interaction. The tool MOLLOC [132] and the image based method by Merelli *et al.* [134] compute alignments by comparing images of the surface from

oriented points to determine corresponding points on surfaces. Since the number of oriented points is large, these tools are computationally intensive and require fine tuning for efficient execution. PROBIS [133] aligns proteins/nucleic acids by aligning the centroids of sub-residue level functional units (aliphatic/ aromatic rings, hydrogen donor/acceptors etc) that lie close to the molecular surface. The tool SURFCOMP aligns small ligand surfaces using either surface curvature or electrostatic potential on the molecular surfaces. SURFCOMP identifies local maxima of these scalar fields as features and attempts to align them across pairs of surfaces. The tool PBSALIGN aligns protein-protein interaction surfaces using an approach similar to SURFCOMP with a few key differences. It defines feature points as surface points that are closest to C_α atoms and computes a feature vector comprising of principal curvatures and statistics of electrostatic potential and hydrophobicity near each feature point.

6.2 Contributions

We present MS3ALIGN, a Multi-Scale, Morse-Smale, Molecular Surface aligner. The tool MS3ALIGN begins by computing mean curvature at all points on the surface and then segmenting significant protrusions. Segmentation is performed by a topological analysis of the surface mean curvature using the Morse-Smale complex. Correspondences between pairs of protrusions on either surface are then established using a two-step procedure, computation of a multi-scale curvature descriptor followed by neighbor identification in the descriptor space. These correspondences are then grouped together into maximal sets. Each maximal set is used to compute a rigid body transformation that aligns the first surface to the second surface. These alignments are evaluated and ordered using a distance measure that is based on the RMS distance between surfaces and the corresponding

area fraction.

The key benefit of the method is that it is agnostic to how the molecular surface was generated. We show experiments where surfaces generated from atom location data in the PDB as well as density maps in the cryo-Electron-Microscopy data bank are aligned. Thus, the method is not dependent on a particular source of data. Rather, the only input to the method is the surface data. This is a key advantage over existing methods such as SURFCOMP, PBSALIGN, and PYMOL, which rely on protein sequence data and other derived scalar values such as the electro-static potential, which are often not directly available/computable. Another benefit of our method is that the implementation allows for visual analysis of the alignment, which leads to direct comprehension of the results. A third benefit is that our method is purely based on the local geometric structure. Thus, it is applicable in scenarios where local geometric context is preserved.

Our method may be viewed as an improvement of SURFCOMP [27], and hence we perform a detailed comparison with their results. We validate MS3ALIGN using surface representations of ligands and their site of interaction. Specifically, we use the POCKETMATCH [136] tool to quantify structural variation between various interaction sites and compare their alignments obtained using MS3ALIGN. In this experiment, we also demonstrate the benefit of visual analysis where we visually validate results.

6.3 Background

We briefly review the additional background material relevant for this application. In particular, we discuss the mathematical notion of curvature.

A *regular curve* is defined as a twice differentiable function $l(t) : \mathbb{I} \rightarrow \mathbb{R}^3$ from the unit interval to 3D space where the magnitude of the tangent equals one,

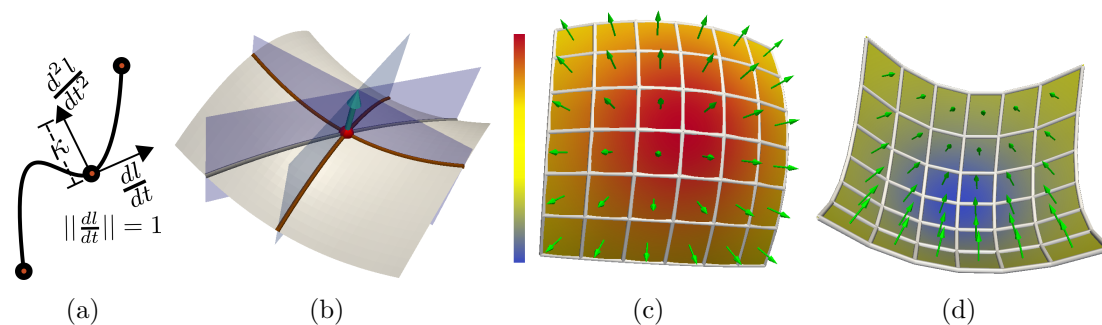


Figure 6.1: (a) Curvature κ of a regular curve at a point is defined as the magnitude of the second differential at that point. (b) The principal curvatures of a point x on a smooth surface is defined as the maximal κ_1 and minimal κ_2 curvature of regular curves formed by intersecting planes rotated about the surface normal n_x with the surface. Mean curvature is defined as $(\kappa_1 + \kappa_2)/2$. Mean curvature is (c) high at convex regions (d) and low at concave regions. Green arrows depict surface normals at respective surface points.

$\|\frac{dl}{dt}\| = 1$ [137]. The *curvature* at a point t is defined as the magnitude of the second differential at t , $\kappa(t) = \|\frac{d^2l}{dt^2}\|$.

A family of regular curves through a point x on a surface \mathbb{S} is defined by a collection of planes that contain the surface normal n_x at the point x and the point itself (see Figure 6.1(b)). Each of these curves has an associated curvature κ . For curves on surfaces, one associates a sign with the curvatures given by the dot product of the surface normal and the second differential of the regular curve. The curves with maximum and minimum curvature (with sign) are referred to as the *principal curvatures* at x and are denoted by κ_1 and κ_2 respectively. The *mean curvature* at a point x , $H(x)$ is the mean of κ_1 and κ_2 i.e. $H(x) = (\kappa_1 + \kappa_2)/2$. Convex surfaces have positive κ_1 and κ_2 and therefore have positive mean curvature, whereas concave surfaces have negative mean curvature (see Figures 6.1(c) and 6.1(d)).

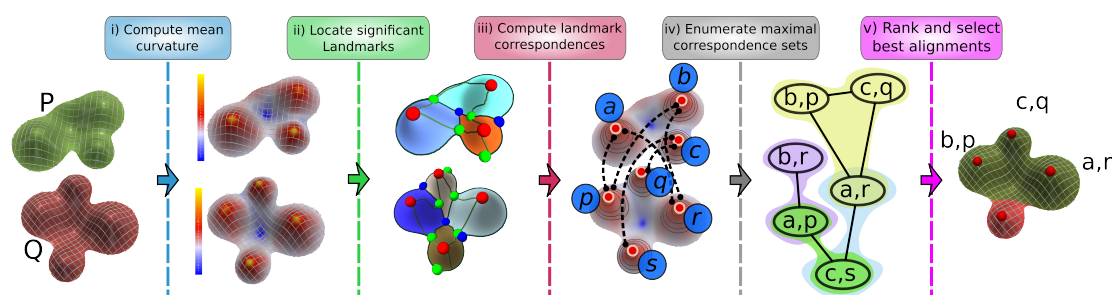


Figure 6.2: The surface alignment pipeline comprises of five stages. P and Q are input surfaces represented as triangle meshes. *i)* Mean curvature is computed for both surfaces. *ii)* Significant landmarks are extracted by first computing the Morse-Smale complex of the curvature field and then simplifying the Morse-Smale complex using topological persistence. The maxima that survive simplification (red spheres) are used as landmark points. *iii)* Correspondences between landmark points on either surface is established by comparing the multi-scale curvature vectors. *iv)* A graph is constructed where each landmark correspondence is considered as a node, and edges are placed between nodes if they satisfy inequalities (6.1) and (6.2). Each maximal clique in this graph generates a maximal correspondence set. *v)* Each maximal correspondence set is evaluated using the measure given by (6.3). Those evaluating to the smallest values are ranked as the best alignments.

6.4 Surface Alignment Pipeline

In this section we describe the design of MS3ALIGN in detail. The tool MS3ALIGN comprises of multiple stages where the primary input is the two surfaces represented as triangle meshes. Figure 6.2 depicts the various stages. In the following sections each stage is explained in detail.

6.4.1 Curvature Computation

The mean curvature at every vertex of the two input surfaces P and Q is computed in the first stage. There exist many algorithms to estimate the mean curvature of triangle meshes. We use the algorithm based on the theory of normal cycles by Cohen-Steiner *et al.* [138]. We choose this approach for two reasons. First, this algorithm guarantees linear convergence to the curvature of a smooth surface, with

a sufficiently well sampled set of points. Second, the definition of the curvature is based on averaging the curvature tensor over a neighborhood of size R_c on the triangle mesh. This allows for a smoother estimate of the mean curvature where meshing artifacts are overcome by the averaging operation. Good selection of the neighborhood size R_c for curvature computation is crucial to avoid isotropy issues introduced by mesh discretization. For computing a smooth curvature estimate at each vertex, we find that the neighborhood size R_c needs to span at least two rings of vertices. The maxima of mean curvature correspond to the protrusions on each surface patch.

6.4.2 Landmark extraction

In the second stage, landmark points, representing significant protrusions of both surfaces P and Q , are identified from the maxima of the mean curvature scalar field. However, several maxima may correspond to near-flat regions. Significant protrusions are identified by a topological analysis using the Morse-Smale complex of the mean curvature field. Segmenting molecular surfaces using the Morse-Smale complex has been reported earlier [139, 140] using the Connolly function [141], which is related to the surface curvature. The Morse-Smale complex of the mean curvature field is first computed [15] and then simplified by iteratively canceling insignificant maxima using the topological cancellation procedure. The significance of a maximum is determined by the notion of *topological persistence* [24] where each maximum is paired with a saddle critical point. The measure of significance of a maximum, referred to as its *persistence*, is the absolute difference in mean curvature value of the maximum and its paired saddle critical point. Thus, maxima are eliminated in increasing order of persistence upto a given threshold, denoted by T_s . The threshold T_s is specified as a fraction of the average of the mean curvature at all local maxima.

6.4.3 Landmark correspondences

In the third stage, correspondences between pairs of landmark points on either surface P and Q is established. This is done by analysis of the curvature at multiple scales. The mean curvature computed using a neighborhood size R_c gives an estimate of curvature at that scale. We compute mean curvature at multiple scales at each landmark point $\mathbf{p} \in P$ and $\mathbf{q} \in Q$. We use 15 uniformly sampled curvature scales from the interval $[R_c, 2R_c]$. Landmark points $\mathbf{p} \in P$ and $\mathbf{q} \in Q$ are declared as a corresponding pair (\mathbf{p}, \mathbf{q}) if the absolute difference between their mean curvatures at every scale is bounded by a threshold T_{ms} . The threshold T_{ms} is specified as a fraction similar to T_s .

6.4.4 Maximal correspondence sets

In the fourth stage, correspondences between landmark points in P and Q are collected into maximal sets of correspondences. A maximal set $\mathbb{C} := \{(\mathbf{p}_1, \mathbf{q}_1), (\mathbf{p}_2, \mathbf{q}_2), \dots, (\mathbf{p}_n, \mathbf{q}_n)\}$ is constructed so that for each pair of correspondences $(\mathbf{p}_i, \mathbf{q}_i)$ and $(\mathbf{p}_j, \mathbf{q}_j)$, the two landmark points \mathbf{p}_i and \mathbf{p}_j in P have relative pairwise geometric properties similar to that of the two landmark points \mathbf{q}_i and \mathbf{q}_j in Q . We use two geometric properties to establish relative pairwise similarity. First, we ensure that the absolute difference between the distances of the two landmarks on either surface is less than a threshold T_{mrd} , referred to as the maximum relative distance threshold. Second, we ensure that the absolute difference in the angles between the surface normals of two landmarks on either surface is less than $\pi/2$. In other words, for a given maximal set $\mathbb{C} := \{(\mathbf{p}_1, \mathbf{q}_1), (\mathbf{p}_2, \mathbf{q}_2), \dots, (\mathbf{p}_n, \mathbf{q}_n)\}$, we ensure that

$$|\|\mathbf{p}_i - \mathbf{p}_j\| - \|\mathbf{q}_i - \mathbf{q}_j\|| < T_{mrd} \quad (6.1)$$

$$|\cos^{-1}(N(\mathbf{p}_i) \cdot N(\mathbf{p}_j)) - \cos^{-1}(N(\mathbf{q}_i) \cdot N(\mathbf{q}_j))| < \pi/2 \quad (6.2)$$

for all $(\mathbf{p}_i, \mathbf{q}_i), (\mathbf{p}_j, \mathbf{q}_j) \in \mathbb{C}$, where $N(\mathbf{p})$ represents the surface normal at point \mathbf{p} . The constructed sets are maximal in the sense that no other correspondence may be added without violating conditions (6.1) and (6.2).

Finding maximal correspondence sets may be recast as the problem of enumerating maximal cliques in graphs. The nodes of the graph are correspondences between landmark points (\mathbf{p}, \mathbf{q}) . Edges exist between pairs of correspondences (\mathbf{p}, \mathbf{q}) and $(\mathbf{p}', \mathbf{q}')$ if they satisfy conditions (6.1) and (6.2) and if $\mathbf{p} \neq \mathbf{p}'$ as well as $\mathbf{q} \neq \mathbf{q}'$. Maximal correspondence sets are found by enumerating maximal cliques from this graph. We use the greedy pivot heuristic modification of the Bron-Kerbosch algorithm [142] by Koch [143, 144] to enumerate maximal cliques in this graph. This modification exhibits near linear complexity in the number of maximal cliques for most graphs.

Indeed, the total number of maximal correspondence sets may be exponential in the number of landmark points. The number of maximal correspondence sets depends on the tolerance for positional uncertainty of the landmark points, captured by T_{mrd} , as well as the number of correspondences, captured by T_{ms} . Choosing a very small T_{ms} and T_{mrd} causes the algorithm to demand near exact matches in the mean curvature as well as the relative positions of the landmarks, whereas higher values allow larger variations.

6.4.5 Surface Alignment

In the final stage, each maximal correspondence set is first used to determine a rigid body transformation (\mathbf{R}, \mathbf{t}) from P to Q via a least squares minimization [145]. Next, for each correspondence set $\mathbb{C} := \{(\mathbf{p}_1, \mathbf{q}_1), (\mathbf{p}_2, \mathbf{q}_2), \dots, (\mathbf{p}_n, \mathbf{q}_n)\}$, we compute a measure $D_{P,Q}(\mathbb{C})$ that quantifies the error of the transformation (\mathbf{R}, \mathbf{t}) ,

given by

$$D'_{P,Q}(\mathbb{C}) := \frac{\sqrt{\frac{\sum_{i=1}^n A_P(\mathbf{p}_i) \|(\mathbf{R}\mathbf{p}_i + \mathbf{t}) - \mathbf{q}_i\|^2}{A_P(\mathbb{C})}}}{A_P(\mathbb{C})/A_P} \quad (6.3)$$

$$D_{P,Q}(\mathbb{C}) := \min\{D'_{P,Q}(\mathbb{C}), D'_{Q,P}(\mathbb{C})\}$$

Here, $A_P(\mathbf{p}_i)$ is the area of the descending manifold of \mathbf{p}_i , $A_P(\mathbb{C})$ is the total area of all landmarks of P in \mathbb{C} i.e. $A_P(\mathbb{C}) := \sum_{i=1}^n A(\mathbf{p}_i)$, and A_P is the area of P . Also, $D_{P,Q}(\mathbb{C})$ is a symmetric version of $D'_{P,Q}(\mathbb{C})$, where $D'_{Q,P}(\mathbb{C})$ inverts the roles of P and Q in $D'_{P,Q}(\mathbb{C})$ with the exception that the transformation (\mathbf{R}, \mathbf{t}) is again applied only to landmarks in P . The numerator of $D'_{P,Q}(\mathbb{C})$ in Equation (6.3) represents a coarse approximation of the RMS distance between the matching portions of both surfaces. The denominator is the fraction of the area of descending manifolds of all landmark points of P in \mathbb{C} with respect to the total area of P . Due to the denominator, correspondence sets having larger area fractions are preferred.

Maximal correspondence sets that result in the least values of this measure are reported along with their transformations. Computing this measure is efficient even with a large numbers of maximal correspondence sets since it only requires landmarks and not all points on the surface. It is possible for some spurious correspondence sets consisting of a few landmarks to align nearly perfectly. Hence, we consider only those correspondence sets so that the area of the corresponding regions of either surface is atleast 15% of its total area.

In general, the choice of Rc and $Tmrd$ are constrained by the scale of the expected size of features. In our experiments, this choice depends on the dataset being studied. In the first experiment (Section 6.5.2), we study alignments by fixing Rc and $Tmrd$ and varying Ts and Tms . We conclude that we are able to compute satisfactory alignments with a value of 0.1 for both parameters. However, in datasets with fewer feature points, it becomes necessary to decrease Ts so that

more feature points are retained and increase Tms to ensure that more correspondences are established. In these cases, it does not adversely impact performance because of the fewer number of feature points in these datasets.

6.5 Experiments

In this section we discuss our evaluation of MS3ALIGN. We begin with a discussion on the molecular surfaces used in our experiments (Section 6.5.1). In the first experiment (Section 6.5.2), we evaluate its performance under conditions of noise, partial overlap, and running times using a random set of 20 proteins from the PDB [2]. In the second experiment (Section 6.5.3), we compare MS3ALIGN with the results discussed in the evaluation of SURFCOMP [27]. As the MOLLOC web-server [132] and the code from Merelli *et al.* [134] is unavailable, we could not compare our results with theirs. In the third experiment (Section 6.5.4), we validate the alignments of binding sites computed by MS3ALIGN comparing them with those generated using PyMol [3]. Here, we quantify the structural variation of binding sites using POCKETMATCH [136]. In the fourth experiment (Section 6.5.5), we use MS3ALIGN to compute alignments of iso-surfaces extracted from cryo-electron microscopy scans.

6.5.1 Molecular surfaces

In our experiments, we use three types of surfaces extracted from proteins/ligands obtained from the PDB [2] which are built upon the van-der-Waal's molecular model. The first type is the *molecular skin surface* (Section 6.5.1) [146]. The second type of surface used is the *ligand surface*, which is representative of the ligand interacting with the protein in a protein-ligand interaction obtained from the PDB [2] (Section 6.5.1). The third kind of surface used is the *pocket surface*

(Section 6.5.1). This is representative of the surface of interaction of the protein with the ligand. Additionally, we generate partially overlapping surfaces by cutting skin surfaces to generate pairs of surfaces with approximately known overlap fractions (Section 6.5.1).

Skin Surface

The molecular skin surface is a mathematically robust surface model that is similar in geometry to the solvent excluded surface. The molecular skin surface of a protein is computed in two steps. First, its atomic locations are augmented with van der Waal radii after adding missing hydrogen atoms using the PDB2PQR tool [147]. Next, the skin surface is extracted using the NANOSHAPER tool [148] using the atomic locations and radii as input.

Ligand Surface

The ligand surface is computed in three steps. First, the CHIMERA [149] program is used to extract the molecular structure of the ligand from the protein's PDB file. Then, together with the protein's PDB file, the molecular structure of the ligand is used to determine atomic positions and radii of the ligand using the PDB2PQR tool. Finally, the ligand position and radii data is used to compute the skin surface of the ligand using NANOSHAPER.

Pocket Surface

The pocket surface is extracted as a subset of the molecular skin surface of the protein. This subset is the part of the surface that belongs to the residues (within 4.5\AA) that interact with the ligand. These residues are referred to as the *pocket residues* or just the *pocket*. A subset of the molecular surface is extracted so that all vertices of the subset are within the Van der Waal's sphere of at least one of the

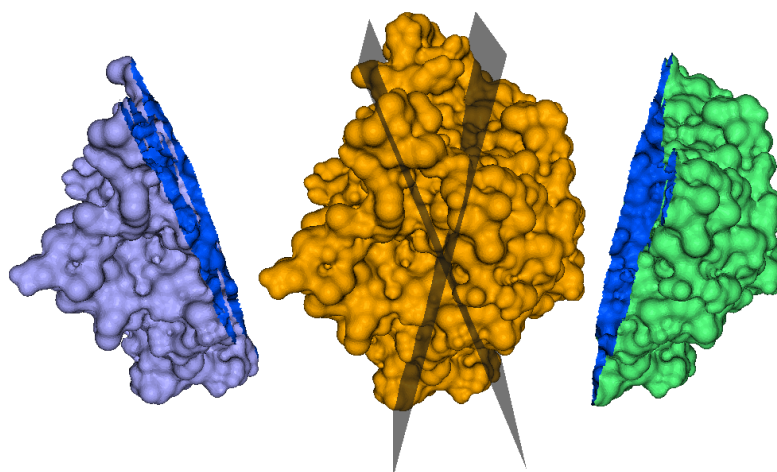


Figure 6.3: Pairs of surfaces which overlap with each other with an approximately known fraction of area of each other are generated by cutting the molecular skin surface using a pair of planes. Here, the skin surface of 4j2m (orange, center) is cut by two planes into two pieces (violet on the left and green on the right) such that 20% of the area of either overlaps with the other.

atoms of the pocket. An additional 0.5\AA is added to the Van der Waal's sphere to account for the possible error introduced when extracting the skin surface, because NANOSHAPER uses a structured grid with edge length 0.5\AA . This subset surface may be disconnected and/or contain holes. This topological noise is repaired using a variant of the dilation-erosion operation applied to triangle meshes [150]. The radius for both steps is set to 1.2\AA , the radius of the hydrogen atom.

Partially overlapping surfaces

In this section, we describe in detail our approach for the generation of overlapping skin surface pairs with approximately known overlap fractions. Each skin surface is split by a pair of planes into two surfaces such that they overlap with each other with an approximately known overlap fraction (see Figure 6.3). Pairs of surfaces are generated for overlap fractions of approximately 20%, 40%, 60%, and 80% of each other. To do this, first, a coordinate system about the centroid of the

molecule is constructed from the principal components of the centroid subtracted positions of the surface’s vertices [151]. The plane containing the first two principal components is rotated about the second component by 0 , $\pi/5$, $2\pi/5$, $3\pi/5$ and $4\pi/5$ radians to obtain five planes, each of which slice the surface into two pieces. The piece on the right of each plain is retained to generate five *partial* surfaces. The first partial surface overlaps with the other four partial surface with approximately 20%, 40%, 60%, and 80% of its area.

6.5.2 Performance analysis

In this experiment, we study three aspects of MS3ALIGN. First, we study its ability to determine correct alignments in the presence of noise. Second, we study its ability to detect alignments in the presence of partial overlaps. Finally, we study the runtime performance of MS3ALIGN. We use twenty structurally different proteins from the PDB [2] having 1500–3000 atoms. We set the parameter $R_c = 3\text{\AA}$. This enables landmarks to be located on protrusions of 2–4 atoms, which is typical of groups such as ammonium, hydroxy, and methyl that are close to the surface. We set the parameter $T_{mrd} = 1\text{\AA}$. Alignments are studied with varying choices of the T_s parameter. The T_{ms} parameter is set to be equal to the T_s parameter.

We now study performance in the presence of noise. Noise is introduced by adding standard Gaussian noise of known variance to all atom locations of the protein. The molecular skin surface of this perturbed molecule is used as the noisy version of the surface. Multiple such noisy versions of the skin surface are generated by adding increasing levels of noise. The level of noise is quantified by computing the RMS distance of all atoms from their original position to their position after adding noise. We compute the RMS distance between the two surfaces by mapping each vertex of the original surface to the closest vertex in the noisy

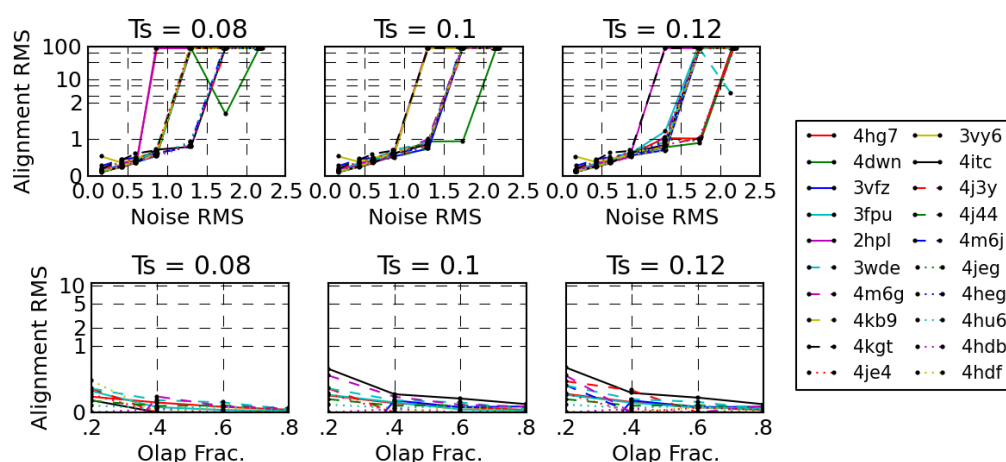


Figure 6.4: (*top-row*) RMS distance between skin surfaces of various molecules after each surface is aligned with noisy versions of itself shown for $T_s = 0.08$, 0.1 and 0.12 respectively. The y-axis is log-scaled beyond 2\AA . (*bottom-row*) RMS distance between a subset of the skin surface and four other surfaces that it partially overlaps with. The y-axis is log-scaled beyond 1\AA .

surface after alignment. We conclude that the alignment is successful if the RMS distance is within 2\AA . The top row of Figure 6.4 shows three graphs for respective simplification threshold T_s values of 0.08, 0.10, and 0.12. With a T_s threshold of 0.1, we observe that surfaces align with RMS distance approximately equal to the RMS distance between the noisy and noiseless surfaces upto 1\AA . We also observe that most alignments fail after the introduced noise causes RMS distance between surfaces to be more than 1\AA . A primary reason for this is the choice of the maximum relative distance threshold $T_{mrd} = 1\text{\AA}$, which specifies the amount acceptable relative movement of the landmark points.

We next study the ability of MS3ALIGN to detect alignments in the presence of partial overlaps. Five partially overlapping surfaces are generated as subsets of the skin surface. The first surface is generated by a cut plane that partitions the skin surface. The remaining surfaces are also generated by rotated cut planes such that the overlap fractions with the first surface is approximately 20%, 40%, 60%, and 80% respectively. In other words, the first and second partial surfaces intersect

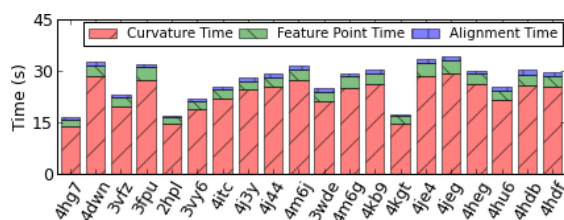


Figure 6.5: Stacked bar graph showing breakup of MS3ALIGN run-times for the surfaces used in the partial overlap study. The time taken for alignment (stages *iii – v*) is significantly lesser than time taken for the first and second stage.

in approximately 20% of the area of each other, the first and third intersect in approximately 40% of the areas of each other, and so on. Since we already know the transformation that aligns the first partial surface with the others, namely the identity transformation, we study the RMS distance from the first partial surface after applying the the alignment transformation determined by MS3ALIGN. The bottom row of Figure 6.4 shows three graphs for respective simplification threshold T_s values of 0.08, 0.10, and 0.12. The alignment RMS distance of the first partial surface is mapped to the y-axis and the overlap fractions with the remaining four partial surfaces is mapped to the x-axis. We find that, in all cases, alignments were successfully determined with RMS distance less than 1\AA .

Next, we present the runtime breakup of the various stages of MS3ALIGN in Figure 6.5. We conducted our experiments on a HP xw7700 workstation with an Intel(R) Xeon(R) CPU E5405 2.00GHz dual quad-core processor and 8 GB of RAM. Figure 6.5 presents the running times for the partial overlap experiment discussed above for a simplification threshold parameter T_s value of 0.1. We note from the figure that the alignment time is consistently under 1 second for each alignment.

6.5.3 A comparison with SurfComp

In this experiment, we compare MS3ALIGN with the SURFCOMP tool [27]. The authors of SURFCOMP validate using the ligand surfaces from two datasets. The

first dataset consists of thermolysin inhibitor ligands of two kinds, the first containing tryptophan and the second with an aliphatic residue at the C-terminal end. The second dataset consists of ligands bound with the Dihydrofolate Reductase (DHFR) enzyme. In both experiments, surfaces obtained from ligand molecules are aligned. Hofbauer *et al.* consider two types of physico-chemical properties on molecular surfaces to determine alignments. Since they conclude that electrostatic potential (ESP) results in better alignments, we only compare against these alignments. The chemical structures of the ligands in both datasets are presented in Figures 6.6 and 6.10. For these experiments, the R_c parameter is set to 1.2\AA since we wish to study alignments at the scale of a single atom. The T_s parameter is set to 0.06 for the first dataset and 0.1 for the second dataset. The T_{ms} parameter is set to 0.09 and 0.15 respectively. The T_{mrd} parameter is set equal to R_c . In the following paragraphs, we discuss the alignments determined by MS3ALIGN for each of the datasets in detail and compare them with SURFCOMP.

In the thermolysin dataset, SURFCOMP compares eight thermolysin inhibitor ligands considered in two groups. The first group consists of ligands from PDBs 1THL, 1TLP, 1TMN and 3TMN. In our experiment, we were unable to extract the ligand structure from 3TMN because of a failure in the PDB2PQR tool which is used in a preprocessing step to compute the molecular skin surface. Hence remove 3TMN from this list. The second group consists of ligands from PDBs 4TLN, 5TLN, 5TMN, and 6TMN. For consistency of labeling of datasets with respect to Hofbauer *et al.*, we use the PDB id to reference the ligand considered. We were able to determine the same alignments as SURFCOMP for the first group. However, in the case of 1THL-1TMN our sixth best alignment corresponds to the SURFCOMP's best alignment. For the same pair 1THL-1TMN, SURFCOMP reports an alignment of the tryptophan moiety of both ligands. MS3ALIGN reports it as the sixth best alignment with other meaningful alignments being detected as

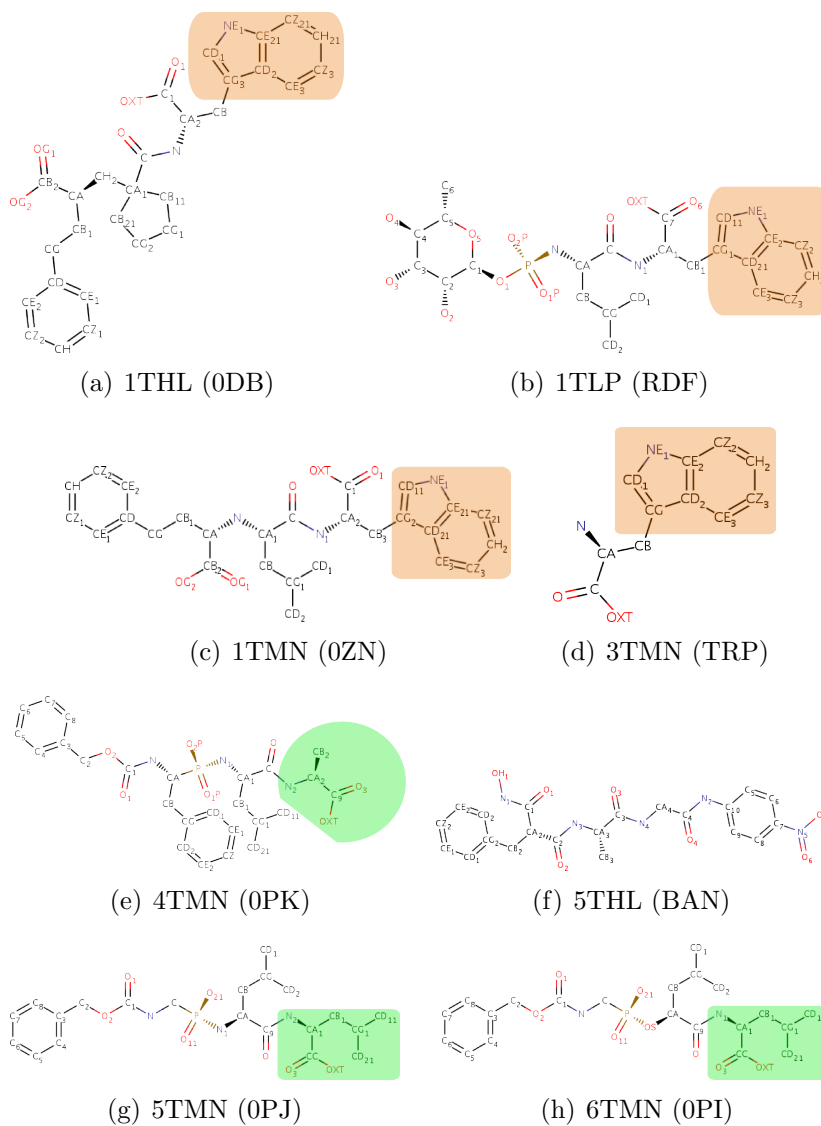


Figure 6.6: Thermolysin Inhibitors. (a)-(d) Ligand structures with Tryptophan shown in the orange boxes. (e)-(h) Ligand structures with aliphatic residue shown within green shaded regions. The ligand structure images were generated via the RCSB's [2] Protein Data Bank website.

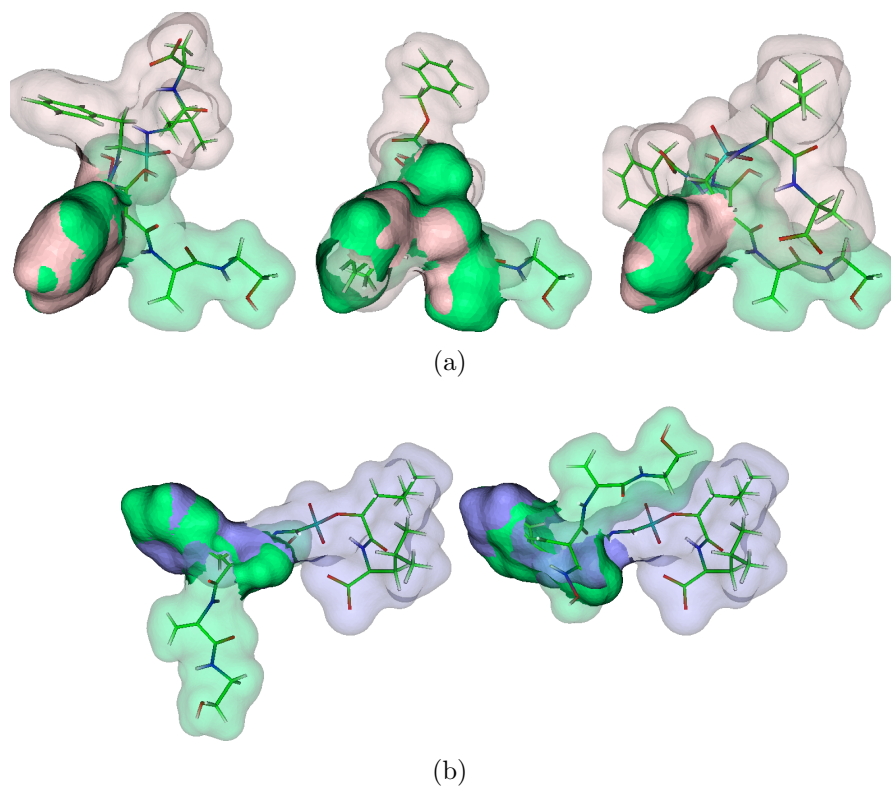


Figure 6.7: Alignments of the ligand surfaces found by MS3ALIGN and not SURF-COMP for the thermolysin inhibitor dataset. The aligning portions of the surface are shown as opaque and the rest is shown with transparency. A stick representation of the ligands is also shown. (a) The best three alignments of the 4TMN's OPK ligand (light pink) with 5TLN's BAN ligand (green) aligns the two aromatic rings of OPK with those of BAN. (b) The best two results of 6TMN's OPJ ligand (dark blue) with the 5TLN's BAN (green) show alignment of aromatic rings from both ligands.

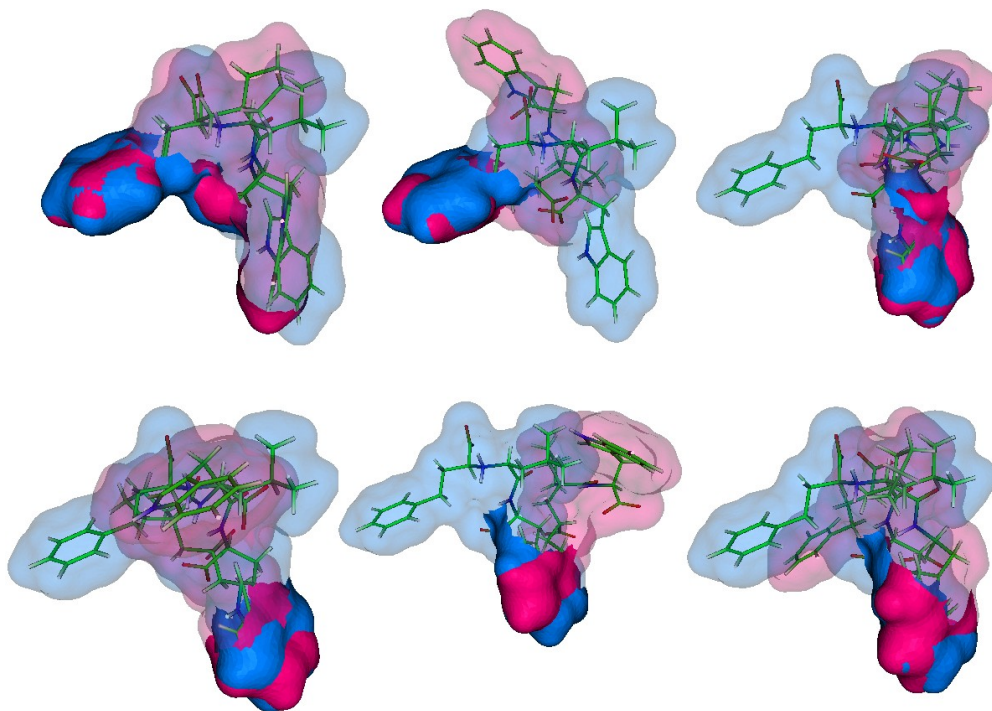


Figure 6.8: Alignment of 1THL with 1TMN exhibits multiple partial alignments. In all figures 1TMN is fixed (blue) and 1THL (pink) is transformed. (*Top Left and Center*): The two mirrored alignments of the aromatic rings in 1THL and 1TMN are detected as the best two alignments. (*Top Right and Bottom Left*): The two mirrored alignments of the benzyl ring of 1THL with the hetero-cycle of tryptophan in 1TMN are detected third and fourth best alignments. (*Bottom Center*): A variant of the the alignment of the aromatic ring in 1THL with the other parts of the penta-cycle of tryptophan in 1TMN. (*Bottom Right*): Alignment of the tryptophan parts of both ligands.

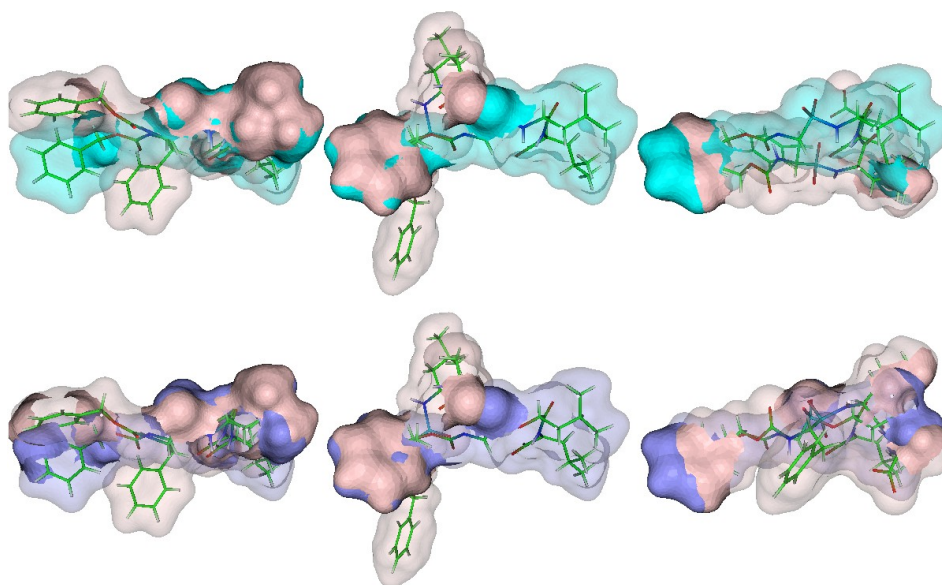


Figure 6.9: Alignments of the 4TMN's 0PK ligand (light pink) with: (*left column*) 5TMN's 0PJ ligand(cyan). (*right column*) 6TMN's 0PK ligand (purple). The alignment of 0PK in 5TMN with BAN in 5TLN is shown in Figure 6.7(a).

better. Figure 6.8 shows the other alignments detected for the pair 1THL-1TMN. From the visualization, we conclude that they are also chemically relevant. Also, Hofbauer *et al.* state that they were not able to determine relevant alignments of the ligand in 5TLN with the other ligands. Figure 6.7 show the best alignments of the ligand in 5TLN with the ligands in 4TMN and 5TMN using MS3ALIGN. Since the ligand 0PI (6TMN) is structurally very similar to the ligand 0PJ (5TMN), the same alignments with respect to other ligands were also found, and thus images of these alignments are omitted. Thus, we find that MS3ALIGN aligns relevant portions of the surfaces of the ligand BAN in 5TLN with all the other considered ligands. For many pairs, we determine partial alignments of other substructures. Figure 6.9 presents the additional alignments of 4TMN with the others in its group.

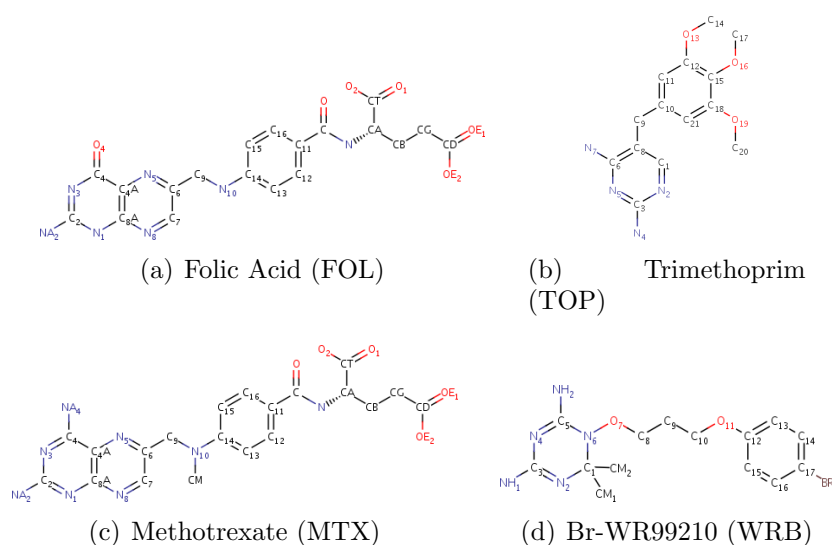


Figure 6.10: DHFR ligands FOL, MTX, TOP, and WRB taken from pdb 1DHF, 1DF7, 1DG5, and 1DG7 respectively. The ligand structure images were generated via the RCSB’s [2] Protein Data Bank website.

Dihydrofolate Reductase Dataset

In the DHFR dataset, alignments of surfaces of four ligands interacting with DHFR are analyzed. The ligands used are Folic acid (FOL), Methotrexate (MTX), Trimethoprim (TMP), and BR-WR99210 (WRB) from PDBs 1DHF, 1DF7, 1DG5 and 1DG7 respectively. Figure 6.10 shows the chemical structures of the considered ligands. We note that SURFCOMP reports only two good alignments of FOL with MTX and WRB. We were able to recover a similar alignment of FOL and MTX. MS3ALIGN does not determine the same alignment reported by SURFCOMP for the case of FOL and WRB. Figure 6.11(a) shows the alignment determined by MS3ALIGN where the alignment is along the C=N-C=N moieties with amino groups at the two and four positions from triazine and pteridine of WRB and MTX respectively. SURFCOMP reports an alignment where the brominated benzyl group of WRB aligns with the central benzyl group of FOL which is different from the best alignment we determine. Relevant alignments determined by MS3ALIGN for

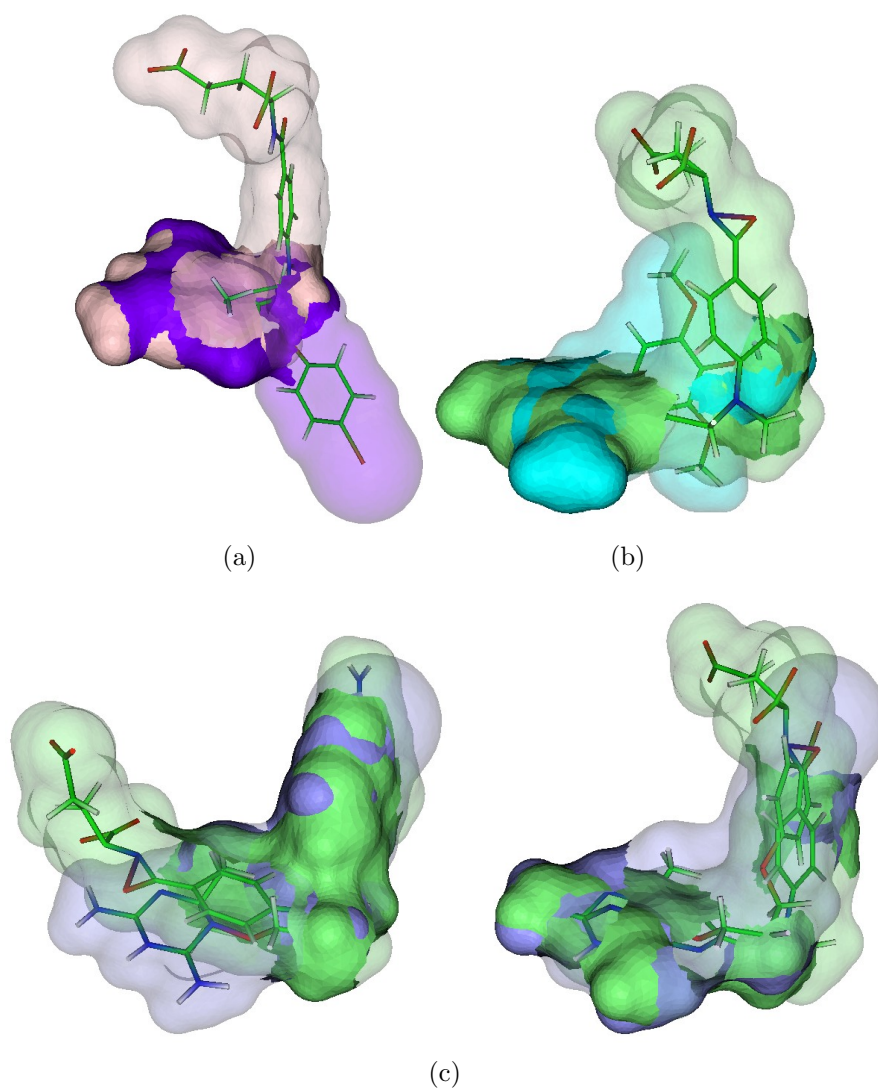


Figure 6.11: Alignments of the ligand surfaces found by MS3ALIGN and not SURFCOMP for the DHFR dataset. The aligning portions of the surface are shown as opaque and the rest is shown with transparency. A stick representation of the ligands is also shown. (a) The best alignment of FOL (light pink) with WRB (purple) is along the two amino groups attached to the aromatic ring in both ligands. (b) Alignment between MTX (green) and TOP (cyan), where the two amino groups attached to an aromatic ring align. (c) The best two alignments MTX (green) and WRB (purple).

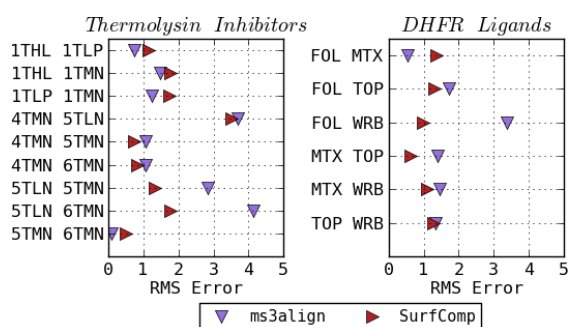


Figure 6.12: Plot of the RMS distance between ligand surfaces after alignment using MS3ALIGN and SURFCOMP. (left) Plot for the dataset of thermolysin inhibitor ligands. The ligands are identified by the PDB they were extracted from. (right) Plot for the dataset of ligands bound to DHFR enzymes. The ligands are identified by their abbreviation.

the pairs MTX-TMP, MTX-WRB, and FOL-WRB, that were not determined by SURFCOMP are shown in Figures 6.11.

Figure 6.12 presents the RMS distances between the pairs of surfaces after alignment using both MS3ALIGN and SURFCOMP. We note here that small RMSD values ($< 1.5\text{\AA}$) are indicative of successful alignments whereas larger RMSD values (1.5–5) do not necessarily indicate failure of the alignment. This is particularly true when the reported alignment is that of a partial alignment of surfaces. Table 6.1 tabulates and compares the aligning portions of the ligands in both datasets using MS3ALIGN and SURFCOMP. In conclusion, we were able to replicate all alignments, except one, reported by Hofbauer *et al.* in their evaluation of SURFCOMP. Additionally, we were able to obtain other chemically relevant alignments in many cases.

6.5.4 A validation using PocketMatch and PyMol

In this experiment, we validate alignments generated using MS3ALIGN against those generated using PYMOL [3]. We curate a dataset of thirty one protein structures from the PDB [2], each interacting with one of eight types of ligands. A variety of ligands ranging from small sugars such as glucose to molecules containing substituted sugars such as NDP, fatty acids such as ACD (arachidonic acid), vitamins such as Biotin (BTN) and Retinoic Acid (REA) are chosen. The structural

Ligand Pair	SURFCOMP Aligns	MS3ALIGN Aligns
1THL-1TLP	Tryptophan moiety	Tryptophan moiety
1THL-1TMN	Tryptophan moiety	Benzyl moiety Tryptophan moiety (6 th best)
1THL-3TMN	Tryptophan moiety	-
1TLP-1TMN	Tryptophan moiety	Tryptophan moiety
1TLP-3TMN	Tryptophan moiety	-
1TMN-3TMN	Tryptophan moiety	-
4TMN-5TLN	-	Benzyl moiety
4TMN-5TMN	L-Alanine moiety	L-Alanine moiety
4TMN-6TMN	L-Alanine moiety	L-LAanine moiety
5TLN-5TMN	-	Benzyl moiety
5TLN-6TMN	-	Benzyl moiety
5TMN-6TMN	4-methyl pentanoic acid group	4-methyl pentanoic acid group
FOL-TOP	-	-
FOL-MTX	Entire ligands	Entire ligands
FOL-WRB	Benzyl moiety	C=N-C=N part of triazine and pteridine groups along with amino groups
TOP-MTX	-	C=N-C=N part of triazine and pteridine groups along with amino groups
TOP-WRB	-	C=N-C=N part of triazine and pteridine groups along with amino groups
MTX-WRB	-	C=N-C=N part of triazine and pteridine groups along with amino groups

Table 6.1: Comparison of the chemically relevant alignments of SURFCOMP and MS3ALIGN. The chemical structures of the ligands are presented in Figures 6.6 and 6.10

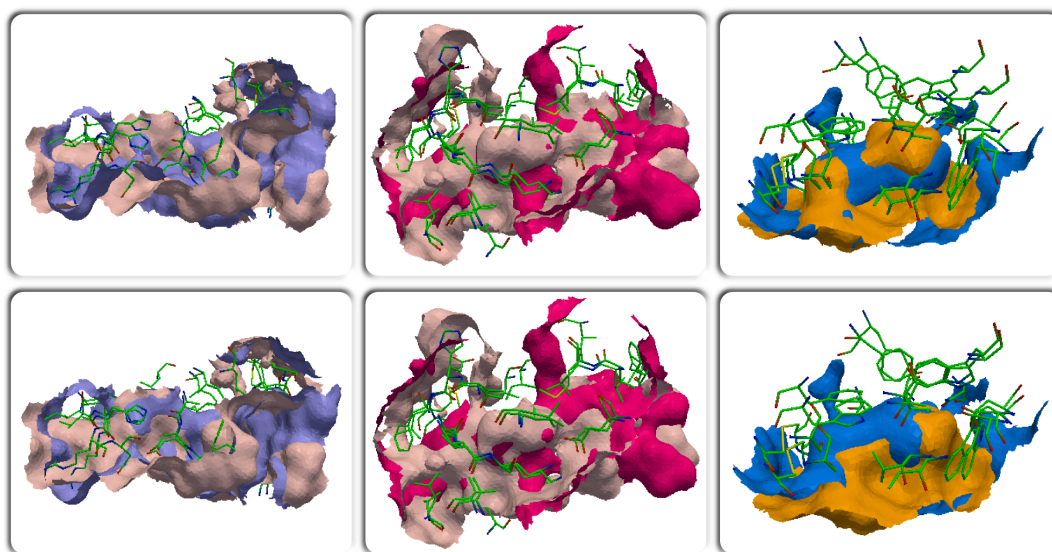


Figure 6.13: Alignments of pockets formed by the ligands STI in 4BKJ and 3HEC (*left*), STI in 3K5V and 3HEC (*middle*), and BTN in 4GGZ and 4JNJ (*right*), using MS3ALIGN (*top-row*) and PYMOL (*bottom-row*).

variability of the pocket is quantified using POCKETMATCH [136]. For each binding site, POCKETMATCH generates 90 sorted lists of distances from the three dimensional coordinates and chemical properties of the site. For a pair of sites, a normalized score based on the similarity of the pair of 90 lists is computed. Sites having a POCKETMATCH Pmax score greater than 0.6 are statistically shown to be structurally similar, with a score of 1 indicating identical sites. Independently, an alignment of the pocket residues is generated using PYMOL’s “super” command. This command aligns proteins using a dynamic programming approach followed by multiple refinement cycles that improve the fit by eliminating pairings with high relative variability.

For this experiment, the pocket surfaces of each ligand type is aligned with each other using MS3ALIGN with $R_c = 1.2\text{\AA}$, $T_s = 0.1$, $T_{ms} = 0.15$, $T_{mrd} = 1.2\text{\AA}$. In Figure 6.14, the RMS distance between the pockets transformed by PYMOL’s and MS3ALIGN’s alignment is plotted for the set of considered pocket pairs. The

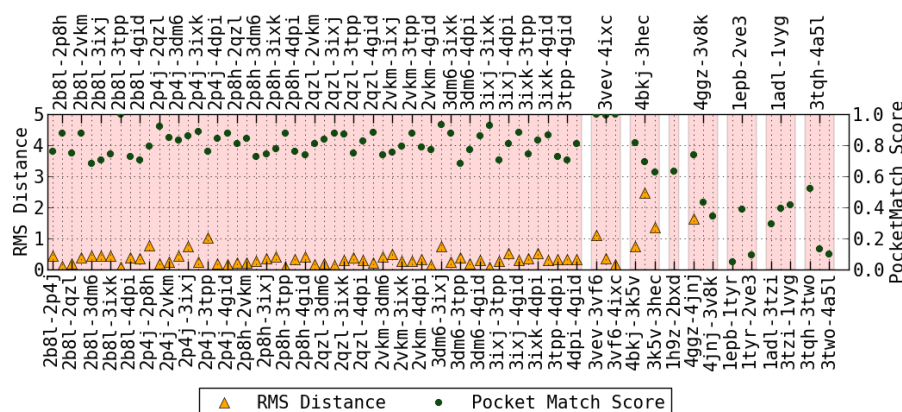


Figure 6.14: The RMS distance between pockets transformed by PYMOL’s and MS3ALIGN’s alignment. The corresponding POCKETMATCH indicate overall consensus between the different methods.

POCKETMATCH score between each pair is also shown for comparison. When the RMS distance between the pockets transformed by PYMOL’s and MS3ALIGN’s alignment is less than 1.2\AA , we conclude that both alignments are equivalent. In Figure 6.13, we visually verify the validity of alignments of pairs where the RMS distance is between 1.2 and 5\AA . We conclude that we were able to successfully determine alignments of pocket surfaces whose POCKETMATCH score is greater than 0.7 and in some cases even 0.6.

6.5.5 Aligning Electron Microscopy Iso-Surfaces

In this experiment, we consider two sets of related iso-surfaces generated from cryo electron microscopy scans obtained from the EMDDataBank [26]. The first dataset comprises of a related set of Fragment antigen binding of HIV antibodies. The second dataset comprises of a related set of HIV antibodies. Both datasets are available at resolutions of approximately 20\AA . Table 6.2 summarizes the details of the two datasets used. An iso-surface is defined as the set of points where the density value is equal to a pre-specified constant. Molecular surfaces are extracted

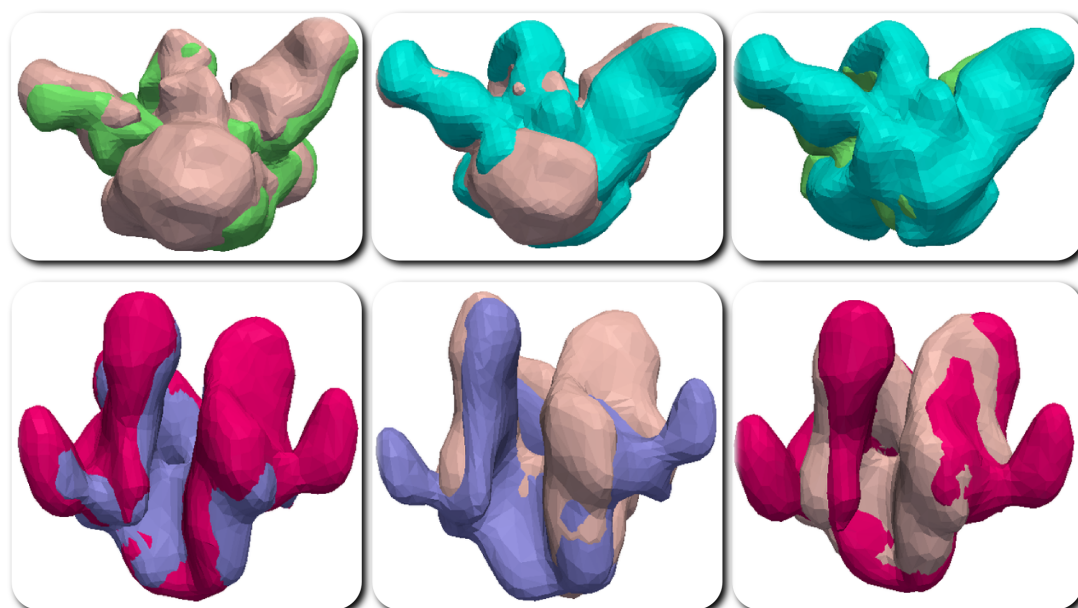


Figure 6.15: Alignments of the iso-surfaces of the cryo Electron Microscopy datasets. Pairwise alignments of the first dataset is shown in the top row and pairwise alignments of the second dataset is shown in the bottom row. Pairwise alignments between: *top-left*) 5918 (skin pink) and 5919 (green), *top-middle*) 5918 (skin pink) and 5920 (cyan), *top-right*) 5919 (green) and 5920 (cyan), *bottom-left*) 5323 (purple) and 5324 (magenta), *bottom-middle*) 5323 (purple) and 5325 (skin pink), and *bottom-right*) 5324 (magenta) and 5325 (skin pink).

EMDB ID	Iso-Value	Imaging Resolution (\AA)	Imaging Box Size ($\text{\AA} \times \text{\AA} \times \text{\AA}$)
5918	2.05	21	$328 \times 328 \times 328$
5919	2.69	19	$348 \times 348 \times 348$
5920	2.27	25	$328 \times 328 \times 328$
5323	2.2	20	$410 \times 410 \times 410$
5324	1.45	20	$410 \times 410 \times 410$
5325	2.0	20	$410 \times 410 \times 410$

Table 6.2: Electron Microscopy datasets used in the alignment experiments. The iso-value refers to the density value suggested by EMData bank for representing the molecular surface.

A	B	RMSD (\AA)
5918	5919	6.69
5918	5920	7.00
5919	5920	8.32
5323	5324	5.71
5323	5325	8.74
5324	5325	10.75

Table 6.3: RMS distances between pairs of iso-surfaces after alignment from both datasets shown in Table 6.2.

from cryo-EM data by computing the iso-surface at a carefully chosen iso-value.

For both datasets, we set the parameter $R_c = 30\text{\AA}$, $T_s = 0.05$, $T_{ms} = 0.1$, $T_{mrd} = 30\text{\AA}$, and computed alignments using MS3ALIGN. Figure 6.15 shows the respective pairwise alignments of the iso-surfaces of both datasets. Table 6.2 shows the RMS distance between pairs surfaces from the two datasets after alignment using MS3ALIGN. Here the RMSD is computed after alignment as the root mean square of the closest distance from every point of the first surface to the second surface and vice versa. Thus, we conclude that the alignments were successful.

6.6 Conclusions

We presented a method to align molecular surfaces by identifying and establishing correspondences between significant protrusions on the surface. We also present MS3ALIGN, a tool that implements this method. A key advantage of our method is robust segmentation of the surface into segments that can be individually evaluated for correspondences. Furthermore, due to its purely geometric design, it is applicable to molecular surfaces arising from various sources such as the PDB and Electron-Microscopy scans. This is a key advantage over existing methods such as SURFCOMP, PBSALIGN, and PYMOL, which rely on protein sequence data and other derived scalar values such as the electro-static potential, which are often not directly available/computable. In the future, we plan to expand MS3ALIGN to align surfaces by including other geometric properties such as spherical harmonics and Zernike coefficients of individual segments. These global properties may be applied to smaller segments resulting in a method for determining alignments using a blend of local and global properties. Currently, we use only the local curvature property. Another potential avenue is the further refinement of the alignment using physico-chemical properties. In terms of leveraging topological properties, the application of the 3D Morse-Smale complex for three-dimensional electron density distributions is another avenue. Here, topology based methods potentially have an advantage of being able to study stable core structures that track the backbone of protein chains. These may then be leveraged to develop efficient identification and alignment algorithms.

Chapter 7

Conclusions

The focus of this thesis has been two-fold. The first part focused on efficient algorithms to compute the Morse-Smale complex of large 2D and 3D domains. The second part focused on two novel applications of the Morse-Smale complex.

Computation

In this part, we described a discrete Morse theory based method to compute the Morse-Smale complex. It begins by first defining a valid discrete Morse function which is then used to compute discrete gradient consistently and in parallel. Next, we describe algorithms to efficiently traverse the structure of the gradient field to determine the combinatorial and geometric structure of the Morse-Smale complex. These methods were implemented for 2D and 3D structure grids. They were both evaluated using synthetic and real world datasets. We found that both implementations exhibited a near linear scaling with data sizes. We compared desktop implementations of other Morse-Smale complex algorithms using standard datasets. We found a significant runtime advantage of our method over other methods, which is an order of magnitude better for all the considered datasets.

We also presented theoretical results that allow us to partially reconcile the simplification of Morse-Smale complexes with the notion of persistence. In particular, we showed that if one were to only topologically cancel extremum-saddle critical points pairs in the Morse-Smale complex in increasing order of the least absolute difference in function value between the two critical points, then resulting set of critical point pairs are persistent critical point pairs.

Applications

In the second part of the thesis, we described two novel applications of the Morse-Smale complex in the field of cosmology and structural biology.

Exploring Cosmic Filaments: In this application, we described a framework to explore the structure of cosmic filaments using the Morse-Smale complex. The Morse-Smale complex plays a key role in identifying filamentary structures. In particular, filaments are determined in a hierarchy of Morse-Smale complexes so as to be able to interactively explore the topologically rich filamentary structure. We used two classes of cosmological simulation datasets to test our framework and showcase its capabilities. We first compared cosmic filaments extracted with those found by other methods. We reconfirm that we are able to identify similar structures and in few cases more relevant structures. Then, we used our framework to explore filaments in three different density regimes, those that existed in high density regions, those that existed in low-density regions, and those that stretch from high density to low density regions.

Aligning Molecular Surfaces: In this application, we used the Morse-Smale complex to identify and segment significant protrusions on molecular surfaces. These protrusions allowed us develop an efficient method to align molecular surfaces. The alignment method was implemented as a tool named MS3ALIGN. We

evaluated the performance of MS3ALIGN to study its resilience to noise and also to determine alignments when only parts of the surfaces overlapped. We compared the method with related software and showcased its advantages. We validated the tool with the alignments determined using PYMOL's [3] and scored using POCKETMATCH [136].

Future Directions

In recent years, the body of applications that leverage Morse-Smale complexes for topological analysis has been growing rapidly. Therefore, near real time computation is of immediate value to anyone looking for such analysis. Furthermore, our methods are easily accessible due to its deployability on commodity CPU and GPU systems. We believe that our methods will find wide-spread application in time-varying datasets as well as ensemble datasets, where rapidly emerging data needs to be quickly and automatically analyzed for features. Examples of such data include climate simulations, cosmological simulations, fluid dynamics, and hydrodynamic mixing simulations. In such data streams, identifying key frames in the data is of vital interest due to volume and rapidity of the data.

Bibliography

- [1] W. Wang, C. Bruyere, and B. Kuo, “Competition data set and description in 2004 IEEE Visualization design contest,” <http://vis.computer.org/vis2004contest/data.html>. 1, 67, 68
- [2] H. M. Berman *et al.*, “The protein data bank,” *Nucleic Acids Res*, vol. 28, pp. 235–242, 2000. 1, 9, 132, 135, 139, 143, 145
- [3] “The PyMOL Molecular Graphics System, version 1.2r3pre, Schrödinger, LLC.” 1, 132, 145, 154
- [4] D. Laney, P. T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci, “Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1053–1060, 2006. 2, 3, 28, 31
- [5] A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbotham, and B. Hamann, “Topologically clean distance fields,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1432–1439, 2007. 3, 24, 28, 31
- [6] W. Feng, J. Huang, T. Ju, and H. Bao, “Feature correspondences using Morse Smale complex.” *The Visual Computer*, 2013. 3

- [7] F. Cazals, F. Chazal, and T. Lewiner, “Molecular shape analysis based upon the Morse-Smale complex and the Connolly function,” in *Proc. 19th Ann. ACM Sympos. Comput. Geom.*, 2003, pp. 351–360. [3](#), [5](#), [25](#), [26](#), [49](#), [62](#)
- [8] E. Danovaro, L. D. Floriani, and M. Vitali, “Multi-resolution Morse-Smale Complexes for Terrain Modeling.” in *ICIAP*, 2007, pp. 337–342. [3](#)
- [9] T. Sousbie, “The persistent cosmic web and its filamentary structure I. theory and implementation,” *Monthly Notices of the Royal Astronomical Society*, vol. 414, no. 1, pp. 350–383, 2011. [3](#), [29](#), [30](#), [31](#), [90](#), [107](#)
- [10] T. Sousbie, C. Pichon, and H. Kawahara, “The persistent cosmic web and its filamentary structure ii. illustrations,” *MNRAS*, vol. 414, no. 1, pp. 384–403, 2011. [3](#), [29](#), [31](#), [90](#), [107](#)
- [11] D. Gunther, R. Boto, J. Contreras-Garcia, J. Piquemal, and J. Tierny, “Characterizing molecular interactions in chemical systems,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 99, no. PrePrints, p. 1, 2015. [3](#), [29](#), [31](#)
- [12] A. Gyulassy, P. T. Bremer, V. Pascucci, and B. Hamann, “A practical approach to Morse-Smale complex computation: scalability and generality,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1619–1626, 2008. [4](#), [5](#), [6](#), [25](#), [26](#), [27](#), [34](#), [35](#), [59](#), [62](#), [75](#), [76](#), [95](#)
- [13] V. Robins, P. Wood, and A. Sheppard, “Theory and algorithms for constructing discrete Morse complexes from grayscale digital images,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1646–1658, 2011. [4](#), [25](#), [26](#), [27](#), [62](#), [63](#), [95](#)
- [14] D. Günther, J. Reininghaus, H. Wagner, and I. Hotz, “Memory efficient

- computation of persistent homology for 3D image data using discrete Morse theory,” in *Sibgrapi 2011 - Technical Papers*, 2011. [4](#), [6](#), [25](#), [27](#), [62](#), [75](#), [76](#)
- [15] N. Shivashankar, S. M, and V. Natarajan, “Parallel computation of 2D Morse-Smale complexes,” *IEEE Trans. Vis. & Comp. Graphics*, vol. 18, no. 10, pp. 1757–1770, 2012. [4](#), [27](#), [95](#), [128](#)
- [16] N. Shivashankar and V. Natarajan, “Parallel computation of 3D Morse-Smale complexes,” *Computer Graphics Forum*, vol. 31, no. 3pt1, pp. 965–974, 2012. [4](#), [27](#), [95](#)
- [17] —, “Efficient software for programmable visual analysis using the morse-smale,” in *Workshop on Topology-Based methods in Visualization*, 2015. [5](#)
- [18] N. Shivashankar, P. Pranav, V. Natarajan, R. van de Weygaert, E. G. P. Bos, and S. Rieder, “Felix: A topology based framework for visual exploration of cosmic filaments,” *IEEE Transactions on Visualization and Computer Graphics*, Under Submission. [5](#), [30](#), [31](#)
- [19] N. Shivashankar, S. Patil, A. Bhosle, N. Chandra, and V. Natarajan, “MS3ALIGN: An efficient molecular surface aligner using the topology of surface curvature,” *BMC Bioinformatics*, Under Submission. [5](#), [31](#), [32](#)
- [20] D. Günther, H.-P. Seidel, and T. Weinkauff, “Extraction of dominant extremal structures in volumetric data using separatrix persistence,” *Comp. Graph. Forum*, vol. 31, no. 8, pp. 2554–2566, 2012. [5](#), [90](#)
- [21] R. Forman, “A user’s guide to discrete Morse theory,” *Séminaire Lotharingien de Combinatoire*, vol. 48, 2002. [6](#), [11](#), [17](#), [21](#), [24](#), [95](#)
- [22] G. van Rossum, “Python tutorial,” Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, 1995. [6](#)

- [23] “Gnu lesser general public license,” Free Software Foundation. [Online]. Available: <http://www.gnu.org/licenses/lgpl.html> 6, 10
- [24] H. Edelsbrunner, D. Letscher, and A. Zomorodian., “Topological persistence and simplification,” *Disc. & Comp. Geom.*, vol. 28, no. 4, pp. 511–533, 2002. 7, 14, 76, 90, 128
- [25] J. R. Bond, L. Kofman, and D. Pogosyan, “How filaments of galaxies are woven into the cosmic web,” *Nature*, vol. 380, pp. 603–606, 1996. 7, 85
- [26] C. L. Lawson, M. L. Baker *et al.*, “EMDataBank.org: unified data resource for CryoEM,” *Nucleic Acids Research*, vol. 39, 2011. 9, 148
- [27] C. Hofbauer, H. Lohninger, and A. Aszódi, “SurfComp: a novel graph-based approach to molecular surface comparison.” *J. Chem. Inf. Comp. Sci.*, vol. 44, no. 3, pp. 837–847, 2004. 10, 123, 125, 132, 137
- [28] Y. Matsumoto, *An Introduction to Morse Theory*. Amer. Math. Soc., 2002, translated from Japanese by K. Hudson and M. Saito. 11
- [29] J. R. Munkres, *Elements of Algebraic Topology*. Addison-Wesley, 1984. 11, 20, 21
- [30] H. Edelsbrunner, J. Harer, and A. Zomorodian., “Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds,” *Discrete and Computational Geometry*, vol. 30, no. 1, pp. 87–107, 2003. 14, 20, 23, 24, 25, 78, 82, 89
- [31] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci., “Morse-Smale complexes for piecewise linear 3-manifolds,” in *Proc. 19th Ann. Sympos. Comput. Geom.*, 2003, pp. 361–370. 20, 24, 25, 95
- [32] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2001. 20

- [33] H. Edelsbrunner and E. P. Mücke, “Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms,” *ACM Trans. Graph.*, vol. 9, no. 1, pp. 66–104, 1990. [22](#), [38](#)
- [34] S. Smale, “Generalized Poincaré’s conjecture in dimensions greater than four,” *Ann. of Math.*, vol. 74, pp. 391–406, 1961. [23](#)
- [35] —, “On gradient dynamical systems,” *Ann. of Math.*, vol. 74, pp. 199–206, 1961. [23](#)
- [36] P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci, “A topological hierarchy for functions on triangulated surfaces,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 4, pp. 385–396, 2004. [24](#)
- [37] A. Gyulassy, V. Natarajan, V. Pascucci, P. T. Bremer, and B. Hamann, “A topological approach to simplification of three-dimensional scalar fields,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 474–484, 2006. [24](#), [25](#), [35](#)
- [38] J. Milnor, “Morse theory,” *Journal of Mathematical Physics*, 1963. [24](#), [89](#)
- [39] J. Reininghaus and I. Hotz, “Combinatorial 2d vector field topology extraction and simplification,” in *Topological Methods in Data Analysis and Visualization*, ser. Mathematics and Visualization, V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, Eds. Springer Berlin Heidelberg, 2011, pp. 103–114. [24](#)
- [40] J. Reininghaus, C. Lowen, and I. Hotz, “Fast combinatorial vector field topology,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1433–1443, 2011. [24](#)

- [41] U. Bauer, C. Lange, and M. Wardetzky, “Optimal topological simplification of discrete functions on surfaces,” *Discrete and Computational Geometry*, pp. 1–31, 2011. [24](#)
- [42] T. Lewiner, H. Lopes, and G. Tavares, “Applications of Forman’s discrete Morse theory to topology visualization and mesh compression,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 5, pp. 499–508, 2004. [26](#), [62](#)
- [43] T. Peterka, R. Ross, A. Gyulassy, V. Pascucci, W. Kendall, H.-W. Shen, T.-Y. Lee, and A. Chaudhuri, “Scalable parallel building blocks for custom data analysis,” in *Proc. IEEE Symposium on Large-Scale Data Analysis and Visualization (LDAV)*, 2011. [27](#), [62](#)
- [44] A. Gyulassy, V. Pascucci, T. Peterka, and R. B. Ross, “The parallel computation of Morse-Smale Complexes.” in *International Parallel and Distributed Processing Symposium*, 2012, pp. 484–495. [27](#), [62](#)
- [45] S. Gerber, V. Pascucci, and R. T. Whitaker, “Morse-smale regression,” *Journal of Computational and Graphical Statistics*, 2012. [30](#), [31](#)
- [46] G. Blelloch, “Scans as primitive parallel operations,” *IEEE Transactions on Computers*, vol. 38, pp. 1526–1538, 1987. [65](#)
- [47] G. E. Blelloch, “Prefix sums and their applications,” in *Synthesis of Parallel Algorithms*, J. H. Reif, Ed. Morgan Kaufmann, 1991. [65](#)
- [48] “Boost C++ library.” [Online]. Available: <http://www.boost.org/> [71](#)
- [49] R. van de Weygaert and J. R. Bond, “Clusters and the Theory of the Cosmic Web,” in *A Pan-Chromatic View of Clusters of Galaxies and the Large-Scale*

- Structure*, ser. Lecture Notes in Physics, Berlin Springer Verlag, M. Plionis, O. López-Cruz, and D. Hughes, Eds., vol. 740, 2008, p. 335. [85](#)
- [50] A. H. Guth, “Inflationary universe: A possible solution to the horizon and flatness problems,” *Physical Review D*, vol. 23, pp. 347–356, 1981. [85](#)
- [51] A. D. Linde, “Coleman-Weinberg theory and the new inflationary universe scenario,” *Physics Letters B*, vol. 114, pp. 431–435, 1982. [85](#)
- [52] —, “Scalar field fluctuations in the expanding universe and the new inflationary universe scenario,” *Physics Letters B*, vol. 116, pp. 335–339, 1982. [85](#)
- [53] D. N. Spergel *et al.*, “Three-year wilkinson microwave anisotropy probe (wmap) observations: Implications for cosmology,” *The Astrophysics Journal Supplement Series*, vol. 170, pp. 377–408, 2007. [85](#)
- [54] E. Komatsu *et al.*, “Seven-year wilkinson microwave anisotropy probe (wmap) observations: Cosmological interpretation,” *The Astrophysical Journal Supplement Series*, vol. 192, no. 2, 2011. [85](#)
- [55] P. Collaboration *et al.*, “Planck 2013 results. xvi. cosmological parameters,” *arXiv preprint arXiv:1303.5076*, 2013. [85](#)
- [56] B. S. Ryden and A. L. Melott, “Voids in Real Space and in Redshift Space,” *Astrophys. J.*, vol. 470, p. 160, 1996. [86](#), [121](#)
- [57] J. D. Schmidt, B. S. Ryden, and A. L. Melott, “The Size and Shape of Voids in Three-dimensional Galaxy Surveys,” *Astrophys. J.*, vol. 546, pp. 609–619, 2001. [86](#), [121](#)
- [58] D. Park and J. Lee, “Void Ellipticity Distribution as a Probe of Cosmology,” *Physical Review Letters*, vol. 98, no. 8, p. 081301, Feb. 2007. [86](#), [121](#)

- [59] G. Lavaux and B. D. Wandelt, “Precision cosmology with voids: definition, methods, dynamics,” *MNRAS*, vol. 403, pp. 1392–1408, 2010. [86](#), [121](#)
- [60] E. G. P. Bos, R. van de Weygaert, K. Dolag, and V. Pettorino, “The darkness that shaped the void: dark energy and cosmic voids,” *MNRAS*, vol. 426, pp. 440–461, 2012. [86](#), [106](#), [121](#)
- [61] G. Lavaux and B. D. Wandelt, “Precision Cosmography with Stacked Voids,” *Astrophys. J. Lett.*, vol. 754, p. 109, 2012. [86](#), [121](#)
- [62] J. Clampitt, Y.-C. Cai, and B. Li, “Voids in modified gravity: excursion set predictions,” *MNRAS*, vol. 431, pp. 749–766, 2013. [86](#), [121](#)
- [63] P. M. Sutter, A. Pisani, B. D. Wandelt, and D. H. Weinberg, “A measurement of the alcock-paczyński effect using cosmic voids in the sdss,” *MNRAS*, vol. 443, pp. 2983–2990, 2014. [86](#), [121](#)
- [64] J. Tyson, *Gravitational Lensing by Clusters of Galaxies*, 2000. [86](#)
- [65] J.-P. Kneib and P. Natarajan, “Cluster lenses,” *Astronomy & Astrophysics*, vol. 19, p. 47, 2011. [86](#)
- [66] J. P. Dietrich, N. Werner, D. Clowe, A. Finoguenov, T. Kitching, L. Miller, and A. Simionescu, “A filament of dark matter between two clusters of galaxies,” *Nature*, vol. 487, pp. 202–204, 2012. [86](#)
- [67] F. Hoyle, “The Origin of the Rotations of the Galaxies,” in *Problems of Cosmical Aerodynamics*, J. M. Burgers and H. C. van de Hulst, Eds., 1951, p. 195. [86](#)
- [68] P. J. E. Peebles, “Origin of the Angular Momentum of Galaxies,” *Astrophys. J.*, vol. 155, p. 393, 1969. [86](#)

- [69] S. D. M. White, “Angular momentum growth in protogalaxies,” *Astrophys. J.*, vol. 286, pp. 38–41, Nov. 1984. [86](#)
- [70] J. Lee and U.-L. Pen, “Cosmic Shear from Galaxy Spins,” *Astrophys. J. Lett.*, vol. 532, pp. L5–L8, 2000. [86](#)
- [71] B. J. T. Jones, R. van de Weygaert, and M. A. Aragón-Calvo, “Fossil evidence for spin alignment of Sloan Digital Sky Survey galaxies in filaments,” *MNRAS*, vol. 408, pp. 897–918, 2010. [86](#)
- [72] E. Tempel, R. S. Stoica, and E. Saar, “Evidence for spin alignment of spiral and elliptical/S0 galaxies in filaments,” *MNRAS*, vol. 428, pp. 1827–1836, 2013. [86](#)
- [73] P. Peebles, *The Large Scale Structure of the Universe*. Princeton Univ. Press, Princeton, New Jersey, 1980. [87](#)
- [74] S. F. Shandarin and I. B. Zeldovich, “Topology of the large-scale structure of the universe,” *Comments on Astrophysics*, vol. 10, pp. 33–45, 1983. [88](#)
- [75] J. D. Barrow, S. P. Bhavsar, and D. H. Sonoda, “Minimal spanning trees, filaments and galaxy clustering,” *MNRAS*, vol. 216, 1985. [88](#)
- [76] M. Alpaslan, A. S. G. Robotham, D. Obreschkow, S. Penny, S. Driver, P. Norberg, S. Brough, M. Brown, M. Cluver, B. Holwerda, A. M. Hopkins, E. van Kampen, L. S. Kelvin, M. A. Lara-Lopez, J. Liske, J. Loveday, S. Mahajan, and K. Pimbblet, “Galaxy and Mass Assembly (GAMA): fine filaments of galaxies detected within voids,” *MNRAS*, vol. 440, pp. L106–L110, 2014. [88](#)
- [77] M. A. Aragón-Calvo, B. J. T. Jones, R. van de Weygaert, and J. M. van

- der Hulst, “Spin Alignment of Dark Matter Halos in Filaments and Walls,” *Astrophys. J. Lett.*, vol. 655, pp. L5–L8, 2007. [88](#)
- [78] —, “The multiscale morphology filter: identifying and extracting spatial patterns in the galaxy distribution,” *Astronomy & Astrophysics*, vol. 474, pp. 315–338, 2007. [88](#), [91](#), [108](#), [115](#)
- [79] N. A. Bond, M. A. Strauss, and R. Cen, “Crawling the cosmic network: identifying and quantifying filamentary structure,” *MNRAS*, vol. 409, pp. 156–168, 2010. [88](#)
- [80] M. Cautun, R. van de Weygaert, and B. J. T. Jones, “NEXUS: tracing the cosmic web connection,” *MNRAS*, vol. 429, pp. 1286–1308, 2013. [88](#), [91](#), [95](#), [108](#)
- [81] O. Hahn, C. Porciani, C. M. Carollo, and A. Dekel, “Properties of dark matter haloes in clusters, filaments, sheets and voids,” *MNRAS*, vol. 375, pp. 489–499, 2007. [88](#)
- [82] J. E. Forero-Romero, Y. Hoffman, S. Gottlöber, A. Klypin, and G. Yepes, “A dynamical classification of the cosmic web,” *MNRAS*, vol. 396, pp. 1815–1824, 2009. [88](#)
- [83] N. I. Libeskind, Y. Hoffman, A. Knebe, M. Steinmetz, S. Gottlöber, O. Metuki, and G. Yepes, “The cosmic web and the orientation of angular momenta,” *MNRAS*, vol. 421, pp. L137–L141, 2012. [88](#)
- [84] M. Cautun, R. van de Weygaert, B. J. T. Jones, and C. S. Frenk, “Evolution of the cosmic web,” *ArXiv e-prints*, 2014. [88](#), [89](#), [97](#)
- [85] S. F. Shandarin, “The multi-stream flows and the dynamics of the cosmic web,” *JCAP*, vol. 5, pp. 15–+, 2011. [89](#)

- [86] T. Abel, O. Hahn, and R. Kaehler, “Tracing the dark matter sheet in phase space,” *MNRAS*, vol. 427, pp. 61–76, 2012. [89](#)
- [87] M. C. Neyrinck, “Origami constraints on the initial-conditions arrangement of dark-matter caustics and streams,” *MNRAS*, vol. 427, pp. 494–501, 2012. [89](#)
- [88] S. Shandarin, S. Habib, and K. Heitmann, “Cosmic web, multistream flows, and tessellations,” *Phys. Rev. D*, vol. 85, 2012. [89](#)
- [89] R. S. Stoica, V. J. Martínez, and E. Saar, “A three-dimensional object point process for detection of cosmic filaments.” *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 56 (4), 459–477, vol. 56, p. 1, 2007. [89](#)
- [90] E. Tempel, R. S. Stoica, E. Saar, V. J. Martinez, L. J. Liivamägi, and G. Castellan, “Detecting filamentary pattern in the cosmic web: a catalogue of filaments for the SDSS,” *preprint arXiv:1308.2533*, 2013. [89](#)
- [91] C. Genovese, M. Perone-Pacifico, I. Verdinelli, and L. Wasserman, “Minimax Manifold Estimation,” *ArXiv e-prints*, 2010. [89](#)
- [92] H. Blum, “A Transformation for Extracting New Descriptors of Shape,” *Models for the Perception of Speech and Visual Form*, 1967. [89](#)
- [93] E. Platen, R. van de Weygaert, and B. J. T. Jones, “A cosmic watershed: the WVF void detection technique,” *MNRAS*, vol. 380, pp. 551–570, 2007. [90](#), [108](#)
- [94] M. A. Aragón-Calvo, E. Platen, R. van de Weygaert, and A. S. Szalay, “The Spine of the Cosmic Web,” *Astrophysical Journal*, vol. 723, pp. 364–382, 2010. [90](#), [107](#)

- [95] J. Reininghaus, N. Kotava, D. Günther, J. Kasten, H. Hagen, and I. Hotz, “A scale space based persistence measure for critical points in 2d scalar fields,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2045–2052, 2011. [90](#)
- [96] T. Weinkauff and D. Günther, “Separatrix persistence: extraction of salient edges on surfaces using topological methods,” in *Proc. of Symp. on Geom. Proc.*, 2009, pp. 1519–1528. [90](#)
- [97] W. E. Schaap and R. van de Weygaert, “Continuous fields and discrete samples: reconstruction through delaunay tessellations,” *arXiv*, vol. 363, pp. L29–L32, 2000. [95](#), [106](#), [108](#)
- [98] R. van de Weygaert and W. Schaap, “The Cosmic Web: Geometric Analysis,” in *Data Analysis in Cosmology*, ser. Lecture Notes in Physics, Berlin Springer Verlag, V. J. Martínez, E. Saar, E. Martínez-González, and M.-J. Pons-Bordería, Eds., vol. 665, 2009, pp. 291–413. [95](#), [108](#)
- [99] “Computational Geometry Algorithms Library,” <http://www.cgal.org>. [95](#)
- [100] A. Gyulassy, V. Natarajan, V. Pascucci, and B. Hamann, “Efficient computation of Morse-Smale complexes for three-dimensional scalar functions,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 6, pp. 1440–1447, 2007. [95](#)
- [101] S. Codis, C. Pichon *et al.*, “Connecting the cosmic web to the spin of dark haloes: implications for galaxy formation,” *MNRAS*, vol. 427, pp. 3320–3336, 2012. [96](#)
- [102] P. J. E. Peebles, “The Void Phenomenon,” *Astrophys. J.*, vol. 557, pp. 495–504, 2001. [97](#), [121](#)

- [103] M. A. Aragón-Calvo, R. van de Weygaert, and B. J. T. Jones, “Multiscale phenomenology of the cosmic web,” *MNRAS*, vol. 408, pp. 2163–2187, 2010. [97](#), [114](#), [115](#)
- [104] A. Gyulassy, N. Kotava, M. Kim, C. Hansen, H. Hagen, and P.-T. Bremer, “Direct feature visualization using Morse-Smale complexes,” *IEEE Trans. Vis. & Comp. Graphics*, vol. 18, no. 9, pp. 1549–1562, 2012. [100](#)
- [105] W. Schroeder, K. Martin, and B. Lorensen, *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th Edition*, 4th ed. Kitware, 2006. [100](#)
- [106] A. Maximo, R. Marroquim, and R. Farias, “Hardware-assisted projected tetrahedra,” in *Proceedings of the 12th Eurographics / IEEE - VGTC conference on Visualization*. Eurographics Association, 2010, pp. 903–912. [100](#)
- [107] P. Shirley and A. Tuchman, “A polygonal approximation to direct scalar volume rendering,” *SIGGRAPH Comput. Graph.*, vol. 24, no. 5, pp. 63–70, 1990. [100](#)
- [108] K. Moreland and E. Angel, “A fast high accuracy volume renderer for unstructured data,” in *Proc. of IEEE Symp. on Vol. Vis. and Graphics*, 2004, pp. 9–16. [101](#)
- [109] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, 1975. [101](#)
- [110] R. van de Weygaert and E. van Kampen, “Voids in Gravitational Instability Scenarios - Part One - Global Density and Velocity Fields in an Einstein - De-Sitter Universe,” *MNRAS*, vol. 263, 1993. [102](#), [121](#)
- [111] R. K. Sheth and R. van de Weygaert, “A hierarchy of voids: much ado about nothing,” *MNRAS*, vol. 350, pp. 517–538, 2004. [102](#), [121](#)

- [112] R. v. d. Weygaert, “Voronoi tessellations and the cosmic web: Spatial patterns and clustering across the universe,” in *ISVD '07: Proc. of Symp. on Voronoi Diagrams in Science and Engineering*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 230–239. [102](#), [121](#)
- [113] M. A. Aragon-Calvo and A. S. Szalay, “The hierarchical structure and dynamics of voids,” *MNRAS*, vol. 428, pp. 3409–3424, 2013. [102](#), [121](#)
- [114] R. van de Weygaert and V. Icke, “Fragmenting the universe. II - Voronoi vertices as Abell clusters,” *Astron. Astrophys.*, vol. 213, pp. 1–9, 1989. [102](#)
- [115] R. van de Weygaert, *Voids and the geometry of large scale structure*, 1991, PhD thesis, University of Leiden. [102](#)
- [116] E. Komatsu *et al.*, “Five-year wilkinson microwave anisotropy probe observations: Cosmological interpretation,” *The Astrophysical Journal Supplement Series*, vol. 180, no. 2, 2009. [105](#)
- [117] T. Ishiyama, S. Rieder *et al.*, “The Cosmogrid Simulation: Statistical Properties of Small Dark Matter Halos,” *Astrophysical Journal*, vol. 767, 2013. [105](#)
- [118] Y. B. Zel’dovich, “Gravitational instability: An approximate theory for large density perturbations.” *Astronomy & astrophysics*, vol. 5, pp. 84–89, 1970. [105](#)
- [119] S. Rieder, R. van de Weygaert, M. Cautun, B. Beygu, and S. Portegies Zwart, “Assembly of filamentary void galaxy configurations,” *MNRAS*, vol. 435, pp. 222–241, 2013. [105](#)

- [120] C. De Boni, K. Dolag, S. Ettori, L. Moscardini, V. Pettorino, and C. Baccigalupi, “Hydrodynamical simulations of galaxy clusters in dark energy cosmologies - I. General properties,” *MNRAS*, vol. 415, pp. 2758–2772, 2011. [106](#)
- [121] K. Dolag, S. Borgani, G. Murante, and V. Springel, “Substructures in hydrodynamical cluster simulations,” *MNRAS*, vol. 399, pp. 497–514, 2009. [106](#)
- [122] R. van de Weygaert, K. Kreckel *et al.*, “The Void Galaxy Survey,” in *Environment and the Formation of Galaxies: 30 Years Later*, I. Ferreras and A. Pasquali, Eds., 2011, p. 17. [121](#)
- [123] K. Kreckel, E. Platen *et al.*, “Only the Lonely: H I Imaging of Void Galaxies,” *Astron.J.*, vol. 141, p. 4, 2011. [121](#)
- [124] —, “The Void Galaxy Survey: Optical Properties and H I Morphology and Kinematics,” *Astron.J.*, vol. 144, p. 16, 2012. [121](#)
- [125] A. S. Konagurthu, J. C. Whisstock, P. J. Stuckey, and A. M. Lesk, “MUSTANG: a multiple structural alignment algorithm.” *Proteins*, vol. 64, no. 3, pp. 559–574, 2006. [123](#)
- [126] L. Holm and C. Sander, “Mapping the protein universe,” *Science*, vol. 273, pp. 595–603, 1996. [123](#)
- [127] P. Shealy and H. Valafar, “Multiple structure alignment with msTALI,” *BMC Bioinformatics*, vol. 13, no. 1, pp. 1–17, 2012. [123](#)
- [128] K. Yeturu and N. Chandra, “PocketAlign: A novel algorithm for aligning binding sites in protein structures,” *J. Chem. Info. and Modeling*, vol. 51, no. 7, pp. 1725–1736, 2011. [123](#)

- [129] A. Heifets and R. H. Lilien, “LigAlign: Flexible ligand-based active site alignment and analysis,” *J. Mol. Graphics and Modelling*, vol. 29, no. 1, pp. 93 – 101, 2010. [123](#)
- [130] J. F. Gibrat, T. Madej, and S. H. Bryant, “Surprising similarities in structure comparison.” *Cur. Opinion in Struct. Bio.*, vol. 6, no. 3, pp. 377–385, 1996. [123](#)
- [131] M. Gao and J. Skolnick, “iAlign: a method for the structural comparison of protein-protein interfaces,” *Bioinformatics*, 2010. [123](#)
- [132] S. Angaran, M. E. Bock, C. Garutti, and C. Guerra, “MolLoc: a web tool for the local structural alignment of molecular surfaces.” *Nucleic Acids Research*, vol. 37, pp. W565–70, Jul 2009. [123](#), [132](#)
- [133] J. Konc and D. Janezic, “ProBiS-2012: web server and web services for detection of structurally similar binding sites in proteins,” *Nucleic Acids Research*, vol. 40, 2012. [123](#), [124](#)
- [134] I. Merelli *et al.*, “Image-based surface matching algorithm oriented to structural biology,” *IEEE/ACM Trans. Comp. Biology Bioinform.*, vol. 8, no. 4, pp. 1004–1016, 2011. [123](#), [132](#)
- [135] B. Pang, D. Schlessman *et al.*, “PBSalign: Sequence-independent local alignment of protein binding sites,” *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on*, 2013. [123](#)
- [136] K. Yeturu and N. Chandra, “PocketMatch: a new algorithm to compare binding sites in protein structures.” *BMC Bioinformatics*, vol. 9, no. 1, 2008. [125](#), [132](#), [147](#), [154](#)

- [137] M. P. do Carmo, *Differential geometry of curves and surfaces*. Prentice Hall, 1976. [126](#)
- [138] D. Cohen-Steiner and J. M. Morvan, “Restricted Delaunay triangulations and normal cycle,” in *Symp. Comp. Geom.*, 2003, pp. 312–321. [127](#)
- [139] F. Cazals, F. Chazal, and T. Lewiner, “Molecular shape analysis based upon the morse-smale complex and the connolly function,” in *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, 2003. [128](#)
- [140] V. Natarajan, Y. Wang, P.-T. Bremer, V. Pascucci, and B. Hamann, “Segmenting molecular surfaces,” *Comput. Aided Geom. Des.*, 2006. [128](#)
- [141] M. L. Connolly, “Measurement of protein surface shape by solid angles,” *J. Mol. Graph.*, 1986. [128](#)
- [142] C. Bron and J. Kerbosch, “Algorithm 457: finding all cliques of an undirected graph,” *Commun. ACM*, vol. 16, no. 9, pp. 575–577, 1973. [130](#)
- [143] I. Koch, “Enumerating all connected maximal common subgraphs in two graphs,” *Theoretical Comp. Sci.*, vol. 250, no. 12, pp. 1 – 30, 2001. [130](#)
- [144] F. Cazals and C. Karande, “A note on the problem of reporting maximal cliques,” *Theor. Comput. Sci.*, 2008. [130](#)
- [145] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987. [130](#)
- [146] H. Edelsbrunner, “Deformable smooth surface design,” *Disc. Comp. Geom.*, vol. 21, no. 1, pp. 87–115, 1999. [132](#)

- [147] T. J. Dolinsky, J. E. Nielsen, J. A. Mccammon, and N. A. Baker, “PDB2PQR: an automated pipeline for the setup of PoissonBoltzmann electrostatics calculations,” *Nucleic Acids Res*, vol. 32, pp. 665–667, 2004. [133](#)
- [148] S. Decherchi and W. Rocchia, “A general and robust ray-casting-based algorithm for triangulating surfaces at the nanoscale,” *PLoS ONE*, vol. 8, no. 4, 04 2013. [133](#)
- [149] E. F. Pettersen *et al.*, “UCSF Chimera—a visualization system for exploratory research and analysis.” *J. Comp. Chem.*, vol. 25, no. 13, pp. 1605–1612, 2004. [133](#)
- [150] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983. [134](#)
- [151] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901. [135](#)