1-1-2011

Simplifying The Non-Manifold Topology Of Multi-Partitioning Surface Networks

Trung Nguyen
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

School of Engineering and Applied Science

Department of Computer Science and Engineering

Thesis Examination Committee:
Tao Ju, Chair
Robert Pless
Jeremy Buhler

SIMPLIFYING THE NON-MANIFOLD TOPOLOGY OF MULTI-PARTITIONING SURFACE NETWORKS

by

Trung Duc Nguyen

A thesis presented to the School of Engineering
of Washington University in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

May 2011
Saint Louis, Missouri

ABSTRACT OF THE THESIS

Simplifying the Non-Manifold Topology of Multi-Partitioning Surface Networks

by

Trung Duc Nguyen

Master of Science in Computer Science

Washington University in St. Louis, 2011

Research Advisor: Professor Tao Ju

In bio-medical imaging, multi-partitioning surface networks (MPSNs) are very useful to model complex organs with multiple anatomical regions, such as a mouse brain. However, MPSNs are usually constructed from image data and might contain complex geometric and topological features. There has been much research on reducing the geometric complexity of a general surface (non-manifold or not) and the topological complexity of a closed, manifold surface. But there has been no attempt so far to reduce redundant topological features which are unique to non-manifold surfaces, such as curves and points where multiple sheets of surfaces join.

In this thesis, we design interactive and automated means for removing redundant non-manifold topological features in MPSNs, which is a special class of non-manifold surfaces. The core of our approach is a mesh surgery operator that can effectively simplify the non-manifold topology while preserving the validity of the MPSN. The operator is implemented in an interactive user interface, allowing user-guided simplification of the input. We further develop an automatic algorithm that invokes the operator following a greedy heuristic. The algorithm is based on a novel, abstract representation of a non-manifold surface as a graph, which allows efficient discovery and scoring of possible surgery operations without the need for explicitly performing the surgeries on the mesh geometry.

Acknowledgments

It is a great chance to show my appreciation and gratitude to all the people who have supported me during my studies at Washington University. Without their help I could not have finished this thesis.

First, I would like to show my utmost gratitude to my advisor: Dr. Tao Ju. I really appreciate his patient guidance. He has made his great support available in a number of ways, from teaching me very basic concepts to even debugging programs with me. It is just impossible to recount his help here. I really learned a lot from him.

I would like to thank Dr. Gruia-Catalin Roman for the support he gave to me and the students in the department. Discussions with him were always exciting and encouraging, giving me more confidence and hope. I also would like to thank the faculty and staff in the Department of Computer Science for their continuous help.

I would like to thank Dr. Robert Pless and Dr. Jeremy Buhler for being members of my committee and for their valuable suggestions.

I am also grateful to all the Media and Machines Lab members for making the lab a great working environment. It has been my pleasure to work with professors and students in the lab. I specially thank Stephen Schuh and Lu Liu for their help in many ways at many times.

I would like to thank Ly Phan and other friends who have helped since I came to Saint Louis. I definitely have had a great time with them in this amazing city.

Last but not least, I would like to thank my parents and my younger brother for their love and support. Their love and encouragement always make me stronger.

Trung Duc Nguyen

Washington University in Saint Louis
May 2011

Dedicated to my parents

For your everlasting love, patience and sacrifice

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Advances in medical imaging have presented medical and biology researchers with an unprecedented amount of spatial information, which needs to be visualized, processed, and analyzed. Computational tasks involving medical images often make use of a geometric, surface representation that depicts the 3D boundary of a biological structure. For example, visualizing the surface of a structure of interest in a 3D volume (e.g., MRI or CT) is much more intuitive than directly projecting the entire volume onto the 2D screen. Geometric representations also enable powerful physical simulations, such as simulating fluid flow in a vessel or interaction between tissue and virtual surgery knives. Last but not least, surface models of anatomical structures are often used as templates (called atlases) that can be deformed and matched to the shape of structures in images of different individuals, a process known as registration. Registration allows images of individuals with diverse anatomical shapes to be aligned, classified, and compared.

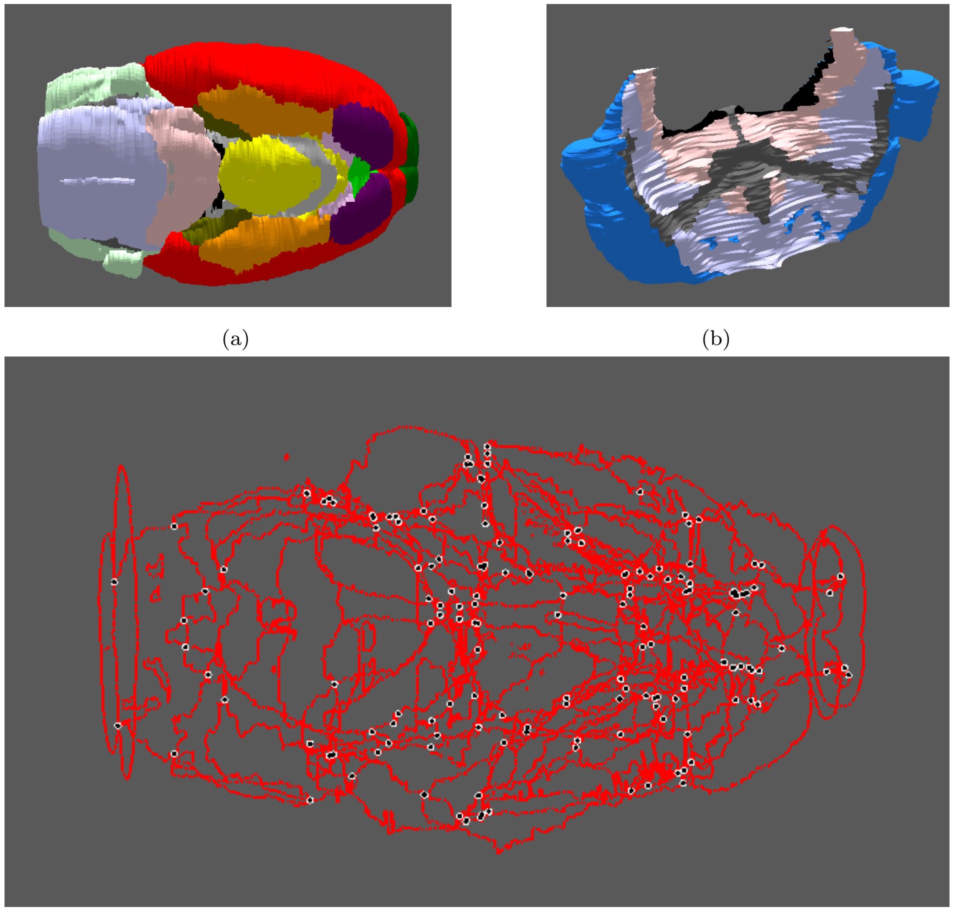
Traditionally, surface models in bio-medicine are limited to closed, manifold surfaces. These surfaces partition the space into disjoint inside (e.g., tissue) and outside (e.g., background) regions, and contain no borders or junctions (where multiple surface pieces join). Surfaces like these are suitable to represent a homogeneous tissue, such as a bone, or a compartment within a complex organ, such as the cortex of the brain. However, they are less suitable to represent a complex organ (e.g., the brain) made up of multiple anatomical regions (e.g., cortex, mid-brain, cerebellum, etc.). Simply juxtaposing multiple manifold surfaces, each representing one subdivision, would easily leave gaps or create intersecting geometry between neighboring anatomical regions.

Alternatively, a complex organ with interior subdivision can be modeled as a non-manifold, multi-partitioning surface network (MPSN), which consists of multiple surface pieces joining together to partition the space into abutting regions. A MPSN avoids the issues (e.g., gaps and intersections) in combining multiple manifold surfaces, since two abutting regions share a single, common piece of surface. Figure 1.1 demonstrates an example of a MPSN representing the mouse brain, which consists of 16 distinct anatomical regions. The picture in (a) shows the exterior surface of the brain, where each surface piece is colored by the anatomical region it bounds. The picture in (b) shows a single anatomical region (the cerebellum), where the surface pieces are colored by the neighboring regions abutting the cerebellum.

1.2 Problem Statement and Contributions

Recent research has seen an increasing number of methods for constructing non-manifold surfaces, particularly MPSNs from image data (see more detailed review in the next section). For example, the MPSN in Figure 1.1 is produced by one of these methods from a stack of labelled tissue images [8]. Unfortunately, the resulting geometry is often too complex for downstream applications. The complexity of a general surface (manifold or not) can be appreciated from two aspects:

- *Geometric complexity:* A surface is typically represented discretely as a triangulated mesh. Geometric complexity refers to the number of vertices and triangles in the triangulation.
- *Topological complexity:* Topological complexity refers to the number of topological features, which are independent of how the surface is tessellated and also invariant under continuous deformation of the geometry. On a manifold surface, typical topological features are the connected surface components and the handles (e.g., a torus has one handle while a sphere has none). A non-manifold



(a) (b) (c)

Figure 1.1: An example of a MPSN prepresenting a mouse brain model: (a) exterior surface of the mouse brain, with each surface colored by the anatomical region it bounds. (b) a single anatomical region in the mouse brain (the cerebellum region). (c) network of non-manifold curves (colored red) and set of non-manifold points (highlighted) in the MPSN.

surface introduces additional topological features where three or more pieces of surfaces join, either along a curve or at a point.

A geometrically complex surface is computationally expensive to visualize or process. On the other hand, a significant amount of topological complexity may prevent the geometry from being simplified to a desirable degree, as the topological features persist during geometric coarsening. In addition, an incorrect set of topological features is detrimental to many computations, such as simulation of fluid flows within the surface.

There has been much research in the geometry processing community on methods for reducing the geometric complexity of a general surface (manifold or not). Recent research also presents algorithms for simplifying the topology of a manifold surface (see more detailed reviews in the next section). However, there has been no work so far on taming the topological features that are unique to non-manifold surfaces. These non-manifold features can be numerous. Figure 1.1(c) highlights the curves and points where multiple surface pieces join in the mouse brain example. Note that many of them are caused by errors during mesh creation rather than reflecting the physical reality.

In this thesis, we design interactive and automated means for removing redundant non-manifold topological features in MPSNs, which are a special class of non-manifold surfaces. The core of our approach is a mesh surgery operator that can effectively simplify the non-manifold topology while preserving the validity of the MPSN (Chapter 2). The operator is implemented in an interactive user interface, allowing user-guided simplification of the input (Chapter 3). We further develop an automatic algorithm that invokes the operator following a greedy heuristic (Chapter 4). The algorithm is based on a novel, abstract representation of a non-manifold surface as a graph, which allows efficient discovery and scoring of possible surgery operations without the need for explicitly performing the surgeries on the mesh geometry.

1.3 Related Work

Creating MPSNs: Recent years have seen several methods capable of creating non-manifold surface representations, particularly MPSNs, from real-world data. When the input is a 3D scalar volume whose voxels have been assigned with two or more labels, an MPSN can be constructed, separating voxels with different labels using iso-surfacing techniques [7, 3, 12]. A different type of inputs consists of a set of planes (parallel or oblique), each containing a multi-partitioning curve network that describes the cross-section geometry of the subject, which can be connected to form MPSNs [8, 10, 1].

Geometry simplification: Simplifying the geometric complexity of a surface has been well-studied in the literature for a few decades. Popular schemes include edge-collapsing [4] and vertex clustering [6]. In particular, the edge-collapsing scheme can be applied to both manifold and non-manifold surfaces [14]. Topological constraints during edge-collapsing have also been studied [2], ensuring that the topological structure of the surface is preserved after geometric simplification.

Topology simplification: Numerous methods have been proposed in the literature for simplifying the topology of a closed, manifold surface in 3D. Typically, topological redundancies on such surfaces appear as small surface handles or tunnels. Such handles can be removed by a variety of means such as morphological operations [11], spatial carving [13], cutting and gluing [5], and graph editing [15, 9]. However, we are not aware of any methods that deal with redundant non-manifold topological features on a non-manifold surface.

Chapter 2

Mesh Surgery for Topology Simplification

As mentioned earlier, the goal of our work is to reduce the topological complexity in a non-manifold surface network, and particularly the non-manifold features where multiple surface pieces join. We will first introduce some notions that will facilitate our discussion. We will then introduce the core of our approach, a mesh surgery operator that locally reduces the number of non-manifold features. This operator will be used in two different settings in the next two chapters.

2.1 Notions and Definitions

2.1.1 Manifold Topology

We start by introducing several topological concepts for triangulated surfaces in 3D. A surface is said to be manifold at a point if the local neighborhood of the point on the surface is topologically equivalent to a 2D disk. By this definition, a triangulated surface is always manifold at any interior points on a triangle. We can classify the local topology of an edge or vertex as follows:

- **Edge:** an edge is said to be manifold if it is shared by 2 triangles. Otherwise, the edge is non-manifold.

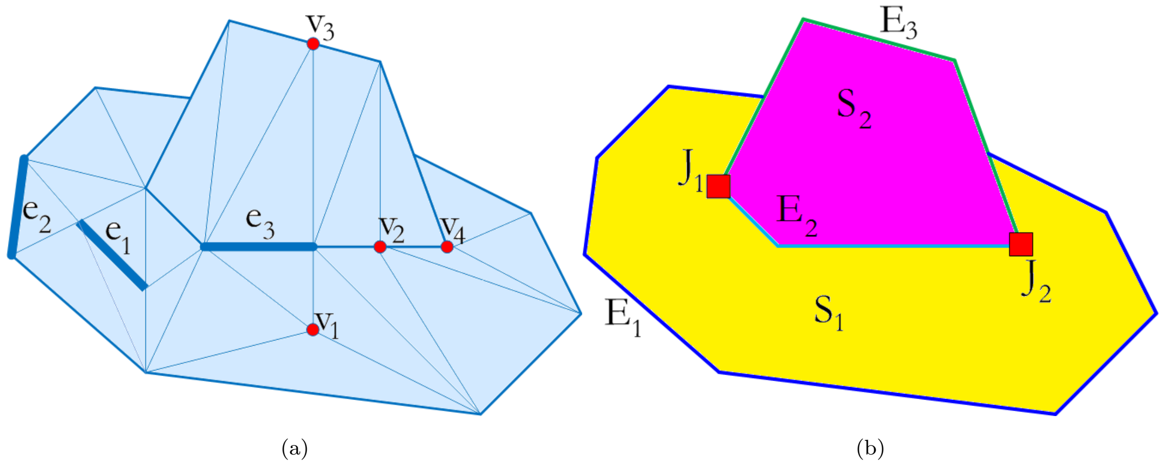


Figure 2.1: Example of manifold topology classification and topological features: (a) different types of edges and vertices: $e1$ is a manifold edge while $e2$ and $e3$ are non-manifold edges; $v1$ is a manifold vertex, $v2$ and $v3$ are non-manifold vertices of type-I and $v4$ is a non-manifold vertex of type-II. (b)different topological features: $S1$ and $S2$ are sheets; $E1$, $E2$ and $E3$ are seams; $J1$ and $J2$ are joints.

- **Vertex:** a vertex is said to be manifold if it is shared by a ring of triangles that forms a topological 2-disk (such as $v1$ in Figure 2.1(a)). Otherwise, the vertex is non-manifold, and can be further classified into two types:
  - A non-manifold vertex is of type-I if it is shared by triangles that form one, three or more fans spanning two edges. The local topology of a type-I vertex is equivalent to a collection of half-disks around an axis (such as $v2, v3$ in Figure 2.1(a)).
  - Otherwise, the non-manifold vertex is of type-II (such as $v4$ in Figure 2.1(a)).

These notions are illustrated in Figure 2.1(a) using a simple triangulation. Here, edge e1 is manifold while edges $e2, e3$ are non-manifold. Vertex $v1$ is a manifold vertex, $v2$ and $v3$ are type-I non-manifold vertices, and $v4$ is a type-II non-manifold vertex.

2.1.2 Topological Features

Topological features on a surface are independent of the tessellation level and invariant under continuous deformation. They can be defined by maximally grouping neighboring geometric elements sharing the same local topology. In this work, we are mostly interested in three types of features:

- A *sheet* is a maximally connected set of triangles (e.g., one that cannot be further expanded). Here, two triangles are said to be connected if they share a common, manifold edge. A sheet represents a manifold portion of the surface.
- A *seam* is a maximally connected set of non-manifold edges. Here, two non-manifold edges are said to be connected if they share a common, type-I non-manifold vertex. A seam lies at the border of a sheet, or the junction of more than two sheets.
- A *joint* is a type-II non-manifold vertex. A joint lies at the end of a seam, or the junction of multiple seams, or the junction of multiple sheets (but away from the seams).

These structures are illustrated in Figure 2.1(b), which contains two sheets ($S1, S2$), three seams ($E1, E2, E3$), and two joints ($J1, J2$). Note that a sheet may be bounded by one or more seams or joints, or not bounded by any seam or joint (e.g., a sphere). A seam can be bounded by two joints (e.g., a curve segment), one joint (e.g., a curve segment with identical ends), or no joint (e.g., a curve loop).

2.1.3 Multi-Partitioning Surface Networks

A multi-partitioning surface network (MPSN) is a triangulated surface that partitions the 3D space into disjoint regions (e.g., air, cortex, cerebellum, etc.). A MPSN is valid if the collection of triangles bounding each region forms a closed surface without holes or boundaries, and every triangle lies on the interface of two different regions. Intuitively, a valid MPSN captures the interface between multiple solid objects with different materials.

In a valid MPSN, a sheet represents a contiguous interface between two neighboring regions, while a seam or joint typically indicates the junction of more than two regions. Figure 1.1(c) visualizes all seams and joints in the MPSN in Figure 1.1(a).

2.2 Detachment Operator

Our aim is to reduce the number of topological features (e.g., sheets, seams, and joints) presented in an input MPSN. Note that it is important to preserve the validity of the MPSN during our operations, so that the MPSN still represents a physically plausible partitioning of a solid object. In addition, we would like to make as little change as possible to the geometry of the surface during topology simplification. We next present a mesh surgery operator that fulfills these goals and constraints.

2.2.1 Observation

We start with an intuitive observation that motivates our approach. As mentioned earlier, the topological features (i.e., sheets, seams and joints) lie at the interface of two or more neighboring regions. Intuitively, if these regions are detached so that they are no longer abutting each other, then the corresponding topological features would disappear.

The intuition is illustrated in the examples in Figure 2.2. Figure 2.2(a) shows the sheets bounding a single anatomical region (say A) in the mouse brain (again, each sheet is colored by the abutting anatomical region). The insert shows a small (bright blue) sheet amidst a large (light blue) sheet. Let us call the abutting region on the small sheet B, and the abutting region on the large sheet C. The small interface between A and B in fact is an artifact caused by the surface reconstruction algorithm, and the two anatomical regions do not have a physical interface at that location. To remove the small sheet (along with seam that bounds it), one solution would be to detach regions A and B and to fill in the "vacant" space left behind by region C. Figure 2.2(b) shows a slightly more complex situation, in which the small (gray) sheet that needs to be removed is surrounded by two large sheets (green and pink).

Let us call the region bounded by these sheets A, and the regions abutting the gray, green and pink sheets B, C, and D respectively. Similar to the previous example, we can remove the small gray sheet by detaching regions A and B and by filling the vacancy using either region C or region D.

While intuitive to understand, the "detaching" operation is not that easy to implement. In the simpler example of Figure 2.2(a), detachment can be achieved by first pulling off the sheet between regions B and C (not shown in the picture) away from the seam that bounds the small sheet in question, then filling the hole left on the pulled sheet with a patch of triangles. However, such "pulling" and "filling" are not straightforward for the more complex example in Figure 2.2(b), in which there are more sheets involved (e.g., between B and C, between B and D, and between C and D).

In the following, we describe a robust way to perform detachment of neighboring regions, which works in arbitrarily complex configurations and maintains the validity of the MPSN. The operator applies not only to regions sharing a common sheet, but also to regions meeting only at a common seam or a joint. We will first explain the operator in 2D, which is easier to understand. We will then discuss the full implementation in 3D.

2.2.2 Detachment in 2D

We first consider a 2D scenario, in which the input is a multi-partitioning curve network (MPCN) made up of a network of polylines that partition the plane into disjoint, labelled regions. There are two kinds of topological features of a MPCN that we are interested in (in contrast to three in 3D):

- A *segment* is a maximal component of edges connected by manifold vertices (e.g., those shared by exactly 2 edges). A segment represents an un-branched portion of the curve network.
- A *joint* is a non-manifold vertex, and lies either at the end of a segment or junction of multiple segments.

As an example, the MPCN in Figure 2.3(a1) contains 4 segments and 2 joints. As in 3D, these topological features lie at the interface of multiple regions, which we will attempt to remove by detaching the involved regions.

The input to our detachment operator is a triple $< I, D, F >$, where I is a topological feature, either a segment or a joint (called the detached feature), D is a region to be detached from I (called the detached region), and F is a region that will be used to fill the vacancy space left out by the detaching action (called the fill region). The operator proceeds in two steps, which we illustrate using the simple example of Figure 2.3(a1):

1. **Step 1 (Splitting):** Divide region D into two sub-regions, $D1$ and $D2$, by adding a new polyline within D, such that $D1$ completely contains the feature I (shown in Figure 2.3(a2)).

2. **Step 2 (Merging):** Merge sub-region $D1$ with region F by removing the edges lying on the interface of F and $D1$ (shown in Figure 2.3(a3)).

Note that the operator maintains a valid MPCN, since the result of each step is a partitioning of the plane into disjoint regions. The operator can be applied to the complex scenario of abutting regions (see Figure 2.3(b1-b3)), and the detached feature (I) can be either a segment or joint (see Figure 2.3(c1-c3)).

To introduce minimal geometric changes during detachment, the new polyline added in the splitting step is chosen to lie close to the detached feature (I), which tends to minimize the total length of edges added during splitting and those removed during merging. Such choice also minimizes the area in the detached region (D) that gets filled by the fill region (F).

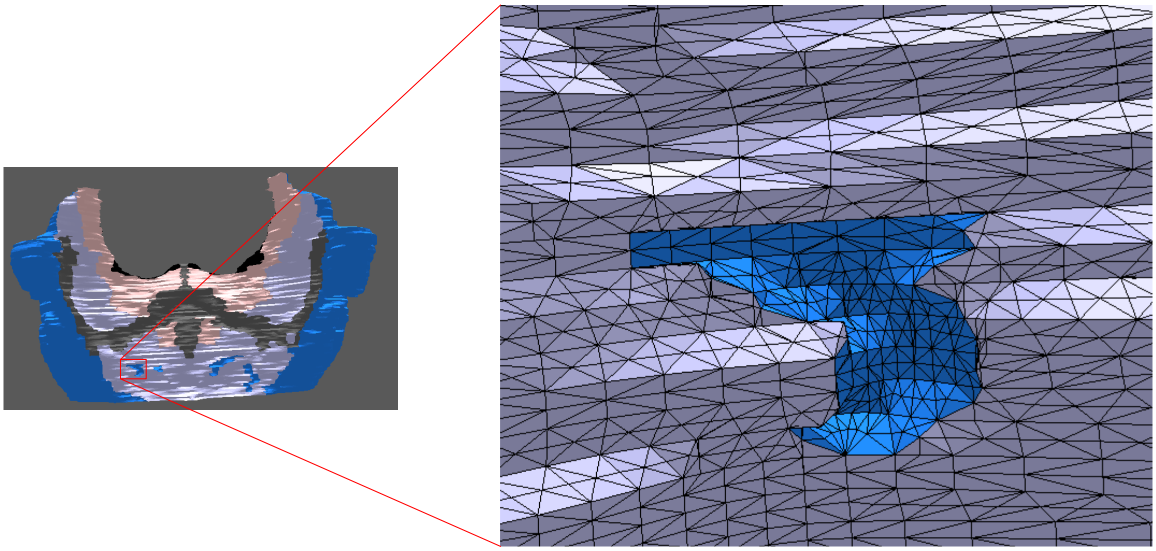
It is important to note that not all detachment operations reduce the number of topological features in the MPCN. For example, the number of segments and joints is unchanged after detachment in Figure 2.3(b). We are interested in operations that do result in a reduction of topological complexity (e.g., Figure 2.3(a,c)). The question of what detachment operations to perform given an input MPCN (or MPSN) will be addressed in the next two chapters.

2.2.3 Detachment in 3D

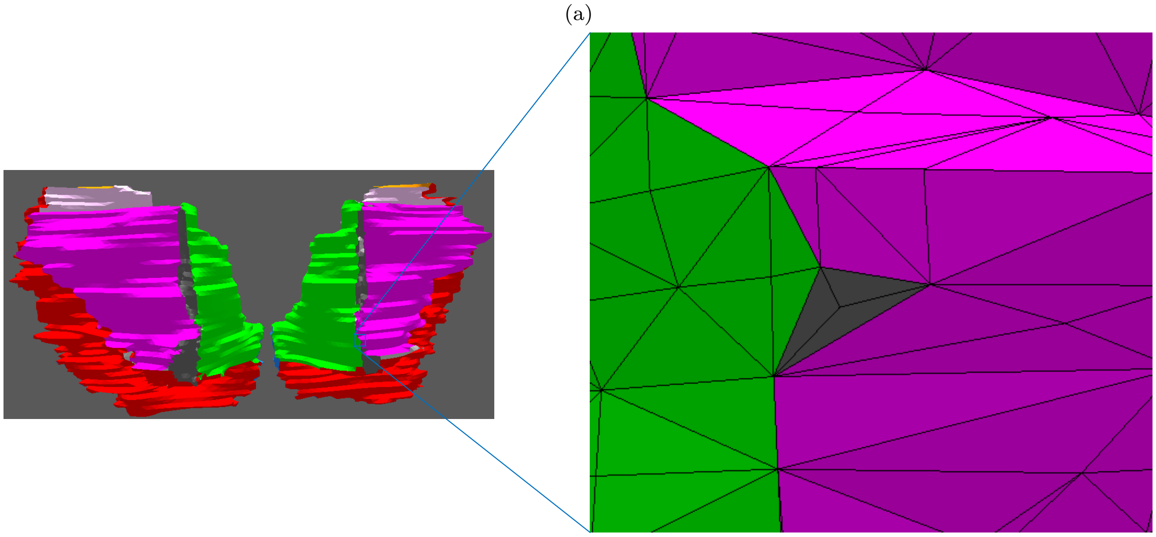
The detachment operator on a 3D MPSN is a straight forward extension of the one on a 2D MPCN. The input to the operator is a triple $< I, D, F >$, where the detached feature I can be a sheet, seam or joint, the detached region D is to be separated from I, and the fill region F will fill the vacancy left behind. The operator proceeds in two steps:

1. **Step 1 (Splitting):** Divide region D into two sub-regions, $D1$ and $D2$, by adding a new triangular surface within D, such that $D1$ completely contains the feature I.
2. **Step 2 (Merging):** Merge sub-region $D1$ with region F by removing triangles lying on the interface of F and $D1$.

The operator maintains a valid MPSN since the result of each step is a partitioning of the space into disjoint regions. Similar to 2D, we minimize the geometric change to the MPSN by adding a triangular surface close to the detached feature (I) in the splitting step. Figure 2.4 illustrates the operator in three examples in which the detached feature (I) is a sheet, seam or joint.



(a)



(b)

Figure 2.2: Examples of detaching MPSNs: (a) Sheets bounding a single anatomical region A in the mouse brain: the blue one is the interface between regions A and B and the light blue one is the interface between regions A and C. (b) More complex situation on a single anatomical region A in a mouse brain: gray, green and pink sheets are interfaces between region A and regions B, C, and D respectively.

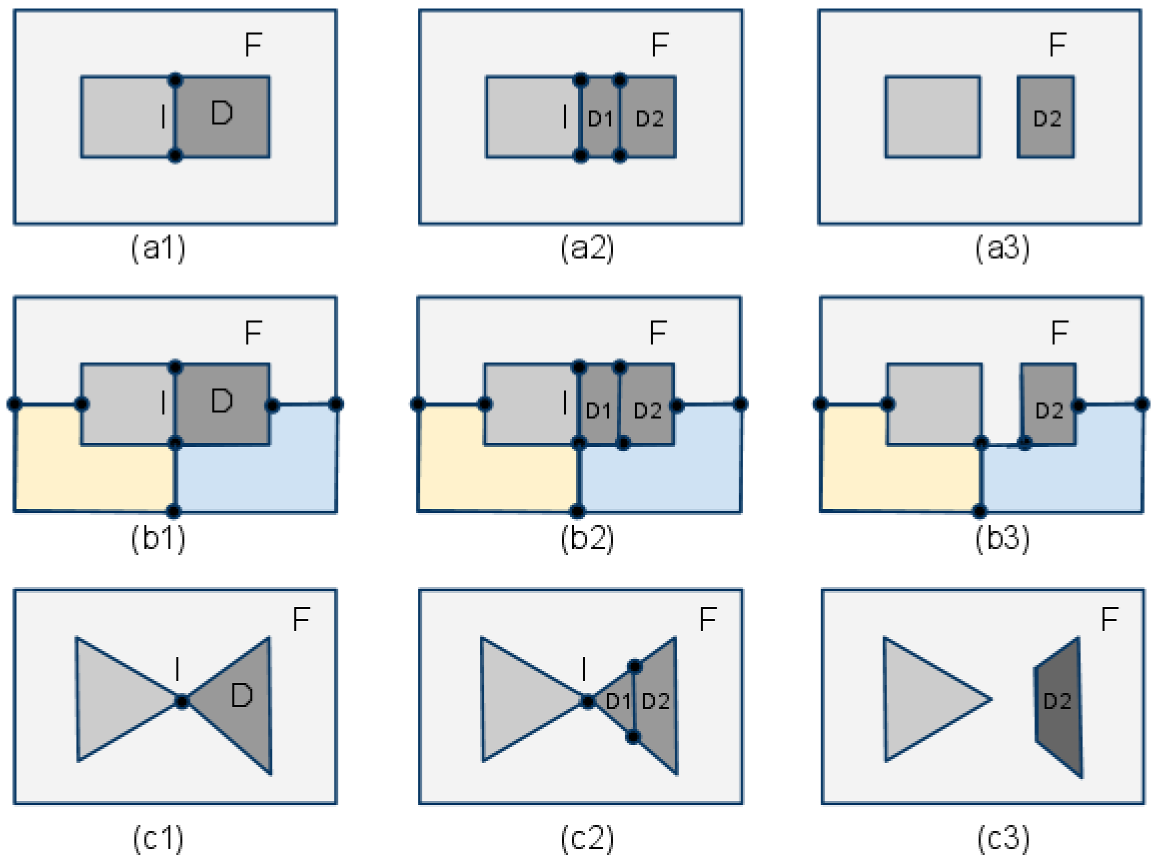
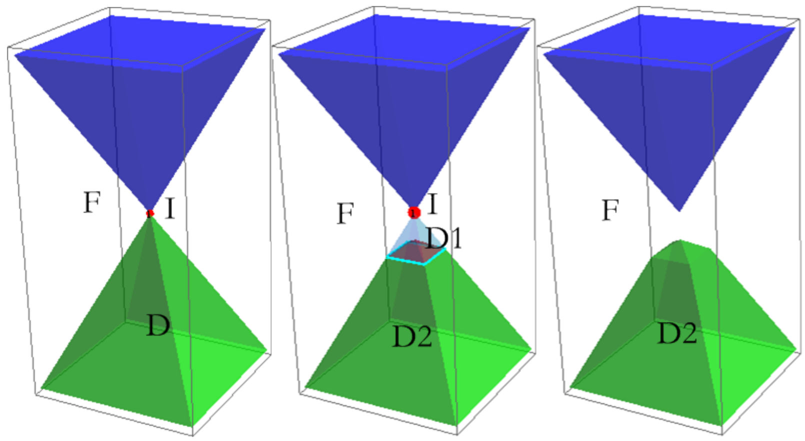
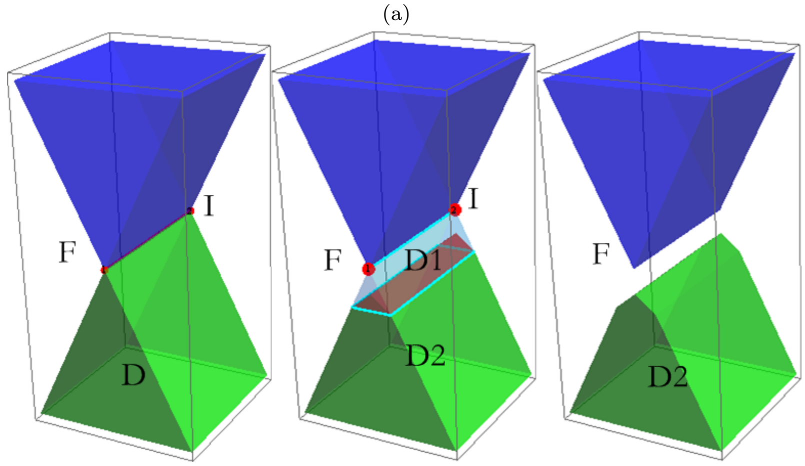


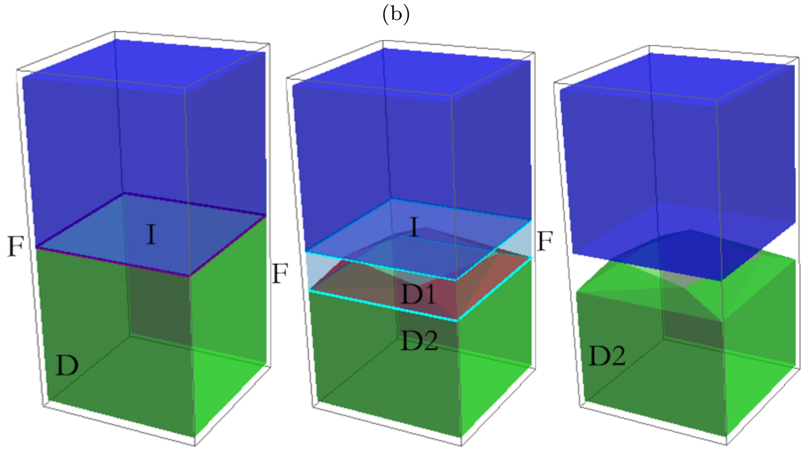
Figure 2.3: Detachment examples in 2D MPCNs (one per row). In each example, the first picture shows an input MPCN (segments and joints are highlighted, and regions are colored) with a detached feature I, a detached region D and a fill region F. The second and third pictures show results after the splitting and merging steps during the detachment.



(a)



(b)



(c)

Figure 2.4: Detachment examples in 3D MPSNs (one per row). In each example, the first picture shows an input MPSN (seams and joins are highlighted, and sheets are colored) with a detached feature I (the joint, seam or sheet in the middle), a detached region D (colored green) and a fill region F. The second and third pictures show results after the splitting and merging steps during the detachment.

Chapter 3

Interactive Topology Simplification

The previous chapter describes a local detachment operator on a MPSN that can reduce its topological complexity (in terms of number of topological features) without destroying its validity. The operator can generally be applied to any triple of detached feature, detached region and fill region. For a given MPSN, there are many possible choices of such triple. However, as we have observed earlier (Section 2.2.2), not all such choices would lead to a reduction in topology. Also, we would like to remove only redundant topological features, which are introduced by the surface construction process and does not reflect physical reality.

To perform detachment operations that would reduce redundant topological features, we develop an interactive environment where the user can decide which topology features are redundant and the system automatically removes those features. In the following sections, we first describe the user interface, and then present a couple of examples.

3.1 An Interactive Tool

The user interface of our interactive tool, shown in Figure 3.1, consists of four main windows as follows:

- **Main Window** is the largest part of the user interface, namely Atlas Builder. This window displays the 3D geometry of any loaded meshes. Additionally, the main menu is located at the top left of this main window.

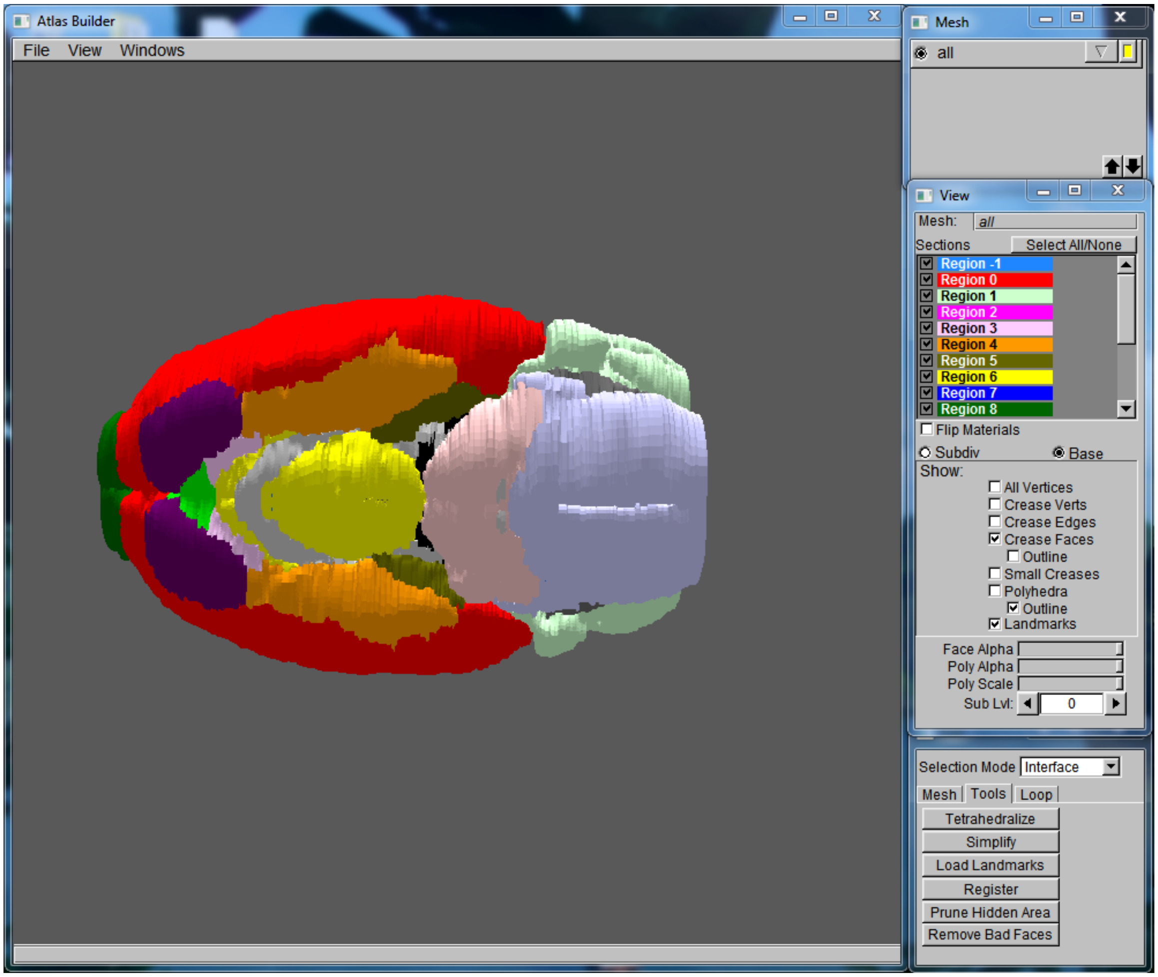
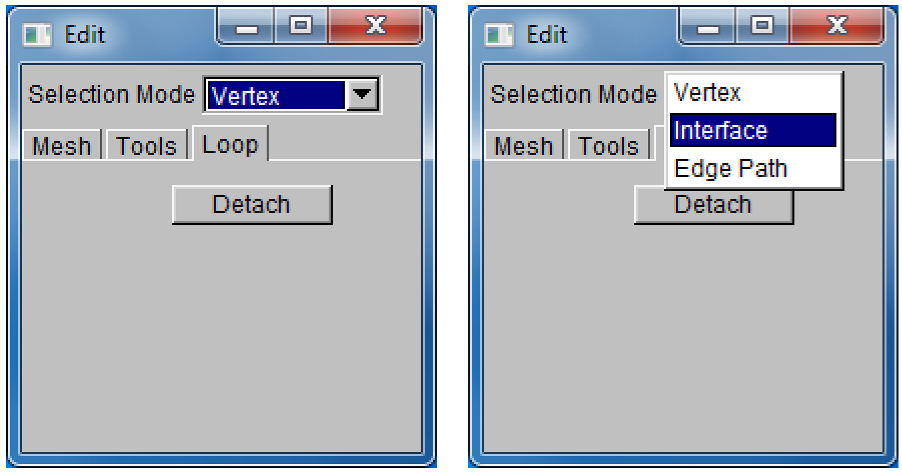


Figure 3.1: The user interface of the Interactive topology simplification tool.

- **Mesh Window** displays a list of currently loaded meshes and allows the user to select a working mesh.
- **View Window** allows the user to control how the selected mesh is visualized via options, such as: selecting regions or topological features to display, or highlighting interesting topological features.
- **Edit Window** contains controls for mesh surgery.

Figure 3.2(a) shows the interactive topology simplification function provided in the *Loop* tab of the *Edit* window. A detachment operation can be done in three steps:



(a) (b)

Figure 3.2: Topology simplification function user interface

1. **Select a topological feature:** In Figure 3.2(b), a drop list named "Selection Mode" allows the user to select topological feature to be simplified. Specifically, the user can use the "Interface" option to select a sheet, "Edge Path" to select a seam, or the "Vertex" option to select a joint.

2. **Choose parameters for detachment operator:** Once a topological feature has been selected, the user can click the button "Detach". A diaglog named "Detach Interface" will then appear to allow the user to select a triple $< I, D, F >$ for the detachment operator: topological feature I is the topological feature selected in the previous step, detached region D is the one selected in the *Cut* list and fill region F is the one selected in the *Fill* list.

3. **Execute detachment:** After selecting the detachment triple, the user can click the "OK" button to execute or click the "Cancel" button to cancel the operation.

An example of a detachment process done by our interactive tool is shown in Figure 3.3, in which we demonstrate how to remove the blue sheet in Figure 2.2(a).

We observe that users might have difficulty finding undesired topological features, especially when these features have small sizes. Our interactive tool assists users by providing a way to let users highlight the features they want to see. Users can choose to assess them either by their sizes or by their complexities:

- **By size:** Typically small sheets and seams are unwanted. This option is shown in Figure 3.4(a). Users can specify a threshold for size of each type of topological feature. All sheets with fewer faces than the sheet threshold and all seams with fewer edges than the seam threshold will be highlighted. On the opposite side, joints with higher degree, i.e. having more seams coming in, are usually unwanted. All joints with degree larger than the joint threshold are highlighted.
- **By complexity:** Seams and joints can also be evaluated by their degree: seams with degree different than 3 and joints with degree different from 3 will be highlighted. This option is enabled by by checking the checkbox in Complexity option (Figure 3.4(a)).

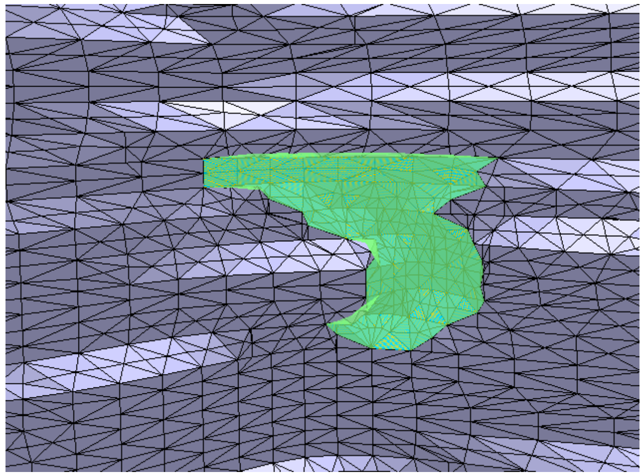
Figure 3.4(b) shows highlighted sheets, seams and joints in the cerebelum region of the mouse brain model.

3.2 Examples

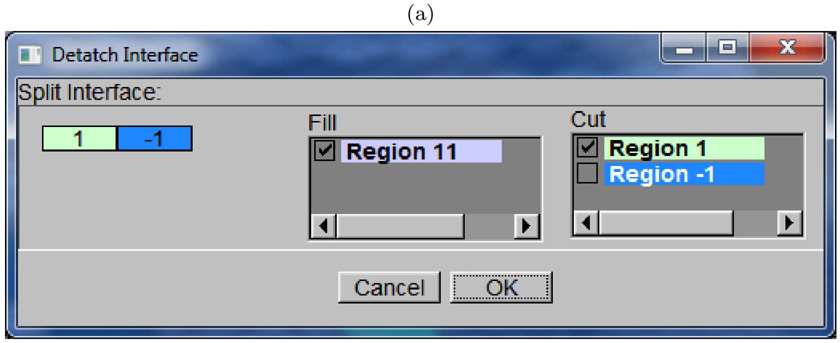
In this section, we provide an example to demostrate the ability of our interactive tool to simplify the topology of a MPSN containing three anatomical regions (cerebellum, medulla, and ventricles) of the mouse brain model. We will first show outputs after one or two individual detachments, and then show the output after completely fixing all regions. Next, we will simplify both the input and output MPSNs down to some certain small sizes (i.e. number of faces in the models) to compare them by using an edge-collapsing mesh simplification tool (QSLIM [4]) with topological preservation extension [2].

Figure 3.5 shows the process of fixing the input MPSN. We start with the original MPSN input in Figure 3.5(a). Its network of non-manifold curves and set of non-manifold points are displayed in Figure 3.5(b). Users use the topological features highlights function to get help finding interesting features (Figure 3.5(c)). Users can then select sheets, seams or joints to simplify the model (Figure3.5(d,e)). After fixing topological problems, we get the final result in Figure 3.5(f). The final output's network of non-manifold curves and set of non-manifold points are much simpler. Finally, we simplify both input and output MPSNs down to different levels: 1000 faces and 500 faces in order to examine and compare them. The comparisons, shown in Figure 3.6 and Figure 3.7, confirm that the topology-fixed MPSN produces much better simplification versions.

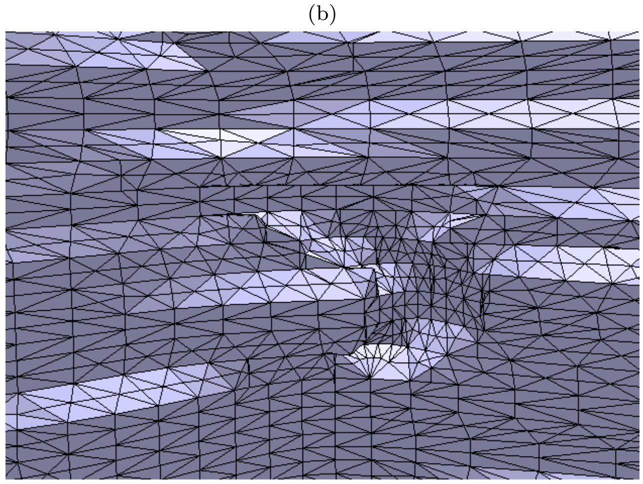
In our example, the output MPSNs have simpler networks of non-manifold curves and smaller sets of non-manifold points, allowing them to have better mesh simplification results.



(a)

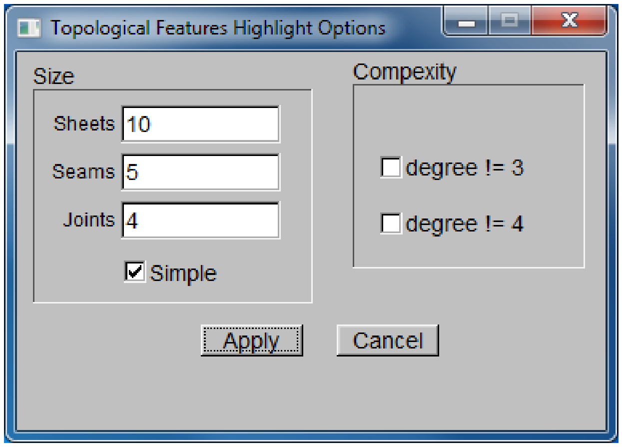


(b)

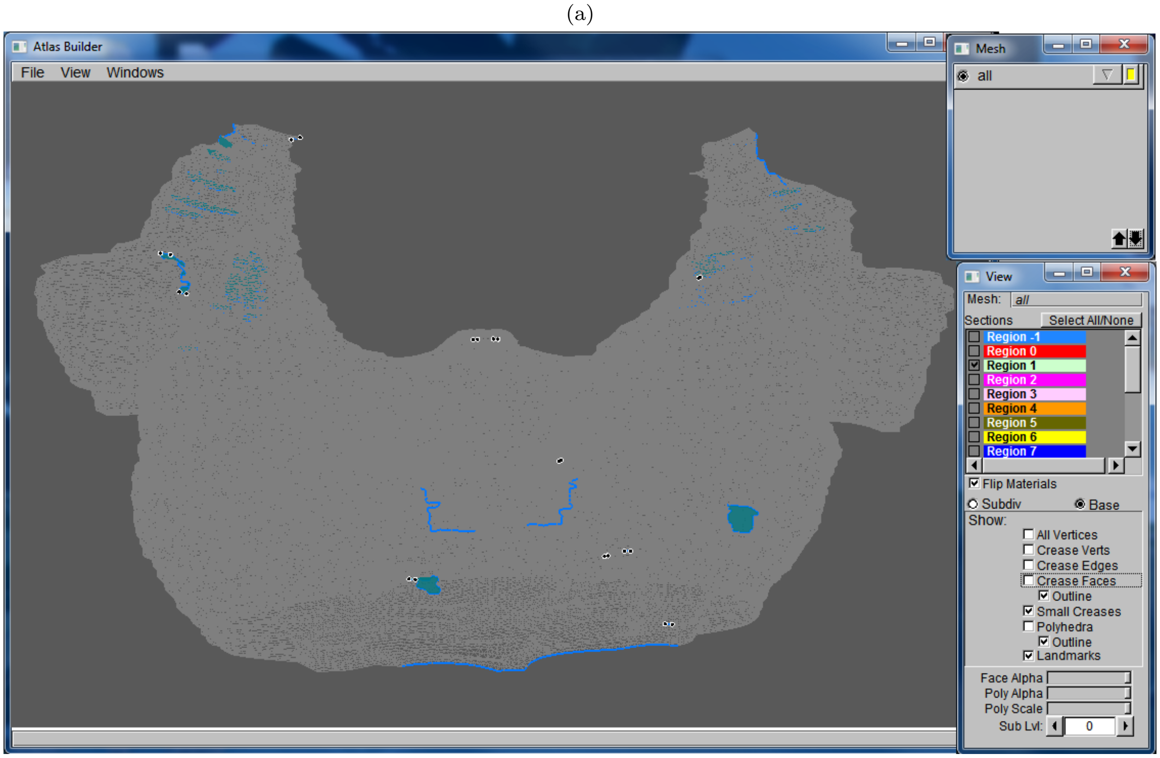


(c)

Figure 3.3: Detachment process using the Interactive topology simplification tool: how to remove the redundant blue sheet in Figure 2.2(a). (a) Step 1: a topological feature, a sheet, is selected. (b) Step 2 and step 3: A detachment triple is decided (I is the selected sheet, D is region -1 and F is region 11) and the button "OK" is pressed to proceed the detachement. (c) The result of the detachment operation: the sheet between regions 1 and -1 no longer exists.



(a)



(b)

Figure 3.4: Topological features highlights in the Interactive topology simplification tool: (a) highlight options (b) highlighted topological features

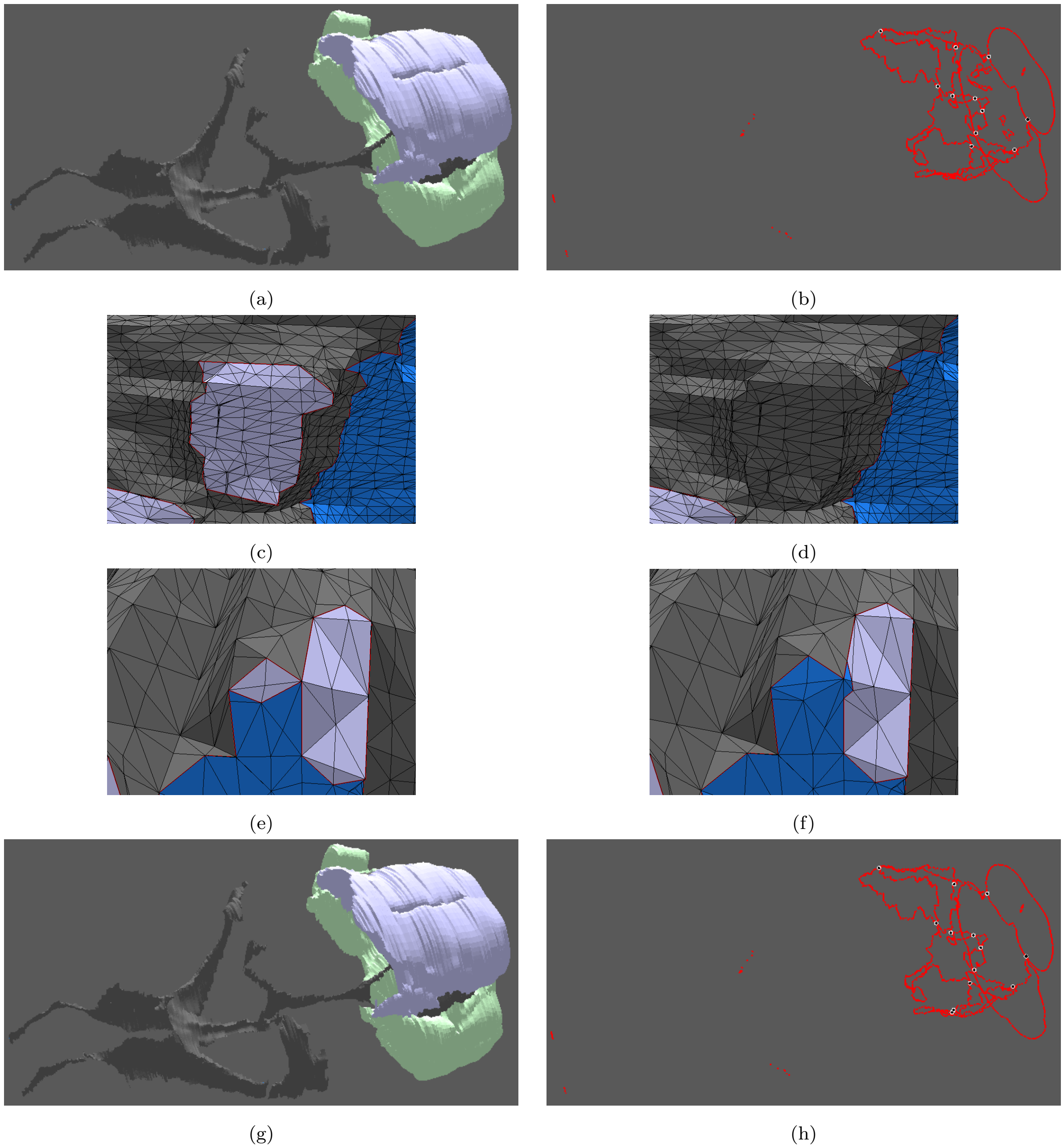
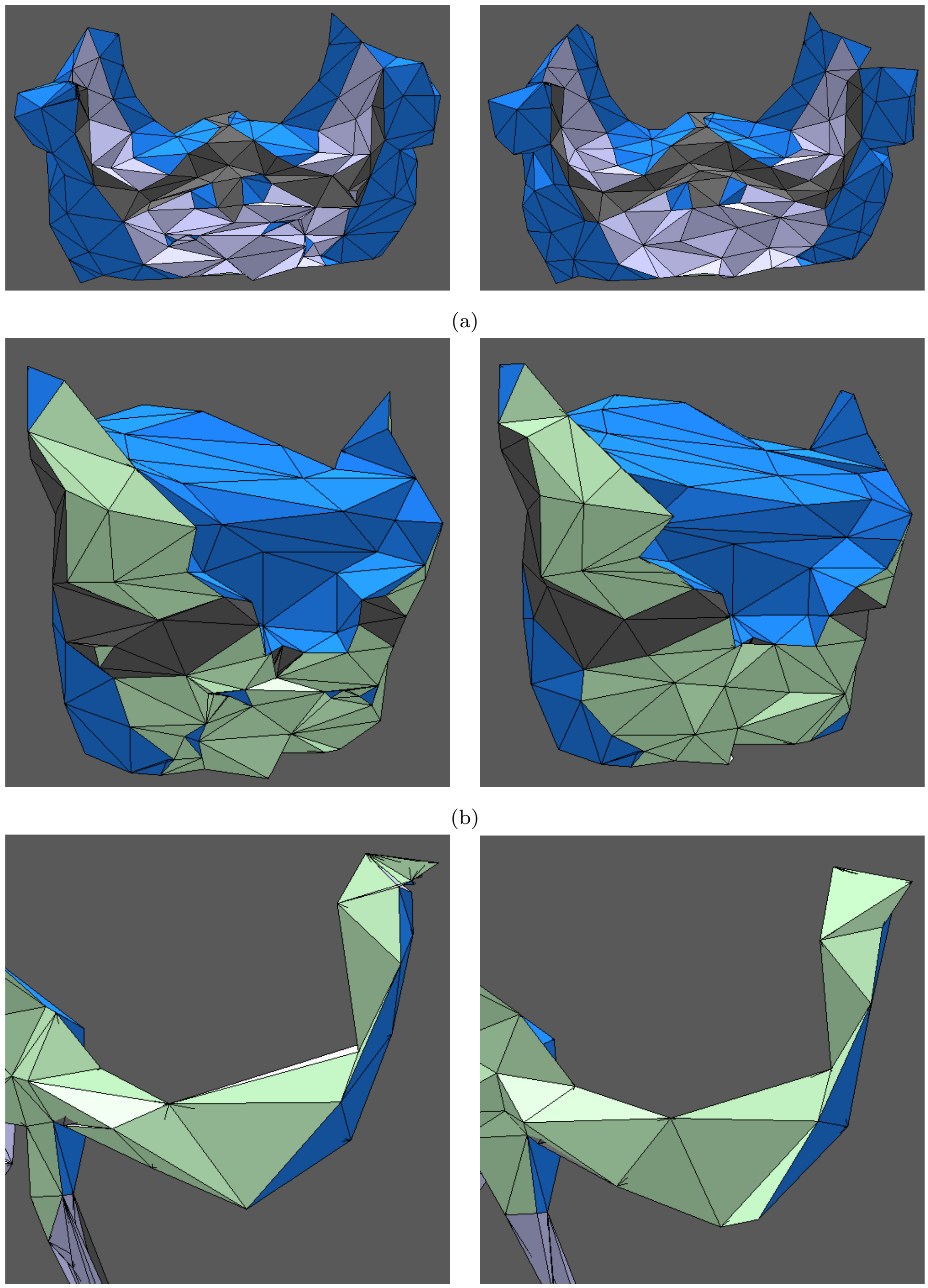


Figure 3.5: Example of the detachment process on the mouse brain model using our interactive tool: (a) the MPSN input containing three anatomical regions (cerebellum, medulla, and ventricles) of the mouse brain. (b) network of non-manifold curves and set of non-manifold points of the input MPSN. (c) one undesired sheet (light blue color). (d) the surface after removing the undesired sheet in (c). (e) an undesired sheet (in the middle, light blue color, containing four faces). (f) the surface after removing the undesired sheet in (e). (g) the final MPSN output. (h) non-manifold structures in the MPSN output, which are much simpler than those of the original MPSN input.

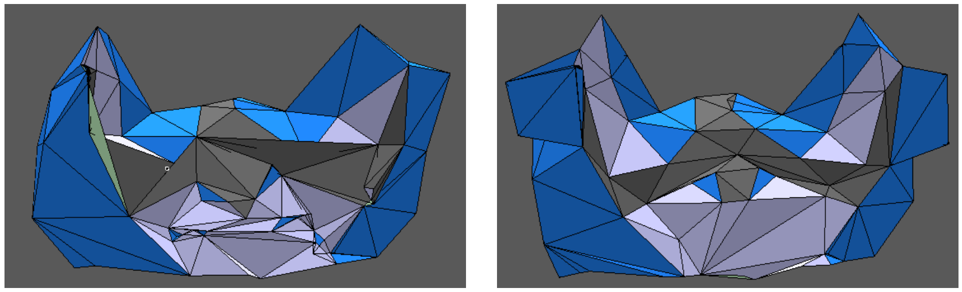


(a)

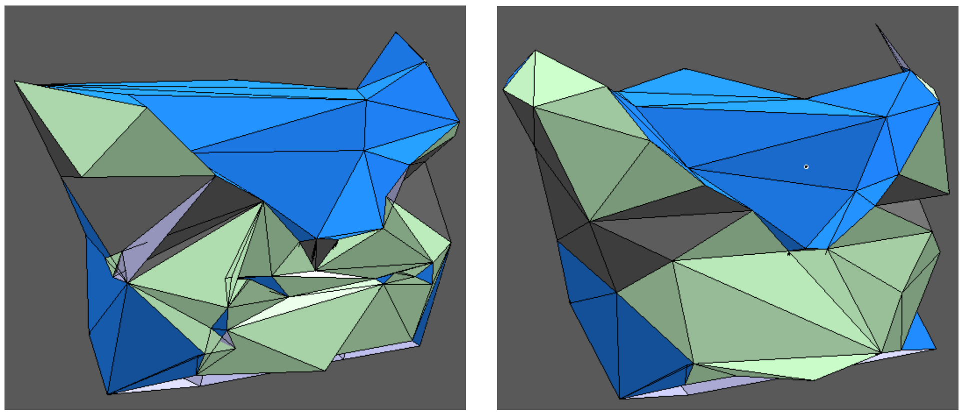
(b)

(c)

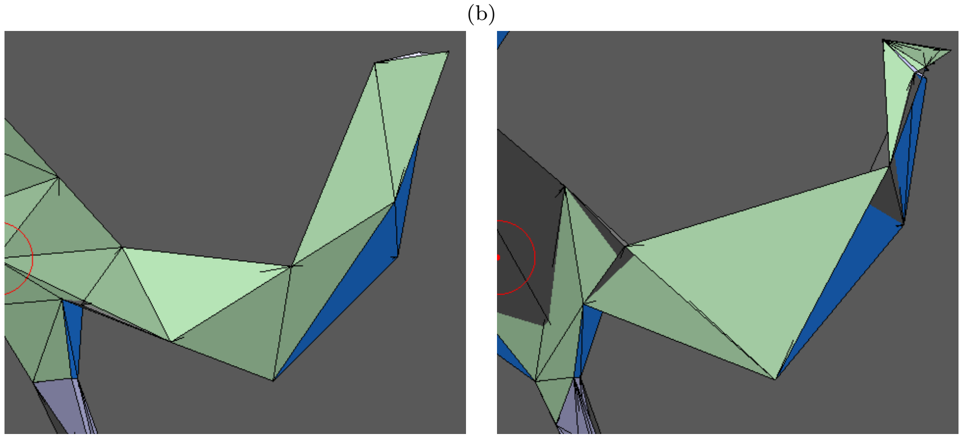
Figure 3.6: Comparison between the simplified versions of size of 1000 faces of the original and the output MPSNs in Figure 3.5. In this figure, the images on the left show parts of the simplified version of the original MPSN and the images on the right show parts of the simplified version of the original MPSN. (a) one spot on the cerebellum region. (b) one spot on the medulla region. (c) one spot on the ventricles region.



(a)



(b)



(c)

Figure 3.7: Comparison between the simplified versions of size of 500 faces of the original and the output MPSNs in Figure 3.5. In this figure, the images on the left show parts of the simplified version of the original MPSN and the images on the right show parts of the simplified version of the original MPSN. (a) one spot on the cerebellum region. (b) one spot on the medulla region. (c) one spot on the ventricles region.

Chapter 4

Automated Topology Simplification

While the interactive tool ensures removal of only redundant topological features, it can be very time-consuming to manually process an entire model, such as the mouse brain example shown in Figure 1.1. In this chapter, we present an automated algorithm that invokes the detachment operator guided by both topological and geometric objectives.

4.1 Overview

Since our goal is to simplify the topology of a MPSN, our preference among possible detachment operations would be ones that yield the most reduction in topological complexity (e.g., number of topological features). Also, as we have observed in the previous chapter, redundant topological features are typically small in size (e.g., area of the sheet, or length of the seam). One can imagine that a greedy, iterative solution would, at each iteration, identify and perform the detachment operation whose detached feature has the least size and whose execution would best simplify the topology.

The iterative algorithm requires exploring a potentially large number of detachment operations at each iteration before executing the selected one. However, the exploration phase can be quite costly using the detachment operator we discussed above, which involves traversing and modification of mesh geometry around each topological feature in the MPSN. In addition, as the geometric complexity of the mesh increases, the time complexity of each detachment operation would increase as well.

Here we describe a more efficient, tessellation-independent way of implementing the iterative algorithm. Our implementation is built on a graph representation of the topological features in the MPSN that captures their adjacency structure. The graph can monitor topological changes in each detachment operation without actually performing the mesh surgeries. As a result, the time complexity of exploring possible detachment candidates scales with the number of topological features rather than the number of mesh elements.

In the following, we start with an elaboration on the graph representation, and then detail the iterative algorithm for topology simplification. Finally, we demonstrate one example produced by our prototype implementation of the algorithm in *Mathematica*.

4.2 A Graph Representation

Our graph representation (called a manifold graph) is a minimal structure that captures the topological features and their adjacency in the MPSN. The manifold graph naturally extends from the conventional arc-node graph, and contains additional elements to capture the adjacency of three types of topological features (sheets, seams and joints). Using the manifold graph, a detachment operation on the MPSN can be interpreted as a sequence of elementary graph operations that are extended from those in a conventional graph.

4.2.1 The 2D Case: The Conventional Graph

We start again by investigating the 2D scenario. Note that the segments and joints in a 2D MPCN can be naturally captured by arcs and nodes in a conventional graph. We allow an arc to have no end nodes (i.e., a closed loop), or two end nodes that can be identical. For example, the 4 segments and 2 joints in the MPCN in Figure 2.3(a1) can be represented by 4 arcs and 2 nodes, in which the arc representing the outermost segment contains no end nodes. On the other hand, the MPCN in Figure

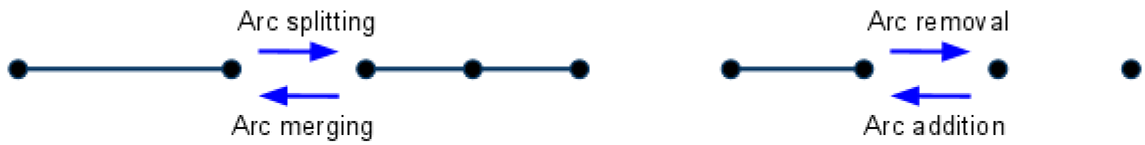


Figure 4.1: Four types of 2D elementary graph operations: arc splitting, merging, addition, and removal.

2.3(c1) can be represented by 3 arcs and 1 node, in which the two interior arcs have identical end nodes.

A detachment operation on the MPCN can be interpreted as a sequence of elementary graph operations, including arc splitting, merging, addition, and removal. These elementary operations are illustrated in Figure 4.1.

Given a triple $< I, D, F >$ with the detached feature I, detached region D and fill region F, the two steps in the detaching operation can be re-formulated on the graph as follows (illustrated in Figure 4.2 for the example in Figure 2.3(a1-a3)):

1. **Step 1 (partitioning):** Split the arc(s) next to I and contained in D (Figure 4.2(b)), and add an arc that connects the newly introduced nodes (Figure 4.2(c)).
2. **Step 2 (merging):** Remove the arc(s) lying between D1 and F (Figure 4.2(d)), and merge the arc(s) so that the arcs and nodes capture the segments and joints in the modified MPCN (Figure 4.2(e)).

4.2.2 The 3D Case: The Manifold Graph

Compared to a 2D MPCN, a 3D MPSN has an additional type of topological feature (sheet) that cannot be easily captured in the conventional arc-node graph. We therefore consider an extended graph formulation called *manifold graph*, which consists of three types of elements, nodes, arcs, and patches. As in 2D, an *arc* connects zero or two (possibly identical) nodes. A *patch* connects zero or several *rings*, where each ring consists of either a single node or a closed circuit of arcs. A sheet, seam or joint

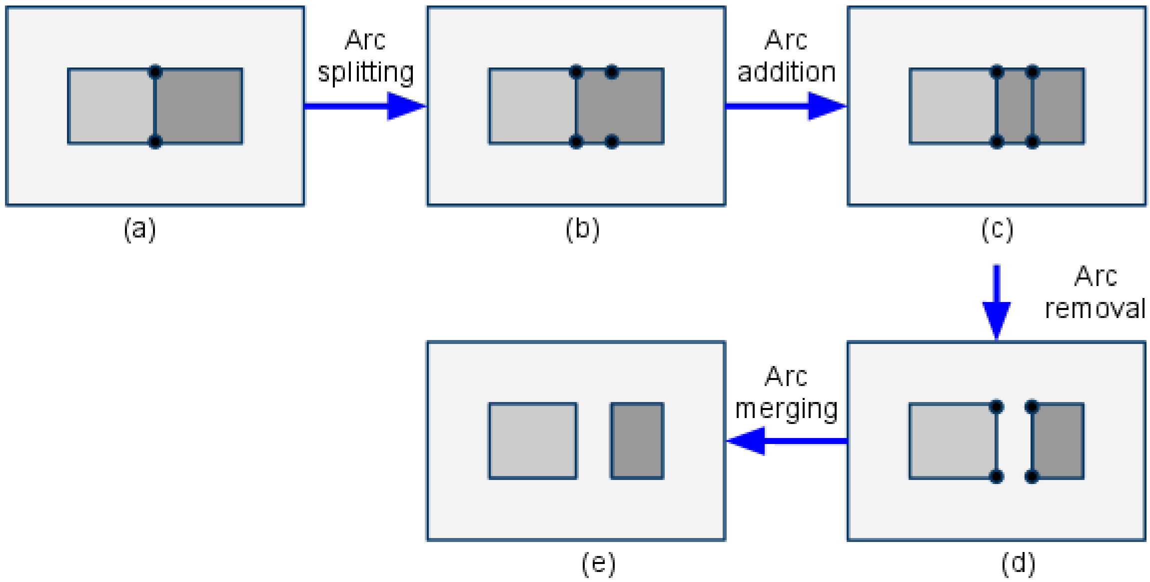


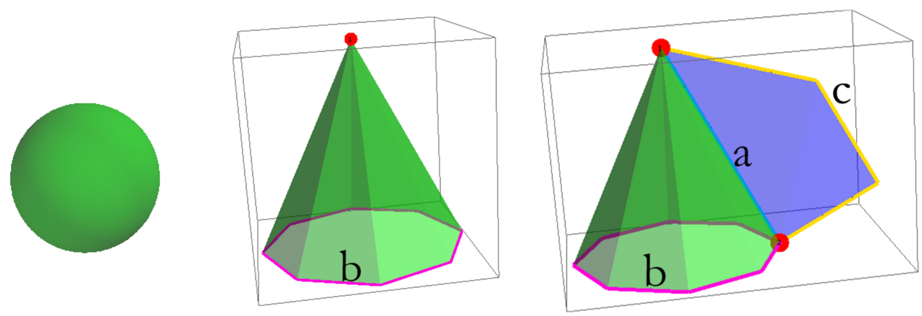
Figure 4.2: The 2D detachment operator in Figure 2.3(a) as four graph operations.

on a MPSN is captured respectively by a patch, arc or node in the manifold graph. In particular, each boundary of a sheet, which can be either a joint or a closed loop of seams, is represented by a ring in the corresponding patch.

As examples, the MPSNs in Figure 4.3 are respectively represented by a manifold graph with a single patch with no rings (a), a single patch with two rings where one of the rings is a single node and the other is made up of a single arc with zero end node (b), and two patches each consisting of a single ring made up of arcs $\{a, b, a\}$ and $\{a, c\}$ (c).

We can now extend the elementary operations in an arc-node graph to a manifold graph.

Addition: Addition creates a new node, a new arc without end-node or connecting two (possibly identical) existing nodes, or a new patch made up of zero or multiple rings of existing nodes and arcs.



(a) (b) (c)

Figure 4.3: Example of different types of patches in manifold graph: (a) a single patch with no rings. (b) a single patch with two rings where one of the rings is a single node and the other made up of a single arc with zero end node. (c) two patches each consisting of a single ring made up of arcs a,b,a and a,c.

Removal: Removal deletes a patch and its associated rings, or an arc that is not shared by any patches, or a node that is not shared by any arcs or patches.

Merging: We separately discuss the merging operations of patches and arcs. If an arc is used exactly twice in all patch rings, patch merging removes the arc and modifies the involved rings and patches as follows. If the arc appears twice in the same ring, the ring will be split into two, both belonging to the original patch (see Figure 4.4(a)). If the arc appears in two rings of a same patch, the two rings are joined (see Figure 4.4(b)). If the arc appears in the rings of two patches, the two rings are joined, and all rings of the two patches belong to a new merged patch (see Figure 4.4(c)).

Arc merging proceeds in a similar way. If a node is used exactly twice in all arcs and never in any patch rings, and if both incident arcs appear successively in their incident patch rings, merging removes the node, joints the two arcs into a new arc, and replaces the successive appearance of the original two arcs in patch rings with the new arc (see Figure 4.4(d)).

Splitting: Splitting reverses the merging operation. Patch splitting adds a new arc between two (possibly identical) nodes incident to the arcs in the patch's rings, and modifies the rings and patches as follows. If the two nodes belong to two different

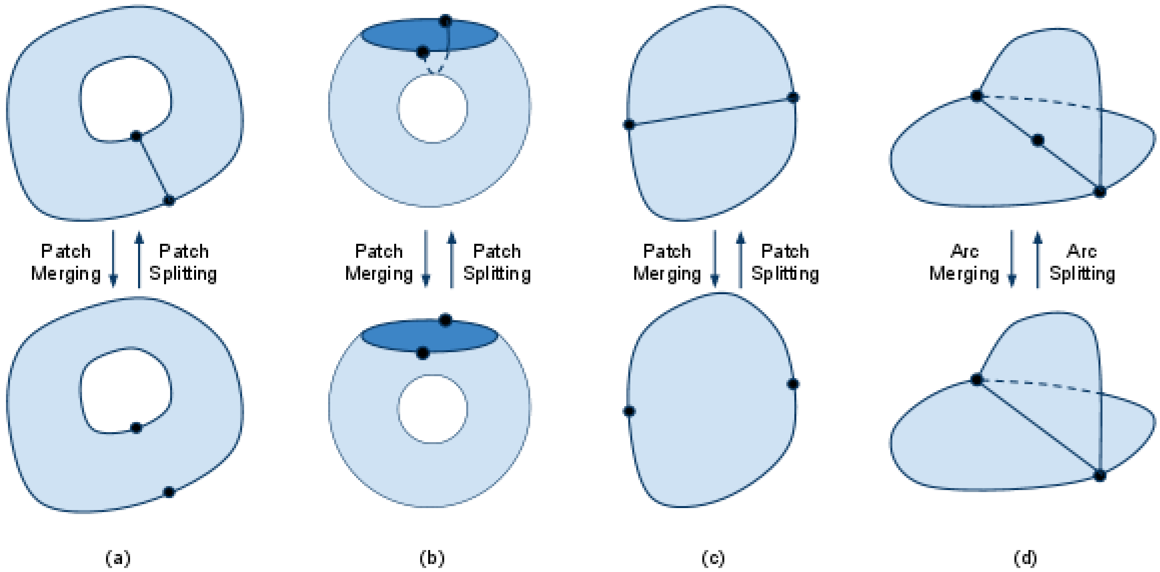


Figure 4.4: The 3D detachment operator as manifold graph operations: Patch merging and splitting (a-c), arc merging and splitting (d).

rings, the two rings are joined by the new arc and still belong to the same patch (Figure 4.4(a)). If the two nodes belong to the same ring, and if the arc does not disconnect the patch, the ring is split into two rings by the new arc, and both rings still belong to the same patch (Figure 4.4(b)). If the two nodes belong to the same ring and the arc disconnects the patch, the ring is split into two rings, each belonging to one of the split patches (Figure 4.4(c)).

Arc splitting adds a new node, splits one arc into two arcs (or one arc with two identical end nodes) using that node, and replaces the appearance of the original arc in all patch rings by the new arc(s) (Figure 4.4(d)).

The detachment operation on a MPSN can be interpreted as a sequence of these elementary graph operations. Given a triple $< I, D, F >$ with the detached feature I, detached region D and fill region F, the two steps in the detaching operation can be re-formulated as follows (illustrated in Figure 4.5 for the example in Figure 2.4(a)):

1. **Step 1 (partitioning):** Split the arc(s) (first) and patches (next) that are next to I and contained in D (Figure 4.5(a)), and add a patch whose only ring consists of the newly created arcs (Figure 4.5(b)).

2. **Step 2 (merging):** Remove the patch(es) lying between $D1$ and F (Figure 4.5(c)), and merge the arc(s) and patch(es) whenever possible (Figure 4.5(d)), so that the patches, arcs and nodes capture the sheets, seams and joints in the modified MPSN.

4.3 The Simplification Algorithm

To simplify the topology of an MPSN, we first need to construct the manifold graph from the mesh. To do so, we first represent each mesh vertex as a node, each mesh edge as an arc connecting two nodes, and each mesh triangle as a patch containing a single ring that is made up of three arcs. To obtain a minimal representation, we perform the merging operation on patches and arcs until no more merging is possible. Note that merging effectively groups triangles (or edges) that belong to the same connected, manifold surface (or curve) on the MPSN. Hence the result of exhaustive merging is a graph whose patches, arcs and nodes have one-to-one correspondence with the sheets, seams and joints on the MPSN. An example of the merging result is shown in Figure 4.6.

Once the graph is built, we take the iterative, greedy strategy mentioned at the beginning of this chapter. At each iteration, all possible triples of detached feature, detached region, fill region are explored on the manifold graph. For each triple, the detachment operation is performed on the graph, and the "score" of the triple is obtained as a vector of five numbers collected on the detached graph::

1. Number of patches (P)

2. Number of arcs (A)

3. Number of nodes (N)

4. Number of patches sharing an arc, summed over all arcs (SA)

5. Number of patches sharing a node, summed over all nodes (SN).

The triple with the minimal score (in lexicographical order) is selected. Mesh surgery is carried out (as discussed in Chapter 2) if the following two criteria are met: 1) the triple has a smaller score than the current graph, and 2) its geometric size (i.e., area of a sheet or length of a seam) is smaller than a user-given threshold. The iteration continues until the minimum-scoring triple fails either criterion.

We have implemented a prototype of the algorithm in *Mathematica*, and examples of the execution of the algorithm on simple MPSNs are shown in Figure 4.7 and Figure 4.8.

In Figure 4.7 the input graph consists of one patch, two arcs and two nodes and has a score of $\{1, 2, 2, 5, 2\}$. The detachment made on the arc $\{1, 2\}$ has a lower score (Figure 4.7(b)) than those of the detachments made on the nodes (Figure 4.7(c)). Therefore the detachment on the arc is chosen and the graph in Figure 4.7(b) is the final result of our algorithm.

Figure 4.8 illustrates the steps of our algorithm on an input graph consisting of four patches, one arc and one node. At each step, the algorithm selects the detachment with best score. The final graph has no node or arc but only three patches.

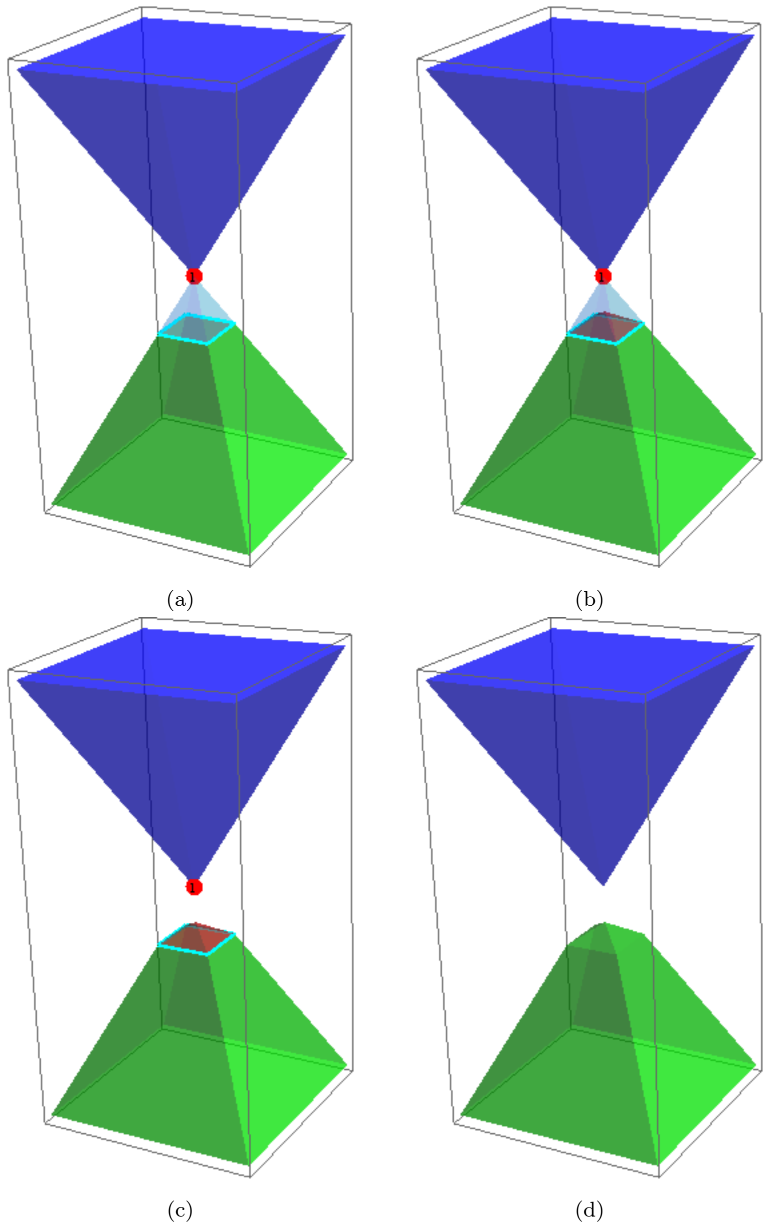
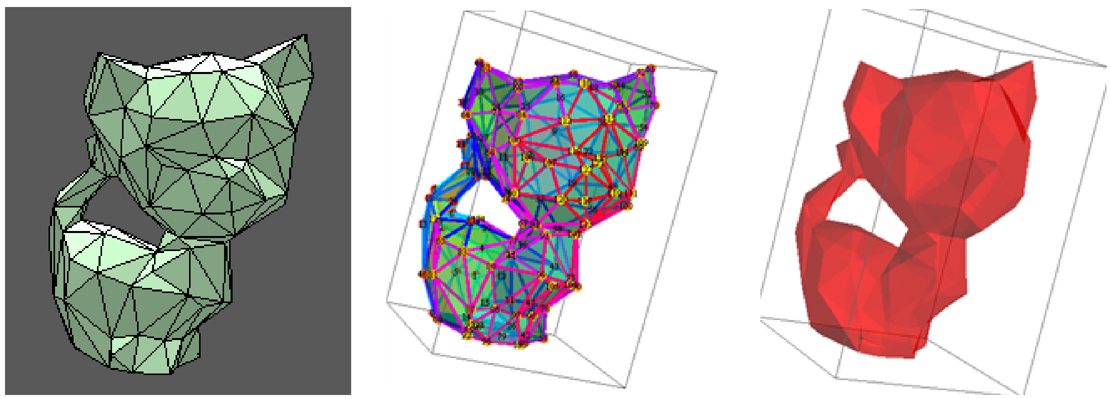
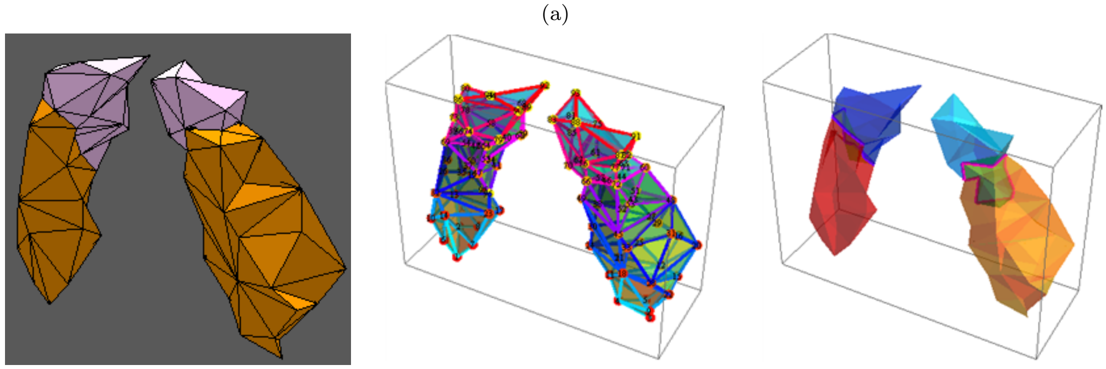


Figure 4.5: The 3D graph operator as four graph operations: (a) patch splitting, (b) patch addition, (c) patch removal, (d) arc and node merging.

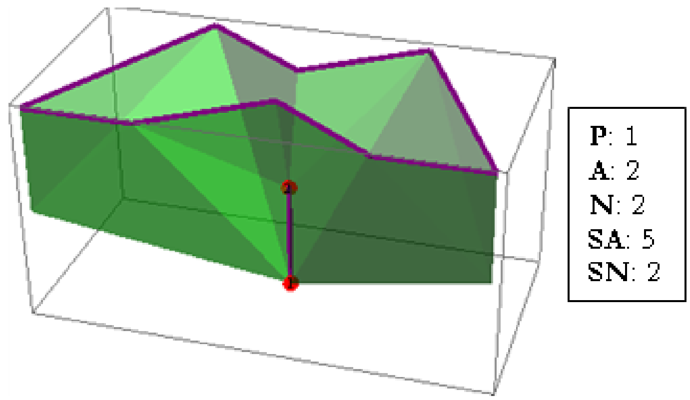


(a)

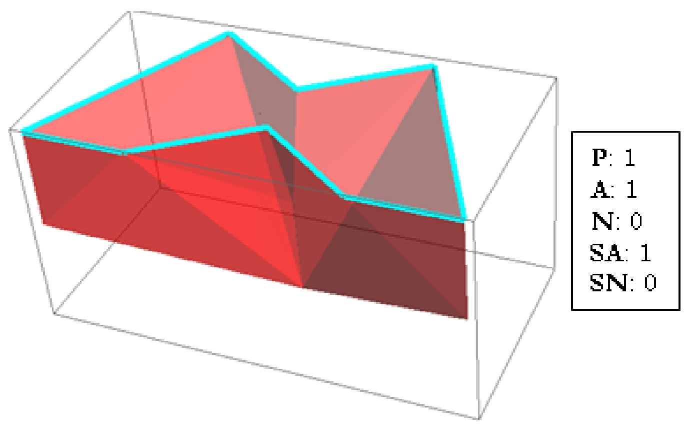


(a)

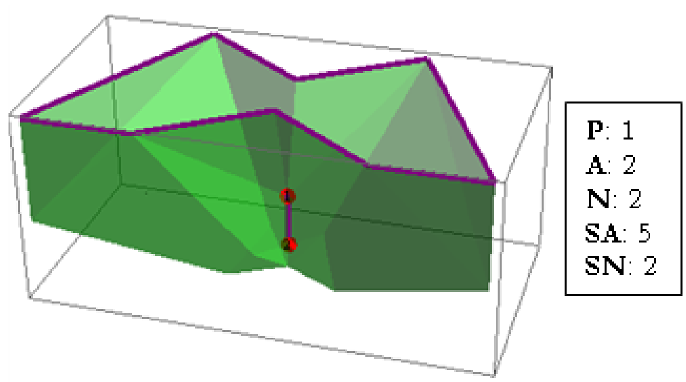
Figure 4.6: Examples of manifold graph construction from MPSNs. In each example, we start with an input 3D MPSN on the left, then build an initial graph in the middle to do exhaustive merging and finally get the minimal graph representation on the right. (a) A kitten MPSN. (b) A MPSN containing three anatomical regions (vortex, basal forebrain and amygdala) in the mouse brain model



(a)



(b)



(c)

Figure 4.7: Different choices to change topological complexity on a graph: (a) Original graph has a score of $\{1, 2, 2, 5, 2\}$. (b) Graph after detaching the arc 1,2 has a score of $\{1, 1, 0, 1, 0\}$. (c) Graph after detaching the node 1 has a score of $\{1, 2, 2, 5, 2\}$.

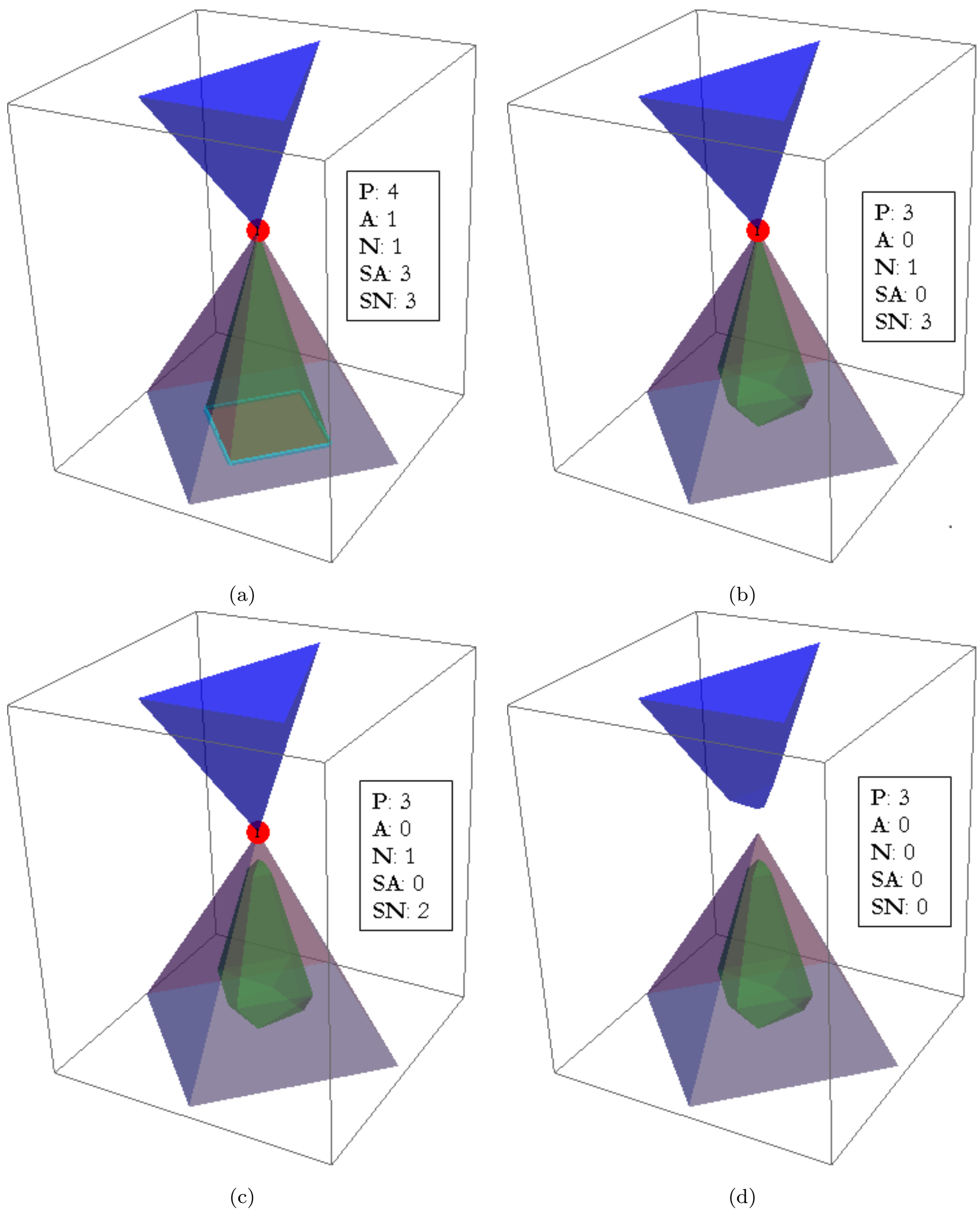


Figure 4.8: An example of topology simplification process by the automated algorithm: (a) Original graph has a score of $\{4, 1, 1, 3, 3\}$. (b) Step 1: Detach the middle patch at the bottom from the green region, resulting in a graph with a score of $\{3, 0, 1, 0, 3\}$. (c) Step 2: Detach the only node from the green region, resulting in a graph with a score of $\{3, 0, 1, 0, 2\}$. (d) Step 3: Detach the only node from the blue region, resulting in a graph with a score of $\{3, 0, 0, 0, 0\}$. The graph in (d) is our final result.

Chapter 5

Conclusion and Future Work

We have introduced a topology simplification algorithm to reduce the topological complexity of multi-partitioning surface networks. Our interactive and automated means help remove redundant non-manifold features in MPSNs. Our mesh surgery operator, implemented in an interactive tool, can effectively simplify the non-manifold topology while preserving the valildity of the MPSNs under user guidance. However, removing undesired topological features interactively might be very time-consuming on complicated MPSNs, such as the mouse brain model example shown in the thesis. Our proposed automated topology simplifcation method, which is guided by both topological and geometric objectives and is tessellation-indepedent, has a great potential to enable a faster and more robust process.

We wish to make our algorithm more efficient. For example, our algorithm currently selects the detachment triple greedly and the entire list of candidates and their scores get updated after every iteration. This process is very time consuming and for the most part redundant since the candidates and their scores need to be updated only when their neighborhoods change after a detachment operation. Therefore, local updates of the list of candidates and their scores are preferred. We would also like to explore more implementation optimizations, such as using a priority queue for storing detachment candidates, and parallel execution of non-adjacent detachment operations. We also wish to make our algorithm "less" greedy, i.e. to find better heuristics which can suggest more optimal solutions.

Furthermore, our current implementation of the automated algorithm, written in *Mathematica*, is quite slow due to the nature of *Mathematica* as a symbolic language. In the future, we wish to transform the implementation into an imperative programming language like C or C++. This will enable the algorithm to run much faster.

References

[1] Gill Barequet and Amir Vaxman. Reconstruction of multi-label domains from partial planar cross-sections. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1327–1337, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[2] Tamal K. Dey, Herbert Edelsbrunner, Sumanta Guha, and Dmitry V. Nekhayev. Topology preserving edge contraction. *Publ. Inst. Math. (Beograd) (N.S*, 66:23–45, 1998.

[3] Powei Feng, Tao Ju, and Joe D. Warren. Piecewise tri-linear contouring for multi-material volumes. In *GMP*, pages 43–56, 2010.

[4] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[5] Igor Guskov and Zoë J. Wood. Topological noise removal. In *Graphics Interface*, pages 19–26, 2001.

[6] Hugues Hoppe. Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '96, pages 99–108, New York, NY, USA, 1996. ACM.

[7] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. *ACM Trans. Graph.*, 21:339–346, July 2002.

[8] Tao Ju, Joe D. Warren, James Carson, Gregor Eichele, Christina Thaller, Wah Chiu, Musodiq Bello, and Ioannis A. Kakadiaris. Building 3d surface networks from 2d curve networks with application to anatomical modeling. *The Visual Computer*, 21(8-10):764–773, 2005.

[9] Tao Ju, Qian-Yi Zhou, and Shi-Min Hu. Editing the topology of 3d models by sketching. In *ACM SIGGRAPH 2007 papers*, SIGGRAPH '07, New York, NY, USA, 2007. ACM.

[10] Lu Liu, C. Bajaj, Joseph Deasy, Daniel A. Low, and Tao Ju. Surface reconstruction from non-parallel curve networks. *Comput. Graph. Forum*, 27(2):155–163, 2008.

[11] Fakir S. Nooruddin and Greg Turk. Simplification and repair of polygonal models using volumetric techniques. *IEEE Trans. Vis. Comput. Graph.*, 9(2):191–205, 2003.

[12] M. Haitham Shammaa, Hiromasa Suzuki, and Yutaka Ohtake. Extraction of isosurfaces from multi-material ct volumetric data of mechanical parts. In *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, SPM '08, pages 213–220, New York, NY, USA, 2008. ACM.

[13] Andrzej Szymczak and James Vanderhyde. Extraction of topologically simple isosurfaces from volume datasets. In *IEEE Visualization*, pages 67–74, 2003.

[14] Fabien Vivodtzev, Georges-Pierre Bonneau, and Paul Le Texier. Topology-preserving simplification of 2d nonmanifold meshes with embedded structures. *The Visual Computer*, 21(8-10):679–688, 2005.

[15] Zoë J. Wood, Hugues Hoppe, Mathieu Desbrun, and Peter Schröder. Removing excess topology from isosurfaces. *ACM Trans. Graph.*, 23(2):190–208, 2004.

Vita

Trung Duc Nguyen

Place of Birth	Ha Nam, Vietnam
Degrees	B.Eng. Computer Science, Hanoi University of Technology, July 2006 M.S. Computer Science, Washington University, May 2011
Experience	Headquarter Information Systems-Billings Department, Vietnam Mobile Services Company, October 2006 - July 2007

May 2011

Simplifying Non-Manifold Topology, Nguyen, M.S. 2011