# Characterizing Retinotopic Mapping Using Conformal Geometry <br> and Beltrami Coefficient: a Preliminary Study <br> by <br> Duyan Ta 

A Thesis Presented in Partial Fulfillment of the Requirement for the Degree

Master of Science

Approved November 2013 by the Graduate Supervisory Committee:

Yalin Wang, Chair
Ross Maciejewski Peter Wonka


#### Abstract

Functional magnetic resonance imaging (fMRI) has been widely used to measure the retinotopic organization of early visual cortex in the human brain. Previous studies have identified multiple visual field maps (VFMs) based on statistical analysis of fMRI signals, but the resulting geometry has not been fully characterized with mathematical models. This thesis explores using concepts from computational conformal geometry to create a custom software framework for examining and generating quantitative mathematical models for characterizing the geometry of early visual areas in the human brain. The software framework includes a graphical user interface built on top of a selected core conformal flattening algorithm and various software tools compiled specifically for processing and examining retinotopic data. Three conformal flattening algorithms were implemented and evaluated for speed and how well they preserve the conformal metric. All three algorithms performed well in preserving the conformal metric but the speed and stability of the algorithms varied. The software framework performed correctly on actual retinotopic data collected using the standard travellingwave experiment. Preliminary analysis of the Beltrami coefficient for the early data set shows that selected regions of V1 that contain reasonably smooth eccentricity and polar angle gradients do show significant local conformality, warranting further investigation of this approach for analysis of early and higher visual cortex.


## DEDICATIONS

To my parents, Tai and Linda, for their unconditional love and support. To my grandparents who have always emphasized and encouraged the pursuit of higher education and knowledge. I hope that I have made them proud.

## ACKNOWLEDGEMENTS

This thesis would not have been possible without the direct and indirect help of many wonderful people. First, I would like to thank my committee chair and advisor, Professor Yalin Wang. His enthusiasm, dedication, and passion for research was inspiring and influential throughout my graduate studies. I am forever grateful for his help and guidance. Second, I would like to thank Professor Ross Maciejewski and Professor Peter Wonka for serving on my thesis committee. Third, I would like to thank our collaborators, Professor Alyssa Brewer and Professor Zhong-Lin Lu, for taking time out of their busy schedule to answer our questions and review our writings and results. Fourth, I would like to especially thank Professor Alyssa Brewer and her graduate student, Brian Barton, for providing our research group with timely and valuable fMRI retinotopic data. Without their data, the research conducted in this thesis would not have been possible. Fifth, I would like to thank my fellow lab members Jie Shi, Liang Xu, and Xing Gao for showing me how to process the research data and answering my long list of questions. Special thanks to Jie for showing me how to process retinotopic mapping data with her custom software tools. Finally, I would like to send a special thank you to all the people who have made a positive difference in my life. I would not be who I am today without them. They include my parents, grandparents, uncles, aunts, cousins, teachers, and friends.

## TABLE OF CONTENTS

Page
LIST OF TABLES ..... vi
LIST OF FIGURES ..... viii
CHAPTER
1 INTRODUCTION ..... 1
1.1 Motivation and Related Works ..... 4
1.2 Contribution ..... 6
1.3 Thesis Direction and Layout ..... 7
2 METHOD ..... 9
2.1 Software Tool Requirements ..... 16
2.2 Conformal Flattening Algorithms ..... 16
2.2.1 Non-Linear Heat Diffusion Spherical Harmonic Mapping ..... 17
2.2.2 Harmonic and Holomorphic 1-Form ..... 20
2.2.3 Surface Discrete Ricci Flow ..... 22
2.3 Supplemental Topics ..... 28
2.3.1 Differential Geometry ..... 29
2.3.2 Exterior Differential Calculus ..... 39
2.3.3 Algebraic Topology ..... 40
2.3.4 Half-Edge Data Structure ..... 44
3 RESULTS ..... 46
4 DISCUSSION ..... 81
4.1 Test Data ..... 81
4.2 Retinotopic Data ..... 81
4.3 Beltrami Coefficient analysis ..... 82
5 CONCLUSION ..... 83
5.1 Future Work ..... 83
REFERENCES ..... 84
APPENDIX
A PROCESSING DETAILS AND DATA ..... 88

## LIST OF TABLES

Table ..... Page
3.1 Algorithm Results. ..... 47
3.2 Non-Linear Heat Diffusion Algorithm Results ..... 47
3.3 Non-Linear Heat Diffusion Test Data. ..... 47
3.4 Harmonic and Holomorphic 1-form Algorithm Results ..... 48
3.5 Harmonic and Holomorphic 1-form Test Data. ..... 48
3.6 Discrete Ricci Flow Algorithm Results. ..... 48
3.7 Discrete Ricci Flow Test Data. ..... 49
3.8 Retinotopic Non-Linear Heat Diffusion Algorithm Results. ..... 49
3.9 Retinotopic Discrete Ricci Flow Algorithm Results. ..... 50
A. 1 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (1-16) ..... 88
A. 2 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (17-32) ..... 89
A. 3 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (33-48) ..... 90
A. 4 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (49-64) ..... 91
A. 5 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (65-80) ..... 92
A. 6 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (81-96) ..... 93
A. 7 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (97-112) ..... 94
A. 8 Color Map Used for Retinotpic Data. 256 RGB colors converted toHSV and Sorted. Colors (113-128)95
A. 9 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (129-144) ..... 96
A. 10 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (145-160) ..... 97
A. 11 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (161-176) ..... 98
A. 12 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (177-192) ..... 99
A. 13 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (193-208) ..... 100
A. 14 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (209-224) ..... 101
A. 15 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (225-240) ..... 102
A. 16 Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (241-256) ..... 103

## LIST OF FIGURES

Figure Page
2.1 Typical Processing Pipeline for Retinotopic Mapping ..... 9
2.2 Applications in Computer Graphics. [12] [42] ..... 11
2.3 Applications in Medical Imaging. [37] [38] [39] [39] [40] [36] ..... 11
2.4 Standard Travelling-wave Diagram for Eccentricity. ..... 12
2.5 Standard Travelling-wave Diagram for Polar Angle. ..... 12
2.6 Decoding Eccentricity and Polar Angle from Color Map. ..... 13
2.7 Plot of Recovered Parameterization ..... 14
2.8 Retinotopic Surface Matching Using Conformal Mapping. ..... 15
2.9 Flattening Using Non-Linear Heat Diffusion Spherical Method. ..... 16
2.10 Flattening Using Harmonic and Holomorphic 1-Form Method. ..... 16
2.11 Flattening Using Discrete Ricci Flow Method ..... 17
2.12 Functional Diagram for the Retinotopic Processing Software ..... 18
2.13 Conformal Mapping of Brain to Sphere ..... 20
2.14 Visualization of Harmonic and Holomorphic 1-Form Using Texture Mapping. ..... 23
2.15 Holomorphic 1-Form of Visual Region Cortical Mesh ..... 23
2.16 Discrete Ricci Flow Output ..... 26
2.17 Circle Packing Weight Computation ..... 27
2.18 General Parameterized Curves and Surfaces. ..... 29
2.19 Normal Curvature at Curve $\gamma$. ..... 35
2.20 Canonical Fundamental Group Basis ..... 43
2.21 Half Edge Data Structure ..... 45
3.1 Histogram A.1a: Sphere Test Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=2502$. Mean $=-0.0064$ ..... 51

## Figure

3.2 Histogram A.1b: Sphere Test Non-Linear Heat Diffusion Histogram
(Real). Vertices Count $=2502$. Mean $=0.0014$ ..... 52
3.3 Histogram A.2a: Sphere Test Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=10002$. Mean $=0.0027 \ldots \ldots \ldots \ldots \ldots$.
3.4 Histogram A.2b: Sphere Test Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=10002$. Mean $=0.0068$
3.5 Histogram B.1a: Face Test Discrete Ricci Flow Histogram (Complex). Vertices Count $=5087$. Mean $=0.0345 \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots . . \ldots . . \ldots$
3.6 Histogram B.1b: Face Test Discrete Ricci Flow Histogram (Real). Ver- tices Count $=5087$. Mean $=-0.0322$ ..... 56
3.7 Histogram B.2a: Face Test Discrete Ricci Flow Histogram (Complex). Vertices Count $=$ 20135. Mean $=0.0091$ ..... 57
3.8 Histogram B.2b: Face Test Discrete Ricci Flow Histogram (Real). Ver- tices Count $=20135$. Mean $=-0.0010$ ..... 58
3.9 Histogram B.3a: Face Test Discrete Ricci Flow Histogram (Complex). Vertices Count $=80114$. Mean $=0.0083$ ..... 59
3.10 Histogram B.3b: Face Test Discrete Ricci Flow Histogram (Real). Ver-tices Count $=80114$. Mean $=-0.0094$60
3.11 Histogram C.1a: AABS1VFM Left Non-Linear Heat Diffusion His- togram (Complex). Vertices Count $=22817$. Mean $=0.0053$ ..... 61
3.12 Histogram C.1b: AABS1VFM Left Non-Linear Heat Diffusion His- togram (Real). Vertices Count $=22817$. Mean $=-0.0229$ ..... 623.13 Histogram C.2a: LLS1PP Left 3deg Non-Linear Heat Diffusion His-togram (Complex). Vertices Count $=25586$. Mean $=0.0030 \ldots \ldots \ldots .63$
3.14 Histogram C.2b: LLS1PP Left 3deg Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=25586$. Mean $=-0.0222 \ldots \ldots \ldots \ldots$.
3.15 Histogram C.3a: SADS3VFM Leftt 3deg Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=16523$. Mean $=0.0024 \ldots \ldots 65$
3.16 Histogram C.3b: SADS3VFM Leftt 3deg Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=16523$. Mean $=-0.0159 \ldots \ldots \ldots .66$
3.17 Histogram C.4a: SADS3VFM Right 3deg Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=13525$. Mean $=-0.0016 \ldots \ldots .67$
3.18 Histogram C.4b: SADS3VFM Right 3deg Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=13525$. Mean $=-0.0131 \ldots \ldots \ldots .68$
3.19 Histogram D.1a: BBCS3PP Right 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=16322$. Mean $=0.0344 \ldots \ldots \ldots \ldots \ldots .$.
3.20 Histogram D.1b: BBCS3PP Right 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=16322$. Mean $=-0.0406$70
3.21 Histogram D.2a: LLS1PP Right 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=21075$. Mean $=0.0291 \ldots \ldots \ldots \ldots \ldots .71$
3.22 Histogram D.2b: LLS1PP Right 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=21075$. Mean $=-0.0386$
3.23 Histogram D.3a: SADS3VFM Leftt 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=16523$. Mean $=0.0320 \ldots \ldots . .73$
3.24 Histogram D.3b: SADS3VFM Leftt 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=16523$. Mean $=-0.0375 \ldots \ldots \ldots \ldots .74$
3.25 Histogram D.4a: SADS3VFM Right 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=13525$. Mean $=0.0267 \ldots \ldots \ldots .75$

Figure
3.26 Histogram D.4b: SADS3VFM Right 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=13525$. Mean $=-0.0353 \ldots \ldots \ldots \ldots .76$
3.27 Histogram E.1a: LLS1PP Left 3deg Parameterization Histogram V1 (Complex). Vertices Count $=1238$. Mean $=0.0587$
3.28 Histogram E.1b: LLS1PP Left 3deg Parameterization Histogram V1 (Real). Vertices Count $=1238$. Mean $=0.2413$
3.29 Histogram F.1a: SADS3VFM Left 3deg Parameterization Histogram V1 (Complex). Vertices Count $=887$. Mean $=0.0358$79
3.30 Histogram F.1b: SADS3VFM Left 3deg Parameterization Histogram V1 (Real). Vertices Count $=887$. Mean $=0.1983$80

## Chapter 1

## INTRODUCTION

In modern medicine, medical imaging is essential and vital for the proper and early diagnosis of medical conditions. The World Health Organization's (WHO) website describes medical imaging as follows:

Medical imaging comprises different imaging modalities and processes to image human body for diagnostic and treatment purposes and therefore has an important role in the improvement of public health in all population groups. Furthermore, medical imaging is justified also to follow the course of a disease already diagnosed and/or treated. Area of medical imaging is very complex and, depending on a context, requires supplementary activities of medical doctors, medical physicists, biomedical engineers as well as technicians. Medical imaging, especially X-ray based examinations and ultrasonography, is crucial in every medical setting and at all levels of heath care. In public health and preventive medicine as well as in curative medicine, effective decisions depend on correct diagnosis. Though medical/clinical judgment maybe sufficient in treatment of many conditions, the use of diagnostic imaging services is paramount in confirming, correctly assessing and documenting course of the disease as well as in assessing response to treatment. [41]

Modern medical imaging research is focussing on using computer science and computational conformal geometry to augment and enhance traditional imaging data in order to provide highly detailed three dimensional representations that can be used to more accurately visualize and detect the minor differences between normal and the very early stages of disease progression. Novel ideas come from a wide variety of research areas including but not limited to computer science,
computer engineering, mathematics, and physics. Imaging modalities are combined with image processing techniques and computational conformal geometry algorithms to generate highly accurate and detailed anatomical data that can be used for cross subject comparisons to study the early signs of diseases. Research from computer science and mathematics have provided faster and better imaging algorithms while hardware researchers have been able to continuously provide faster and more efficient computing power. This combination of hardware and software algorithms is making it possible for researchers to gather large amounts of quality imaging data faster which translates to more opportunities for discoveries and progress in medicine and biology.

Medical imaging is not only a diagnosis tool for doctors but also a valuable research tool for researchers in biology, bioengineering, and medicine. The noninvasive nature of medical imaging allows researchers from biology and medicine to collect large amounts of in vivo research data which is necessary in order to study and fully understand biological processes. Researchers have been able to confirm known theories and also discover previously unknown facts about how the human body works using the available in vivo data. One example research topic that requires in vivo data is human brain mapping. Functional magnetic resonance imaging (fMRI) technology and many new computational geometry algorithms have profoundly change our understanding of how the brain works.

A relatively new research area that has greatly impacted the medical imaging field is computational conformal geometry. In Computational Conformal Geometry [12], computational conformal geometry is defined as, "an interdisciplinary field, combining modern geometry theories from pure mathematics with computational algorithms from computer science." Fig. 2.2 provides some examples of the various applications of computational conformal geometry.

Applications for computational conformal geometry are found in many areas because the conformal structure is both flexible and restrictive. "Roughly speaking, conformal structure is more rigid than topological structure and more flexible than Riemannian metric. Conformal geometry is between topology and Riemannian geometry." [12]. An angle preserving map is restrictive enough that dissimilar surfaces will not generate the same mapping while allowing more flexibility than an isometric mapping. Fig. 2.3 provides some examples of modern medical imaging research that uses conformal geometry. Computational conformal geometry has provided many new tools for solving a variety of geometric problems. Tasks of identifying and matching shapes that were once thought to be capable only by a human are now possible with computers. These tasks are grouped into what is known as high level shape operators. They include classification, comparison, matching, and recognition. The algorithms for performing these tasks with efficiency and high precision rely on the following powerful concept from computational conformal geometry:

All surfaces in real life can be deformed to three canonical shapes: the sphere, the plane, and the disk. The deformation preserves angles and is determined by a small number of control parameters, such as several landmarks. Therefore, all geometric problems in three dimensional Euclidian space $\mathbb{R}^{3}$ can be converted to two dimensional problems on the plane. [12]

This concept is surprising at first because there are so many possible surfaces that we can identify. However, as stated above, there are only three canonical shapes that all surfaces can deform to. By deforming surfaces to their canonical domain, the matching or categorizing problem is greatly simplified.

Computational conformal geometry is currently being used by researchers to solve a variety of problems in medical imaging. Applications can also be found in robotics vision, intelligent systems, and network routing. There are many biological problems that are now being investigated due to the availability of high quality
medical imaging data. One of these is finding a quantitative model that can fully characterize the geometry of the human visual areas. Human retinotopic organization has been studied using functional magnetic resonance imaging (fMRI) in [8] and [34]. Functional magnetic resonance imaging (fMRI) uses what is known as the blood-oxygen-level dependent (BOLD) contrast [14]. It measures the cerebral blood flow to neurons when they are activated instead of directly measuring the electrical activity. This indirect measurement technique works because neuronal activation is highly dependent on the amount of blood flowing to the activation area. Studies so far have not yielded a quantitative model that can fully characterize the human retinotopic mapping. This thesis is looking to explore this problem using computational conformal geometry algorithms and the Beltrami coefficient. The goal is to study whether the application of computational conformal geometry algorithms and Beltrami coefficient to retinotpic mapping data will yield any new insights into finding a fully quantitative model.

### 1.1 Motivation and Related Works

The efficiency of the measurements and the relatively large amplitude of functional magnetic resonance imaging (fMRI) signals in visual cortical areas have made it possible to develop quantitative models of functional responses within specific maps in individual subjects. Retinotopic mapping of human visual cortex generates visual field maps by analyzing the stimulus-referred fMRI response to each of the fragments in each voxel $[7,9,10,13,29,33]$. The maps elucidate the spatial organization of the neuronal responses to visual images on the retina $[7,9,10,25,29,31]$ and have contributed greatly to our understanding of the human visual system and the development of human cerebral cortex [35]. They also hold great promises to further our understanding of plasticity in the human visual cortex in normal and abnormal populations.

Although numerous studies have been devoted to retinotopic mapping, most of them took an experimental approach to discover various visual areas and study the relationship between them. Missing at this time is a mathematical model that fully considers the intrinsic geometrical features of the underlying cortical structures. Instead most studies have focused on 2D mappings but lots of distortions have already been introduced when the 3D cortical surface is flattened to the 2D. Ju's paper [17] found variations in run time and metric preservation across the three evaluated conformal flattening methods. Therefore it is important to note that not all conformal flattening algorithms produce the same results. More recently Balasubramanian and Schwartz have published a paper showing that near isometric flattening of brain surfaces reveals that the organization of the visual areas of the brain are much more alike across subjects than has been shown before. [4] A typical retinotopy is usually generated in three steps: (1) Flatten the cortical surfaces using structural scans; (2) Project the functional data onto the flattened surfaces; (3) Generate a phase map of the retina image on the flattened surface based on the visual stimuli on the retina. However, there are a number of issues: (1) Large distortions are usually introduced in the cortical flattening process; (2) Although the current method generates maps, there is no concrete mathematical description of these maps and no direct way to quantitatively compare the maps. These difficulties made retinotopic mapping mainly an experimental study in which experimental results obtained in small samples pose significant challenges for a population level integration and analysis. Because of the lack of a theoretical model, research on retinotopic mapping is strongly limited by available experimental protocols. For example, some large veins close to fovea in many subjects significantly diminish the fMRI response accuracy and distort the retinotopic map. This problem is alleviated only recently with high resolution fMRI and optimized methods [23].

Schwartz [26] proposed an analytical expression which is a conformal mapping to describe the retinotopic map in V1. The simple and convenient complex-log transform (or variants thereof [3, 21, 27]) has become the de facto standard for describing the shape of human V1. Over the years, a variety of new models have been proposed aiming to solve some counterintuitive predictions in Schwartz's model (e.g. [24]). An important feature of this kind of approach is to apply multidimensional scaling method (MDS) $[2,5,18,19,28,30,32]$ to compute isometric (or near-isometric) mapping from the original brain surface to the Euclidean plane. However, the drawback of MDS is that it does not consider any surface geometric features and the results are only some approximation to the isometric parameterization. Inevitably, the flattening procedure introduces lots of distortions that make the subsequent analysis inaccurate. In [22], Qiu et al. computed a hyperbolic conformal map of visual V1 area using the circle packing method $[15,16]$. However, the mapping was only used for visualization and no quantitative models were developed to describe and compare the retinotopic maps.

### 1.2 Contribution

This thesis explores using concepts from computational conformal geometry to create a custom software framework for examining and generating quantitative mathematical models for characterizing the geometry of early visual areas in the human brain. The goal for the software is to automate tedious processing tasks and minimize human error for data processing tasks. The software framework includes a graphical user interface built on top of a selected set of known conformal flattening algorithm and various software tools compiled specifically for processing and examining retinotopic data. Additional software tools created for retinotopic mapping are also combined along with several other mesh processing tools to form a complete software package for future research. Conformal flattening algorithms are
implemented and studied in detail to ensure that the needs of retinotopic mapping research are met. The conformal metric is evaluated by generating histogram plots of the computed Beltrami coefficient between the original and the deformed mesh. The Beltrami coefficient is explored as a possible quantitative measurement that can be used to characterize the geometry of retinotopic organization in the visual areas. Using the Beltrami coefficient in this way has not been done before. A preliminary quantitative model using the Beltrami coefficient to describe the retinotopic mapping is proposed.

### 1.3 Thesis Direction and Layout

This thesis will first discuss the implementation and evaluation of three computational conformal geometry mapping algorithms: non-linear heat diffusion spherical harmonic, harmonic and holomorphic 1-form, and discrete Ricci flow. These three algorithms are evaluated to study how flattening affects retinotopic data and if flattened data is more helpful in identifying similarities across subjects than non-flattened data. Second, the software requirements for the retinotopic mapping tool will be presented and discussed. The software tool was created to facilitate with data processing. Lastly, the results from processing real retinotopic research data using the software tool will be discussed along with conclusions drawn from the data. It is important to note that characterizing human retinotopic mapping using fMRI data is still an open problem. As a result, it is not the goal of this paper to fully derive and describe a complete quantitative model that can fully characterize human retinotopic mapping. Instead, the goal is to approach the problem by developing a software package around some well known concepts from computational conformal geometry in order to describe the geometry of the early visual areas. Once that has been achieved, then finding a complete quantitative model would be more plausible.

The layout of this thesis is structured like a science research paper. Chapter 2 is the methods section. It discusses all the computational conformal geometry algorithms used and their code implementation. It also briefly highlights the mathematical theory for these algorithms. Chapter 3 is the results section. The performance and accuracy test results for these algorithms are presented in graphs and tabular form. Chapter 4 is the discussion section. In depth analysis of the data and plots from Chapter 3 are discussed and additional insights into the behavior of these algorithms are also discussed. Chapter 5 is the conclusion section. Everything is briefly reviewed and the main results are highlighted. Recommendations are provided and future works are discussed.

## Chapter 2

## METHOD

The retinotopic research data were collected and provided to our group by Professor Alyssa Brewer and her student Brian Barton from UCI Laboratory of Visual Neuroscience. The retinotopic data were collected using the standard travelling-wave method as described in [8] and [34]. Fig. 2.4 and 2.5 depicts the standard-travelling wave method. The human subject is provided visual stimuli while fMRI data is collected. Visual stimuli consists of black and white, drifting checkerboards comprising rotating wedges and expanding rings to measure the cortical representations of polar angle and eccentricity, respectively. These representations were then projected onto a three-dimensional cortical mesh of each hemisphere. Collected retinotopic data consists of three-dimensional cortical mesh with vertex color data attribute corresponding to eccentricity and polar angle. Eccentricity and polar angle values can be retrieved from the data set using the color map selected for data collection. Fig. 2.6 illustrates the decoding of the vertex color data to retrieve the radius r and angle $\theta$. The $u, v$ coordinates can then be computed as $u=r \cos (\theta)$ and $v=r \sin (\theta)$. Fig. 2.7 shows what a plot of the recovered parameterization looks like.

The steps for processing the research data and applying the conformal geometry algorithms are summarized in fig. 2.1.


Figure 2.1: Typical Processing Pipeline for Retinotopic Mapping

The initial step is to cut the visual regions, located on the occipital lobes, from the the rest of the brain. After the cut, the original closed genus zero surface is divided into two open genus zero surfaces. Only the genus zero surface containing the visual
regions are retained for conformal flattening and evaluation. A diagram describing how the conformal mapping surface matching idea is applied to retinotopic mapping is shown in fig. 2.8. Flattening a three-dimensional surface with a conformal parameterization changes a three-dimensional matching problem to a two-dimensional one. The flattening step is where we apply three different conformal geometry algorithms to see which one preserves the metric best.

The discrete Ricci flow and harmonic/holomorphic 1-form algorithms are known to be more robust and stable than the non-linear heat diffusion algorithm because their conformal solutions are obtained by solving several linear systems while the non-linear heat diffusion method uses the gradient descent method which is heavily dependent on step size selection and the initial condition choice. The efficiency of discrete Ricci flow and harmonic/holomorphic 1-form algorithms are dependent on the quality of the linear solver used. Matlab linear solver was used as the backend for my research implementation of these algorithms. Efficiency comparison of additional linear solvers can be found in Polygon Mesh Processing [6]. Another difference is the non-linear heat diffusion method requires the double covering step to convert the open boundary genus-zero surface to a closed genus-zero surface in order to apply the algorithm. On the other hand, the holomorphic 1-form and the discrete Ricci flow method do not require this step because these versions of the algorithms work directly with open boundary surfaces. Fig. 2.9, Fig. 2.10, and Fig. 2.11 show the processing pipeline for each of the three algorithms.


Figure 2.2: Applications in Computer Graphics. [12] [42]


Figure 2.3: Applications in Medical Imaging. [37] [38] [39] [39] [40] [36]


Figure 2.4: Standard Travelling-wave Diagram for Eccentricity.


Figure 2.5: Standard Travelling-wave Diagram for Polar Angle.


Figure 2.6: Decoding Eccentricity and Polar Angle from Color Map.


Figure 2.7: Plot of Recovered Parameterization.


Figure 2.8: Retinotopic Surface Matching Using Conformal Mapping.


Figure 2.9: Flattening Using Non-Linear Heat Diffusion Spherical Method.


Figure 2.10: Flattening Using Harmonic and Holomorphic 1-Form Method.

### 2.1 Software Tool Requirements

A graphical user interface (GUI) was implemented on top of the conformal flattening algorithms to facilitate with retinotopic data processing. The goal for making the GUI is to simplify and minimize errors during data processing. Creation of folders and files are automated to avoid naming collisions and provide naming consistency. Once files are generated they can be viewed readily using any available mesh viewer tool. Fig. 2.12 diagrams the required software functions.

### 2.2 Conformal Flattening Algorithms

Conformal flattening algorithms used in freely available software packages were evaluated by Ju [17]. The paper [17] evaluated FreeSurfer, CirclePack, and LSCM and found variations in metric preservation across all three tools. The algorithms


Figure 2.11: Flattening Using Discrete Ricci Flow Method.
used by each of the software package all had the same goal of flattening while preserving certain metrics. Ju's paper [17] found significant variations in this goal across the tools. As a result, we will implement three known algorithms ourselves and specifically evaluate their pros and cons with respect to flattening retinotopic data.

### 2.2.1 Non-Linear Heat Diffusion Spherical Harmonic Mapping

All closed genus zero surfaces can be mapped conformally onto the sphere. This algorithm uses the nonlinear heat diffusion equation

$$
\frac{\mathrm{d} f(t)}{\mathrm{d} t}=-\Delta f(t)
$$

Each step of the algorithm only requires the computation of the normal and tangential component of the Laplacian. This algorithm solves the harmonic map function by slowly converging towards a specified minimum harmonic energy. The


Figure 2.12: Functional Diagram for the Retinotopic Processing Software
solutions are not unique and differ by a Möbius transformation. A Möbius transformation is a mapping where the following is true

$$
\phi: z \rightarrow \frac{a z+b}{c z+d}, \quad a, b, c, d \in \mathbb{C}, \quad a d-b c=1
$$

It is important to note that this algorithm works because for a sphere, "a map is conformal if and only if it is harmonic." [12] This is why the goal of this algorithm is to iteratively minimize the harmonic energy. It is also important to note that in general, it is not the case that when a mapping is harmonic that it is also conformal. The full algorithm from [12] is as follows:

Algorithm 17: Spherical conformal mapping input: Mesh M, step length $\delta t$, energy difference threshold $\delta E$ output: A harmonic map $\mathbf{f}: M \rightarrow \mathbb{S}^{2}$, which satisfies the zero mass-center
constraint.
Compute a degree one map, such as Gauss map $\mathbf{g}: M \rightarrow \mathbb{S}^{2}$
Initialize $\mathbf{f} \leftarrow \mathbf{g}$, compute harmonic energy $E_{0}$;
repeat
forall vertex $v \in M$ do
Compute the Laplacian $\Delta \mathbf{f}$; Compute the normal component;

$$
\Delta \mathbf{f}^{\perp}=<\Delta \mathbf{f}, \mathbf{f}>\mathbf{f}
$$

Compute the tangential component;

$$
\Delta \mathbf{f}^{\|}=\Delta \mathbf{f}-\Delta \mathbf{f}^{\perp}
$$

Update $\mathbf{f}(v)$ by

$$
\mathbf{f}(v)=\mathbf{f}(v)-\delta t \times \Delta \mathbf{f}^{\|}
$$

end
Compute Möbius transformation $\varphi: \mathbb{S}^{2} \rightarrow \mathbb{S}^{2}$, such that the mass center of $\varphi \circ \mathbf{f}$ is the sphere center;
$\mathbf{f} \leftarrow \varphi \circ \mathbf{f} ;$
$E_{0} \leftarrow E$;
Compute the harmonic energy $E(\mathbf{f})$.
until Harmonic energy difference $\left|E-E_{0}\right|$ is less than $\delta E$;
Return $\mathbf{f}$ [12]
The normalization step in the algorithm above is computationally expensive and requires an optimization for practical usage. The full algorithm from [12] to approximate the Möbius normalization step is:

Algorithm 18: Normalization
input: Mesh M, a mapping to the sphere $\mathbf{f}: M \rightarrow \mathbb{S}^{2}$
output : Normalized mapping $\tilde{\mathbf{f}}$, whose mass center is at the sphere center
Compute the the mass center of $\mathbf{f}$ :

$$
\mathbf{c} \leftarrow \int_{\mathbb{S}^{2}} \mathbf{f} \mathrm{~d} \sigma .
$$

where $\mathrm{d} \sigma$ is the area element on the original mesh M.
forall vertex $v \in M$ do

$$
\tilde{\mathbf{f}} \leftarrow \mathbf{f}(v)-\mathbf{c}
$$

end

$$
\begin{aligned}
& \text { forall vertex } v \in M \text { do } \\
& \qquad \tilde{\mathbf{f}} \leftarrow \frac{\tilde{\mathbf{f}}}{|\tilde{f}(v)|} . \\
& \text { end }[12]
\end{aligned}
$$

This conformal mapping method requires careful selection of the step length $\delta t$ in order for the energy to decrease and converge. This gradient descent method is not always guaranteed to converge. The stability of the gradient descent method depends on the step length choice and the initial map. The initial map quality will affect the stability and speed of the convergence. The Gauss map is a good initial choice but there are others. If a better initial map can be found that makes this method more stable and converges faster, then this method may become a competitive choice. Fig. 2.13 shows the spherical mapping output of the test brain mesh.


Figure 2.13: Conformal Mapping of Brain to Sphere

### 2.2.2 Harmonic and Holomorphic 1-Form

This algorithm computes the harmonic and holomorphic forms for a multi-holed annulus. The general steps for a conformal mapping of a multi-holed annuli as described in [12] has been reduced to only two boundaries for our research need here. The surface mesh cut that contains the visual region can be conformally
mapped to a disk by puncturing an arbitrary hole and applying the algorithm for conformally mapping a multi-holed annuli. It is important that the punctured hole not include any of the visual region of interest (V1 and V2). These regions can be traced and cut out from the unit disk after the mapping is complete. The cut region will be conformal to the original pre-flattened mesh of that region. The full algorithm described on page 241 in [12] has been modified for a single internal boundary as follows.

Algorithm: Conformal Mapping for Multi-Holed Annuli input: Genus Zero Annulus with a Single Internal Boundary mesh M output : A conformal map to a disk $\mathbf{f}: M \rightarrow \mathbb{S}^{2}$

The set of boundaries of the annulus is $\partial M=\gamma_{0}-\gamma_{1}$.
The outer most boundary of the annulus is taken to be $\gamma_{0}$.
NOTE: An annulus is an outer large boundary encircling smaller circles.

1. Compute the harmonic measures of $M$ by solving the following Dirichlet problem,

$$
\delta f_{i} \equiv 0,\left.\quad f_{i}\right|_{\gamma_{j}}=\delta_{i j}= \begin{cases}1, & i=j \\ 0, & i \neq j\end{cases}
$$

The exact harmonic 1-form is

$$
\omega_{0}=\mathrm{d} f .
$$

2. Find shortest path connecting inner boundary and outer boundary $\tau$. Generate open mesh $\bar{M}$ by slicing mesh M along $\tau$.
The cut generates two boundaries along $\tau, \tau^{+}$and $\tau^{-}$.
3. Compute harmonic function $g: \bar{M} \rightarrow \mathbb{R}$ such that
$g\left|\tau^{+}=1, g\right| \tau^{-}=0$.
Denote each edge in $\tau^{+}$as $e^{+}$and $\tau^{-}$as $e^{-}$. The differential $\mathrm{d} g\left(e^{+}\right)=\mathrm{d} g\left(e^{-}\right)=0$, so the differential is well defined on $M$. Let

$$
\tau=d g, \quad \int_{\gamma_{0}} \tau=1
$$

Diffuse to find $\omega_{1}$ by solving the following

$$
\sum_{j} w_{i j}\left[\tau\left(\left[v_{i}, v_{j}\right]\right)+h\left(v_{j}\right)-h\left(v_{i}\right)\right]=0
$$

Then $\omega_{1}=\tau+\mathrm{d} h$.
5. Compute holomorphic 1-form by solving the following linear system

$$
\left(\begin{array}{cc}
\int_{M} \omega_{0} \wedge * \omega_{0} & \int_{M} \omega_{0} \wedge * \omega_{1} \\
\int_{M} \omega_{1} \wedge * \omega_{0} & \int_{M} \omega_{1} \wedge * \omega_{1}
\end{array}\right)=\left(\begin{array}{cc}
\int_{M} \omega_{0} \wedge \omega_{0} & \int_{M} \omega_{0} \wedge \omega_{1} \\
\int_{M} \omega_{1} \wedge \omega_{0} & \int_{M} \omega_{1} \wedge \omega_{1}
\end{array}\right)\left(\begin{array}{ll}
\lambda_{00} & \lambda_{10} \\
\lambda_{01} & \lambda_{11}
\end{array}\right)
$$

Integrate to obtain mapping $\phi: \bar{M} \rightarrow C$
Choose any vertex $v_{0}$ as the initial root vertex for integration.

$$
\phi\left(v_{i}\right)=\int_{v_{0}}^{v_{i}} \omega_{1}+\sqrt{-1} * \omega_{1}
$$

Visualization of the holomorphic 1-form using checkboard pattern is shown in fig. 2.14. The top left and middle picture shows the triangle punched hole and the sliced path connecting the inner boundary and outer boundary. The top right picture shows the applied checkerboard texture map on the mesh using the holomorphic 1-form as $(u, v)$ coordinates. The bottom row shows a different holomorphic 1-form $\omega=\sum_{i=1}^{n} \lambda_{i}\left(\psi_{i}+\sqrt{-1} * \psi_{i}\right)$ with a different constant value $\lambda$ introduced to induce a disk mapping. The bottorm row of three images also shows a zooming progression towards the tip of the nose. The texture pattern shows up only when you are close enough to the tip. An example of holomorphic 1-form applied on the visual cortical mesh is shown in fig. 2.15. The runtime to obtain this is less than one minute compared to about forty to fifty hours using the non-linear spherical harmonic method.

### 2.2.3 Surface Discrete Ricci Flow

This algorithm computes a conformal mapping for a multi-holed annulus using discrete Ricci flow. This algorithm uses the Newton's method for gradient descent to optimize the convex energy. This is similar to the non-linear heat diffusion method described above which also uses gradient descent to optimize the harmonic


Figure 2.14: Visualization of Harmonic and Holomorphic 1-Form Using Texture Mapping.


Figure 2.15: Holomorphic 1-Form of Visual Region Cortical Mesh
energy. The full algorithm for generating a conformal mapping of a multi-holed annuli using discrete Ricci flow from [12] is shown below. Once again we will only be using this algorithm with annuli that have a single internal boundary. We will puncture an arbitrary hole on a genus zero open surface to create the internal boundary. Then we will apply the algorithm for conformally mapping a multi-holed annuli. It is important that the punctured hole not include any of the visual region of interest (V1 and V2). These regions can be traced and cut out from the unit disk after the mapping is complete. The cut region will be conformal to the original pre-flattened mesh of that region.

Algorithm 43: Newton's method of discrete Ricci flow for multi-holed annulus
input : A multi-holed annulus M
output : A flat circle packing metric $(M, \Gamma, \Phi)$ which maps the boundaries to circles

Compute the initial circle packing metric $\left(M, \Gamma_{0}, \phi\right)$;
Compute the initial curvature $K$.
Trace boundaries, store each boundary as a list of ordered vertices

$$
\Gamma_{1}, \Gamma_{2}, \ldots, \Gamma_{m}
$$

Set $\Gamma_{1}$ as the outer boundary,

$$
\theta_{1} \leftarrow 2 \pi, \theta_{2} \leftarrow-2 \pi, \ldots, \theta_{m} \leftarrow-2 \pi .
$$

forall Interior Vertex $v \in M$ do

$$
\bar{K}(v) \leftarrow 0
$$

end
forall Boundaries $\Gamma_{k}$ do
forall $v \in \Gamma_{i}$ do

$$
\bar{k}(v) \leftarrow \frac{\theta_{k}}{\left|\Gamma_{k}\right|}
$$

end

## end

repeat
Compute target metric according to $\bar{K}$ using Newton's method in algorithm 42;
forall Boundaries $\Gamma_{k}$ do end forall $v_{i} \in \Gamma_{i}$ do

$$
\bar{k}(v) \leftarrow \theta_{k} \frac{l_{i-1, i}+l_{i, i+1}}{2 s_{k}}
$$

end
until The maximal difference of current target curvature and previous target curvature is less than $\varepsilon$; [12]

The total target curvature is set to $2 \pi$ for the outer boundary and $-2 \pi$ for the inner boundaries. Algorithm 43 calls algorithm 42 to compute the target curvature over and over again until the difference between the current target curvature and the previous target curvature is less than a certain threshold $\varepsilon$. The full algorithm from [12] that computes the target curvature is shown below.

```
Algorithm 42: Newton's method of discrete Ricci flow
input: A mesh M embedded in \(\mathbb{R}^{3}\), target curvature \(\bar{K}\), curvature
threshold \(\varepsilon\)
output: A circle packing metric \((M, \Gamma, \Phi)\) which induces \(\bar{K}\)
Compute the initial circle packing metric \(\left(M, \Gamma_{0}, \phi\right)\);
Compute the initial curvature \(K\).
    \(\mathbf{u} \leftarrow 0\).
while \(\max \left|K_{i}-\bar{K}_{i}\right|>\varepsilon\) do
    forall edge \(e=\left[v_{i}, v_{j}\right] \in M\) do
        Compute the edge weight \(w_{i} j(\mathbf{u})\) to form the Hessian matrix.
    end
    \(\mathrm{d} \mathbf{u} \leftarrow \Delta^{-1}(\bar{K}-K)\)
    \(\mathbf{u} \leftarrow \mathbf{u}+\mathrm{d} \mathbf{u}\)
    \(\mathbf{k} \leftarrow K(\mathbf{u})\).
end
\(\overline{\mathbf{u}} \leftarrow \mathbf{u}[12]\)
```

Fig. 2.16 shows the test data used for the discrete Ricci flow method. The top left figure shows the original mesh punctured at the nose with a hole. The resulting disk from the discrete Ricci flow method is shown on the top right of the figure. Texture mapping applied to the resulting disk is shown with the bottom image of the figure. Zooming in is required to see the texture mapping because of the resolution of the


Figure 2.16: Discrete Ricci Flow Output
resulting mesh. This algorithm uses the circle packing metric which are the edge lengths between two adjacent circles and is denoted $l_{i j}$. Each vertex $v_{i}$ in the mesh is encircled by a circle with $v_{i}$ as the center. The edge length of an edge $\left[v_{i}, v_{j}\right]$ is determined by $l_{i j}=\gamma_{i}+\gamma_{j}$. Intersecting circles form an acute angle $\Phi: E \rightarrow[0,2 \pi)$. The intersecting angle for an edge $\left[v_{i}, v_{j}\right]$ is $\Phi\left(\left[v_{i}, v_{j}\right]\right)=\phi_{i j}$. Two circles are tangent to each other if $\phi_{i j}$ is zero. The edge length between two circles is $l_{i j}=\sqrt{\gamma_{i}^{2}+\gamma_{j}^{2}+2 \gamma_{i} \gamma_{j} \cos \phi_{i j}}$. The edge length satisfies the triangle inequality $l_{i j}+l_{j k}>l_{k i}$. The discrete Gaussian curvature for a vertex $v_{i}$ is

$$
K_{i}=2 \pi-\sum_{j k} \theta_{i}^{j k}
$$

where $\theta_{i}^{j k}$ is the corner angle at vertex $v_{i}$ in face $\left[v_{i}, v_{j}, v_{k}\right]$. For boundary cases

$$
K_{i}=\pi-\sum_{j k} \theta_{i}^{j k}
$$

The Ricci flow will conformally deform the metric by transforming circles radii. The deformation will preserve the edge weights of all the edge lengths $l_{i j}=\sqrt{\gamma_{i}^{2}+\gamma_{j}^{2}+2 \gamma_{i} \gamma_{j} \cos \phi_{i j}}$ which is $\cos \phi_{i j}$. The radii will change but the edge weights will be preserved which in this case is the angle. The deformation is therefore angle preserving. Let $u_{i}=\log \gamma_{i}$ and the target curvature at vertex $v_{i}$ be $\bar{K}_{i}$, define convex energy as

$$
E(u)=\int^{u} \sum_{i=1}^{n}\left(\bar{K}_{i}-K_{i}\right) \mathrm{d} u_{i}
$$

where $u=\left(u_{1}, u_{2}, \ldots, u_{n}\right)$
The Hessian matrix of $E$ is

$$
\nabla E=\left(\bar{K}_{1}-K_{1}, \bar{K}_{2}-K_{2}, \ldots, \bar{K}_{n}-K_{n}\right)^{T}
$$

It is positive definite. Each edge weight is defined to be

$$
w_{i j}=\frac{\left|o_{l}-o_{k}\right|}{\left|v_{i}-v_{j}\right|}
$$

where $o_{k}$ and $o_{l}$ are two circle centers. These two circles are orthogonal to the three vertex circles of a face. Fig. 2.17 shows these two circles. We can setup the following equation


Figure 2.17: Circle Packing Weight Computation

$$
\mathrm{d} K=\Delta \mathrm{d} u
$$

The Laplacian is defined for all edges ij .

$$
\Delta_{i j}=\left\{\begin{array}{l}
-w_{i j} \quad\left[v_{i}, v_{j}\right] \in M \\
\sum_{k} w_{i k} \quad i=j \\
0 \text { other }
\end{array}\right.
$$

For each vertex $v_{i}$, the differential of the curvature at $v_{i}$ is

$$
\mathrm{d} K_{i}=\sum_{\left[v_{i}, v_{j}\right] \in M} w_{i j}\left(\mathrm{~d} u_{i}-\mathrm{d} u_{j}\right)
$$

### 2.3 Supplemental Topics

Computational conformal geometry algorithms used in this thesis require knowledge from a variety of computer science topics and mathematical areas. They have been summarized here so that the reader has a reference point for additional research if interested. Each topic or area is only discussed in the context of how they explain or support the computational conformal geometry flattening algorithms. Many of the foundations behind these topics and areas are assumed and are only briefly discussed in some cases for clarity. Concepts from algebraic topology are only discussed by providing examples when applicable and explaining how they are used in the algorithms. The theory behind some of them are very abstract and difficult to understand. These theoretical concepts will not be thorougly discussed in detail here because of the amount of background materials that need to be presented in order to explain them. The reader is encouraged to pick up a book on algebraic topology if interested for additional in depth study. However, the reader is warned that these concepts require extensive knowledge of groups, spaces, and modern algebraic language constructs to fully understand them.

Using algebraic topology to study the geometry of surfaces is relatively new when compared to using classical differential geometry. It is more difficult to understand because its definitions and concepts are rigorously defined using modern algebraic language. However, it has the advantage over classical differential geometry in that topological concepts and theorems can be easily converted to algorithms and used to solve topological problems. Solving topological problems using algorithms allows us to easily describe their difficulty using computational complexity.

### 2.3.1 Differential Geometry

Differential geometry is the classical approach for studying surfaces. The concepts and theories are formulated using the tangent space concept along with differential, integral, and vector calculus that are familiar to many of us. It will be discussed in detail here because retinotopic data processing is all done on three dimensional brain surfaces. The three dimensional brain surfaces are represented using triangle meshes that were generated from multiple slices of functional magnetic resonance imaging (fMRI) data. Flattening these three dimensional surfaces to a plane can be achieved easily if the surface can be parameterized. Fig. 2.18 shows a general three-dimensional surface $\mathbb{S}$ and the mapping between a curve $\alpha$ on $\mathbb{S}$ to the $(u, v)$ domain.


Figure 2.18: General Parameterized Curves and Surfaces.

A mapping such as this preserves the local geometry of the surface. The surface patch $r(U)$ shown above is a regular surface patch in $\mathbb{R}^{3}$ and $U=(u, v)$ are called the coordinate parameters of the surface when vectors $r_{u}$ and $r_{v}$ are linearly independent, $r_{u} \wedge r_{v} \neq 0$. Vectors $r_{u}$ and $r_{v}$ are computed by taking the partial derivative with respect to $u$ and $v, r_{u}=\left(\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right)$ and $r_{v}=\left(\frac{\partial x}{\partial v}, \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}\right)$ The normal of the surface is given by $n(u, v)=\frac{r_{u} \wedge r_{v}}{\left|r_{u} \wedge r_{v}\right|}$. If we apply a parameter
transformation to the regular surface above, then we may end up with a different parametric representation of the same surface. Let $r: \mathbb{S} \rightarrow \mathbb{R}^{3}$ be a surface patch and $\phi:\left(u^{\prime}, v^{\prime}\right) \in \mathbb{S}^{\prime} \rightarrow(u, v) \in \mathbb{S}$ be a parameter transformation, then $\phi$ is a diffeomorphism. If the Jacobian is $\neq 0$, then another parametric representation can be written as $r\left(u^{\prime}, v^{\prime}\right)=r \circ \phi\left(u^{\prime}, v^{\prime}\right)=r\left(u\left(u^{\prime}, v^{\prime}\right), v\left(u^{\prime}, v^{\prime}\right)\right): \mathbb{S}^{\prime} \rightarrow \mathbb{R}^{3}$. The Jacobian is the determinant of the Jacobian matrix

$$
\frac{\partial(u, v)}{\partial\left(u^{\prime}, v^{\prime}\right)}=\left|\begin{array}{cc}
\frac{\partial u\left(u^{\prime}, v^{\prime}\right)}{\partial u^{\prime}} & \frac{\partial v\left(u^{\prime}, v^{\prime}\right)}{\partial u^{\prime}} \\
\frac{\partial u\left(u^{\prime}, v^{\prime}\right)}{\partial v^{\prime}} & \frac{\partial v\left(u^{\prime}, v^{\prime}\right)}{\left.\partial v^{\prime}\right)}
\end{array}\right|
$$

The first derivative vector at $t=i$ of the curve $\alpha(t)=r(u(t), v(t))$ on our surface patch $r(U)$ is

$$
\left.\frac{\mathrm{d} r(t)}{\mathrm{d} t}\right|_{t=i}=\left.r_{u} \frac{\mathrm{~d} u}{\mathrm{~d} t}\right|_{t=i}+\left.r_{v} \frac{\mathrm{~d} v}{\mathrm{~d} t}\right|_{t=i}
$$

The first derivative vector is tangent to the surface at $r(u(i), v(i))$. The set of all tangent vectors at a point, $p \in \mathbb{S}$, form the tangent space of $\mathbb{S}$ at $p$. The definitions above describe the local geometry of our surface $\mathbb{S}$ and will be used next to describe several theories of surfaces.

An important theorem for surfaces is Gauss's Theorema Egregium. This theorem is readily observable by trying the following example. Suppose you take a flat piece of paper and fold it into a cylinder, you will notice that the surface curvature of the flat piece of paper changes. However, suppose you shrink yourself to the size of an ant and walk along the surface of the cylinder, you will notice that the local change is the same as that of the flat piece of paper. In fact, they are the same. This observation is what Gauss's Theorema Egregium describes. This theorem tells us that Gaussian curvature remains unchange for embedded surfaces and only the first fundamental form is needed to determine it. In other words, this tells us that locally the surface curvature does not change for embedded surfaces.

We discuss several local theory of surfaces here which are fundamental and necessary for us to show Gauss's Theorema Egregium.

For our surface $\mathbb{S}$ in fig. 2.18 and its parametric representation $r(u, v)$, we can write the tangent vector as $\mathrm{d} r=r_{u} \mathrm{~d} u+r_{v} \mathrm{~d} v$. Taking the inner product of $\mathrm{d} r$, $<\mathrm{d} r, \mathrm{~d} r>$ gives us the length of $\mathrm{d} r$.

$$
<\mathrm{d} r, \mathrm{~d} r>=\mathrm{d} u^{2}<r_{u}, r_{u}>+2 \mathrm{~d} u \mathrm{~d} v<r_{u}, r_{v}>+\mathrm{d} v^{2}<r_{v}, r_{v}>
$$

We use the letters $E, F$, and $G$ to represent the inner products $\left\langle r_{u}, r_{u}\right\rangle$, $\left.<r_{u}, r_{v}\right\rangle$, and $<r_{v}, r_{v}>$ respectively. Substituting and rewriting into matrix form, we get the following,

$$
I=\left(\begin{array}{cc}
\mathrm{d} u & \mathrm{~d} v
\end{array}\right)\left(\begin{array}{ll}
E(u, v) & F(u, v) \\
F(u, v) & G(u, v)
\end{array}\right)\binom{\mathrm{d} u}{\mathrm{~d} v}
$$

. This is known as the first fundamental form and is denoted as $I$. If we change the basis for the tangent space to $\left(u^{\prime}, v^{\prime}\right)$, the transformation is directly written as $\mathrm{d} u$ and $d v$ as

$$
\begin{aligned}
\mathrm{d} u & =\frac{\partial u}{\partial u^{\prime}} \mathrm{d} u^{\prime}+\frac{\partial u}{\partial v^{\prime}} \mathrm{d} v^{\prime} \\
\mathrm{d} v & =\frac{\partial v}{\partial u^{\prime}} \mathrm{d} u^{\prime}+\frac{\partial v}{\partial v^{\prime}} \mathrm{d} v^{\prime}
\end{aligned}
$$

The first fundamental form takes on the following form

$$
I=E \mathrm{~d} u^{2}+2 F \mathrm{~d} u \mathrm{~d} v+G \mathrm{~d} v^{2}=E^{\prime} \mathrm{d} u^{\prime 2}+2 F^{\prime} \mathrm{d} u^{\prime} \mathrm{d} v^{\prime}+G^{\prime} \mathrm{d} v^{\prime 2}
$$

The Jacobian matrix is

$$
J=\left(\begin{array}{ll}
\frac{\partial u}{\partial u^{\prime}} & \frac{\partial v}{\partial u^{\prime}} \\
\frac{\partial u}{\partial v^{\prime}} & \frac{\partial v}{\partial v^{\prime}}
\end{array}\right)
$$

In matrix form, we can write the following

$$
\left(\begin{array}{ll}
E^{\prime} & F^{\prime} \\
F^{\prime} & G^{\prime}
\end{array}\right)=J\left(\begin{array}{ll}
E & F \\
F & G
\end{array}\right) J^{T} .
$$

Let $g$ be the first fundamental form. If we have two tangent vectors in the tangent plane of our surface $\mathbb{S}$

$$
\begin{aligned}
& v_{1}=r_{u} \mathrm{~d} u+r_{v} \mathrm{~d} v \\
& v_{2}=r_{u} \delta u+r_{v} \delta v
\end{aligned}
$$

we can compute the angle $\theta$ between them as

$$
\cos ^{-1} \frac{<v_{1}, v_{2}>g}{\sqrt{<v_{1}, v_{1}>g} \sqrt{<v_{2}, v_{2}>g}}
$$

For each tangent plane on our surface $\mathbb{S}$, we can compute a vector that is perpendicular to it. This vector is call the normal vector and denoted $n$. The set of all normals on our surface is also another surface, $n(u, v)$. This parametric surface would have the following form for its tangent vector, $\mathrm{d} n=n_{u} \mathrm{~d} u+n_{v} \mathrm{~d} v$. The tangent vector for our parametric surface $r(u, v)$ from before is $\mathrm{d} r=r_{u} \mathrm{~d} u+r_{v} \mathrm{~d} v$. The second fundamental form is defined as

$$
I I=-<\mathrm{d} r, \mathrm{~d} n>
$$

. Since the normal is perpendicular to the tangent plane, we have $<r_{u}, n>=0,<r_{v}, n>=0$. Partial differentiation of the equations gives

$$
\begin{aligned}
& <r_{u u}, n>+<r_{u}, n_{u}>=0, \quad<r_{u v}, n>+<r_{u}, n_{v}>=0, \\
& <r_{v u}, n>+<r_{v}, n_{u}>=0, \quad<r_{v v}, n>+<r_{v}, n_{v}>=0 .
\end{aligned}
$$

Again we use another set of letters $L, M$, and $N$ for the inner products to get the following,
$L=<r_{u u}, n>=-<r_{u}, n_{u}>$,
$M=<r_{u v}, n>=-<r_{u}, n_{v}>=-<r_{v}, n_{u}>$,
$N=<r_{v v}, n>=-<r_{v}, n_{v}>$.

Substituting and rewriting in matrix form, we have

$$
I I=\left(\begin{array}{cc}
\mathrm{d} u & \mathrm{~d} v
\end{array}\right)\left(\begin{array}{ll}
L(u, v) & M(u, v) \\
M(u, v) & N(u, v)
\end{array}\right)\binom{\mathrm{d} u}{\mathrm{~d} v}
$$

Similar to the first fundamental form for parameter transformation $\left(u^{\prime}, v^{\prime}\right)$, we have the following

$$
\left(\begin{array}{cc}
L^{\prime} & M^{\prime} \\
M^{\prime} & N^{\prime}
\end{array}\right)=J\left(\begin{array}{cc}
L & M \\
M & N
\end{array}\right) J^{T}
$$

All normal vectors of points on a surface can be mapped to the unit sphere. This mapping is the Gauss map $G: \mathbb{S} \rightarrow \mathbb{S}^{2}, r(u, v) \rightarrow n(u, v)$. The derivative map of the Gauss map is called the Weingarten map. Let $T_{p} \mathbb{S}$ denote the tangent space of surface $\mathbb{S}$ at point $p$ and $T_{n(p)} \mathbb{S}^{2}$ denote the tangent space of unit sphere surface $\mathbb{S}^{2}$ at point $p$, the Weingarten map is written as

$$
\mathbb{W}: T_{p} \mathbb{S} \rightarrow T_{n(p)} \mathbb{S}^{2}, \mathrm{~d} r \rightarrow \mathrm{~d} n
$$

If you notice that the $T_{p} \mathbb{S}$ and $T_{n(p)} \mathbb{S}^{2}$ are parallel to each other in Euclidian $\mathbb{R}^{3}$, then the Weingarten map is just mapping a plane to another plane. To find the coefficients matrix

$$
\left(\begin{array}{ll}
a & b \\
c & d
\end{array}\right)
$$

for the transformation, set up the equations using $\left\{r_{u}, r_{v}\right\}$ as the basis

$$
\begin{aligned}
& \mathbb{W}\left(r_{u}\right)=-n_{u}=a r_{u}+b r_{v} \\
& \mathbb{W}\left(r_{v}\right)=-n_{v}=c r_{u}+d r_{v}
\end{aligned}
$$

In matrix form, the equation is

$$
-\binom{n_{u}}{n_{v}}\left(\begin{array}{ll}
r_{u} & r_{v}
\end{array}\right)=\left(\begin{array}{ll}
a & b \\
c & d
\end{array}\right)\binom{r_{u}}{r_{v}}\left(\begin{array}{ll}
r_{u} & r_{v}
\end{array}\right)
$$

Using the first and second fundamental form definitions, we get the following

$$
\left(\begin{array}{ll}
a & b \\
c & d
\end{array}\right)=\left(\begin{array}{cc}
L & M \\
M & N
\end{array}\right)\left(\begin{array}{ll}
E & F \\
F & G
\end{array}\right)^{-1}=\frac{1}{E G-F^{2}}\left(\begin{array}{ll}
L G-M F & M E-L F \\
M G-N F & N E-M F
\end{array}\right)
$$

Solving for the roots of

$$
k^{2}-\frac{L G-2 M F+N E}{E G-F^{2}} k+\frac{L N-M^{2}}{E G-F^{2}}=0
$$

gives us the eigen values. The eigen values give us the principal curvatures and the eigen vectors give us the principal directions. Principal curvatures are usually denoted as $k_{1}$ and $k_{2}$. The mean curvature, normally denoted as H , is $H=\frac{1}{2}\left(k_{1}+k_{2}\right)$. The Gaussian curvature is the product of the principal curvatures, $K=k_{1} k_{2}$. Mean curvature and Gaussian curvature using first and second fundamental form definitions are

$$
\begin{aligned}
& H= \frac{1}{2} \frac{L G-2 M F+N E}{E G-F^{2}} \\
& K=\frac{L N-M^{2}}{E G-F^{2}}
\end{aligned}
$$

The Jacobian of the Gauss map is $K$. It is the ratio of the surface region area to the corresponding Gauss map image. On the surface, $r_{u} \wedge r_{v}$ is the area element.

Similarly on the Gauss unit sphere, $n_{u} \wedge n_{v}$ is the area element. Therefore we can also write $K$ as

$$
K=\frac{n_{u} \wedge n_{v}}{r_{u} \wedge r_{v}}
$$

Normal curvature of a surface along a tangent vector $v$ is defined to be the curvature of the curve $\gamma$ at a point $p$ where the curve $\gamma$ is at the intersection of the surface with a plane determined by the tangent vector $v \in T_{p} \mathbb{S}$ and the normal vector $n$. See fig. 2.19 for a general depiction of the planes and the curve $\gamma$ with respect to $n$ and $v$.


Figure 2.19: Normal Curvature at Curve $\gamma$.

The equation for normal curvature along $v$ is

$$
k_{n}(v)=<\mathbb{W}(v), v>=k_{1} \cos ^{2} \theta+k_{2} \sin ^{2} \theta
$$

Next we discuss the construction of an orthonormal moveable frame. We begin with our surface $\mathbb{S}$ with parametric representation $r(u, v)$. We arbitrarily choose two vector fields and denote them $e_{1}$ and $e_{2}$. The chosen vector fields have to satisfy the following properties $<e_{1}, e_{1}>=<e_{2}, e_{2}>=1,<e_{1}, e_{2}>=0$, and smooth with respect to $(u, v)$. The unit normal field of surface $\mathbb{S}$ is $e_{3}=e_{1} \wedge e_{2}$. Together
$r, e_{1}, e_{2}, e_{3}$ form the orthonormal frame field of the surface. Let $\omega_{1}$ and $\omega_{2}$ be differential 1-forms $\omega_{1}(v)=<e_{1}, v>$ and $\omega_{2}(v)=<e_{2}, v>$, then we can write the equation for the tangent vector $\mathrm{d} r=r_{u} \mathrm{~d} u+r_{v} \mathrm{~d} v$ using vector fields as $\mathrm{d} r=\omega_{1} e_{1}+\omega_{2} e_{2}$. The first fundamental form written using vector fields is

$$
I=<\mathrm{d} r, \mathrm{~d} r>=\omega_{1} \omega_{1}+\omega_{2} \omega_{2}
$$

The second fundamental form written using vector fields is

$$
I I=-<\mathrm{d} r, \mathrm{~d} e_{3}>=-\omega_{1} \omega_{31}-\omega_{2} \omega_{32}=\omega_{1} \omega_{13}+\omega_{2} \omega_{23}
$$

The differential of $e_{3}$ above is $\mathrm{d}_{3}=\omega_{31} e_{1}+\omega_{32} e_{2}+\omega_{33} e_{3}$ where $\omega_{3 i}=<\mathrm{d} e_{3}, e_{j}>, \quad j \in\{1,2,3\}$. The following definition is useful for simplification when working with differential 1-forms

$$
<\mathrm{d} e_{i}, e_{j}>+<e_{i}, \mathrm{~d} e_{j}>=0 \quad \text { where } i, j \in\{1,2,3\}
$$

This implies the following

$$
\omega_{i j}+\omega_{j i}=0, \quad \omega_{i i}=0
$$

The above equations for $\mathrm{d} r$ and $\mathrm{d} e_{i}$ together is call the motion equation of the orthonormal frame of the surface. They are written as follows

$$
\begin{gathered}
\mathrm{d} r=\omega_{1} e_{1}+\omega_{2} e_{2} \\
\left(\begin{array}{l}
\mathrm{d} e_{1} \\
\mathrm{~d} e_{2} \\
\mathrm{~d} e_{3}
\end{array}\right)=\left(\begin{array}{ccc}
0 & \omega_{12} & \omega_{13} \\
-\omega_{12} & 0 & \omega_{23} \\
-\omega_{13} & -\omega_{23} & 0
\end{array}\right)\left(\begin{array}{l}
e_{1} \\
e_{2} \\
e_{3}
\end{array}\right)
\end{gathered}
$$

The first and second fundamental forms are

$$
\begin{gathered}
I=\omega_{1} \omega_{1}+\omega_{2} \omega_{2} \\
I I=\omega_{1} \omega_{13}+\omega_{2} \omega_{23}
\end{gathered}
$$

Now suppose that we construct another orthonormal frame field for our surface. We denote the vector fields for this new orthonormal frame field as $e_{1}^{\prime}, e_{2}^{\prime}, e_{3}^{\prime}$. We restrict $e_{3}^{\prime}=e_{3}$ and call the angle between $e_{1}$ and $e_{1}^{\prime}$ as $\theta(u, v)$.

$$
\begin{gathered}
e_{1}^{\prime}=\cos \theta e_{1}+\sin \theta e_{2} \\
e_{2}^{\prime}=-\sin \theta e_{1}+\cos \theta e_{2}
\end{gathered}
$$

If we assume $\omega_{i j}^{\prime}$ where $i, j=\{1,2,3\}$ to be the coefficients of the motion equation for our new orthonormal frame defined by $e_{1}^{\prime}, e_{2}^{\prime}, e_{3}^{\prime}$, then the following results can be directly obtained

$$
\begin{aligned}
& \binom{\omega_{1}^{\prime}}{\omega_{2}^{\prime}}=\left(\begin{array}{cc}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{array}\right)\binom{\omega_{1}}{\omega_{2}} \\
& \binom{\omega_{31}^{\prime}}{\omega_{32}^{\prime}}=\left(\begin{array}{cc}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{array}\right)\binom{\omega_{31}}{\omega_{32}}
\end{aligned}
$$

The following is therefore true

$$
\begin{gathered}
\omega_{1}^{\prime} \omega_{1}^{\prime}+\omega_{2}^{\prime} \omega_{2}^{\prime}=\omega_{1} \omega_{1}+\omega_{2} \omega_{2} \\
\omega_{1}^{\prime} \omega_{31}^{\prime}+\omega_{2}^{\prime} \omega_{32}^{\prime}=\omega_{1} \omega_{31}+\omega_{2} \omega_{32}
\end{gathered}
$$

The results above show that the first and second fundamental form are independent of what we choose as the orthonormal frame. If we write $\omega_{13}$ and $\omega_{23}$ as a linear combination of $\omega_{1}$ and $\omega_{2}$, we get the following

$$
\binom{\omega_{13}}{\omega_{23}}=\left(\begin{array}{ll}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}\right)\binom{\omega_{1}}{\omega_{2}}
$$

with matrix

$$
H=\left(\begin{array}{ll}
h_{11} & h_{12} \\
h_{21} & h_{22}
\end{array}\right)
$$

representing the Weingarten mapping

$$
\mathrm{d} r=\omega_{1} e_{1}+\omega_{2} e_{2} \rightarrow-\mathrm{d} e_{3}=\omega_{31} e_{1}+\omega_{32} e_{2}
$$

The principal curvature, Gaussian curvature, and mean curvature can all be computed from matrix H .

The orthonormal frame and the motion equations together defines a surface. If we choose an orthonormal frame as parameterization $r$ and vector fields $e 1, e 2, e 3$, then the motion equations are

$$
\begin{gathered}
\mathrm{d} r=\omega_{1} e_{1}+\omega_{2} e_{2} \\
\mathrm{~d} e_{i}=\sum_{j=1}^{3} \omega_{i j} e_{j}, \quad \omega_{i j}+\omega_{j i}=0, \quad i=1,2,3
\end{gathered}
$$

We can get the following equations from using the fact that $\mathrm{d}^{2} r=0$,

$$
\omega_{1} \wedge \omega_{13}+\omega_{2} \wedge \omega_{23}=0
$$

and also

$$
\left(h_{12}-h_{21}\right) \omega_{1} \wedge \omega_{2}=0
$$

We also get this set of equations

$$
\begin{aligned}
& \mathrm{d} \omega_{1}=\omega_{2} \wedge \omega_{21} \\
& \mathrm{~d} \omega_{2}=\omega_{1} \wedge \omega_{12} \\
& \mathrm{~d} \omega_{12}=\omega_{13} \wedge \omega_{32} \\
& \mathrm{~d} \omega_{13}=\omega_{12} \wedge \omega_{23} \\
& \mathrm{~d} \omega_{23}=\omega_{21} \wedge \omega_{13}
\end{aligned}
$$

Equation $\mathrm{d} \omega_{12}=\omega_{13} \wedge \omega_{32}$ is the Gauss equation and can be derived by using $\mathrm{d}^{2} e_{i}=0$ for $i=1,2$. Equations $\mathrm{d} \omega_{13}=\omega_{12} \wedge \omega_{23}$ and $\mathrm{d} \omega_{23}=\omega_{21} \wedge \omega_{13}$ are the Codazzi equations. Previously we discussed the H matrix when talking about
representing $\omega_{13}$ and $\omega_{23}$ as a linear combination of $\omega_{1}$ and $\omega_{2}$. Using $\omega_{13}=h_{11} \omega_{1}+h_{12} \omega_{2}$ and $\omega_{23}=h_{21} \omega_{1}+h_{22} \omega_{2}$, we get the following

$$
\mathrm{d} \omega_{12}=-\left(h_{11} h_{22}-h_{12}^{2}\right) \omega_{1} \wedge \omega_{2}=-K \omega_{1} \wedge \omega_{2}
$$

### 2.3.2 Exterior Differential Calculus

Exterior differential calculus is not completely different from regular differential and integral calculus. The manifold in exterior differential calculus is like the tangent planes in regular calculus. If you take all the tangent planes of a surface and placed them side by side you would have what is called a differential atlas in exterior differential calculus. Similarly other concepts like tangent, differential, cross-product, integral, gradient, and Stokes theorem all have a corresponding exterior calculus version of how to compute them. Tangent, differential, and integral do not have different names. The cross-product is call the wedge product. The gradient used in Stokes theorem is call the exterior derivative. The differential and integral forms of exterior calculus are better suited for computation on discretized meshes. Discretized meshes are more practical for real world application where data sampling is not infinite.

A triangle mesh is a simplicial complex. It is composed of simplexes: points, edges, and faces glued together. Functions defined on vertices, halfedges, and oriented faces are called simplicial 0 -form, 1 -form, and 2 -form respectively. Let $v_{0}$, $v_{1}$, and $v_{2}$ be the vertices of an oriented face. The differential of a 0 -form is a 1 -form.

$$
\mathrm{d} f\left(\left[v_{0}, v_{1}\right]\right)=f\left(v_{1}\right)-f\left(v_{0}\right)
$$

The differential of a 1 -form is a 2 -form

$$
\mathrm{d} \omega\left(\left[v_{0}, v_{1}, v_{2}\right]\right)=\omega\left(\left[v_{0}, v_{1}\right]\right)+\omega\left(\left[v_{1}, v_{2}\right]\right)+\omega\left(\left[v_{2}, v_{0}\right]\right)
$$

This type of computation is easy to compute and can be used on discretized surfaces. Integration for exterior calculus is also straightforward. The integral of a

2-form will yield a 1 -form and the integral of a 1 -form will yield a 0 -form. So the result of integrating a connected triangle patch (2-form) will result in a closed loop (1-form) and the integration of a loop will result in a vertex (0-form). This is like Green's and Stoke's theorem for line and flux integrals.

### 2.3.3 Algebraic Topology

The concept of homotopy, homology, and cohomology groups from algebraic topology will be discussed in this section. They are directly used by the harmonic holomorphic 1-form conformal mapping algorithm used in this thesis. For these concepts, only the theorem results and their application to the study of surfaces are presented. The rigorous definitions are left up to the reader to explore on their own. Homotopy and homology groups are geometric and can be easily depicted using texture mapping on surfaces. Cohomology groups are not geometric like homotopy and homology groups but are very useful for studying topology. They are difficult to explain but their computation is simpler than the previous groups.

Before discussing homotopy, homology, and cohomology groups, some concepts of topological surfaces will be reviewed here. The retinotopic data collected for this thesis are projected onto a closed surface mesh representation of the brain for visualization in three dimensions. The brain mesh is a topological surface. Three dimensional surfaces in topology are treated like they are made of an elastic rubber material. They can be stretched or compressed but not cut. These two permitted transformations allow us to change the shape of the surface but not its topological classification. Topological surfaces are classified according to their genus number. This number tells us the number of handles a surface has. A sphere has no handles and so is classified as a genus zero surface. A torus or donut shaped surface has one handle and is therefore classified as a genus one surface. Higher genus surfaces can be constructed by joining more than one donut together. Two
donuts are joined together by removing a hole from each and joining them at this location. Surfaces joined this way is call a connected sum. A connected sum of $n$ tori is a genus- $n$ surface. All genus zero surfaces can be topologically deformed to a sphere. Similarly all genus one surfaces can be deformed to a torus. However it is not possible to deform a sphere into a donut without cutting it open. This observation shows us that genus is a topological invariant property. Handles used in genus classification are not boundaries. If you were to travel along the surface of a donut, you will not notice the handle. Travelling along a surface restricts you to a two dimensional world. You can no longer see the handle like you can in three dimensions. A boundary on a surface is a hole cut out of the surface. Cutting a hole in the surface allows you to see the inside of it. If you travel along a surface with a boundary, you will be able to identify the boundary when you approach it. After you reach a boundary, you can follow it and restrict yourself to one dimension of freedom. Surfaces without boundaries are closed surfaces while those with boundaries are open surfaces. A sphere and a donut are examples of closed surfaces. A disk is an example of a open surface. It is not possible to deform a closed surface to an open surface without cutting. This is why boundary is also another topological invariant property. A spherical surface has an inside and an outside. We know this rather intuitively. Some surfaces however are not orientable like the Möbius strip. We cannot deform a surface which is orientable and deform it so that it is non-orientable. As a result, orientability is another topological invarian property. These topological invariants are the same for topologically equivalent surfaces. All closed connected surfaces are topologically equivalent to only three types of closed surfaces. This is summed up in the classification theorem for surfaces from Computational Conformal Geometry [12] :

Theorem 2.3 (Classification Theorem for Surfaces).

Any closed connected surface is homeomorphic to exactly one of the following surfaces: a sphere, a finite connected sum of tori, or a sphere with a finite number of disjoint discs removed and with crosscaps glued in their place. The sphere and connected sums of tori are orientable surfaces, whereas surfaces with crosscaps are unorientable. [12]

Homotopy is used to describe the existence of a set of continuous mappings in between two continuous mappings. A way to think about this is the morphing of two equivalent topological surfaces. A football for example can be deformed into a sphere since they are both closed genus zero surfaces. We write the mapping as $f_{0}: F \rightarrow S$. An intermediate shape in between the deformation could also be deformed into a sphere. We write the mapping of this intermediate shape to a sphere as $f_{1}: I \rightarrow S$. Between the football and the intermediate shape is a set of shapes that could also map to the sphere. This set of shapes and their mapping to a sphere exists so the mappings $f_{0}$ and $f_{1}$ are considered homotopic to each other. Homotopy equivalence is when two topological spaces can be mapped to each other in both direction. Using the football to sphere example, the definition for their homotopy equivalence is that there must exist continuous maps $f: F \rightarrow S$ and $g: S \rightarrow F$ such that

$$
\begin{aligned}
& g \circ f \cong i d_{F}: F \rightarrow F \\
& f \circ g \cong i d_{S}: S \rightarrow S
\end{aligned}
$$

Two closed loops on a surface are homotopic to each other if we can deform one to the other with the restriction that the loops cannot leave the surface. The set of homotopic closed loops that start and end at the same point $p$ form a homotopy group. This is important for identifying the set of loops on a surface for cutting. Being able to identify the set of homotopic loops is useful for finding where to slice the mesh so that we can unfold it to a plane. The harmonic and holomorphic 1-form algorithm that we use in this thesis requires an open cut mesh as its input.

The path on a surface mesh that when cutting along it opens up the surface mesh and creates a simply connected patch is form from what is called the canonical fundamental group basis. The definition from Computational Conformal Geometry [12] is :

## Definition 2.17 (Canonical Fundamental Group Basis).

A fundamental group basis $\left\{a_{1}, b_{1}, a_{2}, b_{2}, \ldots, a_{g}, b_{g}\right\}$ is canonical if

1. $a_{i}$ and $b_{i}$ intersect at the same point $p$.
2. $a_{i}$ and $a_{j}, b_{i}$ and $b_{j}$ only touch at $p$. [12]

Fig. 2.20 gives a picture of what the a canonical fundamental group basis looks like on a genus two surface. The point in the middle of the double torus where all the line goes through is the point p mentioned in the definition above.


Figure 2.20: Canonical Fundamental Group Basis

The brain meshes processed in this thesis are triangulated meshes. The original brain is a smooth and continuous surface whereas our data is only a discrete approximated representation of the imaging data. A triangular mesh is a collection of triangular simplexes glued together. A triangle simplex is a standard simplex because it is the minimal convex set for its vertices. A triangle simplex has orientation determined by the order of its vertices. Topological concepts like genus, boundary, and orientability can all be detected using straightforward algorithms for topological surfaces that have been approximated using a triangular mesh. For a
triangular mesh surface, it is useful to extract groups of connected edges or triangle faces. A curve on a mesh surface is a set of consecutive oriented edges. A patch on a mesh surface is a set of adjacent oriented faces. Using chain group language, a curve on a mesh is a 1 -chain and a patch is a 2-chain. Closed 1-chain loops are those that do not have any boundary vertices while open 1-chain loops are those with boundary vertices. An exact 1-chain loop is a closed 1-chain loop that also encloses a patch. A $q$ dimensional homology group $H$ of a chain complex $M$ is the quotient group of the $q$ dimensional closed chain $C$ group over the $q$ dimensional boundary chain group $B$.

$$
H_{q}(M)=\frac{C_{q}(M)}{B_{q}(M)}
$$

The cohomology group is defined also as a $q$ dimensional quotient group where the closed chain group and boundary chain group are substituted with cochain group and coboundary group respectively. The cochain and coboundary group are easier to compute than regular closed chain and boundary group. It is used when computing harmonic and holomorphic 1-form for one of the flattening algorithms looked at in this thesis.

### 2.3.4 Half-Edge Data Structure

The applications of these algorithms relies on an understanding of topology and its discrete representation for computer processing. This section provides a brief description of the triangle mesh discrete representation. A triangle mesh is a simplicial complex. Simplexes found in a triangle mesh are points, lines, and triangle. The corresponding names for them in a triangle mesh data structure are vertices, edges, and faces. There are many different ways to store the triangle mesh as a data structure in a computer. One of them is the half-edge data structure. The initial setup of the structure is a little more complex than some of the simpler ones. However the half-edge structure provides for a very efficient way to search for
adjacent edges and vertices. Adjacent searching for edges and vertices is used extensively in many topological algorithms. Fig. 2.21 shows the various pointers for the half edge data structure.


Figure 2.21: Half Edge Data Structure

## Chapter 3

## RESULTS

The first part of the results section is a table summary of the flattening algorithms. Tab. 3.1 lists the pros and cons of the three algorithms. Next are tables listing the performance of these algorithms. The run time for each algorithm and the mesh properties of the test data are listed in tab. 3.2, tab. 3.4, and tab. 3.6. The accuracy of each algorithm was checked using the Beltrami coefficient and are shown using histogram plots following the tables. The histogram plots show how conformal the deformed shapes are compared to the original shapes. The smaller the standard deviation of the histogram plot the better the conformal mapping. The second part of the results section focuses on showing data from actual retinotopic data processing. Flattened V1 region for five test data sets are shown along with the histogram plot comparing how the the discrete Ricci flow method compared with the non-linear heat diffusion method. The final set of histograms compares the recovered parameterization mesh versus the actual V1 region mesh for a selected few data sets that actually show valid results.

| Algorithm | Surface Types | Pros and Cons |
| :---: | :---: | :---: |
| Non-Linear Heat Diffusion Spherical | Sphere | Straightforware implementation. Long run time. Mö bius computations required. Convergence not always guaranteed. |
| Harmonic and Holomorphic 1-form | Sphere, Closed or Open Genus Zero Surfaces, High Genus Surfaces | Fastest. Stable. Linear solver required. Need to slice mesh and find boundaries. Minimal convergence issues. |
| Directe Ricci Flow | Sphere, Closed or Open Genus Zero Surfaces, High Genus Surfaces | Fast and Stable. Linear solver required. Minimal convergence issues. |

Table 3.1: Algorithm Results.

| Mesh | Time (Average 3 runs) | Conformal |
| :--- | :--- | :--- |
| Brain | $20-30$ minutes | Histogram A.1a and A.1b |
| Brain Subdivision 1 | $4-5$ hours | Histogram A.2a and A.2b |
| Brain Subdivision 2 | $40+$ hours | No Data |

Table 3.2: Non-Linear Heat Diffusion Algorithm Results.

| Mesh | Vertices | Faces | Edges |
| :--- | :--- | :--- | :--- |
| Brain | 2502 | 5000 | 7500 |
| Brain Subdivision 1 | 10002 | 20000 | 30000 |
| Brain Subdivision 2 | 40002 | 80000 | 120000 |

Table 3.3: Non-Linear Heat Diffusion Test Data.

| Mesh | Time (Average 3 runs) | Conformal |
| :--- | :--- | :--- |
| Face | $9-10$ seconds | N/A |
| Face Subdivision 1 | $40-50$ seconds | N/A |
| Face Subdivision 2 | $2-3$ minutes | N/A |

Table 3.4: Harmonic and Holomorphic 1-form Algorithm Results.

| Mesh | Vertices | Faces | Edges |
| :--- | :--- | :--- | :--- |
| Face | 5101 | 9999 | 15100 |
| Face Subdivision 1 | 20201 | 39996 | 60197 |
| Face Subdivision 2 | 80398 | 159984 | 240382 |

Table 3.5: Harmonic and Holomorphic 1-form Test Data.

| Mesh | Time (Average 3 runs) | Conformal |
| :--- | :--- | :--- |
| Face | $1-2$ minutes | Histogram B.1a and B.1b |
| Face Subdivision 1 | $3-4$ minutes | Histogram B.2a and B.2b |
| Face Subdivision 2 | $14-15$ minutes | Histogram B.3a and B.3b |

Table 3.6: Discrete Ricci Flow Algorithm Results.

| Mesh | Vertices | Faces | Edges |
| :--- | :--- | :--- | :--- |
| Face | 5087 | 9961 | 15048 |
| Face Subdivision 1 | 20135 | 39844 | 59979 |
| Face Subdivision 2 | 80114 | 159376 | 239490 |

Table 3.7: Discrete Ricci Flow Test Data.

| Subject | Time | Conformal |
| :--- | :--- | :--- |
| ABS1VFM Left | 58 hours | Histogram C.1a and <br> C.1b |
| BBCS3PP Right | Failed | N/A Failed |
| LLS1PP Left | 8 hours | Histogram C.2a and <br> C.2b |
| LLS1PP Right | $60+$ hours | N/A Stopped after 60 <br> hours |
| SADS3VFM Left | 53 hours | Histogram C.3a and <br> C.3b |
| SADS3VFM Right | 18 hours | Histogram C.4a and <br> C.4b |

Table 3.8: Retinotopic Non-Linear Heat Diffusion Algorithm Results.

| Subject | Time | Conformal |
| :--- | :--- | :--- |
| ABS1VFM Left | Failed | N/A |
| BBCS3PP Right | 3 min | Histogram D.1a and <br> D.1b |
| LLS1PP Left | Failed | N/A |
| LLS1PP Right | 4 min | Histogram D.2a and <br> D.2b |
| SADS3VFM Left | 3 min | Histogram D.3a and <br> D.3b |
| SADS3VFM Right | 2 min | Histogram D.4a and <br> D.4b |

Table 3.9: Retinotopic Discrete Ricci Flow Algorithm Results.


Figure 3.1: Histogram A.1a: Sphere Test Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=2502$. Mean $=-0.0064$


Figure 3.2: Histogram A.1b: Sphere Test Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=2502$. Mean $=0.0014$


Figure 3.3: Histogram A.2a: Sphere Test Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=10002$. Mean $=0.0027$


Figure 3.4: Histogram A.2b: Sphere Test Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=10002$. Mean $=0.0068$


Figure 3.5: Histogram B.1a: Face Test Discrete Ricci Flow Histogram (Complex). Vertices Count $=5087$. Mean $=0.0345$


Figure 3.6: Histogram B.1b: Face Test Discrete Ricci Flow Histogram (Real). Vertices Count $=5087$. Mean $=-0.0322$


Figure 3.7: Histogram B.2a: Face Test Discrete Ricci Flow Histogram (Complex). Vertices Count $=$ 20135. Mean $=0.0091$


Figure 3.8: Histogram B.2b: Face Test Discrete Ricci Flow Histogram (Real). Vertices Count $=20135$. Mean $=-0.0010$


Figure 3.9: Histogram B.3a: Face Test Discrete Ricci Flow Histogram (Complex). Vertices Count $=80114$. Mean $=0.0083$


Figure 3.10: Histogram B.3b: Face Test Discrete Ricci Flow Histogram (Real). Vertices Count $=80114$. Mean $=-0.0094$


Figure 3.11: Histogram C.1a: AABS1VFM Left Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=22817$. Mean $=0.0053$


Figure 3.12: Histogram C.1b: AABS1VFM Left Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=22817$. Mean $=-0.0229$


Figure 3.13: Histogram C.2a: LLS1PP Left 3deg Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=25586$. Mean $=0.0030$


Figure 3.14: Histogram C.2b: LLS1PP Left 3deg Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=25586$. Mean $=-0.0222$


Figure 3.15: Histogram C.3a: SADS3VFM Leftt 3deg Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=16523$. Mean $=0.0024$


Figure 3.16: Histogram C.3b: SADS3VFM Leftt 3deg Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=16523$. Mean $=-0.0159$


Figure 3.17: Histogram C.4a: SADS3VFM Right 3deg Non-Linear Heat Diffusion Histogram (Complex). Vertices Count $=13525$. Mean $=-0.0016$


Figure 3.18: Histogram C.4b: SADS3VFM Right 3deg Non-Linear Heat Diffusion Histogram (Real). Vertices Count $=13525$. Mean $=-0.0131$


Figure 3.19: Histogram D.1a: BBCS3PP Right 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=16322$. Mean $=0.0344$


Figure 3.20: Histogram D.1b: BBCS3PP Right 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=16322$. Mean $=-0.0406$


Figure 3.21: Histogram D.2a: LLS1PP Right 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=21075$. Mean $=0.0291$


Figure 3.22: Histogram D.2b: LLS1PP Right 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=21075$. Mean $=-0.0386$


Figure 3.23: Histogram D.3a: SADS3VFM Leftt 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=16523$. Mean $=0.0320$


Figure 3.24: Histogram D.3b: SADS3VFM Leftt 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=16523$. Mean $=-0.0375$


Figure 3.25: Histogram D.4a: SADS3VFM Right 3deg Discrete Ricci Flow Histogram (Complex). Vertices Count $=13525$. Mean $=0.0267$


Figure 3.26: Histogram D.4b: SADS3VFM Right 3deg Discrete Ricci Flow Histogram (Real). Vertices Count $=13525$. Mean $=-0.0353$


Figure 3.27: Histogram E.1a: LLS1PP Left 3deg Parameterization Histogram V1 (Complex). Vertices Count $=1238$. Mean $=0.0587$


Figure 3.28: Histogram E.1b: LLS1PP Left 3deg Parameterization Histogram V1 (Real). Vertices Count $=1238$. Mean $=0.2413$


Figure 3.29: Histogram F.1a: SADS3VFM Left 3deg Parameterization Histogram V1 (Complex). Vertices Count $=887$. Mean $=0.0358$


Figure 3.30: Histogram F.1b: SADS3VFM Left 3deg Parameterization Histogram V1 (Real). Vertices Count $=887$. Mean $=0.1983$

## Chapter 4

DISCUSSION
The run times collected for each of the three algorithms implemented showed a wide gap between the linear solver based algorithms and the gradient descent method. It was expected that the convergence times for the non-linear heat diffusion method to be slow but not as slow as was observed. This is a really big drawback for this method since its implementation is relatively straight forward. However it did manage to to converge on a few of the retinotopic data that the discrete Ricci flow method could not. The harmonic/holomorphic 1-form flattening to a disk was not successful and therefore was not used on retinotopic data. There were issues with the output $u, v$ coordinates being too small or too large for the mesh viewer program to handle. The speed of the harmonic/holomorphic 1-form is very fast however. It is a direct computation rather than in iterative convergence algorithm so it can be solved much more quickly than the discrete Ricci flow method.

### 4.1 Test Data

The algorithms all performed well using their respective test data. The non-linear heat diffusion method convergence time went into days by the time the mesh was subdivided a second time. As the amount of vertices increased the amount of time required for convergence grew by a factor of about eight. Ricci flow and harmonic/holomorphic 1-from grew by a factor of around three to four. This is about the same factor that the vertices grew by using subdivision.

### 4.2 Retinotopic Data

Flattening retinotopic data results for Ricci flow and heat diffusion had significantly different convergence rates but the conformal factor was similar. Both algorithms had Beltrami coefficients plots that were close to zero for all retinotopic data that each algorithm was able to flatten. Ricci flow was able to achieve a mapping for
some cases when non-linear heat diffusion failed to finish. On the other hand, some subject data such as ABS1VFM left retinotopic data could not be flatten using the Ricci flow method. No matter the sizes of the punctured hole tried, the algorithm would not converge. This needs to be further explored since the non-linear heat diffusion method was able to successfully converge and flatten this mesh.

### 4.3 Beltrami Coefficient analysis

Two subjects were selected to test the parameterization retrieval method using the color map. The resulting histogram plots of the $u, v$ meshes versus their conformal flattened versions reveal that the shape of the histograms are very close to each other. More analysis of the shape and the distribution of these histograms across subjects will be useful to determine if the human visual areas can be characterized using this model. One of the problem that has to be addressed is the difference in size of V1 across subjects. Subject SADS3VFM has only 887 vertices in the left V1 while subject LLS1PP has 1238 vertices.

## Chapter 5

## CONCLUSION

This thesis explored using conformal geometry and Beltrami coefficient for characterizing human retinotopic mapping. The problem was to find a function to describe the mapping between the human visual field and the primary cortical visual areas. The approach was to treat the problem as a shape deformation matching problem. Matching problems in two-dimensions are simpler than three-dimensional ones. Therefore, computational conformal geometry flattening algorithms were implemented and evaluated. These algorithms were used to flatten test data and retinotopic data. Beltrami coefficients were computed and plotted as histograms between the visual field mesh and flattened cortical visual areas mesh for two initial test subject data. Comparing the flattened visual cortical areas of the brain across two initial test subjects revealed similar histogram shapes. This finding demonstrates that flattening the visual cortical mesh and computing the Beltrami coefficient gives a valid quantitative way to characterize retinotopic mapping data.

### 5.1 Future Work

Quasi-conformal mapping [11] [1] [20] will be looked at. Retinotopic organization varies in cones receptor density from high to low when moving away from the central foveal region. More visual cortical neurons are devoted to the foveal region compared the other regions. Objects appearing in our peripheral vision do not appear sharp as when they are focussed in our foveal region. As a result, it is hypothesized that conformal mapping equates to clarity in our visions. We are able to identify what angle is between two lines if we are looking at it directly instead of indirectly. Foveal vision may explain why the histograms were more quasiconformal than conformal.

## REFERENCES

[1] Ahlfors, L. V., Lectures on quasiconformal mappings, vol. 38 (American Mathematical Society, 2006), second edition edn.
[2] Balasubramanian, M., J. Polimeni and E. Schwartz, "Exact geodesics and shortest paths on polyhedral surfaces", IEEE Trans. Patt. Anal. Mach. Intell. 31, 6, 1006-1016 (2009).
[3] Balasubramanian, M., J. Polimeni and E. L. Schwartz, "The V1 -V2-V3 complex: quasiconformal dipole maps in primate striate and extra-striate cortex", Neural Netw 15, 1157-1163 (2002).
[4] Balasubramanian, M., J. R. Polimeni and E. L. Schwartz, "Near-isometric flattening of brain surfaces", Neuroimage 51, 2, 694-703 (2010).
[5] Balasubramanian, M., J. R. Polimeni and E. L. Schwartz, "Near-isometric flattening of brain surfaces", Neuroimage 51, 694-703 (2010).
[6] Botsch, M., L. Kobbelt, M. Pauly, P. Alliez and B. Levy, Polygon Mesh Processing (A K Peters, 2010).
[7] DeYoe, E. A., G. J. Carman, P. Bandettini, S. Glickman, J. Wieser, R. Cox, D. Miller and J. Neitz, "Mapping striate and extrastriate visual areas in human cerebral cortex", Proc. Natl. Acad. Sci. U.S.A. 93, 2382-2386 (1996).
[8] Engel, S. A., G. H. Glover and B. A. Wandell, "Retinotopic organization in human visual cortex and the spatial precision of functional MRI", Cereb. Cortex 7, 2, 181-192 (1997).
[9] Engel, S. A., G. H. Glover and B. A. Wandell, "Retinotopic organization in human visual cortex and the spatial precision of functional MRI", Cereb. Cortex 7, 181-192 (1997).
[10] Engel, S. A., D. E. Rumelhart, B. A. Wandell, A. T. Lee, G. H. Glover, E. J. Chichilnisky and M. N. Shadlen, "fMRI of human visual cortex", Nature 369, 525 (1994).
[11] Gardiner, F. P. and N. Lakic, Quasiconformal Teichmüller Theory, vol. 76 (American Mathematical Society, 2000).
[12] Gu, X. D. and S. T. Yau, Computational Conformal Geometry (International Press, 2008).
[13] Hansen, K. A., S. V. David and J. L. Gallant, "Parametric reverse correlation reveals spatial linearity of retinotopic human V1 BOLD response", Neuroimage 23, 233-241 (2004).
[14] Huettel, S. A., A. W. Song and G. McCarthy, Functional Magnetic Resonance Imaging (Sinauer, 2009), 2 edn.
[15] Hurdal, M. K. and K. Stephenson, "Cortical cartography using the discrete conformal approach of circle packings", NeuroImage 23, S119-S128 (2004).
[16] Hurdal, M. K. and K. Stephenson, "Discrete conformal methods for cortical brain flattening", NeuroImage 45, S86-S98 (2009).
[17] Ju, L., M. K. Hurdal, J. Stern, K. Rehm, K. Schaper and D. Rottenberg, "Quantitative evaluation of three cortical surface flattening methods", Neuroimage 28, 4, 869-880 (2005).
[18] Kruskal, J. B., "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis", Psychometrika 29, 1, 1-27, URL http://www.springerlink.com/index/10.1007/BF02289565 (1964).
[19] Kruskal, J. B., "Nonmetric multidimensional scaling: A numerical method", Psychometrika 29, 2, 115-129, URL http://ideas.repec.org/a/spr/psycho/v29y1964i2p115-129.html (1964).
[20] Lui, L. M., T. W. Wong, P. M. Thompson, T. Chan, X. Gu and S.-T. Yau, "Shape-based diffeomorphic registration on hippocampal surfaces using Beltrami holomorphic flow", Med Image Comput Comput Assist Interv 13, 323-330 (2010).
[21] Polimeni, J. R., M. Balasubramanian and E. L. Schwartz, "Multi-area visuotopic map complexes in macaque striate and extra-striate cortex", Vision Res. 46, 3336-3359 (2006).
[22] Qiu, A., B. J. Rosenau, A. S. Greenberg, M. K. Hurdal, P. Barta, S. Yantis and M. I. Miller, "Estimating linear cortical magnification in human primary visual cortex via dynamic programming", Neuroimage 31, 1, 125-138 (2006).
[23] Schira, M. M., C. W. Tyler, M. Breakspear and B. Spehar, "The foveal confluence in human visual cortex", J. Neurosci. 29, 9050-9058 (2009).
[24] Schira, M. M., A. R. Wade and C. W. Tyler, "Two-dimensional mapping of the central and parafoveal visual field to human visual cortex", J. Neurophysiol. 97, 6, 4284-4295 (2007).
[25] Schneider, W., D. C. Noll and J. D. Cohen, "Functional topographic mapping of the cortical ribbon in human vision with conventional MRI scanners", Nature 365, 150-153 (1993).
[26] Schwartz, E. L., "The development of specific visual connections in the monkey and the goldfish: outline of a geometric theory of receptotopic structure", J. Theor. Biol. 69, 655-683 (1977).
[27] Schwartz, E. L., "Cortical mapping and perceptual invariance: a reply to Cavanagh", Vision Res. 23, 831-835 (1983).
[28] Schwartz, E. L., A. Shaw and E. Wolfson, "A numerical solution to the generalized mapmaker's problem: Flattening nonconvex polyhedral surfaces", IEEE Trans. Patt. Anal. Mach. Intell. 11, 9, 1005-1008 (1989).
[29] Sereno, M. I., A. M. Dale, J. B. Reppas, K. K. Kwong, J. W. Belliveau, T. J. Brady, B. R. Rosen and R. B. Tootell, "Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging", Science 268, 889-893 (1995).
[30] Shepard, R., "The analysis of proximities: Multidimensional scaling with an unknown distance function. ii", Psychometrika 27, 219-246, URL http://dx.doi.org/10.1007/BF02289621, 10.1007/BF02289621 (1962).
[31] Tootell, R. B., J. B. Reppas, K. K. Kwong, R. Malach, R. T. Born, T. J. Brady, B. R. Rosen and J. W. Belliveau, "Functional analysis of human MT and related visual cortical areas using magnetic resonance imaging", J. Neurosci. 15, 3215-3230 (1995).
[32] Torgerson, W., "Multidimensional scaling: I. theory and method", Psychometrika 17, 4, 401-419, URL http://ideas.repec.org/a/spr/psycho/v17y1952i4p401-419.html (1952).
[33] Vanni, S., L. Henriksson and A. C. James, "Multifocal fMRI mapping of visual cortical areas", Neuroimage 27, 95-105 (2005).
[34] Wandell, B. A., "Computational neuroimaging of human visual cortex", Annu. Rev. Neurosci. 22, 145-173 (1999).
[35] Wandell, B. A. and J. Winawer, "Imaging retinotopic maps in the human brain", Vision Res. 51, 718-737 (2011).
[36] Wang, Y., W. Dai, Y.-Y. Chou, X. Gu, T. Chan, A. Toga and P. Thompson, "Studying brain morphometry using conformal equivalence class", in "Computer Vision, 2009 IEEE 12th International Conference on", pp. 2365-2372 (2009).
[37] Wang, Y., X. Gu, T. F. Chan, P. M. Thompson and S.-T. Yau, "Intrinsic brain surface conformal mapping using a variational method", in "Proc. SPIE Medical Imaging", pp. 241-252 (2004).
[38] Wang, Y., L. M. Lui, X. Gu, K. M. Hayashi, T. F. Chan, A. W. Toga, P. M. Thompson and S.-T. Yau, "Brain surface conformal parameterization using Riemann surface structure", IEEE Trans. Med. Imag. 26, 6, 853-865 (2007).
[39] Wang, Y., J. Shi, X. Yin, X. Gu, T. Chan, S.-T. Yau, A. Toga and P. Thompson, "Brain surface conformal parameterization with the Ricci flow", IEEE Trans Med Imaging In Press (2011).
[40] Wang, Y., L. Yuan, J. Shi, A. Greve, J. Ye, A. W. Toga, A. L. Reiss and P. M. Thompson, "Applying tensor-based morphometry to parametric surfaces can improve MRI-based disease diagnosis", Neuroimage 74, 209-230 (2013).
[41] WHO, W., "Medical imaging", URL
http://www.who.int/diagnostic_imaging/en/ (2013).
[42] Xianfeng, D. G., "Computational conformal geometry", URL http: //www.cs.sunysb.edu/~zengwei/nsf_poster/nsf_poster.html\#APPLICATIONS (2010).

## APPENDIX A

PROCESSING DETAILS AND DATA

| R | G | B | H | S | V |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.87451 | 0.07451 | 0.07451 | 0 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.12549 | 0 | 0.864407 | 0.92549 |
| 0.87451 | 0.11373 | 0.07451 | 2.9415 | 0.914798 | 0.87451 |
| 0.92549 | 0.16471 | 0.12549 | 2.9415 | 0.864407 | 0.92549 |
| 0.87451 | 0.14902 | 0.07451 | 5.58825 | 0.914798 | 0.87451 |
| 0.92549 | 0.2 | 0.12549 | 5.58825 | 0.864407 | 0.92549 |
| 0.87451 | 0.18824 | 0.07451 | 8.52975 | 0.914798 | 0.87451 |
| 0.92549 | 0.23922 | 0.12549 | 8.52975 | 0.864407 | 0.92549 |
| 0.92549 | 0.27451 | 0.12549 | 11.1765 | 0.864407 | 0.92549 |
| 0.87451 | 0.22745 | 0.07451 | 11.4705 | 0.914798 | 0.87451 |
| 0.87451 | 0.26275 | 0.07451 | 14.118 | 0.914798 | 0.87451 |
| 0.92549 | 0.31373 | 0.12549 | 14.118 | 0.864407 | 0.92549 |
| 0.87451 | 0.30196 | 0.07451 | 17.0588 | 0.914798 | 0.87451 |
| 0.92549 | 0.35294 | 0.12549 | 17.0588 | 0.864407 | 0.92549 |
| 0.87451 | 0.33725 | 0.07451 | 19.7055 | 0.914798 | 0.87451 |
| 0.92549 | 0.38824 | 0.12549 | 19.7063 | 0.864407 | 0.92549 |

Table A.1: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (1-16)

| $R$ | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.87451 | 0.37647 | 0.07451 | 22.647 | 0.914798 | 0.87451 |
| 0.92549 | 0.42745 | 0.12549 | 22.647 | 0.864407 | 0.92549 |
| 0.92549 | 0.46275 | 0.12549 | 25.2945 | 0.864407 | 0.92549 |
| 0.87451 | 0.41569 | 0.07451 | 25.5885 | 0.914798 | 0.87451 |
| 0.87451 | 0.45098 | 0.07451 | 28.2352 | 0.914798 | 0.87451 |
| 0.92549 | 0.50196 | 0.12549 | 28.2352 | 0.864407 | 0.92549 |
| 0.87451 | 0.48627 | 0.07451 | 30.882 | 0.914798 | 0.87451 |
| 0.92549 | 0.53725 | 0.12549 | 30.882 | 0.864407 | 0.92549 |
| 0.92549 | 0.57255 | 0.12549 | 33.5295 | 0.864407 | 0.92549 |
| 0.87451 | 0.52549 | 0.07451 | 33.8235 | 0.914798 | 0.87451 |
| 0.87451 | 0.56078 | 0.07451 | 36.4702 | 0.914798 | 0.87451 |
| 0.92549 | 0.61176 | 0.12549 | 36.4702 | 0.864407 | 0.92549 |
| 0.87451 | 0.6 | 0.07451 | 39.4118 | 0.914798 | 0.87451 |
| 0.92549 | 0.65098 | 0.12549 | 39.4118 | 0.864407 | 0.92549 |
| 0.87451 | 0.63529 | 0.07451 | 42.0585 | 0.914798 | 0.87451 |
| 0.92549 | 0.68627 | 0.12549 | 42.0585 | 0.864407 | 0.92549 |

Table A.2: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (17-32)

| R | G | B | H | S | V |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.87451 | 0.67451 | 0.07451 | 45 | 0.914798 | 0.87451 |
| 0.92549 | 0.72549 | 0.12549 | 45 | 0.864407 | 0.92549 |
| 0.92549 | 0.76078 | 0.12549 | 47.6467 | 0.864407 | 0.92549 |
| 0.87451 | 0.71373 | 0.07451 | 47.9415 | 0.914798 | 0.87451 |
| 0.87451 | 0.74902 | 0.07451 | 50.5882 | 0.914798 | 0.87451 |
| 0.92549 | 0.8 | 0.12549 | 50.5882 | 0.864407 | 0.92549 |
| 0.87451 | 0.78824 | 0.07451 | 53.5298 | 0.914798 | 0.87451 |
| 0.92549 | 0.83922 | 0.12549 | 53.5298 | 0.864407 | 0.92549 |
| 0.87451 | 0.82353 | 0.07451 | 56.1765 | 0.914798 | 0.87451 |
| 0.92549 | 0.87451 | 0.12549 | 56.1765 | 0.864407 | 0.92549 |
| 0.87451 | 0.86275 | 0.07451 | 59.118 | 0.914798 | 0.87451 |
| 0.92549 | 0.91373 | 0.12549 | 59.118 | 0.864407 | 0.92549 |
| 0.85098 | 0.87451 | 0.07451 | 61.7648 | 0.914798 | 0.87451 |
| 0.90196 | 0.92549 | 0.12549 | 61.7648 | 0.864407 | 0.92549 |
| 0.86275 | 0.92549 | 0.12549 | 64.7055 | 0.864407 | 0.92549 |
| 0.81176 | 0.87451 | 0.07451 | 64.7062 | 0.914798 | 0.87451 |

Table A.3: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (33-48)

| R | G | B | H | S | V |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.77647 | 0.87451 | 0.07451 | 67.353 | 0.914798 | 0.87451 |
| 0.82353 | 0.92549 | 0.12549 | 67.647 | 0.864407 | 0.92549 |
| 0.78824 | 0.92549 | 0.12549 | 70.2937 | 0.864407 | 0.92549 |
| 0.73725 | 0.87451 | 0.07451 | 70.2945 | 0.914798 | 0.87451 |
| 0.69804 | 0.87451 | 0.07451 | 73.2353 | 0.914798 | 0.87451 |
| 0.74902 | 0.92549 | 0.12549 | 73.2353 | 0.864407 | 0.92549 |
| 0.66275 | 0.87451 | 0.07451 | 75.882 | 0.914798 | 0.87451 |
| 0.71373 | 0.92549 | 0.12549 | 75.882 | 0.864407 | 0.92549 |
| 0.62353 | 0.87451 | 0.07451 | 78.8235 | 0.914798 | 0.87451 |
| 0.67451 | 0.92549 | 0.12549 | 78.8235 | 0.864407 | 0.92549 |
| 0.58824 | 0.87451 | 0.07451 | 81.4702 | 0.914798 | 0.87451 |
| 0.63529 | 0.92549 | 0.12549 | 81.765 | 0.864407 | 0.92549 |
| 0.54902 | 0.87451 | 0.07451 | 84.4118 | 0.914798 | 0.87451 |
| 0.6 | 0.92549 | 0.12549 | 84.4118 | 0.864407 | 0.92549 |
| 0.5098 | 0.87451 | 0.07451 | 87.3532 | 0.914798 | 0.87451 |
| 0.56078 | 0.92549 | 0.12549 | 87.3533 | 0.864407 | 0.92549 |

Table A.4: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (49-64)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.47843 | 0.87451 | 0.07451 | 89.706 | 0.914798 | 0.87451 |
| 0.52549 | 0.92549 | 0.12549 | 90 | 0.864407 | 0.92549 |
| 0.43922 | 0.87451 | 0.07451 | 92.6467 | 0.914798 | 0.87451 |
| 0.4902 | 0.92549 | 0.12549 | 92.6467 | 0.864407 | 0.92549 |
| 0.4 | 0.87451 | 0.07451 | 95.5882 | 0.914798 | 0.87451 |
| 0.45098 | 0.92549 | 0.12549 | 95.5882 | 0.864407 | 0.92549 |
| 0.36471 | 0.87451 | 0.07451 | 98.235 | 0.914798 | 0.87451 |
| 0.41569 | 0.92549 | 0.12549 | 98.235 | 0.864407 | 0.92549 |
| 0.32549 | 0.87451 | 0.07451 | 101.176 | 0.914798 | 0.87451 |
| 0.37647 | 0.92549 | 0.12549 | 101.176 | 0.864407 | 0.92549 |
| 0.2902 | 0.87451 | 0.07451 | 103.823 | 0.914798 | 0.87451 |
| 0.33725 | 0.92549 | 0.12549 | 104.118 | 0.864407 | 0.92549 |
| 0.25098 | 0.87451 | 0.07451 | 106.765 | 0.914798 | 0.87451 |
| 0.30196 | 0.92549 | 0.12549 | 106.765 | 0.864407 | 0.92549 |
| 0.21176 | 0.87451 | 0.07451 | 109.706 | 0.914798 | 0.87451 |
| 0.26275 | 0.92549 | 0.12549 | 109.706 | 0.864407 | 0.92549 |

Table A.5: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (65-80)

| R | G | B | H | S | V |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.17647 | 0.87451 | 0.07451 | 112.353 | 0.914798 | 0.87451 |
| 0.22745 | 0.92549 | 0.12549 | 112.353 | 0.864407 | 0.92549 |
| 0.13725 | 0.87451 | 0.07451 | 115.294 | 0.914798 | 0.87451 |
| 0.18824 | 0.92549 | 0.12549 | 115.294 | 0.864407 | 0.92549 |
| 0.10196 | 0.87451 | 0.07451 | 117.941 | 0.914798 | 0.87451 |
| 0.14902 | 0.92549 | 0.12549 | 118.235 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.08627 | 120.882 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.13725 | 120.882 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.12549 | 123.824 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.17647 | 123.824 | 0.864407 | 0.92549 |
| 0.12549 | 0.92549 | 0.21176 | 126.47 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.16471 | 126.765 | 0.914798 | 0.87451 |
| 0.07451 | 0.87451 | 0.2 | 129.412 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.25098 | 129.412 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.23922 | 132.353 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.2902 | 132.353 | 0.864407 | 0.92549 |

Table A.6: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (81-96)

| $R$ | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.07451 | 0.87451 | 0.27451 | 135 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.32549 | 135 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.31373 | 137.941 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.36471 | 137.941 | 0.864407 | 0.92549 |
| 0.12549 | 0.92549 | 0.4 | 140.588 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.35294 | 140.882 | 0.914798 | 0.87451 |
| 0.07451 | 0.87451 | 0.38824 | 143.53 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.43922 | 143.53 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.42745 | 146.471 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.47843 | 146.471 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.46275 | 149.118 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.51373 | 149.118 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.49804 | 151.765 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.54902 | 151.765 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.53725 | 154.705 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.58824 | 154.706 | 0.864407 | 0.92549 |

Table A.7: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (97-112)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.07451 | 0.87451 | 0.57255 | 157.353 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.62353 | 157.353 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.61176 | 160.294 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.66275 | 160.295 | 0.864407 | 0.92549 |
| 0.12549 | 0.92549 | 0.69804 | 162.941 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.65098 | 163.235 | 0.914798 | 0.87451 |
| 0.07451 | 0.87451 | 0.68627 | 165.882 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.73725 | 165.882 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.72549 | 168.824 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.77647 | 168.824 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.76078 | 171.47 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.81176 | 171.47 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.8 | 174.412 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.85098 | 174.412 | 0.864407 | 0.92549 |
| 0.12549 | 0.92549 | 0.88627 | 177.059 | 0.864407 | 0.92549 |
| 0.07451 | 0.87451 | 0.83922 | 177.353 | 0.914798 | 0.87451 |

Table A.8: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (113-128)

| $R$ | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.07451 | 0.87451 | 0.87451 | 180 | 0.914798 | 0.87451 |
| 0.12549 | 0.92549 | 0.92549 | 180 | 0.864407 | 0.92549 |
| 0.07451 | 0.83922 | 0.87451 | 182.647 | 0.914798 | 0.87451 |
| 0.12549 | 0.88627 | 0.92549 | 182.941 | 0.864407 | 0.92549 |
| 0.07451 | 0.8 | 0.87451 | 185.588 | 0.914798 | 0.87451 |
| 0.12549 | 0.85098 | 0.92549 | 185.588 | 0.864407 | 0.92549 |
| 0.07451 | 0.76078 | 0.87451 | 188.53 | 0.914798 | 0.87451 |
| 0.12549 | 0.81176 | 0.92549 | 188.53 | 0.864407 | 0.92549 |
| 0.07451 | 0.72549 | 0.87451 | 191.176 | 0.914798 | 0.87451 |
| 0.12549 | 0.77647 | 0.92549 | 191.176 | 0.864407 | 0.92549 |
| 0.07451 | 0.68627 | 0.87451 | 194.118 | 0.914798 | 0.87451 |
| 0.12549 | 0.73725 | 0.92549 | 194.118 | 0.864407 | 0.92549 |
| 0.07451 | 0.65098 | 0.87451 | 196.765 | 0.914798 | 0.87451 |
| 0.12549 | 0.69804 | 0.92549 | 197.059 | 0.864407 | 0.92549 |
| 0.12549 | 0.66275 | 0.92549 | 199.705 | 0.864407 | 0.92549 |
| 0.07451 | 0.61176 | 0.87451 | 199.706 | 0.914798 | 0.87451 |
| Ta A 9. | $00 r$ | U 98 |  |  |  |

Table A.9: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (129-144)

| R | G | B | H | S | V |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.07451 | 0.57255 | 0.87451 | 202.647 | 0.914798 | 0.87451 |
| 0.12549 | 0.62353 | 0.92549 | 202.647 | 0.864407 | 0.92549 |
| 0.12549 | 0.58824 | 0.92549 | 205.294 | 0.864407 | 0.92549 |
| 0.07451 | 0.53725 | 0.87451 | 205.295 | 0.914798 | 0.87451 |
| 0.07451 | 0.49804 | 0.87451 | 208.235 | 0.914798 | 0.87451 |
| 0.12549 | 0.54902 | 0.92549 | 208.235 | 0.864407 | 0.92549 |
| 0.07451 | 0.46275 | 0.87451 | 210.882 | 0.914798 | 0.87451 |
| 0.12549 | 0.51373 | 0.92549 | 210.882 | 0.864407 | 0.92549 |
| 0.07451 | 0.42745 | 0.87451 | 213.529 | 0.914798 | 0.87451 |
| 0.12549 | 0.47843 | 0.92549 | 213.529 | 0.864407 | 0.92549 |
| 0.07451 | 0.38824 | 0.87451 | 216.47 | 0.914798 | 0.87451 |
| 0.12549 | 0.43922 | 0.92549 | 216.47 | 0.864407 | 0.92549 |
| 0.07451 | 0.35294 | 0.87451 | 219.118 | 0.914798 | 0.87451 |
| 0.12549 | 0.4 | 0.92549 | 219.412 | 0.864407 | 0.92549 |
| 0.07451 | 0.31373 | 0.87451 | 222.059 | 0.914798 | 0.87451 |
| 0.12549 | 0.36471 | 0.92549 | 222.059 | 0.864407 | 0.92549 |

Table A.10: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (145-160)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.07451 | 0.27451 | 0.87451 | 225 | 0.914798 | 0.87451 |
| 0.12549 | 0.32549 | 0.92549 | 225 | 0.864407 | 0.92549 |
| 0.07451 | 0.23922 | 0.87451 | 227.647 | 0.914798 | 0.87451 |
| 0.12549 | 0.2902 | 0.92549 | 227.647 | 0.864407 | 0.92549 |
| 0.07451 | 0.2 | 0.87451 | 230.588 | 0.914798 | 0.87451 |
| 0.12549 | 0.25098 | 0.92549 | 230.588 | 0.864407 | 0.92549 |
| 0.07451 | 0.16471 | 0.87451 | 233.235 | 0.914798 | 0.87451 |
| 0.12549 | 0.21176 | 0.92549 | 233.53 | 0.864407 | 0.92549 |
| 0.07451 | 0.12549 | 0.87451 | 236.176 | 0.914798 | 0.87451 |
| 0.12549 | 0.17647 | 0.92549 | 236.176 | 0.864407 | 0.92549 |
| 0.07451 | 0.08627 | 0.87451 | 239.118 | 0.914798 | 0.87451 |
| 0.12549 | 0.13725 | 0.92549 | 239.118 | 0.864407 | 0.92549 |
| 0.14902 | 0.12549 | 0.92549 | 241.765 | 0.864407 | 0.92549 |
| 0.10196 | 0.07451 | 0.87451 | 242.059 | 0.914798 | 0.87451 |
| 0.13725 | 0.07451 | 0.87451 | 244.705 | 0.914798 | 0.87451 |
| 0.18824 | 0.12549 | 0.92549 | 244.706 | 0.864407 | 0.92549 |

Table A.11: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (161-176)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.17647 | 0.07451 | 0.87451 | 247.647 | 0.914798 | 0.87451 |
| 0.22745 | 0.12549 | 0.92549 | 247.647 | 0.864407 | 0.92549 |
| 0.21176 | 0.07451 | 0.87451 | 250.294 | 0.914798 | 0.87451 |
| 0.26275 | 0.12549 | 0.92549 | 250.295 | 0.864407 | 0.92549 |
| 0.25098 | 0.07451 | 0.87451 | 253.235 | 0.914798 | 0.87451 |
| 0.30196 | 0.12549 | 0.92549 | 253.235 | 0.864407 | 0.92549 |
| 0.33725 | 0.12549 | 0.92549 | 255.882 | 0.864407 | 0.92549 |
| 0.2902 | 0.07451 | 0.87451 | 256.177 | 0.914798 | 0.87451 |
| 0.32549 | 0.07451 | 0.87451 | 258.823 | 0.914798 | 0.87451 |
| 0.37647 | 0.12549 | 0.92549 | 258.823 | 0.864407 | 0.92549 |
| 0.36471 | 0.07451 | 0.87451 | 261.765 | 0.914798 | 0.87451 |
| 0.41569 | 0.12549 | 0.92549 | 261.765 | 0.864407 | 0.92549 |
| 0.4 | 0.07451 | 0.87451 | 264.412 | 0.914798 | 0.87451 |
| 0.45098 | 0.12549 | 0.92549 | 264.412 | 0.864407 | 0.92549 |
| 0.43922 | 0.07451 | 0.87451 | 267.353 | 0.914798 | 0.87451 |
| 0.4902 | 0.12549 | 0.92549 | 267.353 | 0.864407 | 0.92549 |

Table A.12: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (177-192)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.52549 | 0.12549 | 0.92549 | 270 | 0.864407 | 0.92549 |
| 0.47843 | 0.07451 | 0.87451 | 270.294 | 0.914798 | 0.87451 |
| 0.5098 | 0.07451 | 0.87451 | 272.647 | 0.914798 | 0.87451 |
| 0.56078 | 0.12549 | 0.92549 | 272.647 | 0.864407 | 0.92549 |
| 0.54902 | 0.07451 | 0.87451 | 275.588 | 0.914798 | 0.87451 |
| 0.6 | 0.12549 | 0.92549 | 275.588 | 0.864407 | 0.92549 |
| 0.63529 | 0.12549 | 0.92549 | 278.235 | 0.864407 | 0.92549 |
| 0.58824 | 0.07451 | 0.87451 | 278.53 | 0.914798 | 0.87451 |
| 0.62353 | 0.07451 | 0.87451 | 281.177 | 0.914798 | 0.87451 |
| 0.67451 | 0.12549 | 0.92549 | 281.177 | 0.864407 | 0.92549 |
| 0.66275 | 0.07451 | 0.87451 | 284.118 | 0.914798 | 0.87451 |
| 0.71373 | 0.12549 | 0.92549 | 284.118 | 0.864407 | 0.92549 |
| 0.69804 | 0.07451 | 0.87451 | 286.765 | 0.914798 | 0.87451 |
| 0.74902 | 0.12549 | 0.92549 | 286.765 | 0.864407 | 0.92549 |
| 0.73725 | 0.07451 | 0.87451 | 289.706 | 0.914798 | 0.87451 |
| 0.78824 | 0.12549 | 0.92549 | 289.706 | 0.864407 | 0.92549 |

Table A.13: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (193-208)

| R | G | B | H | S | V |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0.82353 | 0.12549 | 0.92549 | 292.353 | 0.864407 | 0.92549 |
| 0.77647 | 0.07451 | 0.87451 | 292.647 | 0.914798 | 0.87451 |
| 0.81176 | 0.07451 | 0.87451 | 295.294 | 0.914798 | 0.87451 |
| 0.86275 | 0.12549 | 0.92549 | 295.294 | 0.864407 | 0.92549 |
| 0.85098 | 0.07451 | 0.87451 | 298.235 | 0.914798 | 0.87451 |
| 0.90196 | 0.12549 | 0.92549 | 298.235 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.86275 | 300.882 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.91373 | 300.882 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.82353 | 303.823 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.87451 | 303.823 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.78824 | 306.47 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.83922 | 306.47 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.74902 | 309.412 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.8 | 309.412 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.71373 | 312.059 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.76078 | 312.353 | 0.864407 | 0.92549 |

Table A.14: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (209-224)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.87451 | 0.07451 | 0.67451 | 315 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.72549 | 315 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.63529 | 317.941 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.68627 | 317.941 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.6 | 320.588 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.65098 | 320.588 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.56078 | 323.53 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.61176 | 323.53 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.52549 | 326.177 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.57255 | 326.47 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.48627 | 329.118 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.53725 | 329.118 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.45098 | 331.765 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.50196 | 331.765 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.41569 | 334.411 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.46275 | 334.706 | 0.864407 | 0.92549 |

Table A.15: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (225-240)

| R | G | B | H | S | V |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0.87451 | 0.07451 | 0.37647 | 337.353 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.42745 | 337.353 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.33725 | 340.294 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.38824 | 340.294 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.30196 | 342.941 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.35294 | 342.941 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.26275 | 345.882 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.31373 | 345.882 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.22745 | 348.53 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.27451 | 348.823 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.18824 | 351.47 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.23922 | 351.47 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.14902 | 354.412 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.2 | 354.412 | 0.864407 | 0.92549 |
| 0.87451 | 0.07451 | 0.11373 | 357.059 | 0.914798 | 0.87451 |
| 0.92549 | 0.12549 | 0.16471 | 357.059 | 0.864407 | 0.92549 |

Table A.16: Color Map Used for Retinotpic Data. 256 RGB colors converted to HSV and Sorted. Colors (241-256)

