

Old Dominion University

ODU Digital Commons

Computational Modeling & Simulation
Engineering Theses & Dissertations

Computational Modeling & Simulation
Engineering

Summer 2017

Multi-Material Mesh Representation of Anatomical Structures for Deep Brain Stimulation Planning

Tanweer Rashid

Old Dominion University, Tanweer.Rashid@gmail.com

Follow this and additional works at: https://digitalcommons.odu.edu/msve_etds



Part of the [Bioimaging and Biomedical Optics Commons](#), [Biotechnology Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Rashid, Tanweer. "Multi-Material Mesh Representation of Anatomical Structures for Deep Brain Stimulation Planning" (2017). Doctor of Philosophy (PhD), Dissertation, Computational Modeling & Simulation Engineering, Old Dominion University, DOI: 10.25777/07s9-6064
https://digitalcommons.odu.edu/msve_etds/10

This Dissertation is brought to you for free and open access by the Computational Modeling & Simulation Engineering at ODU Digital Commons. It has been accepted for inclusion in Computational Modeling & Simulation Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact digitalcommons@odu.edu.

**MULTI-MATERIAL MESH REPRESENTATION OF ANATOMICAL STRUCTURES
FOR DEEP BRAIN STIMULATION PLANNING**

by

Tanweer Rashid

B.S. August 2008, North South University, Bangladesh

M.S. May 2011, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY

August 2017

Approved by:

Michel A. Audette (Director)

Frederic D. McKenzie (Member)

Khan M. Iftekharuddin (Member)

Gregory S. Fischer (Member)

ABSTRACT

MULTI-MATERIAL MESH REPRESENTATION OF ANATOMICAL STRUCTURES FOR DEEP BRAIN STIMULATION PLANNING

Tanweer Rashid
Old Dominion University, 2017
Director: Michel A. Audette

The Dual Contouring algorithm (DC) is a grid-based process used to generate surface meshes from volumetric data. However, DC is unable to guarantee 2-manifold and watertight meshes due to the fact that it produces only one vertex for each grid cube. We present a modified Dual Contouring algorithm that is capable of overcoming this limitation. The proposed method decomposes an ambiguous grid cube into a set of tetrahedral cells and uses novel polygon generation rules that produce 2-manifold and watertight surface meshes with good-quality triangles. These meshes, being watertight and 2-manifold, are geometrically correct, and therefore can be used to initialize tetrahedral meshes.

The 2-manifold DC method has been extended into the multi-material domain. Due to its multi-material nature, multi-material surface meshes will contain non-manifold elements along material interfaces or shared boundaries. The proposed multi-material DC algorithm can (1) generate multi-material surface meshes where each material sub-mesh is a 2-manifold and watertight mesh, (2) preserve the non-manifold elements along the material interfaces, and (3) ensure that the material interface or shared boundary between materials is consistent. The proposed method is used to generate multi-material surface meshes of deep brain anatomical structures from a digital atlas of the basal ganglia and thalamus. Although deep brain anatomical structures can be labeled as functionally separate, they are in fact continuous tracts of soft tissue in close proximity to each other. The multi-material meshes generated by the proposed DC

algorithm can accurately represent the closely-packed deep brain structures as a single mesh consisting of multiple material sub-meshes. Each sub-mesh represents a distinct functional structure of the brain.

Printed and/or digital atlases are important tools for medical research and surgical intervention. While these atlases can provide guidance in identifying anatomical structures, they do not take into account the wide variations in the shape and size of anatomical structures that occur from patient to patient. Accurate, patient-specific representations are especially important for surgical interventions like deep brain stimulation, where even small inaccuracies can result in dangerous complications. The last part of this research effort extends the discrete deformable 2-simplex mesh into the multi-material domain where geometry-based internal forces and image-based external forces are used in the deformation process. This multi-material deformable framework is used to segment anatomical structures of the deep brain region from Magnetic Resonance (MR) data.

Copyright, 2017, by Tanweer Rashid, All Rights Reserved.

This dissertation is dedicated to my family and friends, and to the pursuit of knowledge in order to achieve a more complete understanding of the universe and all that it contains, for the sake of Allah, the All Knowing, The Omniscient, The Infinite.

And if whatever trees upon the earth were pens and the sea [was ink], replenished thereafter by seven [more] seas, the words of Allah would not be exhausted. Indeed, Allah is Exalted in Might and Wise.

– The Holy Quran, Surah 31, Verse 27

ACKNOWLEDGMENTS

This dissertation was only possible through the will and mercy of Allah, the Infinitely Compassionate and Merciful, Lord of all that is in between the heavens and the earth. There is none other worthy of worship except Allah, and Mohammad (peace and blessings of Allah be upon him) is the last Messenger of Allah.

This dissertation represents almost 6 years of hard work and effort, sweat, and tears, and I would not have made it to this point without the help, support and guidance of my advisor Dr. Michel Audette. I would also like to express my gratitude to Dr. Rick Mckenzie for the help, support and encouragement that he has given me. Many thanks also go to Dr. Greg Fischer and Dr. Khan Iftekharuddin for taking the time and effort to be invaluable members of my dissertation committee. I would also like to thank Dr. Powei Feng and Dr. Benjamin Gilles for providing with software that I used to get a head start on my research. It probably would have taken me even longer to complete my Ph.D. otherwise. Finally, many thanks to Dr. Andrey Chernikov, and Dr. Yongjie Zhang for providing me extremely helpful suggestions and feedback on my work.

It goes without saying that and the staff and my colleagues in the MSVE Department have given me tremendous help, support, and encouragement in all the years that I have been a graduate student in the Department. I would like to especially thank my dear friend Sharmin, whom I have used as a sounding board for my many wild thoughts and ideas regarding research and non-research matters, and who also took the time and patience to listen though my rants, fears, dreams, and goals. And last, but not the least, I would like to thank my mother and father, my brothers and sister, my grandmother, and my aunts and uncles for all their support, and for always believing in me.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
Chapter	
1. INTRODUCTION	1
1.1 Motivation	3
1.3. Contributions	5
1.4. Organization	8
2. FOUNDATIONAL CONCEPTS	9
2.1 Dual Contouring	9
2.2 Simplex Meshes	13
2.3 Volume Smoothing.....	23
3. 2-MANIFOLD DUAL CONTOURING.....	29
3.1 Background and Literature Review.....	30
3.2 2-Manifold Dual Contouring.....	31
3.3 Tetrahedral Decomposition of Ambiguous Cubes	34
3.4 Polygon Generation.....	37
3.5 Detection of Non-manifold Edges and Vertices and Boundary Edges	41
3.6 Integration into Dual Contouring	42
3.7 Results and Discussion.....	44
3.8 Comparison with Existing Methods	58
3.9 Mesh Smoothing.....	60
3.10 Tetrahedral Mesh Generation	63
3.11 Conclusions	65
4. MULTI-MATERIAL AND 2-MANIFOLD DUAL CONTOURING	68
4.1 Literature Review	68
4.2 Multi-material vs. 2-Manifold.....	70
4.3 Ambiguous vs. Unambiguous	74
4.4 Tetrahedral Decomposition of Ambiguous Cubes	78
4.5 Polygon Generation.....	81
4.6 Verifying 2-Manifoldness in Multi-material Meshes.....	82
4.7 Mesh Smoothing.....	83
4.8 Results and Discussion.....	84
4.9 Multi-Material Tetrahedral Mesh Generation	94
4.10 Conclusions	96

5. MULTI-MATERIAL 2-SIMPLEX DEFORMABLE MESH	97
5.1 Background	97
5.2 Initializing a Multi-Material 2-Simplex Mesh	103
5.3 Description of Multi-Material 2-Simplex Meshes	106
5.4 Overview of Deformable Multi-material 2-Simplex Meshes.....	109
5.5 Multi-material Deformation Using a Synthetic Example.....	111
5.6 Segmentation of the Subthalamic Nucleus and Substantia Nigra	115
5.7 Segmentation of the Globus Pallidus and Striatum.....	120
5.8 Validation	133
5.9 Conclusion.....	140
6. CONCLUSIONS AND FUTURE WORK	142
6.1 Multi-Material and 2-Manifold Dual Contouring	142
6.2 Multi-Material 2-Simplex Deformable Meshes	145
REFERENCES	149
VITA.....	160

LIST OF TABLES

Table	Page
1. Results of Surfaces Generated Using Standard Dual Contouring on Unprocessed Data	50
2. Results of Surfaces Generated Using the Proposed Dual Contouring Method on Unprocessed Data.....	50
3. Results of Surfaces Generated Using Standard Dual Contouring on Preprocessed Data	54
4. Results of Surfaces Generated Using the Proposed Dual Contouring Method on Preprocessed Data	54
5. Mesh Quality for Meshes Generated by the Proposed Method Using Unprocessed Data and Preprocessed Data	57
6. Mesh Statistics and Quality	59
7. Mesh Quality Report for Striatum, Internal Capsule, and Globus Pallidus Group	89
8. Mesh Quality Report for Components of the Thalamus.....	92
9. Summary of Deformation Errors	140

LIST OF FIGURES

Figure	Page
1. Components of deep brain stimulation	2
2. An illustration of the duality, in 2D, between Marching Cubes algorithm (left) and Dual Contouring algorithm (right).....	9
3. Tree representation of an image.....	10
4. Difference between vertex computation for MC and DC.....	11
5. Minimal edge in an adaptive octree. AB and BC are minimal edges because they do not contain any edge smaller than themselves. AC is not a minimal edge because it contains AB and BC.....	13
6. Examples of Simplex meshes. (left) 1-simplex and (right) 2-simplex	14
7. Geometric representation of a vertex P and its relation to its three neighboring vertices P1, P2 and P3 in a 2-simplex mesh.	15
8. Relationship between $\mathbf{F}_{\text{Normal}}$ and $\mathbf{F}_{\text{Tangent}}$ with respect to the neighbors of point \mathbf{P}_i	18
9. Topological operators for the 2-simplex mesh.	22
10. An example of binary morphology.....	25
11. Surface meshes of a volume smoothed using binary morphology operations.....	26
12. An example of multi-material smoothing using Gaussian blurring.....	28
13. The 14 fundamental configurations for a grid cube.....	33
14. Non-manifold vertex due to complimentary Case 4.	34
15. Creation of the face diagonal.	36
16. An illustration of the Minimal Edge Rule.....	38
17. An illustration of the Face Diagonal Rule.	39
18. An illustration of the Interior Edge Rule.	42
19. Two examples of a non-manifold vertex being replaced by a 2-manifold mesh.....	44

20. Examples of applying the proposed DC method on synthetic data.	46
21. Surfaces generated from a digital deep brain atlas using standard DC.....	48
22. Surfaces generated from a digital deep brain atlas using the proposed method.	49
23. Surfaces generated using standard DC on preprocessed atlas data.....	52
24. Surfaces generated using the proposed method on preprocessed atlas data.	53
25. An illustration of the digital deep brain atlas.....	55
26. Meshes of the Asian Dragon model.....	60
27. Two meshes of the Nucleus lateropolaris thalami (atlas label 26) generated using the proposed method.	62
28. The results of applying Laplacian smoothing on a surface mesh of the Nucleus lateropolaris thalami (atlas label 26).....	63
29. Tetrahedral meshes created using the surface mesh of the Nucleus lateropolaris thalami (atlas label 26) as the input PLC.....	65
30. A 2D example of a uniform grid superimposed on a multi-material domain. The four corners of the square occupy three different material domains as well as the background.....	71
31. A simple example of a multi-material of two materials, with a shared boundary.	72
32. A synthetic example of a multi-material mesh with three materials, having non-manifold edges and vertex.....	74
33. Examples of ambiguous cubes. (a) Face ambiguity, and the two possible surfaces (b) Interior ambiguity and the two possible surfaces	76
34. Identifying ambiguity for multi-material domains.	77
35. An illustration of ambiguous cube decomposition.	80
36. Taubin smoothing. (left) Unsmoothed mesh. (right) Mesh smoothed using Taubin smoothing, with 10 iterations.....	84
37. A multi-material representation of the striatum (purple), internal capsule (yellow), Globus Pallidus (green), Globus Pallidus External (red) and Globus Pallidus Internal (blue).	86

38. An illustration of the complexity of the shared boundaries between the (a) striatum, (b) internal capsule, (c) Globus Pallidus, (d) Globus Pallidus Internal and (e) Globus Pallidus External. The different colors represent the parts of the mesh that are shared between the structures.	87
39. A coronal slice of the multi-material mesh of the Striatum (s), Internal Capsule (ic), Globus Pallidus (GP), Globus Pallidus Internal (GPI) and Globus Pallidus External (GPE). The different colors represent the different shared boundaries between two structures.	88
40. A multi-material representation of the thalamus.	90
41. A cross section of the multi-material mesh of the thalamus. The different colors represent the different constituent components of the thalamus.	91
42. An example of a multi-material tetrahedral mesh.....	95
43. Converting a triangular mesh into a simplex mesh.....	104
44. Conversion of a multi-material triangular mesh of two boxes, colored red and blue and having a shared boundary, shown in green, into a multi-material 2-simplex mesh.....	106
45. An example of a multi-material 2-simplex mesh. The highlighted vertex on the non-manifold edge has more than 3 neighboring vertices.	108
46. An example of 2-simplex sub-meshes. (left) The whole multi-material 2-simplex mesh. (right) the constituent pure 2-simplex sub-meshes	108
47. A flowchart of the multi-material 2-simplex deformation process.....	110
48. Initialization of the multi-material 2-simplex deformation.	113
49. State of the mesh deformation with respect to the image gradient.	114
50. Graph of absolute displacement for the deformation of the synthetic box example.	114
51. A rendering of the labels in the Wang atlas.	116
52. Meshes of the SN and STN.....	117
53. Coronal slice of the T2-weighted 7T MRI. (Inset) The subthalamic nucleus and substantia nigra	118

54. State of the deformation of the SN and STN. The red outline represents the outline of the STN and the blue outline represents the outline of the SN.....	119
55. Graph of absolute displacement for the deformation of the SN and STN.....	120
56. Final surface mesh (left) and wireframe (right) of the SN (blue) and STN (red).	120
57. The differing contrasts of the globus pallidus and striatum in T1-weighted MRI (left) and T2-weighted MRI (right).....	121
58. The surface mesh representation of the GP. (Top) Wang atlas, (bottom) the three parts of the GP in Chakravarty's atlas. (Green) Globus Pallidus, (Yellow) Globus Pallidus External, (Blue) Globus Pallidus Internal.	122
59. Simplex mesh generation for the Striatum and GP.....	123
60. Cross-sections of the striatum and globus pallidus during deformation using T1-weighted MRI.	125
61. Graph of absolute displacement for the deformation of the GP and striatum using only T1-weighted MRI.....	126
62. The final surface mesh (left) of the striatum and GP after deformation using only T1-weighted MRI. (Middle and right) A wireframe rendering of the striatum and GP.....	126
63. Cross-sections of the striatum and globus pallidus during deformation using T2-weighted MRI.	128
64. Graph of absolute displacement for the deformation of the GP and striatum using only T2-weighted MRI.....	129
65. The final surface mesh (left) of the striatum and GP after deformation using only T2-weighted MRI. (Middle and right) A wireframe rendering of the striatum and GP.....	129
66. Cross-sections of the striatum and globus pallidus during deformation using both T1 and T2-weighted MRI.....	131
67. Graph of absolute displacement for the deformation of the GP and striatum using both T1 and T2-weighted MRI.	132
68. The final surface mesh (left) of the striatum and GP after deformation using both T1 and T2-weighted MRI.....	132
69. (Top row) mesh of the STN, (bottom row) mesh of the SN.	135

70. Histogram of segmentation errors for the STN.....	135
71. Histogram of segmentation errors for the SN.	136
72. Final deformation of the GP using: (left) only T1-weighted MRI, (Middle) only T2-weighted MRI, and (right) T1 and T2-weighted MRI.	138
73. Histogram of segmentation errors for the GP using only T1-weighted MRI.	138
74. Histogram of segmentation errors for the GP using only T2-weighted MRI.	139
75. Histogram of segmentation errors for the GP using both T1 and T2-weighted MRI.	139

CHAPTER 1

INTRODUCTION

Deep brain stimulation (DBS) is a treatment where electrodes are surgically implanted into the brain, which are then used to apply electrical impulses into targeted anatomical structures. DBS is used for treating neurological disorders such as Parkinson's Disease (PD), Dystonia, Tourette syndrome and epilepsy, and psychological disorders such as treatment-resistant depression (TRD), and obsessive-compulsive disorders (OCD) [1]. The targets for implantation depend on the disorder being treated: the thalamus and the globus pallidus for dystonia [2-5], the centromedian-parafascicular complex of the thalamus, the internal segment of the globus pallidus (GPi), and the anterior limb of the internal capsule for Tourette syndrome [6-9]. In the case of Parkinson's disease, there has been numerous studies with respect to deep brain stimulation, but the most promising implantation targets are the subthalamic nucleus (STN) [10] and GPi [11]¹.

A DBS system consists of three components: the lead or electrode, the extension, and the neurostimulator (Fig. 1). The lead is surgically implanted into the targeted deep brain regions. Surgeons identify the target region using Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) in conjunction with printed and/or digital atlases. The neurostimulator is a small battery powered device that is placed under the skin near the collarbone, lower chest, or abdomen. The extension connects the electrodes to the neurostimulator. The neurostimulator transmits electrical pulses to the electrode, and can be calibrated/programmed by a trained technician or nurse.

¹ IEEE Transactions and Journals style is used in this thesis for formatting figures, tables, and references.

There are two approaches to DBS: the awake procedure and asleep procedure. In the awake procedure, fiducials are attached to the patient's head using a stereotactic frame to set up a reference system for imaging. Patients are kept awake so that they are able to provide feedback to the surgeon. Typically, MR images are the modality of choice because MR imaging offers better visualization of the brain's soft tissue structures. The stereotactic MR images of the brain are obtained before the surgical procedure and are often augmented by coregistering with stereotactic CT images. The insertion trajectory can be based on either the MR/CT data alone, or by superimposing anatomical atlases on the MR/CT images. The skin of the skull is anesthetized using local anesthetic and a patch of hair on the skull is shaved and cleaned. A hole is drilled in the skull and the electrodes are implanted into the deep brain regions (see Fig. 1 inset). On the other hand, in the asleep procedure the patient is given a general anesthetic, and remains unconscious throughout the procedure. The asleep procedure can also make use of a stereotactic frame and stereotactic imaging for trajectory planning, and does not require the patient to provide feedback to the surgeon.

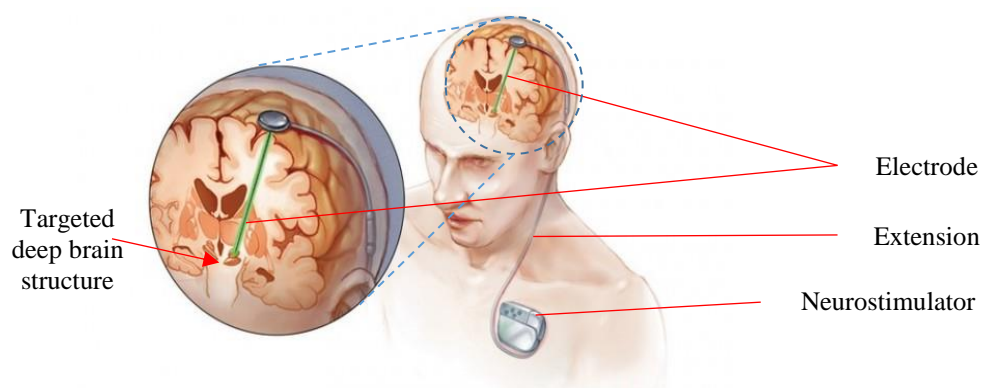


Fig. 1. Components of deep brain stimulation. Picture source: <http://vnstherapy-herb.blogspot.com/2013/07/deep-brain-stimulation-bleeding-edge-of.html>.

1.1 Motivation

For many neurological afflictions, such as Parkinson's disease (PD), dystonia, and epilepsy, DBS is the prescribed treatment. DBS has also been used for treating severe cases of psychological disorders like treatment-resistant depression (TRD) and obsessive-compulsive disorder (OCD). As many as 1 million Americans are living with PD, and nearly 60,000 Americans are diagnosed with PD each year [12]. Epilepsy is considered the fourth most common neurological problem and affects nearly 3 million Americans [13]. Major depression has a lifetime prevalence of 15%-20%, and it is one of the leading causes of disability worldwide. TRD affects up to 50%-60% of patients [14]. OCD has an estimated lifetime prevalence of 2%, and if left untreated, it can destroy a person's capacity to function at work, at home, and socially [15].

The traditional DBS procedure first involves a preoperative MR imaging of the patient's brain. A stereotactic frame is then affixed to the patient's head and fiducials are attached for referencing. This is followed by CT scans of the patient's head with the attached stereotactic frame. The traditional insertion strategy is based on the registered MR and CT images, along with a brain atlas in digital or printed form [16]. This framework of DBS limits the choice of approach, and surgical planning requires considerable mental computation by the surgeon. Modern DBS practice relies on microelectrode recordings (MER) for the confirmation of a successful implantation, but the use of MER comes at a cost of extended operating times and increases the potential for complications [17]. Intraoperative MRI (ioMRI)-guided DBS was proposed in [18] with the use of a MRI-compatible frame. However, this technique relies on difficult manual adjustment of trajectory guides in the scanner. Moreover, this adjustment

appears to be implemented by the user rather than computed through registration-based brain shift estimation.

Traditional and currently available image-guided approaches to DBS do not take into account the brain shift that occurs during implantation. This neglect increases the risk for complications. Under certain conditions, brain shift may be negligible [19], but other studies such as in [20-22] suggest that shifts averaging 2 mm and reaching 4 mm can occur anterior-posteriorly and laterally in targeting the STN. According to [23], the worst case is a cerebrovascular complication with significant rates of “*morbidities and mortalities resulting from the multiple trajectories used during physiological exploration of the brain target*”. In one study [24], there were 16 out of 81 patients whose ventricular walls had been penetrated, making this type of risk an important consideration. According to [25], there is also a risk of electric current being delivered outside the target location to nearby structures, and this can lead to neurological sequelae. Because of brain shift and subsequent incorrect targeting, there would be a need for multiple needle insertion to target a specific structure, and this can lead to corticospinal fluid loss and further shift.

The STN is a very small structure, averaging 5.9 mm in antero-posterior, 3.7 mm in mediolateral and 5 mm in dorsoventral dimensions [26]. The STN consists of a sensorimotor, associative, and limbic components where only the sensorimotor is targeted for PD. It is well within the realm of possibility for an electrode array to overlap most, or even the full length, of the STN. The risk of delivering current to nearby the associative and limbic components, which is exacerbated by brain shift, is consistent with behavioral changes seen after stimulation of the STN. Behavioral changes include cognitive problems seen in 41%, depression in 8% and hypomania in 4% of patients, according to [27]. Similar findings are also discussed recently in

[28]: *“Cognitive and behavioural disturbances in patients with Parkinson’s disease seem to be relatively more frequent after deep brain stimulation of the subthalamic nucleus... electrode misplacements or current spreading to non-motor circuits involving the subthalamic nucleus may give rise to cognitive and behavioural disturbances after subthalamic implants.”*

In conclusion, it is fair to say that current DBS techniques do not take into account the amount of brain shift that can occur during implantation. This increases the risk of inaccurate targeting of the anatomical structures thereby causing behavioral changes in the patient. Current research studies show that brain shift is a dire problem in DBS. In order to alleviate the risks involved in targeting, it is necessary to first obtain an accurate representation of a patient’s deep brain regions, and then have a mechanism by which brain shifts can be tracked, preferably in real-time.

1.2 Contributions

This section will briefly discuss the anticipated contributions that will result from the proposed research.

1.2.1 Watertight and 2-Manifold Dual Contouring Algorithm

Dual contouring (DC) [29] is a surface meshing algorithm similar to Marching Cubes (MC) [30], but with the added benefit of being fast and able to reproduce sharp features. However, one limitation of DC is that it is incapable of producing 2-manifold meshes, especially for complex surfaces and topologies. To date, there are few variations or enhancements of the DC algorithm that can reproduce sharp features while ensuring watertightness and 2-manifoldness of the surface meshes.

The main contribution of this chapter is a modified Dual Contouring algorithm that is capable to producing 2-manifold and watertight surface meshes. Furthermore, the proposed DC algorithm is topology-preserving, as well as capable of reproducing sharp features. *Ambiguous* grid cubes are decomposed into a set of tetrahedral cells, whose centroids are used as vertices of the output surface mesh. Novel polygon generation rules are devised which ensure error-free, 2-manifold and watertight triangular surface meshes.

1.2.2 Multi-material and 2-Manifold Dual Contouring Algorithm

Although deep brain anatomical structures can be labeled as functionally separate, they are in fact continuous tracts of soft tissue in close proximity to each other. It is therefore important to treat such structures as a whole, rather than separate. Surface mesh representations of these structures should, likewise, reflect on the continuity between such structures; failure to do so risks interfaces with small distances between and disparities in the deformation computation between neighboring surfaces.

The main contribution in this chapter is an extension of the above mentioned modified DC algorithm that is capable of generating multi-material and 2-manifold surface meshes where material *sub-meshes* have shared boundaries. Material information is implemented as pair-wise integers assigned to triangles of the surface mesh. Ordinarily, surface meshes containing non-manifold elements would be considered ‘defective’. However due to their multi-material nature, multi-material surface meshes will inherently contain non-manifold elements. In the proposed method, the multi-material surface meshes are defined as being 2-manifold in the sense that the sub-meshes of each material are, by themselves, completely 2-manifold and watertight. The proposed method is used to generate multi-material surface meshes of deep brain anatomical

structures from a digital atlas of the basal ganglia and thalamus. These meshes can accurately represent the closely-packed deep brain structures as a single mesh consisting of multiple material sub-meshes. Each sub-mesh represents a distinct functional structure of the brain.

1.2.3 Multi-Material 2-Simplex Mesh with Shared Boundaries

The proposed system will utilize a multi-material 2-simplex mesh having shared boundaries. This 2-simplex mesh will be initialized using multi-material triangular meshes created using the previously mentioned multi-material 2-manifold Dual Contouring algorithm. The use of this type of simplex mesh is preferred over a triangular mesh because the simplex mesh has well-defined geometry-based internal forces and image-based external forces [31, 32]. Further, the multi-material 2-simplex will be used to accurately represent closely packed anatomical structures in a manner not done before. In existing implementations, such as in [31, 33], the simplex meshes used were either of one single material, or consisted of several surfaces with boundaries independent of each other's. In the case of multiple simplex meshes [33], overlap amongst meshes was prevented using collision detection and handling. This collision detection entailed that overlapping surfaces would push each other away, but had no way of ensuring that shared boundaries were perfectly flush with each other, as this multi-surface approach could result in small spaces between surfaces. The main contribution here is a multi-material 2-simplex mesh with shared boundaries between materials.

In this research, an especial emphasis on the shared boundary is placed because firstly, deep brain anatomical structures, despite being labeled as functionally separate, are continuous tracts of soft tissue and a shared boundary more accurately reflects this physical aspect of deep brain structures. Secondly, a shared boundary negates the need for potential computationally

expensive collision detection algorithms to avoid mesh overlap. It is also important to mention high-fidelity multi-surface mesh surfaces of the anatomy has broad applicability in biomedical engineering, for example in orthopedics, given that a spine model can better simulate how a load propagates if surfaces are flush with each other than in the absence of this characteristic.

1.3 Organization

This dissertation is divided into the following sections: Chapter 2 discusses some of the foundational concepts and methodologies used throughout this thesis. Chapter 3 provides details about the proposed watertight and 2-manifold Dual Contouring algorithm. Chapter 4 presents the multi-material version of the watertight and 2-manifold Dual Contouring algorithm. Section 5 introduces the concept of the multi-material 2-simplex deformable surface mesh. Chapter 6 concludes with a discussion on how the current limitations of the proposed methods might be addressed and on the future directions of the work presented here.

CHAPTER 2

FOUNDATIONAL CONCEPTS

This section will discuss relevant foundational concepts and methods that have been used throughout this research effort.

2.1 Dual Contouring

Dual Contouring (DC) [29] is a method used for extracting the surface of an implicit volume. The method is dual in the sense that vertices generated by DC are topologically dual to faces generated by the Marching Cubes [30] (MC) algorithm, as shown in Fig. 2. The DC algorithm is a hybrid of grid-based and tree-based (octrees for 3D, and quadtrees for 2D) methods. A labeled input volume is first subdivided using a uniform grid of an appropriate size. Fig. 3 (a) shows a multi-labeled 2D surface divided using a uniform grid of squares. An octree/quadtree is then used to represent and parse through the grid cubes. Fig. 3 depicts a 2D surface divided into grid squares, and its corresponding quadtree representation.

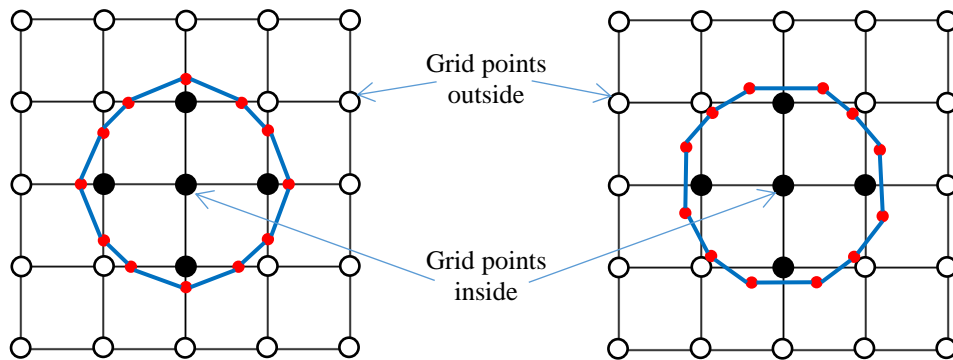


Fig. 2. An illustration of the duality, in 2D, between Marching Cubes algorithm (left) and Dual Contouring algorithm (right).

Once the volume is divided using the uniform grid, the values of the corners of each cube are noted and stored. In standard, single-material DC, the corners are labeled as inside or outside the volume. This results in a single material mesh. In multi-material DC, each corner of the cube is assigned an integer value, where each integer represents a particular material of the labeled volume. The leaves of the tree represent the grid cubes at their finest level of division, as shown in Fig. 3(b). A cube or leaf whose 8 corners all have the same material value implies that the cube is fully inside that particular material. The tree can therefore be further simplified by removing such leaves. This is demonstrated in Fig. 3(c), where the leaves numbered 2 and 4 are removed from the quadtree because all four of their corners have the same value, i.e. the cubes lie outside the volume.

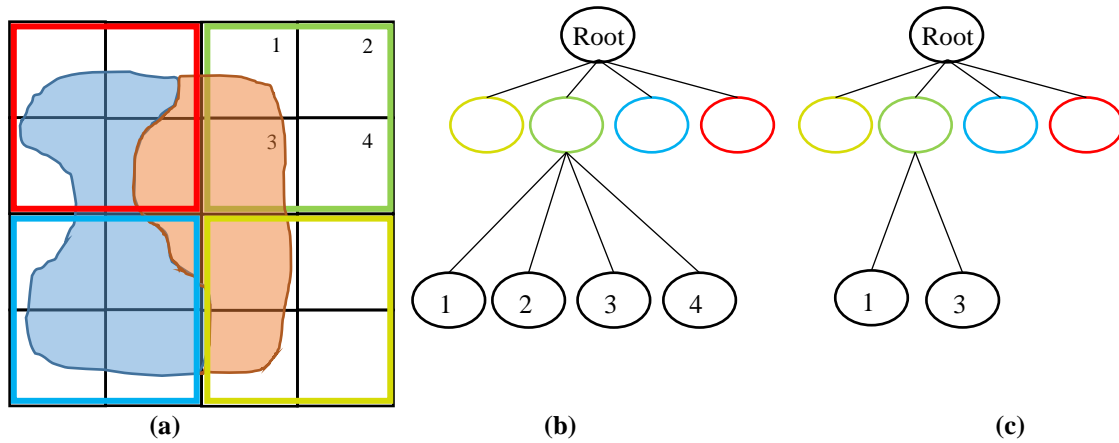


Fig. 3. Tree representation of an image. (a) A multi-labeled surface divided into grid squares, (b) its quadtree representation, and (c) the quadtree after simplification.

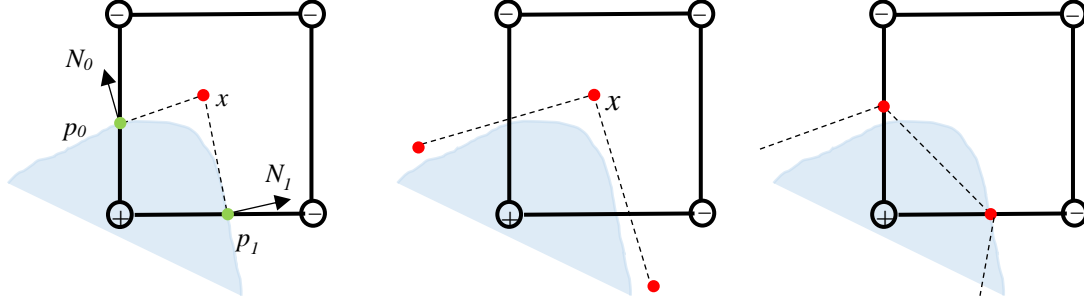


Fig. 4. Difference in vertex computation between MC and DC. (left) Formulation of Quadratic Error Functions. The blue region represents the surface/volume. (Middle) Edges as well as a sharp feature generated with DC, (right) Edges generated with MC.

For each cube that intersects the volume, *dual vertices* or *minimizers* are computed using Quadratic Error Functions (QEFs). The general formulation of a QEF is given in Equation (1).

$$E[x] = \sum_{i=1}^n ((x - p_i) \cdot N_i)^2 \quad (1)$$

Fig. 4 (left) depicts a 2D visualization of QEFs. In this figure, the bounding surface of the volume shown in light blue color intersects the lower left corner of a cube. The lower left corner of the cube is marked with a “+” sign indicating that it lies inside the volume while the remaining corners of the cube are marked with a “-” sign indicating that they lie outside the volume.

Furthermore, the surface intersects the left and bottom edges of the cube at points p_0 and p_1 (green points), respectively. If a tangent were drawn from points p_0 and p_1 and extended inside the cube, they would intersect each other somewhere inside the cube at x (red point). This point would be a vertex of the isosurface. Typically, one minimizer is computed for each grid cube that intersects the volume of interest. This minimizer can theoretically be anywhere inside the grid

cube, rather than being restricted to the edges of the cube as in MC. This feature allows DC to produce meshes with sharp features, as shown in Fig. 4 (middle), whereas MC cannot, as shown in Fig. 4 (right).

As can be seen from Equation (1) and Fig. 4 (left), the QEF depends on the intersection points and the normal at those intersection points. The function $E[x]$ can be expressed as the inner product $(Ax - b)^T(Ax - b)$ where A is a matrix whose rows are the normal N_i and b is a vector whose entries are $(N_i \cdot p_i)$. The function $E[x]$ can then be expanded into the following:

$$E[x] = x^T A^T A x - 2x^T A^T b + b^T b. \quad (2)$$

In Equation (2) $A^T A$ is a symmetric 3x3 matrix, $A^T b$ is a column vector of length three and $b^T b$ is a scalar. This representation of a QEF can be solved using the QR decomposition method of Golub and Van Loan [34], or by computing the pseudoinverse of the matrix $A^T A$ using Singular Value Decomposition (SVD) [35, 36].

Once the octree is generated and simplified, and all the minimizers for leaf cells are computed, the recursive functions **EdgeProc()**, **CellProc()** and **FaceProc()** as defined in [29] are used to locate the common *minimal edge* shared by four neighboring octree cells. The minimal edge is defined as the smallest edge shared by 4 neighboring octree cells. The concept of a minimal edge is necessary due to the fact that some octrees used in DC are adaptive octrees, meaning that the leaves of octree are not all at the same level. In such a case, it may be that the four-neighboring octree cells are not always the same size. Fig. 5 depicts the concept of a minimal edge.

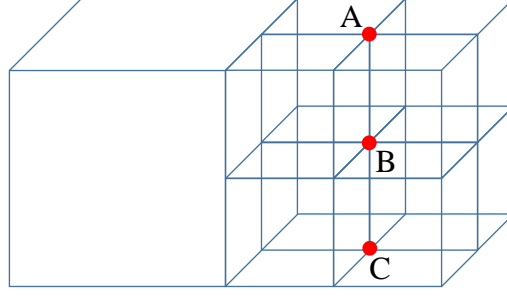


Fig. 5. The minimal edge in an adaptive octree. AB and BC are minimal edges because they do not contain any edge smaller than themselves. AC is not a minimal edge because it contains AB and BC.

The function **CellProc** receives an octree cell c as parameter, the function **FaceProc** receives two face adjacent cells, and the function **EdgeProc** receives four adjacent cells. **CellProc** recursively calls itself for each leaf of c , i.e. eight times. It then makes twelve calls to **FaceProc** with every pair of face adjacent leaf-cells. Lastly, **CellProc** makes six calls to **EdgeProc** with four leaf-cells sharing a minimal edge. **FaceProc** receives two cells sharing a common face f , and calls itself four times with every pair of leaf-cells contained in f . **FaceProc** then makes four calls to **EdgeProc** with every four leaf-cells that share an edge contained in f . **EdgeProc** receives four edge adjacent leaf-cells and makes two recursive calls to itself.

2.2 Simplex Meshes

A simplex mesh is a discrete deformable model introduced by Delingette [31, 32], and used for 3D shape representation and segmentation. A vertex in a k -simplex mesh is connected to $(k + 1)$ neighboring vertices by edges. This ensures constant vertex connectivity of the mesh.

Depending on k , a simplex mesh can be used to represent curves ($k = 1$), surfaces ($k = 2$) or volumes ($k = 3$), as shown in Fig. 6.

2.2.1 Mesh Geometry

According to [31, 32], a 2-simplex mesh M of \mathbb{R}^3 is defined as the pair $(V(M), N(M))$ where:

$$V(M) = \{P_i\}, \{i = 1, \dots, n\}, P_i \in \mathbb{R}^3 \quad (3)$$

$$N(M): \{1, \dots, n\} \rightarrow \{1, \dots, n\}^{k+1}, \quad (4)$$

$$i \rightarrow (N_1(i), N_2(i), \dots, N_{k+1}(i)),$$

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, k+1\}, \forall l \in \{1, \dots, k+1\}, l \neq j$$

$$N_j(i) \neq i \quad (5)$$

$$N_l(i) \neq N_j(i). \quad (6)$$

$V(M)$ is the set of vertices of the simplex mesh, and $N(M)$ is the associated connectivity function that represents the neighboring vertices connected to each vertex. Equations (5) and (6) are responsible for preventing loops.

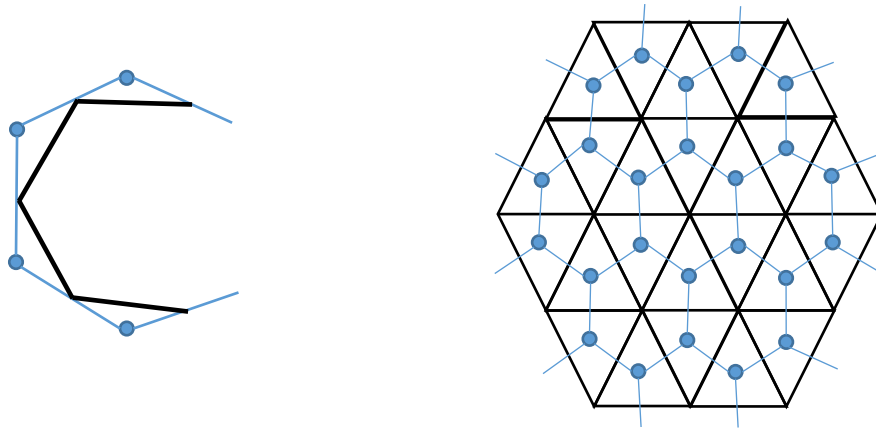


Fig. 6. Examples of Simplex meshes. (left) 1-simplex and (right) 2-simplex.

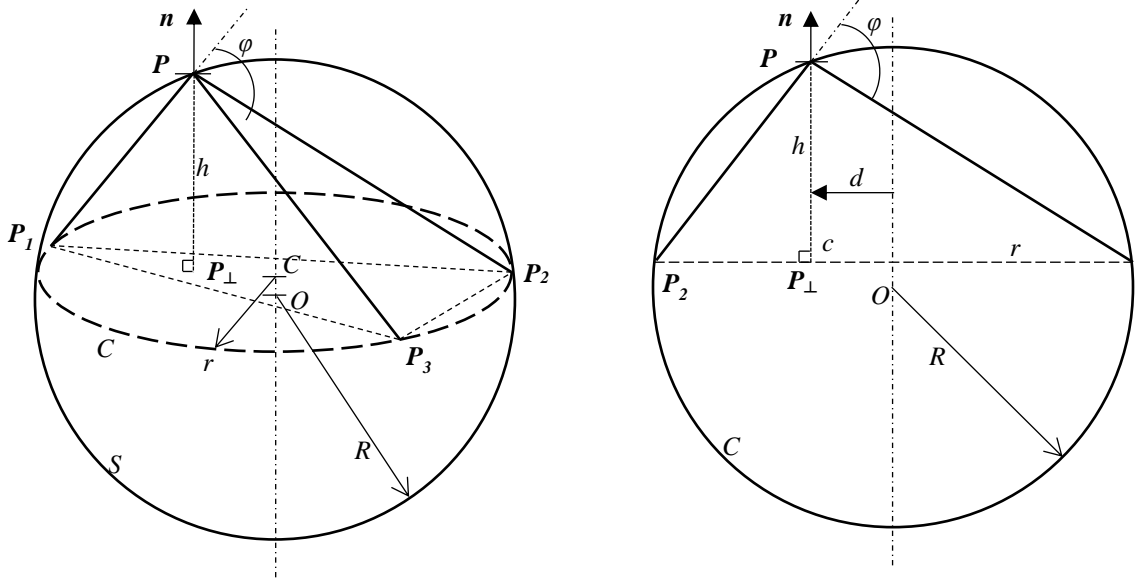


Fig. 7. Geometric representation of a vertex P and its relation to its three neighboring vertices P_1 , P_2 and P_3 in a 2-simplex mesh.

Geometrically, a vertex P of a 2-simplex mesh can be defined with respect to its three neighboring vertices P_1, P_2 and P_3 , as shown in Fig. 7. The sphere S , with its center at O and radius of R , is the circumscribed sphere containing P, P_1, P_2 and P_3 . C is the circumscribing circle, with its center at c and radius of r , containing P_1, P_2 and P_3 .

The simplex angle ϕ of the vertex P can be defined [31, 32] by the following equation:

$$\begin{aligned} \phi &\in [-\pi, \pi]: \\ \sin(\phi) &= \frac{r}{R} \text{sign}((P_1 - P) \cdot n) \\ \cos(\phi) &= \frac{\|C - O\|}{R} \text{sign}((C - O) \cdot n). \end{aligned} \tag{7}$$

Using this formulation of φ , the vertex \mathbf{P} can be defined with respect to its neighbors in the following manner:

$$\mathbf{P}(\epsilon_1, \epsilon_2, \varphi) = \epsilon_1 \mathbf{P}_1 + \epsilon_2 \mathbf{P}_2 + (1 - \epsilon_1 - \epsilon_2) \mathbf{P}_3 + L(\varphi) \cdot \mathbf{n}, \quad (8)$$

$$\mathbf{n} = \frac{\mathbf{P}_1 \mathbf{P}_3 \wedge \mathbf{P}_1 \mathbf{P}_2}{\|\mathbf{P}_1 \mathbf{P}_3 \wedge \mathbf{P}_1 \mathbf{P}_2\|}, \quad (9)$$

$$\epsilon_1 + \epsilon_2 + \epsilon_3 = 1, \quad (10)$$

$$L(\varphi) = \frac{(r^2 - d^2) \tan(\varphi)}{r + \epsilon \sqrt{r^2 + (r^2 - d^2) \tan^2(\varphi)}}, \quad d \leq r \quad (11)$$

$$\epsilon = \begin{cases} 1, & \text{if } \varphi < |\pi/2| \\ -1, & \text{if } \varphi > |\pi/2| \end{cases}$$

$$r = \|\mathbf{C} \mathbf{P}_1\|, \quad d = \|\mathbf{C} \mathbf{P}_\perp\|.$$

Equation (10) represents the metric parameters ϵ_1 , ϵ_2 and ϵ_3 .

Gilles in [33] contends that the simplex angle φ can have ambiguous values in situations where $d \leq r$ does not hold true, and uses the elevation h of \mathbf{P} above \mathbf{P}_\perp as a meaningful substitute. The elevation h is more accurately equated as:

$$h_p = h St^{-1/\alpha}. \quad (12)$$

In Equation (12), St refers to the area of the triangle formed by the three neighboring vertices \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 , and α tunes the scale invariant aspect. The vertex \mathbf{P} can be newly defined as:

$$\mathbf{P}(\epsilon_1, \epsilon_2, h_p) = \epsilon_1 \mathbf{P}_1 + \epsilon_2 \mathbf{P}_2 + (1 - \epsilon_1 - \epsilon_2) \mathbf{P}_3 + h_p St^{1/\alpha} \mathbf{n}, \quad (13)$$

$$St = \frac{\|\mathbf{P}_1 \mathbf{P}_3 \wedge \mathbf{P}_1 \mathbf{P}_2\|}{2}. \quad (14)$$

2.2.2 Mesh Forces

The simplex mesh is a mass-spring type of deformable surface model, meaning that the position of each vertex is governed by the general Newtonian law of motion, shown in Equation (15).

$$m_i \frac{d^2 \mathbf{P}_i}{dt^2} = -\gamma \frac{d\mathbf{P}_i}{dt} + \mathbf{F}_{int} + \mathbf{F}_{ext}. \quad (15)$$

In the above equation, m_i is the mass, and \mathbf{P}_i is the position of a vertex of the mesh. In Equation (15), \mathbf{F}_{int} represents all internal forces and \mathbf{F}_{ext} represents all the external forces acting on \mathbf{P}_i .

Delingette in [31, 32] defines the internal force \mathbf{F}_{int} as a summation of normal forces and tangential forces, as shown in Equation (16).

$$\mathbf{F}_{int} = \mathbf{F}_{Tangent} + \mathbf{F}_{Normal}. \quad (16)$$

The purpose of the tangential internal force is to provide control for the vertex position with respect to its three neighbors \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 in the tangent plane. Fig. 8 shows the relationship between $\mathbf{F}_{Tangent}$ and \mathbf{F}_{Normal} with respect to the three neighbors \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 of the vertex \mathbf{P}_i .

The tangential force $\mathbf{F}_{Tangent}$ and the normal force \mathbf{F}_{Normal} are defined as:

$$\mathbf{F}_{Tangent} = (\tilde{\epsilon}_1 - \epsilon_1)\mathbf{P}_1 + (\tilde{\epsilon}_2 - \epsilon_2)\mathbf{P}_2 + (\tilde{\epsilon}_3 - \epsilon_3)\mathbf{P}_3, \quad (17)$$

$$\mathbf{F}_{Normal} = (L(r_i, d_i, \tilde{\varphi}_i) - L(r_i, d_i, \varphi_i)) \mathbf{n}. \quad (18)$$

where $\tilde{\epsilon}_1$, $\tilde{\epsilon}_2$ and $\tilde{\epsilon}_3$ are reference metric parameters corresponding to the prescribed value of the metric parameters, and the function $L(r_i, d_i, \varphi_i)$ is as defined in Equation (11).

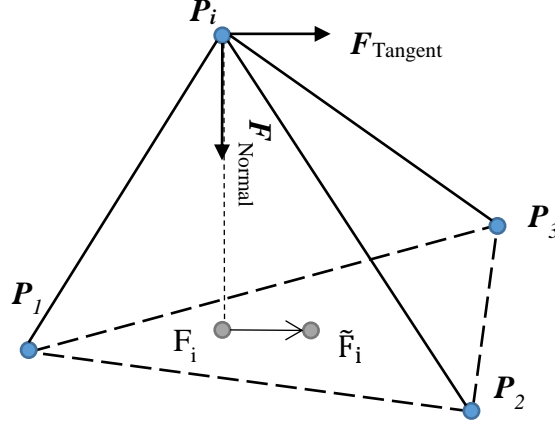


Fig. 8. Relationship between $\mathbf{F}_{\text{Normal}}$ and $\mathbf{F}_{\text{Tangent}}$ with respect to the neighbors of point \mathbf{P}_i .

Gilles [33] defines additional internal forces based on Laplacian smoothing, shape memory and global volume preservation, as follows respectively:

$$\tilde{\mathbf{P}} = \frac{s_1 \mathbf{P}_1 + s_2 \mathbf{P}_2 + s_3 \mathbf{P}_3}{s_1 + s_2 + s_3} \langle h_i \rangle_\eta \mathbf{n}, \quad (19)$$

$$\tilde{\mathbf{P}} = \tilde{\epsilon}_1 \mathbf{P}_1 + \tilde{\epsilon}_2 \mathbf{P}_2 + (1 - \tilde{\epsilon}_1 - \tilde{\epsilon}_2) \mathbf{P}_3 + \tilde{h}_p St^{1/\alpha} \mathbf{n}, \quad (20)$$

$$\tilde{\mathbf{P}} = \mathbf{P} + \frac{\tilde{V} - V}{S} \mathbf{n}. \quad (21)$$

Equation (19) represents the barycentric weighted Laplacian smoothing based internal force where s_1 , s_2 , and s_3 represents the surfaces associated with the neighboring vertices \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 . The operator $\langle \cdot \rangle_\eta$ represents the averaging around the neighborhood η of vertex \mathbf{P} . $\langle h_i \rangle_\eta$ therefore represents the average elevation around the neighborhood of \mathbf{P} .

A reference shape model can be used to represent prior shape information in deformation. In Equation (20), $\tilde{\epsilon}_1$, $\tilde{\epsilon}_2$ and \tilde{h}_p represent predefined metric parameters.

Volume preservation can be used to exploit the fact that biological tissue is incompressible. In Equation (21), \tilde{V} represents the target volume, V represents the current volume and S represents the surface area of the closed mesh.

In the context of segmentation of anatomical structures, the main criterion by which an organ or anatomical structure is identified, without human assistance, is through the use of image gradients that delineate the boundaries of the target structure. For the 2-simplex discrete deformable mesh, external forces are derived from maximal image gradients. The idea is to derive a force function based on the gradient values around the neighborhood of a vertex's position. The neighborhood of a vertex's position can be defined as a fixed space along the bidirectional surface normal of the vertex. For a vertex \mathbf{P} , with normal \mathbf{n} , and a predefined stepsize s , the sample points for the image gradient can be defined as:

$$\tilde{\mathbf{P}} = \mathbf{P} + j\mathbf{s}\mathbf{n}, \quad (22)$$

where j is an integer representing optimal shifts in the surface normal direction. An interpolation method such as trilinear interpolation can be used to interpolate values between voxels of gradient image intensities. The maximal gradient magnitude at \tilde{j} can be obtained using:

$$\tilde{j} = \underset{-\frac{d}{s} < j < \frac{d}{s}}{\operatorname{argmin}} (\|\nabla T(\mathbf{P} + j\mathbf{s}\mathbf{n})\|), \quad (23)$$

where ∇T represents the image gradient of the input image T .

In order to achieve model-to-image registration, the goal is to align a source image S to the target T using model deformations. The model, initially aligned to S , and whose initial vertex positions are given by \mathbf{P}_0 , is iteratively deformed until each vertex matches the target, where a similarity metric Δ in the vertex neighborhood η is maximal:

$$\tilde{j} = \underset{-\frac{d}{s} < j < \frac{d}{s}}{\operatorname{argmin}} \Delta \left(S(\eta(\mathbf{P}_0)), T(\eta(\mathbf{P} + j\mathbf{s}\mathbf{n})) \right). \quad (24)$$

As mentioned above, the vertex neighborhood η is constructed by sampling points using Equation (22) along the direction of the surface normal \mathbf{n} .

2.2.3 Mesh Evolution

As described above, each vertex of a 2-simplex mesh can be considered as a particle with a mass m , and obeying the Newtonian law of equilibrium. The internal and external forces act upon the vertex, thus altering the shape of the mesh. The Newtonian law of motion can be decomposed into the first order differential equation system:

$$\begin{aligned}\mathbf{V} &= \frac{\partial \mathbf{P}}{\partial t}, \\ M \frac{\partial \mathbf{V}}{\partial t} &= \mathbf{F}(\mathbf{P}, \mathbf{V}),\end{aligned}\tag{25}$$

where \mathbf{V} , \mathbf{P} and M represent the velocity, position and mass of the vertex, respectively. The function \mathbf{F} is the force vector, which depends on the velocity and position. This can also include internal and external forces, as well as damping forces which help to reduce oscillations in the system.

Discretizing the first order differential system yields the following:

$$\begin{aligned}\mathbf{P}_{t+dt} - \mathbf{P}_t &= \mathbf{V}dt, \\ \mathbf{V}_{t+dt} - \mathbf{V}_t &= \frac{\mathbf{F}(\mathbf{P}, \mathbf{V})}{M}dt.\end{aligned}\tag{26}$$

In the manner of [37], a state vector $\mathbf{Q}_t = [\mathbf{P}_t, \mathbf{V}_t]$ can be used to rewrite the first order differential system as follows:

$$\mathbf{Q}_{t+dt} - \mathbf{Q}_t = \mathbf{Q}'_{t+dt}dt,\tag{27}$$

and applying the first order Taylor expansion yields:

$$\mathbf{Q}_{t+dt} - \mathbf{Q}_t = \mathbf{Q}'_tdt + \alpha \frac{\partial \mathbf{Q}'}{\partial \mathbf{Q}}(\mathbf{Q}_{t+dt} - \mathbf{Q}_t)dt,\tag{28}$$

where α represents the implicit coefficient. When $\alpha = 0$, the system represents the explicit Forward Euler method, which is unstable for larger step sizes. On the other hand, $\alpha = 1$ represents the implicit Backward Euler method which is stable for any step size. When applied to a mechanical system, Equation (28) leads to:

$$\begin{aligned} \mathbf{P}_{t+dt} - \mathbf{P}_t &= (\mathbf{V}_t + \alpha(\mathbf{V}_{t+dt} - \mathbf{V}_t))dt, \\ \mathbf{V}_{t+dt} - \mathbf{V}_t &= \mathbf{H}^{-1}\mathbf{Y}, \end{aligned} \quad (29)$$

$$\begin{aligned} \mathbf{H} &= \mathbf{I} - \alpha\mathbf{M}^{-1}\frac{\partial\mathbf{F}}{\partial\mathbf{V}}dt - \alpha^2\mathbf{M}^{-1}\frac{\partial\mathbf{F}}{\partial\mathbf{P}}dt^2, \\ \mathbf{Y} &= \mathbf{M}^{-1}\mathbf{F}(\mathbf{P}_t, \mathbf{V}_t)dt + \alpha\mathbf{M}^{-1}\frac{\partial\mathbf{F}}{\partial\mathbf{P}}\mathbf{V}_tdt^2. \end{aligned} \quad (30)$$

The resolution of this system depends on the inversion of $\mathbf{H}\mathbf{X} = \mathbf{Y}$ using the conjugate gradient method described in [38], and used in [33, 39]. The general algorithm is as follows:

```

 $\beta = 0, \mathbf{X} = 0, \mathbf{R} = \mathbf{Y}$ 
while  $\beta > \varepsilon$ 
     $\alpha = \mathbf{R}^T\mathbf{R}$ 
    if  $(\beta \neq 0)$  then
         $\mathbf{T} = \mathbf{R} + (\alpha/\beta)\mathbf{T}$ 
    else
         $\mathbf{T} = \mathbf{R}$ 

     $\beta = \mathbf{T}^T\mathbf{H}\mathbf{T}$ 
     $\mathbf{R} = \mathbf{R} - (\alpha/\beta)\mathbf{H}\mathbf{T}$ 
     $\mathbf{X} = \mathbf{X} + (\alpha/\beta)\mathbf{T}$ 
     $\beta = \alpha$ 

```

Here, the error factor is β , and the predefined threshold is ε . The algorithm iterates until the error factor becomes less than the threshold.

2.2.4 Topological Operators

Delingette introduces four topological operators for the 2-simplex mesh in [31, 32]: TO_1 , TO_2 , TO_3 and TO_4 . These operators can be used to improve the topological quality of the simplex mesh (uniformity among vertices and cells). Out of these four operators, TO_1 and TO_2 preserves the overall topology of the mesh while TO_3 and TO_4 alters the topology.

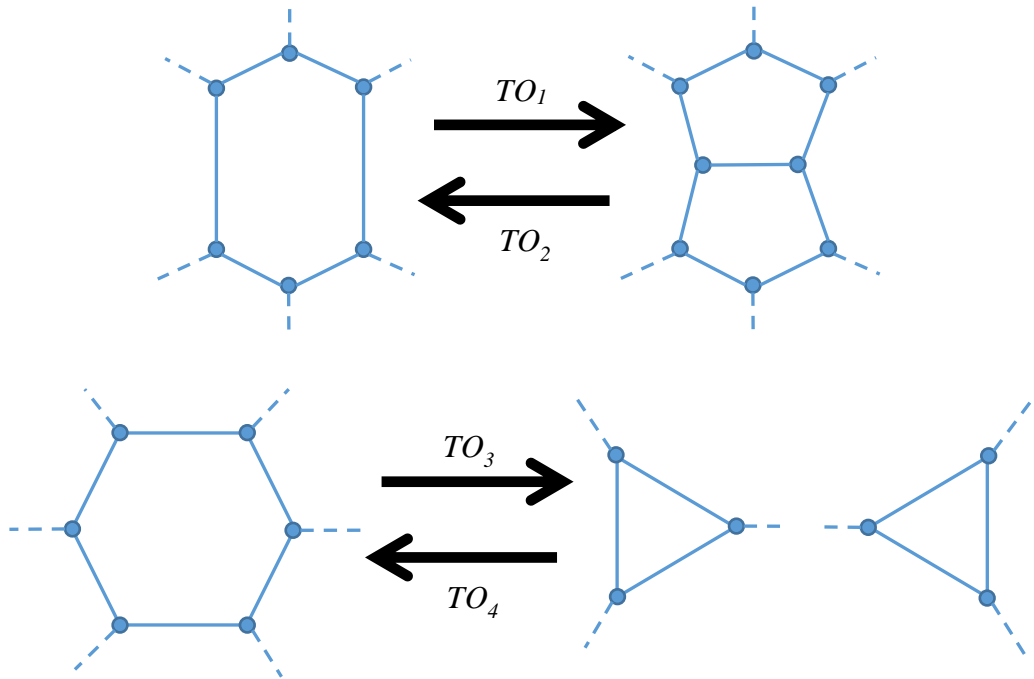


Fig. 9. Topological operators for the 2-simplex mesh.

As can be seen in Fig. 9, a TO_2 operation has the effect of splitting a cell into 2 adjacent cells, and conversely, a TO_1 operation can be used to merge two adjacent cells into one. These features can allow a user to potentially use the TO_1 and TO_2 operators for targeted or localized mesh decimation and resolution control while still preserving mesh topology.

2.3 Volume Smoothing

For many applications involving images (and/or volumes), it is sometimes necessary to perform some manner of preprocessing on the images in order to simplify them, or enhance desirable features, or remove unwanted artifacts and noise. In this section, two methods for volume smoothing are described: binary morphology and Gaussian blurring. In this research, binary morphology was used to smooth single material or binary images, and Gaussian blurring was used to smooth multi-material volumes.

2.3.1 Binary Morphology

Mathematical morphology is based on set theory, and describes techniques and operations that can be applied to an image in order to affect changes to the image. Operations such as region filling, closing and opening are performed on the image to enhance or simplify the image, whereas operations such as boundary extraction and skeletonization are performed on the image in order to generate new information about the image.

Binary morphology describes morphological operations applied to binary images. The two fundamental operations, as described in [40], are erosion and dilation. For an image A and a structuring element B , dilation is defined as:

$$A \oplus B = \{z | B_z \cap A \neq \emptyset\}, \quad (31)$$

and erosion is defined as:

$$A \ominus B = \{z | B_z \cap A \subseteq A\}. \quad (32)$$

Dilation has the effect of adding pixels or voxels to the image A and erosion has the effect of removing pixels or voxels from A , depending on the shape and size of the structuring

element. The structuring element is typically a simple geometric shape such as a cube or sphere or diamond, though other shapes can be defined and used.

Based on these two fundamental operators, two other operations, opening and closing, are defined, respectively, as:

$$A \circ B = (A \ominus B) \oplus B, \quad (33)$$

and

$$A \cdot B = (A \oplus B) \ominus B. \quad (34)$$

In Equation (33) and Equation (34), the set A represents the image of interest and the set B is the structuring element. Opening describes an erosion operation on A , followed by a dilation operation on the eroded A , using the structuring element B . Opening smooths the image by removing thin regions and protrusions. Closing describes a dilation operation on A , followed by an erosion operation on the dilated A , using the structuring element B . Closing smooths the image by removing holes and filling gaps. Both the opening and closing operations, by themselves, are referred to as *unidirectional* or *biased* operations, in the sense that each operation removes only one type of “noise” [41]. In order to create a bidirectional and unbiased smooth image, it is necessary to apply the opening and closing operations in sequence using the same structuring element. The sequence of operations can be either closing followed by opening, or opening followed by closing. The resulting smoothed images of open-close and close-open operations are similar but not identical.

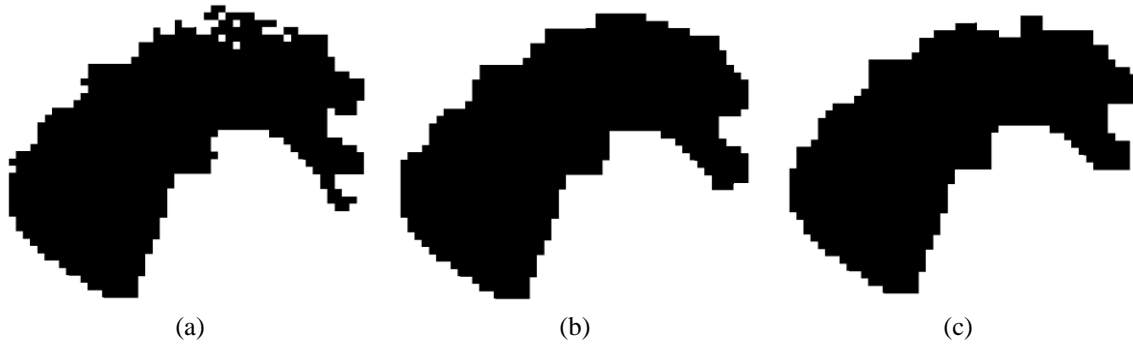


Fig. 10. An example of binary morphology. (a) Original volume, (b) smoothing using close-open sequence, (c) smoothing using open-close sequence.

Fig. 10 show an example of using the open-close and close-open sequence for smoothing. Fig. 11 shows the surface meshes of the volumes in Fig. 10. All the surface meshes in this figure were generated using the proposed Dual Contouring method described in Chapter 3. Fig. 11(a) shows the surface mesh of the original unsmoothed volume, Fig. 11(b) shows the surface mesh of the volume smoothed by first applying closing followed by opening, and Fig. 11(c) shows the surface mesh of the volume smoothed by first applying opening followed by closing. The structuring element used is a 3x3x3 cube. The square inset in Fig. 11 shows an example of a small feature, in this case a loop-like structure, and the dashed circle shows a sharp protrusion, that can appear on the surface mesh if the volume is left unsmoothed. As can be seen in Fig. 11, both close-open and open-close smoothing results in volume that is similar, but not identical.

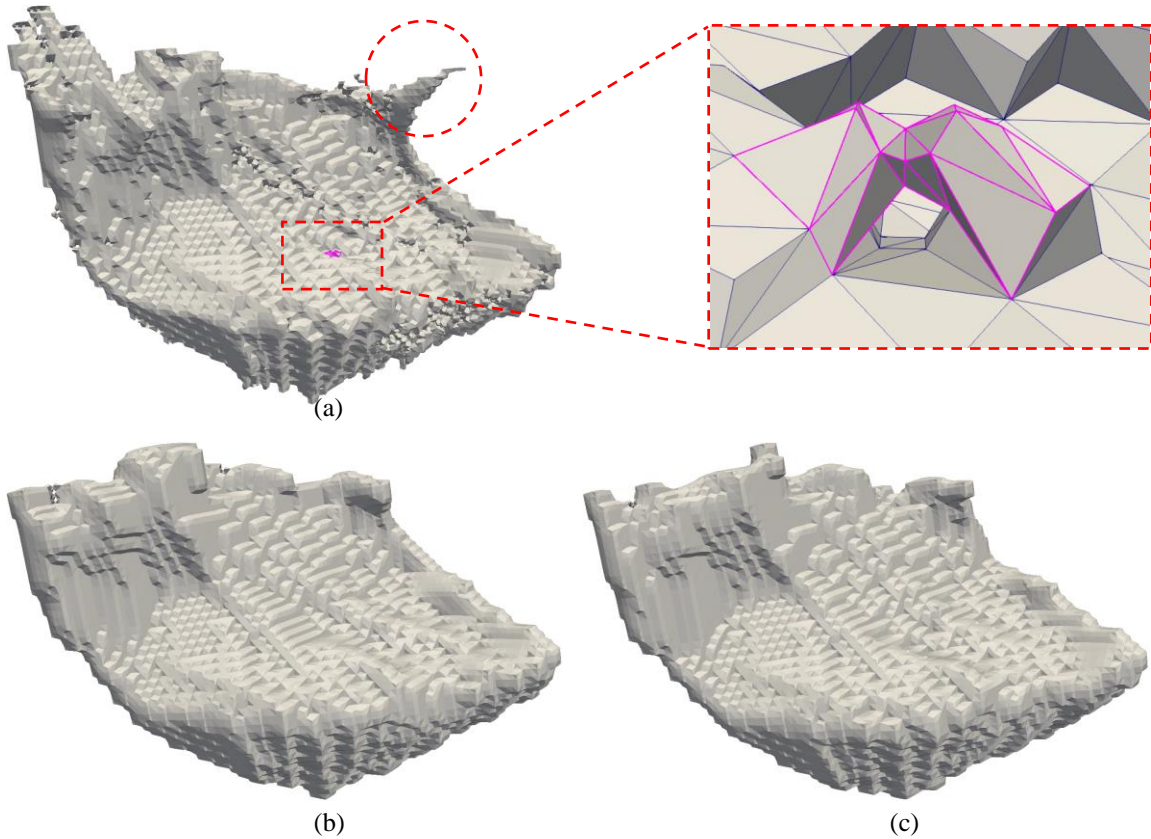


Fig. 11. Surface meshes of a volume smoothed using binary morphology operations. (a) Mesh of original unsmoothed volume, (b) mesh of volume smoothed using closing followed by opening, (c) mesh of volume smoothed using opening followed by closing.

2.3.2 Multi-Material Volume Smoothing Using Gaussian Blurring

Although binary morphology offers a powerful set of tools for image smoothing, they can only be readily used in conjunction with binary datasets. In the case of multi-label or multi-material datasets, it is necessary to first isolate each label or material into separate binary volumes, apply binary morphology-based smoothing to each of the separated material volumes, and then recombine all the separate material volumes into a single, smoothed multi-material

volume. Since binary morphology both removes and adds foreground voxels to the image, it is conceivable that empty voxels, or voxels with conflicting material values, may occur during the recombination phase for voxels that are adjacent to another material. This problem can be somewhat mitigated using a *voting* algorithm that determines the value of the problematic voxel based the values of neighboring voxels. However, it is possible that the new voxels inserted by the voting algorithm can lead to undesired features in the resulting image.

Another approach to smoothing multi-material volumes is through the use of blurring kernels, such as a Gaussian kernel which acts as a low-pass filter. The smoothing process begins by first generating separate binarized volumes for each material, and then applying a Gaussian blurring filter with a specified variance to each material volume. For this research, the opensource ITK's (Insight Segmentation and Registration Toolkit) *DiscreteGaussianImageFilter* was used to achieve Gaussian blurring. The surface of the structures represented in the material volumes would now be blurred, having a real-numbered value between 0 and 1, with the values decreasing as the edge moves away from the center of the structure. It is therefore necessary to apply a threshold in order to reconstruct the smoothed surface. The default threshold value is 0.5, but higher or lower values based on user inspection can be used as well. The thresholded volumes are then recombined into a single multi-material volume.

Since the volume being smoothed is a multi-material volume, there are two types of surfaces encountered: the surface between a material and the background, and the surface between two materials. An alternative, therefore, is to use two different threshold values, one for voxels between the material and background (*mat2bkgnd*), and another for voxels between two materials (*mat2mat*).

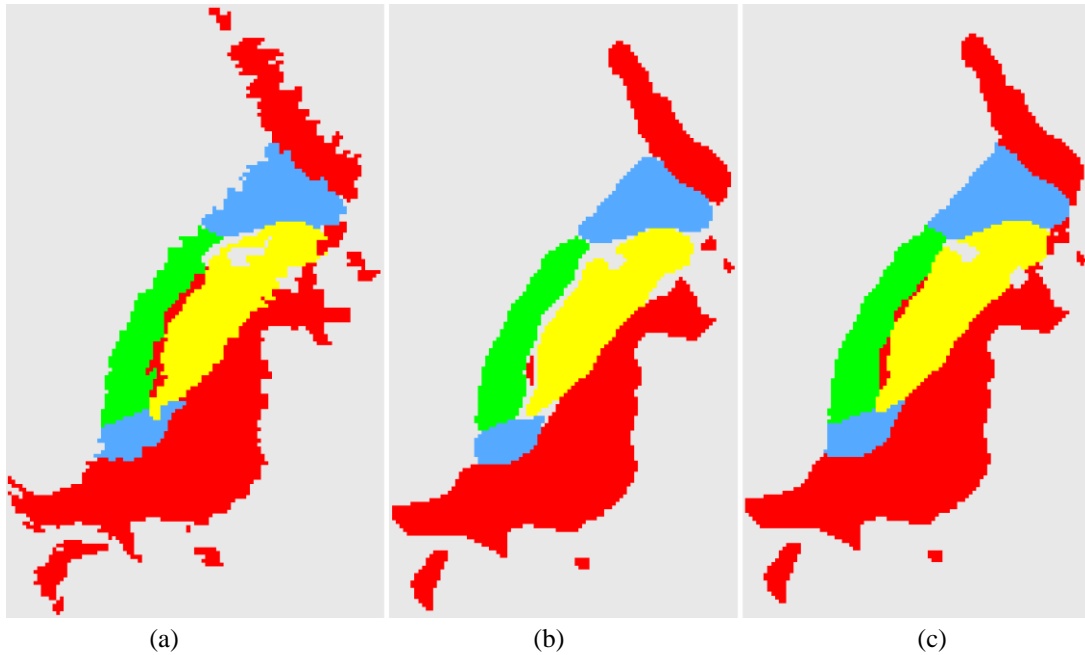


Fig. 12. An example of multi-material smoothing using Gaussian blurring. (a) Original multi-material volume, (b) threshold values $\text{mat2bkgnd} = 0.5$, $\text{mat2mat} = 0.5$, (c) threshold values $\text{mat2bkgnd} = 0.5$, $\text{mat2mat} = 0.3$.

Fig. 12 shows a demonstration of the implemented multi-material smoothing using a Gaussian blurring kernel. Fig. 12(a) shows the original and unaltered volume. Fig. 12(b) shows the resulting smoothed volume where the threshold was kept the same at 0.5, and Fig. 12(c) shows the resulting smoothed volume where the mat2bkgnd threshold is 0.5 and the mat2mat threshold is 0.3. As can be seen, there is a noticeable difference when using a lower value for the mat2mat threshold. In Fig. 12(a) the thin region between the green and yellow labels is originally labeled red, but in Fig. 12(b) the thin region is completely removed due to the higher mat2mat threshold value being used. In Fig. 12(c) the thin region is reconstructed due to the use of a smaller mat2mat threshold value.

CHAPTER 3

2-MANIFOLD DUAL CONTOURING

Surface meshing is an invaluable tool and one of the most commonly used methods in scientific research for visualizing volumetric data. A surface mesh of a real-world object can be generated in one of two ways: (1) by using a scanning device such as the NextEngine 3D Laser Scanner or Microsoft's Kinect, or (2) by isosurface extraction from volumetric data such as MRI or CT using contouring algorithms such as *Marching Cubes* (MC) [30], *Dual Contouring* (DC) [29] or *Dynamic Particle Systems* [42]. In both cases, the resulting polyhedral mesh may contain geometric errors such as non-manifold edges and/or vertices, holes, and intersecting polygons, especially if the surface being meshed is complex. A survey performed by Ju in [43] discusses the wide range of techniques that has been developed for repairing polygonal models.

Non-manifold geometry is problematic for a variety of situations, such as rendering of refractive surfaces, computation of surface normals and curvatures, bounding tetrahedral meshes suitable for finite element analysis and fluid simulations, as well as CAD-based manufacturing and 3D printing. The repairing of geometric errors in meshes is an active research area and there is no one-fits-all algorithm that can fix all the different types of geometric errors. Of course, this is not to say that topologically and geometrically correct surface mesh generation is a poorly researched field. Reference [44] presents an extensive review of the many variants of the MC algorithm that have been developed over the years. *Tight cocone* [45] is a meshing algorithm that guarantees watertight meshes. *Marching Tetrahedra* [46] is another method similar to MC that can produce topologically correct meshes.

This chapter focuses primarily on Dual Contouring. DC offers the advantage of producing meshes with sharp features [29]. In MC, the newly created vertices are constrained to

the edges of the grid while in DC, the vertices can be anywhere inside the grid cube. However, the standard DC algorithm produces non-manifold edges and vertices in certain situations. Presented in this work is a modification to the Dual Contouring algorithm that is capable of generating 2-manifold meshes and thereby avoid non-manifold geometric errors in the first place.

3.1 Background and Literature Review

One of the main disadvantages of DC is that it does not guarantee 2-manifold, watertight, and intersection-free surfaces. A polygonal mesh is considered as being 2-manifold if each edge of the mesh is shared by only two faces, and if the neighborhood of each vertex of the mesh is the topological equivalent of a disk. Ju and Udeshi address the issue of intersecting triangles in [47] by proposing a hybrid method where dual vertices (inside grid cubes) as well as face vertices and edge vertices (inserted on the cube's face and edges, respectively) are used to create polygons according to new polygon generation rules. Zhang et al. in [48] present a topology-preserving algorithm for surface simplification using vertex clustering and an enhanced cell representation, but this method is unable to avoid non-manifold edges and vertices. Varadhan et al. [49] suggest an approach that combines edge intersection testing, adaptive subdivision, and dual contouring to reconstruct thin features. Schaefer et al. use a vertex clustering method in [14], where they present an additional topology criterion that must be satisfied to ensure manifoldness.

Zhang and Qian in [50] take a different approach by first generating a base mesh using standard DC, and then analyzing and categorizing the leaves into 31 topology groups. For *ambiguous* cubes, multiple minimizers, as many as three in some instances, are inserted whereby

a new topologically correct mesh is created by reconnecting the vertices of the mesh with the newly inserted minimizers. Zhang and Qian [51] also use a topology-preserving decomposition method to divide ambiguous cubes into twelve tetrahedral cells, each having one minimizer (the centroid), and construct a series of tetrahedra and polyhedra to create tetrahedral meshes. This method can avoid topological ambiguities in tetrahedral meshes but does not produce surface meshes. The chief advantage in using surface meshes is the fact that compared to volumetric meshes like tetrahedral and hexahedral meshes, surface meshes are sparser. Surface meshes consist only of vertices and cells delineating the surface boundary whereas volumetric meshes have vertices and cells describing the internal volume in addition to vertices and cells for the surface boundary. This makes surface meshes a more logical choice to reduce computational overhead.

The proposed method uses an approach similar to that in [51] by decomposing an ambiguous cube into several tetrahedral cells. Novel polygon generation rules are presented that result in 2-manifold and watertight triangular surface meshes.

3.2 2-Manifold Dual Contouring

The proposed method in this dissertation begins the same way as in standard Dual Contouring (DC) by superimposing a uniform virtual grid onto the implicit volume. Depending on the isovalue chosen, the corners of each cube of the grid can have 2^8 or 256 possible configurations. By taking rotation and symmetry into account, these configurations can be reduced into 14 fundamental cases, as shown in Fig. 13. Cases 0, 1, 2, 5, 8, 9 and 11 are simple unambiguous cases, meaning there is only one possible surface intersecting the grid cube (no surface for Case 0). Cases 3, 4, 6, 7, 10, 12 and 13 are ambiguous, meaning that there is more

than one possible surface that intersects the cube. It is the presence of these ambiguous cubes, as well as the fact that standard DC produces only one minimizer for each cube, that causes non-manifold surfaces to arise. Additionally, the experiments have found that the complement of Case 4 (that is, a situation where the two diagonally opposite corners of the cube are in background and the rest are in the foreground) is also responsible for the generation of non-manifold vertices, as shown in Fig. 14. These particular non-manifold vertices occur inside the surface mesh. In [52], Sohn shows that a cubic cell can be decomposed into a set of tetrahedral cells while preserving the topology of the isosurface inside the cube. However, the number of tetrahedral cells created, as well as their shapes and sizes, is dependent on the number of face and body saddle points. The tetrahedral decomposition method in [51] is much more advantageous because all ambiguous cubes are decomposed into a maximum of 12 tetrahedral cells of similar shape and size.

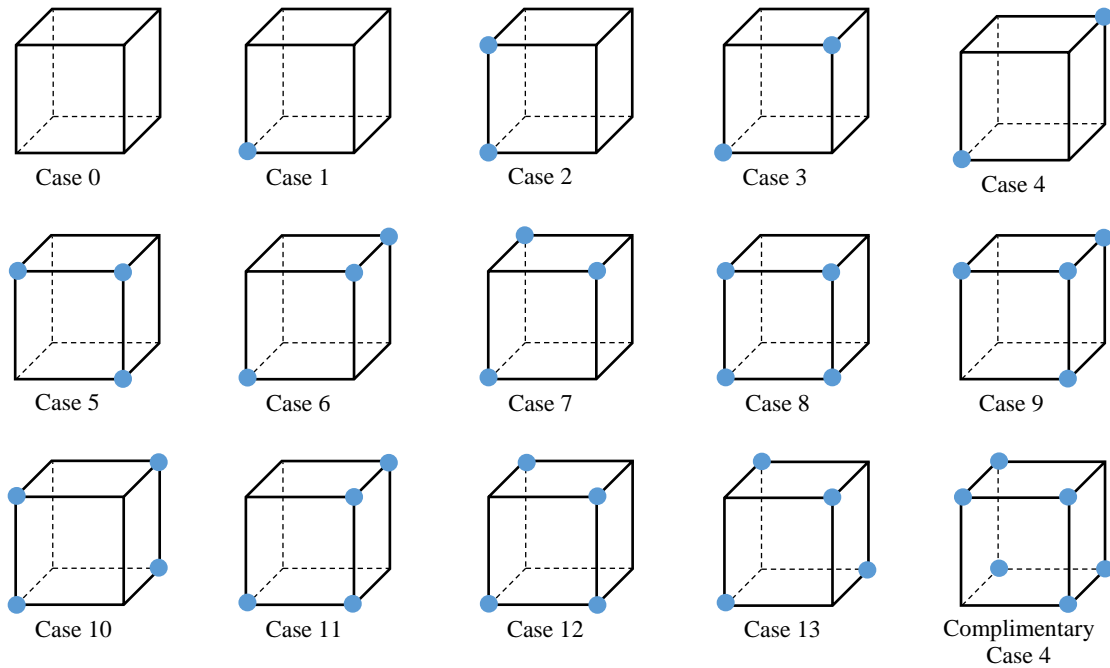


Fig. 13. The 14 fundamental configurations for a grid cube.

The solution presented in this dissertation involves first detecting ambiguous cubes and then subdividing the cube into several tetrahedral cells in a manner similar to [51]. The centroid of the tetrahedral cells are used as minimizers. This approach allows an ambiguous cube to have multiple minimizers, *thus overcoming the one minimizer limitation of standard DC*. More details on the exact mechanism of tetrahedral decomposition is presented in the following section.

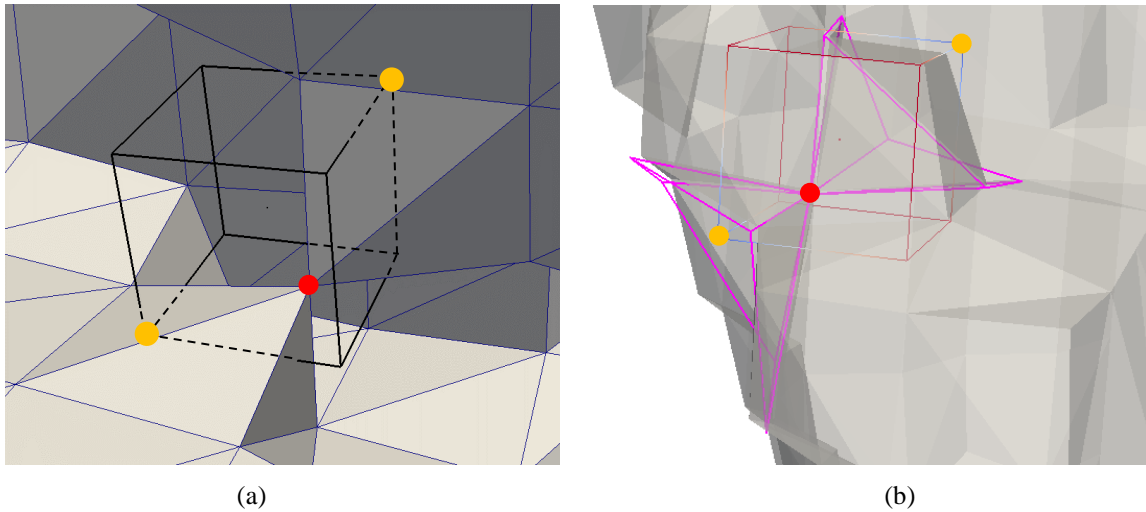


Fig. 14. Presence of a non-manifold vertex due to complimentary Case 4. (a) This is an inside view of the surface mesh with the non-manifold vertex highlighted by the red dot. (b) This is a view from outside the surface mesh (rendered transparent). The highlighted polygons share the non-manifold vertex. The two yellow dots represent the two diagonally opposite corners of the cube and they lie outside the implicit volume.

3.3 Tetrahedral Decomposition of Ambiguous Cubes

The volume is first subdivided into a uniform grid of an appropriate size. An octree whose nodes represent the grid cubes is then used for fast parsing of the grid cubes and polygon generation. Each corner of a grid cube is assigned an inside or outside label based on its location within the volume of interest. For each unambiguous cube, one minimizer is computed.

In the case of ambiguous cubes, the approach followed is similar to that of Zhang and Qian in [51] and the ambiguous cube is subdivided into a maximum of twelve tetrahedral cells. The center of the cube is a common point for all the newly created tetrahedral cells, as shown in Fig. 15(a) and Fig. 15(d). A tetrahedral cell is made up of the center along with three corners of a

cube face. Each face of the ambiguous cube forms the base of two tetrahedra by joining the diagonally opposite corners of the face. Fig. 15(b) and Fig. 15(e) illustrates two ways for creating the diagonal, which is further explained below:

- If the two diagonally opposite corners are contiguous with each other through the interior of the volume, then create a diagonal between the two corners, shown in Fig. 15(b).
- If the two diagonally opposite corners are inside the volume, but not contiguous with each other through the interior, then create a diagonal using the other two corners Fig. 15(e).
- For all other cases, any appropriate diagonal can be used.

The choice of creating the diagonal is important because the resulting polygonization can lead to topological changes. Fig. 15(c) depicts a situation where the two corners of the bottom face of the ambiguous cube are contiguous with each other inside the volume, and the diagonal is created as shown in Fig. 15(b). Fig. 15(f) shows a situation where the two corners of the bottom face of the ambiguous cube are inside the volume, but not contiguous through the interior, and the diagonal for the bottom face is created as in Fig. 15(e). In both examples, the center of the chosen face is sampled to determine whether that point lies inside or outside the volume. This rule for creating a face diagonal is important because it allows adjacently situated ambiguous cubes to have consistent face diagonals.

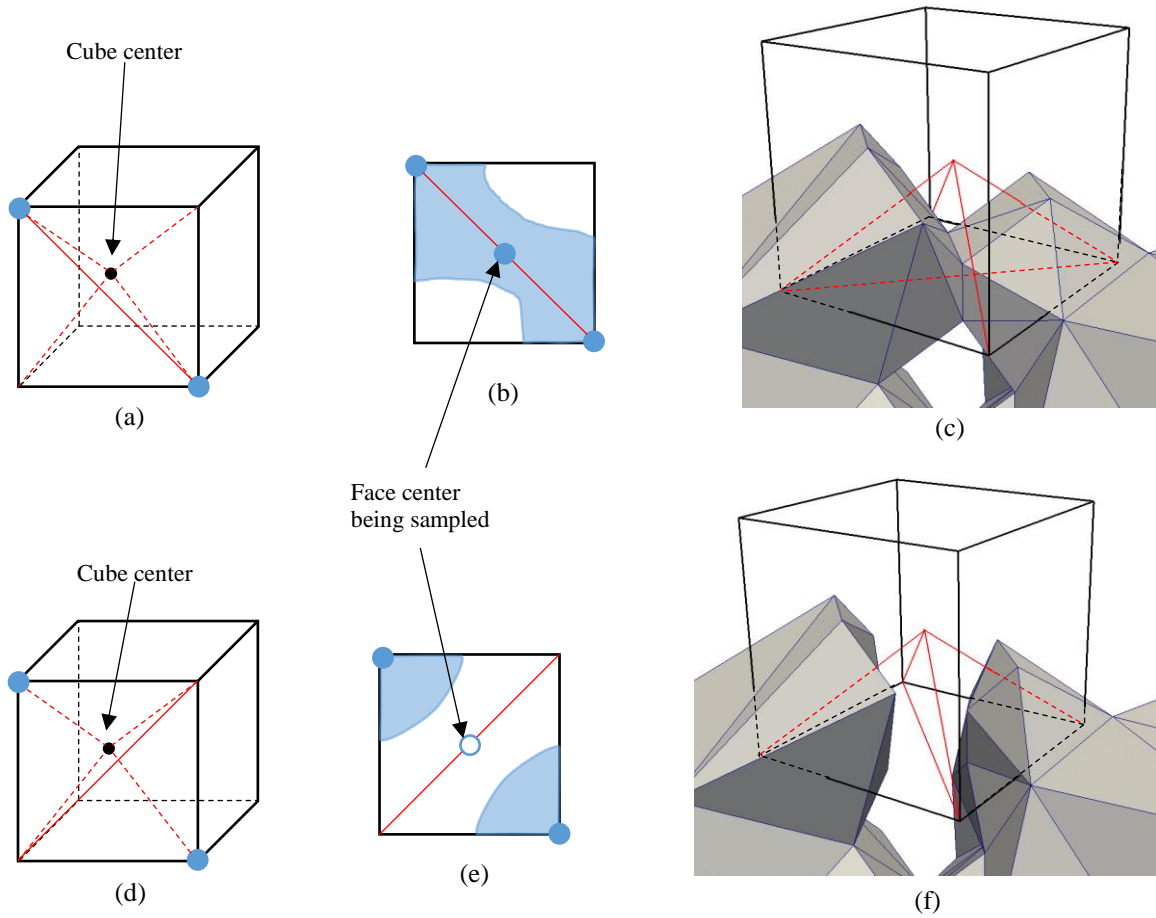


Fig. 15. Creation of the face diagonal for ambiguous cubes. (a, d) Two ways in which a diagonal can be created on the front-most face of an ambiguous cube to generate two tetrahedra. (b) The corners of the cube are contiguously inside the volume, (e) the corners of the cube are inside the volume, but not contiguously (c, f). Two differing topologies can occur due to different choice of diagonals.

It is useful to define a few terms at this point. An *interior edge* is an edge of a tetrahedron that is made up of a corner of the parent cube and the center of the cube. A *sign change edge* is an edge of a tetrahedron or a grid cube whose end points have inside and outside labels. A *valid*

tetrahedron is one in which at least one edge is a sign change edge. The proposed strategy in this dissertation makes use of only valid tetrahedra for polygon generation.

During polygon generation, the centroid of each tetrahedron is used as a minimizer. The method of Zhang and Qian in [51] presents two general rules that generate tetrahedra/polyhedrons using the minimizers of unambiguous grid cubes as well as the minimizers of the tetrahedral cells from ambiguous grid cubes. The effect of incorporating these rules into the standard DC algorithm results in a mesh comprising of both surface and tetrahedral meshes.

In this strategy are presented novel rules that result in surface meshes, rather than tetrahedral meshes. The method is effective in generating 2-manifold meshes and can be easily incorporated into the DC algorithm. The details of the rules are presented in the following section.

3.4 Polygon Generation

During the polygon generation phase, the method follows the standard DC approach by analyzing minimal edges as well as other sign change edges. Each minimal edge is an edge that is characterized by a sign change and that is shared between four grid cubes. If all four grid cubes sharing that edge are unambiguous, then we create two triangles using the four minimizers of the four grid cubes. On the other hand, if any one of the four incident grid cubes is ambiguous, it is necessary to apply the following rules: (1) Minimal Edge Rule, (2) Face Diagonal Rule, and (3) Interior Edge Rule.

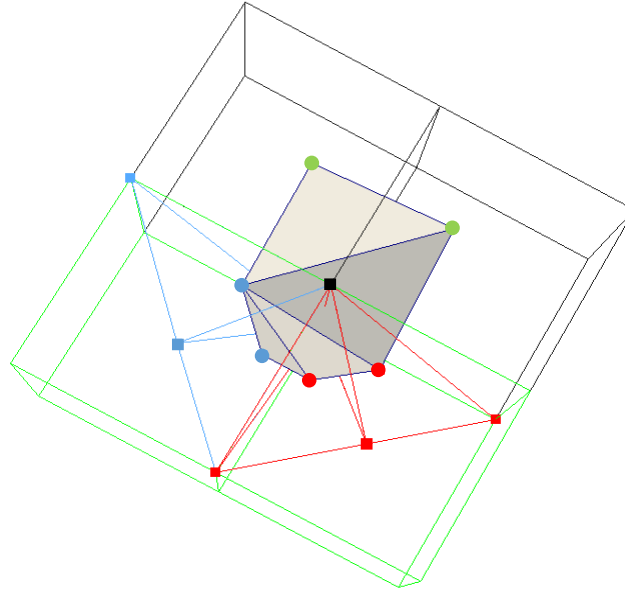


Fig. 16. An illustration of the Minimal Edge Rule. The two grey cubes are unambiguous cubes and the two green cubes are ambiguous cubes. The black square represents one end of the minimal edge. The blue and red squares represent the vertices of the tetrahedral cells. The two blue and two red circles represent the minimizers of the tetrahedra incident on the minimal edge.

Minimal Edge Rule: Create an n -sided polygon, or n -gon, using the minimizers of all the unambiguous cubes and tetrahedral cells that contain the minimal edge.

This rule follows the same concept as in standard DC: if the minimal edge is a sign-change edge, then there must be a surface intersecting the minimal edge. The n -sided polygon is generated by linking together the minimizers of unambiguous grid cubes and tetrahedral cells that share the minimal edge, and then triangulating the n -gon. Each ambiguous grid cube will have exactly two valid tetrahedral cells sharing the minimal edge. It should be noted that the resulting n -gon does not necessarily have to be convex. It is also worth mentioning that extra

care should be taken when using third-party polygon generation algorithms, such as Delaunay tessellation, which have a tendency to generate convex polygons. Fig. 16 illustrates the Minimal Edge Rule. In this figure, the small black square represents one end of the minimal edge that is shared by the four grid cubes. There are two ambiguous grid cubes (colored green) and two unambiguous grid cubes (colored grey). The blue and red squares represent the vertices of the tetrahedral cells. The blue and red lines represent the four tetrahedral cells of the two ambiguous grid cubes. All four tetrahedral cells share the minimal edge. In Fig. 16, a 6-sided polygon is created by first linking together the minimizers of the two unambiguous grid cubes, as well as the minimizers of the four tetrahedral cells sharing the minimal edge, and then triangulating the polygon.

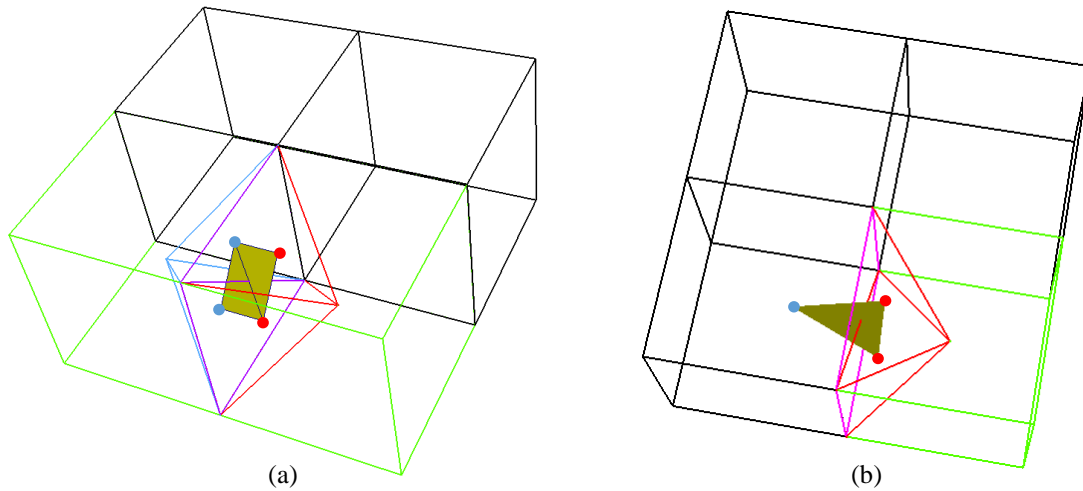


Fig. 17. An illustration of the Face Diagonal Rule. The purple square is the shared face between two cubes. Ambiguous cubes are colored green and unambiguous cubes are colored black. (a) The case of two ambiguous cubes sharing a face. (b) The case of an ambiguous cube and an unambiguous cube sharing a face.

Face Diagonal Rule: For any ambiguous cube sharing a face with another cube, if the face diagonal of the shared face is a sign change edge, then create a polygon using all the minimizers surrounding the face diagonal.

Ambiguous grid cubes can share a face with another ambiguous or unambiguous grid cube. In the case of two ambiguous cubes sharing a face, if the face diagonal is a sign change edge, then there are four valid tetrahedral cells that share the face diagonal. The four minimizers are used to generate a quadrangle, which equates with two triangles. In the case of an ambiguous cube sharing a face with an unambiguous cube, there are two valid tetrahedral cells whose bases comprise the shared face. The minimizers of these two tetrahedral cells as well as the minimizer of the ambiguous cube are used to generate a triangle. Fig. 17 illustrates the application of this rule. In Fig. 17(a), the two green cubes are the ambiguous cubes while the red and blue lines represent the four tetrahedral cells sharing the face diagonal. The face diagonal is a sign change edge. The black cubes are unambiguous. The purple square represents the shared face between the two ambiguous cubes. The red and blue round points represent the four minimizers used to generate the two yellow triangles. In Fig. 17(b), the green cube is the sole ambiguous cube and the red lines represent the two tetrahedral cells making up the shared face with a neighboring unambiguous cube (purple square). The two red round points are the minimizers of the two tetrahedral cells and the blue round point is the minimizer of the unambiguous cube in question. The three minimizers are used to generate a triangle (yellow).

Interior Edge Rule: For a sign-change interior edge of a tetrahedral cell which has one end point that is also shared with the minimal edge, create a polygon using the minimizers of all the tetrahedral cells of the parent cube that share the sign-change interior edge.

An interior edge can be shared by multiple tetrahedral cells. If the interior edge is a sign-change edge, then it follows that there is a surface intersecting the interior edge, and this surface can be constructed using the minimizers of the surrounding tetrahedral cells. Fig. 18 depicts the Interior Edge Rule. In this figure, there is one ambiguous grid cube (green) and three unambiguous grid cubes (grey). The center of the ambiguous cube (yellow square) is a shared vertex for all the tetrahedral cells. The small red squares are the vertices of the tetrahedral cells. The white and black round points make up the minimal edge. The interior edge with a sign-change edge in this figure is made of the white round point and the center of the ambiguous cube. The red lines represent four tetrahedral cells that share the sign change interior edge. The blue round points are the minimizers of the tetrahedral cells. A polygon is created using these four minimizers.

3.5 Detection of Non-manifold Edges and Vertices and Boundary Edges

The proposed method is a modified Dual Contouring algorithm that can produce 2-manifold and watertight surface meshes. In the course of this work, we did not rely only on visual inspection to detect the presence or absence of non-manifold edges and vertices and boundary edges. This method used MeshLab [53] to detect the presence or absence of non-manifold edges and vertices and boundary edges. MeshLab is an open source mesh processing tool that makes extensive use of the VCG (Visualization and Computer Graphics) Library.

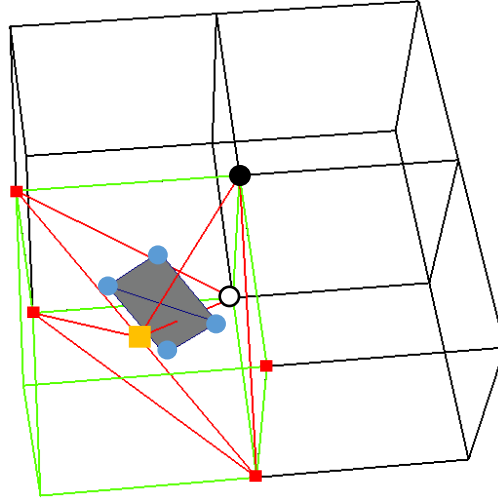


Fig. 18. An illustration of the Interior Edge Rule. The white and black round points represent the minimal edge. The red squares depict the vertices of the four tetrahedral cells incident on the minimal edge: the vertices of the minimal edge are also vertices of two tetrahedral cells. The green round point and the orange square make up the sign-change interior edge. The red lines represent the four tetrahedral cells that share the sign-change edge. The blue round points are the minimizers of the tetrahedral cells that are used to create a polygon.

3.6 Integration into Dual Contouring

This section provides information on the integration of the proposed method into the standard Dual Contouring algorithm. The first part, identifying ambiguous and unambiguous grid cubes can be done during the octree generation phase. For unambiguous cubes, only one minimizer is computed. Ambiguous cubes are first decomposed into a maximum of twelve tetrahedral cells. The centroid of each tetrahedron is then used as the minimizer. As previously mentioned, not all twelve tetrahedral cells of an ambiguous cube are used. Only valid tetrahedral cells, i.e. tetrahedral cells having a sign change edge are used.

Once the octree is generated in standard DC, three recursive functions, **cellProc()**, **faceProc()** and **edgeProc()** are used to locate the minimal edge and the cubes containing the minimal edge. In **edgeProc()**, if the four octree cells are all leaf nodes, then the four minimizers of the octree cells are used to generate a quad or two triangles.

For 2-manifold polygon generation, we insert the proposed polygon generation rules into **edgeProc()** with the following criteria: If all four octree cells are leaf cells and if all four octree cells are unambiguous, then use the four minimizers of the four cells to create two triangles. If any one of the four octree cells are ambiguous cells, then use the *MinimalEdgeRule()*, *FaceDiagonalRule()* and *InteriorEdgeRule()* to generate polygons. The three rules can be called in any order. The modified **edgeProc()** algorithm, in which the 4 octree leaves at the same level c_1, c_2, c_3 and c_4 is passed as input parameters, is given below:

```

edgeProc( $c_1, c_2, c_3, c_4$ ) {
    IF ( $c_i$  are all leaf nodes) {
        IF ( $c_i$  are all unambiguous) {
            Generate a quad or two triangles using the minimizers of the four leaf
            nodes
        }
        ELSE IF (Any  $c_i$  are ambiguous) {
            MinimalEdgeRule()
            FaceDiagonalRule()
            InteriorEdgeRule()
        }
    }
    ELSE {
        Two calls to edgeProc()
    }
}

```

3.7 Results and Discussion

A modified version of the Dual Contouring algorithm has been presented that is capable of overcoming some of the limitations of standard DC. This proposed method uses tetrahedral decomposition of ambiguous cubes to generate 2-manifold and watertight surface meshes.

Fig. 19(a) shows a non-manifold vertex that was created due to complimentary Case 4 cube configuration. Fig. 19(b) shows the 2-manifold tube-like mesh replacing the non-manifold vertex using the proposed method. Fig. 19(c) shows another example of a non-manifold vertex caused by Case 4 ambiguity, and Fig. 19(d) shows that it has been replaced by mesh that is separate and not tubular.

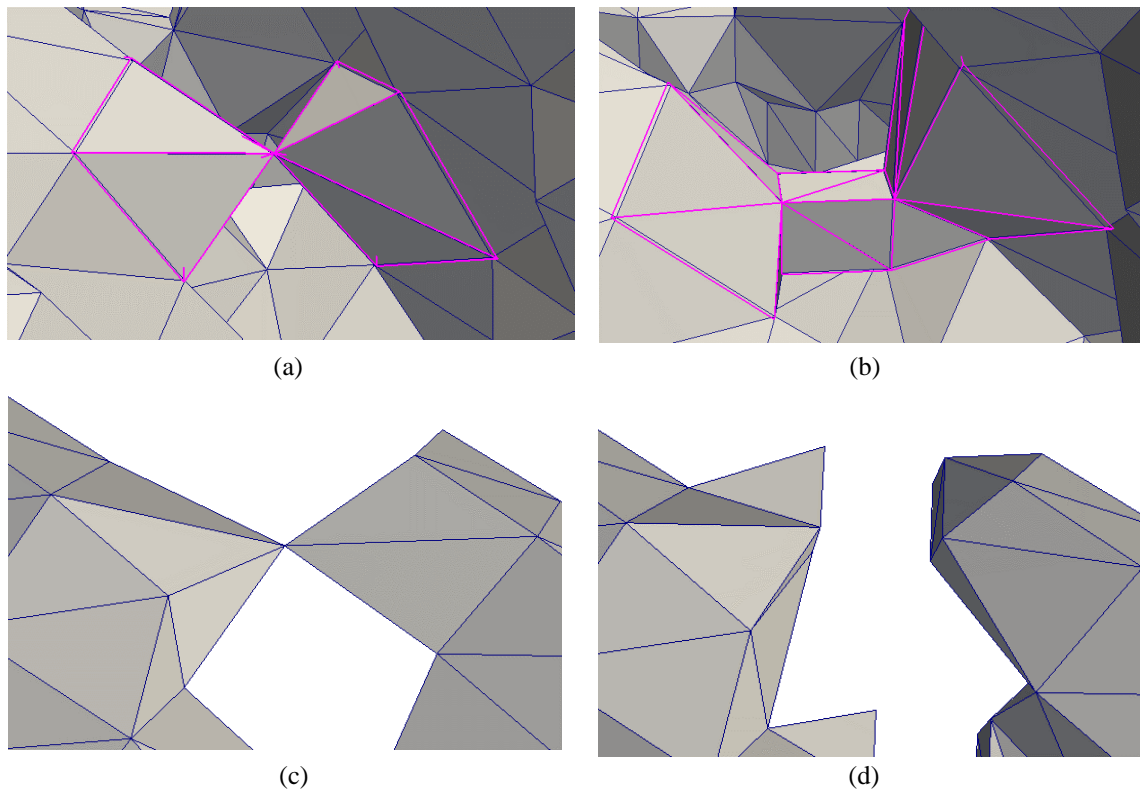


Fig. 19. Two examples of a non-manifold vertex being replaced by a 2-manifold mesh. (a, c) shows two different non-manifold vertices, (b, d) shows their corresponding 2-manifold solutions.

Fig. 20 shows a few examples of applying the proposed method on synthetic datasets. The top row of Fig. 20 shows non-manifold meshes generated using traditional DC (the red dot depicts a non-manifold vertex and the red lines depicts non-manifold edges), while the bottom row shows 2-manifold meshes generated using the proposed method. Fig. 20(a) shows a situation having Case 4 ambiguity. There are two possible 2-manifold solutions for the Case 4 situation, and the proposed method is able to generate both solutions, as shown in the bottom row of Fig. 20(a). Similar to the concept described in the previous section, the inside/outside label for the center of the cube is the determining factor for the 2-manifold mesh. If the two diagonally opposite corners of the cube as well as the center of the cube are inside the volume, then a tunnel like structure is the best representation of the volume, and if the two diagonally opposite corners are inside the volume but the center is outside, then the resulting mesh would consist of two disjointed parts. Similarly, Fig. 20(b) shows a situation having Case 10 ambiguity. The top part of Fig. 20(b) shows the non-manifold mesh generated using standard DC and the bottom part shows two of three possible 2-manifold solutions. The top figure of Fig. 20(c) shows a situation having Case 7 ambiguity, and the bottom part shows one of seven possible 2-manifold solutions.

Although Fig. 20 shows only three ambiguous cases, the proposed method is capable of generating watertight and 2-manifold meshes for all ambiguous cases. The important point to consider here is that *the polygon generation rules can be applied to any ambiguous case*, and it is not necessary to explicitly identify the ambiguous cases.

The proposed algorithm has been applied to a digital atlas [54] of deep brain structures, specifically the basal ganglia and thalamus created using serial histological data. This atlas is in MINC 2.0 (Medical Imaging NetCDF) format and contains a total of 92 labeled structures. The atlas consists of $334 \times 334 \times 334$ voxels with a stepsize of 0.3 mm. Both the standard DC and the

proposed method has been used to generate surface meshes of deep brain structures in two experiments.

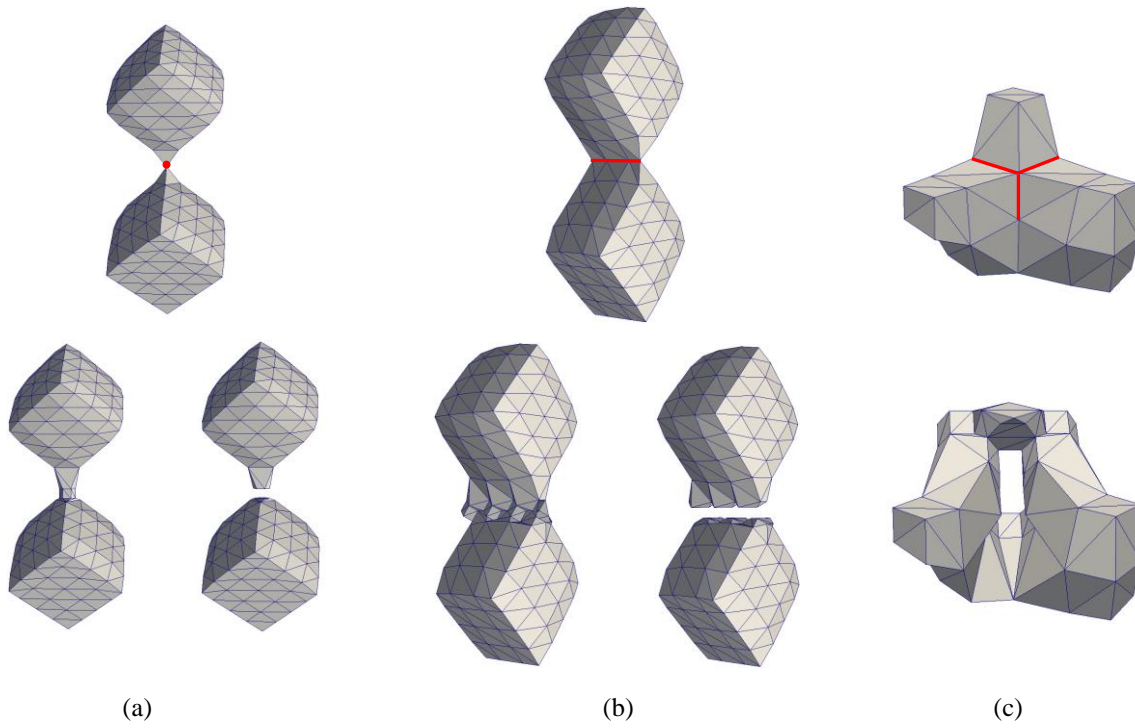


Fig. 20. Examples of applying the proposed DC method on synthetic data. The meshes on the top row are non-manifold meshes generated using traditional DC, and the meshes on the bottom are 2-manifold meshes generated using the proposed method. (a) Case 4 ambiguity, (b) Case 10 ambiguity, (c) Case 7 ambiguity.

3.7.1 Experiment without Preprocessed Data

In the first experiment, the atlas data were not preprocessed in any way. Fig. 21 shows the surface meshes generated using traditional DC (yellow colored meshes). In this figure, the red lines depict non-manifold edges that standard DC is unable to avoid. TABLE 1 reports the number of triangles and vertices, as well as the number of non-manifold edges and vertices for the meshes in Fig. 21. Fig. 22 shows the surface meshes generated using the proposed 2-manifold DC method (grey colored meshes). In this figure, all the meshes are 2-manifold and watertight. TABLE 2 reports the number of triangles and vertices, along with the number of non-manifold edges and vertices for the meshes in Fig. 22. In both TABLE 1 and TABLE 2, the names of the anatomical structures and their corresponding atlas label numbers are given.

As evidenced in TABLE 1, standard DC produces meshes containing non-manifold elements, in the case of large meshes, more than 600 non-manifold edges, while the proposed method produces surface meshes completely free of non-manifold elements. The only concern here is that the meshes generated using the proposed method have a significantly higher number of vertices and triangles. This is because the proposed method inserts multiple minimizers (as many as twelve in some cases) in ambiguous cubes through tetrahedral decomposition, thus increasing the number of vertices, and the number of triangles.

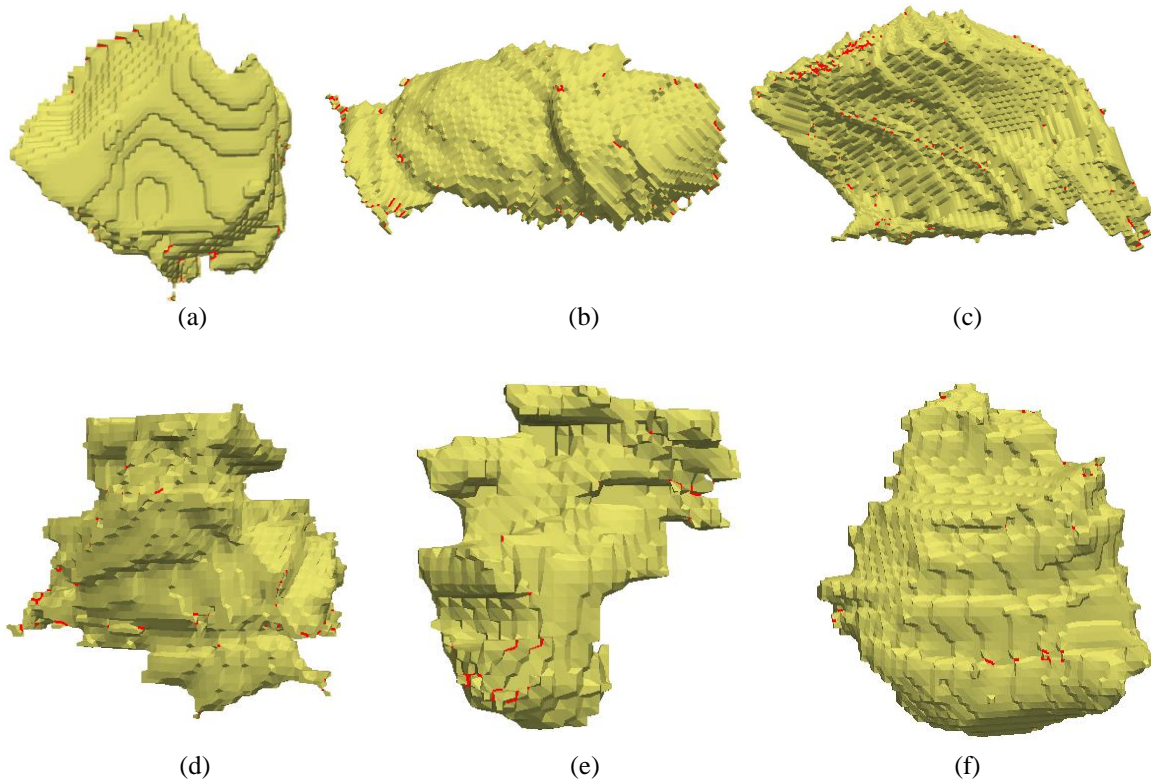


Fig. 21. Surfaces generated from a digital deep brain atlas using standard DC. The atlas was not preprocessed in any way. The red lines depict non-manifold edges. (a) Globus Pallidus (b) Globus Pallidus Internal (c) Globus Pallidus External (d) Nucleus lateropolaris thalami (e) Nucleus fasciculosus thalami (f) Subthalamic Nucleus.

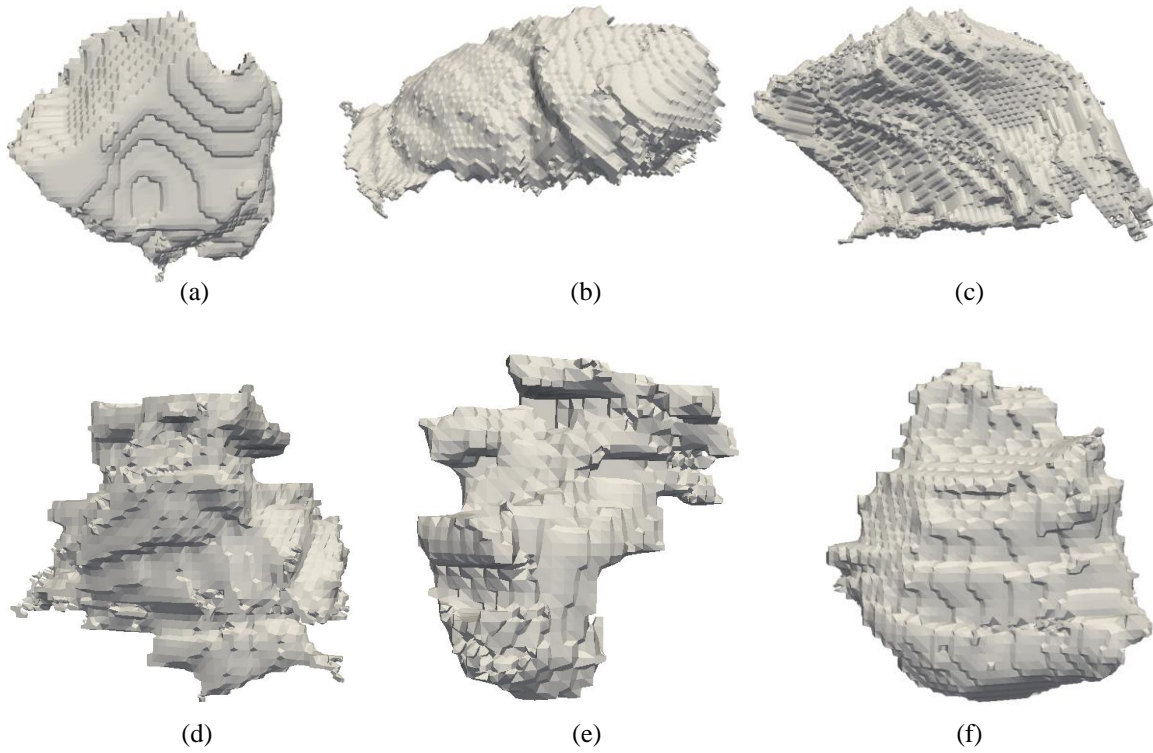


Fig. 22. Surfaces generated from a digital deep brain atlas using the proposed method. The atlas was not preprocessed in any way. All the meshes are watertight and 2-manifold. (a) Globus Pallidus (b) Globus Pallidus Internal (c) Globus Pallidus External (d) Nucleus lateropolaris thalami (e) Nucleus fasciculosus thalami (f) Subthalamic Nucleus.

TABLE 1
RESULTS OF SURFACES GENERATED USING STANDARD DUAL CONTOURING ON
UNPREPROCESSED DATA

	Atlas Label	Structure Name	Number of Vertices/Tris	Number of Non- manifold Edges/Vertices
Fig. 21 (a)	05	Globus Pallidus	26068/53460	656/70
Fig. 21 (b)	11	Globus Pallidus Internal	23396/48088	642/67
Fig. 21 (c)	12	Globus Pallidus External	26408/53644	453/39
Fig. 21 (d)	26	Nucleus lateropolaris thalami	13401/27424	333/31
Fig. 21 (e)	27	Nucleus fasciculosus thalami	4971/10096	82/9
Fig. 21 (f)	39	Subthalamic Nucleus	9939/20152	148/8

TABLE 2
RESULTS OF SURFACES GENERATED USING THE PROPOSED DUAL CONTOURING
METHOD ON UNPREPROCESSED DATA

	Atlas Label	Structure Name	Number of Vertices/Tris	Number of Non- manifold Edges/Vertices
Fig. 22 (a)	05	Globus Pallidus	35750/72670	0/0
Fig. 22 (b)	11	Globus Pallidus Internal	32369/66018	0/0
Fig. 22 (c)	12	Globus Pallidus External	32707/66226	0/0
Fig. 22 (d)	26	Nucleus lateropolaris thalami	18123/36806	0/0
Fig. 22 (e)	27	Nucleus fasciculosus thalami	6231/12490	0/0
Fig. 22 (f)	39	Subthalamic Nucleus	12006/24276	0/0

3.7.2 Experiment with Preprocessed Data

A second experiment was performed on the same deep brain structures using both traditional DC and the proposed method. For this experiment, some preprocessing was performed on the atlas data. Each deep brain structure was separated and binarized into a single volume. A crude smoothing was performed using the binary morphological operations opening and closing. The structuring element was a cube of size $3 \times 3 \times 3$. The sequence of morphological operations were closing, followed by opening. This preprocessing simplified the topology of the volume considerably. Fig. 23 shows the meshes generated using standard DC (yellow colored meshes), and Fig. 24 shows the meshes generated using the proposed method (grey colored meshes).

TABLE 3 reports the number of triangles and vertices, as well as the number of non-manifold edges and vertices for the meshes in Fig. 23. TABLE 4 reports the number of triangles and vertices, along with the number of non-manifold edges and vertices for the meshes in Fig. 24. In both TABLE 3 and TABLE 4, the names of the anatomical structures and their corresponding atlas label numbers are given.

The purpose of this second experiment is to demonstrate the fact that even after preprocessing, standard DC is unable to produce 2-manifold meshes in almost all cases, with Fig. 23(e) being the exception. Preprocessing using binary morphological operations considerably simplified the topology of the surface. That is why there are fewer non-manifold edges and vertices in the meshes generated using standard DC. The simplified volume also contains fewer ambiguous cubes. So the number of vertices and triangles in the meshes generated by the proposed method are not significantly higher than the number of vertices and triangles of the meshes generated by standard DC. In the case of Fig. 23(e), the preprocessing operation

simplified the volume to such an extent that there were no ambiguous cubes present, and therefore, the proposed method behaved exactly like standard DC. For the other cases, the proposed method generated meshes that are completely 2-manifold and watertight.

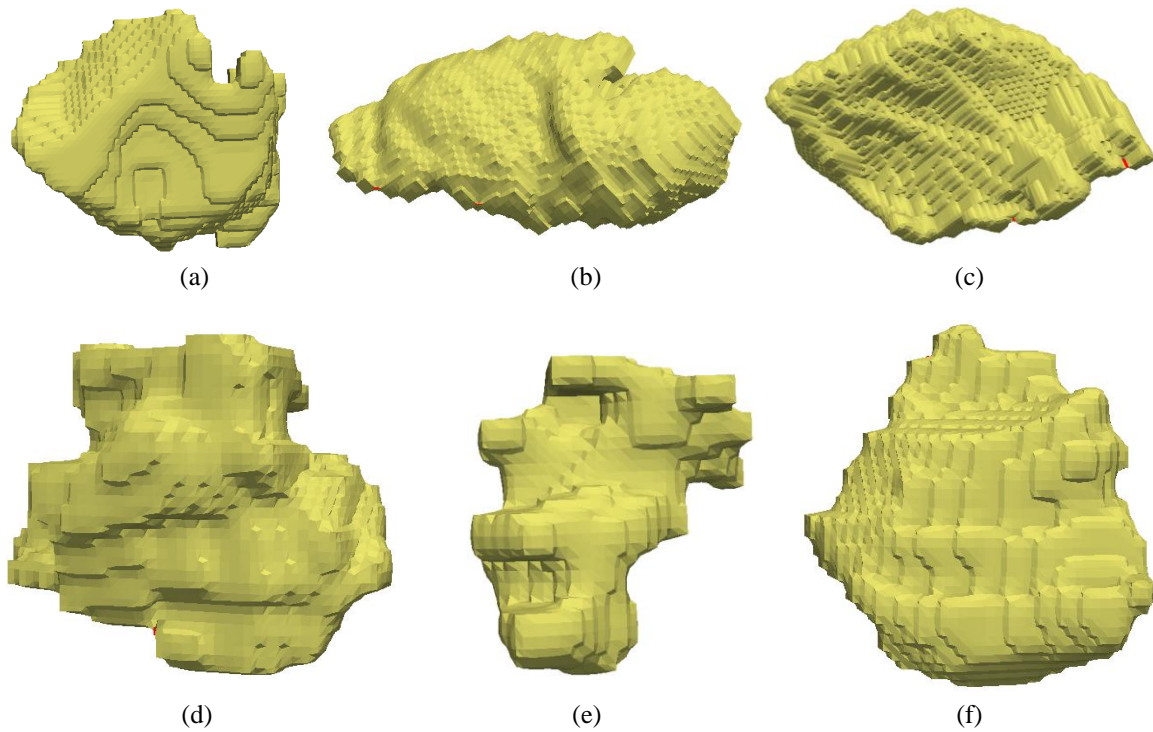


Fig. 23. Surfaces generated using standard DC on preprocessed atlas data. The red lines depict non-manifold edges. (a) Globus Pallidus (b) Globus Pallidus Internal (c) Globus Pallidus External (d) Nucleus lateropolaris thalami (e) Nucleus fasciculosus thalami (f) Subthalamic Nucleus.

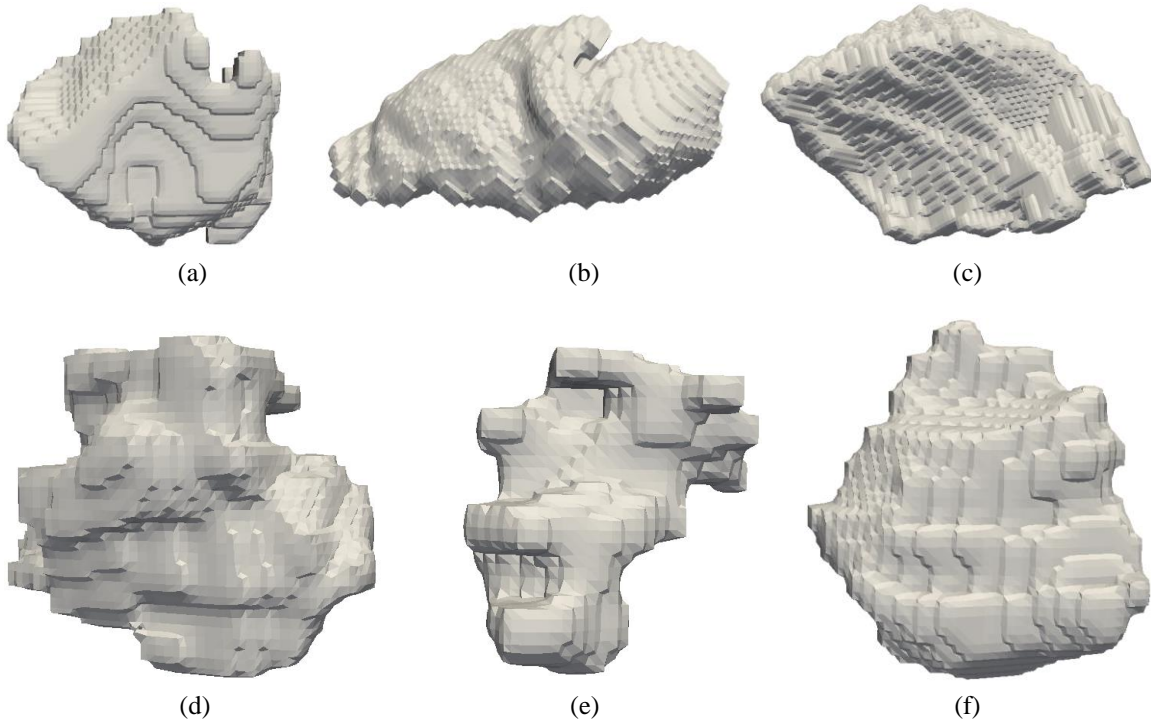


Fig. 24. Surfaces generated using the proposed method on preprocessed atlas data. These meshes are completely 2-manifold. (a) Globus Pallidus (b) Globus Pallidus Internal (c) Globus Pallidus External (d) Nucleus lateropolaris thalami (e) Nucleus fasciculosus thalami (f) Subthalamic Nucleus.

TABLE 3
RESULTS OF SURFACES GENERATED USING STANDARD DUAL CONTOURING ON
PREPROCESSED DATA

	Atlas Label	Structure Name	Number of Vertices/Tris	Number of Non- manifold Edges/Vertices
Fig. 23 (a)	05	Globus Pallidus	16858/33712	2/0
Fig. 23 (b)	11	Globus Pallidus Internal	15736/31480	7/0
Fig. 23 (c)	12	Globus Pallidus External	20411/40828	7/0
Fig. 23 (d)	26	Nucleus lateropolaris thalami	9608/19212	3/0
Fig. 23 (e)	27	Nucleus fasciculosus thalami	3550/7092	0/0
Fig. 23 (f)	39	Subthalamic Nucleus	8165/16324	1/0

TABLE 4
RESULTS OF SURFACES GENERATED USING THE PROPOSED DUAL CONTOURING
METHOD ON PREPROCESSED DATA

	Atlas Label	Structure Name	Number of Vertices/Tris	Number of Non- manifold Edges/Vertices
Fig. 24 (a)	05	Globus Pallidus	16890/33776	0/0
Fig. 24 (b)	11	Globus Pallidus Internal	15847/31698	0/0
Fig. 24 (c)	12	Globus Pallidus External	20506/41016	0/0
Fig. 24 (d)	26	Nucleus lateropolaris thalami	9657/19314	0/0
Fig. 24 (e)	27	Nucleus fasciculosus thalami	3550/7092	0/0
Fig. 24 (f)	39	Subthalamic Nucleus	8181/16358	0/0

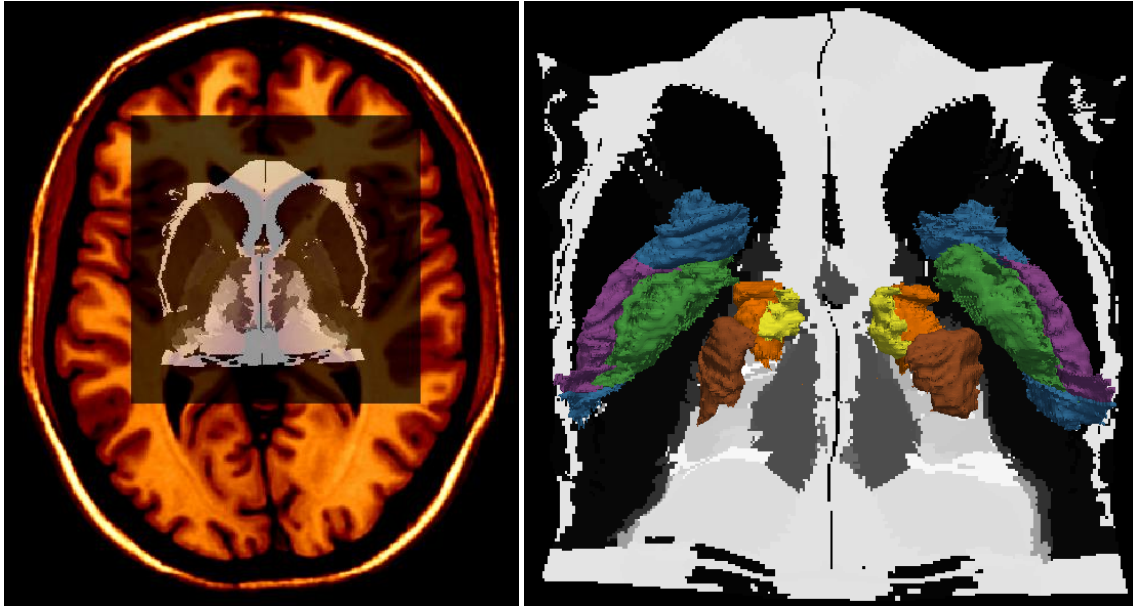


Fig. 25. An illustration of the digital deep brain atlas. (left) The digital atlas superimposed over the brain. (right) The anatomical structures of Fig. 22 superimposed over a slice along the XY plane ($Z = 163$) of the digital atlas.

Fig. 25 (left) shows the digital atlas superimposed over a corresponding slice of the T1-weighted Colin 27 average brain (in hot-metal coloring). Fig. 25 (right) shows all the anatomical structures of Fig. 22 superimposed over a magnified XY slice of the digital deep brain atlas. The atlas is depicted in gray scale, and the mesh coloring is as follows: Globus Pallidus (blue), Globus Pallidus Internal (green), Globus Pallidus External (purple), Nucleus lateropolaris thalami (orange), Nucleus fasciculosus thalami (yellow) Subthalamic Nucleus (brown).

3.7.3 Quality of Triangles

A commonly used metric to describe the quality of triangles in surface meshes is the radius ratio described in [55], as shown in Equation (35).

$$ratio = 2 \frac{r_{in}}{r_{circ}}. \quad (35)$$

In Equation (35) r_{in} describes the radius of the circle inscribed in the triangle, and r_{circ} is the radius of the circumscribing circle. A value close to 1 indicates a very good quality triangle (close to an equilateral triangle) and a value near zero indicates a poor quality triangle (a triangle which is collapsing to an edge). The average values for the structures in Fig. 22 and Fig. 24 are reported in TABLE 5. Also included in TABLE 5 are the number of triangles whose ratio values fall below a threshold (0.2 in this case). As can be seen, there are a number of triangles whose radius ratios are less than ideal. Most of these poor quality triangles are created by the Minimal Edge Rule. As mentioned before, the n -gon created in this rule does not necessarily have to be convex. In our implementation, we use an ad-hoc method to triangulate the n -gon but this triangulation method is not configured towards producing high quality triangles as Delaunay-based methods do. Delaunay-based tessellation methods are avoided because they tend produce convex triangulations, and enforcing the n -gon to be always convex can introduce non-manifold edges in the resulting surface mesh.

TABLE 5
MESH QUALITY FOR MESHES GENERATED BY THE PROPOSED METHOD USING
UNPREPROCESSED DATA AND PREPROCESSED DATA

		Meshes from unpreprocessed data		Meshes from preprocessed data	
Atlas label	Structure name	Average Radius ratio	No of tris (No of tris with radius ratio ≤ 0.2)	Average Radius ratio	No of tris (No of tris with radius ratio ≤ 0.2)
05	Globus Pallidus	0.744	72670 (503)	0.782	33776 (102)
11	Globus Pallidus Internal	0.741	66018 (482)	0.781	31698 (82)
12	Globus Pallidus External	0.747	66226 (487)	0.775	41016 (144)
26	Nucleus lateropolaris thalami	0.745	36806 (271)	0.781	19314 (78)
27	Nucleus fasciculosus thalami	0.756	12490 (81)	0.784	7092 (29)
39	Subthalamic Nucleus	0.747	24276 (155)	0.778	16358 (60)

3.8 Comparison with Existing Methods

In this section, the results of the proposed method are compared with the results of the opensource implementation of the Intersection-free Contouring algorithm (IC) [47] available from GitHub (<https://github.com/aewallin/dualcontouring>). For input data, the Asian Dragon polygonal model was used and is freely available at The Stanford 3D Scanning Repository (<http://graphics.stanford.edu/data/3Dscanrep>).

For the proposed method, the opensource visualization library VTK's (Visualization ToolKit) *vtkPolyDataToImageStencil* filter was used to create a volume with appropriate spacing from the Dragon model. The volume was then used as input in the proposed method. The octree was set to depth 9 also. One single mesh was generated.

For the IC method, the same basic procedure was followed as described in [47]: The Dragon model was converted into Hermite data using the PolyMender software [56], and then applied the Hermite data to the IC software. The only difference is was setting the octree depth of 9. IC uses adaptive simplification, based on an error threshold to simplify the resulting polygonal mesh. Larger threshold values result in smaller number of vertices and triangles. Three error threshold values were used for adaptive simplification and generated three polygonal meshes. Since [47] does not report any error threshold values, this method used values so that number of vertices and triangles of the three meshes would be greater than, close to, and less than the number of vertices and triangles of the mesh generated using the proposed method. The results are summarized in TABLE 6.

TABLE 6
MESH STATISTICS AND QUALITY

Mesh Generation Method	Number of vertices	Number of triangles	Number of non-manifold edges/vertices	Average Radius ratio	No of tris with radius ratio ≤ 0.2
IC, threshold = 0.7	194778	390550	544/186	0.430	108924
IC, threshold = 0.5	226408	453986	536/225	0.452	132929
IC, threshold = 0.3	212770	626940	525/259	0.465	197812
Proposed method	237117	474454	0/0	0.763	4591

Fig. 26 shows the four meshes generated using the IC method and the proposed method. As reported in TABLE 6, the mesh generated using the proposed method is completely 2-manifold whereas the meshes generated using the IC method contain large numbers of non-manifold edges and vertices. Furthermore, the quality of the triangles of the mesh generated using the proposed method is superior to the meshes generated by the IC method. However, the downside of having such high-quality triangles is that the mesh generated by the proposed method has a blocky appearance (explained in the next section), as shown in the right inset in Fig. 26. In contrast, the meshes generated using the IC method have much smoother appearance, at the cost of having lower quality triangles.

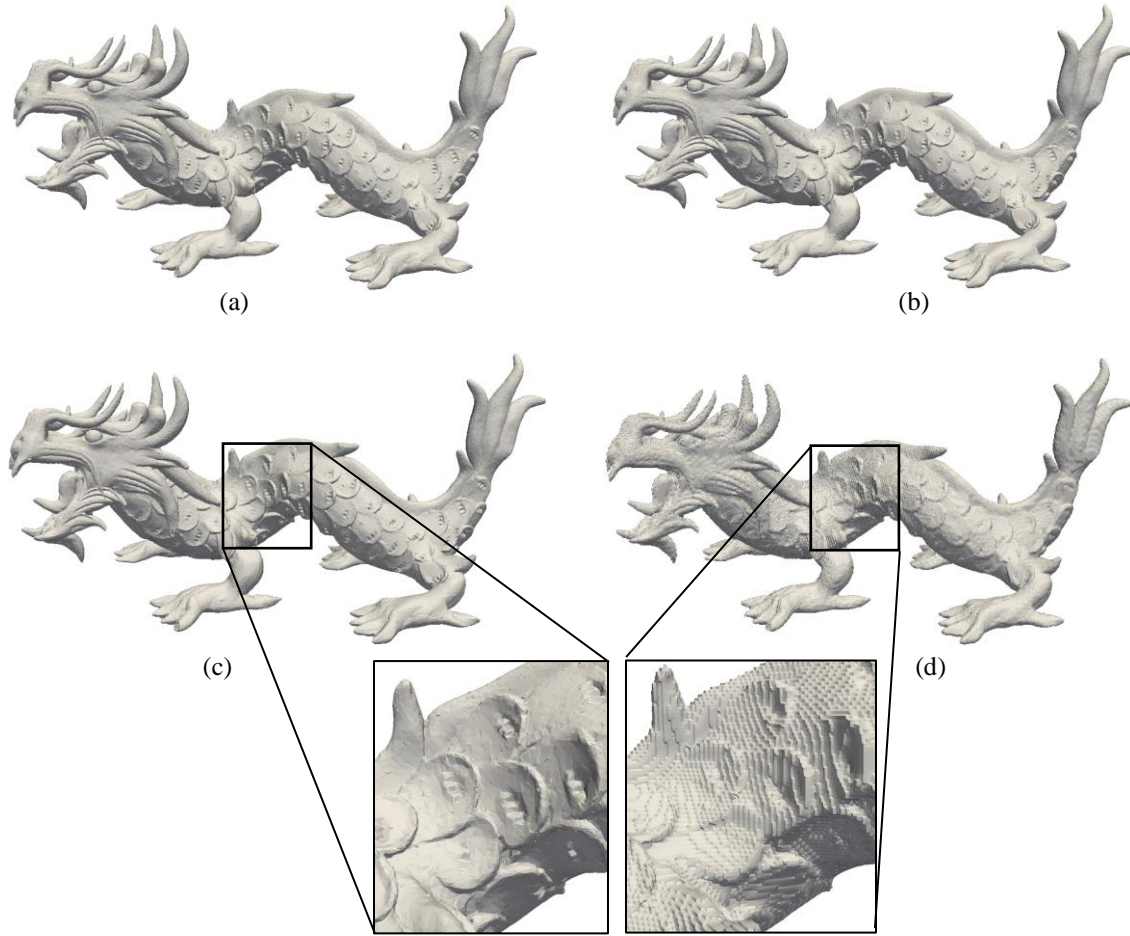


Fig. 26. A comparison of meshes of the Asian Dragon model. (a) IC method, threshold = 0.3, (b) IC method, threshold = 0.5, (c) IC method, threshold = 0.7, (d) proposed method. The two insets near the bottom show a close-up of the mesh generated using the proposed method and one generated using the IC method.

3.9 Mesh Smoothing

The meshes generated using the proposed method, as shown in Fig. 22, Fig. 24, and Fig. 26(d), all show a staircase-like appearance, even though they are 2-manifold and watertight. This section will elaborate on the reason for the staircase-like appearance, and offer a solution.

One of the limitations of the proposed solution is that it assumes that minimizers are always inside their respective grid cubes. This is not an issue in the case of tetrahedral cells (within ambiguous cubes) because we use their centroids as minimizers, and the centroid will always lie within the tetrahedron and its parent cube. In the case of unambiguous cubes, the minimizers are computed using Quadratic Error Functions, and the method used to solve these QEFs is an important factor to consider. In certain situations, the computation of minimizers may result in the minimizer being placed outside its respective cube. This allows the resulting mesh to have a smoother appearance, as shown in Fig. 27 (right), but at the cost of containing intersecting triangles (inset in Fig. 27) and/or cracks. Facilities in MeshLab were used to detect the presence of intersecting triangles in the meshes. Schmitz et al. uses an iterative particle-based method to compute minimizers in [57] which results in good approximation of the isosurface, but does not guarantee a solution.

In order to obtain watertight surfaces in our solution, we constrain the minimizers of unambiguous cubes to remain within their respective cubes. While this gives the resulting meshes their staircase-like appearance, the meshes are watertight and do not contain any intersecting triangles and/or cracks as shown in Fig. 27 (left).

One solution to rectify the staircase-like appearance of the meshes generated by the proposed method is to use some form of mesh smoothing technique as a post-processing step. Laplacian smoothing is a simple and basic smoothing algorithm that changes the position of all vertices by computing a new position based on neighboring vertices and triangles/faces. Fig. 28 shows two example of using the Laplacian smoothing technique as a post-processing stage on a surface mesh of the Nucleus lateropolaris thalami (atlas label 26) generated by the proposed method.

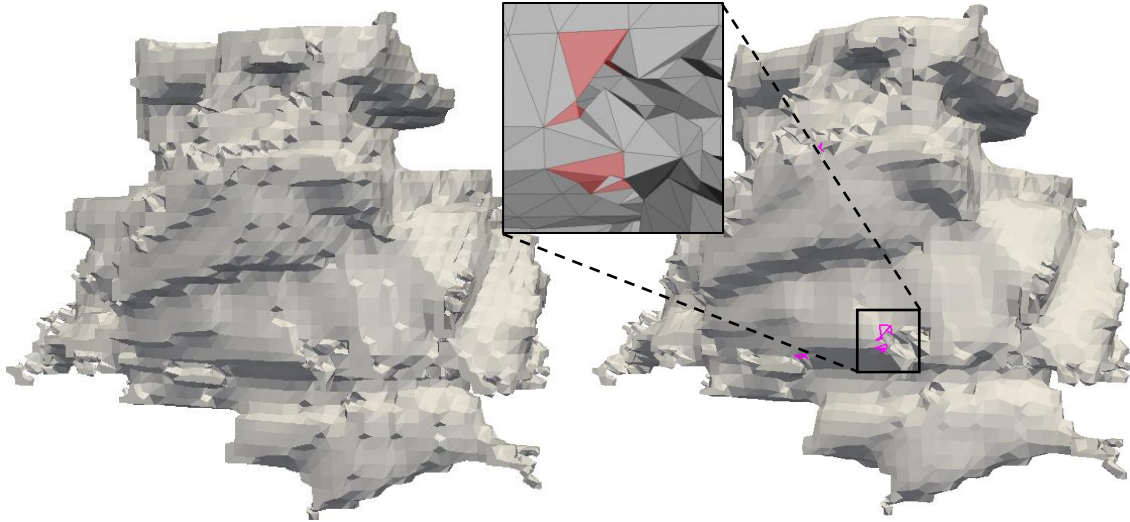


Fig. 27. Two meshes of the Nucleus lateropolaris thalami (atlas label 26) generated using the proposed method. (left) Minimizers were constrained to remain within their respective grid cube, resulting in the blocky appearance of the mesh. (right) Minimizers were allowed to be positioned outside their respective grid cubes, resulting in a smoother appearing mesh, but containing intersecting triangles (inset).

In the case of using Laplacian-based smoothing, the amount of smoothing depends on the number of iterations used in the smoothing process. More iterations result in a much smoother surface. One problem with using a basic Laplacian smoothing is that it can cause shrinkage, as well as requiring a high number of iterations in order to be effective. A better alternative for post-processing smoothing is to use the Taubin filter, which can avoid shrinkage and requires fewer iterations.

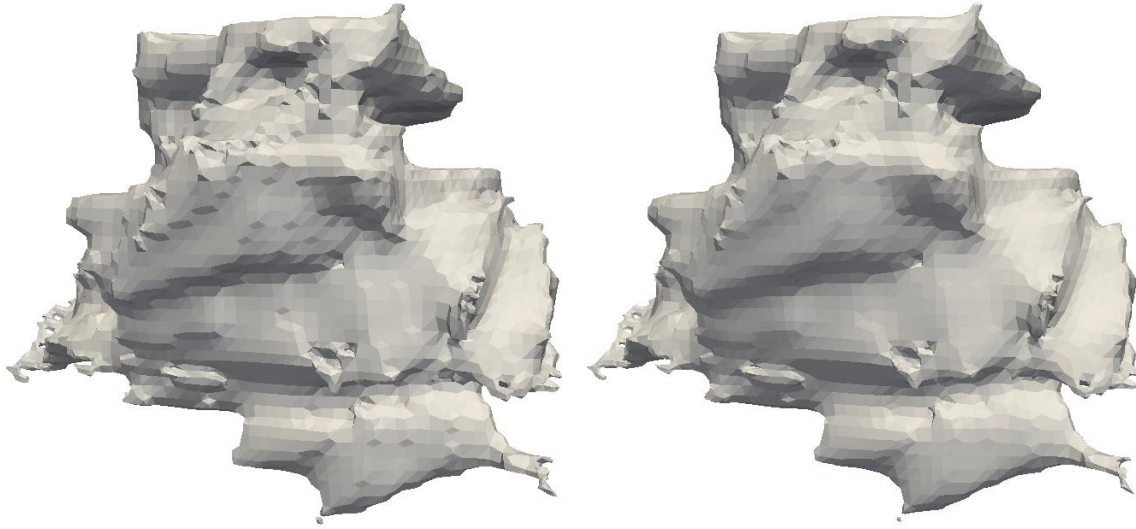


Fig. 28. The results of applying Laplacian smoothing on a surface mesh of the Nucleus lateropolaris thalami (atlas label 26). (left) The number of iterations was set to 100, (right) the number of iterations was set to 200.

In any case, the biggest drawback of using any post-processing smoothing is that since vertex positions are being changed, the quality of triangles may also adversely change: triangles may be stretched into thin slivers, and intersecting triangles may occur near thin structures.

3.10 Tetrahedral Mesh Generation

One type of mesh that is commonly used in engineering and biomedical research is the tetrahedral mesh. Tetrahedral mesh generation can be classified [58] into the following four categories: (1) Octree-based, (2) Delaunay, (3) Advancing Front and (4) Optimization-based. Among these categories, Delaunay based techniques are the most frequently used. In many applications, an initial surface mesh, known as a piecewise linear complex (PLC) that coincides

with the boundary of the problem domain is used as an initial starting point for the tetrahedralization process. In such cases, the user has to ensure (either manually or by using mesh editing software) that the input surface mesh does not contain geometric errors such as holes, slivers, intersecting triangles and non-manifold elements.

Software such as *TetGen* [59] and the opensource library *CGAL* (Computational Geometry Algorithms Library) [60] are able to generate tetrahedral meshes from an input PLC using Delaunay-based methods. Fig. 29 shows two cross-sections of tetrahedral meshes generated using TetGen, using the surface mesh of the Nucleus lateropolaris thalami (Fig. 22(d)) as input. The purple colored triangles represent the tetrahedral elements and the blue colored triangles represent the input surface mesh. Fig. 29 (Bottom left) shows a coarse tetrahedralization (when the edge-radius ratio is set to 1.5) and Fig. 29 (bottom right) shows a finer tetrahedralization (when the edge-radius ratio is set to 1.0). Although TetGen and CGAL both provide facilities for mesh refinement and optimal tetrahedralization, none of those facilities were utilized during the generation of the meshes in Fig. 29. The main purpose of this section is to emphasize the fact that the surface meshes generated using the proposed methodology are *error free* (2-manifold, watertight and intersection free) and can be readily used in the generation of tetrahedral meshes.

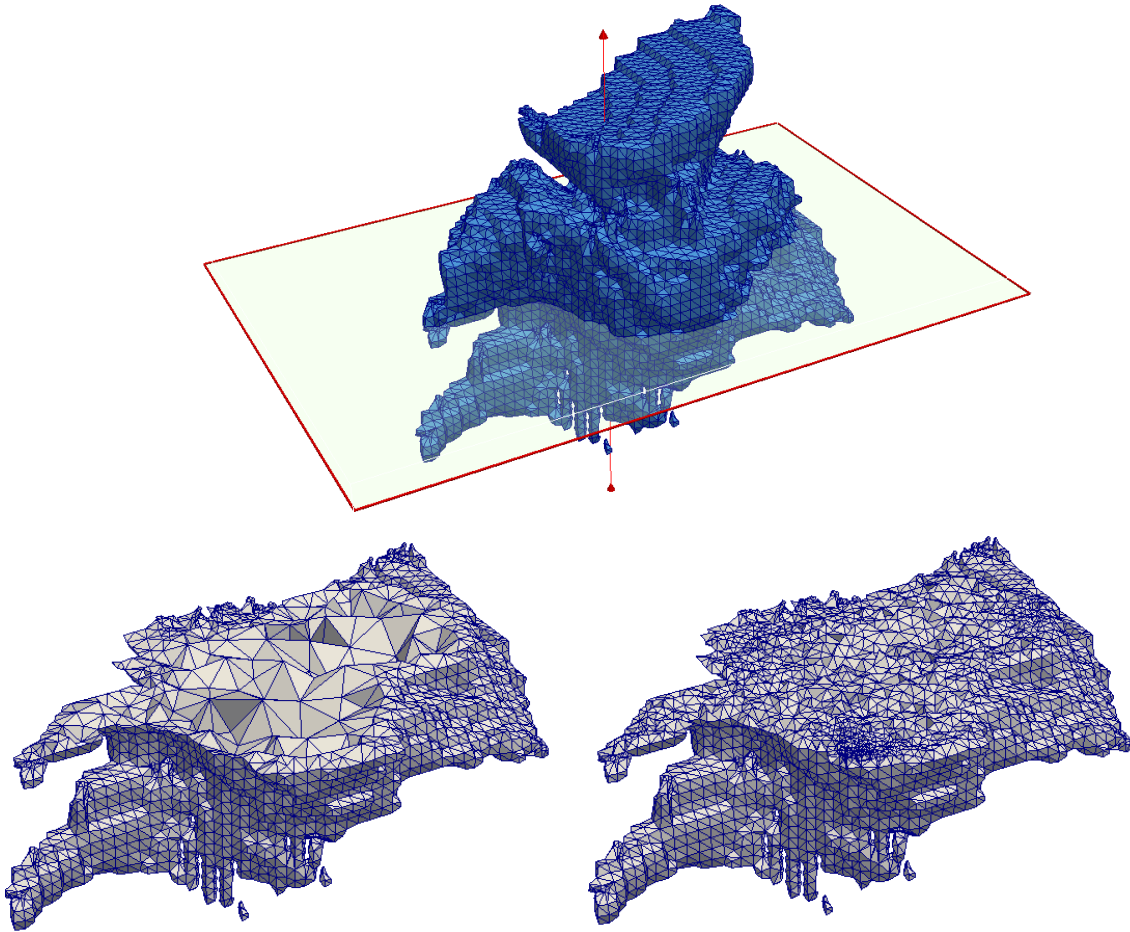


Fig. 29. Tetrahedral meshes created using the surface mesh of the Nucleus lateropolaris thalami (atlas label 26) as the input PLC. (Top) Surface mesh, with cutting plane, (Bottom left) coarse tetrahedralization (Bottom right) Finer tetrahedralization.

3.11 Conclusions

The standard Dual Contouring (DC) algorithm can be made to use adaptive as well as non-adaptive octrees. In the adaptive approach, the grid cubes can be of different sizes (different levels of the octree) whereas in the non-adaptive approach, all the grid cubes are of the same size (lowest level of the octree). One of the limitations of our proposed solution is that it is only

applicable in non-adaptive octrees where a minimal edge will always have four grid cubes sharing the edge.

The proposed method generates watertight and 2-manifold surface meshes that have an overall unsmoothed and staircase-like appearance because of limitations placed on the computed minimizers' positions. This can be circumvented by using post-processing mesh smoothing techniques, but caution must be exercised when using such techniques.

For the results presented thus far, we have not included the runtime for any of the meshes. This is because some parts of the proposed method have not yet been optimally/efficiently implemented. In general, total runtime is between 3 minutes (for the meshes shown in Fig. 22 and Fig. 24) to around 13 minutes (for the Asian Dragon mesh in Fig. 26 (d)), with most of the runtime spent reading the volume. Optimizing all aspects of the implementation of the proposed method would reduce the runtime.

A modified Dual Contouring algorithm is presented that is capable of generating 2-manifold surface meshes. Since non-manifoldness occurs due to the presence of ambiguous grid cubes, we proposed a method to subdivide an ambiguous cube into tetrahedral cells. The centroids of these tetrahedral cells are used as minimizers, and allow an ambiguous cube to have multiple minimizers. Novel polygon generation rules that result in 2-manifold surfaces have also been provided.

Two sets of results were presented using unpreprocessed and preprocessed data. For both sets of data, standard DC and the proposed method to generate surface meshes were applied. MeshLab facilities were used, instead of relying on visual inspection, to confirm the presence or absence of boundary edges and non-manifold edges and vertices. In both cases, the proposed method produced meshes that were 2-manifold and watertight, while traditional DC produced

meshes with non-manifold edges and vertices. Our proposed strategy is simple and effective, and can be easily integrated into the traditional Dual Contouring algorithm. Furthermore, the resulting surface meshes are error free, and can easily be used for the generation of tetrahedral meshes.

CHAPTER 4

MULTI-MATERIAL AND 2-MANIFOLD DUAL CONTOURING

Chapter 3 presented a modified Dual Contouring algorithm that is capable of producing watertight and 2-manifold surface meshes. The surface meshes, as well as the accompanying literature review presented thus far deal solely with single material surface meshes. In this chapter, the 2-manifold DC algorithm is extended to produce multi-material surface meshes.

4.1 Literature Review

So far, the algorithms discussed previously deal mostly with Dual Contouring for single material mesh generation, with the exception of [51]. Multi-material surface meshing is the next evolution in surface mesh generation. For a multi-labeled data, Bloomenthal and Ferguson [61] decompose a cube into six tetrahedral cells. For each tetrahedron that intersects the surface, the edges, faces and interior of each tetrahedron is examined for vertices, and polygons are produced. Hege et al. [62] present a variation of the MC algorithm that can produce meshes for non-binary volumes. This involves subdividing a cube into a number of sub-cells whose probable material indices are established using trilinear interpolation. In [63] Bonnell et al. use volume fraction information on a dual grid constructed from regular hexahedral grids, which are then split into six tetrahedral cells to generate non-manifold multi-material boundary surfaces. Wu and Sullivan [64] extend the classical Marching Cubes algorithm to produce 2D and 3D multi-material meshes where every faceted surface separates two materials. Reitingner et al. [65] present a modified MC-based method that can produce consistent non-manifold meshes from multi-labeled datasets. A cube whose corners have more than one material label is subdivided into eight smaller cubes, and faces of the cube whose corners have more than one material label are

subdivided into four sub-cells. A constrained Laplacian filter is used on the output mesh to reduce staircase-like artifacts. Bertram et al. [66] use a method similar to [62] but relies on a dual grid. In [67] Bischoff and Kobbelt simplify topological ambiguities by subdividing voxels that contain critical edges or vertices before extracting the material interface. Boissonnat and Oudot in [68] present a Delaunay-based surface mesh generator that produces provably good meshes, and Oudot et al. extend this to volumetric meshing in [69]. Pons et al. [70] further extend these Delaunay-based methods to produce surface and volumetric meshes from multi-labeled medical datasets. Zhang and Qian [51] use a modified Dual Contouring algorithm where ambiguous grid cubes are decomposed into tetrahedral cells to generate tetrahedrons and polyhedrons for multi-material datasets. Dillard et al. [71] present an interpolation method using a simple physical model to find likely region boundaries between image slices, and then apply Marching Tetrahedra [46] to produce a surface mesh. Smoothing and simplification methods are also presented to reduce the number of triangles. Meyers presents a variation of the Dynamic Particle Systems [72] to produce multi-material surface and tetrahedral meshes.

The standard Dual Contouring algorithm is unable to produce 2-manifold and watertight surface meshes due to the presence of ambiguous cubes. Chapter 3 presented a modified Dual Contouring algorithm that is capable of overcoming the limitations of standard DC. Ambiguous cubes were subdivided into a number of tetrahedral cells having regular size and shape. The centroids of these tetrahedral cells were used as minimizers, and novel polygon generation rules were used to produce single material surface meshes that were 2-manifold and watertight.

The work presented here is an extension of the algorithm presented in Chapter 3. The main contribution is the generation of geometrically correct multi-material surface meshes which contain non-manifold elements at the junctions where materials meet. The *sub-meshes* of each

individual material are 2-manifold and watertight on their own. The presented method is capable of producing multi-material interfaces or shared boundaries that are consistent between sub-meshes. It is also demonstrated that these multi-material surface meshes can be used to easily initialize multi-material tetrahedral meshes.

4.2 Multi-material vs. 2-Manifold

A mesh is defined as being 2-manifold if every edge of the mesh is shared by exactly two faces, and if the neighborhood of each vertex is the topological equivalent of a disk. A closed 2-manifold mesh \mathbf{S} satisfies the Euler-Poincaré condition $n + t - k = 2$. The closed surface mesh is also watertight if it does not contain any holes or cracks.

Since a multi-material surface will inherently contain non-manifold elements along material interfaces due to its multi-material nature, it is important to present a formal description of a mesh being multi-material, watertight and 2-manifold. A triangular mesh can be described as a set $\mathbf{S} = \{\mathbf{V}, \mathbf{E}, \mathbf{F}\}$ where \mathbf{V} is a set of n vertices $\{v_0, v_1, v_2, \dots, v_{n-1}\}$ with each $v_i \in \mathbb{R}^3$, \mathbf{E} is a set of k edges $\{e_0, e_1, e_2, \dots, e_{k-1}\}$ where each tuple e_i is an edge made up of two vertices, and \mathbf{F} is a set of t triangular faces $\{f_0, f_1, f_2, \dots, f_{t-1}\}$ where each tuple f_i describe a triangle made up of three vertices.

In a regular, *single-material*, grid-based surface meshing algorithms like Marching Cubes [30] or Dual Contouring [29], the input data is a binary volume or implicit function which describes a point as being either greater than or less than a given isovalue. A uniform grid is superimposed on the input data, and the corners of the grid cubes are designated as inside or outside the volume.

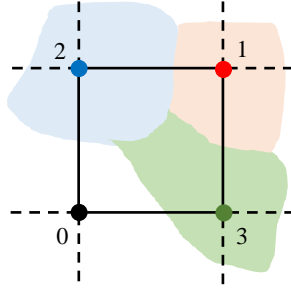


Fig. 30. A 2D example of a uniform grid superimposed on a multi-material domain. The four corners of the square occupy three different material domains as well as the background.

Multi-label or multi-material input data have *material indices* describing the different material subdomains of the input data. The material indices are typically implemented in the form of integer values with 0 indicating background and the positive integers describing different materials/labels. The corners of grid cubes are assigned these integer material indices, indicating that the cube's corner resides inside that material region of the input data. Fig. 30 shows an example of a 2D uniform grid superimposed on a multi-material domain. There are three materials in this example, red with material index 1, blue with material index 2 and green with material index 3, along with the background whose material index is 0. The four corners of the square are within each of these regions and are assigned their respective regions' material indices.

A multi-material mesh would have additional material information associated with its vertices and/or faces. In our implementation of a multi-material and 2-manifold Dual Contouring algorithm, we assign pairwise integer values to faces only, with the mechanism of the assignment explained in the next section. The triangular faces of a multi-material mesh \mathbf{S}_M of M materials can be described by the set $F = \{f_i^{p,q}\}$, where p and q are the assigned pairwise material indices

for a triangular face, and $p, q \in \mathbf{M}$ and $p \neq q$. For example, $f_0^{0,1}$ denotes the first triangle of the mesh with material indices 0 and 1. This means that the first triangle forms a plane between the background and Material 1. As another example, the face $f_{80}^{2,3}$ describes the triangle with index 80 forming a plane between materials 2 and 3. Fig. 31 depicts a more complete example. Fig. 31(a) shows a simple mesh comprising two materials with material indices 1 and 2, in red and blue colors, respectively. This is the actual, whole and complete mesh. Fig. 31(b) shows a cutout of the mesh. The green colored part of the mesh is the interface or shared boundary that lies between the two materials. In this example, all the faces of Material 1 (red) are of the form $f_i^{0,1}$, all faces of Material 2 (blue) are of the form $f_0^{0,2}$ and all faces of the shared boundary (green) are of the form $f_0^{1,2}$. Fig. 31(c) and (d) shows what the sub-meshes of each individual material and the shared boundary would look like if they were viewed separately.

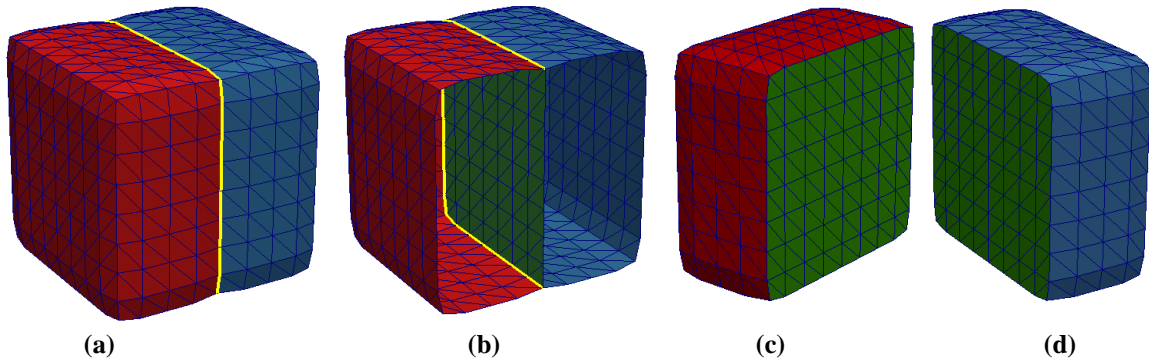


Fig. 31. A synthetic example of a multi-material mesh of two materials, with a shared boundary. (a) The whole and complete multi-material mesh, (b) a cutout of the mesh showing the green colored shared boundary, (c) the sub-mesh for Material 1, (d) the sub-mesh for Material 2.

One aspect of a multi-material mesh having shared boundaries is that non-manifold elements will occur at the junction where two or more materials meet. In Fig. 31(a) and (b), the junction where Materials 1 and 2 meet is outlined in yellow, and these edges are non-manifold, i.e., they are shared by more than two faces. While the multi-material mesh (Fig. 31(a)), as a whole, is not purely 2-manifold due to the presence of these non-manifold elements, the sub-meshes (Fig. 31(c) and (d)) of individual materials along with the shared boundary themselves are completely 2-manifold and watertight.

Fig. 32 shows another synthetic example of a multi-material mesh containing non-manifold elements, with sub-meshes for Material 1, 2 and 3 colored in red, blue and purple, respectively. In this figure, there is an additional non-manifold vertex, as shown in Fig. 32(a) and Fig. 32(b), which is shared between the sub-meshes of Material 1 and Material 2, along with non-manifold edges (demarcated in yellow). Fig. 32(b) shows a cutout of the mesh, and the shared boundary between the sub-meshes for Material 1 and Material 3 is colored green. Fig. 32(d), (e) and (f) shows the separated sub-meshes for each of the individual materials. Again, although the whole and complete multi-material mesh is not purely 2-manifold, the three sub-meshes, by themselves, are 2-manifold and watertight.

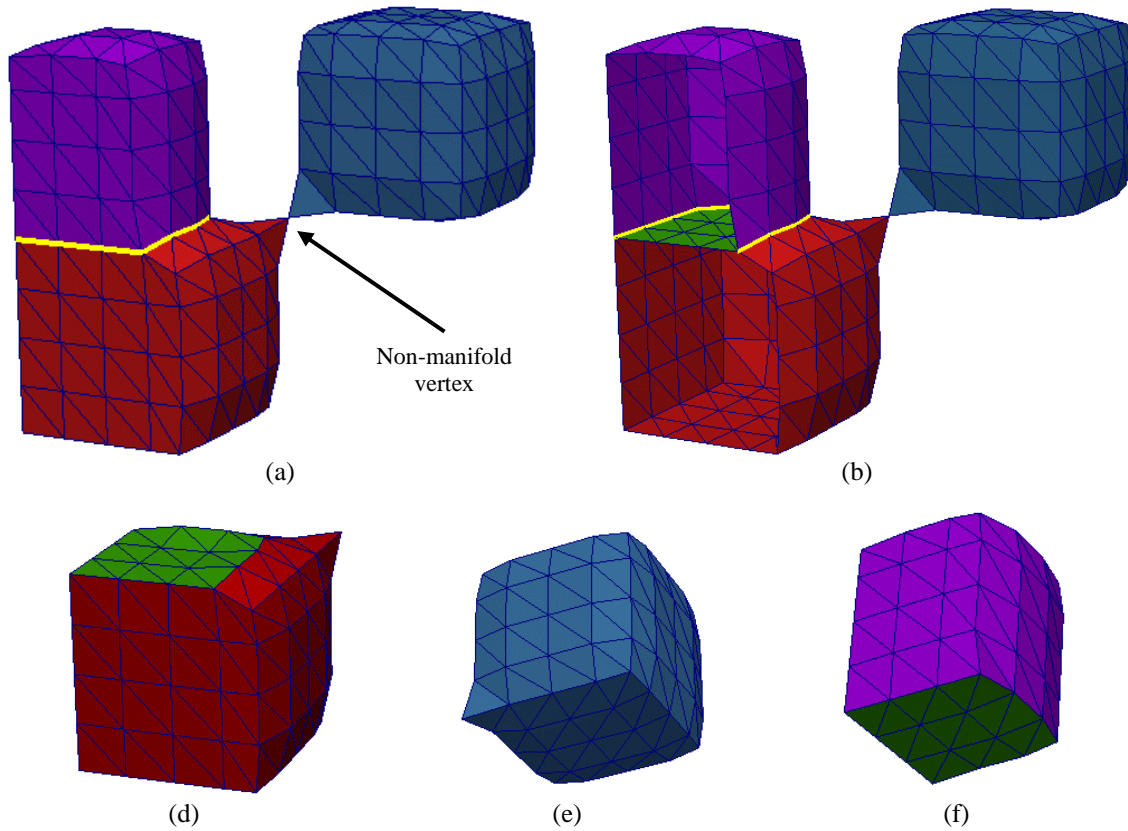


Fig. 32. A synthetic example of a multi-material mesh with three materials, having non-manifold edges and vertex. (a) The whole and complete multi-material mesh, (b) a cutout of the mesh to show the shared boundary between the red and purple meshes, (c) sub-mesh for material 1, (d) sub-mesh for Material 2, (e) sub-mesh for Material 3. All three sub-meshes are 2-manifold and watertight.

4.3 Ambiguous vs. Unambiguous

The surface meshing process begins by superimposing a uniform grid over the multi-labeled volume in question. As stated earlier, the corners of the grid cubes are assigned material indices to indicate which material domain they occupy. An octree is used to represent the

uniform grid. The corner configuration of the grid cubes can be of two types: (1) all the corners have the same material index, and (2) at least one corner has a material index that is different from the rest. The cubes whose corners all have the same material index are completely inside a particular material and are therefore discarded from the octree. Quadratic Error Functions (QEF), as shown in Equation (1) are used to generate a single minimizer for each grid cube.

In standard DC, the presence of ambiguous cubes, along with its single minimizer, is the reason non-manifold elements occur in the resulting surface mesh. Once the octree is generated, it is therefore necessary to identify *ambiguous* and *unambiguous* grid cubes. A grid cube is called ambiguous because there are more than one possible surface intersecting the cube. Ambiguous cubes have either a *face ambiguity* or an *interior ambiguity*. Face ambiguity occur when the material index of two diagonal corners of a face are the same while the other two corners have different material indices. Interior ambiguity occur when the material index of two diagonally opposite corners of the cube have the same value. Fig. 33(a) and Fig. 33(b) show an example of face and interior ambiguities, respectively.

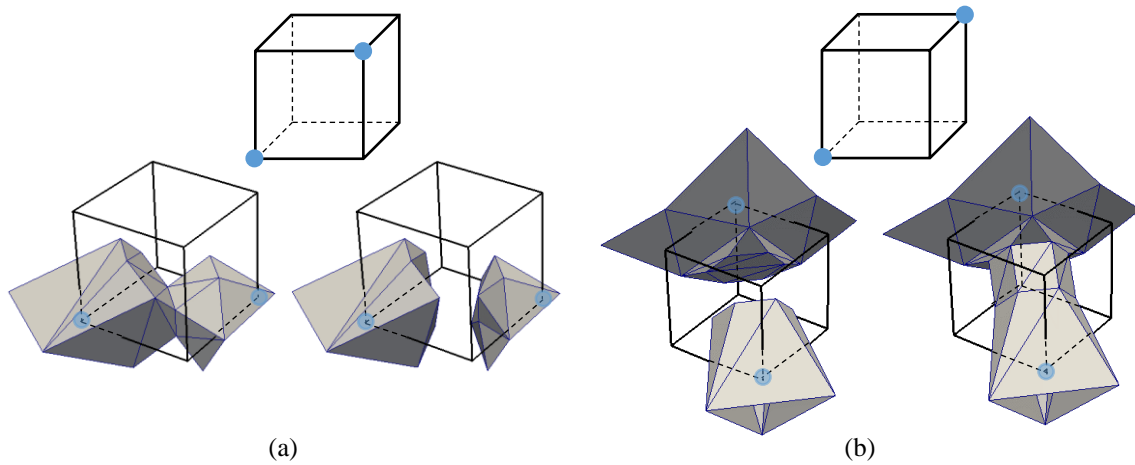


Fig. 33. Examples of ambiguous cubes. (a) Face ambiguity, and the two possible surfaces (b) Interior ambiguity and the two possible surfaces.

In the single material case each grid cube can have 2^8 or 256 possible configurations, however these can be reduced to 14 fundamental cases when rotation and symmetry are taken into account, as shown in Fig. 13. Out of these 14 configurations, Cases 0, 1, 2, 5, 8, 9 and 11 are simple cases for which there is only a single surface that intersects the grid cube (no surface for Case 0). Reference [50] presents a comprehensive list of 31 possible cases and their surface intersections when face and interior ambiguity are taken into consideration.

It may be possible, though costly, to identify each ambiguous case in the single material case. However, in the multi-material case where a cube can have a total of 8^8 permutations, it is simply not feasible to identify ambiguity for each grid cube. Indicator variables can be used, as demonstrated in [51], to reduce a multi-labeled cube into a series of binary labeled cubes for each material, and then determine ambiguity for each material.

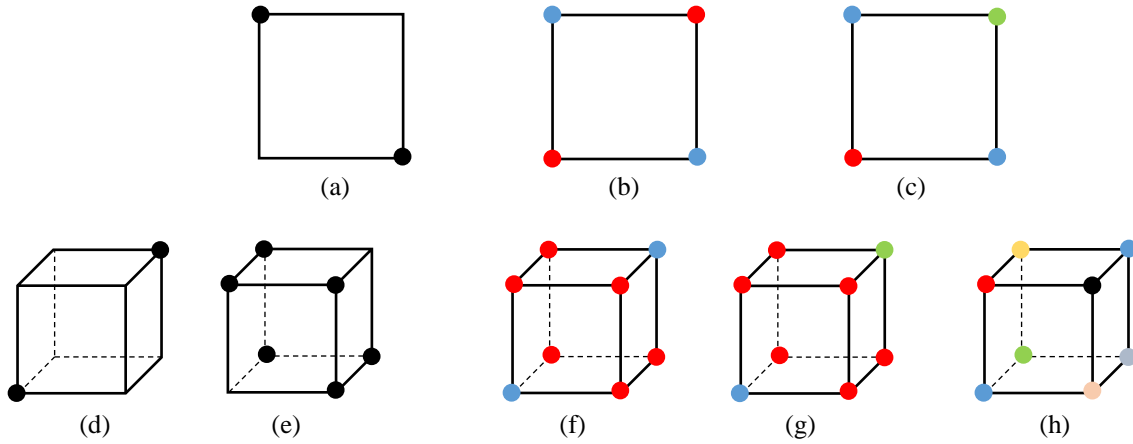


Fig. 34. Identifying ambiguity for multi-material domains. (Top row) Face ambiguity, (a) One possible configuration for single material contouring, (b-c) two possible configurations for multi-material contouring. (Bottom row) Interior ambiguity. (d-e) two possible configurations for single material contouring, (f-h) three possible configurations for multi-material contouring.

To determine ambiguity, we note that every ambiguous cube (Cases 3, 6, 7, 10, 12, and 13), with the exception of Case 4, show at least one instance of face ambiguity. One face of the cube consisting of four points can have a total of 2^4 and 4^4 permutations, for single material and multi-material situations, respectively. After taking symmetry and rotation into account, the number of permutations for a single face exhibiting face ambiguity can be reduced to 1 and 2, for single material and multi-material situations, respectively, as shown in Fig. 34 (top row).

The situation is slightly more complex for Case 4 ambiguity. In the single material case, standard DC produces a non-manifold vertex when (1) only two diagonally opposite corners of the cube are inside the volume and the remaining corners are outside the volume, as shown in Fig. 34(d), or (2) when only two diagonally opposite corners are outside the volume and the remaining corners are inside the volume, as shown in Fig. 34(e). This concept holds true in the

multi-material case involving two materials, three materials, or more than three materials in the most extreme circumstance, as shown in Fig. 34(f-h).

In the proposed multi-material and 2-manifold Dual Contouring method, a cube is identified as either ambiguous or unambiguous, by determining face or interior ambiguity. The exact nature of the ambiguity (i.e. Case 3 or Case 4 or Case 10 or Case 12) is not necessary. This allows the proposed method to be extremely generalized in its approach.

4.4 Tetrahedral Decomposition of Ambiguous Cubes

Once grid cubes have been classified into ambiguous and unambiguous cubes, the next stage is to compute minimizers for each cubes. In standard DC, one minimizer is computed for every cube. This single minimizer in ambiguous cubes is insufficient to adequately represent the different topologies that are possible in ambiguous cubes. Hence, when used in conjunction with the minimizers of neighboring cubes to generate surfaces, non-manifold edges and vertices appear in the resulting surface mesh.

Volume smoothing is one method that can be used to reduce the chances of generating non-manifold edges and vertices. Volume smoothing methods like binary or grayscale morphology, or blurring kernels such as a Gaussian kernel, can simplify the topology of the data in question, thus reducing the number of ambiguous cubes. However, such approaches do not address the underlying issue of what to do when ambiguous cubes are inevitably encountered. Since the presence of a single minimizer in ambiguous cubes is the main reason for the creation of non-manifold elements, the obvious solution is to introduce multiple minimizers into ambiguous cubes. In [50], based on the intersections between the isosurface and the cube,

multiple minimizers are introduced into the cube and new polygons are created by reconnecting vertices of the mesh to the newly inserted minimizers.

Sohn [52] shows that a cube can be decomposed into a number of tetrahedrons while preserving the topology. The manner of the tetrahedral decomposition is dependent on the number of face and body saddle points of the cube. This concept was exploited by Zhang and Qian in [51] by decomposing a cube into twelve tetrahedral cells of similar size and shape. Our proposed method decomposes an ambiguous cube into a maximum of twelve tetrahedral cells in the same manner as [51]. The center of the cube acts as a common point for all tetrahedrons. The center, along with three corners of a face make up the four vertices of a single tetrahedron. Each face of the cube forms the base for two tetrahedrons. Fig. 35 (left) shows an example of two tetrahedrons created from the front face of a cube.

The choice of the face diagonal is important because it can cause a potential change in the local topology. The basic rules for creating the diagonal are as follows:

- If the two diagonally opposite corners are contiguous with each other through the interior of the volume, then create a face diagonal between the two corners.
- If the two diagonally opposite corners are inside the volume, but not contiguous with each other through the interior, then create a face diagonal using the other two corners.
- For all other cases, any appropriate face diagonal can be used.

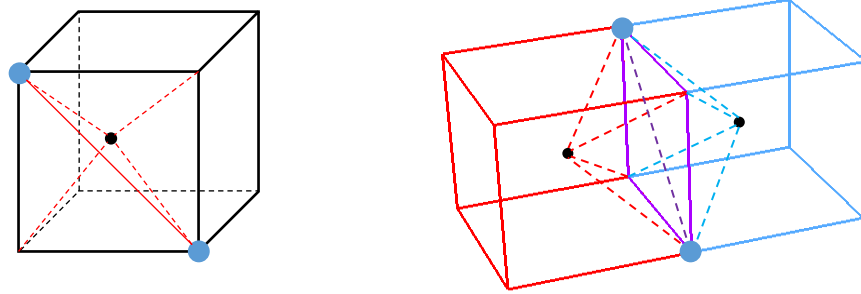


Fig. 35. An illustration of the decomposition of ambiguous cubes. (left) A partially decomposed ambiguous cube having two tetrahedrons. The front face is the base of two tetrahedrons (red lines), and the center of the cube is a common point for all tetrahedra. (right) Two adjacently placed ambiguous cubes sharing a face (purple square). The face diagonal (purple dashed line) across the shared face is consistent for both cubes.

This rule also allows two adjacently placed ambiguous cubes sharing a face to have consistent face diagonals. The consistency of the diagonal of shared faces between two ambiguous cubes is particularly important during polygon generation. Fig. 35 (right) shows an example of two ambiguous cubes with a shared face (colored purple). The red and blue dashed lines represent two tetrahedrons for the red and blue cubes, respectively, and the shared face serves as the base for all four tetrahedrons. Notice that the face diagonal (purple dashed line) is the same for both cubes.

An *interior edge* is an edge of a tetrahedron that is made up of a corner of the parent cube and the center of the cube. A *sign change edge* is an edge of a tetrahedron or a grid cube whose end points have different material indices. A *valid* tetrahedron is one in which at least one edge is a sign change edge. Although an ambiguous cube is decomposed into a total of twelve

tetrahedrons, our proposed strategy makes use of only valid tetrahedra for polygon generation. For each valid tetrahedron, the centroid is computed and used as its minimizer.

As mentioned above, in the case of single material surfacing, a cube can have 31 unique configurations when face and interior ambiguity are taken into consideration. The number of configurations drastically increase when dealing with multi-material surfacing. Each cube configuration is representative of a different surface and/or topology, and establishing polygon generation rules for each configuration is difficult at best for single material surfacing, and near impossible when dealing with multi-material surfacing. In order to be effective and feasible, polygon generation rules need to be generalized enough to handle unambiguous cubes, as well as any ambiguous case, for both single and multi-material surfacing.

The next section details a set of polygon generation rules which are applied to unambiguous cubes as well as any ambiguous cubes that have been subdivided into a number of tetrahedral cells. These rules are capable of producing multi-material, 2-manifold and watertight surfaces irrespective of whether the cubes are ambiguous or not. The rules also work independent of the number of tetrahedral cells of an ambiguous cube.

4.5 Polygon Generation

In standard DC, polygon generation is based on analyzing minimal edges: for each minimal edge, create a polygon using the minimizers of all cubes that contain the minimal edge. For DC using non-adaptive octrees, a minimal edge will be contained by 4 cubes, and the cubes' four minimizers are used to create a quad or two triangles. The proposed method to generate multi-material surfaces also relies on analyzing minimal edges contained by four cubes. Additionally, since ambiguous cubes will contain a number of interior edges due to their

decomposition into a set of tetrahedral cells, it is necessary to analyze these interior edges and face diagonals as well. Zhang and Qian follow a similar approach in [51] to generate tetrahedral meshes.

The proposed method to generate multi-material and 2-manifold surface meshes is only applicable to non-adaptive octrees. Minimal edges will always be shared by four cubes. If all four cubes are unambiguous, then we create two triangles using the minimizers of the four cubes. On the other hand, if any of the four cubes exhibit ambiguity, then the following rules are applied: (1) Minimal Edge Rule, (2) Face Diagonal Rule, and (3) Interior Edge Rule. These three rules are similar as the ones presented in Chapter 3, except that these rules can now be used to produce multi-material meshes.

4.6 Verifying 2-Manifoldness in Multi-material Meshes

Multi-material meshes are inherently non-manifold due to their multi-material nature. Specifically, the junctions where two or more materials meet will contain non-manifold edges and/or vertices. The proposed algorithm is capable of generating multi-material surface meshes where each material's sub-mesh is a watertight and 2-manifold mesh. In order to verify this, the whole multi-material mesh is split into each material's sub-meshes, as shown in Fig. 31 and Fig. 32. For each sub-mesh, the opensource software MeshLab [53] is used, instead of relying on visual inspection, to detect the presence of non-manifold edges and vertices and boundary edges. MeshLab is an open source mesh processing tool that makes extensive use of the VCG (Visualization and Computer Graphics) Library. Our experiments show that even though the whole multi-material mesh can have non-manifold elements, the individual material's sub-mesh is always watertight and 2-manifold.

4.7 Mesh Smoothing

As discussed in Chapter 3, one limitation of the proposed method is that the surface of the resulting multi-material meshes exhibit a staircase-like effect. The reason for this is because the polygon generation rules always assume that minimizers will always be inside their respective cubes. This is not an issue for minimizers of tetrahedral cells because these minimizers are the centroids of their respective tetrahedrons. In the case of unambiguous cubes, the QR decomposition method used to compute minimizers can, in some situations, result in the minimizer being placed outside its respective cube. When minimizers are computed to be outside their respective cubes the surface mesh has an overall smoother look while containing cracks and intersecting triangles, as shown in Fig. 27 (right). On the other hand, when minimizers are constrained to remain within their respective cubes, the surface mesh is crack-free and has no intersecting triangles, but exhibits a staircase effect, as shown in Fig. 27 (left).

Mesh smoothing can be performed as either a pre-processing stage or a post-processing stage. In a pre-processing stage, the input volume can be smoothed using such as grayscale or binary morphological operations, or blurring with a Gaussian kernel. Extra care needs to be taken for multi-material volumes so that material interfaces are not corrupted. In a post-processing stage, mesh smoothing techniques such as *Laplacian* or *Taubin* smoothing can be used. Both Laplacian and Taubin smoothing are algorithms that changes the position of all vertices by computing a new position based on neighboring vertices and triangles/faces. Laplacian smoothing can produce shrinkage whereas Taubin smoothing can avoid shrinkage. We used the *vtkWindowedSincPolyDataFilter* class (which is an implementation of a Taubin filter) available in VTK (Visualization Toolkit) to smooth our multi-material surface meshes as a post-processing stage. Another advantage of using Taubin filters over Laplacian filters is that it takes relatively

fewer number of iterations to have discernable effects. The results of the smoothing process are shown in Fig. 36. Of course, care must be taken when using any sort of smoothing filters. Too many iterations can potentially lead to undesirable changes, such as triangles becoming slivers.

4.8 Results and Discussion

Fig. 31 and Fig. 32 shows two examples of multi-material and 2-manifold surface meshes generated with the proposed method. The proposed method has also been applied on a digital atlas [54] of the basal ganglia and thalamus to produce multi-material surface meshes of anatomical structures. The atlas is in MINC 2.0 format (Medical Imaging NetCDF) and contains a total of 92 labeled structures. The atlas consists of $334 \times 334 \times 334$ voxels with a stepsize of 0.3 mm.

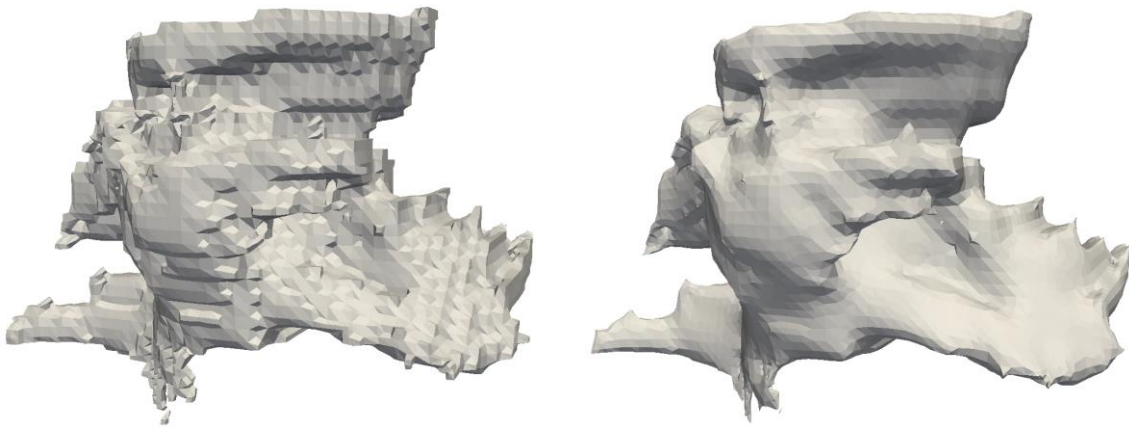


Fig. 36. Taubin smoothing. (left) Unsmoothed mesh. (right) Mesh smoothed using Taubin smoothing, with 10 iterations.

A multi-material mesh of five anatomical structures in close proximity to each other has been generated using the proposed method. Fig. 37 shows a multi-material mesh of the striatum (purple), interior capsule (yellow), Globus Pallidus (green), Globus Pallidus External (red) and Globus Pallidus Internal (blue). The upper figure shows both (left and right) parts of the mesh, and the lower figure shows a close-up of the right side of the mesh. Typically the striatum and the internal capsule almost completely surrounds the Globus Pallidus, so the striatum and internal capsule are rendered transparent in this figure. The white lines denote the non-manifold edges along the material interfaces. For the sake of simplicity, the shared boundaries between materials have not been differentiated in Fig. 37, even though they do exist. The shared boundaries are shown in Fig. 38. Fig. 39 shows a cross-section of the multi-material mesh in Fig. 37 and Fig. 38. TABLE 7 shows the quality of the triangles of the meshes. The overall quality of the meshes of the triangles of the meshes is roughly 0.74 on average. The poor quality triangles of the mesh are a result of the implementation of the Minimal Edge Rule. In the current implementation of the proposed method, Delaunay-based triangulation schemes have not been used due to their tendency to enforce convex polygonization.

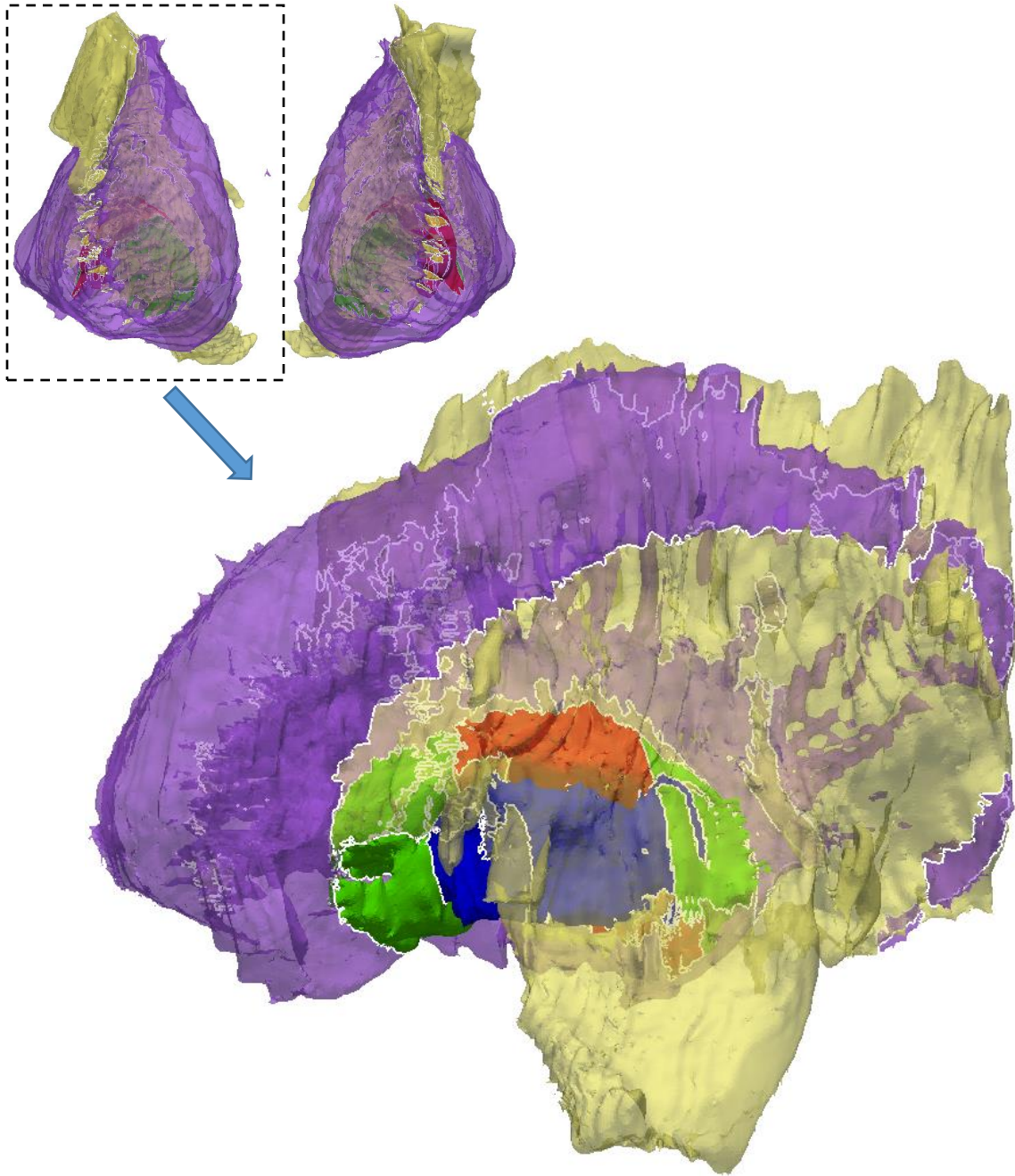


Fig. 37. A multi-material representation of the striatum (purple), internal capsule (yellow), Globus Pallidus (green), Globus Pallidus External (red) and Globus Pallidus Internal (blue). The striatum and internal capsule are rendered transparent. The white lines represent the non-manifold edges where two materials meet. Each structure has a single color. The shared boundaries are not depicted here.

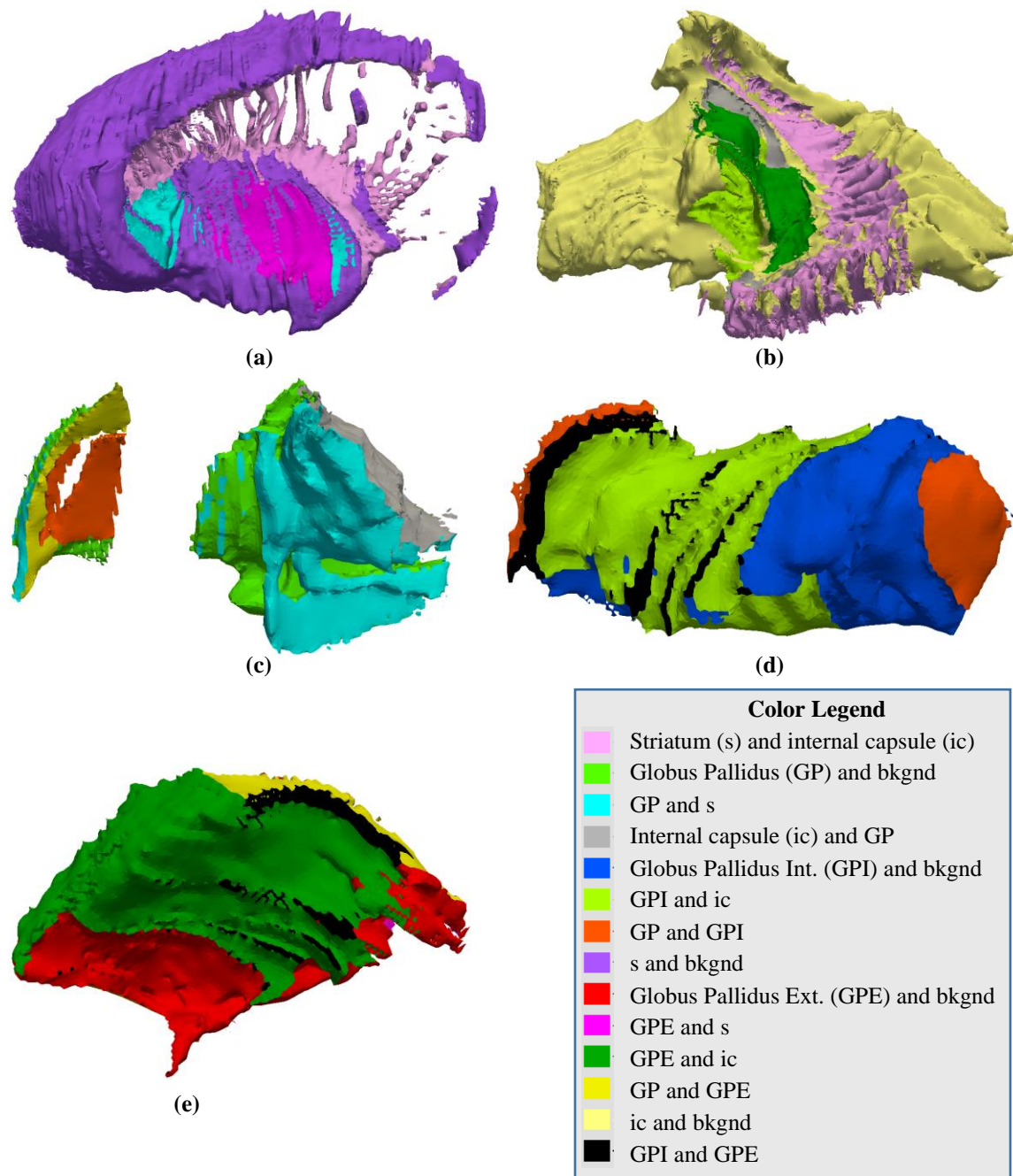


Fig. 38. An illustration of the complexity of the shared boundaries between the (a) striatum, (b) internal capsule, (c) Globus Pallidus, (d) Globus Pallidus Internal and (e) Globus Pallidus External. The different colors represent the parts of the mesh that are shared between the structures.

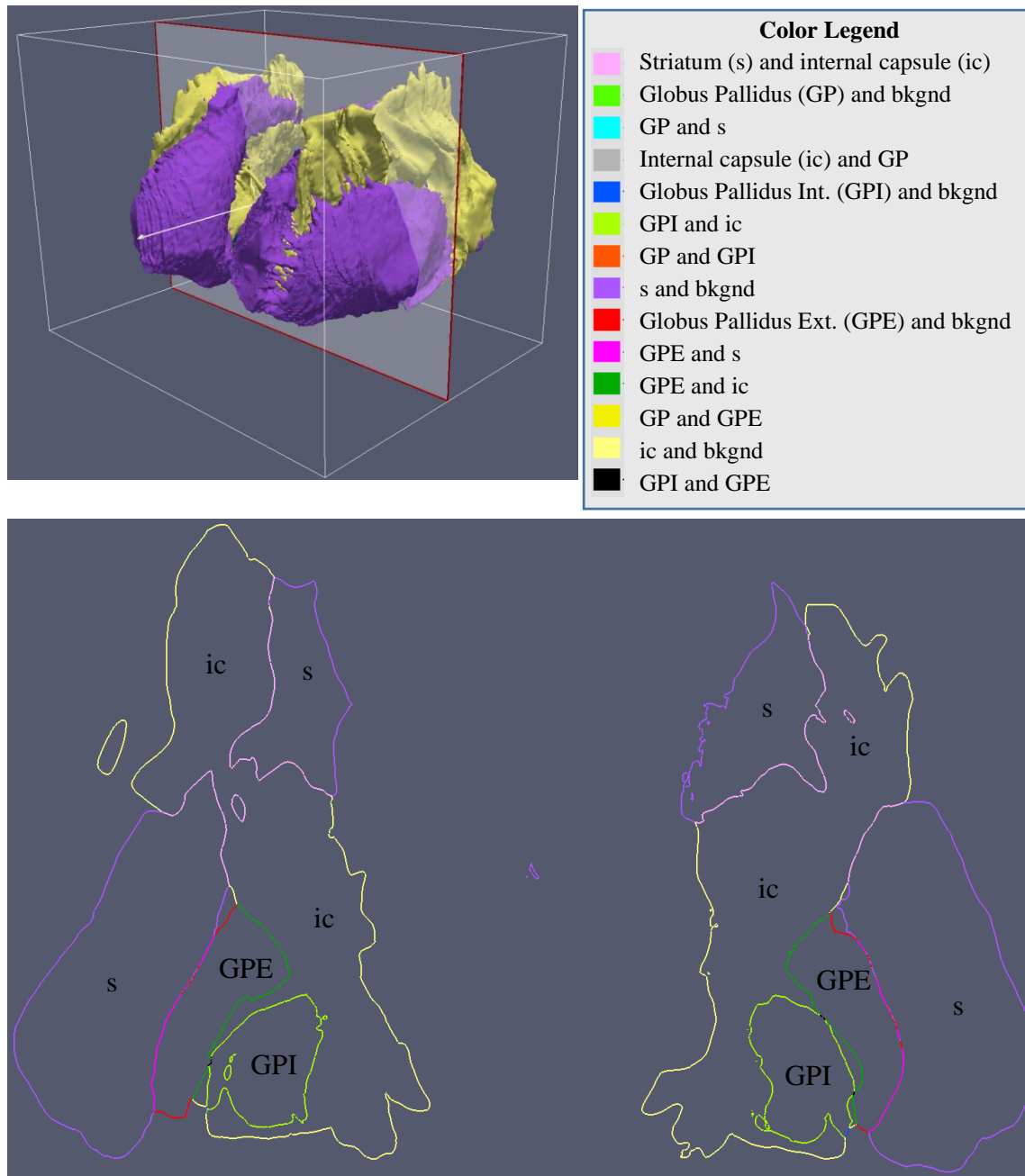


Fig. 39. A coronal slice of the multi-material mesh of the Striatum (s), Internal Capsule (ic), Globus Pallidus (GP), Globus Pallidus Internal (GPI) and Globus Pallidus External (GPE). The different colors represent the different shared boundaries between two structures.

TABLE 7
MESH QUALITY REPORT FOR STRIATUM, INTERNAL CAPSULE, AND GLOBUS
PALLIDUS GROUP

	Atlas Label	Structure Name	Number of triangles	Number of vertices	Average Radius Ratio (worst value)
Fig. 37		Whole mesh	1600205	777287	0.742725 (0.0000349219)
Fig. 38 (a)	1	Striatum	843580	420016	0.741234 (0.0000349219)
Fig. 38 (b)	4	Internal Capsule	1024134	509799	0.743144 (0.0000349219)
Fig. 38 (c)	5	Globus Pallidus	76966	38083	0.73666 (0.00165458)
Fig. 38 (d)	11	Globus Pallidus Internal	73816	36456	0.733899 (0.00126407)
Fig. 38 (e)	12	Globus Pallidus External	74998	37249	0.733607 (0.000459615)

The triangulation in the Minimal Edge Rule is a very basic and naïve approach, and is not configured towards producing well-shaped triangles, and sometimes can result in poor quality triangles.

Another multi-material surface mesh has been generated using the proposed method. This mesh, shown in Fig. 40, represents the thalamus. This mesh consists of 34 separate labels. In Fig. 40 the different colors represent the different parts of the thalamus, and the white lines represent the non-manifold edge of material interfaces. Fig. 41 shows a cross-section of the mesh of the thalamus.

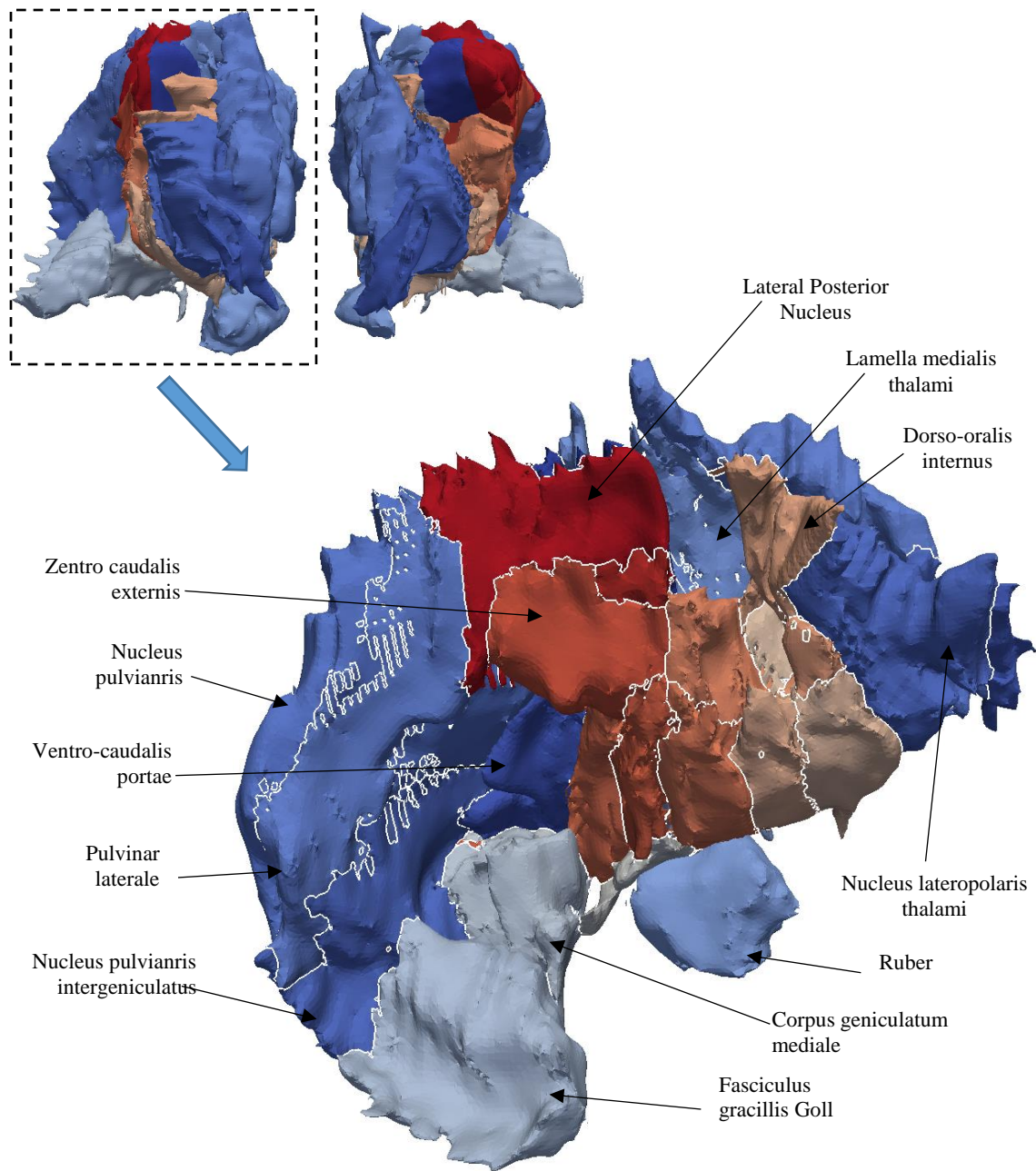


Fig. 40. A multi-material representation of the thalamus.

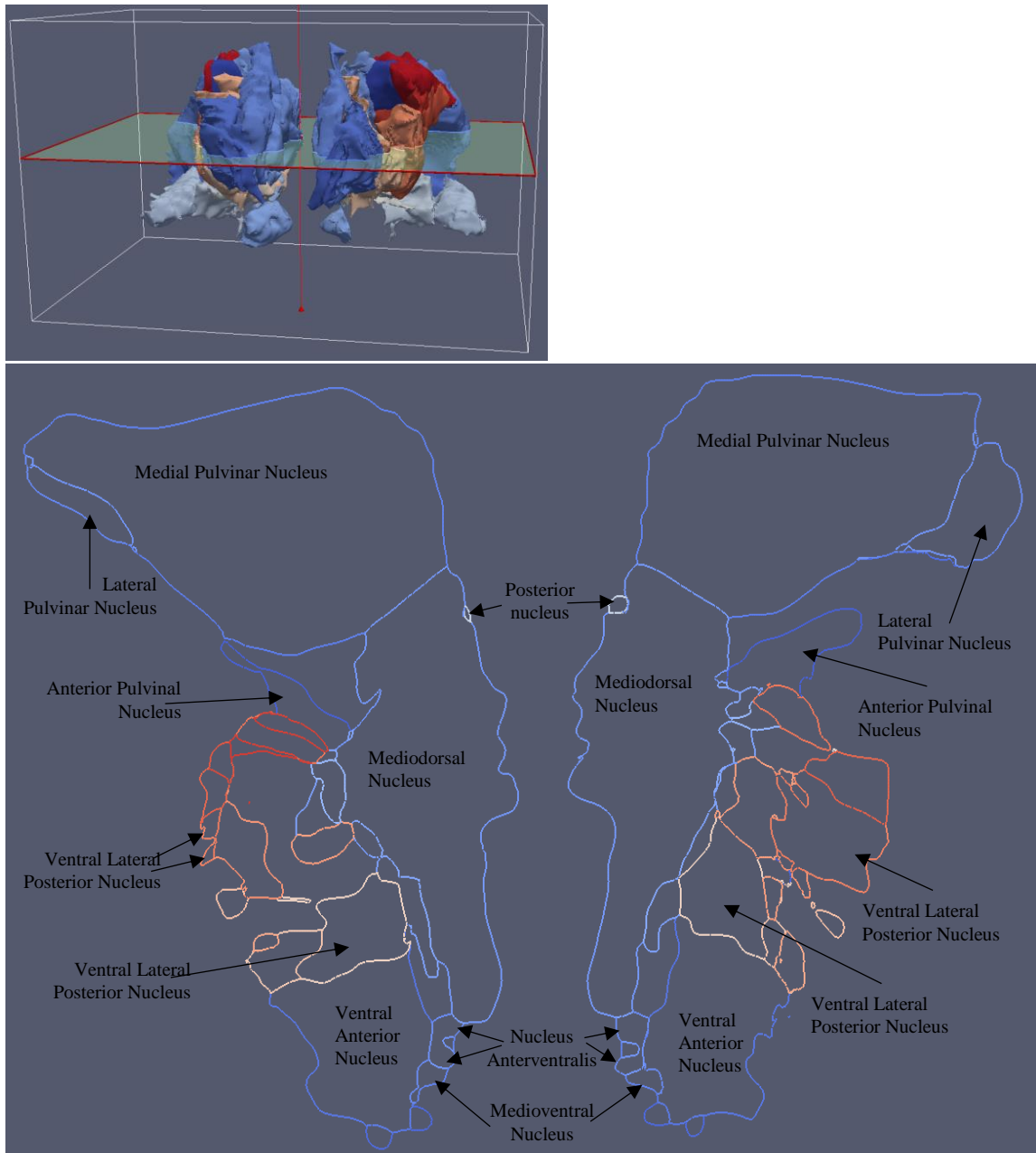


Fig. 41. A cross section slice of the multi-material mesh of the thalamus. The different colors represent the different constituent components of the thalamus.

TABLE 8
MESH QUALITY REPORT FOR COMPONENTS OF THE THALAMUS

Atlas Label	Structure Name	Number of Triangles	Number of vertices	Average Radius Ratio (Worst value)
	Whole Mesh	661493	307493	0.735966 (0.000063037)
22	Stria medullaris thalami (st. m)	6050	3033	0.76577 (0.00395043)
26	Nucleus lateropolaris thalami (Lpo)	39504	19680	0.741478 (0.00113207)
27	Nucleus fasciculosus thalami (Fa)	13204	6564	0.746772 (0.000264277)
28	Nucleus Anterior Principalis (Apr)	22454	11183	0.748473 (0.00318234)
37	Nucleus Medialis (M)	82294	40793	0.739118 (0.0017311)
40	Lamella medialis thalami (La. M.)	68054	33337	0.726119 (0.000307618)
48	Ruber (Ru)	14874	7351	0.743886 (0.00555594)
49	Nucleus Centralis (Ce.)	42170	20705	0.723844 (0.000759698)
51	Nucleus Parafasciculairs (Pf.)	22088	10828	0.709835 (0.000788477)
60	Fasciculus gracillis Goll (G)	24244	12002	0.727113 (0.00374679)
68	Corpus geniculatum mediale (G.m/G.Md)	41594	20573	0.737691 (0.000436662)
70	Nucleus Limitans (Li)	20700	10286	0.736441 (0.000451716)
71	Ventro-caudalis parvocell (V.c.pc)	9742	4831	0.744686 (0.000759698)
81	Ventro-oralis medialis (V.o.m.)	7670	3835	0.734398 (0.00333334)
86	Ventro-oralis internus (V.o.i.)	22588	11204	0.735069 (0.00108321)
87	Ventro-oralis anterior (V.o.a)	26420	13034	0.721205 (0.000506435)

TABLE 8 (cont.)

88	Ventro-oralis posterior (V.o.p.)	18284	9034	0.730065 (0.00126571)
89	Dorso-oralis internus (D.o.i)	19044	9410	0.73692 (0.0058984)
90	Zentrolateralis oralis (Z.o.)	4346	2137	0.716008 (0.00813926)
91	Ventro-intermedius internus (V.im.i)	20438	10077	0.728127 (0.000991324)
92	Zentro-lateralis externus (Z.im.e)	12276	6026	0.724353 (0.000265455)
94	Ventro-intermedius externus (V.im.e)	23398	11567	0.727223 (0.00260416)
95	Ventro-caudalis internus (V.c.i)	23586	11651	0.730712 (0.00644611)
96	Ventro-caudalis anterior internus (V.c.a.e)	12306	6117	0.737476 (0.00371692)
97	Zentro caudalis externis (Z.c.e)	15456	7684	0.733257 (0.00440189)
98	Zentro caudalis internis (Z.c.i)	8906	4439	0.736896 (0.00606832)
99	Dorso-caudalis (D.c.)	3890	1933	0.740567 (0.00468903)
100	Nucleus pulvinaris orolateralis (Pu.o.l.)	34284	16996	0.737573 (0.00468903)
101	Nucleus pulvinaris oromedialis (Pu.o.m.)	26474	13139	0.736356 (0.000479491)
102	Ventro-caudalis portae (V.c.por)	25712	12764	0.749486 (0.00212645)
104	Nucleus ventrointermedius internus (V.im.i)	4148	2056	0.725728 (0.00270778)
105	Nucleus pulvinaris intergeniculatus (Pu.ig)	35828	17580	0.721556 (0.000063037)
106	Nucleus pulvinaris (Pu.m)	86460	43066	0.746412 (0.000191482)
107	Pulvinar laterale (Pu.l)	37484	18458	0.724573 (0.000888158)

4.9 Multi-Material Tetrahedral Mesh Generation

As discussed in Chapter 3, among the various techniques [58] available for generating tetrahedral meshes, Delaunay-based methods are the most popular. An initial surface mesh, known as a Piecewise Linear Complex (PLC), is used to define the boundary of the domain to be meshed. The difficulty here is the need to ensure that the surface mesh is 2-manifold, and watertight. This difficulty is further compounded in a multi-material scenario where, in addition to 2-manifoldness and watertightness requirements, the user must also ensure that the material interface (or shared boundary) is consistent.

The proposed DC algorithm can produce multi-material 2-manifold surface meshes where the *shared boundary or material interface is consistent*, and each sub-mesh is *watertight and 2-manifold*. Such multi-material meshes can be readily used as the input PLC in the generation of multi-material tetrahedral meshes.

Fig. 42 shows an example of a multi-material tetrahedral mesh. Fig. 42(a) shows a multi-material surface mesh of the subthalamic nucleus and substantia nigra that has been generated using the proposed DC algorithm, and is used in TetGen [59] to generate a multi-material tetrahedral mesh. Fig. 42(b) shows a cutout of the generated multi-material tetrahedral mesh, with the internal tetrahedra colored purple. Fig. 42(c) shows a cutout of the tetrahedral mesh with the internal tetrahedra rendered transparent in order to show the material interface (colored blue).

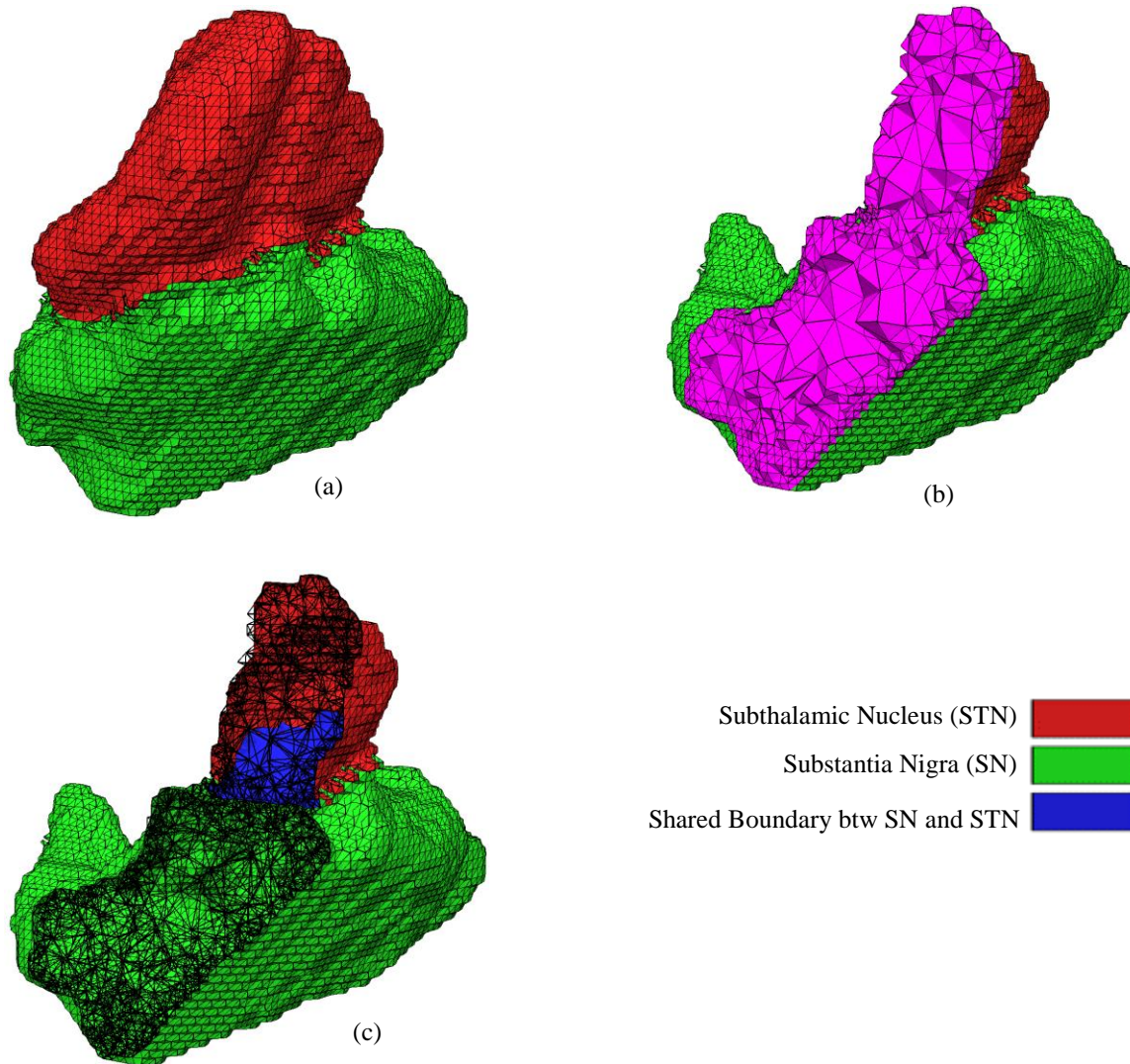


Fig. 42. An example of a multi-material tetrahedral mesh. (Top left) The multi-material 2-manifold surface mesh, (top right) cutout of the tetrahedral mesh, where the purple color indicates internal tetrahedra, (lower left) cutout of the tetrahedral mesh, with the internal tetrahedra rendered transparent to show the shared boundary.

4.10 Conclusions

In this chapter, the 2-manifold Dual Contouring method from Chapter 3 has been extended to produce multi-material surface meshes. These meshes are multi-material as well as 2-manifold in the sense that the sub-meshes of individual materials are, by themselves, watertight and 2-manifold, even though the whole mesh can contain non-manifold elements along material interfaces. The sub-meshes have consistent shared boundaries. Each triangle of the mesh is identified using pairwise material indices. The proposed method is effective in generating geometrically correct, as well as accurate representation of anatomical structures.

The proposed method exhibits some of the same limitations as the single material 2-manifold DC method, namely that the mesh has a staircase-like effect. This limitation can be somewhat mitigated through careful use of a post-processing smoothing filter.

CHAPTER 5

MULTI-MATERIAL 2-SIMPLEX DEFORMABLE MESH

Printed and digital atlases are important tools for medical interventions. While these atlases are able to provide reasonable guidance in identifying anatomical structures, they do not take into account the large variations in the shape and size of anatomical structures that occur from patient to patient. An accurate depiction of the anatomy is especially important for surgical interventions like deep brain stimulation, where even small inaccuracies can result in potentially dangerous complications. In these situations, a patient-specific representation of the anatomical structures of interest is preferred, rather than a generic printed or digital atlas. Deformable surface meshes are one way to achieve such patient-specific representations. An initial mesh model of the structures of interest can be generated using the digital atlas, and then deformed using patient-specific CT or MR data. Not only does the deformed surface mesh accurately represent the structures of interest, the model, being a surface mesh, is sparser than volumetric representations, such as tetrahedral or hexahedral meshes, and thus reduces computational overhead.

This chapter presents an extension of the discrete deformable 2-simplex mesh. The innovation here is a multi-material implementation of deformable meshes which can be initialized with relative ease, and deformed using MR data to accurately represent anatomical structures of the deep brain region.

5.1 Background

Segmentation, in the context of medical imaging, is the process where an image or volume is partitioned into non-overlapping connected regions using some characteristics intrinsic

to the image or volume. Segmentation is important in medical imaging because it provides a considerable amount of non-invasive information about the structures of the human body, as well as aid in studying anatomical structures and track the progression of diseases.

Atlas based segmentation is a paradigm where anatomical atlases are used as a guide for the segmentation of new images, which effectively transforms a segmentation problem into a non-rigid registration problem. This category of segmentation can be divided into two parts: single atlas-based and multiple atlas-based. In single atlas-based segmentation, an atlas is constructed from one or more labeled images, and then registered to the target image. The accuracy of the segmentation depends on the registration process. Examples of single atlas-based segmentation can be found in [73-76]. In multi atlas-based segmentation, many independently built atlases are registered to the target image, and the resulting segmentation labels are combined. The advantage of multiple atlas-based segmentation over the single atlas-based approach is that more information is available due to the use of many independent atlases, and the drawback is the number of registration steps required to produce the final segmentation. Examples of multiple atlas-based segmentation are [77-79].

Deformable models were first presented in 1986 by Sederberg and Parry in [80]. Terzopoulos in [81] coined the term deformable models and applied physical properties to objects. The basic idea behind the use of deformable models for segmentation is that the model will evolve using internal and external forces, eventually coincide with the anatomical boundary, and the interior of this boundary is considered the tissue or organ of interest. The internal force will ensure a smooth surface, and the external force will move the surface of the model towards the object boundary. To date, there are various categories of deformable models based on

deformable splines, mass-spring models, tensor-mass models, finite element models, etc. Meier provides a thorough survey of the various deformable models in [82].

Snakes [83] is the first deformable model to be used for segmentation using spline-based internal and image-features-based external forces. In order to be effective, the Snakes model needed to be positioned close to the boundary of the object in order to be effective. Cootes et al. [84] introduced the notion of using statistical shape information (called active shape models) to aid in the segmentation process. This process is dependent upon having landmarks for a number of training images, and one-to-one point correspondence between the boundaries of the objects in the training dataset.

Delingette formulated a specific type of deformable models: the k -simplex mesh [31, 32], for 3D shape reconstruction and segmentation. A k -simplex mesh is defined as a k -manifold discrete mesh where each vertex is linked to exactly $k + 1$ neighboring vertices. Delingette specifies a simplex angle and metric parameters, which can be used to represent the position of any vertex with respect to its neighbors. Fig. 7 shows an illustration of the vertex \mathbf{P} , along with its three neighbors \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 . In Fig. 7, the simplex angle φ , defined by Equation (7), represents the angle between the segments that join \mathbf{P} to the projection of the circle C . Gilles in [33] computes the simplex angle as the height h of \mathbf{P} above the plane made by its three neighbors \mathbf{P}_1 , \mathbf{P}_2 and \mathbf{P}_3 . 2-Simplex meshes are dual to triangular meshes, as shown in Fig. 6. Resolution control can also be achieved through the use of topology operators, as illustrated in Fig. 9. Simplex forces (where internal forces are based on mesh geometry, and external forces are based on input image gradients), together with enhancements such as shape constraints [85], smoothing parameters, shape memory and internal and external constraints [33, 39] and statistical shape information [86-88] have allowed simplex meshes to be used for accurate

segmentation of anatomical structures. Montagnat extended the simplex mesh for segmentation of the liver from CT scans, ventricles from MRI brain data and the myocardium from SPECT images [85, 89-91]. Gilles introduce a multi-surface 2-simplex model with collision detection and handling to segment muscles and bone from MR data [33, 39]. Tejos extended the 2D Diffusion Snakes process [92-94] and combined it with simplex meshes and statistical shape information for segmenting the patellar and femoral cartilage. Recently, Haq et al. [95-97] used a shape-aware based multi-surface simplex model to segment the lumbar spine, along with healthy and pathological intervertebral discs. Sultana et al. [98-100] used an implementation of the 1-simplex active contour incorporating shape statistics-aware deformation forces to segment the intracranial portion of ten pairs of cranial nerves attached to the brainstem, and build a patient-specific atlas of cranial nerves.

Registration is the process for determining the correspondence of features between images collected at different times or using different imaging modalities. Registration can be performed with images/volumes as well as discrete models (like surface and tetrahedral meshes). Therefore registration can be categorized into (1) image-to-image, (2) model-to-image and (3) model-to-model registration. The main goal of registration in the medical field is to find corresponding anatomical or functional locations in those images and models. Registration can be applied to images of the same subject but of different modalities. This is called multimodal registration. Registration can also be applied to images of the same subject but acquired over different time periods. This is called serial image registration. Another application of registration is to align images acquired from different subjects. The typical registration process consists of a transformation model which defines the transformation between images, a similarity metric which measures the degree of alignment between images, and an optimization method which

maximizes the similarity metric over the space of admissible transformations. Audette et al. [101] present a survey of registration methods based on the choice of transformation, similarity criterion and optimization method.

The transformation or deformation model could be physics-based [102-106], interpolation-based, knowledge-based or have task-specific constraints. The most common interpolation-based deformation models are use Radial Basis Functions [107-111] and Elastic Body Splines, as well as Free-Form Deformation. Holden [112] provides a review of different geometric transformation techniques used for non-rigid registration.

The choice of the similarity metric depends on the modality of the images to be registered. Images can be mono-modal, meaning that the images were produced by the same device. Simple metrics such as Sum of Squared Differences or Sum of Absolute Differences can be used, or more sophisticated methods such as Correlation Coefficient and Cross Correlation [113-115] can be used. In the case of multi-modal registration where the images can be of different modalities (like MR and CT images) the similarity metric is often based on information-theoretic approaches. Methods like Mutual Information [116-119], and its variations are the most common similarity metrics used. Other methods include landmark matching [120-122] for images and Iterative Closest Point (ICP) method [123] for surfaces and contours.

Optimization methods can be broadly categorized as continuous, discrete a heuristics-based approach. Continuous methods uses real valued variables, and the most commonly used methods are the Gradient Descent method, the Conjugate Gradient Descent method, Newton-type methods, Levenberg-Marquardt and Stochastic gradient descent. Discrete optimization methods uses a discrete set of values, and common techniques are graph-based or linear programming-based.

Fluck et al. [124] surveys a variety of GPU-accelerated registration approaches, and including different programming models and interfaces for developing software on a GPU. A similar discussion on image registration using multicore systems (high performance computing architecture, symmetric multiprocessing, massively multiprocessing, architectures with distributed memory and non-uniform memory access) is provided by Shams et al. in [125]. Liu et al. in [126] present a cooperative parallel architecture and a method to parallelize non-rigid registration algorithms using this architecture.

Archip et al. in [127] implement a non-rigid registration method as well as a system for visualizing multimodal data (augmented reality visualization with fMRI and DT-MRI) during neurosurgery. Intra-operative data was transferred to a super computing facility, processed, and then transferred back and displayed in the operating room (OR) during neurosurgery. A mean residual displacement of 1.82 mm after non-rigid registration is reported.

Wittek et al. [128] applied non-rigid registration methods for determining craniotomy-induced brain shift. Their procedure used hexahedral and tetrahedral meshes to represent the brain. The deformation field was predicted using specialized non-linear finite element algorithms for six cases, which exploits shape function precomputations based on a Lagrangian formulation. For each of the six cases, computation of the deformation field is reported to have taken less than 4 seconds using GPU implementation. The 95th percentile Hausdorff distance (previously used in [127, 129, 130]) between the registered surface of the ventricles of the preoperative segmentation and the intraoperative surface of the ventricles determined from intraoperative image segmentation was considered as the registration error measure. The average 95th percentile error for all six cases was reported as 1.733 mm, and the 75th percentile error was 0.9 mm.

Liu et al. [131] present a feature point-based non-rigid registration method to compensate for brain shift during tumor resection. Their method claims to be 16 times more accurate than rigid registration methods. However, they do not provide calculation times, and even though they used intraoperative MRI data for their evaluations, they are unclear as to whether the non-rigid registration process can be applied for real-time visualization of brain shift.

Oguro et al. [132] use a B-spline-based non-rigid registration algorithm for MRI-guided prostate brachytherapy. The registration was guided by image similarity based on Mutual Information. The processing time is reported as less than 5 minutes, and though the authors imply that this is feasible for clinical intervention, they admit that further validation is required before their procedure can be applied to clinical interventions.

5.2 Initializing a Multi-Material 2-Simplex Mesh

A 2-simplex mesh is 2-manifold discrete mesh where every vertex is connected to three neighboring vertices. A 2-simplex mesh undergoes deformations based on geometry-based internal forces and image-based external forces. The dynamics of each vertex can be modeled using the Newtonian law of motion shown in Equation (36).

$$m_i \frac{d^2 \mathbf{P}_i}{dt^2} = -\gamma \frac{d\mathbf{P}_i}{dt} + \mathbf{F}_{int} + \mathbf{F}_{ext} \quad (36)$$

where m_i is the mass, and \mathbf{P}_i is the position of a vertex of the mesh. \mathbf{F}_{int} represents all internal forces and \mathbf{F}_{ext} represents all the external forces acting on \mathbf{P}_i and γ represents a damping coefficient.

A 2-simplex mesh is the topological dual of a triangular mesh [32]. This geometric duality can be exploited to generate 2-simplex meshes from triangular surface meshes. Fig. 43 shows an illustration of the duality.

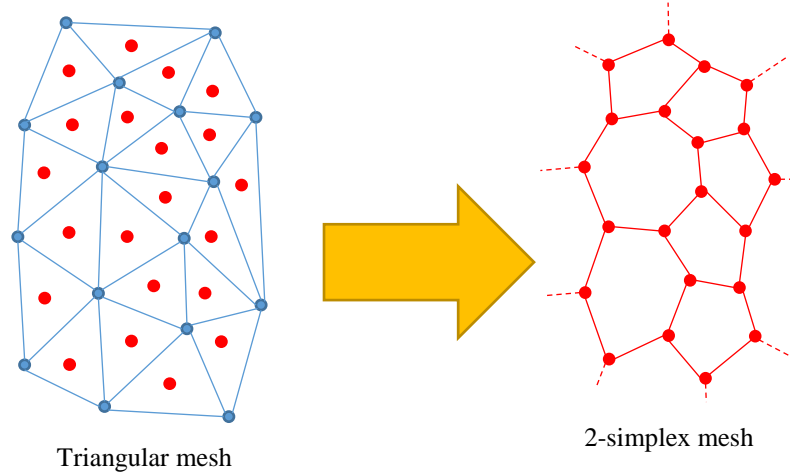


Fig. 43. Converting a triangular mesh into a simplex mesh using duality.

The centroids of triangles in the triangular mesh coincide with the vertices of the simplex mesh, and edges of the simplex mesh are created by linking these simplex vertices. The general algorithm of the conversion process can be summarized as follows:

- Step 1: Compute the centroids of each triangle of the triangular mesh.
- Step 2: For each i^{th} vertex of the triangular mesh,
 - Step 2.1: Locate all the triangles incident on the i^{th} vertex
 - Step 2.2: Use the centroids of these triangles to create one simplex cell.

Fig. 43 shows illustrates the process. In this figure, the red dots on the triangular mesh represents the centroids that eventually become vertices of the 2-simplex mesh. There are two assumptions typically made when converting a triangular mesh into a corresponding simplex mesh using the above process:

1. The triangular mesh is assumed to be watertight (no holes or gaps),

2. The triangular mesh is assumed to be 2-manifold.

A closed 2-simplex mesh is a watertight 2-manifold mesh with no gaps and/or boundary edges. On the other hand, a multi-material 2-simplex (MM2S) mesh will contain non-manifold edges and/or vertices. This situation is analogous to the multi-material triangular surface meshes discussed in the previous section. The Multi-material and 2-manifold Dual Contouring method described in the previous section were used to generate multi-material triangular surface meshes, and use these to initialize MM2S meshes.

While the duality between 2-simplex meshes and triangular meshes remains true even for multi-material meshes, the above algorithm needs to be adjusted slightly to account for the multi-material nature of the meshes. The multi-material triangular meshes contain material information associated with triangles, and this information can be exploited to produce MM2S meshes in the following manner:

- Step 1: Compute the centroids of each triangle of the triangular mesh.
- Step 2: For each material index
 - Step 2.1: For each i^{th} vertex of the triangular mesh,
 - Step 2.1.1: Locate all the triangles with the current material index that contain the i^{th} vertex
 - Step 2.1.2: Use the centroids of these triangles to create one simplex cell.

Since simplex vertices and cells are being created for each material index, care must be taken to avoid duplicate and overlapping cells along the shared boundaries. Fig. 44 illustrates the conversion process for a multi-material triangular mesh. Fig. 44(a) shows a synthetic box comprising two materials, Fig. 44(b) shows a wireframe rendering of the box. In this figure, the red colored part of the mesh represents one material while the blue colored part represents the

second material. The green colored part of the mesh represents the shared boundary. In Fig. 44(c) the dots represent the centroids of triangles, and Fig. 44(d) shows the duality of triangles to 2-simplex cells. Fig. 44 (e) and (f) show the multi-material 2-simplex and its wireframe representation, respectively.

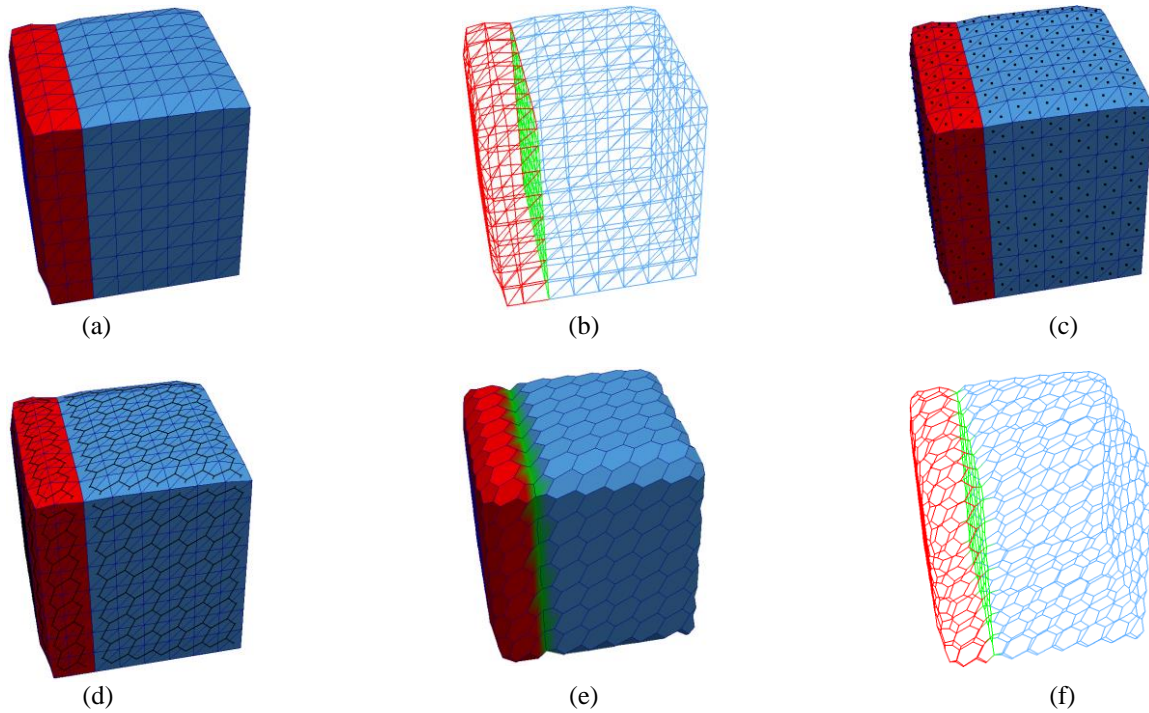


Fig. 44. Conversion of a multi-material triangular mesh of two boxes, colored red and blue and having a shared boundary, shown in green, into a multi-material 2-simplex mesh.

5.3 Description of Multi-Material 2-Simplex Meshes

In the previous section, the multi-material nature of the triangular surface meshes were described by assigning pairwise material indices to triangles. Since the vertices of 2-simplex

meshes are dual to triangles in triangular meshes, it is reasonable to assign the triangles' pairwise material indices to their corresponding dual vertices in the 2-simplex mesh. This procedure ensures the preservation of material information in the conversion process. The number of vertices of the 2-simplex mesh will be the same as the number of triangles in the triangular mesh. Furthermore, as mentioned previously, a MM2S mesh will have shared boundaries as well as non-manifold edges and vertices. Because of its multi-material nature, this type of a k-simplex mesh is not a true 2-simplex mesh in the sense that vertices along the non-manifold edges of the shared boundary can have more than 3 neighboring vertices. Fig. 45 shows such an example. The vertex represented by the yellow dot is a vertex on the non-manifold edge of the shared boundary and is connected to five vertices. The two red colored dots represent two vertices belonging to the red material group, while the blue colored dot represents two vertices belonging to the blue material group. The lone green dot represents a vertex belonging to the shared boundary.

A MM2S mesh can be described as the set $\mathbf{S}^M = \{\mathbf{V}, \mathbf{E}\}$ where \mathbf{V} is the set of n vertices $\{v_i^{p,q}\}, \{i = 0, \dots, n\}, v_i \in \mathbb{R}^3, \{p, q \in \mathbf{M}\}, p \neq q$ where $\mathbf{M} \in \mathbb{N}^+$ is the set of positive integers describing material indices, and p and q are the pairwise material indices assigned to each vertex. \mathbf{E} is the set of m edges $\{\{v_i, v_j\}_m\}, \forall v_i \in \mathbf{V}, \forall v_j \in \mathbf{V}, i \neq j$. In the previous section, multi-material triangular surface meshes (which contain non-manifold edges and vertices) were described as being 2-manifold in the sense that the sub-mesh of each material was purely 2-manifold and watertight. Fig. 46 (left) shows an example of the whole multi-material 2-simplex mesh and Fig. 46 (right) shows the material sub-meshes that are pure 2-simplex. The same analogy can be applied to the case of MM2S meshes: each material sub-mesh of a multi-material simplex mesh is a pure 2-simplex mesh in the sense that all vertices of the sub-mesh have exactly

3 neighboring vertices. This aspect of the MM2S mesh can be effectively utilized when performing deformation of the mesh.

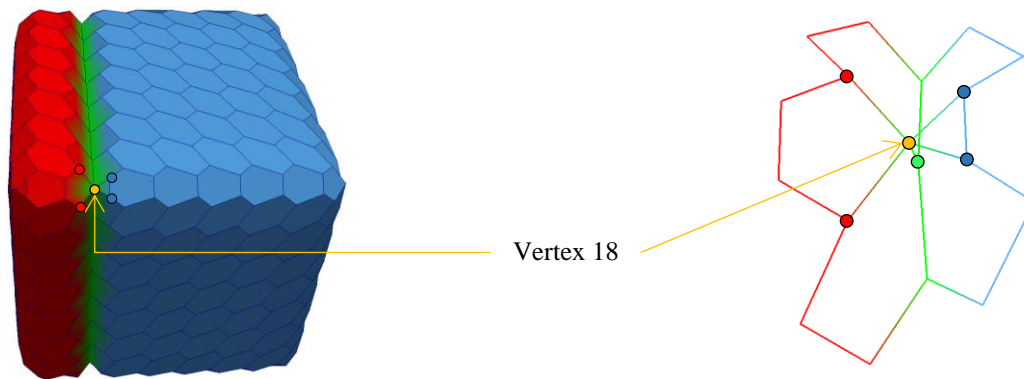


Fig. 45. An example of a multi-material 2-simplex mesh. The highlighted vertex on the non-manifold edge has more than 3 neighboring vertices.

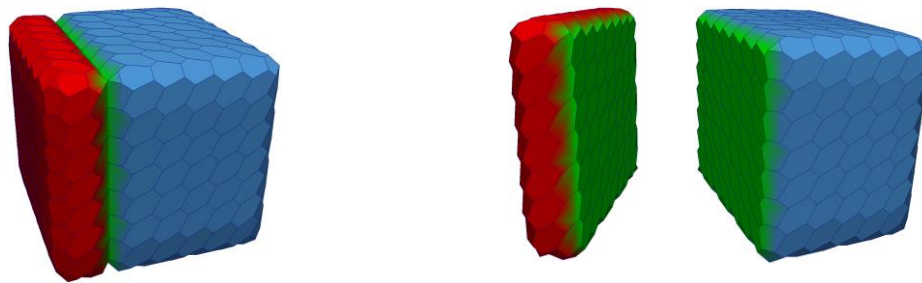


Fig. 46. An example of sub-meshes in a multi-material 2-simplex. (left) The whole multi-material 2-simplex mesh. (right) the constituent pure 2-simplex sub-meshes.

5.4 Overview of Deformable Multi-material 2-Simplex Meshes

Once the initial MM2S mesh is generated from the triangular mesh, it is split into its constituent sub-meshes. Both the MM2S mesh and its sub-meshes are kept in memory. As mentioned above, each sub-mesh is a pure 2-simplex mesh where every vertex is connected to exactly three neighboring vertices.

For every iteration of deformation, internal and external forces are computed. Internal forces are based on mesh geometry, and external forces are based on an input image or volume. The input image or volume is not used directly in the deformation process. Instead, edge-preserving anisotropic diffusion smoothing filter (implemented in VTK's *vtkImageAnisotropicDiffusion3D* filter) is applied, and then the gradient of the volume is computed. This gradient image is used to determine external forces for each vertex. Both internal and external forces are computed independent of each sub-mesh. The forces are then used to separately deform each sub-mesh sequentially using the mesh evolution process described in Chapter 2.

Since forces are computed independently of sub-meshes, the corresponding vertices making up the shared boundary may not necessarily remain consistent after deformation. It is therefore necessary, after each deformation iteration, to ensure that all corresponding vertices of the sub-meshes making up the shared boundary are aligned and consistent. This is done by averaging the positions of each corresponding shared boundary vertex, and then updating the shared boundary vertices in the MM2S mesh as well as the sub-meshes with these newly computed vertex positions. Fig. 47 shows a flowchart of the deformation process.

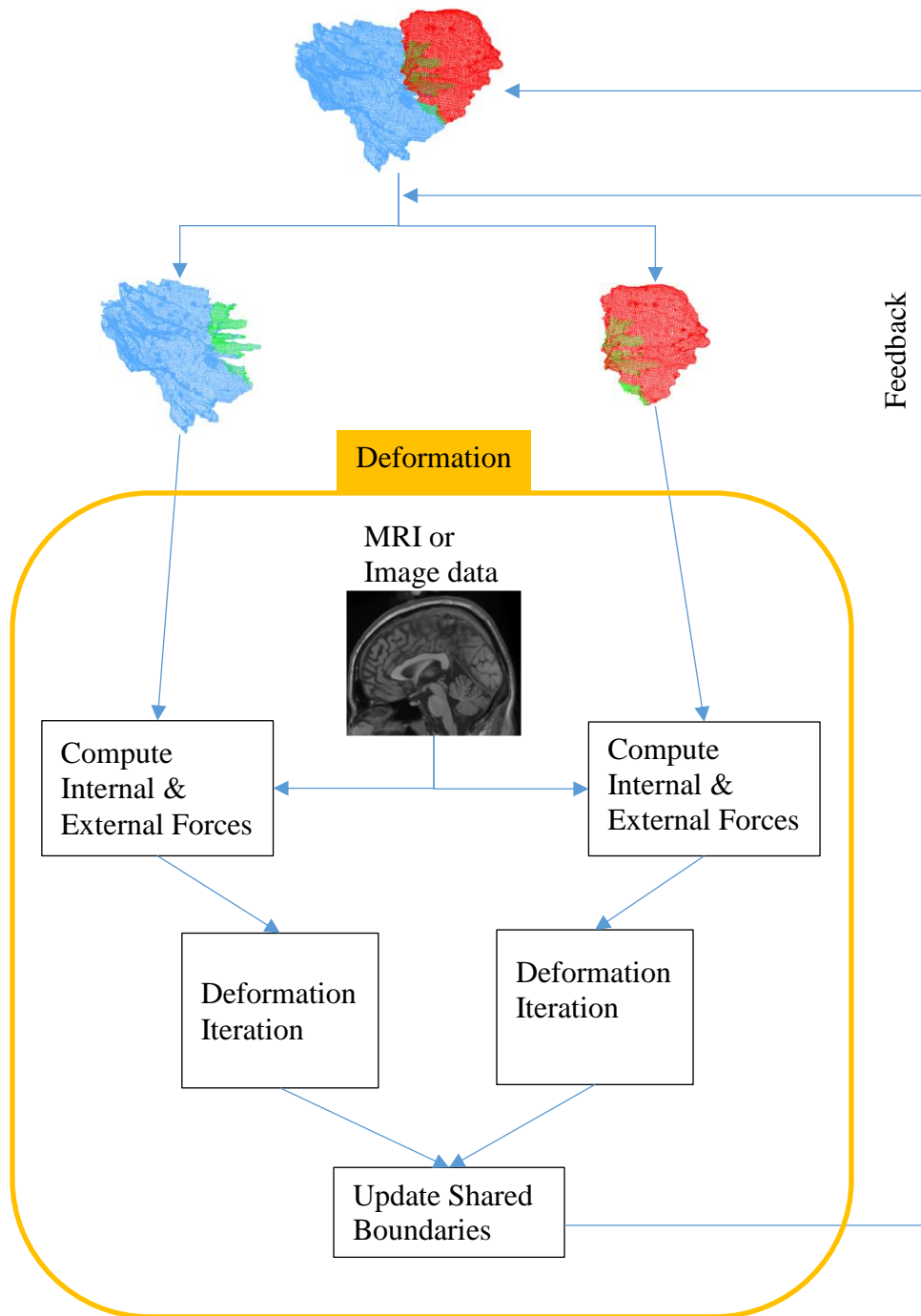


Fig. 47. A flowchart of the multi-material 2-simplex deformation process.

This process for deforming a multi-material 2-simplex mesh offers two advantages: (1) the proven single surface 2-simplex mesh deformation framework of [31, 32] can be easily utilized, (2) the shared boundary between the sub-meshes will always remain consistent, and (3) there is no need to worry about the non-manifold edges of the shared boundaries since the deformation occurs only on sub-meshes. The only disadvantage to this process is that it may be slightly time consuming, especially for large meshes, due to the need to separately update the vertices of the shared boundaries for every deformation iteration.

5.5 Multi-material Deformation Using a Synthetic Example

As an example, consider a synthetic multi-material box, consisting of two materials which will be deformed using MM2S deformation procedure described above. The input mesh is a multi-material triangular surface mesh, which is converted into a multi-material 2-simplex mesh, as seen in Fig. 48 (top row).

The input volume is a simple cube where half of the voxels are of one value, and the remaining voxels are of a different value. This volume is anisotropically smoothed [133] to perverse edges and remove noise, and the gradient image is computed. The MM2S mesh undergoes an affine or rigid transformation such that it lies within the gradient image, as shown in Fig. 48 (bottom row).

Fig. 49 (top row) shows a slice of the gradient image, as well as the slice of the initial MM2S mesh, along a specific axis. Fig. 49 (bottom row) shows a series of slices of the MM2S mesh undergoing deformation with respect to the gradient image. As can be seen, the initial MM2S mesh is contained by the boundary coinciding with the image gradient. Subsequent

deformations enlarge the MM2S mesh until it coincides with a location of strong image gradient magnitude at the 200th iteration.

A useful method for tracking the state of the deformation process is by computing the sum of absolute vertex displacement for all vertices of the current iteration's mesh, with respect to the previous iteration. While this metric can be a potential indicator as to state of the deformation, it is not completely reliable as a terminating condition. Fig. 50 shows the graph of absolute vertex displacement for the deformation of the synthetic multi-material box. It can be seen that after approximately 50 iterations, the overall displacement per iteration of the mesh very small. Fig. 49 shows that the deformation is much better after 200 deformation iterations, instead of 60 iterations.

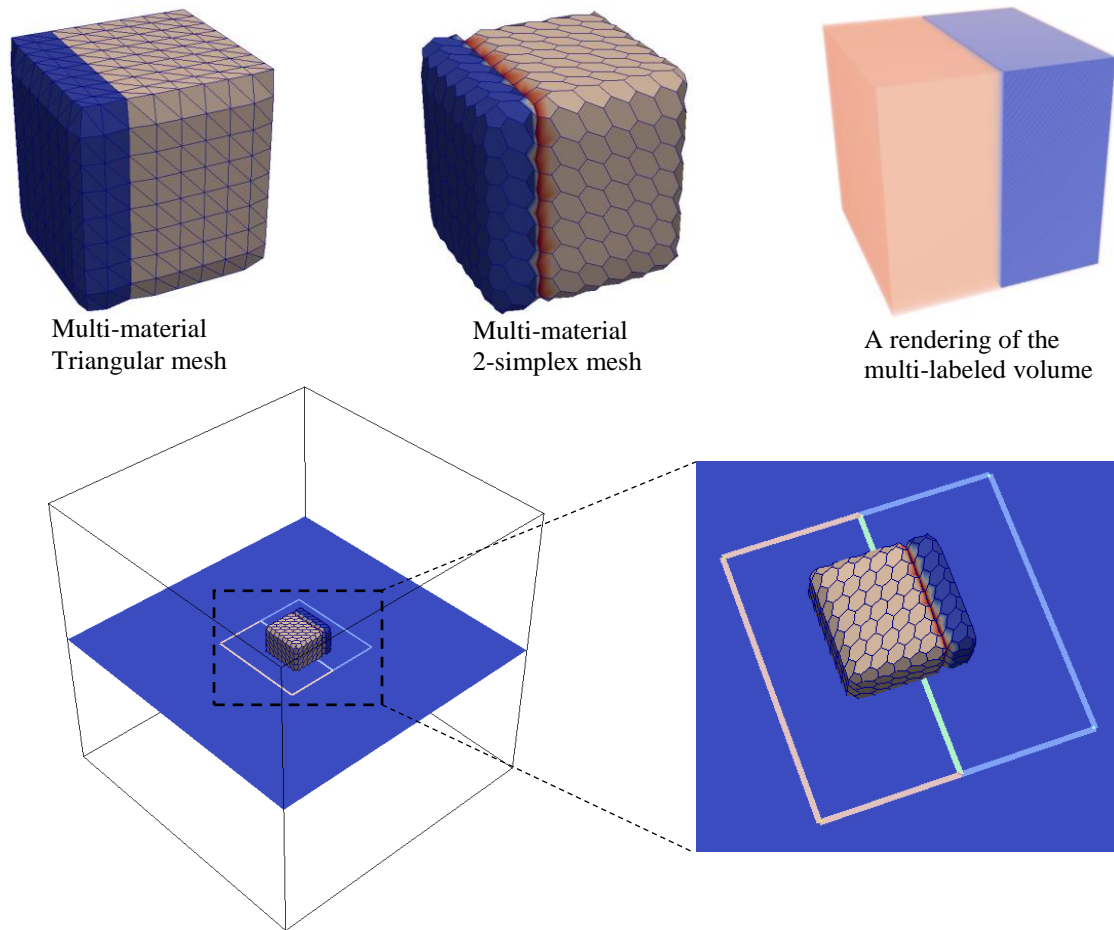


Fig. 48. Initialization of the multi-material 2-simplex deformation.

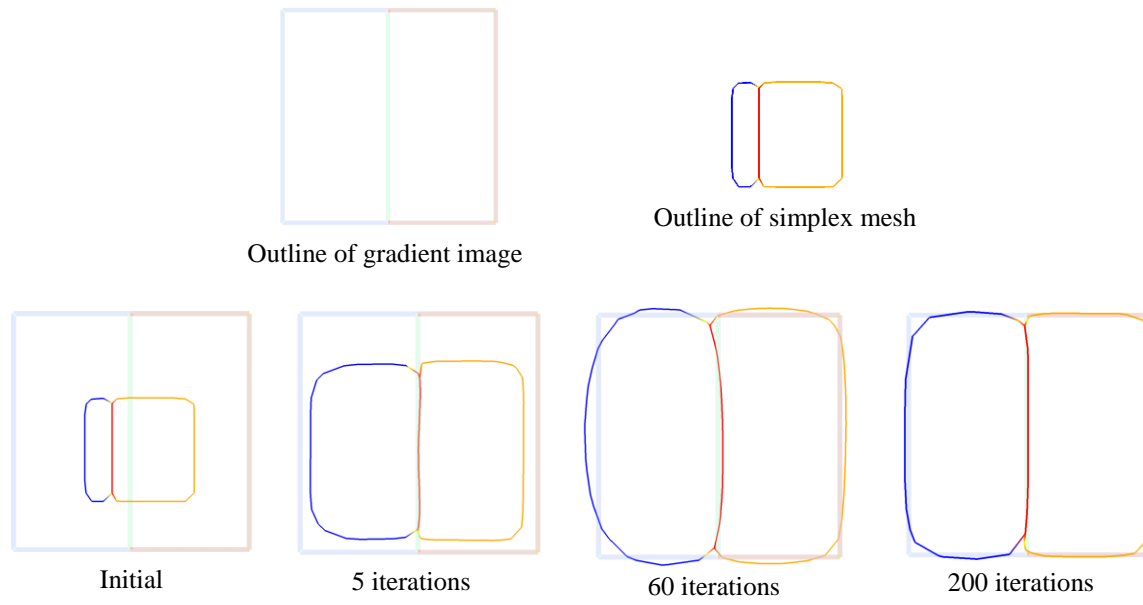


Fig. 49. State of the mesh deformation of the synthetic box example with respect to the image gradient.

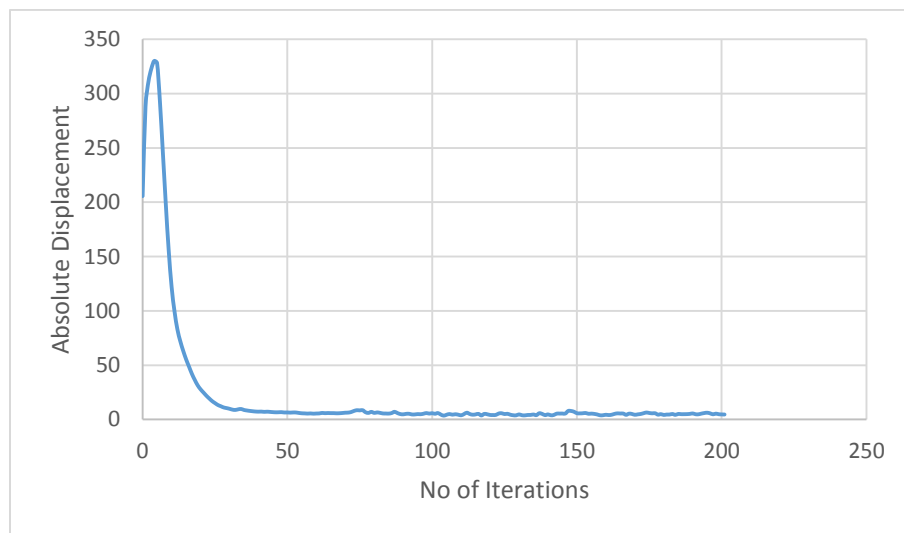


Fig. 50. Graph of absolute displacement for the deformation of the synthetic box example.

5.6 Segmentation of the Subthalamic Nucleus and Substantia Nigra

The proposed multi-material 2-simplex deformable system has been also used on realistic data to achieve meaningful segmentation of anatomical structures. The subthalamic nucleus (STN) and the substantia nigra (SN) are two deep brain structures that are difficult to detect and segment from MRI. The proposed system has also been used to segment the globus pallidus (GP), the image gradient of which is better defined. T1-weighted and T2-weighted MR data was used. O’Gorman [134] studies the visibility of the STN and GP internal segment using eight different MR protocols, and concludes that different iron levels present in the basal ganglia structures can affect the contrast-to-noise ratio.

The MR data used in this section are freely available from *Neuroimaging Informatics Tools and Resources Clearinghouse* (<https://www.nitrc.org/projects/deepbrain7t>), and was produced as a part of the research in [135]. In [135], T1 and T2-weighted MR images of 12 healthy control subjects were acquired using a 7T MR scanner, and an unbiased average template with T1w and T2w contrast was generated using groupwise registration. Both images have a dimensions of 267x367x260 voxels, and 0.6 mm isotropic spacing. A labeled volume (shown in Fig. 51) is also provided (referred to as the Wang atlas), containing segmentations of the left and right globus pallidus, mammillary body, red nucleus, substantia nigra and subthalamic nucleus. This labeled volume, recently made public, serves as ground truth for validating our multi-surface atlas-to-image registration approach.

An initial watertight and 2-manifold multi-material triangular mesh of the left SN-STN was constructed from Chakravarty’s atlas [54] using the multi-material Dual Contouring algorithm described in Chapter 4. This atlas has a step size of 0.3 mm, and a triangular mesh generated from this resolution is simply too large (approximately 410k triangles and 20k

vertices) to be practical. Therefore, the atlas was downsampled to an appropriate size, and a much coarser multi-material triangular mesh was generated (2.5k triangles and 1.2k vertices). Fig. 52(a) shows a mesh representation of the SN and STN from the Wang atlas, Fig. 52(b) and (c) shows the surface mesh and wireframe mesh, respectively, of the SN and STN constructed using Chakravarty's atlas.

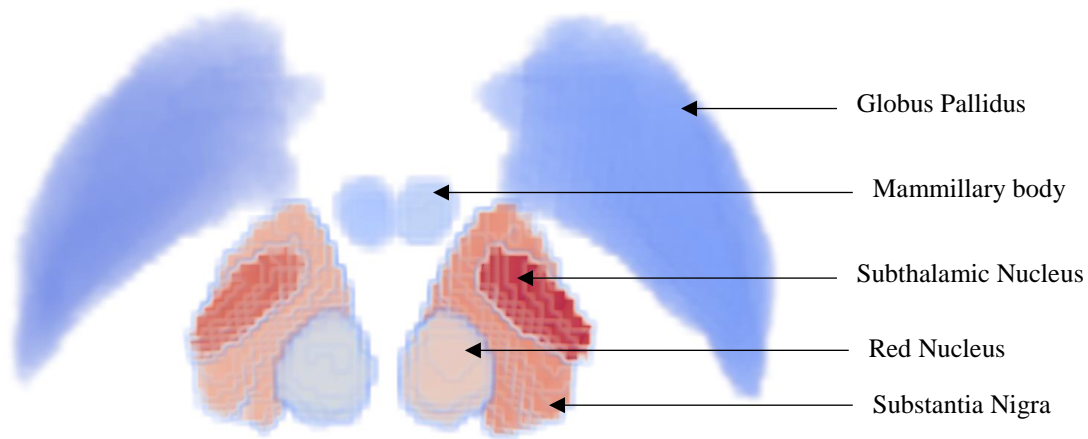


Fig. 51. A rendering of the labels in the Wang atlas.

For the deformation process, the T2-weighted MR image was used because the SN and STN are more visible than in T1-weighted MR images [136], as shown in Fig. 53. The image was anisotropically, and then the gradient image was computed. The external forces for the deformation were computed using the gradient image. Laplacian-based internal forces were used to achieve a smooth mesh. Fig. 54 shows the deformation of the SN and STN mesh for several iterations. Fig. 55 shows a graph of the absolute mesh displacement. As can be seen, the

deformation reaches a steady-state after 150 iterations. Fig. 56 shows the final deformed mesh of the SN and STN, at 160 iterations.

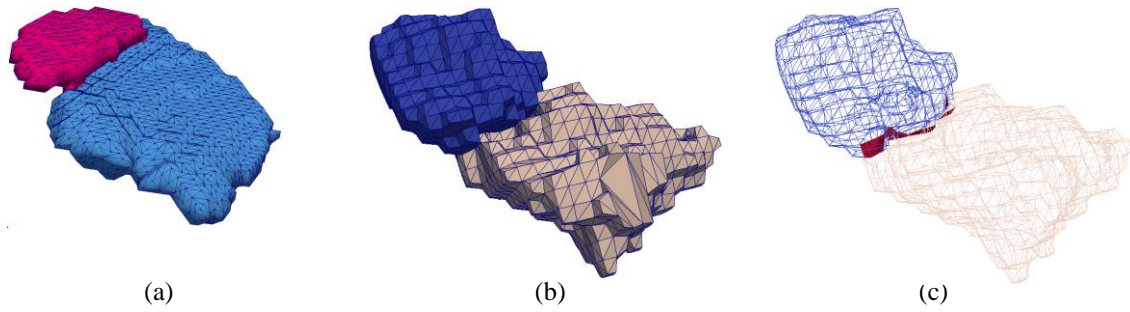


Fig. 52. Meshes of the SN and STN. (a) A mesh representation of the SN (blue) and STN (pink) from Wang atlas, (b) the multi-material triangular surface mesh of the SN (yellow) and STN (blue) after clipping, (c) the wireframe representation of the mesh, where the red part represents the shared boundary.

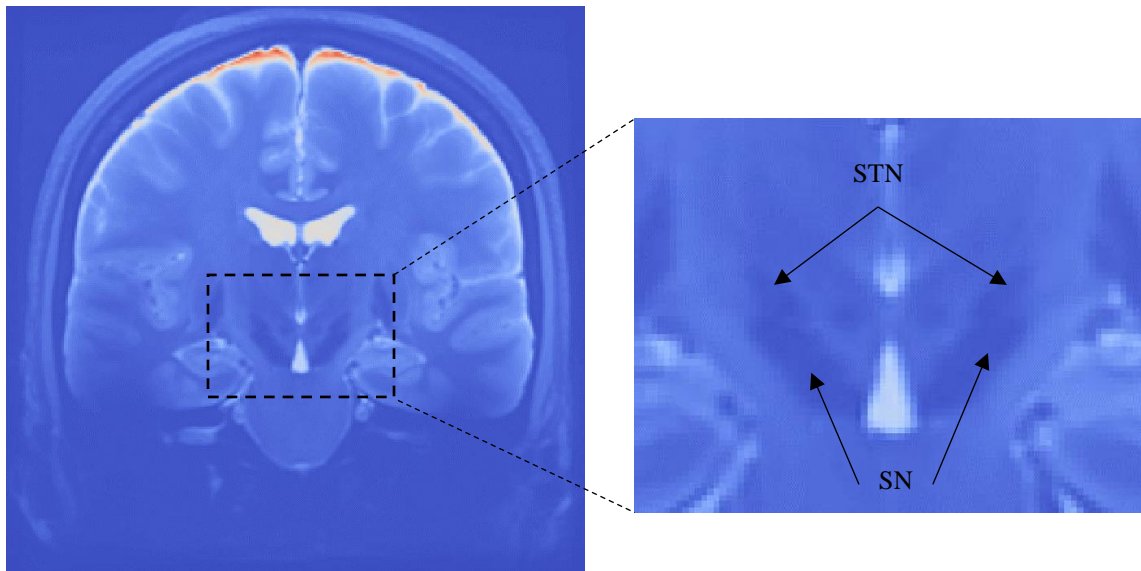


Fig. 53. Coronal slice of the T2-weighted 7T MRI. (Inset) The subthalamic nucleus and substantia nigra.

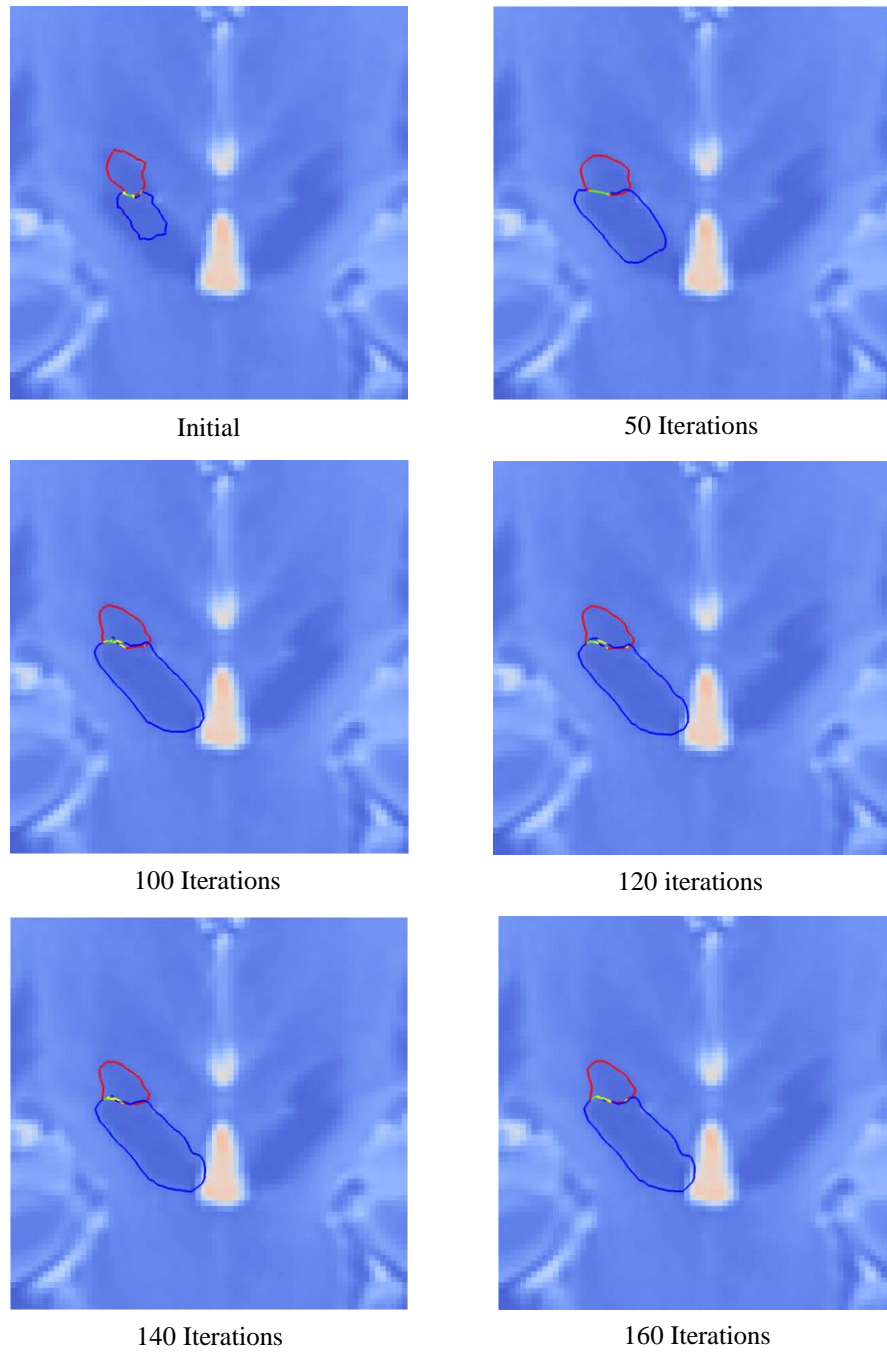


Fig. 54. State of the deformation of the SN and STN. The red outline represents the outline of the STN and the blue outline represents the outline of the SN. The green outline represents the shared boundary.

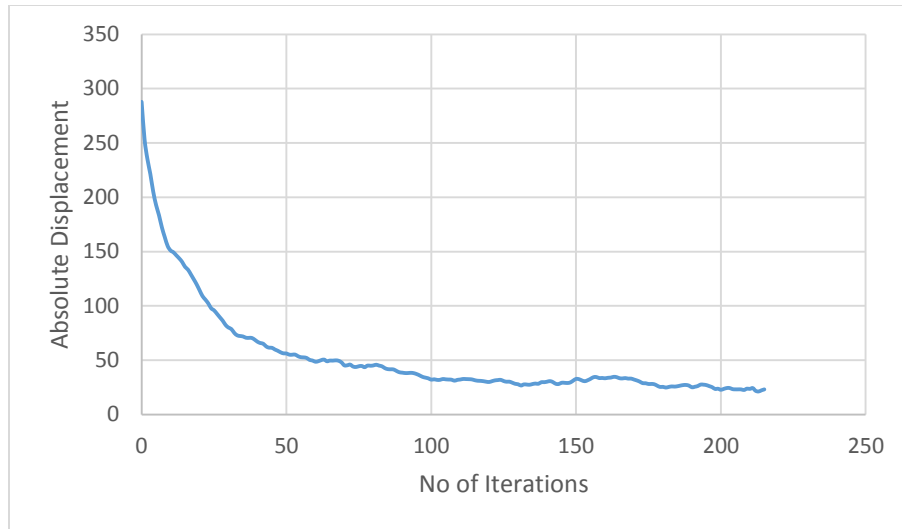


Fig. 55. Graph of absolute displacement for the deformation of the SN and STN.

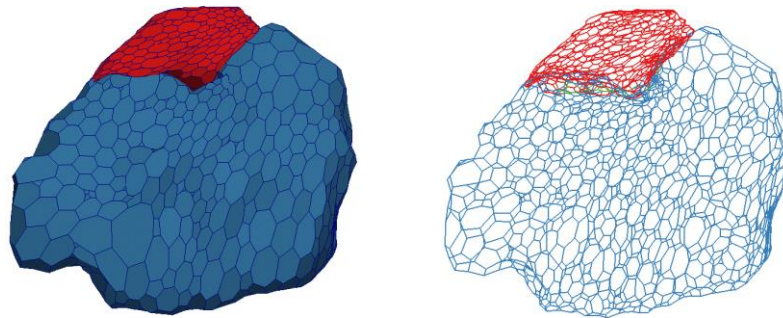


Fig. 56. Final surface mesh (left) and wireframe (right) of the SN (blue) and STN (red).

5.7 Segmentation of the Globus Pallidus and Striatum

In this section, a series of three experiments will be attempted to segment the striatum (comprising the putamen and the caudate nucleus) and the globus pallidus (GP) using high resolution 7 Tesla MR data from [135]. In a T1-weighted MR image, both the striatum and GP have similar intensities, whereas in a T2-weighted MR image the GP appears much more

distinct, as shown in Fig. 57. In the three experiments an initial multi-material 2-simplex mesh representation of the striatum and GP will be deformed using:

- 1) only T1-weighted MRI
- 2) only T2-weighted MRI, and
- 3) both T1 and T2-weighted images. Here, the T1-weighted MR image will be used to mainly drive the deformation of the striatum, and the T2-weighted MR image will be used to segment the GP. The shared boundary between the striatum and GP will therefore be influenced by both the T1 and T2-weighted images.

The main purpose of the third experiment is to demonstrate that the proposed multi-material 2-simplex framework can incorporate multi-modal data for deformation.

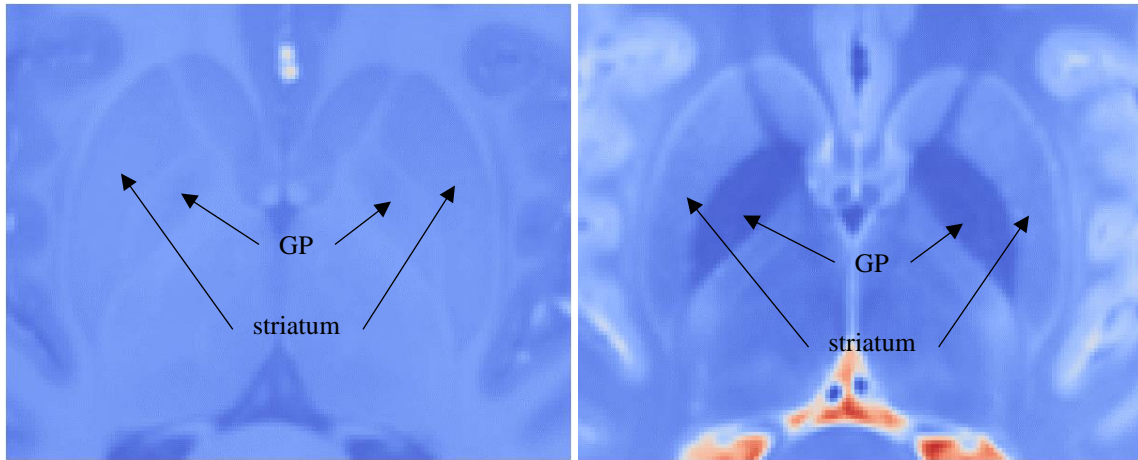


Fig. 57. The differing contrasts of the globus pallidus and striatum in T1-weighted MRI (left) and T2-weighted MRI (right).

5.8.1 Initializing the Multi-material 2-Simplex Mesh

In the Wang atlas, the GP is represented as one single structure, as shown in Fig. 58 (top) whereas in Chakravarty's atlas the GP is represented as three distinct parts: Globus Pallidus (label 5), Globus Pallidus Internal (label 11) and Globus Pallidus External (label 12), as shown in Fig. 58 (bottom). For this experiment, the three parts of the GP in Chakravarty's atlas were combined into a single structure. A new volume was created consisting of only the combined GP as well as the striatum.

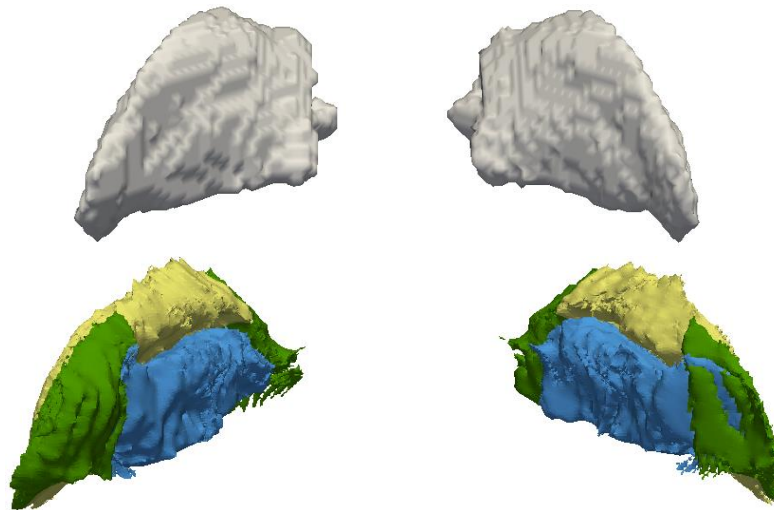


Fig. 58. The surface mesh representation of the GP. (Top) Wang atlas, (bottom) the three parts of the GP in Chakravarty's atlas. (Green) Globus Pallidus, (Yellow) Globus Pallidus External, (Blue) Globus Pallidus Internal.

In order to produce a coarse mesh, the volume was downsampled to an appropriate size and smoothed using the Gaussian blurring-based smoothing process described in Chapter 2. The multi-material DC method was then used to create a watertight multi-material triangular surface

mesh, which in turn was used to create the initial multi-material 2-simplex mesh. In Fig. 59, the upper row illustrates the multi-material triangular mesh created by the DC algorithm, and the bottom shows the converted multi-material 2-simplex mesh models of the striatum and GP. This MM2S mesh was used for all three deformations.

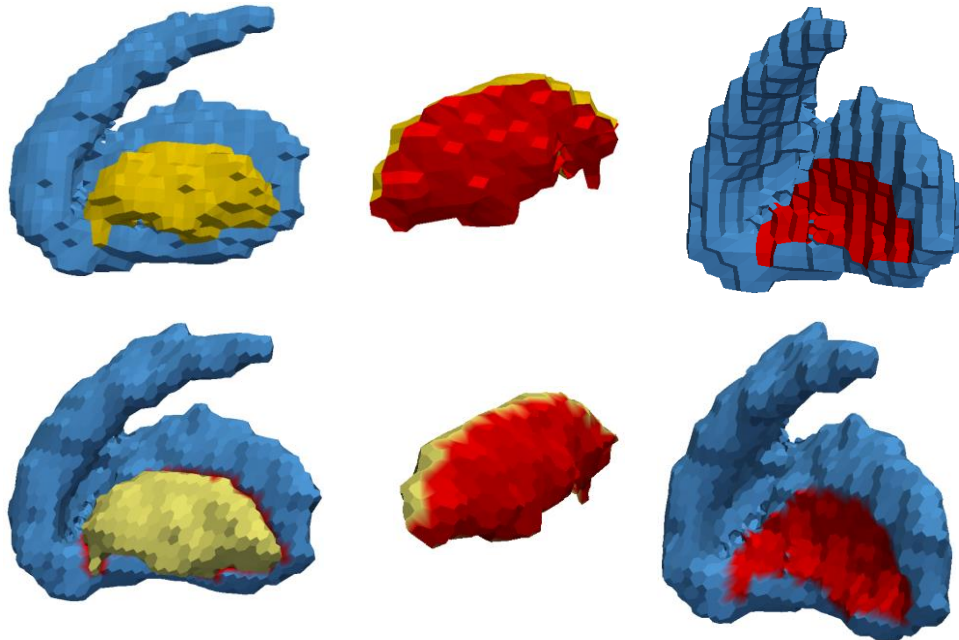


Fig. 59. Simplex mesh generation for the Striatum and GP. (Top) The multi-material triangular mesh of the Striatum and combined Globus Pallidus. (Bottom) The multi-material 2-simplex mesh initialized from the triangular mesh. The red part of the mesh depicts the shared boundary between the GP and St.

For the deformation, the MR image was anisotropically smoothed, and then the gradient image was computed. The external forces for the deformation were computed using the gradient image. For the first experiment, only the T1-weighted MR image was used. For the second

experiment, only the T2-weighted MR image was used. For the third experiment, both gradient images for both T1 and T2-weighted images were computed. Laplacian-based internal forces were used to achieve a smooth mesh.

5.8.2 Segmentation Using Only T1-weighted MRI

Fig. 60 shows the state of the deformation using only T1-weighted MRI over several iterations. In this figure, the blue outline represents the outline of the striatum, the yellow outline represents the outline of the GP, and the red outline shows the shared boundary between the GP and striatum. As shown in Fig. 61, the deformation achieves a steady-state after about 250 iterations. Fig. 62 shows the meshes of the striatum and GP after 250 deformation iterations.

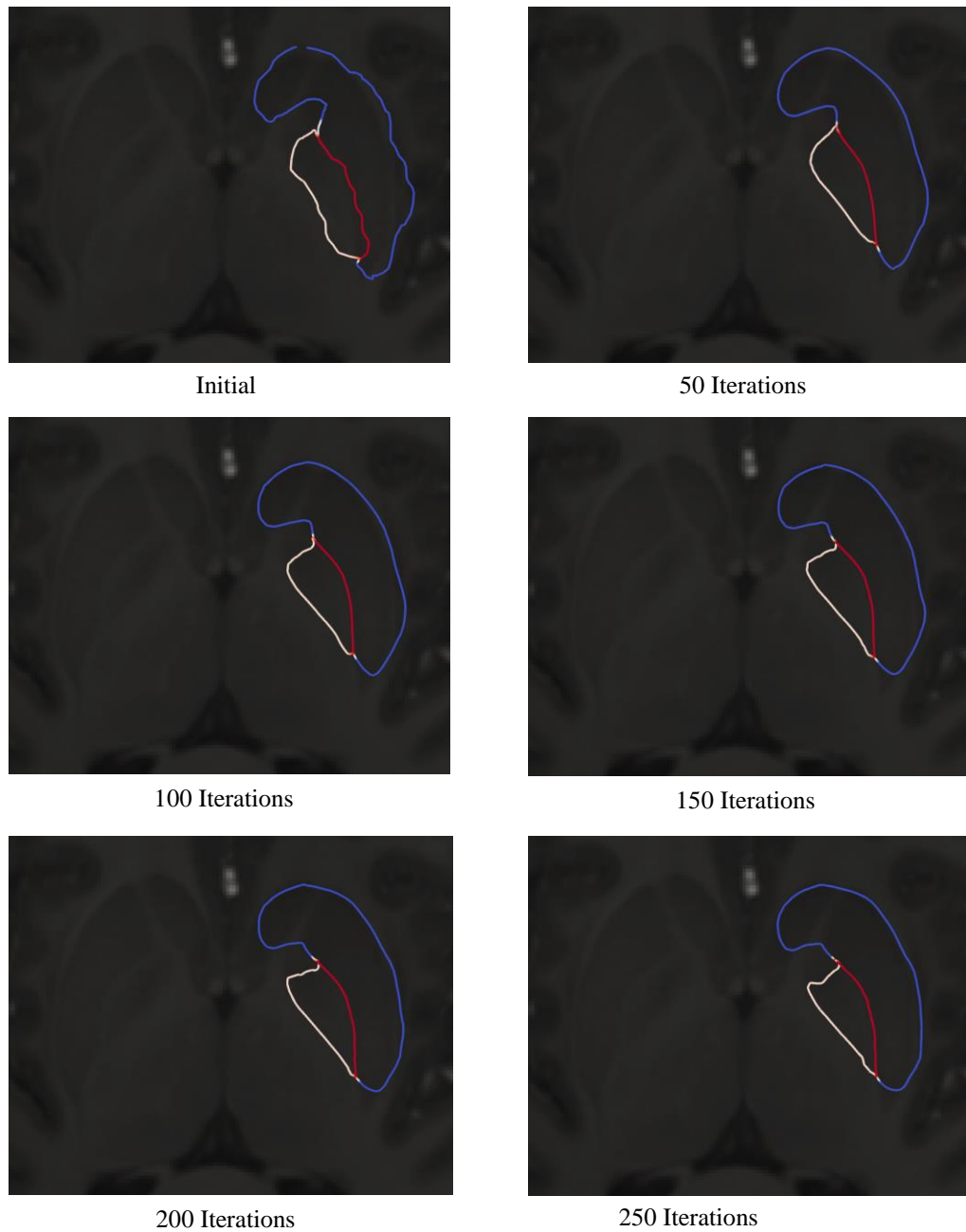


Fig. 60. Cross-sections of the striatum and globus pallidus during deformation using T1-weighted MRI. The blue outline represents the striatum, the yellow outline represents the GP, and the red outline represents the shared boundary.



Fig. 61. Graph of absolute displacement for the deformation of the GP and striatum using only T1-weighted MRI.

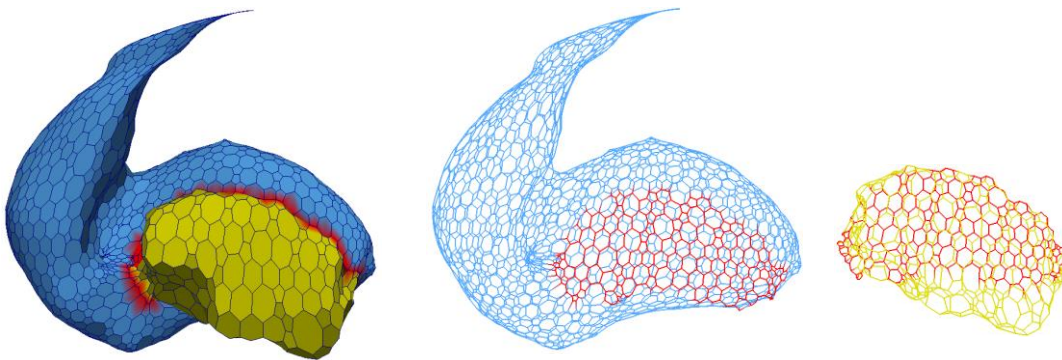


Fig. 62. The final surface mesh (left) of the striatum and GP after deformation using only T1-weighted MRI. (Middle and right) A wireframe rendering of the striatum and GP.

5.8.3 Segmentation Using Only T2-weighted MRI

Fig. 63 shows the state of the deformation using only T2-weighted MRI over several iterations. In this figure, the blue outline represents the outline of the striatum, the yellow outline represents the outline of the GP, and the red outline shows the shared boundary between the GP and striatum. As shown in Fig. 64 the deformation achieves a steady-state after about 350 iterations. Fig. 65 shows the meshes of the striatum and GP after 350 deformation iterations.

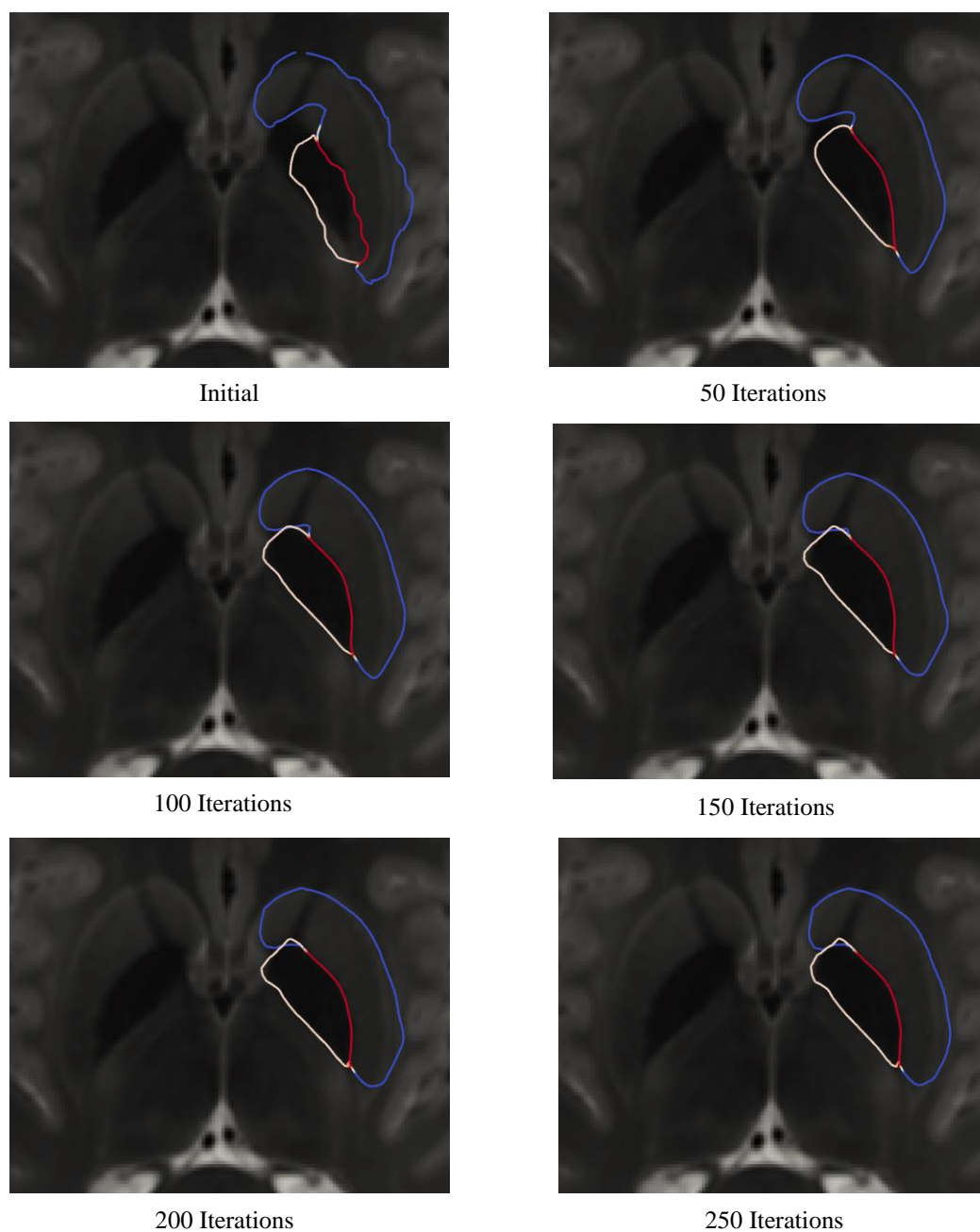


Fig. 63. Cross-sections of the striatum and globus pallidus during deformation using T2-weighted MRI. The blue outline represents the striatum, the yellow outline represents the GP, and the red outline represents the shared boundary.

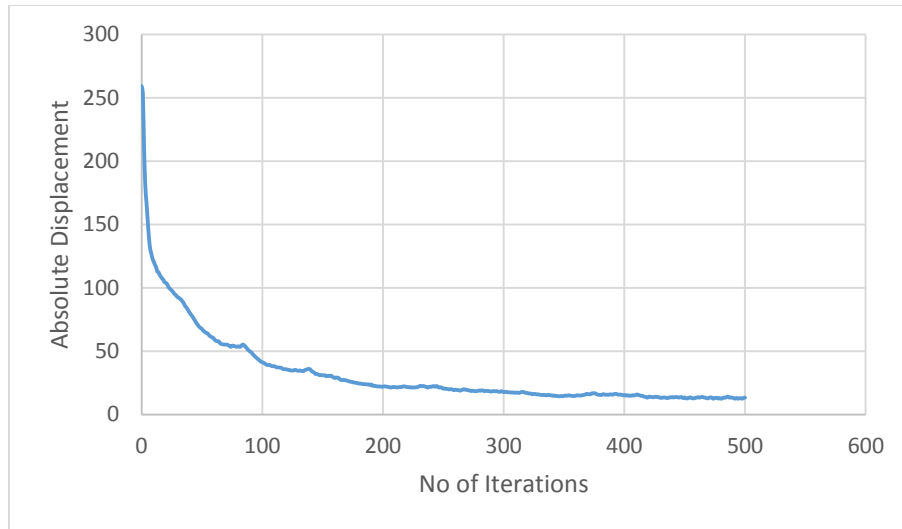


Fig. 64. Graph of absolute displacement for the deformation of the GP and striatum using only T2-weighted MRI.

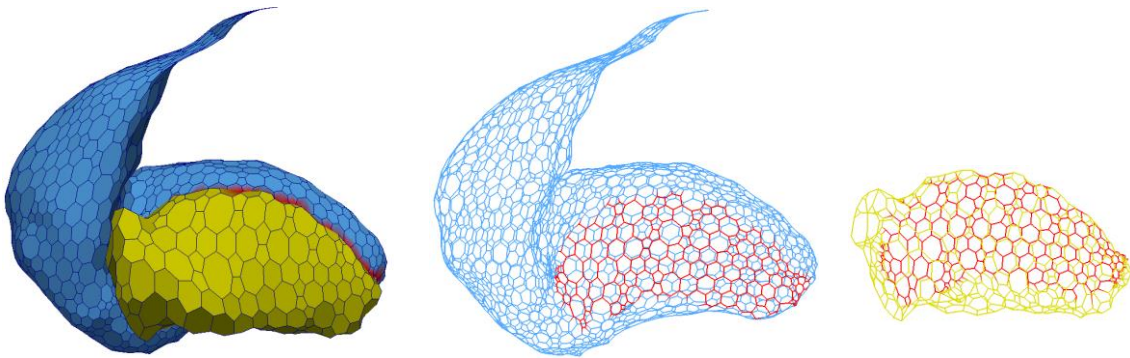


Fig. 65. The final surface mesh (left) of the striatum and GP after deformation using only T2-weighted MRI. (Middle and right) A wireframe rendering of the striatum and GP.

5.8.4 Segmentation Using Both T1 and T2-weighted MRI

Fig. 66 shows the state of the deformation using both T1 and T2-weighted MRI over several iterations. In this figure, the blue outline represents the outline of the striatum, the yellow outline represents the outline of the GP, and the red outline shows the shared boundary between the GP and striatum. As shown in Fig. 67 the deformation achieves a steady-state after about 350 iterations. Fig. 68 shows the meshes of the striatum and GP after 350 deformation iterations.

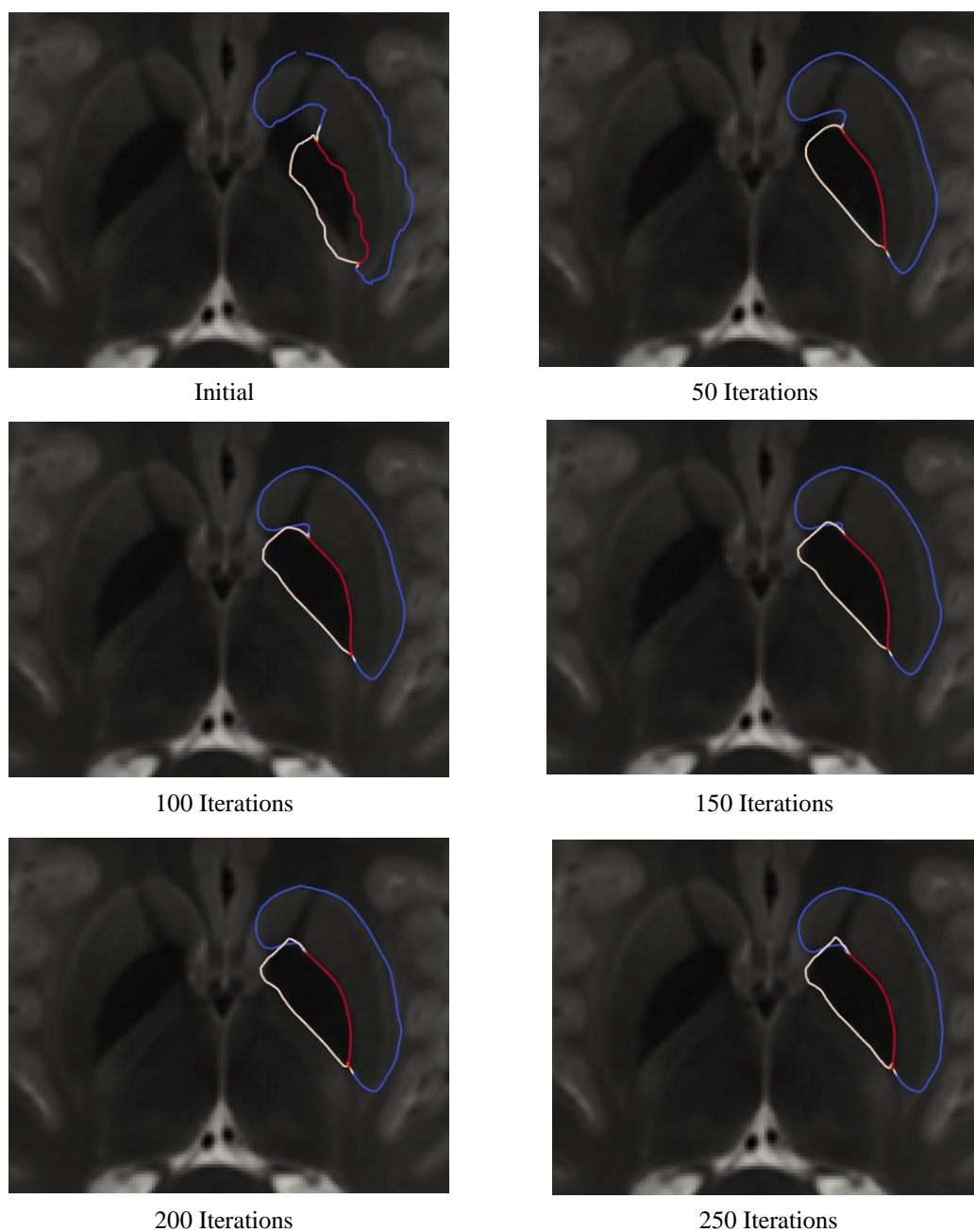


Fig. 66. Cross-sections of the striatum and globus pallidus during deformation using both T1 and T2-weighted MRI. The blue outline represents the striatum, the yellow outline represents the GP, and the red outline represents the shared boundary.

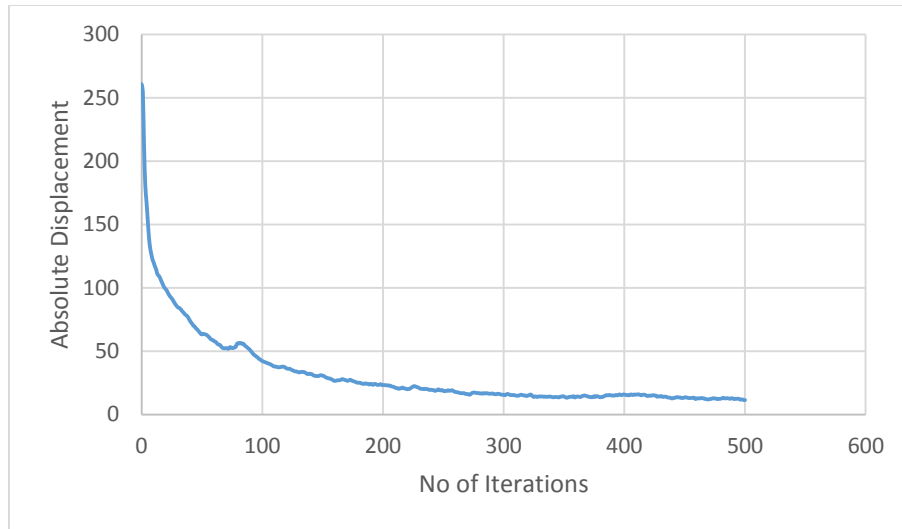


Fig. 67. Graph of absolute displacement for the deformation of the GP and striatum using both T1 and T2-weighted MRI.

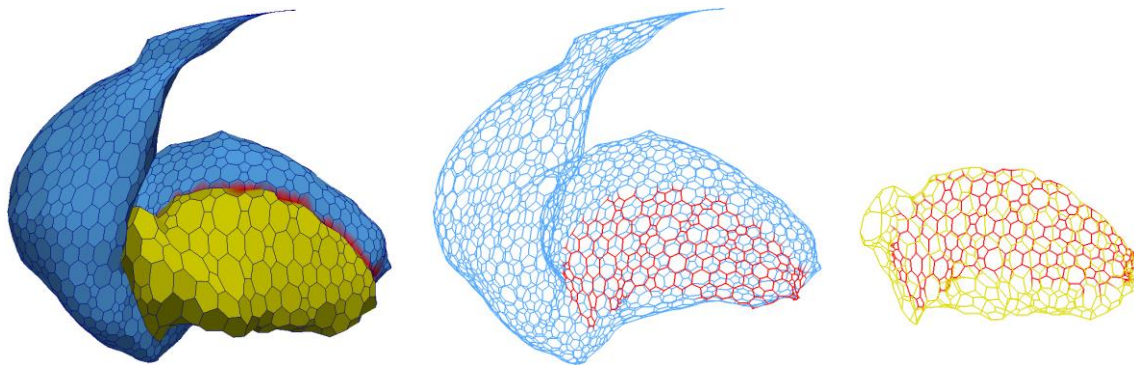


Fig. 68. The final surface mesh (left) of the striatum and GP after deformation using both T1 and T2-weighted MRI. (Middle and right) A wireframe rendering of the striatum and GP.

5.8 Validation

For validation, surface mesh representations of the SN, STN and GP were made from the labeled volume that is a part of the data from [135], and this was used as the ground truth. The opensource VTK libraries implementation of the Marching Cubes algorithm was used to generate the surface meshes. Unfortunately a segmentation of the striatum is not available in the data from [135], and so a quantitative analysis of the striatum is not possible.

The surface-to-surface distance between the deformed mesh and the ground truth mesh of the SN, STN and GP was computed using uniform sampling. The metrics calculated are: Hausdorff Distance (HD), Mean Absolute Distance (MAD), Mean Square Distance (MSD) and Dice's Coefficient (DC), reported in TABLE 9. The Hausdorff distance is the largest error between the deformed mesh and its corresponding ground truth mesh. The Dice's Coefficient measures the amount of similarity between the deformed mesh and the ground truth mesh. The Mean Square Distance reports the average squared difference between the sampled points on the deformed mesh and the ground truth mesh, and the Mean Absolute Distance reports the mean of the absolute difference between the sampled points of the deformed mesh and the ground truth mesh.

Fig. 69 shows the deformed meshes of the SN and STN. The color scheme is as follows: for the deformed meshes, red indicates over-segmentation, and blue represents under-segmentation, and green indicates correct segmentation, with respect to the groundtruth. For the deformation of the SN and STN, the maximum over-segmentation, as shown in Fig. 69, the highest over-segmentation error is approximately 1.55 mm and 1.97 mm for the STN and SN, respectively. The highest under-segmentation error is -0.898 mm and -1.63 mm for the STN and SN, respectively. TABLE 9 shows the HD, MSD, MAD and DC values for the deformed STN

and SN. In both cases, the HD value is smaller compared to the HD of the GP because both the SN and STN are smaller structures, relative to the GP. The MSD and MAD values of the SN and STN are smaller because, along the main body of both structures, the segmentation is fairly accurate (as shown by the green coloring in Fig. 69), with over and/or under-segmentation occurring at the lateral ends of both structures, coinciding with lesser gradient values. The DC values show that there is an approximately 77% and 80% similarity, for the STN and SN respectively, with their corresponding ground truth meshes. Fig. 70 and Fig. 71 shows the distribution of segmentation errors of the STN and SN, respectively, with respect to sample points.

The HD for the STN and SN are 2.10 and 2.5 mm, respectively. The DC value for the STN and SN are 77% and 79%, respectively. The HD, DC, MSD and MAD values for all three experiments of the GP are reported in TABLE 9.

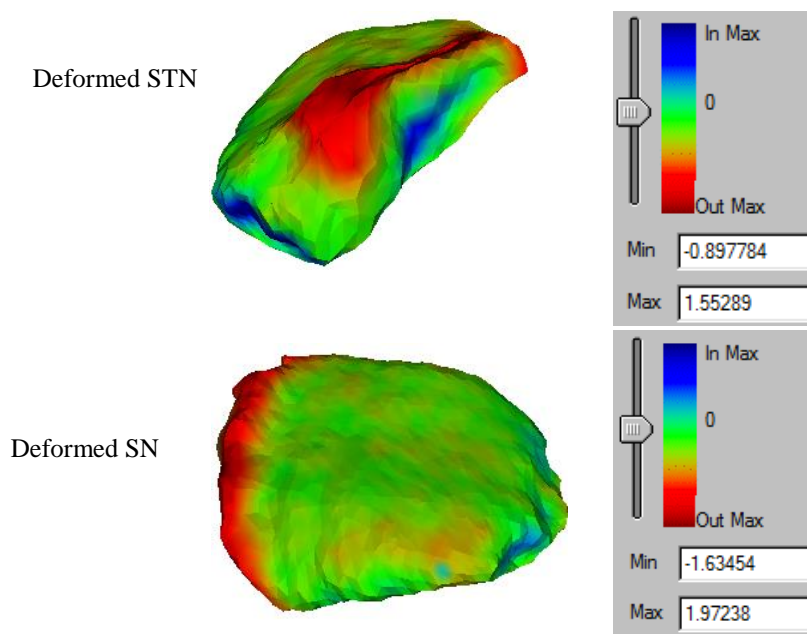


Fig. 69. (Top row) mesh of the STN, (bottom row) mesh of the SN.

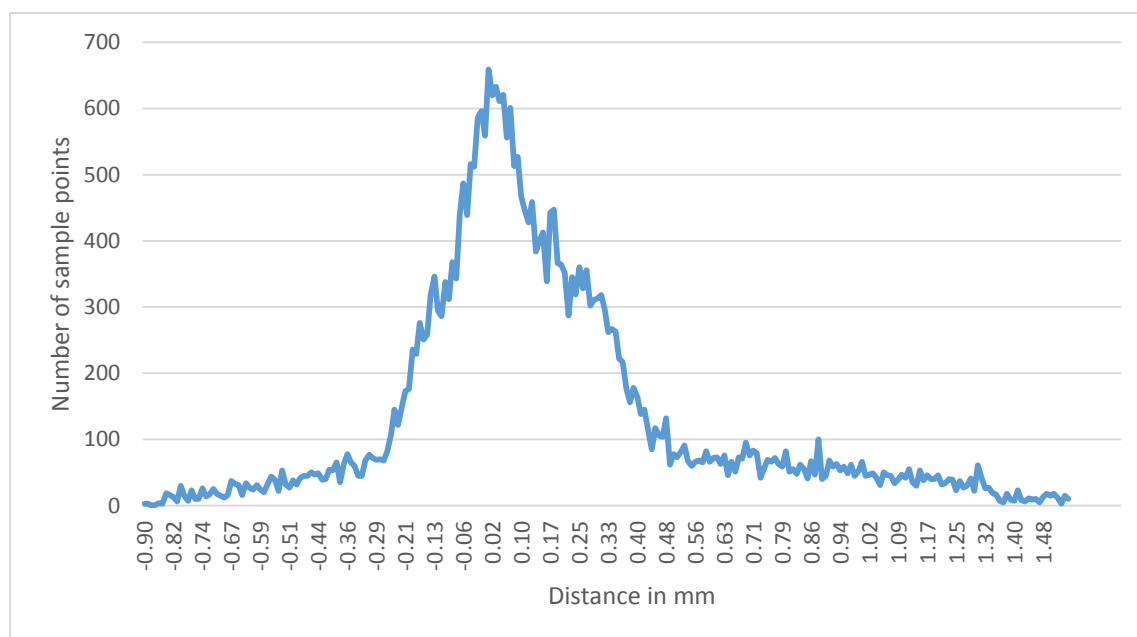


Fig. 70. Histogram of segmentation errors for the STN.

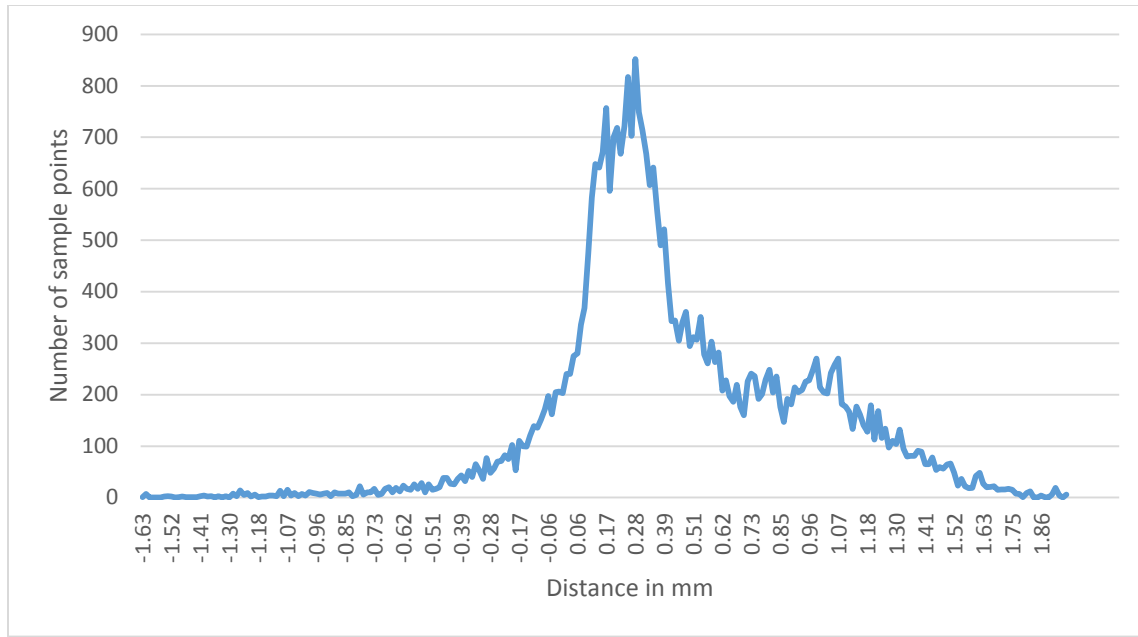


Fig. 71. Histogram of segmentation errors for the SN.

Fig. 72 shows the deformed meshes of the GP from the three experiments. The same color scheme applies: red indicates over-segmentation, blue indicates under-segmentation and green indicates correct segmentation, with respect to the ground truth. Amongst these three results, the mesh obtained from using only T1-weighted MRI shows the worst result with the maximum under-segmentation being approximately -6.23 mm and over-segmentation being approximately 7.1 mm. The deformed meshes from using only T2-weighted MRI and both T1 and T2-weighted MRI show similar segmentation results, with the maximum over-segmentation value being approximately 2.4 mm and the under-segmentation value being -2.1 mm.

For the first experiment using only T1-weighted MRI, the HD is significantly larger than any other entries in the table. This is because the lower left side of the GP, as shown in Fig. 72 (left), is under-segmented, and over-segmented on the lower right side. As mentioned previously,

the GP boundary is not highly visible in T1-weighted MRI, which can account for the segmentation errors. Correspondingly, the MSD and MAD values are the largest for this deformation. The DC value shows only an approximately 70% similarity with the ground truth. Fig. 73 shows the distribution of the segmentation errors for the deformed mesh using only T1-weighted MRI.

In contrast, the DC value is significantly better at 92% similarity for the deformations of the GP using only T2-weighted MRI and combined T1 and T2-weighted MRI. This improvement is the result of the GP's better visibility on the T2-weighted MRI. The HD, MSD and MAD values for these two deformations are similar and also better than the values of the deformation with only T1-weighted MRI. Again, the HD is larger, compared to the SN and STN because the GP is a relatively larger structure. Fig. 74 and Fig. 75 shows the distribution of the segmentation errors for the deformed meshes using only T2 weighted MRI, and both multimodal T1 and T2-weighted MRI, respectively.

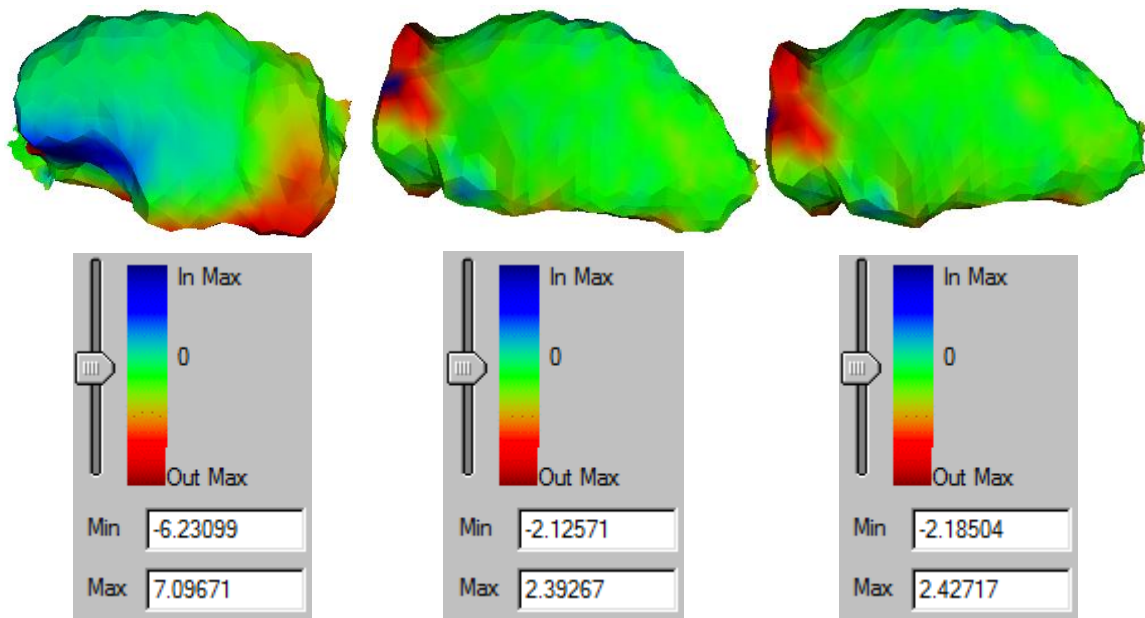


Fig. 72. Final deformation of the GP using: (left) only T1-weighted MRI, (Middle) only T2-weighted MRI, and (right) T1 and T2-weighted MRI.

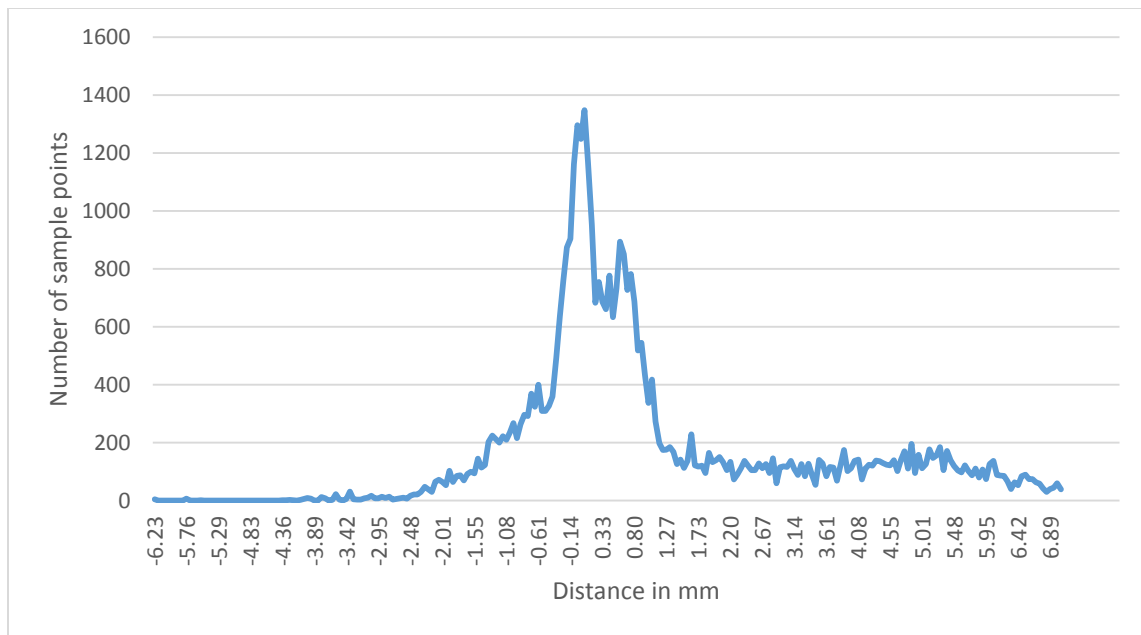


Fig. 73. Histogram of segmentation errors for the GP using only T1-weighted MRI.

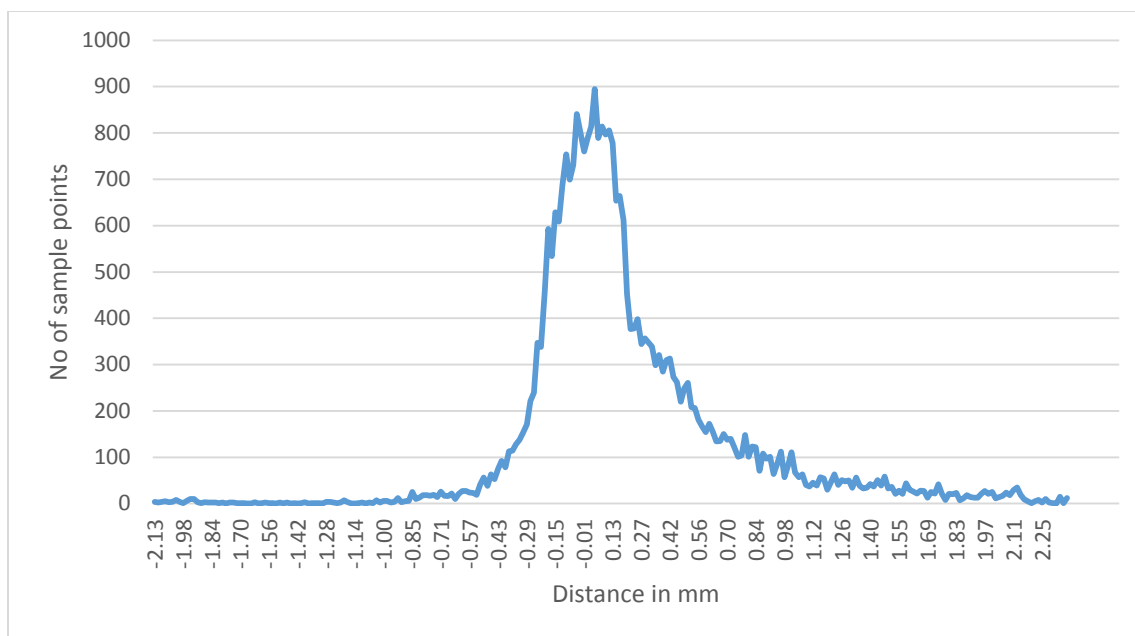


Fig. 74. Histogram of segmentation errors for the GP using only T2-weighted MRI.

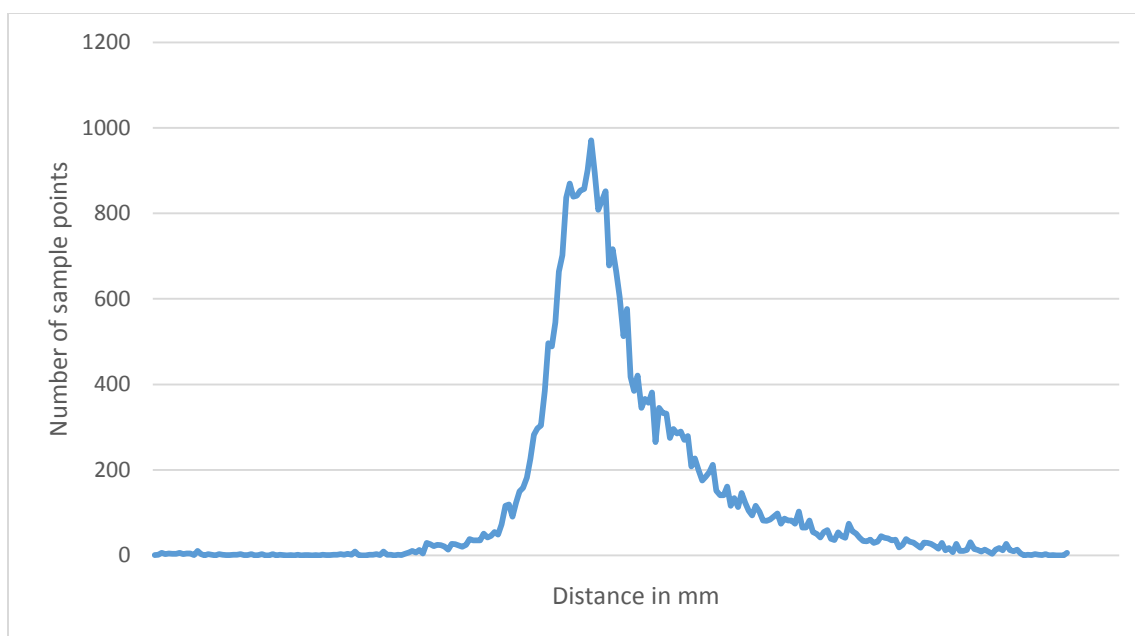


Fig. 75. Histogram of segmentation errors for the GP using both T1 and T2-weighted MRI.

TABLE 9
SUMMARY OF DEFORMATION ERRORS

Anatomical Structures	Hausdorff Distance	Mean Square Distance (MSD)	Mean Absolute Distance (MAD)	Dice's Coefficient
Subthalamic nucleus	2.10539	0.30867	0.35002	0.773219
Substantia nigra	2.54666	0.357749	0.455244	0.799318
Globus pallidus (T1w only)	7.09671	3.9322	1.26715	0.697706
(Globus pallidus (T2w only)	2.39267	0.233799	0.315259	0.924629
(Globus pallidus (T1w & T2w)	2.42717	0.216618	0.307724	0.927084

5.9 Conclusion

Accurate representations of anatomical structures in the deep brain regions is very important for medical modelling and simulation purposes. This chapter presented a deformable multi-material 2-simplex (MM2S) mesh framework. The meshes are multi-material in the sense that they can have consistent shared boundaries with each other.

MM2S meshes can be generated with relative ease because 2-simplex meshes are topologically dual to triangular meshes. It has been shown that this topological duality can be adapted for converting a multi-material triangular mesh into a multi-material 2-simplex mesh such that the mesh's material information is preserved. The MM2S mesh is not a pure 2-simplex mesh in the sense that vertices along the non-manifold edge of the shared boundary may not have exactly 3 neighboring vertices. On the other hand, each material sub-meshes are pure 2-simplex meshes. This feature of the MM2S mesh gives rise to a practical approach for achieving deformation: each sub-mesh is deformed separately, using separately computed internal and

external forces, and then the vertices of the shared boundary are merged, resulting in a consistent multi-material mesh.

Synthetic and realistic examples of MM2S mesh deformations have been presented. In the realistic example, deep brain structures such as the subthalamic nucleus (STN), substantia nigra (SN) and globus pallidus (GP) have been segmented using 7T MRI data. In the case of the SN and STN, the initial multi-material triangular mesh was not very representative of the structures in the 7T MR data, resulting in slight modifications. Even then, Fig. 52 shows that the STN and SN in the ground truth meshes are of different proportions, whereas the initial STN and SN meshes were of similar size. This rather large difference between the initial mesh and the ground truth may account for the low accuracy of the segmentation. Furthermore, the SN and STN are comparatively difficult to detect and segment from standard T2-weighted MRI. On the other hand, the initial mesh of the GP was representative of the ground truth GP. Furthermore, the GP is more visible in T2-weighted MRI than in T1-weighted MRI. The three deformations of the GP showed that the deformation with T1-weighted MRI produced poor results, and deformations of the GP with T2-weighted MRI produced superior results.

Modern medical procedures should be able to utilize multiple sources of information to produce the best possible results. The deformation using both T1 and T2-weighted MRI demonstrated that the multi-material 2-simplex mesh deformation framework is capable of using multi-modal data in order to achieve accurate deformations. In this chapter, only T1 and T2-weighted MR data was utilized, however there are specialized MR protocols, such as the multicontrast, multiecho MR imaging method of [137], or susceptibility-weighted imaging [134], that may be used for more accurate segmentation results.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This section will discuss the limitations of the work presented in this thesis, and discuss possible ways to extend and improve the methods.

6.1 Multi-Material and 2-Manifold Dual Contouring

The standard Dual Contouring (DC) algorithm can generate surface meshes with sharp features. However, a limiting factor of the standard DC algorithm is that it does not guarantee geometrically correct 2-manifold surface meshes. This limitation is because the standard DC algorithm does not examine and handle the situations that gives rise to non-manifold edges and vertices, namely cube ambiguity. The modified DC algorithm presented in Chapter 3 addresses this limitation. Grid cubes are identified as either ambiguous or unambiguous. Ambiguous cubes are divided into tetrahedral cells whose centroids act as minimizers. Three novel polygon generation rules are presented by which the modified DC algorithm is able to produce watertight and 2-manifold surface meshes.

Chapter 4 extends the modified DC algorithm to generate multi-material and 2-manifold surface meshes from labeled volumetric data. By their very nature, these multi-material meshes can have non-manifold edges and vertices. In Chapter 4, a multi-material mesh is defined as being 2-manifold in the sense that each material sub-mesh, if separated, is completely watertight and 2-manifold on their own. Grid cubes are identified as either ambiguous or unambiguous, and the method by which ambiguous cubes are sub-divided into tetrahedral cells ensures that the shared boundaries remain consistent for each sub-mesh. The ambiguity identification method, as well as subsequent polygon generation rules are extremely generalized such that it is not

necessary to handle each ambiguity case separately. Only one set of polygon generalization rules are presented, and these rules are able to produce geometrically correct meshes for any ambiguous cases, and for any combination of ambiguous and unambiguous cubes.

Although the proposed DC algorithm can generate multi-material and 2-manifold surface meshes, there still exists a few limitation in the current implementation, and these limitations and possible solutions are discussed in the following section.

6.1.1 Mesh Smoothness

As can be seen Fig. 22, Fig. 38 and Fig. 40, the surface meshes generated by the proposed method exhibits a staircase-like effect. While this effect can be mitigated by the application of post-processing smoothing filters (like Laplacian or Taubin), the current implementation does not address the underlying cause. The staircase-like effect is mainly due to the fact that the minimizers of unambiguous cubes are constrained to remain within their respective cubes. The method used to compute minimizers can result in some minimizers being positioned outside their respective cube. Fig. 27 shows that when minimizers are allowed to be placed outside their respective cubes, the resulting mesh can have a smoother appearance but contain cracks and/or gaps and intersecting polygons. One solution to address this limitation is to use a different method for computing minimizers. Another approach may to incorporate into the polygon generation process the adjacent grid cube where the minimizer moves to.

A second cause for the staircase-like effects are the centroids of tetrahedral cells being used as minimizers: specifically, these minimizers are static and are not able to move around within their respective tetrahedral cells in the same way that minimizers of unambiguous cubes can. This further exacerbates the staircase-like effects exhibited in the surface meshes. A

mechanism for adjusting the positions of these minimizers based on the corner values of the tetrahedral cells may result in smoother meshes.

6.1.2 Triangle Quality

The proposed multi-material and 2-manifold DC algorithm is able to generate triangles of good quality for the most part. However, some poor quality triangles are being produced, as evidenced in TABLE 5, TABLE 7 and TABLE 8. The reason for these poor quality triangles is that the current implementation of the Minimal Edge Rule is not configured towards producing quality triangles. The Minimal Edge Rule produces an n -gon by linking all the minimizers of all cubic or tetrahedral cells sharing the minimal edge, and then triangulating the n -gon. Ideally, 2D Delaunay-based tessellation methods would be used to produce good-quality triangles, but Delaunay-based methods tend to produce convex polygons. Enforcing convexity on the n -gon generated by the Minimal Edge Rule can result in non-manifold edges in the output mesh. One solution to this limitation is to use constrained Delaunay methods to produce good quality triangles while maintaining the non-convex aspect of the n -gon. Another solution might be to partition the non-convex n -gon into a two or more convex polygons, and then applying 2D Delaunay-based tessellation to produce good quality triangulations.

6.1.3 Adaptive Octree

The current implementation of the proposed DC algorithm only works on non-adaptive octrees. Incorporating an adaptive octree allows the standard DC algorithm to be more flexible in terms of creating producing triangles of differing sizes (larger triangles for flat surfaces). However, the current implementation of the proposed multi-material and 2-manifold DC

algorithm assumes that all grid cubes are of the same size (lowest level of the octree), and each minimal edge shares four cubes. Incorporating an adaptive octree may prove to be difficult in the sense that new cube decomposition methods and polygon generation methods need to be formulated.

6.2 Multi-Material 2-Simplex Deformable Meshes

A multi-material 2-simplex (MM2S) deformable mesh framework has been presented in Chapter 5, along with results using synthetic and realistic data. While the reported results show promise, the current deformation framework is somewhat limited. This section will discuss some of these limitations and present improvements to the multi-material 2-simplex mesh.

6.2.1 Creation of the Initial Mesh

In Chapter 5, a process for converting a multi-material triangular mesh into a multi-material 2-simplex mesh has been presented. One limitation with this method is that the initialized MM2S mesh may not in fact be very representative of the target structure, as evidenced in Fig. 52. Since the initial MM2S mesh plays a significant role in the accuracy of the deformation, it makes sense to incorporate more structural information when generating the initial MM2S mesh. The initial MM2S mesh will undergo deformation in order to represent a specific shape. Therefore, one approach is to utilize statistical shape information in generating the initial MM2S mesh so that the resulting mesh would be a generalization of the specific shape.

6.2.2 Dependency on Image Gradients

The MM2S deformation process uses internal forces which are based on mesh geometry, and external forces which are based on image gradients. The accuracy of the deformation depends significantly on the external force, and therefore it can be said that the deformation process is dependent on image gradients. If the gradients are clear and crisp, as is the case in Section 5.6, the deformation will likely result in a good approximation of the target structure. On the other hand, if the gradients are not very clear, or if there are competing gradients within the search space, then the deformation is likely to result in a poor segmentation, as demonstrated in Section 5.8. When the globus pallidus (GP) is segmented using only T1-weighted MRI, the resulting mesh shows a significantly large error. This is because the GP is not well delineated in T1-weighted MRI, and therefore the gradient of the GP was not well defined. The segmentation of the GP using T2-weighted and multimodal MRI produced superior results, owing to the fact that the GP has stronger gradients in T2-weighted MRI.

One way to alleviate this dependency on image gradients is to utilize specialized data. For example, the subthalamic nucleus (STN) is difficult to detect in standard T1 and T2-weighted MRI. Specialized MR acquisition protocols, such as the multicontrast, multiecho MR imaging method introduced in [137], which can directly delineate the STN and other basal ganglia structures, can be used to produce superior segmentation results.

Another approach is to use *a priori* statistical information to guide the deformation. The idea is to represent statistical shape information as another force in the deformation process. This statistical and shape-aware approach was successfully utilized in [95-97] in the segmentation of lumbar vertebrae and intervertebral discs.

6.2.3 Multi-Resolution Approach to Deformation

Multi-resolution registration, in terms of discrete deformable meshes, implies an ability of the deformable mesh to take on several levels of resolutions, coarser levels for algorithmic robustness, finer levels in order to capture small and fine features of the target. Finer resolution meshes have a larger number of cells and vertices, and thus incur increased processing time. Moreover, if the deformation starts off with a very high resolution mesh, not only will the overall processing time increase, the mesh may be susceptible to being trapped in false minima. The solution is to start the deformation with a coarse mesh, and then after a suitable number of deformation iterations, switch to a higher resolution mesh that is based on the previous deformed coarse mesh. The coarse mesh will deform at a much faster pace since it has fewer cells and vertices.

The current MM2S deformable mesh is only able to deform at a fixed mesh resolution. The results presented in Chapter 5 all use a medium-level mesh throughout their respective deformations. Due to their relatively coarse nature, some of the cells/polygons in the meshes used thus far are rather large, and these meshes are unable to accurately represent small features, or surfaces with high curvature. A multi-resolution approach to the MM2S mesh can potentially result in more robust and accurate segmentation results.

6.2.4 Mesh Decimation and Mesh Uniformity

For 2-simplex meshes, there exists several topology operators that can be used to alter the mesh, as discussed in Chapter 2. Amongst these, the TO_1 and TO_2 operators are topology preserving. The TO_1 operator splits one cell into two adjacent cells, thereby increasing the number of points by two and the number of cells by one. On the other hand, the TO_2 operator

merges two adjacent cells into one cells, thereby decreasing the number of cells and points by one and two, respectively.

It is theoretically possible to achieve mesh decimation by applying a series of TO_2 operators to a 2-simplex mesh. The advantage in this approach to decimation is that, as mentioned above, the TO_2 operator is topology preserving. Small cells can be merged into larger cells, or all the cells within a specific region of the mesh can be merged into a few large cells.

Another application of topological operators is towards producing meshes with uniform cells. The current method of initializing MM2S meshes from triangular meshes is prone to producing non-uniform cells. Some cells are made up of as few as three points while other cells comprise of more than ten points. TO_1 operators can be applied to larger cells in order to break them down into smaller cells, and TO_2 operators can be applied to small cells in order to merge them into larger cells. This would result in a mesh with uniform cell sizes.

One point of concern is that these topology operators assume that the mesh is a pure 2-simplex mesh, in other words, every vertex is connected to exactly three neighboring vertices. This is not the case with a multi-material 2-simplex mesh where vertices on the non-manifold edge of the shared boundary can have more than three neighboring vertices. One approach to overcoming this issue is to apply topology operators on material sub-meshes only, in such a manner that the shared boundary remains consistent. In other words, operations on shared boundaries should be duplicated across the respective sub-meshes. Another approach is to selectively apply the topology operators only on parts of the MM2S mesh that satisfy the connectivity criteria of a 2-simplex mesh. Whichever approach is used, topology operators can be a very useful feature for multi-material 2-simplex meshes.

REFERENCES

- [1] J. S. Perlmutter and J. W. Mink, "Deep brain stimulation," *Annu. Rev. Neurosci.*, vol. 29, pp. 229-257, 2006.
- [2] H. A. Eltahawy, J. Saint-Cyr, N. Giladi, A. E. Lang, and A. M. Lozano, "Primary dystonia is more responsive than secondary dystonia to pallidal interventions: outcome after pallidotomy or pallidal deep brain stimulation," *Neurosurgery*, vol. 54, pp. 613-621, 2004.
- [3] A. M. Lozano, R. Kumar, R. Gross, N. Giladi, W. Hutchison, J. Dostrovsky, *et al.*, "Globus pallidus internus pallidotomy for generalized dystonia," *Movement Disorders*, vol. 12, pp. 865-870, 1997.
- [4] J. L. Vitek, V. Chockkan, J. Y. Zhang, Y. Kaneoke, M. Evatt, M. R. DeLong, *et al.*, "Neuronal activity in the basal ganglia in patients with generalized dystonia and hemiballismus," *Annals of Neurology*, vol. 46, pp. 22-35, 1999.
- [5] D. Yoshor, W. J. Hamilton, W. Ondo, J. Jankovic, and R. G. Grossman, "Comparison of thalamotomy and pallidotomy for the treatment of dystonia," *Neurosurgery*, vol. 48, pp. 818-826, 2001.
- [6] N. J. Diederich, K. Kalteis, M. Stamenkovic, V. Pieri, and F. Alesch, "Efficient internal pallidal stimulation in Gilles de la Tourette syndrome: a case report," *Movement Disorders*, vol. 20, pp. 1496-1499, 2005.
- [7] A. W. Flaherty, Z. M. Williams, R. Amirnovin, E. Kasper, S. L. Rauch, G. R. Cosgrove, *et al.*, "Deep brain stimulation of the anterior internal capsule for the treatment of Tourette syndrome: technical case report," *Neurosurgery*, vol. 57, p. E403, 2005.
- [8] J. Houeto, C. Karachi, L. Mallet, B. Pillon, J. Yelnik, V. Mesnage, *et al.*, "Tourette's syndrome and deep brain stimulation," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 76, pp. 992-995, 2005.
- [9] V. Visser-Vandewalle, Y. Temel, P. Boon, F. Vreeling, H. Colle, G. Hoogland, *et al.*, "Chronic bilateral thalamic stimulation: a new therapeutic approach in intractable Tourette syndrome: report of three cases," *Journal of Neurosurgery*, vol. 99, pp. 1094-1100, 2003.
- [10] M. Rodriguez-Oroz, I. Zamarbide, J. Guridi, M. Palmero, and J. Obeso, "Efficacy of deep brain stimulation of the subthalamic nucleus in Parkinson's disease 4 years after surgery: double blind and open label evaluation," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 75, pp. 1382-1385, 2004.
- [11] V. C. Anderson, K. J. Burchiel, P. Hogarth, J. Favre, and J. P. Hammerstad, "Pallidal vs subthalamic nucleus deep brain stimulation in Parkinson disease," *Archives of Neurology*, vol. 62, pp. 554-560, 2005.

- [12] I. Parkinson's Disease Foundation. (2014, 04/18/2014). *Statistics on Parkinson's - Parkinson's Disease Foundation (PDF)*. Available: http://www.pdf.org/en/parkinson_statistics
- [13] P. O. Shafer and J. I. Sirven. (2013, 04/18/2014). *Epilepsy Statistics / Epilepsy Foundation*. Available: <http://www.epilepsy.com/learn/epilepsy-statistics>
- [14] M. Fava, "Diagnosis and definition of treatment-resistant depression," *Biological Psychiatry*, vol. 53, pp. 649-659, 2003.
- [15] D. Denys, M. Mantione, M. Figee, P. van den Munckhof, F. Koerselman, H. Westenberg, *et al.*, "Deep brain stimulation of the nucleus accumbens for treatment-refractory obsessive-compulsive disorder," *Archives of General Psychiatry*, vol. 67, pp. 1061-1068, 2010.
- [16] D. Sierens, S. Kutz, J. Pilitsis, A. Bakay, and R. Bakay, "Stereotactic surgery with microelectrode recordings," *Movement Disorders Surgery. Thieme, New York*, pp. 83-114, 2008.
- [17] K. E. Lyons, S. B. Wilkinson, J. Overman, and R. Pahwa, "Surgical and hardware complications of subthalamic stimulation A series of 160 procedures," *Neurology*, vol. 63, pp. 612-616, 2004.
- [18] P. A. Starr, A. J. Martin, J. L. Ostrem, P. Talke, N. Levesque, and P. S. Larson, "Subthalamic nucleus deep brain stimulator placement using high-field interventional magnetic resonance imaging and a skull-mounted aiming device: technique and application accuracy," *Journal of Neurosurgery*, vol. 112, p. 479, 2010.
- [19] E. A. Petersen, E. M. Holl, I. Martinez-Torres, T. Foltynie, P. Limousin, M. I. Hariz, *et al.*, "Minimizing brain shift in stereotactic functional neurosurgery," *Neurosurgery*, vol. 67, pp. ons213-ons221, 2010.
- [20] K. A. Sillay, L. Kumbier, C. Ross, M. Brady, A. Alexander, A. Gupta, *et al.*, "Perioperative brain shift and deep brain stimulating electrode deformation analysis: implications for rigid and non-rigid devices," *Annals of Biomedical Engineering*, vol. 41, pp. 293-304, 2013.
- [21] Y. Miyagi, F. Shima, and T. Sasaki, "Brain shift: an error factor during implantation of deep brain stimulation electrodes," *Journal of Neurosurgery*, vol. 107, pp. 989-997, 2007.
- [22] M. F. Khan, K. Mewes, R. E. Gross, and O. Škrinjar, "Assessment of brain shift related to deep brain stimulation surgery," *Stereotactic and Functional Neurosurgery*, vol. 86, pp. 44-53, 2007.
- [23] M. I. Hariz, "Complications of deep brain stimulation surgery," *Movement Disorders*, vol. 17, pp. S162-S166, 2002.
- [24] Y. Gologorsky, S. Ben-Haim, E. L. Moshier, J. Godbold, M. Tagliati, D. Weisz, *et al.*, "Transgressing the ventricular wall during subthalamic deep brain stimulation surgery for

- Parkinson disease increases the risk of adverse neurological sequelae," *Neurosurgery*, vol. 69, pp. 294-300, 2011.
- [25] L. Mallet, M. Schüpbach, K. N'Diaye, P. Remy, E. Bardinnet, V. Czernecki, *et al.*, "Stimulation of subterritories of the subthalamic nucleus reveals its role in the integration of the emotional and motor aspects of behavior," *Proceedings of the National Academy of Sciences*, vol. 104, pp. 10661-10666, 2007.
 - [26] E. O. Richter, T. Hoque, W. Halliday, A. M. Lozano, and J. A. Saint-Cyr, "Determining the position and size of the subthalamic nucleus based on magnetic resonance imaging results in patients with advanced Parkinson disease," *Journal of Neurosurgery*, vol. 100, pp. 541-546, 2004.
 - [27] Y. Temel, A. Kessels, S. Tan, A. Topdag, P. Boon, and V. Visser-Vandewalle, "Behavioural changes after bilateral subthalamic stimulation in advanced Parkinson disease: a systematic review," *Parkinsonism & Related Disorders*, vol. 12, pp. 265-272, 2006.
 - [28] A. Daniele, P. Spinelli, and C. Piccininni, "Cognitive and Behavioural Changes After Deep Brain Stimulation of the Subthalamic Nucleus in Parkinson's Disease," *Explicative Cases of Controversial Issues in Neurosurgery, InTech*, 2012.
 - [29] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," in *ACM Transactions on Graphics (TOG)*, 2002, pp. 339-346.
 - [30] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *ACM Siggraph Computer Graphics*, 1987, pp. 163-169.
 - [31] H. Delingette, "General object reconstruction based on simplex meshes," *International Journal of Computer Vision*, vol. 32, pp. 111-146, 1999.
 - [32] H. Delingette, "Simplex meshes: a general representation for 3D shape reconstruction," in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, 1994, pp. 856-859.
 - [33] B. Gilles and N. Magnenat-Thalmann, "Musculoskeletal MRI segmentation using multi-resolution simplex meshes with medial representations," *Medical Image Analysis*, vol. 14, pp. 291-302, 2010.
 - [34] G. H. Golub and C. F. Van Loan, "Matrix Computations. Johns Hopkins series in the mathematical sciences," *Johns Hopkins University Press, Baltimore, MD*, 1989.
 - [35] L. P. Kobbelt, M. Botsch, U. Schwanecke, and H.-P. Seidel, "Feature sensitive surface extraction from volume data," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, 2001, pp. 57-66.
 - [36] P. Lindstrom, "Out-of-core simplification of large polygonal models," in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 259-262.

- [37] P. Volino and N. Magnenat-Thalmann, "Implicit midpoint integration and adaptive damping for efficient cloth simulation," *Computer Animation and Virtual Worlds*, vol. 16, pp. 163-175, 2005.
- [38] P. Volino and N. M. Thalmann, "Implementing fast cloth simulation with collision response," in *Computer Graphics International, 2000. Proceedings*, 2000, pp. 257-266.
- [39] B. Gilles, L. Moccozet, and N. Magnenat-Thalmann, "Anatomical modelling of the musculoskeletal system from MRI," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2006*, ed: Springer, 2006, pp. 289-296.
- [40] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*: Prentice-Hall, Inc., 2006.
- [41] A. C. Bovik, *The essential guide to image processing*: Academic Press, 2009.
- [42] L. H. de Figueiredo, J. de Miranda Gomes, D. Terzopoulos, and L. Velho, "Physically-based methods for polygonization of implicit surfaces," in *Proceedings of Graphics Interface*, 1992, pp. 250-257.
- [43] T. Ju, "Fixing geometric errors on polygonal models: A survey," *Journal of Computer Science and Technology*, vol. 24, pp. 19-29, 2009.
- [44] T. S. Newman and H. Yi, "A survey of the marching cubes algorithm," *Computers & Graphics*, vol. 30, pp. 854-879, 2006.
- [45] T. K. Dey and S. Goswami, "Tight cocone: a water-tight surface reconstructor," in *Proceedings of the eighth ACM symposium on Solid modeling and Applications*, 2003, pp. 127-134.
- [46] G. M. Treece, R. W. Prager, and A. H. Gee, "Regularised marching tetrahedra: improved iso-surface extraction," *Computers & Graphics*, vol. 23, pp. 583-598, 1999.
- [47] T. Ju and T. Udeshi, "Intersection-free contouring on an octree grid," in *Pacific Graphics*, 2006.
- [48] N. Zhang, W. Hong, and A. Kaufman, "Dual contouring with topology-preserving simplification using enhanced cell representation," in *Visualization, 2004. IEEE*, 2004, pp. 505-512.
- [49] G. Varadhan, S. Krishnan, Y. J. Kim, and D. Manocha, "Feature-sensitive subdivision and isosurface reconstruction," in *Visualization, 2003. VIS 2003. IEEE*, 2003, pp. 99-106.
- [50] Y. Zhang and J. Qian, "Dual contouring for domains with topology ambiguity," *Computer Methods in Applied Mechanics and Engineering*, vol. 217, pp. 34-45, 2012.
- [51] Y. Zhang and J. Qian, "Resolving topology ambiguity for multiple-material domains," *Computer Methods in Applied Mechanics and Engineering*, vol. 247, pp. 166-178, 2012.
- [52] B.-S. Sohn, "Topology preserving tetrahedral decomposition of trilinear cell," in *Computational Science—ICCS 2007*, ed: Springer, 2007, pp. 350-357.

- [53] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "Meshlab: an open-source mesh processing tool," in *Eurographics Italian Chapter Conference*, 2008, pp. 129-136.
- [54] M. M. Chakravarty, G. Bertrand, C. P. Hodge, A. F. Sadikot, and D. L. Collins, "The creation of a brain atlas for image guided neurosurgery using serial histological data," *Neuroimage*, vol. 30, pp. 359-376, 2006.
- [55] J. Shewchuk, "What is a good linear finite element? Interpolation, conditioning, and quality measures," in *Proceedings of the 11th International Meshing Roundtable*, 2002.
- [56] T. Ju, "Robust repair of polygonal models," *ACM Transactions on Graphics (TOG)*, vol. 23, pp. 888-895, 2004.
- [57] L. Schmitz, C. Dietrich, and J. L. Comba, "Efficient and high quality contouring of isosurfaces on uniform grids," in *Computer Graphics and Image Processing (SIBGRAPI), 2009 XXII Brazilian Symposium on*, 2009, pp. 64-71.
- [58] S. J. Owen, "A survey of unstructured mesh generation technology," in *IMR*, 1998, pp. 239-267.
- [59] H. Si, "TetGen, a Delaunay-based quality tetrahedral mesh generator," *ACM Transactions on Mathematical Software (TOMS)*, vol. 41, p. 11, 2015.
- [60] The CGAL Project, *CGAL User and Reference Manual*, 4.9 ed.: CGAL Editorial Board, 2016.
- [61] J. Bloomenthal and K. Ferguson, "Polygonization of non-manifold implicit surfaces," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 1995, pp. 309-316.
- [62] H.-C. Hege, D. STALLING, M. SEEBASS, and M. Zockler, "A Generalized Marching Cubes Algorithm Based On Non-Binary," Citeseer1997.
- [63] K. S. Bonnell, M. A. Duchaineau, D. R. Schikore, B. Hamann, and K. I. Joy, "Material interface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, pp. 500-511, 2003.
- [64] Z. Wu and J. M. Sullivan, "Multiple material marching cubes algorithm," *International Journal for Numerical Methods in Engineering*, vol. 58, pp. 189-207, 2003.
- [65] B. Reitinger, A. Bornik, and R. Beichel, "Constructing smooth non-manifold meshes of multi-labeled volumetric datasets," *WSCG (Full Papers)*, pp. 227-234, 2005.
- [66] M. Bertram, G. Reis, R. H. van Lengen, S. Köhn, and H. Hagen, "Non-manifold Mesh Extraction from Time-varying Segmented Volumes used for Modeling a Human Heart," in *EuroVis*, 2005, pp. 199-206.

- [67] S. Bischoff and L. Kobbelt, "Extracting consistent and manifold interfaces from multi-valued volume data sets," in *Bildverarbeitung für die Medizin 2006*, ed: Springer, 2006, pp. 281-285.
- [68] J.-D. Boissonnat and S. Oudot, "Provably good sampling and meshing of surfaces," *Graphical Models*, vol. 67, pp. 405-451, 2005.
- [69] S. Oudot, L. Rineau, and M. Yvinec, "Meshing volumes bounded by smooth surfaces," in *Proceedings of the 14th International Meshing Roundtable*, 2005, pp. 203-219.
- [70] J.-P. Pons, F. Ségonne, J.-D. Boissonnat, L. Rineau, M. Yvinec, and R. Keriven, "High-quality consistent meshing of multi-label datasets," in *Biennial International Conference on Information Processing in Medical Imaging*, 2007, pp. 198-210.
- [71] S. Dillard, J. Bingert, D. Thoma, and B. Hamann, "Construction of simplified boundary surfaces from serial-sectioned metal micrographs," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1528-1535, 2007.
- [72] M. Meyer, R. Whitaker, R. M. Kirby, C. Ledergerber, and H. Pfister, "Particle-based sampling and meshing of surfaces in multimaterial volumes," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, pp. 1539-1546, 2008.
- [73] B. Li, G. E. Christensen, E. A. Hoffman, G. McLennan, and J. M. Reinhardt, "Establishing a normative atlas of the human lung: intersubject warping and registration of volumetric CT images," *Academic Radiology*, vol. 10, pp. 255-265, 2003.
- [74] H. Park, P. H. Bland, and C. R. Meyer, "Construction of an abdominal probabilistic atlas and its application in segmentation," *Medical Imaging, IEEE Transactions on*, vol. 22, pp. 483-492, 2003.
- [75] I. Sluimer, M. Prokop, and B. van Ginneken, "Toward automated segmentation of the pathological lung in CT," *Medical Imaging, IEEE Transactions on*, vol. 24, pp. 1025-1038, 2005.
- [76] J. Stancanella, P. Romanelli, N. Modugno, P. Cerveri, G. Ferrigno, F. Uggeri, *et al.*, "Atlas-based identification of targets for functional radiosurgery," *Medical Physics*, vol. 33, pp. 1603-1611, 2006.
- [77] I. Isgum, M. Staring, A. Rutten, M. Prokop, M. A. Viergever, and B. van Ginneken, "Multi-atlas-based segmentation with local decision fusion—Application to cardiac and aortic segmentation in CT scans," *Medical Imaging, IEEE Transactions on*, vol. 28, pp. 1000-1010, 2009.
- [78] T. Rohlfing, D. B. Russakoff, and C. R. Maurer Jr, "Performance-based classifier combination in atlas-based image segmentation using expectation-maximization parameter estimation," *Medical Imaging, IEEE Transactions on*, vol. 23, pp. 983-994, 2004.

- [79] X. Artaechevarria, A. Munoz-Barrutia, and C. Ortiz-de-Solorzano, "Combination strategies in multi-atlas image segmentation: Application to brain MR data," *Medical Imaging, IEEE Transactions on*, vol. 28, pp. 1266-1277, 2009.
- [80] T. W. Sederberg and S. R. Parry, "Free-form deformation of solid geometric models," in *ACM Siggraph Computer Graphics*, 1986, pp. 151-160.
- [81] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *ACM Siggraph Computer Graphics*, 1987, pp. 205-214.
- [82] U. Meier, O. López, C. Monserrat, M. C. Juan, and M. Alcaniz, "Real-time deformable models for surgery simulation: a survey," *Computer Methods and Programs in Biomedicine*, vol. 77, pp. 183-197, 2005.
- [83] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321-331, 1988.
- [84] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, pp. 38-59, 1995.
- [85] J. Montagnat and H. Delingette, "Volumetric medical images segmentation using shape constrained deformable models," in *CVRMed-MRCAS'97*, 1997, pp. 13-22.
- [86] J. Schmid, J. A. I. Guitián, E. Gobbetti, and N. Magnenat-Thalmann, "A GPU framework for parallel segmentation of volumetric images using discrete deformable models," *The Visual Computer*, vol. 27, pp. 85-95, 2011.
- [87] J. Schmid and N. Magnenat-Thalmann, "MRI bone segmentation using deformable models and shape priors," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2008*, ed: Springer, 2008, pp. 119-126.
- [88] J. Schmid, A. Sandholm, F. Chung, D. Thalmann, H. Delingette, and N. Magnenat-Thalmann, "Musculoskeletal simulation model generation from MRI data sets and motion capture data," in *Recent Advances in the 3D Physiological Human*, ed: Springer, 2009, pp. 3-19.
- [89] J. Montagnat and H. Delingette, "Space and time shape constrained deformable surfaces for 4D medical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2000*, 2000, pp. 196-205.
- [90] J. Montagnat and H. Delingette, "4D deformable models with temporal constraints: application to 4D cardiac image segmentation," *Medical Image Analysis*, vol. 9, pp. 87-100, 2005.
- [91] J. Montagnat, M. Sermesant, H. Delingette, G. Malandain, and N. Ayache, "Anisotropic filtering for model-based segmentation of 4D cylindrical echocardiographic images," *Pattern Recognition Letters*, vol. 24, pp. 815-828, 2003.

- [92] D. Cremers, C. Schnorr, and J. Weickert, "Diffusion-snakes: combining statistical shape knowledge and image information in a variational framework," in *Variational and Level Set Methods in Computer Vision, 2001. Proceedings. IEEE Workshop on*, 2001, pp. 137-144.
- [93] D. Cremers, C. Schnörr, J. Weickert, and C. Schellewald, "Diffusion–snakes using statistical shape knowledge," in *Algebraic Frames for the Perception-action Cycle*, ed: Springer, 2000, pp. 164-174.
- [94] D. Cremers, F. Tischhäuser, J. Weickert, and C. Schnörr, "Diffusion snakes: Introducing statistical shape knowledge into the Mumford-Shah functional," *International Journal of Computer Vision*, vol. 50, pp. 295-313, 2002.
- [95] R. Haq, R. Aras, D. A. Besachio, R. C. Borgie, and M. A. Audette, "3D lumbar spine intervertebral disc segmentation and compression simulation from MRI using shape-aware models," *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, pp. 45-54, 2015.
- [96] R. Haq, D. A. Besachio, R. C. Borgie, and M. A. Audette, "Using shape-aware models for lumbar spine intervertebral disc segmentation," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, 2014, pp. 3191-3196.
- [97] R. Haq, J. Cates, D. A. Besachio, R. C. Borgie, and M. A. Audette, "Statistical Shape Model Construction of Lumbar Vertebrae and Intervertebral Discs in Segmentation for Discectomy Surgery Simulation," in *International Workshop on Computational Methods and Clinical Applications for Spine Imaging*, 2015, pp. 85-96.
- [98] N. Patel, S. Sultana, T. Rashid, D. Krusienski, and M. A. Audette, "Application and histology-driven refinement of active contour models to functional region and nerve delineation: towards a digital brainstem atlas," in *SPIE Medical Imaging*, 2015, pp. 94150I-94150I-10.
- [99] S. Sultana, P. Agrawal, S. Y. Elhabian, R. T. Whitaker, T. Rashid, J. E. Blatt, *et al.*, "Towards a Statistical Shape-Aware Deformable Contour Model for Cranial Nerve Identification," in *Workshop on Clinical Image-Based Procedures*, 2016, pp. 68-76.
- [100] S. Sultana, J. E. Blatt, Y. Lee, M. Ewend, J. S. Cetas, A. Costa, *et al.*, "Patient-specific cranial nerve identification using a discrete deformable contour model for skull base neurosurgery planning and simulation," in *Workshop on Clinical Image-Based Procedures*, 2015, pp. 36-44.
- [101] M. A. Audette, F. P. Ferrie, and T. M. Peters, "An algorithmic overview of surface registration techniques for medical imaging," *Medical Image Analysis*, vol. 4, pp. 201-217, 2000.
- [102] C. Broit, "Optimal registration of deformed images," 1981.
- [103] J. Ashburner, "A fast diffeomorphic image registration algorithm," *Neuroimage*, vol. 38, pp. 95-113, 2007.

- [104] G. E. Christensen, S. C. Joshi, and M. I. Miller, "Volumetric transformation of brain anatomy," *Medical Imaging, IEEE Transactions on*, vol. 16, pp. 864-877, 1997.
- [105] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, "Deformable templates using large deformation kinematics," *Image Processing, IEEE Transactions on*, vol. 5, pp. 1435-1447, 1996.
- [106] J.-P. Thirion, "Image matching as a diffusion process: an analogy with Maxwell's demons," *Medical Image Analysis*, vol. 2, pp. 243-260, 1998.
- [107] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 567-585, 1989.
- [108] F. L. Bookstein, "Thin-plate splines and the atlas problem for biomedical images," in *Information Processing in Medical Imaging*, 1991, pp. 326-342.
- [109] S. Lowitzsch, "Approximation and interpolation employing divergence-free radial basis functions with applications," Citeseer, 2002.
- [110] A. M. Siddiqui, A. Masood, and M. Saleem, "A locally constrained radial basis function for registration and warping of images," *Pattern Recognition Letters*, vol. 30, pp. 377-390, 2009.
- [111] A. Tristán-Vega and V. García-Pérez, "Comments on: A locally constrained radial basis function for registration and warping of images," *Pattern Recognition Letters*, vol. 32, pp. 586-589, 2011.
- [112] M. Holden, "A review of geometric transformations for nonrigid body registration," *Medical Imaging, IEEE Transactions on*, vol. 27, pp. 111-128, 2008.
- [113] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee, "Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain," *Medical Image Analysis*, vol. 12, pp. 26-41, 2008.
- [114] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys (CSUR)*, vol. 24, pp. 325-376, 1992.
- [115] J. Kim and J. A. Fessler, "Intensity-based image registration using robust correlation coefficients," *Medical Imaging, IEEE Transactions on*, vol. 23, pp. 1430-1444, 2004.
- [116] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International Journal of Computer Vision*, vol. 24, pp. 137-154, 1997.
- [117] W. M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, "Multi-modal volume registration by maximization of mutual information," *Medical Image Analysis*, vol. 1, pp. 35-51, 1996.

- [118] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, "Automated multi-modality image registration based on information theory," in *Information Processing in Medical Imaging*, 1995, pp. 263-274.
- [119] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens, "Multimodality image registration by maximization of mutual information," *Medical Imaging, IEEE Transactions on*, vol. 16, pp. 187-198, 1997.
- [120] H. J. Johnson and G. E. Christensen, "Consistent landmark and intensity-based image registration," *IEEE Transactions on Medical Imaging*, vol. 21, pp. 450-461, 2002.
- [121] K. Rohr, H. S. Stiehl, R. Sprengel, T. M. Buzug, J. Weese, and M. Kuhn, "Landmark-based elastic registration using approximating thin-plate splines," *IEEE Transactions on Medical Imaging*, vol. 20, pp. 526-534, 2001.
- [122] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis: with Applications in R*: John Wiley & Sons, 2016.
- [123] P. J. Besl and N. D. McKay, "Method for registration of 3-D shapes," in *Robotics-DL Tentative*, 1992, pp. 586-606.
- [124] O. Fluck, C. Vetter, W. Wein, A. Kamen, B. Preim, and R. Westermann, "A survey of medical image registration on graphics hardware," *Computer Methods and Programs in Biomedicine*, vol. 104, pp. e45-e57, 2011.
- [125] R. Shams, P. Sadeghi, R. A. Kennedy, and R. I. Hartley, "A survey of medical image registration on multicore and the GPU," *Signal Processing Magazine, IEEE*, vol. 27, pp. 50-60, 2010.
- [126] Y. Liu, A. Fedorov, R. Kikinis, and N. Chrisochoides, "Real-time non-rigid registration of medical images on a cooperative parallel architecture," in *Bioinformatics and Biomedicine, 2009. BIBM'09. IEEE International Conference on*, 2009, pp. 401-404.
- [127] N. Archip, O. Clatz, S. Whalen, D. Kacher, A. Fedorov, A. Kot, *et al.*, "Non-rigid alignment of pre-operative MRI, fMRI, and DT-MRI with intra-operative MRI for enhanced visualization and navigation in image-guided neurosurgery," *NeuroImage*, vol. 35, pp. 609-624, 4/1/ 2007.
- [128] A. Wittek, G. Joldes, M. Couton, S. K. Warfield, and K. Miller, "Patient-specific non-linear finite element modelling for predicting soft organ deformation in real-time; Application to non-rigid neuroimage registration," *Progress in Biophysics and Molecular Biology*, vol. 103, pp. 292-303, 2010.
- [129] A. Fedorov, E. Billet, M. Prastawa, G. Gerig, A. Radmanesh, S. K. Warfield, *et al.*, "Evaluation of brain MRI alignment with the robust Hausdorff distance measures," in *Advances in Visual Computing*, ed: Springer, 2008, pp. 594-603.
- [130] S. Oguro, K. Tuncali, H. Elhawary, P. R. Morrison, N. Hata, and S. G. Silverman, "Image registration of pre-procedural MRI and intra-procedural CT images to aid CT-guided

- percutaneous cryoablation of renal tumors," *International Journal of Computer Assisted Radiology and Surgery*, vol. 6, pp. 111-117, 2011.
- [131] Y. Liu, C. Yao, L. Zhou, and N. Chrisochoides, "A point based non-rigid registration for tumor resection using IMRI," in *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, 2010, pp. 1217-1220.
 - [132] S. Oguro, J. Tokuda, H. Elhawary, S. Haker, R. Kikinis, C. Tempany, *et al.*, "MRI signal intensity based B-Spline nonrigid registration for pre-and intraoperative imaging during prostate brachytherapy," *Journal of Magnetic Resonance Imaging*, vol. 30, pp. 1052-1058, 2009.
 - [133] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 629-639, 1990.
 - [134] R. L. O’Gorman, K. Shmueli, K. Ashkan, M. Samuel, D. J. Lythgoe, A. Shahidiani, *et al.*, "Optimal MRI methods for direct stereotactic targeting of the subthalamic nucleus and globus pallidus," *European Radiology*, vol. 21, pp. 130-136, 2011.
 - [135] B. T. Wang, S. Poirier, T. Guo, A. G. Parrent, T. M. Peters, and A. R. Khan, "Generation and evaluation of an ultra-high-field atlas with applications in DBS planning," in *SPIE Medical Imaging*, 2016, pp. 97840H-97840H-10.
 - [136] S. C. Deoni, T. M. Peters, and B. K. Rutt, "High-resolution T1 and T2 mapping of the brain in a clinically acceptable time with DESPOT1 and DESPOT2," *Magnetic Resonance in Medicine*, vol. 53, pp. 237-241, 2005.
 - [137] Y. Xiao, S. Beriault, G. B. Pike, and D. L. Collins, "Multicontrast multiecho FLASH MRI for targeting the subthalamic nucleus," *Magnetic Resonance Imaging*, vol. 30, pp. 627-640, 2012.

VITA

Tanweer Rashid
 Department of Modelling, Simulation and Visualization Engineering
 Old Dominion University
 Engineering and Computational Sciences Building Suite 1300
 4700 Elkhorn Ave
 Norfolk, VA 23529, USA

Education

Master of Science (M.S.) in Modeling and Simulation

Old Dominion University, Norfolk, VA, USA
 Department of Modeling, Simulation and Visualization Engineering

Bachelor of Science (B.S.) in Computer Engineering

North South University, Dhaka, Bangladesh
 Department of Computer Science and Engineering

Journal and Conference Publications

Rashid, T., Sultana, S., & Audette, M. A. (2016). 2-manifold surface meshing using dual contouring with tetrahedral decomposition. *Advances in Engineering Software*, 102, 83-96.

Sultana, S., Blatt, J., Gilles, B., **Rashid, T.**, & Audette, M. (2017). MRI-based medial axis extraction and boundary segmentation of cranial nerves through discrete deformable 3D contour and surface models. *IEEE Transactions on Medical Imaging*.

Rashid, T., Sultana, S., & Audette, M. A. (2016). Watertight and 2-manifold surface meshes using Dual Contouring with tetrahedral decomposition of grid cubes. *Procedia Engineering*, 163, 136-148.

Sultana, Sharmin, Praful Agrawal, Shireen Y. Elhabian, Ross T. Whitaker, **Tanweer Rashid**, Jason E. Blatt, Justin S. Cetas, and Michel A. Audette. "Towards a statistical shape-aware deformable contour model for cranial nerve identification." In *Workshop on Clinical Image-Based Procedures*, pp. 68-76. Springer International Publishing, 2016.

Audette, M. A., Haq, R., **Rashid, T.**, & Sultana, S. (2015, April). Simplex-based surface and contour models for segmentation and meshing for surgery simulation. In *Proceedings of the Symposium on Modeling and Simulation in Medicine* (pp. 9-15). Society for Computer Simulation International.

Patel, N., Sultana, S., **Rashid, T.**, Krusienski, D., & Audette, M. A. (2015, March). Application and histology-driven refinement of active contour models to functional region and nerve delineation: towards a digital brainstem atlas. In *SPIE Medical Imaging* (pp. 94150I-94150I). International Society for Optics and Photonics.