**Title**

fMRI visualization and methods

**Author**

Gao, James S.

**Publication Date**

2015

# fMRI visualization and methods

By

James Shuang Gao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Vision Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Jack Gallant, Chair
Professor Jose Carmena
Professor Bruno Olshausen

Spring 2015

fMRI visualization and methods

# Abstract

fMRI visualization and methods

by

James Shuang Gao

Doctor of Philosophy in Vision Science

University of California, Berkeley

Professor Jack Gallant, Chair

fMRI is a proven technique for recording brain activity. Although it measures a signal that is coupled with neural firing through a complex chain of events, it is the only method by which we can non-invasively observe an entire healthy brain at millimeter scale resolution. The technique of fMRI is now fairly well known, yet there are still many gaps in the tools we use to analyze it and our understanding of its underpinnings. Indeed, the complexity and depth of the data is the source of many of its shortcomings.

In this dissertation, I detail a set of tools I developed to make better use of fMRI data. In chapter 2, I describe a tool that I created to visualize fMRI data. It provides an intuitive and powerful interface to explore and share your data with others. In chapter 3, I created a device that significantly improves the performance of fMRI by virtually eliminating head motion. This plastic insert relies on commodity 3D printing technology to create personalized helmets that address many of the shortfalls of fast scan sequences. Finally, in chapter 4, I utilize these fast scan sequences to investigate the temporal response of hemodynamics. I show that BOLD timecourses have very little high frequency power, but the high dimensional nature of fMRI can be used to perform classification extremely quickly.

I dedicate this thesis to my dad,
高中立, Zhongli Gao
He dug his way from the Loess
to provide these opportunities for me.

# Contents

**List of Figures** **iv**

**Acknowledgments** **v**

**1 Introduction** **1**

1.1 Prologue . . . . . . . . . . 1

1.2 Function Magnetic Resonance Imaging . . . . . . . . . . 2

1.3 Future research . . . . . . . . . . 3

1.4 Overview . . . . . . . . . . 4

**2 Pycortex: an interactive surface visualizer for fMRI** **5**

2.1 Introduction . . . . . . . . . . 5

2.2 Background . . . . . . . . . . 7

2.2.1 Cortical surface mesh generation . . . . . . . . . . 7

2.2.2 Coregistration . . . . . . . . . . 8

2.2.3 Projection of functional data. . . . . . . . . . . 9

2.3 Innovations in pycortex . . . . . . . . . . 10

2.3.1 Surface generation and coregistration . . . . . . . . . . 10

2.3.2 Pixel-based mapping . . . . . . . . . . 11

2.3.3 WebGL and data sharing . . . . . . . . . . 13

2.4 Pycortex functionality . . . . . . . . . . 14

2.4.1 The segment module . . . . . . . . . . 15

2.4.2 The align module . . . . . . . . . . 17

2.4.3 The quickflat and webgl modules . . . . . . . . . . 17

2.4.4 The overlays module . . . . . . . . . . 20

2.5 Future development . . . . . . . . . . 22

2.6 Acknowledgements . . . . . . . . . . 22

**3 Reducing head motion during fMRI by means of a personalized 3D-printed insert** **24**

3.1 Introduction . . . . . . . . . . 24

3.2 Materials & Methods . . . . . . . . . . 25

3.2.1 Insert design & construction . . . . . . . . . . . . . . . . . . . . 25
3.2.2 MRI methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
3.2.3 Subjects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.2.4 fMRI data analysis . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

**4 Fast fMRI improves signal quality by separating signal and noise spectra 36**
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36
4.2 Methods . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.2.1 Experimental and Stimulus Design . . . . . . . . . . . . . . . . . 37
4.2.2 fMRI Acquisition . . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.2.3 fMRI Preprocessing . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.2.4 Subjects . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.2.5 fMRI Analyses . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.3 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
4.3.1 Total Explainable Variance . . . . . . . . . . . . . . . . . . . . . 41
4.3.2 Coherence spectra . . . . . . . . . . . . . . . . . . . . . . . . . 44
4.3.3 Timepoint correlation . . . . . . . . . . . . . . . . . . . . . . . . 47
4.3.4 High speed stimulus . . . . . . . . . . . . . . . . . . . . . . . . . 49
4.3.5 Physiological noise . . . . . . . . . . . . . . . . . . . . . . . . . 51
4.4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
4.4.1 Timepoint classification . . . . . . . . . . . . . . . . . . . . . . . 51
4.4.2 Acceleration errors . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

**5 Conclusion 55**

**Bibliography 57**

# List of Figures

2.1 Typical fMRI Visualization methods . . . . . 6
2.2 Standard vertex-based mapping . . . . . 9
2.3 Typical fMRI data Projections. . . . . . 12
2.4 Retinotopic data for example subject. . . . . . 13
2.5 Pycortex module diagram. . . . . . 15
2.6 High quality flatmaps generated by pycortex. . . . . . 18
2.7 Static view for web presentation. . . . . . 21

3.1 3D scanning and reconstruction of the space inside the MRI head coil . . . . . 26
3.2 Anatomical scan and reconstruction of the outside surface of the subject's head 27
3.3 Final insert design . . . . . 27
3.4 Fused deposition 3D printer . . . . . 28
3.5 Completed insert . . . . . 29
3.6 Estimated motion parameters with and without the insert . . . . . 32
3.7 Summary of motion parameters . . . . . 33
3.8 Estimated fSNR with and without the insert . . . . . 34

4.1 fMRI Preprocessing Pipeline . . . . . 39
4.2 Total explainable variance . . . . . 42
4.3 Coherence, Signal, and Noise spectra . . . . . 43
4.4 Filtered downsampling vs. subsampling . . . . . 45
4.5 Filtered explainable variance . . . . . 46
4.6 Timepoint correlation . . . . . 48
4.7 Optimal Filter . . . . . 49
4.8 High speed Stimulus . . . . . 50
4.9 Physiological noise . . . . . 52

# Acknowledgments

First, I would like to thank my advisor, Jack Gallant. Although he can be difficult to work with at times, he truly cares about his students. He has been an excellent mentor during my graduate career, providing guidance when I need it, and flexibility when I want to pursue my own goals. Thank you, Jack!

I would not have made it very far in graduate school without my labmates. They have been teachers and companions, commiserators and critics, skeptics and supporters. Alex, thanks for being such an awesome collaborator. My success can only be blamed on you. Your ideas are in every corner of this thesis, and your name will be on all the papers. Natalia, you've always been such a huge supporter of all the random stuff I do, thank you! Mark, our scientific (and non-scientific) debates are always a highlight. Thank you for being such a supporter. To all the rest of my members of Gallant lab, Anwar, Lydia, Storm, Mike, Fatma, Dustin, Shinji, Thomas, and Joe, thank you all for making my graduate career great!

To my awesome roommates, Adam, Mariana, Chat, I've missed you for the last year! We had an awesome time at 955 Oxford. You've always been there for me when I needed the support. I will always miss our wine nights and rad parties. I hope that the next time we feel compelled to read this acknowledgement, it will be from a wing in Castle 955.

To my parents, who have always encouraged me to strive for the best. This thesis is dedicated to my dad, who literally dug his way out of Loess caves in China. He knew the only way out was to study hard, and ensured that I share his ideals. To my mother, a strong woman who followed my dad through political unrest into a foreign country and has thrived. To my little sister Catherine, who I've watched grow from a young child to a college freshman during grad school. You'll always be 8 in my head!

Finally, to my lovely girlfriend Helene. You've seen all the ups and downs of grad school, and you've always been there for me. Whether it's scientific debate or biking down the coast of California, you've followed me everywhere. I am so lucky to have met you during grad school, and I am the luckiest person alive to share this time together.

# Chapter 1

# Introduction

## 1.1 Prologue

The brain remains one of final frontiers in our understanding of the natural world. Understanding the function of the brain would allow us to understand and predict animal and human behaviors. Unfortunately, it is also one of the most difficult things to study in natural science. Unlike other fields, the fundamental processes that govern the operation of the brain are well known. The anatomical description of neurons were well documented by scientists such as Santiago Ramón y Cajal as early as 1890. Mathematical models of neural function were detailed in Hodgkin and Huxley (1952). Despite the extensive understanding of the neural subunits, we still do not have a full predictive model of the brain and the origin of behavior.

Much of this fault lies with the organization of neurons that form the brain. Neurons form complex, highly interconnected networks that are nearly impossible to segment and track. Even with the power of today's computers, only small segments of neural tissue have been fully characterized at an anatomical level. Electrophysiology has been an important tool in understand subsystems in the brain, since it measures function rather than anatomy. However, electrophysiology is fundamentally limited by sampling. Sinking individual needles into the brain allows the modeling of individual neurons. Such individual recordings are unable to provide insight into more global organization.

Fundamentally, neuroscience research has been limited by instrumentation. At one end, electrophysiology provides deep insight into very small neural systems. Before 1990, the global behavior of the brain could only be studied through psychophysics and behavior. Without the necessary instrumentation, the global organization of the brain and how it interfaces with behavior remained a mystery.

In 1990, a pivotal advancement in instrumentation finally provided a tool which allowed us to study the global organization of the brain. Seiji Ogawa's landmark paper, "Brain magnetic resonance imaging with contrast dependent on blood oxygenation." (Ogawa et al. 1990) introduced the direct imaging of brain function at a global scale with functional magnetic resonance imaging. Prior to 1990, MRI had been used for noninvasive imaging of anatomy

(Lauterbur 1973). Ogawa's key contribution was that the oxygenation state of the blood could be measured with MRI. Combined with what was previously known about hemodynamic response (Roy and Sherrington 1890), it became possible to noninvasively study the function of the brain at a global scale.

## 1.2 Function Magnetic Resonance Imaging

Since the invention of fMRI, the technique has revolutionized the field of neuroscience. fMRI has allowed us to peer into the inner workings of the brain (Friston et al. 1996; Huettel et al. 2009). Important discoveries have been made by fMRI such as the existence of functional subdomains (Kanwisher et al. 1997) and validation of neural models at scale (Wandell 1999; Nishimoto et al. 2011). However, there are many technical challenges when using fMRI to study the brain.

On an extremely practical level, MRI measures tissue properties at a grid of points (voxels) in a volume. By recording individual snapshots in time, we add an additional time dimension that further obfuscates the data. The visualization of 3D and 4D data has typically been addressed by simply marginalizing across dimensions. In other words, we look at 2D slices of the data. This simple visualization is easy to generate, but difficult to interpret and understand (Figure 2.1). This is because the brain is not organized along 2D sheets, but is instead a folded 2D manifold in 3D space (Van Essen et al. 2001; Felleman and Van Essen 1991; Kaas 2012).

To address this issue, researchers interested in function of the cortex typically segment the cortical mantle using high resolution anatomical images (Van Essen et al. 2001; Dale et al. 1999). This segmentation is processed with computational geometry tools to generate triangular mesh models of cortex. This mesh model can be used to select voxels for analysis, and results can be plotted across this surface (Fischl et al. 2001; Cox 1996). Even with a mesh model, visualization is still a challenge. Folds in the cortex obscure data plotted deep in sulci, and alternate views such as inflated or flattened surfaces introduce distortions. However, modern 3D rendering algorithms enable interactive exploration of data, a technique that is underutilized in science (chapter 2).

At a more fundamental level, fMRI relies on changes in a tissue's magnetic properties to infer activation. The largest and most robust change that naturally occurs in the brain is the hemodynamic response (Roy and Sherrington 1890). When neurons fire, there is a transient and local burst of metabolic activity that triggers vascular dilation and subsequent reoxygenation (Logothetis et al. 2001; Logothetis 2003). This change in blood oxygenation can be detected by functional MRI. By choosing a TE that maximizes the T2* dephasing effect, oxygen-bound hemoglobin creates changes in tissue contrast that is visible in MRI (Nishimura 2010). In summary, neural firing leads to a small metabolic change that is visible in a tailored MRI sequence. This complex chain of events results in a measurable response peaking 4-8 s after neural firing (Glover 1999; Buxton et al. 2004).

Thankfully, the link between neural firing and hemodynamic response has been well established (Logothetis et al. 2001; Yablonskiy and Haacke 1994; Mukamel et al. 2005; Issa et al. 2013; Heeger and Ress 2002). Many papers have studied the shape of the hemodynamic response function that results from transient stimulation of the brain (Glover 1999; Buxton et al. 2004), and it is very evident that the hemodynamic response function (HRF) applies some type of low-pass filter to underlying neural activity (Logothetis 2008). However, the limits of the temporal frequency space have never been probed until now (chapter 4).

In the last 10 years, a key innovation in MRI acquisition allowed us to measure changes in oxygenation at these high frequency bands. By utilizing coil sensitivity maps across multiple receive channels, it is possible to separate images that have been intentionally aliased together (Larkman et al. 2001; Setsompop et al. 2012). This technique known as "multiband" imaging has moved fMRI far beyond the realm of 0.25 Hz sampling rates. Multiband allows nearly loss-free acceleration of acquisition, however it is not without its downsides. This imaging technique relies on having a static map of coil sensitivities; any disturbance results in incomplete image separation and signal misattribution (Larkman et al. 2001). Unfortunately, awake behaving subjects typically move around, which is exactly what disturbs these maps of coil sensitivities. However, this problem can again be solved with innovative technical solutions (chapter 3).

These innovations in fMRI have allowed us to study the link between brain and behavior at an unprecedented scale. The volumetric measurements provide extremely high dimensional data that spans the entire brain. Our understanding of the link between hemodynamics and neural function allow us to meld the information gleaned using electrophysiology with the measurements taken in fMRI. These technological advances in fMRI have resulted in predictive models of behavior and function, one of the end goals of neuroscience research.

## 1.3 Future research

Now that predictive models of the brain exist, one tantalizing application of these models is that of real time feedback and control. With the right model of brain activity, it may be possible to decode covert intention in a behaving subject. For a healthy subject, real time decoding would merely be a very interesting parlour trick. The science behind accurate decoding can be tested equally well with encoding models (Wu et al. 2006). For subjects with severe disabilities such as locked-in syndrome or late-stage ALS, such an application would restore function and would likely prove life-changing. Further innovation in brain recording techniques could provide neural decoding in a more portable and accessible form.

An appealing early use case for real-time feedback would be for movement control. In Andersson et al. (2013), a subject within a magnet used covert spatial attention shifts to drive a small robot car around an arena. While this was an excellent proof of concept for real-time control, it utilized a slow 20 second block design to maximize the decoded signal. With better models of spatial representation and the latest multiband sequences, it may be possible to provide more naturalistic control with a much shorter feedback cycle.

## 1.4 Overview

In this thesis, I laid the groundwork for a future real-time decoding experiment. I developed tools for the visualization of fMRI data and streamlined the process of voxel selection and display (chapter 2). I tested new fast sequences that will allow better real-time feedback (chapter 4). Finally, I developed a device that will maximize the SNR from these accelerated sequences (chapter 3). It is my hope that these methods that I have developed can be used with a real-time decoding experiment in fMRI in the future.

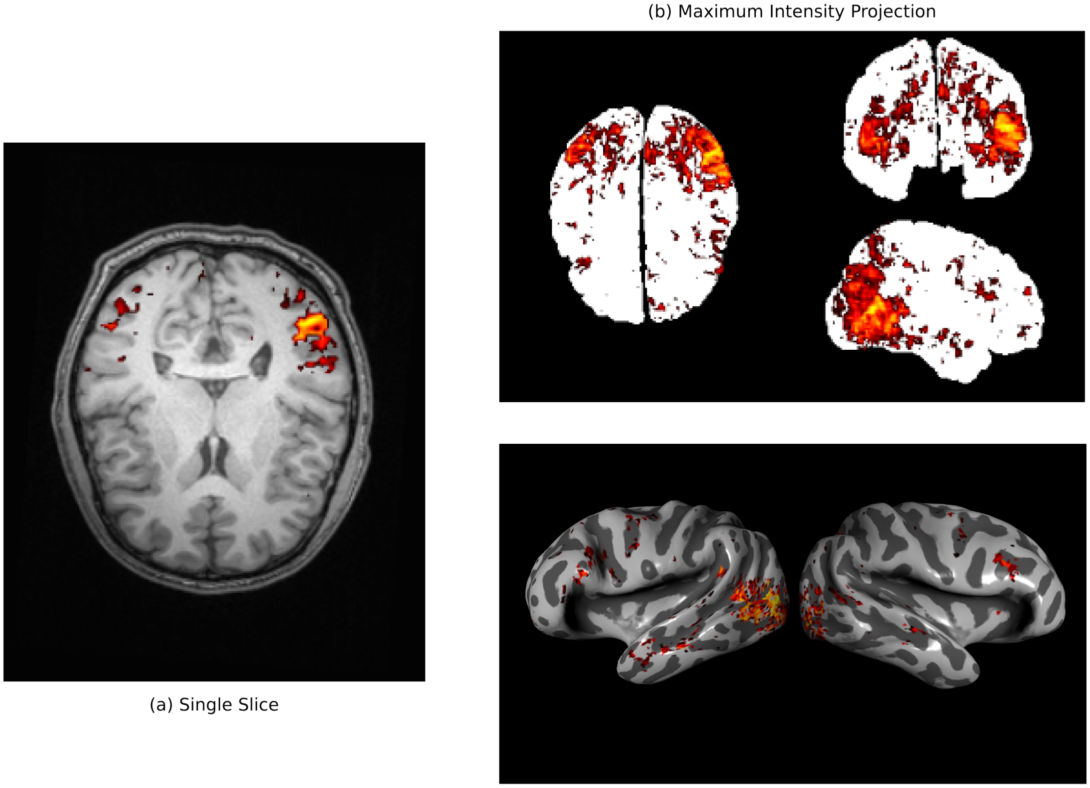
# Chapter 2

# Pycortex: an interactive surface visualizer for fMRI

## 2.1 Introduction

Functional magnetic resonance imaging (fMRI) experiments produce rich data revealing the patterns of hemodynamic activity throughout the brain (Huettel et al. 2009). However, tools for visualization of fMRI data remain relatively primitive. Volumetric views that show single slices or maximum intensity projections (Figure 2.1) reveal only a small portion of the available data. More sophisticated tools use 3D reconstructions of the cortical surface to create inflated or flattened cortical surfaces (Cox 1996; Dale et al. 1999; Goebel 1997; Van Essen et al. 2001). However, most of these tools produce static views of the data so it is often difficult to interpret the relationship between cortical anatomy and inflated and flattened surfaces. Furthermore, current packages use standard computer graphics libraries that are not optimized for accurate visualization of volume projections. They tend to undersample the underlying volumetric data and do not produce optimal visualizations. Finally, no current visualization packages provide a convenient platform for creating interactive online visualizations for a broad audience.

We addressed the shortcomings of 3D visualization tools by developing pycortex, an interactive software package for viewing fMRI data that is optimized for displaying data on the cortical surface. Pycortex streamlines the process of surface visualization and produces interactive displays that switch smoothly between anatomically correct (wrinkled), inflated, and flattened views of the cortical surface. Pycortex implements a pixel-wise mapping algorithm for projecting volumetric data onto the cortex. This method samples the underlying functional data densely and produces accurate, visually appealing renderings of the fMRI data. Finally, pycortex uses WebGL to display the results of the analysis. These WebGL-based visualizations can be created and viewed on the fly, or they can be saved as a web page that can be viewed by anyone with a modern web browser. These standalone visualizations can easily be shared with colleagues, included as links in published articles, or shared online with a broad audience.



(b) Maximum Intensity Projection

(a) Single Slice

(c) Inflated Surface

*Figure 2.1*: *Typical fMRI Visualization methods*
Three typical methods for visualizing fMRI data are used here to visualize a single data set (Huth et al. 2012). (a) A single axial slice from an anatomical image is shown overlain with functional data exceeding statistical threshold. It is difficult to recognize anatomical features in this view and much of the functional data is hidden. (b) A maximum intensity projection (i.e., a glass brain view) is shown along with all functional data that exceeds statistical threshold. This view shows more of the functional data than can be seen in the single slice but the anatomical location of these signals is obscured. (c) An inflated cortical surface is shown with curvature highlighted in grayscale, and overlain with functional data exceeding statistical threshold. The anatomical location of the functional data is clearer than in the other views, but multiple views are required to see all of the data. None of these standard visualizations show all of the data in a succinct and interpretable way.

To see a demonstration of what can be achieved with pycortex, please visit http://www.gallantlab.org/brainviewer/retinotopy_demo. This demo shows the results of a retinotopic mapping fMRI experiment that was performed on one subject. Retinotopic mapping is one of the workhorse tools in fMRI experiments of human vision, and it is used to identify the cortical extent and distribution of many human visual areas (Engel et al. 1997; Sereno 1998; Sereno et al. 1995). In this online demo, the retinotopic mapping stimulus that the subject saw appears on the right, and measured blood oxygenation level-dependent (BOLD) responses measured across the cortical sheet are shown at left.

The rest of this paper is divided into three sections. The first section describes the advantages of cortical surface-based analysis and visualization over other methods. The second explains the innovative aspects of pycortex as a tool for surface visualization. Finally, the third section describes a typical pycortex workflow and presents examples of the major features in pycortex. Readers are encouraged to download the package (https://github.com/gallantlab/pycortex/) to follow along. Additional in-depth examples and explanations are included in the pycortex documentation (http://gallantlab.org/pycortex/docs/).

## 2.2 Background

fMRI generates rich volumetric data which can be difficult visualize. Imaging data are often presented as 3D projections onto 2D planes. However, contiguous functional domains in volume visualizations may appear as unconnected patches. Surface visualization provides an intuitive way to simultaneously view all cortical activity recorded in an fMRI data (Van Essen et al. 2001). The organization of the mammalian cortex ensures that discrete functional domains can be visualized as contiguous patches on the cortical surface (Felleman and Van Essen 1991; Kaas 2012). However, the folding of the cortex obscures information deep in sulci, so functional information is difficult to visualize on the raw surface. To permit better visualization, surface visualizations can unfold the sulci and gyri while maintaining anatomical contiguity.

Many fMRI data analysis packages include a surface visualization module, and these all make use of a standard three-step pipeline: (1) a triangular mesh representation of the cortical surface is extracted from an anatomical scan; (2) functional and anatomical data are coregistered; (3) functional data (or the results of some analysis of the functional data) are projected onto the cortical surface mesh representation. In the following sub-sections, we detail how each of these steps is accomplished.

### 2.2.1 Cortical surface mesh generation

The cortical surface is usually modeled as a triangular mesh in 3D. The mesh is created by first segmenting the brain at the tissue boundaries in a volumetric anatomical scan, then applying a mesh generation algorithm such as marching cubes (Dale et al. 1999). Once the triangular mesh has been created, 3D geometrical operations are performed to inflate and flatten the cortical surface (Fischl et al. 2001). Flattened views of the cortical surface show data across the entire cortex without the need for multiple views in 3D. In order to create a flattened cortical surface representation from the three-dimensional cortical sheet without introducing excessive spatial distortion, relaxation cuts must be introduced into the cortical surface model. This operation is typically performed manually. To avoid splitting regions of interest on the flattened surface it is best to use functional localizer information when determining the location of relaxation cuts.

### 2.2.2 Coregistration

Functional MRI data are typically collected using an imaging sequence that is optimized for functional rather than anatomical tissue contrast (Nishimura 2010). Thus, the functional data and the anatomical data that produced the surface must be spatially aligned before projecting the functional data onto the cortical surface model. This process is called coregistration (Jenkinson and Smith 2001) and results in a transformation matrix that maps between the 3D coordinates of voxels in the functional data and the 3D coordinates of voxels in the anatomical data. Coregistration is typically performed automatically by global optimization of an affine transform from the functional image to the anatomical image used to generate the surface (Jenkinson et al. 2002). Since the contrast between anatomical and functional images are different, these frequently generate poor alignments.

In contrast, recent coregistration algorithms optimize surface intersections with the functional data. By maximizing the gradients across the surface, these algorithms can achieve accurate coregistration with no manual intervention. This technique—called boundary based registration (BBR) (Greve and Fischl 2009)-performs extremely well for data collected using a whole-head slice prescription. However, BBR can still fail unexpectedly. Imaging artifacts related to echo-planar imaging such as distortions and dropout negatively affect the performance of BBR, and it rarely works well with partial-head slice prescriptions. (This is why the accuracy of automatic coregistration should always be verified visually by overlaying the transformed functional image on the anatomical data.)

In most fMRI analysis pipelines, the functional-anatomical transformation estimated by the coregistration procedure is used to re-slice functional data into the same space and resolution as the anatomical scan. Re-slicing allows interpretation of functional results with respect to volumetric anatomical landmarks and provides a straightforward means of transforming data into standardized anatomical spaces (e.g. MNI or Talairach space) (Friston et al. 1995). Re-sliced data can also be projected onto the inflated or flattened cortical surface. However, as we will describe in more detail below, re-slicing data into volumetric anatomical space is not strictly necessary for projection of the data onto the cortical surface. Only the functional-anatomical transformation is necessary.

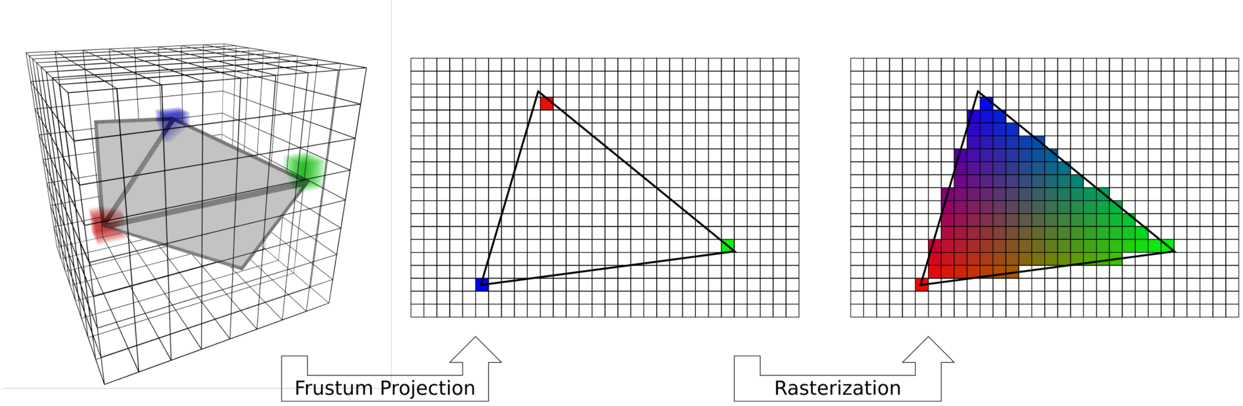


*Figure 2.2*: *Standard vertex-based mapping*
Standard OpenGL rendering implements an algorithm that can be used for vertex-based mapping. First, a fragment of the surface (gray) embedded in the voxel data (3D grid) is used to sample the data. Sampling occurs at only the vertices of the triangle (red, green, blue cubes). The surface triangle is projected to the screen on a regular 2D grid through a standard frustum projection (Woo et al. 1999). Only the data sampled by the vertices are carried over to the screen. Finally, the automatic graphics pipeline rasterizes the triangle by interpolating the colors across a barycentric coordinate space. Vertex-based mapping is generic, but does not adequately sample all underlying data in the case of plotting fMRI data.

### 2.2.3 Projection of functional data.

Visualization of the functional data on the cortical surface is usually accomplished using a 3D graphics pipeline that implements simple vertex-based projection. Vertex-based projection (Woo et al. 1999) can be split into three steps (Figure 2.2). First, each vertex in the cortical surface mesh is mapped into the functional volume. Second, the volumetric functional data are sampled at the vertex locations. If functional data have been re-sliced to anatomical space then this is trivial. However, by using the functional-anatomical transformation information this mapping can be applied directly from the functional data (in its native space) onto the cortex without re-slicing. Finally the color of each pixel on the display is determined by a 3D renderer, usually by linear interpolation between the values of the nearest vertices. This three-step method is not optimal because it requires two separate sampling steps: once from volume space to vertex space, and then again from vertex space to display (pixel) space. If the data are re-sliced to anatomical space then this adds a third sampling step. Each sampling step leads to aliasing and loss of resolution. Furthermore, mesh smoothing and other surface manipulations may cause uneven vertex spacing, and therefore uneven spatial resolution across the cortical surface. This effect is especially apparent with high resolution data (for example, $< 2\text{mm}^3$ voxel data collected using 7T fMRI).

One possible way to address this problem is to subdivide the mesh surface to increase the number of vertices, and thus the number of sampling locations. This is the solution is adopted by BrainVoyager QX. However, increasing the number of vertices greatly increases the computational load, and can only be applied to small portions of the cortex at one time.

## 2.3 Innovations in pycortex

Pycortex improves the process of fMRI visualization in a number of ways. First, pycortex integrates a number of tools to generate high quality cortical surface reconstructions. Pycortex uses these surfaces to sample functional data using a novel projection algorithm that results in much higher resolution visualizations. Finally, we draw on the power of modern graphics cards to provide a highly interactive, accurate, and portable visualization platform that works from within any modern web browser.

### 2.3.1 Surface generation and coregistration

Software packages such as Caret (Van Essen et al. 2001), Freesurfer (Dale et al. 1999), SUMA (Cox 1996), and BrainVoyager QX (Goebel 1997) are typically used to generate a high quality mesh representation of the cortex. Since these surface segmentation and mesh manipulation algorithms are already well developed, they are not reimplemented in pycortex. Instead, pycortex uses surface information output from these packages to create three-dimensional visualizations that can be easily manipulated and viewed. Pycortex is most closely integrated with Freesurfer, a free, open-source software package that is already used by a large community (e.g., the Human Connectome Project (Glasser et al. 2013)). However, pycortex can import most of the 3D formats that are used by standard MRI segmentation packages.

Since the advent of BBR, automatic coregistration algorithms usually produce high quality alignment for whole-brain studies without manual intervention. However, when partial-head slice prescriptions are used then it is best to perform manual coregistration, and it is always wise to visually check any coregistration solution. Pycortex provides an alignment tool that plots the surface mesh overlaid on the functional data. This allows users to view the alignment in orthogonal slice planes (to simulate traditional piecewise linear transformations), or using a global 3D view. The surface may be translated, rotated, and scaled interactively relative to the functional volume. The user can use these tools to visually match the surface with the underlying functional volume.

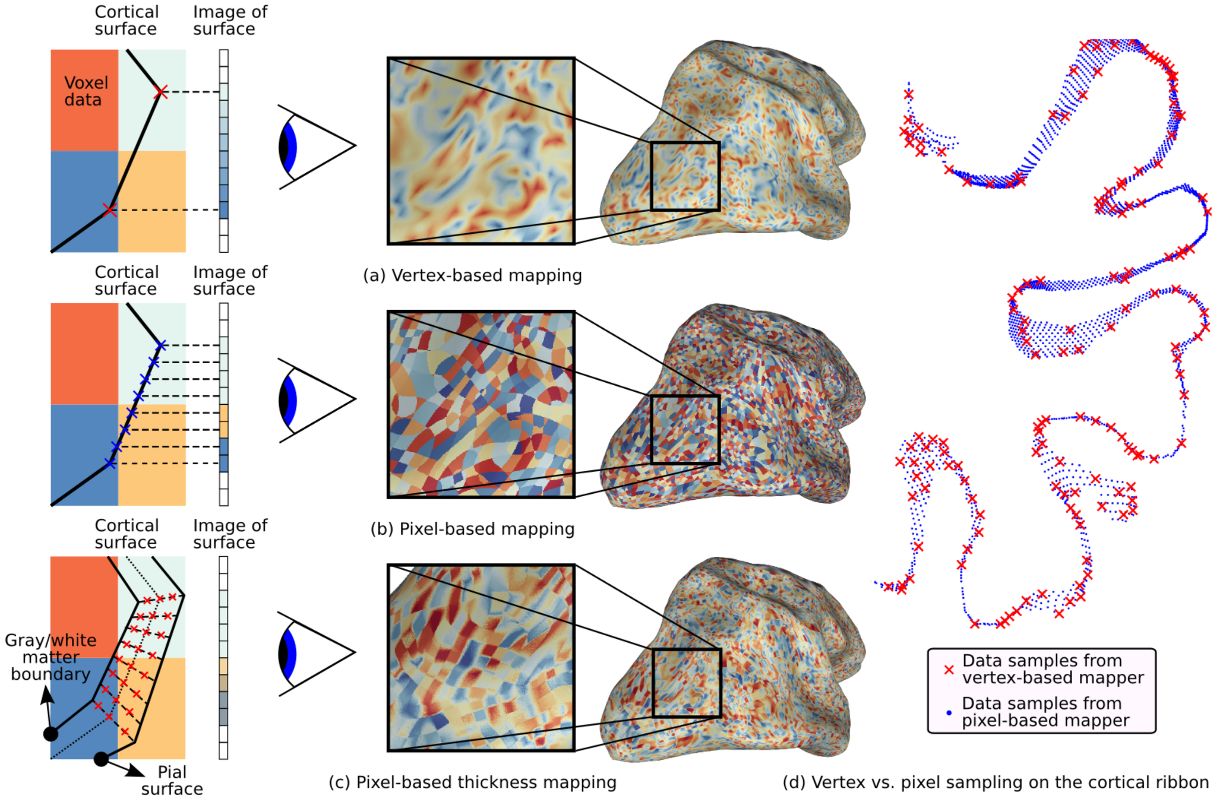
Some of the available tools for coregistration and segmentation are difficult to use. For example, the Freesurfer interface for marking relaxation cuts for surface flattening is extremely poor. Therefore, pycortex integrates several different tools to simplify the process of segmentation and coregistration. Pycortex replaces the default Freesurfer tool with Blender, an open source mesh editing program that is relatively easy to use. Pycortex also integrates with the BBR implementation provided by FSL to provide automatic coregistration that is compatible with surfaces generated by Freesurfer. The simple command pipeline provided by pycortex makes the entire process of surface generation and visualization smooth and relatively straightforward.

### 2.3.2 Pixel-based mapping

As discussed above, vertex-based mapping can be a lossy process that involves unnecessary interpolation. Pycortex implements a simpler, more accurate sampling scheme called pixel-based mapping (Figure 2.4). This scheme replaces the three separate projection steps with a two-step process that only samples the data once. This pixel-based algorithm directly maps the pixel coordinates on the display into the functional volume, thereby eliminating the intermediate vertex space representation. Pixel-based mapping therefore produces much higher fidelity images of the underlying data than those produced by the typical vertex-based method. Compared to other surface visualization methods, pixel-based mapping is computationally costly, since the functional volume must be resampled for every viewpoint in 3D. If the view is rotated even a degree, every pixel must be mapped anew into volume space. However, pycortex renders visualizations smoothly and in real time when used with modern graphics cards and shader pipelines.

Once pixel locations are mapped into volume space, they must sample the underlying data to derive their color. Different sampling methods trade off between speed and accuracy and generate visually distinct images. Pycortex includes several different sampling methods which allow very fine-grained control over this trade-off. The simplest method is nearest-neighbor sampling in which the mapped pixel is assigned the value from the nearest voxel. Nearest-neighbor sampling is fast and easy to compute and simple to interpret. However, nearest-neighbor sampling renders hard edges between adjacent voxels, so it can create a false impression of sharpness in the data. Trilinear sampling interpolates between the eight closest voxels to compute each sample. Trilinear interpolation uses a triangular filter that reduces aliasing compared to nearest-neighbor sampling. However, this suppresses high spatial frequency information and may produce results that are too smooth. Sinc filtering results in samples with the lowest reconstruction error (Oppenheim and Willsky 1996). Sinc filtering can be approximated with a lanczos filter that optimally preserves the spatial frequencies present in the functional data. However, this truncated filter is slow to apply so this sampling scheme cannot be used for real-time rendering. Other sampling schemes can also be implemented in pycortex through an extensible interface.

With other visualization packages such as Caret, only a single position between the pial and white matter surface (typically halfway between) is sampled to generate the visualization. However, human cortex varies in thickness from 1.5-3mm (Fischl and Dale 2000); thus sampling only a single position may ignore voxels which are closer to the white matter or pial surface. Pycortex uses a special sampling scheme called thickness sampling to take multiple samples between pial and white matter surfaces, which captures activity distributed through the thickness of the cortex. In thickness sampling, each pixel is mapped to a line in volume space that stretches between the pial and white matter surfaces. Several samples are taken along this line and the samples are averaged to derive the final pixel value (see Figure 2.3(c)). Alternatively, a single plane within the cortical mantle can be selected so that



(a) Vertex-based mapping

(b) Pixel-based mapping

(c) Pixel-based thickness mapping

(d) Vertex vs. pixel sampling on the cortical ribbon

*Figure 2.3*: *Typical fMRI data Projections.*
On the left, simulated 2D volumes are projected onto a 1D screen. The large colored blocks represent voxel data and the small rectangles represent screen pixels. Three different mapping methods are illustrated along with their effect on the surface. (a) Standard vertex-based mapping uses vertices in the surface mesh to sample the underlying functional data. The vertices (red x's) sample the functional data using a nearest-neighbor algorithm and the values are automatically interpolated by the rasterizer. Although the surface passes through the orange voxel at bottom right, there are orange pixels on the screen because there is no enclosed vertex. (b) Pixel-based mapping projects screen pixels into the volume to sample the underlying data. Orange pixels now appear on the screen since the surface passes through that voxel. (c) Thickness mapping samples data along the entire line between the white matter and the pial surface, thereby reflecting activity throughout the thickness of the cortical mantle. (d) The difference in sampling density between pixel-based and vertex-based mapping is shown for a 1mm slice through the cortical volume as it is mapped onto a flat map. Vertex samples are shown as red x's and pixel samples are shown as blue dots. The increased density improves the accuracy of functional data display, particularly with high-resolution functional data.

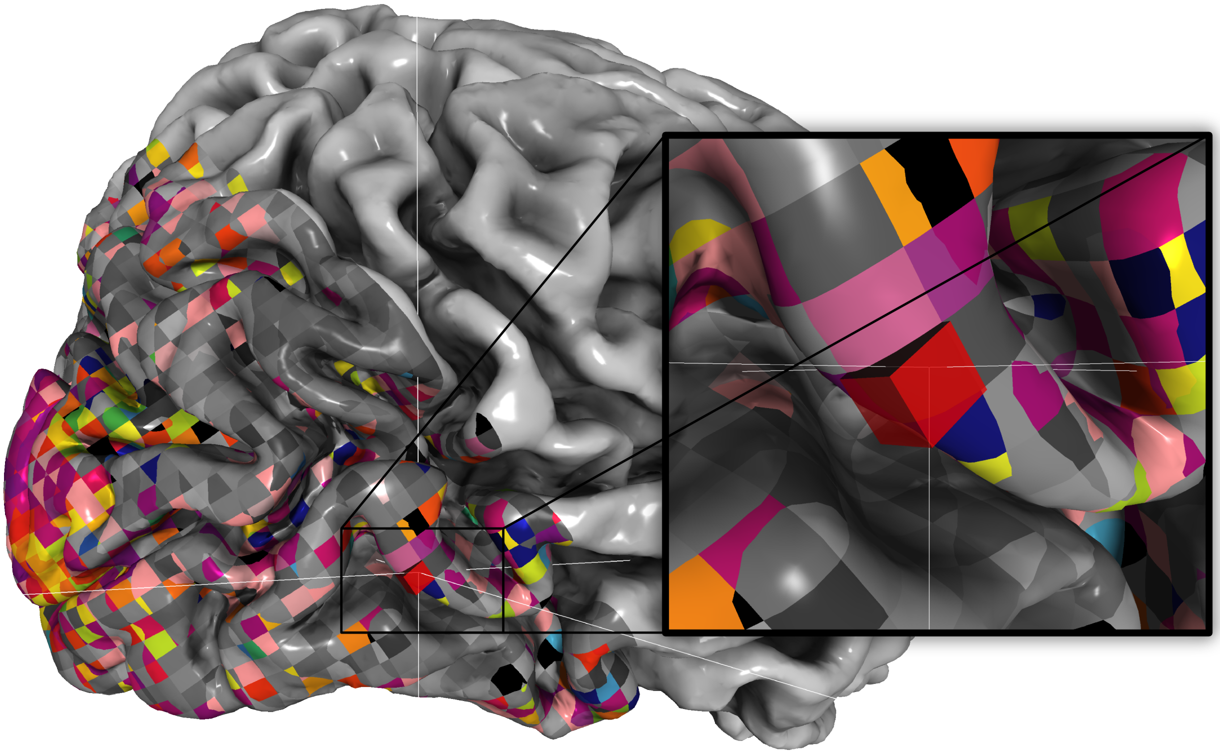


*Figure 2.4*: *Retinotopic data for example subject.*
Pixel-based mapping in pycortex renders voxels true-to-form. Here, a retinotopic map as in Hansen et al. (2007) is plotted using webgl. Note that the slanted slice prescription and the isotropic voxel size is easily visible due to pixel-based mapping and nearest-neighbor sampling. The inset shows how a single voxel intersects the surface.

the data can be viewed anywhere between the pial and the white matter surfaces. Thickness sampling is a costly process to run on a CPU, but it is fast and efficient when implemented using custom shaders and a modern graphics card. To further improve responsiveness with thickness sampling, samples along a random set of positions through the cortical sheet can be averaged. This dithering trades off accuracy in favor of interactivity, but still accurately represents information through the thickness of the cortex.

### 2.3.3 WebGL and data sharing

Graphics card acceleration allows highly complex datasets to be rendered in real time on standard computers. Typical 3D data visualization software relies on programming interfaces like OpenGL to access this powerful hardware. However, software which relies on OpenGL typically requires extensive installation procedures to visualize even simple datasets. WebGL is a new technology which melds the OpenGL programming interface with Javascript, a language used to program websites. This allows powerful data visualizations to be programmed directly inside a web browser. Bringing graphics card acceleration to web pages provides the opportunity to create portable, interactive visualizations of fMRI data.

Pycortex takes full advantage of the power of WebGL by implementing custom shaders on the graphics card. Modern graphics cards include programmable shaders that allow custom code to be uploaded to the card, thus enabling highly parallel rendering operations. By using custom shaders, pycortex can use accelerated rendering algorithms that would otherwise be too slow to be practical. When data is visualized in the WebGL view, only volumetric data and surface structure is passed into the web browser; all other functionality is accomplished by shader programs. Custom shaders included in pycortex enable the surface to be drawn quickly, even when pixel-based mapping and a user-selectable sampling method are used.

Because web browsers are ubiquitous on modern personal computers, no special installation is required to view pycortex visualizations. The use of a web browser as the front end for pycortex also allows an unprecedented level of interactivity. For example, the anatomical surface can be flattened interactively simply by dragging a slider. This interactive design helps the user to develop a clear sense of the correspondence between flattened and anatomically correct surfaces. Pycortex can also display temporally varying time-series data on the cortical surface in real time. This allows simultaneous visualization of the experimental paradigm and the functional data in real time (for an example of such a visualization, see `http://www.gallantlab.org/brainviewer/retinotopy_demo`).

It is simple to post pycortex visualizations to a web page for public viewing. These static visualizations are generated using a simple command that generates a single web page with most resources embedded directly. The surface structure, data, and the webpage can then be posted to any public facing web site. For example, the online Neurovault data repository (http://neurovault.org) now makes use of pycortex, and any fMRI data uploaded to Neurovault can be visualized automatically in pycortex. These visualizations are visible at a static web address that can be referenced in papers and shared with anyone with a web browser.

## 2.4 Pycortex functionality

Pycortex is free, open-source software written in python and javascript. Pycortex adds to the growing body of python tools for neuroscience (Halchenko and Hanke 2012; Millman and Brett 2007; Pedregosa et al. 2012). Installation instructions for pycortex and associated software can be found at `http://pycortex.org`.

In pycortex user interaction is handled through the python command line. Here, we present the typical workflow for pycortex, proceeding from anatomical and functional images to a web-based 3D visualization. In the simplest possible case, only three commands are required to generate a fully interactive surface visualization in pycortex:

```
>>> cortex.segment.init_subject("S1", "T1_anatomical.nii.gz")
>>> cortex.align.automatic("S1", "transform_name",
                          "functional.nii.gz")
>>> cortex.webshow((data, "S1", "transform_name"))
```

These commands illustrate three important python modules for cortical segmentation and visualization. The segment module initializes the cortical segmentation using an anatomical image. The align module provides both automatic and manual coregistration tools for coregistering the surface and functional images. The webgl module is used to generate interactive web visualizations. Two other modules are also documented here to highlight additional pycortex functionality; the overlay module can be used to define surface overlays and regions of interest (ROIs), and the quickflat module is used to generate figure-quality images.

Pycortex makes use of a large amount of internal data such as subject surfaces, alignments and other metadata. All data required for pycortex is kept in a database that is implemented as a simple directory on the user's hard drive. In most cases pycortex uses this database seamlessly without requiring any interaction from the user. (For more information about the database and all supported file formats, consult the pycortex documentation.)

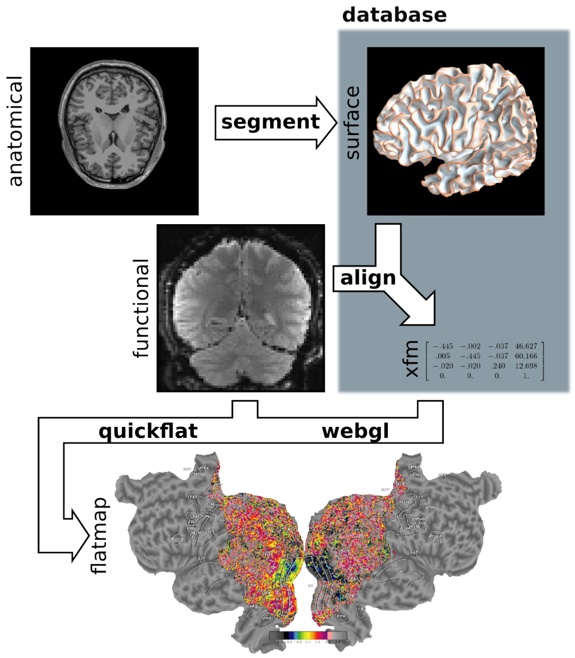


*Figure 2.5*: *Pycortex module diagram.* Pycortex provides an integrated visualization toolkit for fMRI. Several pycortex modules are used to transform the user-provided anatomical and function data into an interactive visualization. The *segment* module integrates with Freesurfer to generate the surface. The *align* module uses the functional data and the surfaces to generate a transform. The *quickflat* and *webgl* modules generate static 2D flatmaps and interactive 3D visualizations, respectively. The *overlays* module is used to generate vertex-based ROIs and extract surface-defined volume ROIs.

### 2.4.1 The segment module

Pycortex integrates with Freesurfer to generate surfaces from anatomical images. For optimal results from Freesurfer, anatomical MRI images should be collected using the scanning protocols defined in the Freesurfer documentation (MGH 2009). Freesurfer is optimized to work with a specific multi-echo T1 scan sequence, and we have found that this sequence maximizes surface quality and minimizes the need for manual editing of the surface after automatic segmentation. Once the requisite anatomical images have been collected, only a two lines of code is necessary to initiate surface segmentation in pycortex:

```
>>> from cortex import segment
>>> segment.init_subject("S1", "T1_anatomical.nii.gz")
```

This command uses Freesurfer to generate the surface files automatically and stores the surface in the pycortex database with the identification code S̈1: (All further pycortex processing steps will refer to this surface by the identification code assigned at this stage of processing). Segmentation is a slow process that can take up to 12 hours, but it can be run unattended.

Freesurfer generally performs very well on normal brains, but minor topological errors may occur in areas of low contrast, such as at the cerebellar boundary and around the optic nerve. Segmentations may also be compromised around diseased tissues that may be present in diseases such as stroke or aneurysm. It is therefore wise to check all surfaces before further processing using one of the following commands:

```
>>> segment.fix_wm("S1")
>>> segment.fix_pia("S1")
```

These commands open an interface that permits segmentation edits to be applied directly to the white matter or pial surfaces. One window is from Freesurfer's segmentation editor tool; white matter voxels can be added or removed in this interface to alter the final surface. A 3D view also opens in another window to view the surface that resulted from the current segmentation. Minor segmentation errors typically manifest as spikes or lumpy areas on the surface. Having both interfaces open simultaneously allows location information to be shared, facilitating manual editing to improve surface extraction. (For more information about how to make these edits, consult the Freesurfer documentation, or follow the segmentation tutorial in the pycortex documentation.) Saving and exiting from all windows will automatically run Freesurfer once more to apply changes and generate new surfaces.

Once the surfaces are deemed satisfactory, relaxation cuts can be introduced to facilitate creation of cortical flatmaps. This is accomplished with one command:

```
>>> segment.cut_surface("S1", "lh")
```

This command automatically exports the surface and opens it in Blender. Vertex selection and face deletion tools can be used to remove the medial wall. Vertices can be marked in conjunction with functional data to facilitate relaxation cutting. For example, retinotopic mapping data can be projected onto the brain to facilitate cutting along the calcarine sulcus to separate visual hemifields. Marked cuts are processed automatically for use in the flattening procedure. (In-depth instructions on performing this step can be found in the documentation for pycortex.) When the changes are saved, pycortex automatically flattens the surface and makes the new flat surface available for visualization. Functional data can immediately be plotted on this flatmap.

If segmentation is performed outside of the segment module, it is still possible to use these surfaces in pycortex. For example, if surfaces from older experiments were generated by CARET, copying the surface files directly into the pycortex database allows them to be used in any pycortex visualization. (For more information about how to use external surfaces, please consult the pycortex documentation.)

### 2.4.2 The align module

To project functional data onto anatomical surfaces accurately the functional data must first be coregistered with the anatomical surface. Pycortex supports automatic coregistration using the BBR tool within FSL (see Background). Pycortex also provides a fully manual alignment tool. Three arguments are required to launch the automatic coregistration tool: the subject, the name of the transform, and a functional reference image. For example,

```
>>> from cortex import align
>>> align.automatic("S1", "test_alignment",
                    "reference_epi.nii.gz")
```

This will automatically coregister the function image with the surface, and store the transform into the pycortex database. After an automatic coregistration, the transform can (and should) be checked with the manual alignment tool to ensure accurate coregistration:

```
>>> align.manual("S1", "test_alignment")
```

This manual alignment tool has three panels that show the current surface slice intersection with the reference image, and a fourth panel that shows the full 3D rendering with slice positions. Showing the data this way facilitates accurate alignment of the gray matter with the functional data. A sidebar contains options to adjust the contrast and brightness, along with some additional settings. The surface can be moved using key commands listed in the sidebar. Hotkeys and buttons in the graphical interface allow the anatomical volume to be translated, rotated and scaled in order to align it optimally with the functional data. The alignment can be saved using a button or by exiting the interface.

Transforms in pycortex are stored in the pycortex database in the form of an affine transform matrix that operates in magnet isocenter right anterior superior (RAS) space (as defined by NIFTI headers). The matrix transforms surface coordinates, which are typically stored with respect to the anatomical space, into the functional space. This format is compatible with AFNI's transform format. Utility functions are included to allow conversion between AFNI/pycortex format and the FSL format.

### 2.4.3 The quickflat and webgl modules

Pycortex provides two visualization tools to plot functional data on surfaces. quickflat visualizations use matplotlib to generate figure-quality 2D flatmaps and webgl uses a web browser for interactive visualizations. Both tools use pixel-based mapping to project functional data onto the cortical surfaces accurately.

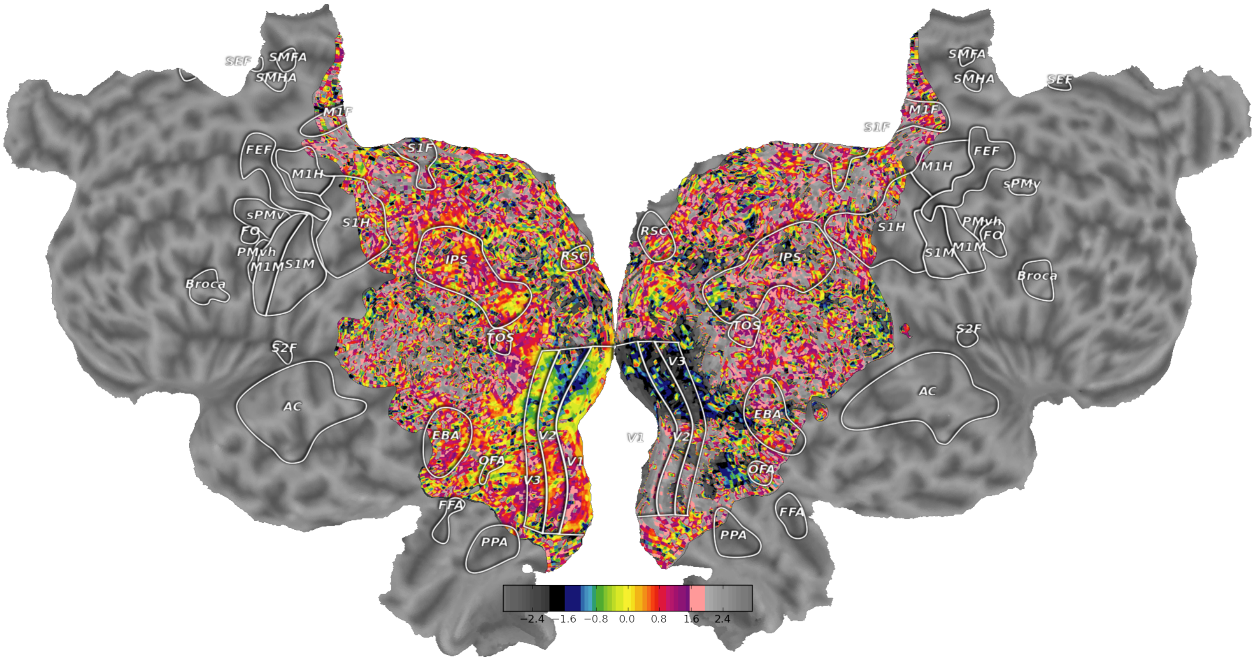


*Figure 2.6*: *High quality flatmaps generated by pycortex.*
Cortical flatmaps can be quickly generated in pycortex using the quickflat module. Matplotlib is used to view figure-quality flatmaps in a standard size. Options are available to include a colorbar, ROI outlines, ROI labels, and curvature information. This figure shows retinotopic organization of visual cortex for one subject as in Hansen et al. (2007)

Quickflat was used to generate the figures in Figure 2.6; to load the same visualization,

```
>>> import cortex
>>> dataset = cortex.load("S1_retinotopy.hdf")
>>> cortex.quickshow(dataset.angle)
```

This sequence of commands loads the example dataset (http://gallantlab.org/pycortex/S1_retinotopy.hdf) and then plots the flatmap in a matplotlib window. (For more informations about additional display options include options to select the sampling function, please consult the pycortex documentation.)

To generate these flatmaps quickly, the quickflat module precomputes a mapping from volumetric samples to figure pixels. These mappings are represented as sparse matrices so visualizations can be generated from new data quickly by taking the dot product of the matrix with the unraveled volume. To generate these sparse matrix mappings, a grid of pixel locations are generated that span the extent of the flatmap surfaces. A Delaunay triangulation is then generated for the flat surfaces and the simplex membership is found for each pixel. Next, the barycentric coordinate on the simplex is generated from the triangulation transform for every pixel. The original surface coordinate is then computed by substituting the mid-cortical vertex (or averaged across multiple depths for thickness sampling) for the flatmap vertex in the Delaunay triangulation, and weighting the vertices with the barycentric coordinate. Finally, a sampler argument determines which function is used to sample the 3D coordinate.

The webgl visualization can be launched using syntax similar to that used for the quickflat visualization:

```
>>> import cortex
>>> dataset = cortex.load("retinotopy.hdf")
>>> cortex.webshow(dataset)
```

This starts a web server in python and opens a browser window to display the visualization. After a brief loading period, the cortical surface is shown with the retinotopy demo data projected on the surface. The rendered 3D view is a fully dynamic visualization that allows real time rotation, panning, and scaling.

The data display can be modified interactively in numerous ways. The dynamic view has two sliding windows that contain display options. The large slider at the bottom linearly interpolates the shape of the cortical mesh between the original (folded) anatomical, inflated, and flattened surfaces. This allows the unfolding process to be visualized continuously, and it clarifies the correspondence between 3D anatomical features and the cortical flatmap. The sliding window located at the top contains options that change how the data is displayed. Different colormaps can be selected and the colormap ranges can be altered dynamically. 2D colormaps are also supported, allowing two datasets to be contrasted simultaneously. Multiple datasets can be loaded and compared directly by simply toggling between them. Sliders are provided to change the transparency of the dropout, overlay, data, and curvature layers.

As explained earlier, pycortex uses custom shaders that implement pixel-based mapping. During 3D graphics rendering, the color of each pixel is determined by some predefined code at the fragment shading step. Under a traditional fixed-function pipeline, fragment shading is performed by a rasterizer that implements vertex-based mapping (Woo et al. 1999). In contrast, the fragment shader in pycortex projects each pixel into the functional space in 3D, and then samples the underlying volume data by reading from a texture. Nearest-neighbor or trilinear sampling is automatically performed by OpenGL when the data is read from the texture. This generates a fully interactive and accurate real-time visualization.

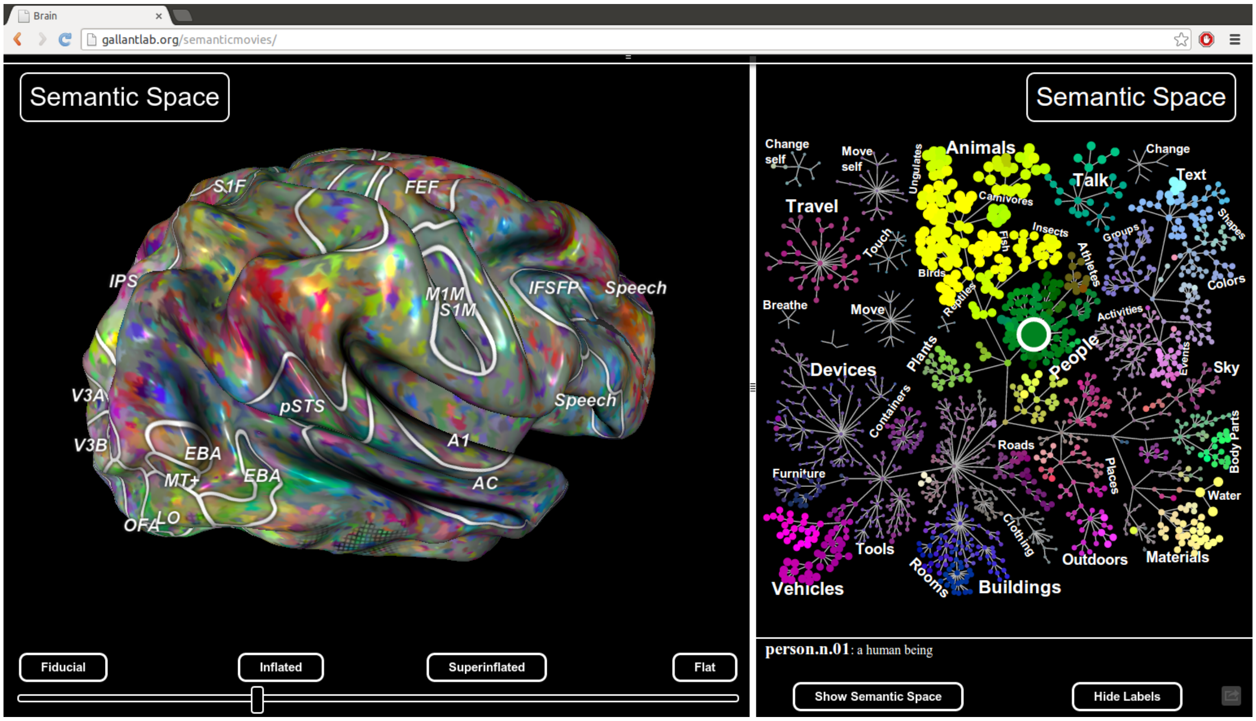
The webgl module contains code that parses and generates the HTML and javascript code required to display surface data in a web browser. It provides two possible use cases: a dynamic view that can be controlled by a back end python web server, and a static view that generates static HTML files for upload into an existing web server. The OpenCTM library (Geelnard 2009) is used to compress the surface mesh into a form that can be utilized by the web browser. If a dynamic view is requested, the webgl module sets up a local web server with all the required surface and data files accessible to the web browser. If a static view is requested, all HTML and javascript code is embedded into a single HTML document and saved to a set of files. Data (in the form of compressed mosaic images) and surface structures are stored separately. These standalone visualizations can then be copied to a web server to be shared with colleagues, included as links in published articles, or shared online with a broad audience.

Pycortex also includes a javascript plugin architecture that allows new interactive visualizations to be developed easily. For example, the static viewer released with Huth et al. (2012) `http://gallantlab.org/brainviewer/huthetal2012/` contains a plugin that allows the user to visualize how 1765 distinct semantic features are mapped across the cortical surface. Clicking a point on the brain picks the closest voxel and the viewer displays the semantic category tuning for the associated voxel.

Finally, pycortex provides a bi-directional communication framework between python and javascript, so that actions in javascript can be scripted and manipulated in python. This powerful interaction dynamic allows exploratory data analysis in a way never before possible.

### 2.4.4 The overlays module

One common requirement of fMRI studies is to visualize regions of interest (ROIs). ROIs are typically defined in volume space, using a statistical threshold applied to a functional localizer contrast (Poldrack 2007). Because these thresholded regions are not anatomically constrained their intersection with the cortical surface is not guaranteed to be contiguous or smooth. Another common requirement is to visualize retinotopic ROI defined by identifying hemifield inversions on a cortical flat map (Hansen et al. 2007). The overlays module provides a means to define overlays, such as ROI borders and other surface markers, directly on the cortical surface. These ROIs are automatically rendered by pycortex as paths or regions on the rendered surfaces.



gallantlab.org/semanticmovies/

person.n.01: a human being

*Figure 2.7: Static view for web presentation.*
Pycortex uses webgl to generate a static view that can be hosted on a web site. The static view allows users to share data with colleagues, collaborators, and the public. No additional software needs to be installed. This figure, published with Huth et al. (2012), shows a typical static view. A static plugin written in javascript on the right interacts with the 3D view, allowing dynamic interaction between the plugin and the view.

To add an ROI, the user must provide contrast data and a named transform:

```
>>> import cortex
>>> cortex.add_roi((contrast_data, "S1", "fullhead"),
                     name='ROI␣name')
```

This automatically starts Inkscape, an open source vector editing program. A flatmap as generated by quickflat is shown with multiple layers corresponding to different overlays. If a closed path is drawn into the ROI layer, pycortex regards it as a complete ROI. A simple utility function can then extract the volumetric mask of this ROI:

```
>>> mask = cortex.get_roi_mask("S1", "fullhead", "V1")
>>> mask['V1'].shape
(31, 100, 100)
```

This returns a volume that indicates the number of ROI vertices within each voxel. The volume can be converted into a binary mask by finding all nonzero voxels. This simple thresholding is equivalent to a nearest-neighbor sampling. Pycortex also provides other projection options that may include additional voxels. (For more information, please consult the pycortex documentation.)

Pycortex stores overlays as 2D vector paths in the standard SVG image format that is easily parsed by many libraries. This allows flexible handling of surface overlays either in pycortex (via Inkscape) or in other programs outside of pycortex. Because the original contrast flatmap is embedded within the SVG the files retain a permanent record of the contrast used to draw the border.

## 2.5 Future development

The pycortex WebGL view provides an unprecedented method for exploration of cortical MRI data. The interactive interface allows results to be manipulated in innovative ways that facilitate comprehension, and the ability to generate static views greatly simplifies data sharing and publication. However, the current WebGL viewer contains a limited set of plugins for interactive data visualization. We plan to develop a large set of interactive plotting tools that will facilitate dynamic data analysis in a web browser. We are also working on extensions to pycortex that allow EcoG and EEG data to be visualized on the cortical surface. Unfortunately because WebGL is a very new standard, support is still unreliable. Therefore, we are also exploring options to stabilize the software on additional platforms (including mobile platforms) and to improve accessibility.

## 2.6 Acknowledgements

Pycortex draws on and makes use of code from a large set of open source libraries, without which it would not exist. Please see the README file included with the software for a full list of acknowledgements. Thanks to members of the Gallant lab who helped beta test and develop this software, including N. Bilenko, T. Cukur, and A. Nunez.

# Chapter 3

# Reducing head motion during fMRI by means of a personalized 3D-printed insert

## 3.1 Introduction

Subject head motion is a continual source of problems for fMRI and diffusion MRI research (Friston et al. 1996; Le Bihan et al. 2006). Head motion can reduce the magnitude of functional signals, create spurious signals (Oakes et al. 2005), and drastically affect estimates of functional connectivity (Power et al. 2012; Van Dijk et al. 2012). These problems are compounded by modern fMRI acquisition techniques such as GRAPPA (Griswold et al. 2002) and simultaneous multi-slice (aka multiband) (Larkman et al. 2001), which offer increased scan efficiency at the cost of increased motion sensitivity. Current methods for reducing head motion include foam padding, inflatable pneumatic pads and bite bars (Bettinardi et al. 1991). However, these methods reduce head motion only modestly and they can be very uncomfortable. One promising method is prospective motion correction, where subject motion is tracked in real time using an infrared camera and the scan window is moved to compensate (Zaitsev et al. 2006). However, any motion during a scan reduces the efficacy of shimming. Furthermore, motion that occurs within a slice acquisition causes shearing of that slice along with misalignments in subsequent slices. Thus, prospective motion correction cannot fix all motion-related artifacts and it is best to eliminate motion at the source.

One previous study immobilized subjects by means of a personalized plaster cast which was attached to the head coil (Edward et al. 2000). This proved very effective at reducing subject motion, but it required substantial effort, time, and skill to produce the cast. Furthermore, each subject must be enrolled well before the experiment in order to allow sufficient time to produce a working plaster cast of the head.

Here we present an alternative method for reducing head motion: a rigid plastic insert that fills the space between the MRI head coil and an individual subject's head. The insert is designed so that it conforms to the inside surface of the head coil and the outside surface of the subject's head. A fused-deposition 3D printer is used to manufacture the insert out of polylactic acid (PLA). The resulting insert constrains head motion far more effectively than padding and it substantially reduces the effort required from the subject to keep still. Furthermore, the rigid plastic insert does not compress over the duration of an experiment, reducing subject drift. One drawback of this method is that it requires a separate insert to be manufactured for each individual subject. However, this cost is likely worthwhile when subjects will be used in multiple experiments, when each subject will participate in many scanning sessions or when subject motion is a particular problem or concern. Furthermore, recent mass-market adoption of 3D printing has made individual manufacture of these inserts a relatively economical option given the cost of subject disqualification due to motion.

We scanned two subjects to test the effectiveness of the insert using an experimental paradigm designed to measure effects on functional signal-to-noise ratio (fSNR). We found that the insert substantially reduced motion parameters estimated by retrospective motion correction and it increased fSNR in much of the brain. Our results suggest that the 3D-printed insert could drastically reduce the problems of head motion in fMRI studies.

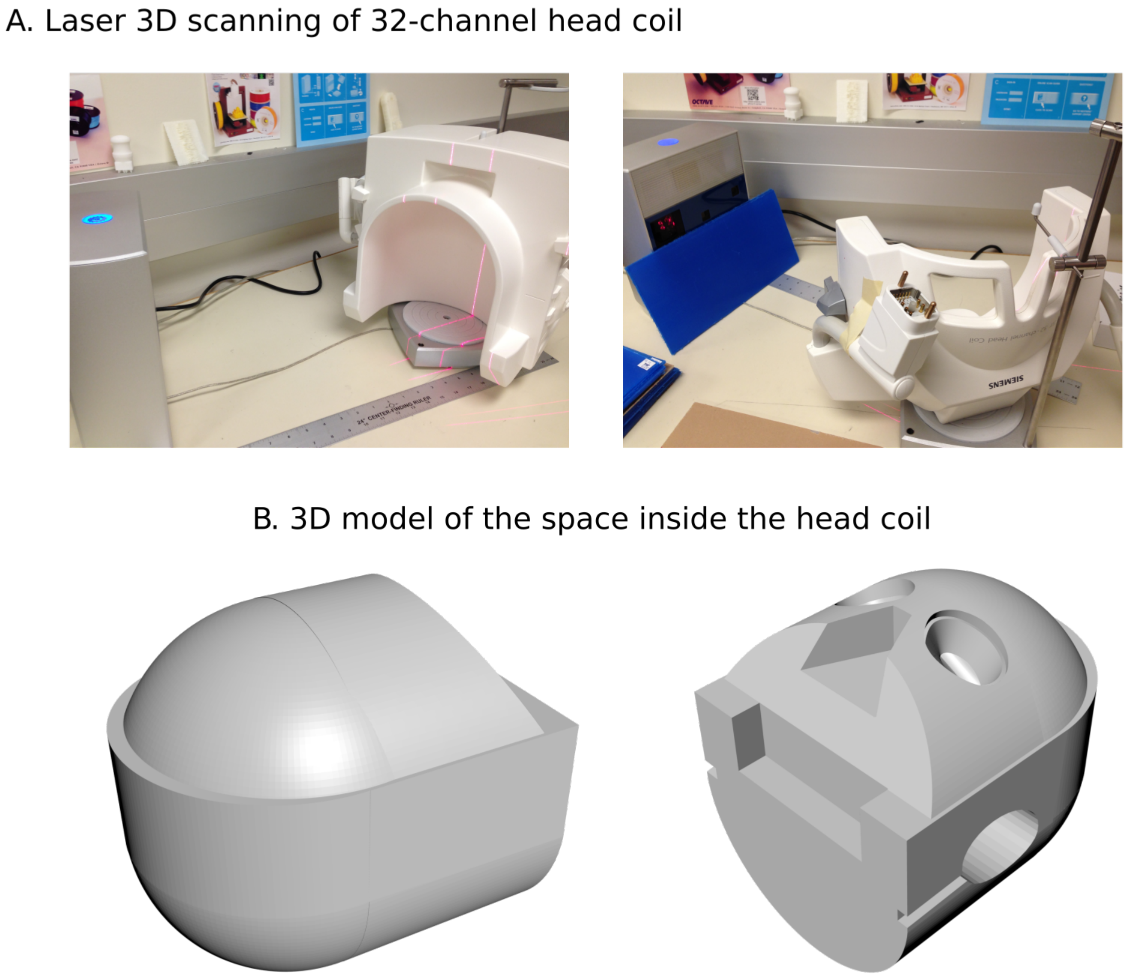
## 3.2 Materials & Methods

### 3.2.1 Insert design & construction

An insert that matches precisely the shapes of both the head coil and subject's head requires a high quality 3D model of both the coil and subject. We used a NextEngine 3D scanner HD (NextEngine, Santa Monica, CA, USA) to obtain a 3D model of the head coil. (A photograph of the laser scanning procedure is shown in Figure 3.1A.) We then used the Blender animation software to model the inside surface of the coil (Figure 3.1B).

To obtain a 3D model of the subject's head, a T1-weighted anatomical MRI was acquired (Figure 3.2A and below). The anatomical image was smoothed and median filtered to reduce noise. A canny edge filter was performed on 2D slices of the image. Holes in the edges were closed with binary dilation followed by binary erosion. A flood fill algorithm was performed in 2D slices to generate a scalp volume. The marching cubes algorithm was used to generate a surface reconstruction (Figure 3.2B). Laplacian smoothing and quadric decimation was applied to smooth and simplify the surface reconstruction. Finally, this scalp surface model was expanded by 2mm along each vertex normal to account for hair and pulsatile expansion of the head.

Once the head model was completed, a model of the interstitial space was obtained by boolean constructive solid geometry operations performed in Blender. This simply involved taking the difference of the inside surface of the coil and the subject's head. Extra holes were created to make room for the ears, nose, eyes, and mouth, and for additional hardware such headphone wires and lenses for refractory correction (Figure 3.3A). The resulting insert model was split in half vertically to allow subject entry, and each half was further divided into top and bottom pieces to facilitate printing (Figure 3.3B).



A. Laser 3D scanning of 32-channel head coil

B. 3D model of the space inside the head coil

*Figure 3.1*: *3D scanning and reconstruction of the space inside the MRI head coil*
(A) A 3D scanner is used to generated a high quality 3D mesh reconstruction of the head coil (B) Open source mesh editing software is used to generate a closed model of the inside of the head coil. Spaces are cut into the model for the eyes, ears, and nose using boolean difference operations.

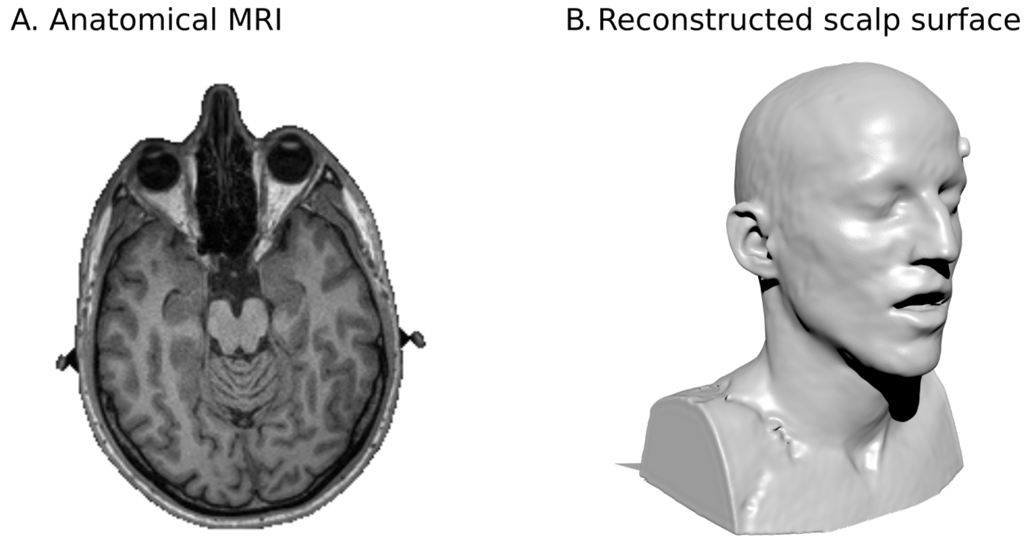


*Figure 3.2*: *Anatomical scan and reconstruction of the outside surface of the subject's head.* (A) A quality T1 image with 1mm isotropic resolution and a wide field of view is collected for the subject. (B) The outside surface of the subject's head is reconstructed from the anatomical MRI using flood-fill and binary morphology algorithms. This surface is expanded by 2mm to account for hair and pulsatile expansion.

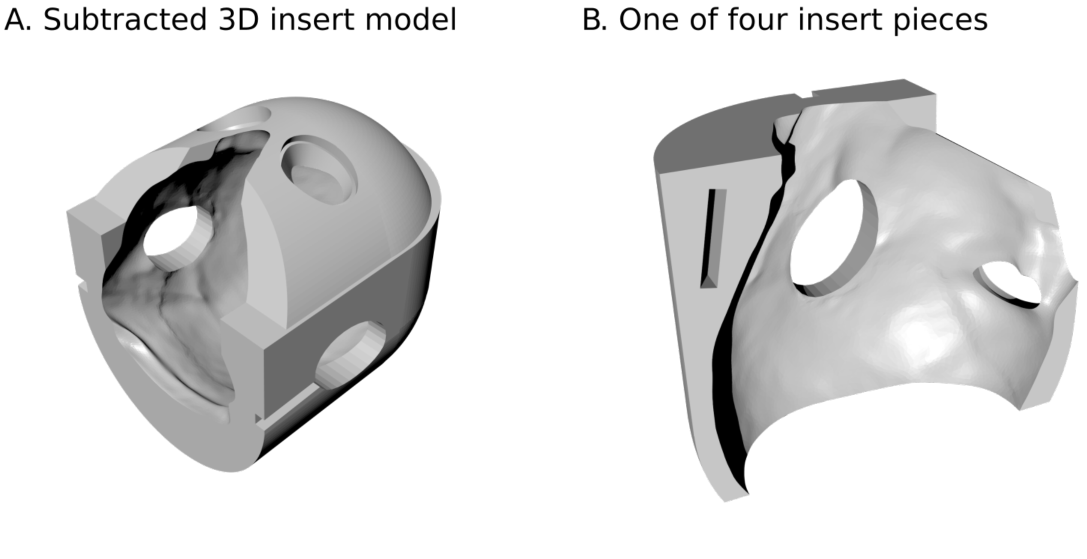


*Figure 3.3*: *Final insert design*
The subject head model is subtracted from the head coil model to obtain a model of the space between the head and the coil. The spaces for eyes, ears, and nose are matched with the head model to ensure adequate space for the subject. (B) This model is split into four pieces to aid 3D printing.

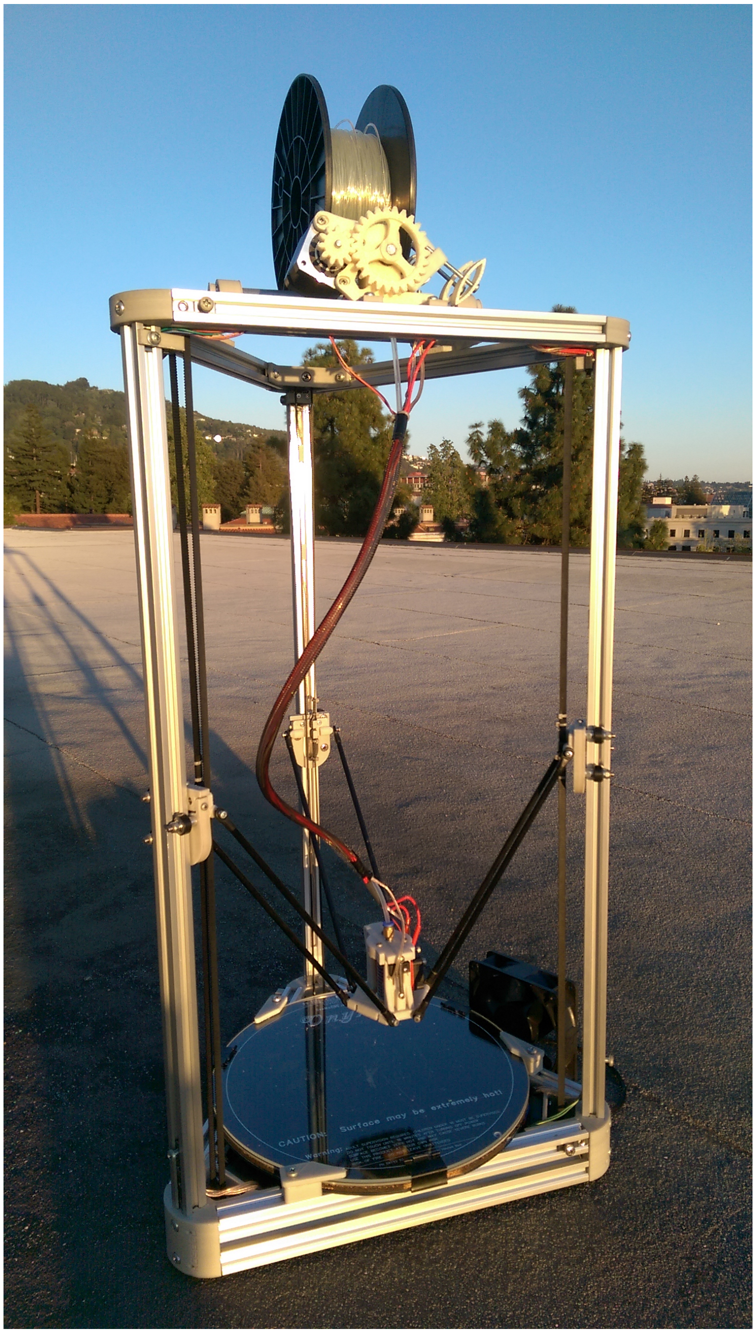


*Figure 3.4*: *Fused deposition 3D printer*
The insert is printed on a fused deposition 3D printer. Each of the four pieces is printed separately. Printing takes approximately 24 hours total.

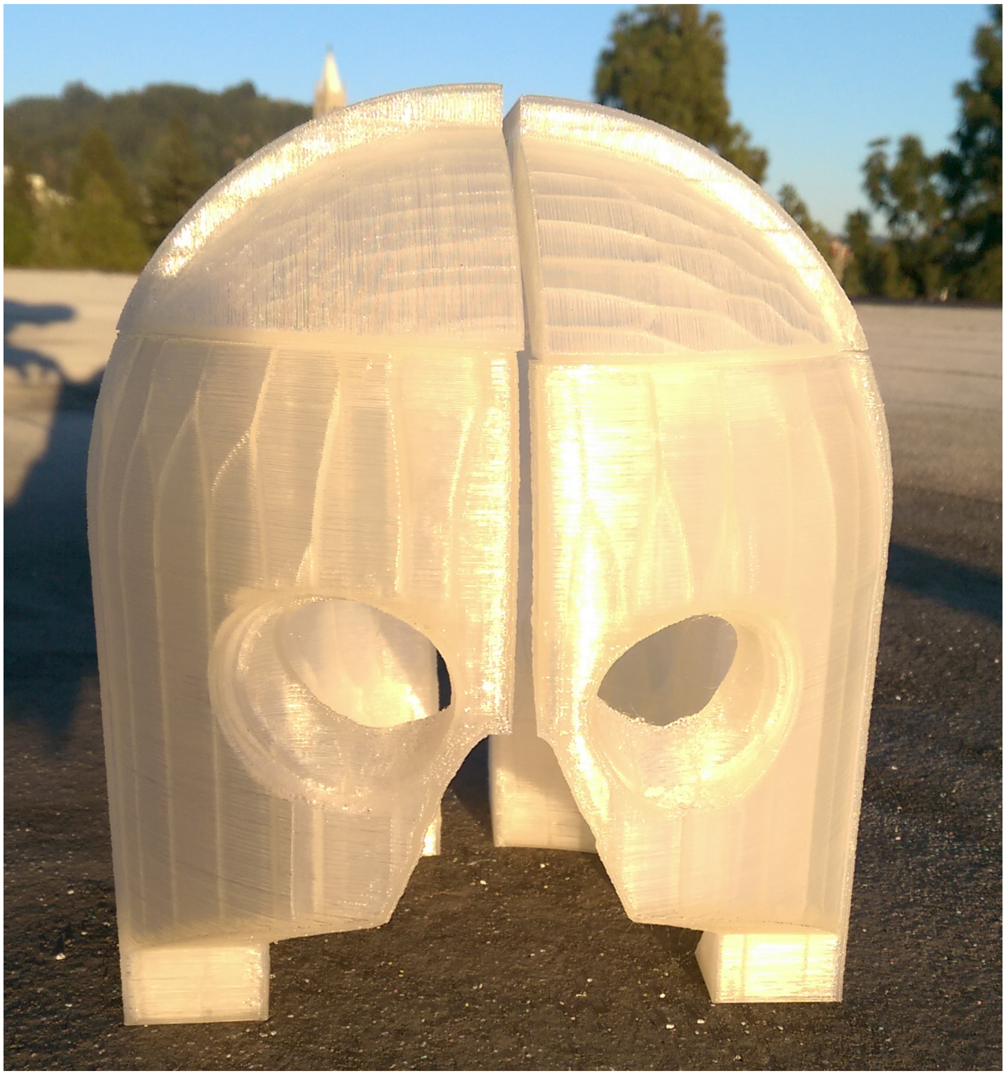


*Figure 3.5*: *Completed insert*
The top and bottom pieces of each half of the insert are glued together. To place the subject in the scanner, the two pieces are held on either side of the head and then the head is lowered into the coil. The insert snugly fills the space between the subject and coil.

Finally, the four pieces were printed out of polylactic acid (PLA) using a custom-built delta robot fused deposition 3D printer (Figure 3.4). PLA was used as the material because it has little to no MRI signal (Wapler et al. 2014), and its low shrinkage results in high fidelity printed models. The top and bottom pieces of each half were then bolted together with nylon bolts and glued together with superglue (Figure 3.5).

Although the device is made of hard plastic, our subjects reported that the insert was reasonably comfortable.

### 3.2.2 MRI methods

In order to generate an accurate scalp segmentation, T1-weighted anatomical images were collected on a 3T Siemens TIM Trio scanner (Siemens, Erlangen, Germany) using a 12-channel Siemens volume coil. This coil was used for anatomical imaging because it produces homogeneous images of the entire head and so improves automatic scalp segmentation.

fMRI data were collected on the same scanner using a 32-channel Siemens volume coil. This coil was used for functional imaging because it provides much higher functional SNR (fSNR) than the 12-channel coil. Functional scans were collected using gradient echo EPI with TR = 2.0045s, TE = 34.5ms, flip angle = 74 degrees, voxel size = 2.24 x 2.24 x 4.1 mm (slice thickness = 3.5 mm plus 18% slice gap), matrix size = 100 x 100 and field of view = 224 x 224 mm. Twenty-four axial interleaved slices were prescribed to cover the majority cortex for the subjects. A standard fat-saturation sequence was used to suppress signal from scalp fat.

fMRI experimental methods. To test the effect of the insert on head motion and fSNR, subjects were shown a 2-minute movie sequence (with sound). This movie consisted of 10-15 second clips assembled from various creative commons licensed videos downloaded from Vimeo.com. Throughout the scan the subject maintained fixation on a small point superimposed at the center of the screen. Scanning took place in two separate sessions separated by two days. In the first session, subjects was first scanned five times while watching the movie without the insert. The subjects then came out of the magnet in order to put the insert in place and five more movie scans were then acquired with the insert. In the second session this procedure was reversed so that the first five scans were with the insert and the last five without.

Visual stimuli were presented using an MR-compatible Avotec projector (Avotec, Stuart, FL, USA) mounted behind the scanner and projecting onto a screen mounted inside the bore of the magnet. The stimuli subtended approximately 24 x 24 degrees of visual angle. Auditory stimuli were presented using Sensimetrics S14 in-ear piezoelectric headphones (Sensimetrics, Malden, MA, USA). A Behringer Ultra-Curve Pro hardware parametric equalizer was used to flatten the frequency response of the headphones based on calibration data provided by Sensimetrics.

### 3.2.3 Subjects

Functional data were collected from two male subjects (both age 28). Subjects were healthy and had normal hearing and vision. Both subjects are experimenters with extensive experience in the scanner and both remain quite still even without the insert.

### 3.2.4 fMRI data analysis

Rigid-body motion parameters (translation and rotation) were estimated for each volume in each functional run using the FMRIB Linear Image Registration Tool (FLIRT) from FSL 5.0 (Jenkinson and Smith 2001). All volumes in each run were then averaged to obtain a high quality image. FLIRT was also used to automatically align this average image for each run to a template image collected in a separate scanning session. The cross-run transformation matrix was then concatenated to the motion-correction transformation matrices obtained using MCFLIRT, and the concatenated transformation was used to resample the original data directly into the overall template space. Low-frequency voxel response drift was modeled as a quadratic polynomial and then subtracted from the signal. The remaining signal for each voxel was then z-scored across time within each scan.

To measure fSNR, the mean explainable variance was computed separately for each voxel in each of the two conditions (i.e. with and without the insert). This was done by computing the fraction of variance in each voxel timecourse for each individual scan that could be explained by the mean timecourse of that same voxel across all ten scans. This metric ranges between 0 when the voxel timecourses for each scan are independent, and 1.0 when a voxel has exactly the same timecourse on every scan. Statistical significance of fSNR was estimated by randomly selecting repeats from the two conditions, computing their explainable variance and taking their difference. This procedure was repeated for 100 bootstrap samples, and the p value for the two actual difference was computed on a per voxel basis.

## 3.3 Results

To determine whether the insert reduced head motion we compared estimated motion parameters with and without the insert present . Both subjects tested here are highly experienced experimenters who remain quite still in the scanner even without the insert. However, examination of the estimated translation and rotation parameters shows that the insert significantly reduced the amount of steady motion for one subject (Figure 3.6). The second subject had exceptionally good motion parameters, but showed a large non-intentional muscle twitch in one trial. Total root-mean-squared (RMS) motion was significantly reduced for all motion parameters except for yaw-rotation in subject 1. RMS motion was significantly reduced in roll rotation and x-axis translation in subject 2. In conclusion, the insert dramatically reduces motion to less than 0.25mm translation in any direction and less than 0.15 degrees rotation for any subject whose motion exceeds these parameters.

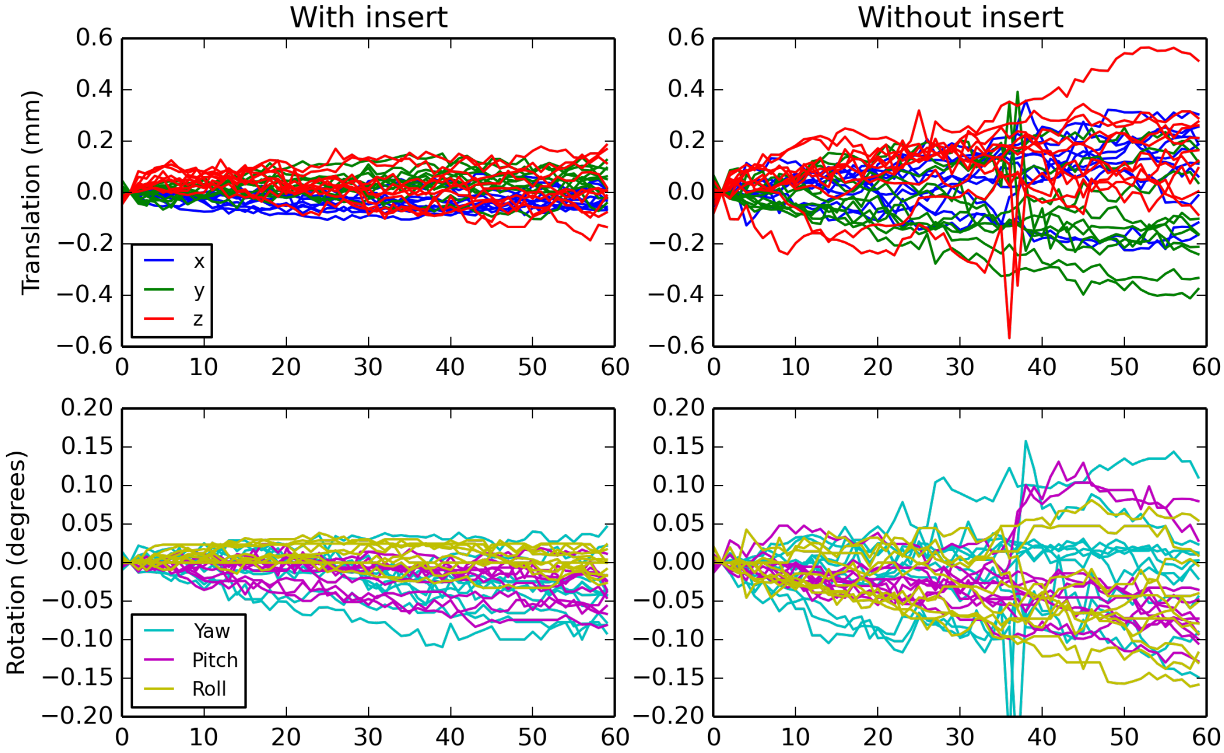


*Figure 3.6*: *Estimated motion parameters with and without the insert*
Subjects were scanned to evaluate motion. FLIRT was used to apply motion correction to each run separately to estimate motion. Motion traces from all 10 runs are plotted for a single subject. In the top panels, estimated translation with and without the insert are plotted as time series. Estimated rotations with and without the insert are plotted in the bottom panels. Estimated motion parameters show that the motion is substantially reduced when the insert is in place.

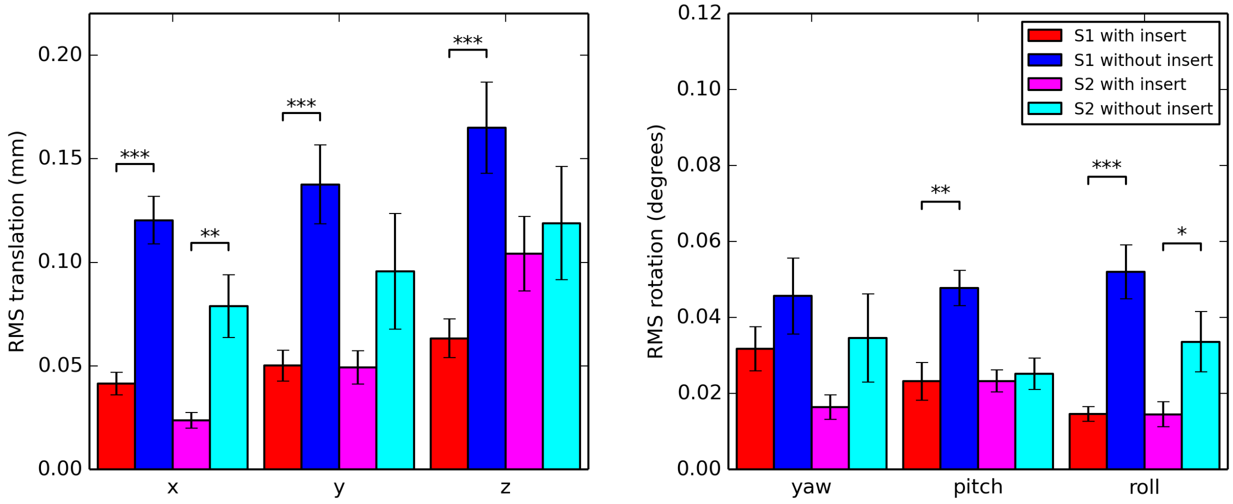


*Figure 3.7*: *Summary of motion parameters*
Motion parameters are summarized here for both subjects with an RMS measure of translation and rotation. Estimated motion parameters are significantly reduced for many motion parameters. * indicates $p < .05$, ** for $p < .01$, and *** for $p < .001$.

Next we compared estimated voxel-wise functional SNR (fSNR) values between the two conditions. For the subject who moved relatively more during the scan, the insert significantly improved fSNR in approximately 15% of the voxels across the cortex ($p < 0.05$). Even for the subject who moved relatively little in total, fSNR was significantly improved in 10% of voxels across the cortex. Thus, the printed insert improves fSNR even in compliant subjects who move very little.

## 3.4 Discussion

Here we showed that subject head motion during fMRI scanning can be reduced substantially by using a personalized 3D-printed plastic insert to stabilize the head. This rigid insert, which is custom designed and manufactured for each subject, completely fills the space surrounding the subject's head and so makes head motion nearly impossible. Preliminary tests showed that the insert significantly reduces motion and increases functional SNR.

The insert substantially improves head stability within a session, but it requires some effort and resources for initial construction. At this point the design and manufacturing process have been developed sufficiently that they are largely automated and require little human intervention. However, an anatomical MRI must be acquired for each subject before the insert can be designed, and 3D printing one full insert still takes at least 24 hours on our home-built 3D printer. These limitations suggest that this method will be most useful when the same subjects will be tested in multiple sessions or when it is particularly important to

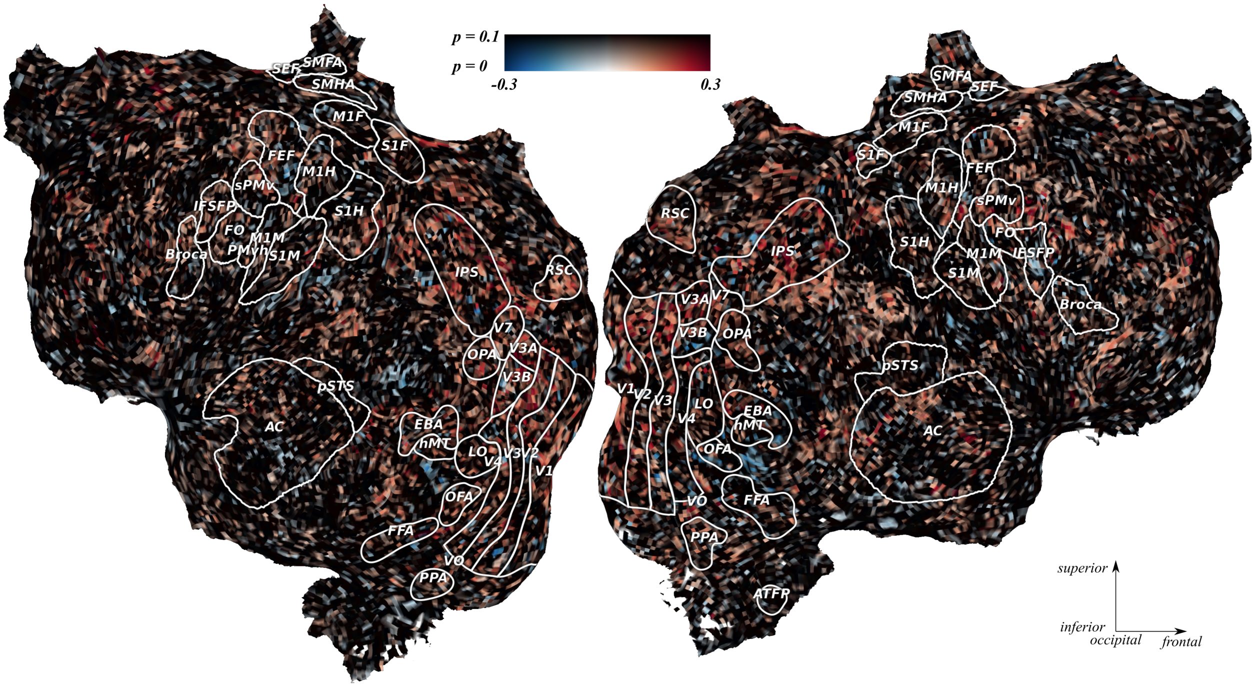


*Figure 3.8*: *Estimated fSNR with and without the insert.*
fSNR was estimated for every voxel in the scanned volume with and without the insert. The difference in fSNR between the two conditions was computed for each voxel and then plotted here on a flattened cortical map for one subject. A significance test was carried out by shuffling the conditions and displaying only voxels where $p < 0.1$. Here red voxels have higher fSNR with the insert and blue voxels have higher fSNR without the insert.

minimize head movement.

While 3D printers are currently very slow, they utilize commodity hardware that can be obtained cheaply. Thus, it might be feasible to perform parallel production of inserts using multiple machines. Furthermore, newer 3D printing technologies such as DLP resin curing might be able to produce an insert in as little as 20 min (Tumbleston et al. 2015). Thus, it might soon be possible to produce an insert almost automatically and effortlessly at the beginning of a scanning session.

The ability to rapidly design and manufacture personalized inserts also opens up many other possibilities for future improvement of MRI data acquisition. For example, it is currently difficult to perform concurrent fMRI and transcranial magnetic stimulation (TMS) because the TMS coils must be positioned precisely over the subject's scalp and then locked firmly into place. This problem might be solved by designing inserts with mechanisms to lock the TMS coils into particular positions.

Radiofrequency (RF) transmit inhomogeneity is a severe problem in ultra-high field fMRI (7+ Tesla). This problem can be alleviated by surrounding the subject with dielectric pads made of materials such as calcium titanate (Haines et al. 2010), but these pads are difficult to position accurately and consistently. This might be solved by designing voids in the insert that can be filled with dielectric material, giving perfect consistency and a high degree of control over the placement of the dielectrics. Since these inserts are generated using a high resolution anatomical scan, finite element modeling could also be used to precisely align additional hardware and model how they interact with the tissues in the head.

Perhaps the most exciting potential application of the approach proposed here is to improve fSNR in fMRI by integrating receive coils directly into the insert. One of the main fSNR limitations in conventional fMRI is the distance from the receive coils to the brain. Because conventional whole-head receive coils are manufactured to accommodate 90% of human heads, the coils are often quite far from the brain. It has recently been shown that flexible surface MRI coils provide higher fSNR than volume coils when imaging shallow structures near the cortical surface (Arias et al. 2014). Thus, one very exciting possibility is that flexible coils could be integrated directly into the insert. This would place the coils as near as possible to each individual's brain, and so could provide substantially higher fSNR than can be obtained with conventional head coils.

All of these possibilities suggest that the customized 3D-printed insert proposed here could help alleviate many problems with current fMRI research.

# Chapter 4

# Fast fMRI improves signal quality by separating signal and noise spectra

## 4.1 Introduction

Recent advances in parallel imaging methods have made it possible to acquire whole-brain fMRI images in as little as 100ms (Feinberg et al. 2010; Setsompop et al. 2012), 20-30 times faster than typical fMRI experiments (Friston et al. 1999). However, the BOLD signal measured by fMRI changes very slowly and takes many seconds to peak after an impulse of neural activity (Glover 1999; Boynton et al. 1996). Therefore it is unlikely that fast measurements contain more functional information than slow measurements. Still, several recent studies have shown that fast fMRI can improve BOLD sensitivity in resting state (Feinberg et al. 2010; Smith et al. 2013), block design, and continuous natural stimulus fMRI experiments (Chen et al. 2014). These results are enticing but do not offer a clear picture of why fast fMRI is helpful.

In this study we explored why fast fMRI improves BOLD sensitivity under a stimulus-driven paradigm. We scanned subjects with simultaneous multi-slice (SMS) fMRI sequences (Setsompop et al. 2012; Larkman et al. 2001) with acceleration factors ranging from 1x (no acceleration) to 12x (12 slices are simultaneously acquired during each excitation) and repetition times (TRs) ranging from 2000ms to 167ms. For each pulse sequence, we scanned subjects 10 times while they watched the same rapidly-changing natural audiovisual stimulus. This design allowed us to separate the true functional signal in each voxel (which is the same across repetitions) from the noise (which is different). Under this model, only only stationary responses to repeated stimuli are considered “signal”. Nonstationary effects such as stimulus adaptation are lumped into “noise”. We characterized the frequency content of the signal and the noise under each sequence with a temporal coherence analysis. Then we tested the temporal precision of the measured responses with a timepoint correlation analysis. Based on these results we developed a recommended protocol for fast fMRI that can be used in any fMRI study.

## 4.2 Methods

### 4.2.1 Experimental and Stimulus Design

Stimuli consisted of 18 color natural movies with sound drawn from Vimeo (www.vimeo.com). A single 5-10s audio-visual clip from each movie was extracted. Kdenlive (www.kdenlive.com) was used to combine clips to make a 2.5 minute audiovisual stimulus. An MR-compatible Avotec projector (Avotec, Stuart, FL, USA) was used to present this audiovisual stimulus to subjects. The movie subtended 24 by 24 degrees of visual angle and the video was played at 24 Hz. A red fixation cross was overlaid on the center of the movie and subjects were asked to maintain fixation throughout the scan. Eye tracking data were collected to ensure subject compliance. Sensimetrics S14 in-ear piezoelectric headphones (Sensimetrics, Malden, MA, USA) were used to present the auditory stimulus. A Behringer Ultra-Curve Pro hardware parametric equalizer was used to flatten the frequency response of the headphones based on calibration data provided by Sensimetrics. Sound level was adjusted for each subject at the beginning of each session.

Subject were scanned 10 times with each of the six fMRI sequences across two sessions. Each scan lasted exactly 2.5 minutes. The stimulus began playing simultaneously with the first acquisition of each scan. Within each session, scans were performed in a balanced, randomized order. Subjects were not aware of the scan order. The magnet was re-shimmed with advanced automatic shimming every 6 scans (approximately every 15 minutes). $B_0$ fieldmaps were acquired after every shim with a GRE sequence (Alecci et al. 2001).

### 4.2.2 fMRI Acquisition

All MRI data were collected on a 3T Siemens TIM Trio scanner (Siemens, Erlangen, Germany) at the UC Berkeley Brain Imaging Center. The 32-channel Siemens head coil was used for head imaging. Functional data were acquired at repetition times (TRs) of 2000ms, 1000ms, 500ms, 250ms, and 167ms. All functional EPI data were collected with the multi-band accelerated EPI sequence (version R011a) from the Center for Magnetic Resonance Research at the University of Minnesota. All functional scans used echo time (TE) = 34.2ms, voxel size = 2.24 x 2.24 x 4.1mm (3.5mm slice thickness with 18% slice gap; these parameters match earlier publications from our group (Huth et al. 2012; Çukur et al. 2013)). The z-axis field of view covered roughly 90% of the brain in each subject. All scans used fat saturation to remove signal from fat.

fMRI acquisition was split into two overlapping experiments. All scan parameters were matched as well as possible between scan speeds. Only the multiband factor and TR were altered between each scan. In Experiment 1, 30 stimulus repeats were collected with TR=2000ms, 1000ms, and 500ms. In Experiment 2, another 30 stimulus repeats were collected with TR=500ms, 250ms, 167ms. A full parameter summary for both experiments can be found in Table 4.1.

| | **Experiment 1** | | | **Experiment 2** | | |
|---|---|---|---|---|---|---|
| TR | 2000ms | 1000ms | 500ms | 500ms | 250ms | 167ms |
| Multiband factor | 1 | 2 | 4 | 4 | 8 | 12 |
| Total volumes | 64 | 128 | 256 | 256 | 512 | 768 |
| Flip angle | 74° | 62° | 47° | 47° | 34° | 28° |
| FFT scale | 1.0x | | | | 1.5x | 2.0x |
| Bandwidth | 2084 Hz | | | 2174 Hz | | |
| Partial Fourier | Full | | | 7/8 | | |
| Echo spacing | 0.57 ms | | | 0.69ms | | |
| TE | 34.2 ms | | | | | |
| Resolution | 2.24 x 2.24 x 4.1mm | | | | | |
| Matrix size | 100 x 100 x 24 | | | | | |

*Table 4.1*: fMRI acquisition parameters

### 4.2.3 fMRI Preprocessing

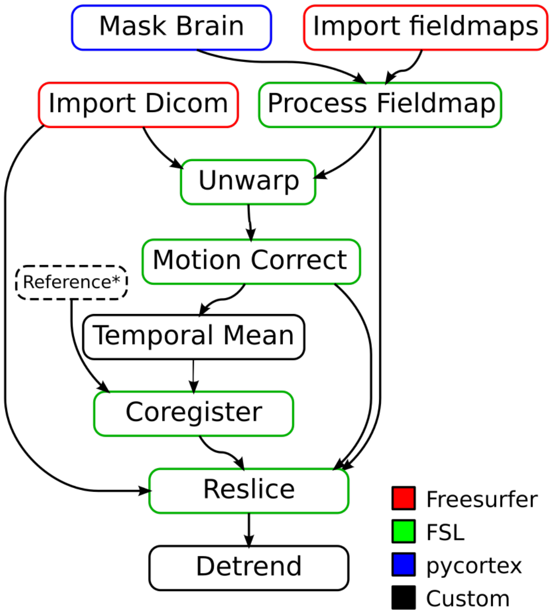
Fieldmap-based unwarping was used to correct for EPI distortion in two of the subjects (AH and AN) but failed in the third (JG). PRELUDE and FUGUE from FSL 5.0 were used to estimate warp fields for the two unwarped subjects (Jenkinson 2004; Alecci et al. 2001). The boundary-based registration (BBR) algorithm in FLIRT from FSL5.0 was first used to align each fieldmap magnitude image to the T1-weighted anatomical image for each subject. The fieldmap mask was generated by reslicing the high quality cortical masks from anatomical space into the fieldmap space. Each fieldmap was then despiked, median filtered, and smoothed using a 3D Gaussian kernel with $\sigma = 2$mm.

MCFLIRT from FSL5.0 was used to motion correct each functional run (Jenkinson and Smith 2001). A high quality template volume was then generated by averaging all volumes in the run. The template volume from the first TR=1000ms run was chosen as the overall template for each subject. FLIRT was then used to automatically coregister the template volume for each run with the overall template. The cross-run transformation matrix was then concatenated to the motion-correction transformation matrices obtained using MCFLIRT. For the two unwarped subjects, the concatenated linear motion-correction and cross-run transformations were combined with the estimated warp fields and applied simultaneously to the untouched input data. For the third subject, the linear transformations alone were applied to the untouched input data. Figure 4.1 shows the schematic preprocessing pipeline for unwarping, motion correction, coregistration, and detrending.

Low-frequency response drift in each run was identified using a 5th order Legendre polynomial then subtracted from the signal. The mean response for each voxel was then subtracted and the remaining response was scaled to have unit variance. To reduce residual drift on scan start, 6 seconds of data were trimmed prior to detrending. To reduce detrending edge artifacts, another 8 seconds were trimmed from the start and end of the scan after detrending. This resulted in exactly 128 seconds of data for analysis (64, 128, 256, 512, and 768 volumes for the 2000, 1000, 500, 250, and 167ms scans). To account for physiological noise in our signal, we recorded pulse oximetry and breathing using a BIOPAC system. We then used RETROICOR (Glover et al. 2000) to model a series of cardiac response functions and downsampled them to the TR for each scan. We fit these channels as nuisance regressors with ordinary least squares, and subtracted them from the signal before additional modeling steps.

Freesurfer was used to generate cortical surface meshes from T1-weighted anatomical scans (Dale et al. 1999). Pycortex was used to carefully hand-check and correct anatomical surface segmentations before surface reconstruction (http://pycortex.org, chapter 2). Pycortex was again used to introduce relaxation cuts on the surface of each hemisphere, then Freesurfer was used to flattened the mesh. Retinotopic mapping data was used as a guide to make the the calcarine sulcus cut along the horizontal meridian in V1 (Hansen et al. 2007).

The BBR automatic alignment tool from FSL was used to align cortical surface to the overall functional template. The line-nearest scheme in pycortex was used to project data onto the surface. This projection scheme samples the functional data at 64 evenly-spaced intervals between the inner (white matter) and outer (pial) surfaces of the cortex, then averages together the samples. Samples are taken using nearest-neighbor interpolation, wherein each sample is given the value of its enclosing voxel.



* Reference volume set to temporal mean of first TR=1000ms scan

*Figure 4.1*: fMRI Preprocessing Pipeline

## 4.2.4 Subjects

Functional data were collected from three male subjects (author JG, age 28; author AH, age 29; author AN, age 28). All subjects were healthy and had normal hearing and normal or corrected-to-normal vision.

## 4.2.5 fMRI Analyses

### Total Explainable Variance

First, we evaluated the total usable signal for each sequence. A functional signal-to-noise ratio (fSNR) was computed using a repeatability metric called explainable variance (EV). EV is computed by normalizing the ratio of variance across repeats and the variance along time for each voxel (Hsu et al. 2004). This metric is 0 for uncorrelated time courses across repeats, and 1 for identical timecourses across repeats. An EV metric was computed independently for each scan sequence and plotted in Figure 4.2, blue line.

### Data filtering

Temporal filtering was applied to many analyses to evaluate its effect on aggregate metrics. To generate low-pass data, an odd-length finite impulse response filter was designed with a hamming window such that the bandwidth of the resulting filter was the same across scan speeds. This resulted in filters with 15, 31, 61, and 91 taps for TR=1000, 500, 250, and 167ms, respectively. This filter was applied once forwards and once backwards to cancel any phase offset, and further analyses were performed on the result.

### Temporal Coherence

To evaluate the frequency content of BOLD, we again looked at the timecourse variance across repeats. We used a measure of temporal coherence to evaluate the signal across frequencies. This coherence metric measures the repeatability of the signal power at each frequency band from 0 Hz to Nyquist, and was computed as in Hsu et al. (2004). First, the ideal signal is estimated as the average timecourse across repeats. The noise was estimated from the residuals, which were computed by subtracting the average across repeats from each individual repeat. Power spectral density (PSD) estimation was independently performed on the signal and the noise to estimate the spectrum of each. Coherence is computed as $\frac{S^2(\omega)}{S^2(\omega)+N^2(\omega)}$. Coherence varies between $\frac{1}{\text{number of repeats}}$ due to chance coherence, and 1 for the exact same signal across repeats. The number of bins used in computing the PSD were normalized such that frequency bins were equal width across scan speeds.

Coherence was computed for voxels within the top 5th percentile in EV across all 6 scans. This corresponds to visual and auditory sensory areas driven by the stimulus (Figure 4.2). Approximately 3000 voxels from each subject were aggregated to plot coherence (Figure 4.3).

### Time Point correlation

Both EV and temporal coherence aggregate over time to derive a voxelwise measure of fSNR. Neither metric attempts to aggregate over voxels to determine a “temporal” fSNR, which we call temporal specificity. In order to directly measure temporal specificity, we created a metric called time point correlation. This metric more closely resembles the conditions of a real-time feedback experiment, where we may attempt to minimize lag between measurement and feedback. In time point correlation, the temporal specificity of a single image is directly measured by evaluating the accuracy of a classifier that best identifies a time point from all others.

To create this classifier, a model of brain activity is first built by averaging 9 repeats of the data, with the 10th held out for testing. In order to select which voxels participate in the classifier, we compute EV within the 9 repeats and select the 1000 best voxels. The same voxels are selected from the testing set and each timepoint is correlated with every other timepoint in the average model across voxels. The timepoint with the highest correlation coefficient is selected as the classification result, and the accuracy is evaluated. This process is repeated holding out each repeat, and the accuracy is averaged across folds. Alternatively, we attempt to sort the heldout timepoints in order of their appearance by correlating the heldout response with the average response. All data are upsampled and evaluated at 10 Hz to compensate for differing TR lengths for each sequence.

Since the data are upsampled (in some cases by a factor of 20), we expect to see lower accuracies if only the exact timepoint were matched. For example, timepoint 5.4 in the heldout data may best match timepoint 5.5 in the average model. This would be considered an error, despite the offset being only 100ms. To account for small offsets in identification, we computed accuracies at a range of match offsets from -2 to +2 seconds. Accuracies at each offset are plotted as a curve with the standard error computed across folds.

## 4.3 Results

### 4.3.1 Total Explainable Variance

First we computed the fraction of the response variance in each voxel that was shared across repetitions of the stimulus (the explainable variance, or EV). If a voxel had very different responses across repetitions (such that the responses average to exactly 0) then the EV would be 0.0. If a voxel had exactly the same response on every repetition then the EV would be 1.0. We computed the total EV for every cortical voxel across all subjects, and made observations based on the top 5% voxels for each scan speed.

Figure 4.2 shows the EV for two scan sequences (2000ms and 500ms from experiment 1) mapped across the cortical surface for one subject. This figure uses a 2D colormap to show EV from both sequences simultaneously: black voxels have low EV in both scans, blue voxels have higher EV in the 2000ms scan than the 500ms scan, red voxels have higher EV in the 500ms scan than the 2000ms scan, and white voxels have equally high EV in both. This cortical flatmap indicates that the complex audiovisual stimulus drives repeatable activity across visual and auditory sensory cortex in both sequences. However, the 2000ms scan has better EV overall. This effect is also visible in the scatter plot in Figure 4.2(a), where the majority of voxels have better EV in the 2000ms scan than the 500ms scan.

Comparing EV across all sequences reveals an inverse relationship between speed and EV. The 2000ms scan sequence has as good or better EV than every other sequence, and the fastest scan sequences suffer the most. This is consistent with the hypothesis that higher acceleration causes loss of fSNR due to a lower flip angle and multiband separation artifacts.

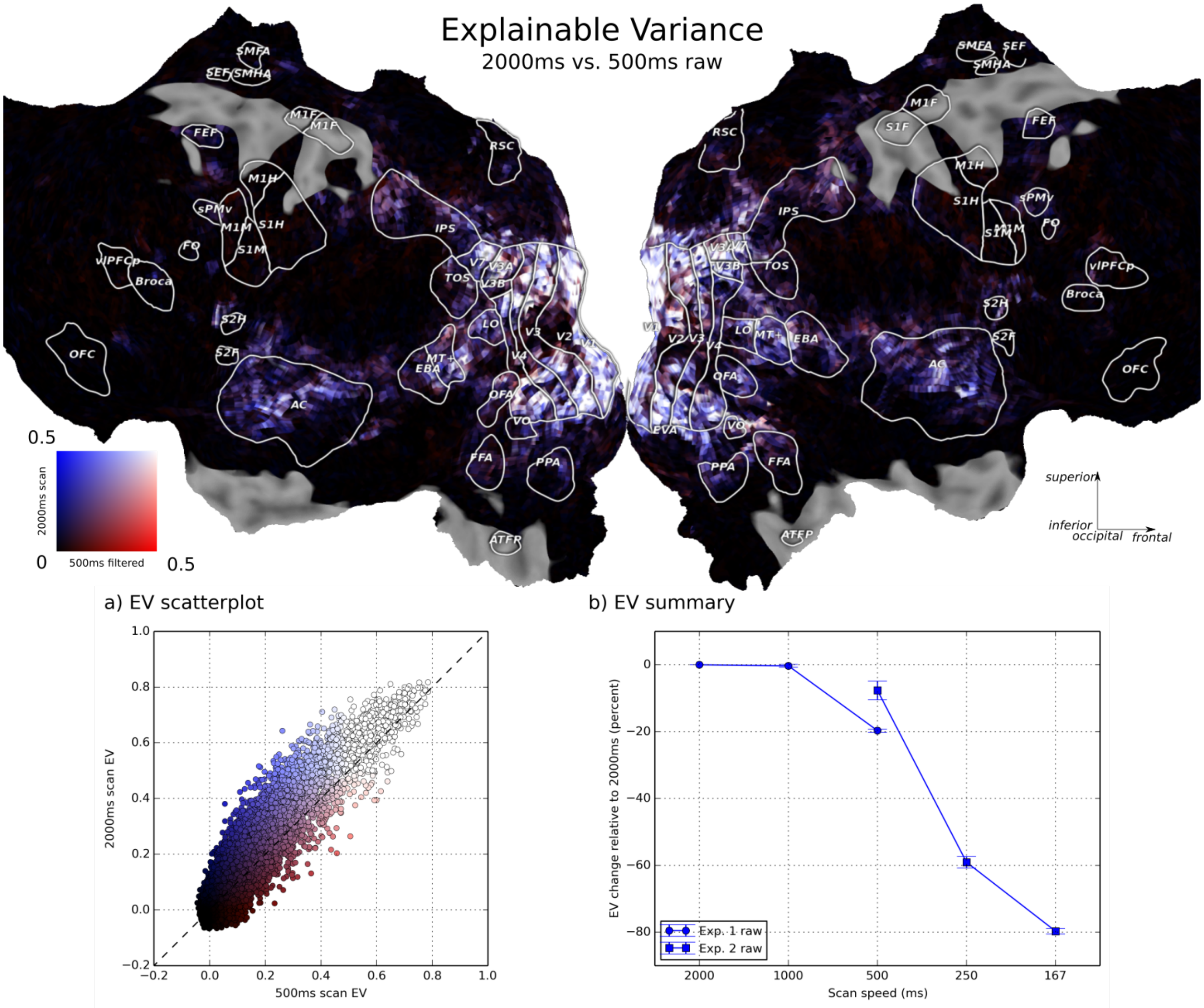
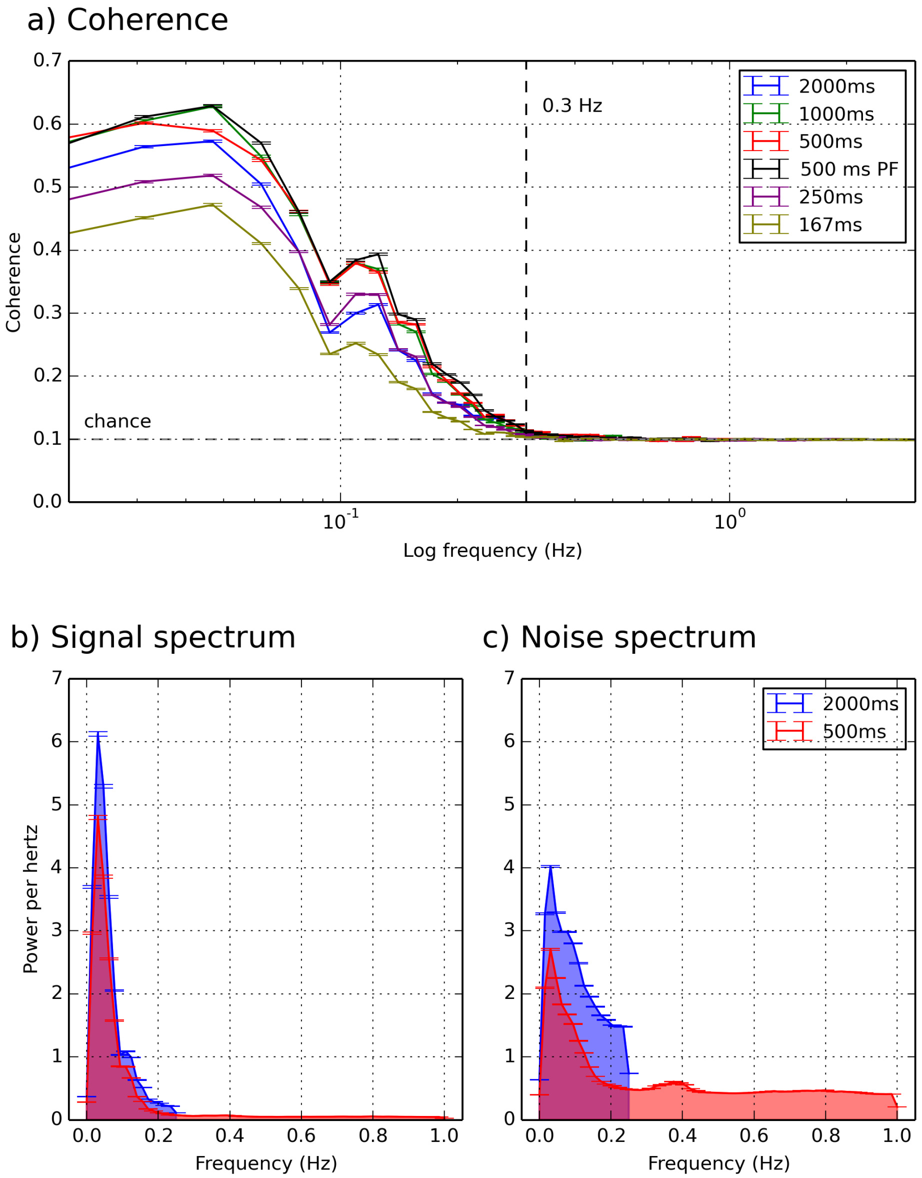


*Figure 4.2*: *Total explainable variance*

Total explainable variance (EV) is a measure of the functional SNR in each voxel. EV varies between 0 for uncorrelated signal across stimulus repetitions, to 1 for same timecourse across stimulus repetitions. Top, EV computed from the 2000ms and 500ms scans are plotted on a cortical flatmap for one subject. Blue indicates areas where the 2000ms scan has better EV than the 500ms scan, red where 500ms is better, white where both are high, and black where both are low. The naturalistic audiovisual stimulus drives repeatable signal primarily in sensory areas. The 2000ms scan is better in the majority of cortex, consistent with SNR loss due to acceleration. (A) Each point represents one voxel's EV computed from the 2000ms vs. 500ms scan. The majority of voxels show better performance for the 2000ms scan (blue). (B) A summary of the effect of scan speed and EV. The percent difference in EV for the top 5% voxels are shown for each scan speed. EV decreases with faster scan speed.



a) Coherence

b) Signal spectrum

c) Noise spectrum

*Figure 4.3: Coherence, Signal, and Noise spectra*
The signal spectrum is computed by taking the power spectrum density (PSD) of the average timecourse across repeats. The noise spectrum is estimated by taking the PSD of the residuals. Coherence normalizes the SNR to lie between 0 and 1 for all frequency bins. (A) Coherence plots are shown for each scan speed. The bulk of repeatable signal in fMRI lies in frequencies below 0.3 Hz. Coherence exceeds the single-band 2000ms scan across all frequencies for all but the 250ms and 167ms scans. (B) The signal spectrum for the 500ms scan is marginally lower than the 2000ms scan. (C) The total noise power is also higher (red area), but is spread over a much broader frequency range. Noise power is located primarily in high frequencies with no signal power. Thus, total SNR can be improved by filtering away high frequency noise.

### 4.3.2 Coherence spectra

To visualize the frequency content of stimulus-driven BOLD signal, we plotted the coherence spectra of the most repeatable voxels (Figure 4.3). We found that coherence is high for all sequences at low frequencies, but falls to chance level at about 0.3Hz. Low frequency coherence (0-0.1 Hz) is between 0.5 and 0.7 for all except the fastest sequence (TR=167ms), which has lower coherence than the other sequences at all frequencies. This analysis suggests that the BOLD responses contain no usable information at frequencies faster than 0.3Hz (corresponding to TR=1666ms).

Despite the higher acceleration factor, our results suggest that the 500ms scan has higher low-frequency coherence than the 2000ms scan. To further examine this effect, we separately plotted the signal and noise power spectra for the 2000ms and 500ms sequences (Figure 4.3). Here we see that the signal spectra for the two sequences are nearly identical, with both falling to chance at around 0.3Hz. However, the noise spectrum for the faster sequence is more distributed than the noise spectrum for the slower sequence. In other words, more of the total noise is in the highest frequencies where there is little or no signal power. By applying a low-pass filter to the voxel data as described in section 4.2.5, it may be possible to retain much of the signal power while eliminating the noise.

To illustrate this effect, we replotted the signal and noise spectrum for the 500ms scan with naive subsampling and filtered resampling. Naive subsampling simply selects every $n$th time sample in order to downsample the signal. Filtered resampling windows the frequency spectrum of the signal before subsampling. Subsampling aliases high frequencies into lower frequencies, altering the signal and noise spectra. Filtered sampling conserves the original noise and signal spectra well. As illustrated in Figure 4.4 (green line), naive subsampling of the 500ms signal results in a noise spectra that is qualitatively very similar to the 2000ms acquisition. Thus, the main difference in SNR between the 2000ms and 500ms scans is the subsampling of the BOLD signal. One might expect this relationship to break down with faster scan speeds. Indeed, we observe this effect already at 250ms (MB=8). At this multi-band factor, the signal spectrum of the subsampled signal is lower than the 2000ms acquisition. The noise spectrum however remains the same. Given these observations, high speed scanning may improve the SNR of the BOLD response by spreading noise power into unusable high frequencies. Correct filtering and resampling would eliminate this noise power.

To test our hypothesis that filtered signals improve EV, we recomputed EV for data which were low-pass filtered at 0.3 Hz (Figure 4.5, green line). We found a 29% increase in EV relative to the 2000ms scan for the 1000ms and 500ms scans ($p < 1 \times 10^{-10}$). Overall, the 500ms sequence from experiment 2 had the highest average EV for these voxels, 40% higher than the 2000ms sequence ($p < 1 \times 10^{-10}$). The fastest sequences (250ms and 167ms) had lower total EV than the other sequences, even after filtering. These results suggest that the amount of usable BOLD signal can be maximized by scanning at a TR of around 500ms (acceleration factor of 4) and then temporally low-pass filtering the data at 0.3 Hz.

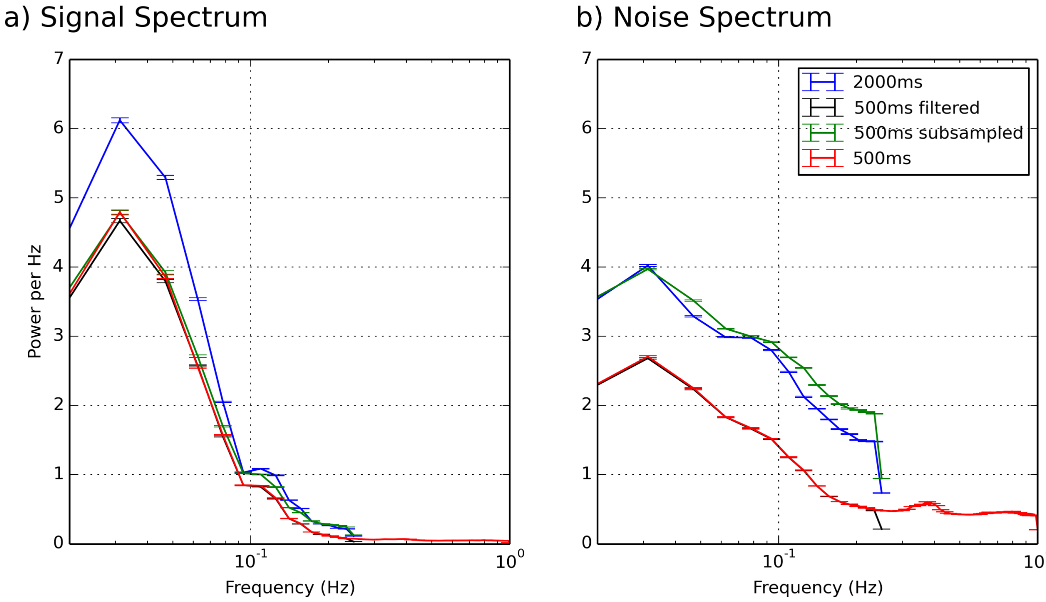


*Figure 4.4*: *Filtered downsampling vs. subsampling*
fMRI images are snapshots in time; sampling introduces aliasing artifacts unless the timecourses are filtered correctly. The signal and noise spectra are plotted for the 2000ms and 500ms scans, along with the spectra of naive subsampled data and filtered downsampled data. Naive subsampling selects every $n$th sample to downsample the signal. Filtered downsampling first windows the signal in fourier space before downsampling. If the 500ms scan is filtered then downsampled (red), the signal and noise spectra match the original 500ms scan and the high frequency noise is eliminated. If the 500ms scan is subsampled with no filtering, the noise power is dramatically increased. All high frequency noise power has been aliased into the lower frequencies and the noise spectrum is remarkably similar to the 2000ms scan.

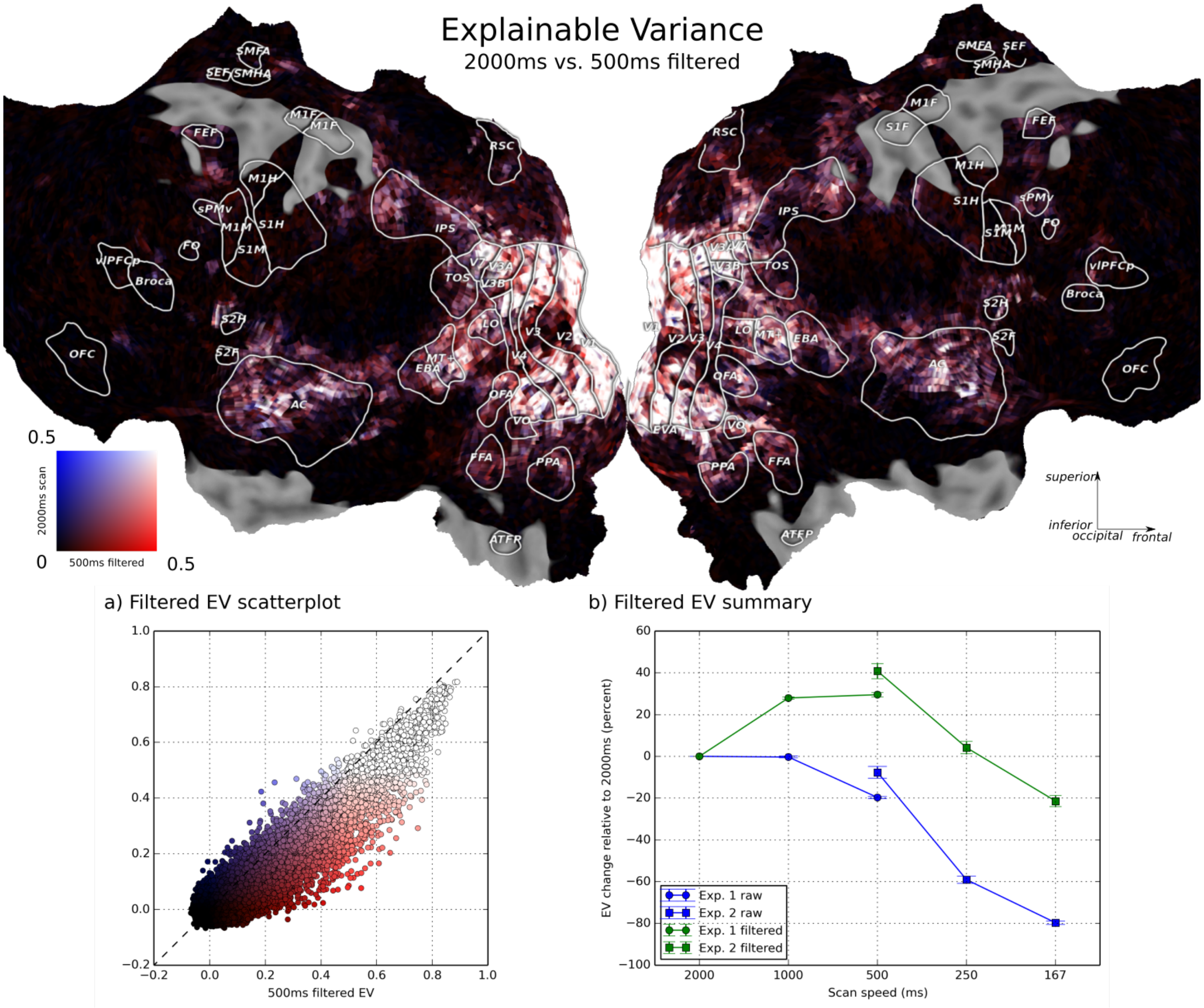


*Figure 4.5*: *Filtered explainable variance*
EV was recomputed for low-pass filtered data. Top, EV from the 2000ms scan is plotted against the filtered 500ms scan on the cortical flatmap. Filtered data recovers much of the EV loss, and dramatically improves total EV across most of the brain. (A) The majority of voxels now have better EV in the filtered 500ms scan compared to the 2000ms scan. (B) Filtering significantly improves EV for all fast scans, $p < 1 \times 10^{-10}$. EV shows a 28% to 40% improvement for the 1000ms and 500ms scans compared to the 2000ms scan ($p < 1 \times 10^{-10}$).

### 4.3.3 Timepoint correlation

In the previous analyses we showed that fast scans do not contain more signal than slow scans. Instead, the faster sequences spread the measurement noise across a wider spectrum allowing their elimination with filtering. However, coherence analysis and total explainable variance are both insensitive to blur. Smoothing the data across time would improve fSNR at the expense of spatial and temporal specificity. One potential goal of scanning faster is reducing feedback delay in a real-time paradigm. Evaluating the temporal specificity of BOLD would allow us to estimate the lag between measurement and decoded result in such a paradigm. To directly investigate the temporal specificity, we performed a timepoint classification analysis. In this analysis, we attempt to sort timepoints in order of their appearance based on the voxel response. In order to put all the scan sequences on equal footing we upsampled all responses to 10Hz.

Figure 4.6 shows the timepoint classification curves for all subjects. These curves peak at an offset of 0 ms, indicating that about 10% of the time we were able to identify timepoints within 100ms. Classification errors with offsets greater than 1.5 seconds are nearly zero. For the 1000ms scan the peak accuracy is somewhat lower and the classification offset is somewhat broader than the 2000ms scan. The 500ms scan sequence has an even lower and broader accuracy peak. This indicates that without filtering, faster scans have less precise temporal information than slower scans.

However, low-pass filtering the fast scans to remove high frequency noise power substantially improves time-point classification performance. With the filtered data, the accuracy curve is noticeably sharper and higher; the 500ms scan achieves a peak performance of 13% in one subject. This indicates that filtering out high frequency noise substantially improves timepoint classification performance.

To quantify the effect of filtering on time-point classification, we fit a gaussian to each accuracy curve and plotted the full-width half-max (FWHM) (Figure 4.6). For unfiltered data, the FWHM is larger with increasing scan speeds; this indicates that classification errors have larger offsets and lower peak performance for the faster scans. However, filtering the data before performing time-point correlation significantly decreases FWHM for faster scans across all subjects ($p < 1 \times 10^{-5}$). The 1000ms and two 500ms scans were significantly better than the 2000ms scan ($p < 0.01$) with filtering. However, this result varies on a per-subject basis and may require tuning for the each subject. Surprisingly, this analysis showed that the temporal accuracy of fMRI data can greatly exceed the time resolution of the scan. Even for the TR=2000ms scan, the timepoint correlation accuracy achieved a FWHM of 850ms. This means that the classifier had a 76% accuracy for matching a 100ms time sample within 850 ms of the target.

Our choice of the 0.3 Hz filter cutoff (section 4.2.5) was a result of a rough estimate of signal power from the coherence analysis. To quantitatively identify the best filter, we performed a grid search on the filter bandwidth and cutoff frequency to optimize FWHM. Since the timepoint correlation metric is not biased by temporal smoothing, optimizing for FWHM will retain low-coherence signal that may reduce total fSNR in a single voxel. This

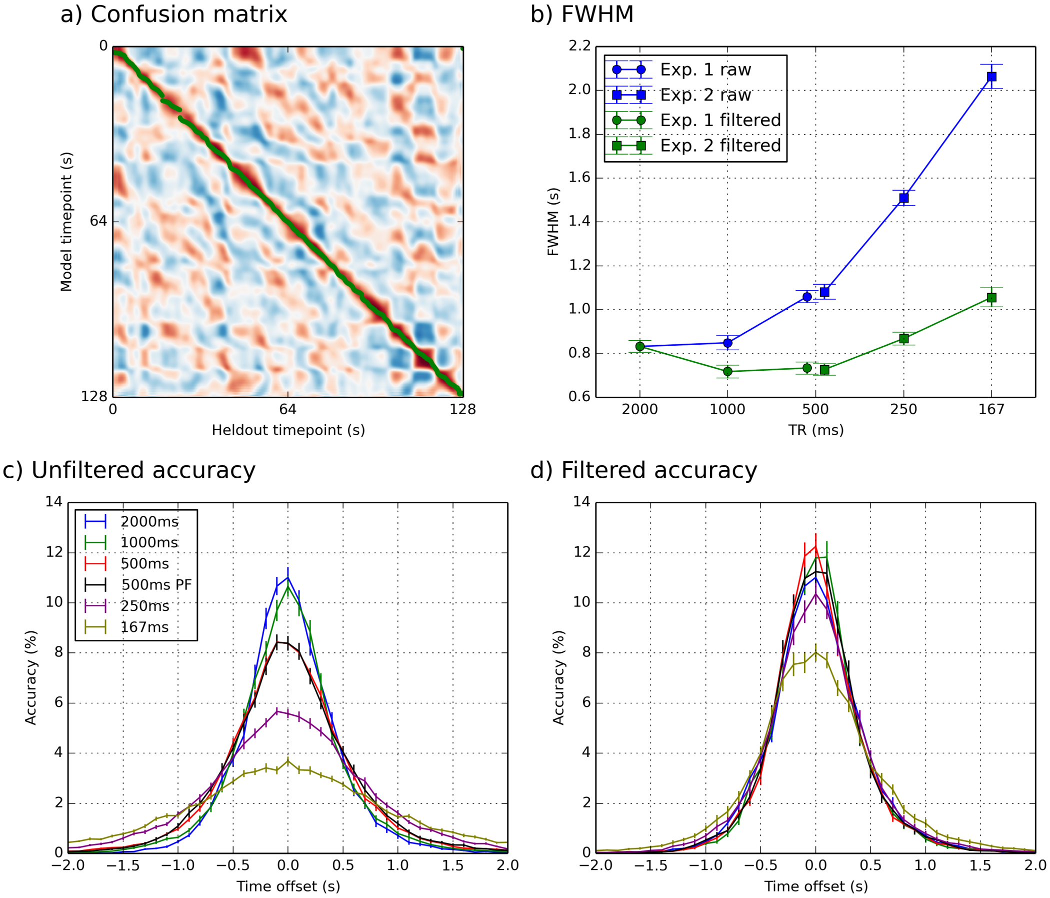


*Figure 4.6*: *Timepoint correlation*

We create a timepoint correlation classifier to evaluate the temporal specificity of the BOLD signal. The timecourse across 9 repetitions are averaged, and the timepoints of the held-out repeat are sorted in time based on voxelwise correlation with the average. All scans are upsampled to 10 Hz to account for different TR. (A) The confusion matrix of the timepoint correlation classifier. Red is for positive correlation, green dots indicate the maximum timepoint correlation. The majority of timepoints are correctly classified (green dots on diagonal). (C) and (D) Accuracy at time offsets from -2s to 2s are plotted for each scan speed. Peak accuracy and temporal specificity drops monotonically with increasing scan speed. However, peak accuracy is recovered and improved for all scans if the timecourses are filtered with a 0.3 Hz low-pass filter. (B) A gaussian function is fit to each accuracy curve to summarize the temporal specificity. Higher FWHM indicates less temporal accuracy. FWHM increases with increasing scan speed, but is reduced for the filtered 1000ms and 500ms scans ($p < 0.01$).

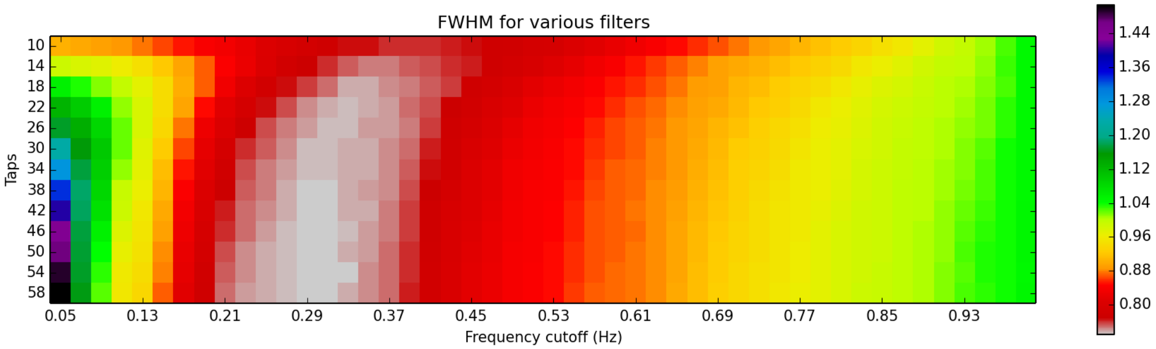


*Figure 4.7*: *Optimal Filter*
The timepoint correlation metric is insensitive to temporal smoothing. Low coherence signals may still be informative if aggregated across voxels. Therefore, we identified the optimal filter by performing a grid search on filter frequency and bandwidth to minimize FWHM. The TR=500ms scan from experiment 1 was filtered with various frequency cutoffs and bandwidths, then timepoint correlation was performed. Minimal FWHM occurs at cutoff frequency 0.29 Hz and a 29 second bandwidth.

low-coherence signal may still be usable information if signals are aggregated across voxels. We find that the FWHM is minimal in the TR=500ms scan with data filtered at 0.29 Hz with a 29 second bandwidth filter. (Figure 4.7).

### 4.3.4 High speed stimulus

In Figure 4.3, we claim that BOLD frequency content falls to zero at about 0.3 hz. However, it is possible that our BOLD frequency spectrum is biased by the stimulus. Since we chose to use a natural movie stimulus, the 1/f temporal spectrum of the stimulus (Dong and Atick 1995) may reduce the amount of high frequency information present in the BOLD signal.

To address this issue, we scanned one subject with a scrambled natural movie stimulus. In this high frequency stimulus, we spliced 3-frame sections of the natural movies with sound into a scrambled order. This results in a movie with a mostly uniform frequency content, and a fundamental frequency at 8 Hz. We chose 8 Hz because this frequency drives the strongest BOLD response (Thomas and Menon 1998) and is well outside the Nyquist frequency of the fastest scan used (1 Hz, 500ms TR). We scanned this subject with the 2000, 1000, and 500ms scans from experiment 1 and computed EV and coherence. Plotting the coherence spectra of this high-speed stimulus reveals no additional power above 0.3 Hz (Figure 4.8). The 500ms scan had significantly more low-frequency coherence than the other two scans, but still fell to chance at 0.3 Hz (Figure 4.8). In addition, peak accuracy for timepoint correlation for the 500ms scan was not significantly different from 1000 and 2000ms ($p > .2$ for both), and

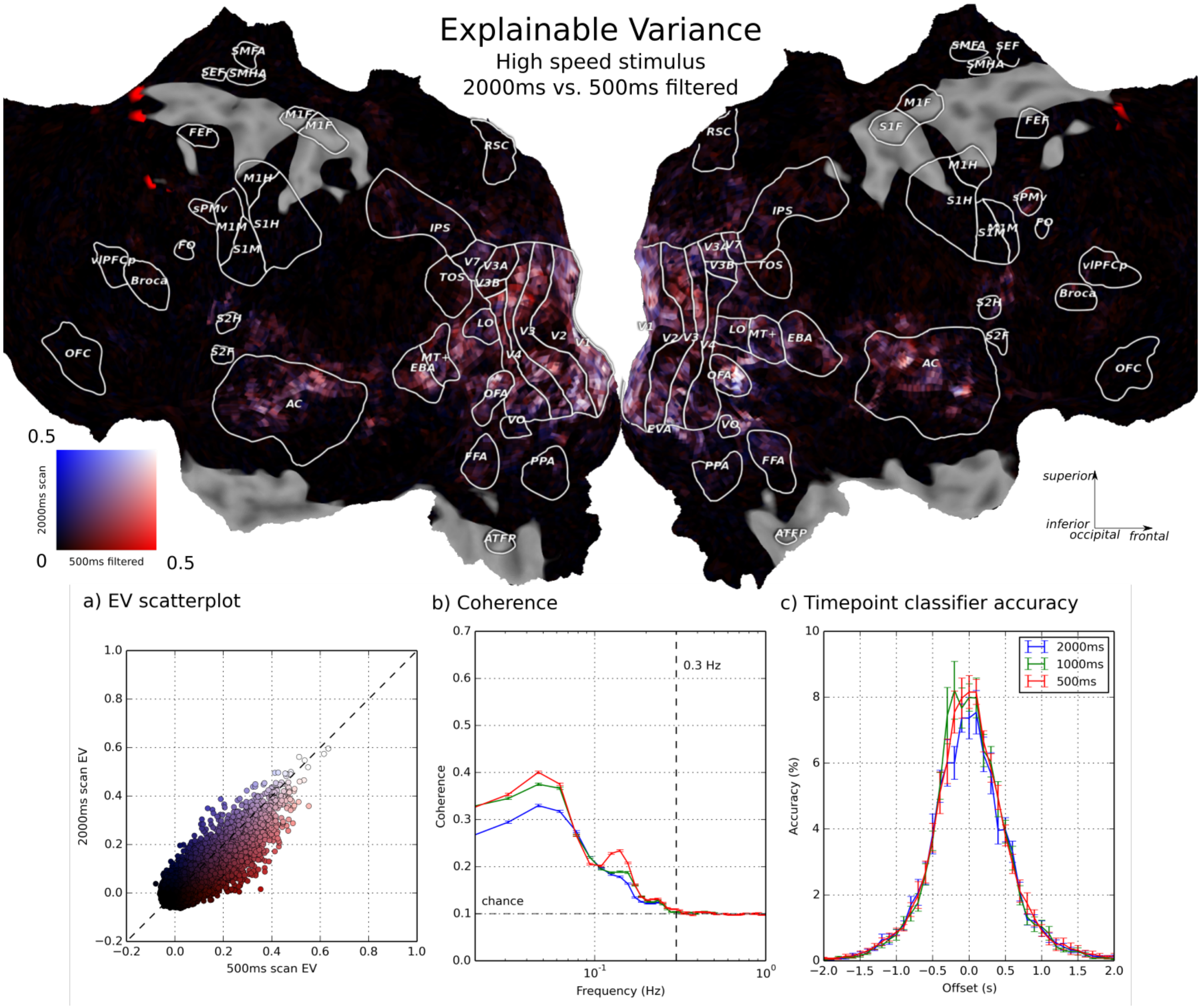


*Figure 4.8*: *High speed stimulus*

If the brain operates as a filter on input, our coherence analysis may have a low frequency bias. This is because our natural stimulus has a 1/f temporal frequency spectrum (Dong and Atick 1995). To control for this low frequency bias, we collected data from one subject with a high frequency stimulus composed of scrambled natural movies. Top, EV computed from the downsampled 500ms scan is plotted on a single subject's flatmap. EV is much lower for scrambled movies compared to coherent natural movies. (A) A scatter plot for each voxel shows the EV as computed from the 2000ms scans and the non-downsampled 500ms scans. Total EV is lower than coherent movies, and downsampled 500ms scans have improved repeatability. (B) No additional high frequency information is present in the coherence spectrum despite the fast stimulus. (C) Timepoint classifier accuracy is lower than the natural stimulus.

all three were significantly lower than the naturalistic stimulus-driven response ($p < .001$ for all).

### 4.3.5 Physiological noise

The highest speed scans used in this study allow the direct observation of cardiac phase without aliasing. Typically, the cardiac signal is aliased down into lower frequencies (Oppenheim and Willsky 1996) that reduce the SNR of low frequency BOLD. To test this effect, we computed coherence spectra and EV on data without physiological regression (Figure 4.9). Overall, physiological regression had relatively minor impact on all analyses. Total EV across all subjects was improved for the 2000ms scan (5% improvement for top 5% voxels, $p < 1 \times 10^{-5}$, Figure 4.9) with a decreasing effect at faster speeds.

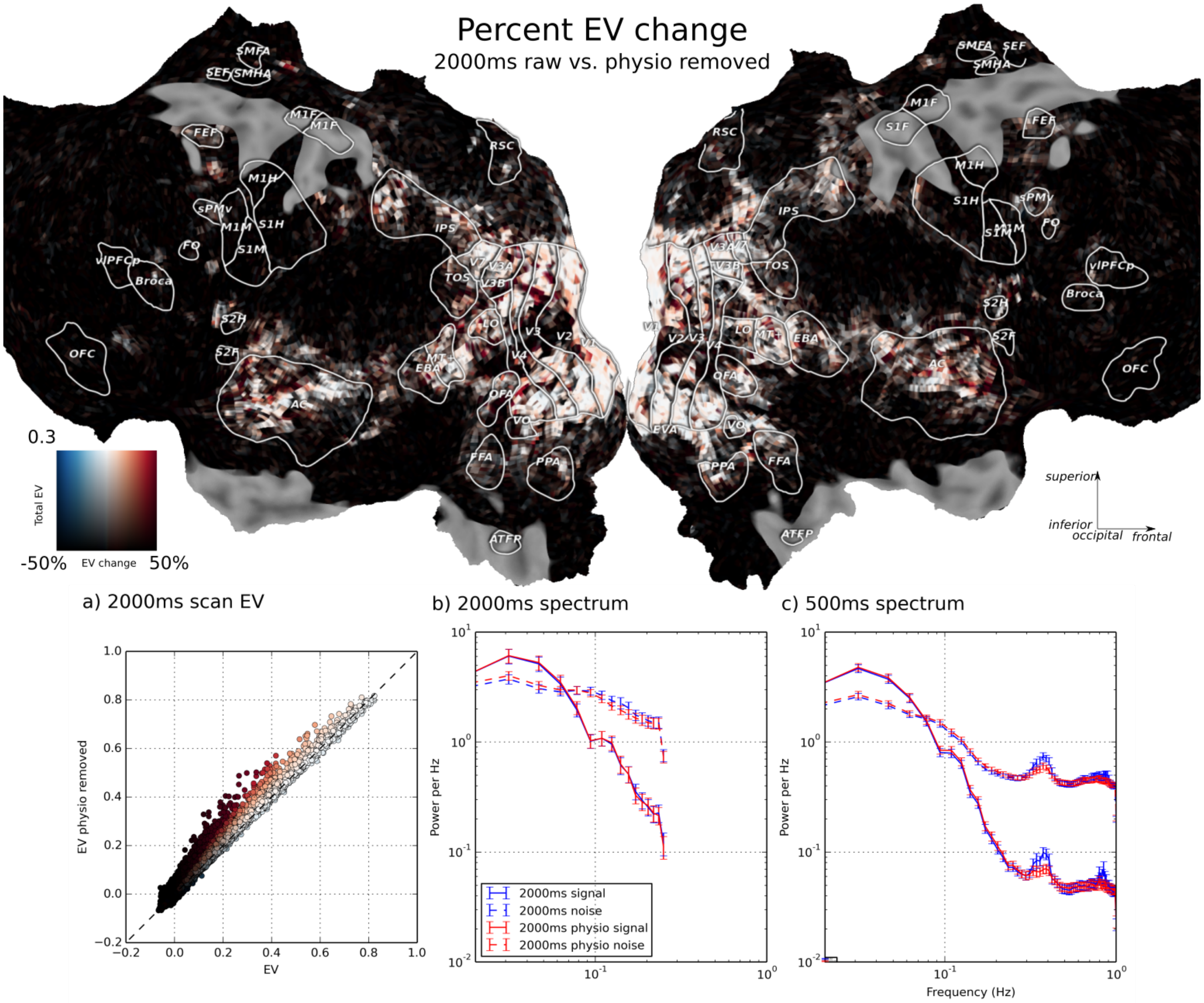
Since physiological noise is uncorrelated with the stimulus presentation (the heart beat phase and frequency is not the same for each repetition of the stimulus), we did not expect to find any changes in the coherence spectrum. However, any heart rate power should be visible in the noise spectrum. Indeed, a small increase in noise power can be seen at 0.4 and 0.8 Hz, corresponding to breathing (5s period) and heart rate (potentially aliased down from 1.2 Hz). Physiological noise regression did reduce the noise power at these two frequencies. However, this makes no effect on filtered analyses for the faster scans. The low-pass filter eliminates all frequencies above 0.3 Hz, thereby eliminating any noise contribution from physiological sources.

## 4.4 Discussion

In this study, we attempted to isolate the frequency response of the HRF in the presence of an audiovisual stimulus. We also attempted to optimize our measurement of the response using the latest multiband and simultaneous multi-slice imaging methods. We replicated the fact that the BOLD response spectrum is heavily biased toward low frequencies (Glover 1999; Boynton et al. 1996). Very little usable signal is left above 0.3 Hz, corresponding to an ideal sampling rate of 1.6 seconds. However, we found that supersampling the BOLD signal at a higher rate (1 Hz) improves SNR. The mechanism behind this SNR improvement is a sampling rate dependent noise power modulation which allows the removal of high frequency noise with a filter to improve low frequency SNR. We found that the ideal tradeoff between acceleration-induced signal loss and supersampling signal gain is around TR=500ms with a multiband factor of 4.

### 4.4.1 Timepoint classification

The timepoint correlation analysis leads to some surprising and seemingly counterintuitive results. Although the hemodynamics have no response faster than 0.3 Hz, we can accurately identify 100ms timepoints within 800ms of where they actually occur. This result



a) 2000ms scan EV

b) 2000ms spectrum

c) 500ms spectrum

*Figure 4.9*: *Physiological noise*
Scan speeds in excess of 500ms can directly measure BOLD variation due to heart rate. We test the effect of physiological noise removal on EV, signal, and noise spectrum. Top, EV computed for the 2000ms scan with and without physiological noise removal is plotted across a single subject flatmap. EV improvement is evident along sulci and in known locations of veins. EV improvement is strongest for the 2000ms scan (shown), since physiological noise is aliased into signal frequencies. (A) Each dot indicates the EV of a single voxel. EV computed from noise-removed data is plotted on the y-axis. A small improvement is observed in many voxels. (B) The signal and noise spectra are plotted for the 2000 ms sequence. Physiological noise removal has nearly no effect on the signal spectrum, but reduces noise power in a broad range of middle to high frequencies. The physiological factors alias down into signal frequencies; removal through regression improves the final SNR. (C) Physiological noise removal makes a more pronounced effect on the noise spectrum for the 500ms scan. Putative noise frequencies at 0.4 Hz and 0.8 Hz are reduced in power.

is due to the aggregation of information across all the voxels in the decoder. As a voxel's time series is upsampled, its individual time course does not provide any additional information above the 0.3 Hz cutoff frequency of our filter. However, the decoder is built using the 1000 best voxels in each fold. Additional temporal specificity is encoded in the trajectory of different voxels, even though each individual voxel provides no additional information. This additional trajectory information across the whole decoding set allows the enhanced matching of timepoints well outside the frequency response of any single voxel.

This result is extremely promising for real-time MRI feedback experiments. Despite the extremely low temporal resolution of fMRI, very fine grained temporal differences can be captured using slow data. By aggregating information across many voxel trajectories, it is possible to make predictions about outcome at a much faster pace than the traditional 2-3s TR. With the use of multi-band acceleration, the boost in SNR from supersampling allows lower decoding latency and shorter feedback cycles. Further advances in data processing and algorithm development need to be made in order to process and predict real-time multiband accelerated data for decoding. For example, image reconstruction of the TR=167ms sequence took nearly 10 minutes, four times slower than the collection time.

### 4.4.2 Acceleration errors

One significant problem with multi-band and simultaneous multi-slice sequences is that multiple slices are collected in a single excitation. Since slices are intentionally aliased together in multiband imaging, minor errors in slice dealiasing could lead to signal misattribution (Larkman et al. 2001). For fMRI studies, this presents the possibility that signal from one anatomical region may be misattributed to signal from another. One effect that is extremely damaging to multiband acceleration is subject movement, especially during the pre-acquisition calibration phase. During this phase, the sequence automatically collects single-band reference images that allow it to allocate signal to each slice using coil sensitivities. If the reference images are misaligned with any subsequent images, the sensitivity profiles become invalid and misattribution becomes a significant problem.

The generalization of our observed SNR improvement finding may depend on the cooperation of the subject. Scanning a naive subject with a multi-band factor of 4 (500ms TR) will not necessarily result in higher SNR relative to 2000ms TR. Large subject movements throughout the scan or during reference scans might reduce the SNR improvement. In order to address this, we have developed a rigid plastic insert that dramatically reduces subject motion (chapter 3), and allows the use of high multi-band acceleration factors with naive subjects. This rigid plastic insert virtually eliminates all subject motion and allows the use of multiband and simultaneous multi-slice scans with much less signal misattribution than without.

## 4.5 Conclusion

Our analyses show that under a stimuli-driven paradigm, the BOLD signal has no stationary response faster than 0.3 Hz. However, SNR at frequencies lower than 0.3 Hz can be improved by temporal supersampling with multiband imaging. Our coherence analysis shows that signal repeatability drops to chance levels at around 0.3 Hz, which is only slightly higher than the Nyquist frequency of ordinary non-accelerated scans. Although multiband alone results in a reduction in total EV compared to non-accelerated scans, proper filtering dramatically improves EV for some of the faster scans. This suggests that fast fMRI allows us to collect more samples that have independent noise. In addition, scanning faster unwraps aliased noise from heart rate and respiration into higher frequency bins, allowing us to eliminate the noise with a low pass filter.

Given the effects observed in our analysis, the ideal scan sequence for a stimulus-driven paradigm is a multiband 4x scan with TR=500ms. Uncorrelated measurement noise can be filtered away because the high speed scan supersamples the signal. The 4-fold multiband acceleration factor results in minimal loss of signal power, may improve total SNR by up to 40%.

# Chapter 5

# Conclusion

Since its conception in 1990, fMRI has proven to be a pivotal tool in the study of neuroscience. It is a technique that is still being developed. Exciting innovations such as multiband imaging allow us to study the brain at higher and higher spatial or temporal resolution. Unfortunately, fMRI is not a magic bullet. The imaging of hemodynamics limits us to only modeling the metabolic activity of the brain, rather than the underlying neural coding. Fast changes in firing cannot be measured by fMRI, and millisecond timing resolution will never be achieved with hemodynamics.

Despite these limitations, predictive models of neurons can still be validated with fMRI. The high spatial dimensionality of fMRI guarantees that organization that varies on a large spatial scale can be observed and predicted. Critically, this prediction of activity hints at a future where computers can read our thoughts. Interaction between humans and computers would not be limited by slow keyboard interfaces, but could be a seamless conversation between brain and machine.

In this thesis, I contributed to the development of fMRI as a tool. I identified some of limitations of MRI, but showed that interesting applications are still possible. I attempted to pave the way for a proof of concept application, where MRI can be used to decode conscious intention.

In chapter 2, I presented a new visualization tool that streamlines surface visualization of MRI data. Not only does it provide an accurate and convenient way to plot volumetric intersections of mesh surfaces, it creates a visualization that is easily shareable with other scientists and the general public alike. This tool also supports interactive data exploration through the dynamic flattening of surfaces, and the simultaneous visualization of the stimulus that elicited brain response. This open source software project promotes the ideals behind open science by encouraging users to share their results and data in an accessible form. My hope is that this software project will continue to be developed by myself and other researchers to expand its functionality and utility.

In chapter 3, I used commodity 3D printers to develop a personalized head stabilization device. This device virtually eliminates all head movement, a problem that plagues the latest multiband sequences. These sequences rely on the static placement of the head to accurately locate signal origins. Although this application was tailored for fMRI imaging, these devices could be generated for medically crucial placement of devices around the body. Such plastic inserts could dramatically improve the outcome of things like radiation therapy for cancer, or fiducial markers for orthopedic surgery.

In chapter 4, I utilized multiband scanning sequences to evaluate the temporal frequency response of the BOLD signal. I found that hemodynamics retain nearly no information that occur faster than 2 seconds; however their spatial distribution is extremely varied and high dimensional. I found that despite the extremely low temporal cutoff of BOLD, the spatial variation can be used to perform very fast classification. This means that realtime feedback experiments can achieve feedback cycles shorter than 10 seconds.

I hope that these technical innovations will lead to the development of a real-time control experiment where the subject can drive a robot car around with their minds as easily as using a joystick. Such an experiment would validate of many theories of spatial representation in the brain, and would provide a tantalizing glimpse into the future of neuroscience.

# Bibliography

Alecci, M., Collins, C. M., Smith, M. B., and Jezzard, P. (2001). Radio frequency magnetic field mapping of a 3 tesla birdcage coil: Experimental and theoretical dependence on sample properties. *Magnetic Resonance in Medicine*, 46(2):379–385.

Andersson, P., Pluim, J. P. W., Viergever, M. A., and Ramsey, N. F. (2013). Navigation of a telepresence robot via Covert Visuospatial Attention and real-time fMRI. *Brain Topography*, 26(1):177–185.

Arias, A. C., Lustig, S., Flynn, A. M., and Corea, J. (2014). MRI receiver coil providing an enhanced signal-to-noise ratio.

Bettinardi, V., Scardaoni, R., Gilardi, M. C., Rizzo, G., Perani, D., Paulesu, E., Striano, G., Triulzi, F., and Fazio, F. (1991). Head holder for PET, CT, and MR studies. *Journal of computer assisted tomography*, 15(5):886–92.

Boynton, G. M., Engel, S. a., Glover, G. H., and Heeger, D. J. (1996). Linear systems analysis of functional magnetic resonance imaging in human V1. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 16(13):4207–21.

Buxton, R. B., Uluda, K., Dubowitz, D. J., and Liu, T. T. (2004). Modeling the hemodynamic response to brain activation. In *NeuroImage*, volume 23.

Çukur, T., Nishimoto, S., Huth, A. G., and Gallant, J. L. (2013). Attention during natural vision warps semantic representation across the human brain. *Nature neuroscience*, 16(6):763–70.

Chen, L., Vu, a., Xu, J., Moeller, S., Ugurbil, K., Yacoub, E., and Feinberg, D. (2014). Evaluation of Highly Accelerated Simultaneous Multi-Slice EPI for FMRI. *NeuroImage*.

Cox, R. W. (1996). AFNI: software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and biomedical research, an international journal*, 29(3):162–173.

Dale, A. M., Fischl, B., and Sereno, M. I. (1999). Cortical surface-based analysis. I. Segmentation and surface reconstruction. *NeuroImage*, 9(2):179–194.

Dong, D. and Atick, J. (1995). Statistics of natural time-varying images. *Computation in Neural Systems*, 6(3):345–358.

Edward, V., Windischberger, C., Cunnington, R., Erdler, M., Lanzenberger, R., Mayer, D., Endl, W., and Beisteiner, R. (2000). Quantification of fMRI artifact reduction by a novel plaster cast head holder. *Human brain mapping*, 11(3):207–13.

Engel, S. A., Glover, G. H., and Wandell, B. A. (1997). Retinotopic organization in human visual cortex and the spatial precision of functional MRI. *Cerebral Cortex*, 7(2):181–192.

Feinberg, D. a., Moeller, S., Smith, S. M., Auerbach, E., Ramanna, S., Gunther, M., Glasser, M. F., Miller, K. L., Ugurbil, K., and Yacoub, E. (2010). Multiplexed echo planar imaging for sub-second whole brain FMRI and fast diffusion imaging. *PloS one*, 5(12):e15710.

Felleman, D. J. and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex*, 1(1):1–47.

Fischl, B. and Dale, A. M. (2000). Measuring the thickness of the human cerebral cortex from magnetic resonance images. *Proceedings of the National Academy of Sciences of the United States of America*, 97(20):11050–11055.

Fischl, B., Liu, A., and Dale, A. M. (2001). Automated manifold surgery: Constructing geometrically accurate and topologically correct models of the human cerebral cortex. *IEEE Transactions on Medical Imaging*, 20(1):70–80.

Friston, K., Ashburner, J., Frith, C., Poline, J., Heather, J., and Frackowiak, R. (1995). Spatial registration and normalization of images. *Human brain mapping*.

Friston, K. J., Williams, S., Howard, R., Frackowiak, R. S., and Turner, R. (1996). Movement-related effects in fMRI time-series. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 35(3):346–55.

Friston, K. J., Zarahn, E., Josephs, O., Henson, R. N., and Dale, A. M. (1999). Stochastic designs in event-related fMRI. *NeuroImage*, 10(5):607–19.

Geelnard, M. (2009). OpenCTM —the Open Compressed Triangle Mesh file format.

Glasser, M. F., Sotiropoulos, S. N., Wilson, J. A., Coalson, T. S., Fischl, B., Andersson, J. L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J. R., Van Essen, D. C., and Jenkinson, M. (2013). The minimal preprocessing pipelines for the Human Connectome Project. *NeuroImage*, 80(16):105–124.

Glover, G. H. (1999). Deconvolution of impulse response in event-related BOLD fMRI. *NeuroImage*, 9(4):416–29.

Glover, G. H., Li, T. Q., and Ress, D. (2000). Image-based method for retrospective correction of physiological motion effects in fMRI: RETROICOR. *Magnetic Resonance in Medicine*, 44(1):162–167.

Goebel, R. (1997). BrainVoyager 2.0: From 2D to 3D fMRI analysis and visualization. *Neuroimage*, 5(S635).

Greve, D. N. and Fischl, B. (2009). Accurate and robust brain image alignment using boundary-based registration. *NeuroImage*, 48(1):63–72.

Griswold, M. A., Jakob, P. M., Heidemann, R. M., Nittka, M., Jellus, V., Wang, J., Kiefer, B., and Haase, A. (2002). Generalized autocalibrating partially parallel acquisitions (GRAPPA). *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 47(6):1202–10.

Haines, K., Smith, N. B., and Webb, A. G. (2010). New high dielectric constant materials for tailoring the B1+ distribution at high magnetic fields. *Journal of magnetic resonance (San Diego, Calif. : 1997)*, 203(2):323–7.

Halchenko, Y. O. and Hanke, M. (2012). Open is Not Enough. Let's Take the Next Step:

An Integrated, Community-Driven Computing Platform for Neuroscience.

Hansen, K. a., Kay, K. N., and Gallant, J. L. (2007). Topographic organization in and near human visual area V4. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(44):11896–911.

Heeger, D. J. and Ress, D. (2002). What does fMRI tell us about neuronal activity? *Nature reviews. Neuroscience*, 3(2):142–151.

Hodgkin, A. L. and Huxley, A. F. (1952). A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerves. *J. Physiol.*, 117:500–544.

Hsu, A., Borst, A., and Theunissen, F. E. (2004). Quantifying variability in neural responses and its application for the validation of model predictions. *Network*, 15(2):91–109.

Huettel, S. A., Song, A. W., and McCarthy, G. (2009). *Functional magnetic resonance imaging*. Sinauer Associates.

Huth, A. G., Nishimoto, S., Vu, A. T., and Gallant, J. L. (2012). A Continuous Semantic Space Describes the Representation of Thousands of Object and Action Categories across the Human Brain. *Neuron*, 76(6):1210–1224.

Issa, E. B., Papanastassiou, A. M., and DiCarlo, J. J. (2013). Large-scale, high-resolution neurophysiological maps underlying FMRI of macaque temporal lobe. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 33(38):15207–19.

Jenkinson, M. (2004). Improving the registration of B0- disorted EPI images using calculated cost function weights. In *Tenth Int. Conf. on Functional Mapping of the Human Brain*.

Jenkinson, M., Bannister, P., Brady, M., and Smith, S. (2002). Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841.

Jenkinson, M. and Smith, S. (2001). A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, 5(2):143–156.

Kaas, J. H. (2012). Evolution of columns, modules, and domains in the neocortex of primates.

Kanwisher, N., McDermott, J., and Chun, M. M. (1997). The fusiform face area: a module in human extrastriate cortex specialized for face perception. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 17(11):4302–4311.

Larkman, D. J., Hajnal, J. V., Herlihy, A. H., Coutts, G. A., Young, I. R., and Ehnholm, G. (2001). Use of multicoil arrays for separation of signal from multiple slices simultaneously excited. *Journal of magnetic resonance imaging : JMRI*, 13(2):313–7.

Lauterbur, P. C. (1973). Image formation by induced local interactions. Examples employing nuclear magnetic resonance. *Clinical orthopaedics and related research*.

Le Bihan, D., Poupon, C., Amadon, A., and Lethimonnier, F. (2006). Artifacts and pitfalls in diffusion MRI. *Journal of magnetic resonance imaging : JMRI*, 24(3):478–88.

Logothetis, N. K. (2003). The underpinnings of the BOLD functional magnetic resonance imaging signal. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 23(10):3963–71.

Logothetis, N. K. (2008). What we can do and what we cannot do with fMRI. *Nature*, 453(7197):869–78.

Logothetis, N. K., Pauls, J., Augath, M., Trinath, T., and Oeltermann, A. (2001). Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412(6843):150–7.

MGH (2009). Morphometry Protocols.

Millman, K. J. and Brett, M. (2007). Analysis of functional magnetic resonance imaging in python. *Computing in Science and Engineering*, 9(3):52–55.

Mukamel, R., Gelbard, H., Arieli, A., Hasson, U., Fried, I., and Malach, R. (2005). Coupling between neuronal firing, field potentials, and FMRI in human auditory cortex. *Science (New York, N.Y.)*, 309(5736):951–4.

Nishimoto, S., Vu, A. T., Naselaris, T., Benjamini, Y., Yu, B., and Gallant, J. L. (2011). Reconstructing visual experiences from brain activity evoked by natural movies. *Current Biology*, 21(19):1641–1646.

Nishimura, D. G. (2010). *Principles of Magnetic Resonance Imaging.* Stanford University.

Oakes, T. R., Johnstone, T., Ores Walsh, K. S., Greischar, L. L., Alexander, A. L., Fox, A. S., and Davidson, R. J. (2005). Comparison of fMRI motion correction software tools. *NeuroImage*, 28(3):529–43.

Ogawa, S., Lee, T. M., Kay, A. R., and Tank, D. W. (1990). Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *Proceedings of the National Academy of Sciences of the United States of America*, 87(24):9868–9872.

Oppenheim, A. and Willsky, A. (1996). *Signals and Systems (2nd Edition).* Prentice Hall.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2012). Scikit-learn: Machine Learning in Python. *. . . of Machine Learning . . .*, 12:2825–2830.

Poldrack, R. A. (2007). Region of interest analysis for fMRI. *Social Cognitive and Affective Neuroscience*, 2(1):67–70.

Power, J. D., Barnes, K. A., Snyder, A. Z., Schlaggar, B. L., and Petersen, S. E. (2012). Spurious but systematic correlations in functional connectivity MRI networks arise from subject motion. *NeuroImage*, 59(3):2142–54.

Roy, C. S. and Sherrington, C. S. (1890). On the Regulation of the Blood-supply of the Brain. *The Journal of physiology*, 11(1-2):85–158.17.

Sereno, M. I. (1998). Brain mapping in animals and humans. *Current opinion in neurobiology*, 8(2):188–194.

Sereno, M. I., Dale, A. M., Reppas, J. B., Kwong, K. K., Belliveau, J. W., Brady, T. J., Rosen, B. R., and Tootell, R. B. (1995). Borders of multiple visual areas in humans revealed by functional magnetic resonance imaging. *Science (New York, N.Y.)*, 268(5212):889–893.

Setsompop, K., Gagoski, B. a., Polimeni, J. R., Witzel, T., Wedeen, V. J., and Wald, L. L. (2012). Blipped-controlled aliasing in parallel imaging for simultaneous multislice echo planar imaging with reduced g-factor penalty. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 67(5):1210–24.

Smith, S. M., Beckmann, C. F., Andersson, J., Auerbach, E. J., Bijsterbosch, J., Douaud, G., Duff, E., Feinberg, D. A., Griffanti, L., Harms, M. P., Kelly, M., Laumann, T., Miller, K. L., Moeller, S., Petersen, S., Power, J., Salimi-Khorshidi, G., Snyder, A. Z., Vu, A. T., Woolrich, M. W., Xu, J., Yacoub, E., Uurbil, K., Van Essen, D. C., and Glasser, M. F. (2013). Resting-state fMRI in the Human Connectome Project. *NeuroImage*, 80:144–168.

Thomas, C. G. and Menon, R. S. (1998). Amplitude response and stimulus presentation frequency response of human primary visual cortex using BOLD EPI at 4 T. *Magnetic Resonance in Medicine*, 40(2):203–209.

Tumbleston, J. R., Shirvanyants, D., Ermoshkin, N., Janusziewicz, R., Johnson, A. R., Kelly, D., Chen, K., Pinschmidt, R., Rolland, J. P., Ermoshkin, A., Samulski, E. T., and DeSimone, J. M. (2015). Continuous liquid interface production of 3D objects. *Science*.

Van Dijk, K. R. A., Sabuncu, M. R., and Buckner, R. L. (2012). The influence of head motion on intrinsic functional connectivity MRI. *NeuroImage*, 59(1):431–8.

Van Essen, D. C., Drury, H. a., Dickson, J., Harwell, J., Hanlon, D., and Anderson, C. H. (2001). An integrated software suite for surface-based analyses of cerebral cortex. *Journal of the American Medical Informatics Association : JAMIA*, 8(5):443–59.

Wandell, B. A. (1999). Computational neuroimaging of human visual cortex. *Annual review of neuroscience*, 22:145–173.

Wapler, M. C., Leupold, J., Dragonu, I., von Elverfeld, D., Zaitsev, M., and Wallrabe, U. (2014). Magnetic properties of materials for MR engineering, micro-MR and beyond. *Journal of magnetic resonance (San Diego, Calif. : 1997)*, 242:233–42.

Woo, M., Neider, J., Davis, T., and Shreiner, D. (1999). *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. Addison-Wesley Longman Publishing Co., Inc, Reading, MA, 1.2 edition.

Wu, M. C.-K., David, S. V., and Gallant, J. L. (2006). Complete functional characterization of sensory neurons by system identification. *Annual review of neuroscience*, 29:477–505.

Yablonskiy, D. A. and Haacke, E. M. (1994). Theory of NMR signal behavior in magnetically inhomogeneous tissues: the static dephasing regime. *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, 32(6):749–63.

Zaitsev, M., Dold, C., Sakas, G., Hennig, J., and Speck, O. (2006). Magnetic resonance imaging of freely moving objects: prospective real-time motion correction using an external optical motion tracking system. *NeuroImage*, 31(3):1038–50.