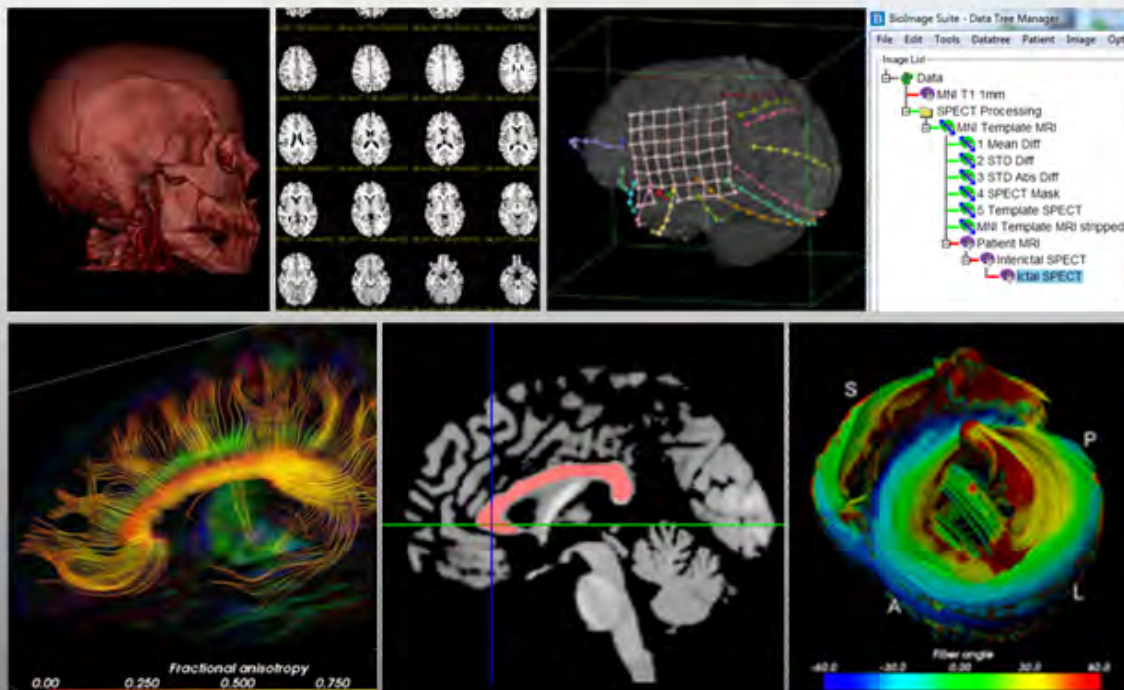


# Bioimage Suite User's Manual



Xenophon Papademetris  
Marcel Jackowski  
Alark Joshi  
Dustin Scheinost  
Cheryl Lacadie  
Marcello DiStasio  
Lawrence Staib



© Copyright 2008

X. Papademetris, M. Jackowski, N. Rajeevan, R.T. Constable, and L.H Staib.

Section of Bioimaging Sciences, Dept. of Diagnostic Radiology, Yale School of Medicine.

ALL RIGHTS RESERVED

# Contents

<b>I</b>	<b>A. Overview</b>	<b>1</b>
<b>1.</b>	<b>Introduction</b>	<b>2</b>
1.1.	BioImage Suite Functionality . . . . .	3
1.2.	BioImage Suite Software Infrastructure . . . . .	4
1.3.	A Brief History . . . . .	4
<b>2.</b>	<b>Background</b>	<b>6</b>
2.1.	Applications of Medical Imaging Analysis: A Brief Overview . . . . .	6
2.2.	Medical Image Processing & Analysis . . . . .	8
2.3.	Software Development Related to Medical Image Analysis . . . . .	9
2.4.	3D Graphics and Volume Rendering . . . . .	11
<b>3.</b>	<b>Starting and Running BioImage Suite</b>	<b>18</b>
3.1.	Installation Overview . . . . .	18
3.2.	Installation Instructions . . . . .	19
3.3.	The Main BioImage Suite Menu . . . . .	22
3.4.	Preferences Editor . . . . .	23
<b>4.</b>	<b>Application Structure</b>	<b>26</b>
4.1.	Application Structure Overview . . . . .	26
4.2.	The File Menu . . . . .	27
4.3.	The Display Menu . . . . .	31
<b>5.</b>	<b>Looking At Images</b>	<b>32</b>
5.1.	Image Formats . . . . .	32
5.2.	The Viewers . . . . .	33
5.3.	The Colormap Editor . . . . .	37
5.4.	Coordinates for NeuroImaging . . . . .	39
5.5.	Atlas Tools . . . . .	44
<b>6.</b>	<b>Advanced Image Visualization</b>	<b>47</b>
6.1.	4D Images . . . . .	47
6.2.	3D Rendering Controls . . . . .	48
6.3.	Volume Rendering . . . . .	48
6.4.	Oblique Slices . . . . .	53
6.5.	The Animation Tool . . . . .	54

---

<b>II</b>	<b>B. Anatomical Image Analysis</b>	<b>59</b>
<b>7.</b>	<b>The Image Processing and Histogram Tools</b>	<b>60</b>
7.1.	Introduction . . . . .	60
7.2.	“Image” and “Results” . . . . .	61
7.3.	Histogram Control . . . . .	62
7.4.	The Image Processing Control . . . . .	64
7.5.	EXAMPLE:Reorientation of Images . . . . .	69
<b>8.</b>	<b>The Interactive Segmentation Tools</b>	<b>71</b>
8.1.	Introduction . . . . .	71
8.2.	The Objectmap Editor Tools . . . . .	73
8.3.	The Surface Editor . . . . .	74
8.4.	Delineating a surface: a step-by-step recipe. . . . .	83
<b>9.</b>	<b>Tissue Classification</b>	<b>84</b>
9.1.	Accessing the Segmentation Tool . . . . .	84
9.2.	Math Morphology . . . . .	85
9.3.	Histogram Segmentation . . . . .	87
9.4.	FSL Brain Extraction Tool . . . . .	88
9.5.	Grey/White Segmentation – using FSL Fast . . . . .	92
9.6.	Bias Field Correction . . . . .	93
9.7.	Appendix: A Little bit of Theory . . . . .	97
<b>10.</b>	<b>Linear Registration</b>	<b>99</b>
10.1.	Accessing the Registration Tools . . . . .	100
10.2.	Registration — Transformation . . . . .	102
10.3.	Manual Registration . . . . .	107
10.4.	Linear Registration (Intensity Based) . . . . .	109
10.5.	Functional Overlay . . . . .	110
10.6.	Image Compare . . . . .	113
10.7.	EXAMPLE: Interactive Registration Tools . . . . .	114
10.8.	Linear Transformations Theory: . . . . .	120
<b>11.</b>	<b>Non Linear Registration</b>	<b>122</b>
11.1.	Introduction . . . . .	122
11.2.	Visualizing NonLinear Transformations . . . . .	122
11.3.	Nonrigid Registration (Intensity Based) . . . . .	124
11.4.	Distortion Correction (Single Axis Distortion) . . . . .	125
11.5.	Batch Mode Registration . . . . .	125
11.6.	Example: Co-register reference 3D brain with individual 3D brain . . . . .	132
11.7.	Checking 3D to Reference non-linear registrations . . . . .	134
11.8.	Remarks . . . . .	135
<b>12.</b>	<b>Landmarks, Surfaces and Point-based Registration</b>	<b>136</b>
12.1.	Introduction . . . . .	136
12.2.	Acquiring Landmarks . . . . .	136
12.3.	The Surface Control and the Surface Objectmap Control . . . . .	140
12.4.	Point-based Registration Tools . . . . .	145

---



---

12.5. Appendix: An Overview of Robust Point Matching . . . . .	147
<b>III C. Functional MRI Analysis</b>	<b>150</b>
<b>13.The Single Subject fMRI Tool</b>	<b>151</b>
13.1. Introduction . . . . .	151
13.2. The fMRI Tool User Interface . . . . .	151
<b>IV D. Multi Subject/Multi Image Analysis</b>	<b>162</b>
<b>14.The Multi-Subject Control</b>	<b>163</b>
14.1. Introduction . . . . .	163
14.2. Setup File Format . . . . .	164
14.3. The Multisubject Tool Graphical User Interface . . . . .	167
14.4. Examples . . . . .	174
14.5. The new SimpleViewer Tool . . . . .	183
<b>15.The Data Tree Manager</b>	<b>185</b>
15.1. Introduction . . . . .	185
15.2. The Tree . . . . .	185
15.3. Space, Anatomical, and Functional Images . . . . .	192
15.4. The Overlay Tab . . . . .	195
15.5. Multiple Image Calculations . . . . .	195
15.6. Functionality for Intracranial Electrode Attributes . . . . .	197
15.7. Options . . . . .	200
<b>V E. Diffusion Weighted Image Analysis</b>	<b>202</b>
<b>16.Diffusion Tensor Image Analysis</b>	<b>203</b>
16.1. Introduction . . . . .	203
16.2. Accessing the Diffusion Tool . . . . .	204
16.3. Tensor Utility . . . . .	204
16.4. Loading diffusion-weighted images (DWI) . . . . .	204
16.5. Specifying gradient directions . . . . .	205
16.6. Loading a mask . . . . .	206
16.7. Computing the tensor . . . . .	207
16.8. Tensor transformations . . . . .	207
<b>17.Diffusion Tensor Analysis</b>	<b>210</b>
17.1. Introduction . . . . .	210
17.2. Loading the diffusion tensor . . . . .	210
17.3. Results . . . . .	211
17.4. Statistics . . . . .	211
17.5. Visualization . . . . .	212
17.6. Transformations . . . . .	215

---

<b>18.Fiber Tracking</b>	<b>216</b>
18.1. Introduction . . . . .	216
18.2. Loading the input images . . . . .	216
18.3. Directionality . . . . .	216
18.4. Tracking . . . . .	217
18.5. Fiber bundles . . . . .	218
18.6. Statistics . . . . .	219
18.7. Results . . . . .	219
18.8. Display . . . . .	219
<b>VI F. Neurosurgery Tools</b>	<b>221</b>
<b>19.Intracranial Electrode Localization</b>	<b>222</b>
19.1. Introduction . . . . .	222
19.2. Getting Started . . . . .	224
19.3. Working with Electrode Grids . . . . .	224
19.4. Electrode Editor Function Catalog . . . . .	227
<b>20.The VVLink Tool</b>	<b>233</b>
20.1. Introduction . . . . .	233
20.2. The BioImage Suite VVLink Interface . . . . .	234
20.3. Connecting and Transferring Images . . . . .	235
20.4. The Data “Tab”, Saving Images and Transformations . . . . .	236
20.5. Real Time Communication . . . . .	236
20.6. Obtaining and Transferring the position of Tools and Landmark points . . . . .	237
<b>21.The Differential SPECT Tool</b>	<b>239</b>
21.1. Introduction . . . . .	239
21.2. Interfacing with the Data Tree Manager . . . . .	239
21.3. Ictal-Interictal Subtraction Analysis by SPM . . . . .	239
21.4. Ictal-Interictal Subtraction Analysis by Bioimage Suite . . . . .	241
21.5. Subtraction Processing . . . . .	243
21.6. Rejoining Blobs . . . . .	245
21.7. Cluster Level Statistics . . . . .	245
21.8. EXAMPLE: Running ISAS and ISAB . . . . .	245
21.9. EXAMPLE: Using the Utilities Tab . . . . .	252
<b>VII G. Cardiovascular Image Analysis</b>	<b>254</b>
<b>22.4D Surface Editor</b>	<b>255</b>
22.1. Introduction . . . . .	255
22.2. Movie Controls (Cine Mode) . . . . .	256
22.3. The “Multi” and “Segment” Tabs . . . . .	256
22.4. Changes in the menu and controls . . . . .	258
<b>23.Estimation of LV Deformation</b>	<b>260</b>

---

<b>24. Angiography Tools</b>	<b>267</b>
24.1. The Vessel Utility Tool . . . . .	268
24.2. The Vessel Tracking Tool . . . . .	271
<b>25. Processing Mouse Hindlimb SPECT/CT Images</b>	<b>273</b>
25.1. Introduction . . . . .	273
25.2. Flip and Crop Images . . . . .	274
25.3. Cropping CT Images . . . . .	275
25.4. Removing the Imaging Table from CT Views . . . . .	277
25.5. Draw Planes on CT Image . . . . .	278
25.6. Segment soft Tissue from CT Images . . . . .	283
25.7. Compute ROI Statistics . . . . .	285
<b>VIII Additional/Miscellaneous Topics</b>	<b>286</b>
<b>26. File Formats</b>	<b>287</b>
26.1. Images . . . . .	287
26.2. Surfaces . . . . .	288
26.3. Landmark Collections . . . . .	293
26.4. Transformations . . . . .	294
26.5. Colormaps . . . . .	295
26.6. Setup Files . . . . .	297
<b>27. Command Line/Batch Mode Tools</b>	<b>298</b>
27.1. Introduction . . . . .	298
27.2. Inspecting Analyze Header Files and Surface Files (.vtk) . . . . .	299
27.3. Image Processing Tasks . . . . .	299
27.4. Segmentation and Bias Field Correction Tools . . . . .	301
27.5. Reslicing Images and other command line tool using transformations . . . . .	303
27.6. Batch Mode Registration . . . . .	304
27.7. Batch Mode Segmentation . . . . .	309
<b>IX Appendices</b>	<b>311</b>
<b>A. Installing Other Software</b>	<b>312</b>
A.1. Installing and Configuring FSL . . . . .	312
A.2. Installing the WFU Pick Atlas . . . . .	313
<b>B. Compiling BioImage Suite</b>	<b>314</b>
B.1. Overview . . . . .	314
B.2. Compiling BioImage Suite . . . . .	315
B.3. Compiling the Prerequisites . . . . .	316
B.4. Miscellaneous . . . . .	321
<b>C. Bioimagesuite FAQ</b>	<b>323</b>
C.1. Working with DICOM data . . . . .	323
C.2. How to install BIS on Debian . . . . .	324

---

---

C.3. Working with TIFF images . . . . .	326
C.4. How can I obtain Bioimagesuite? . . . . .	326
C.5. Is there any documentation for Bioimagesuite? . . . . .	326
C.6. How do I convert matlab format (.mat) into .hdr ? . . . . .	327
<b>References</b>	<b>328</b>

# Part I

## A. Overview

# Chapter 1

## Introduction

BioImage Suite is a collection of image analysis programs which use the same underlying code infrastructure and have the same look and feel but are tuned to specific imaging applications. The current version of BioImage Suite consists of a number of graphical applications (GUI) and a set of command-line utilities providing support for both interactive and batch-mode processing. All software has been tested on Linux, Windows, MAC OS X 10.4 (and at least cursorily run on IRIX

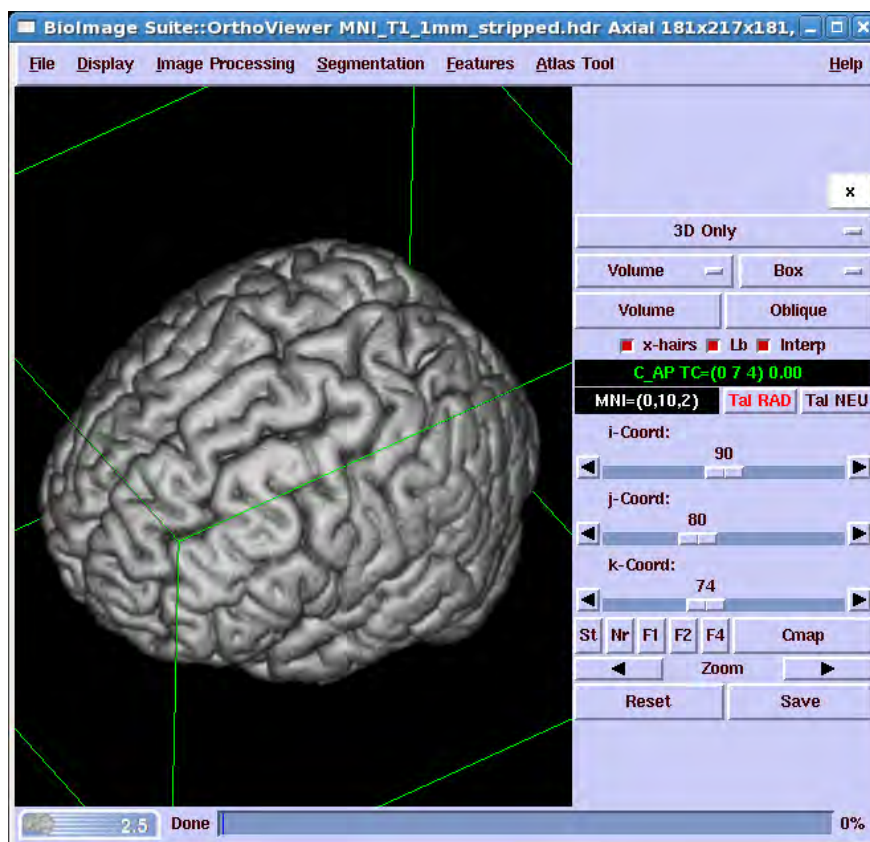


Figure 1.1: An example BioImage Suite Application.I

6.5, Free BSD 6.0 and Solaris 10.0).

This manual describes the basic concepts in BioImage Suite. The following chapters are particularly useful to a first time user:

- The **Starting and Running BioImage Suite** page describes how to start BioImage Suite and how to configure user preferences.
- The **Application Structure (Framework)** page describes the overall structure of all BioImage Suite applications.
- The **Viewers** page describes the BioImage Suite viewers which are at the heart of all applications.

## 1.1 BioImage Suite Functionality

BioImage Suite has facilities for:

**Pre-processing:** Standard image smoothing/filtering, reslicing, cropping reorienting etc. In addition, for bias field correction, it includes a custom reimplementation of the bias field correction method of Styner et al. which incorporates automated histogram fitting for determining the appropriate numbers of classes, and additional spatial constraints.

**Voxel Classification:** Methods for voxel classification are available using simple histogram, single channel MRF and exponential-fit methods.

**Deformable Surface Segmentation:** BioImage Suite has a strong and unique interactive deformable surface editor tool which allows for easy semi-interactive segmentation of different anatomical structures and embeds common snake-like deformable models.

**Registration:** BioImage Suite includes a clean reimplementation of the work of Studholme et al [108]. for rigid/affine registration using a highly efficient conjugate gradient optimization scheme. These methods have been successfully used to align serial MRI data as well as multimodal data (e.g. CT/PET/SPECT to MRI). It also includes a full complement of non-rigid point-based registration methods, intensity-only and integrated feature intensity methods.

**Diffusion Weighted MR Image Analysis:** BioImage Suite includes methods for the computation and visualization of basic voxel-wise measures from diffusion tensor images (e.g. fractional anisotropy) as well as fiber tracking methods using traditional (streamlining) and novel (anisotropic front propagation) methods.

**Cardiac Image Analysis:** The shape-based cardiac deformation method of Papademetris et al. is included in BioImage Suite. This functionality, however, requires the presence of the Abaqus finite element package and license.

**fMRI Activation Detection:** BioImage Suite has a clean and fast reimplementation of the standard General Linear Model (GLM) method for fMRI activation detection, in addition to tools for performing region of interest analysis (ROI), multisubject composite maps, etc. The registration tools (described above) can be used for motion correction, distortion correction and intra-subject registration. (Some of the fMRI tools are not included in the current released version of BioImage Suite because they still use VTK 4.0 and an older set of the common libraries – the rest of BioImage Suite is based on VTK 4.4. We anticipate adding this soon).

## 1.2 BioImage Suite Software Infrastructure

BioImage Suite is developed using a combination of C++ and Tcl in the same fashion as that pioneered by VTK. In practice, most of the underlying computationally expensive algorithms are implemented in C++ (as classes deriving from related VTK or ITK classes) and the user interface is for the most part developed in the Tcl/Tk scripting environment. Further, a custom written C++ wrapper around the Tk graphical user interface library enables the creation of complex graphical components from C++.

## 1.3 A Brief History

BioImage Suite started life as a tool for interactive 4D cardiac segmentation and the original surface editor was presented at the 47th Annual Scientific Session of the American College of Cardiology in 1998. It run exclusively on the Silicon Graphics IRIX platform (6.2,6.3) and used a combination of MOTIF and Open Inventor.

It was subsequently adapted and extended for neuroimaging applications primarily for the needs of an epilepsy image-guided neurosurgery project (2001-). At this point development switched to an explicit multi-platform setup and MOTIF was replaced by a Tcl/Tk environment and Open Inventor was replaced by VTK. (then version 3.1).

Progressively, diffusion weighted imaging (DTI) functionality was added (2002 + M. Jackowski), as well as the development of our fMRI tools (2003 + N. Rajeevan) It was subsequently extended for use in abdominal fat quantification work (2004-) and for use in vascular tree extraction for a mouse hindlimb angiogenesis project. (2005-).

Recently, we have obtained funding from the NIH/NIBIB (R01 EB006494-01 PI: Papademetris, X.) to continue, in the words of the program announcement, “to support the continued development, maintenance, testing and evaluation of existing software”. The BioImage Suite webpage went live in early 2006 and a support forum was established soon afterwards. A first beta version was made publicly available in January 2006. We are (July 2006) in the process of releasing BioImage Suite 2.0 – version 1 was never publicly available but in use at Yale since 2002.

BioImage Suite was originally developed for the needs of the following, NIH-funded, projects at Yale:



- Bioimaging and Intervention in Neocortical Epilepsy – BRP R01-EB000473 PI: Duncan, J. S.
- Dynamic Analysis of LV Deformation from 4D Images –R01-EB002068 PI: Duncan, J. S.
- Integrated Function/Structure Image Analysis in Autism – R01-NS035193 PI: Duncan, J. S.
- Brain Image Segmentation, Comparison, and Measurement – R01-EB000311 PI: Staib, L. H.
- Functional MRI for Neurosurgical Planning in Epilepsy – R01-NS38467 PI: Constable, R.T.
- Functional Heterogeneity of the Hippocampal Formation – R01-NS38467 PI: Constable R.T
- Non-invasive Methods for Imaging Angiogenesis – R01-HL65662 PI: Sinusas, A. J.
- Yale Mouse Metabolic Phenotyping Center – U24-DK59635-01 PI: Shulman, G.I. (Imaging Core PI: Behar, K.L.)

If you use BioImage Suite for a publication please cite it as:

X. Papademetris, M. Jackowski, N. Rajeevan, R.T. Constable, and L.H Staib. BioImage Suite: An integrated medical image analysis suite, Section of Bioimaging Sciences, Dept. of Diagnostic Radiology, Yale School of Medicine. <http://www.bioimagesuite.org>.

## Chapter 2

# Background

### 2.1 Applications of Medical Imaging Analysis: A Brief Overview

BioImage Suite can be applied to help analyze images in a large number of domains. We briefly review some of the key application areas below.

**Structural Neuroimaging:** Measurement of brain structure is important for characterizing differences associated with normal variation (sex, handedness), development and aging, and pathology. Anatomic magnetic resonance images (MRI) are used to measure cortical and subcortical structure. Key problems in neuroimaging include the measurement of gray and white matter volume in the brain as well as segmentation and quantification of individual structures such as the hippocampus or caudate nucleus. Structural differences between groups of images (e.g. autism patients vs controls) can be quantified using non-rigid deformation based techniques (e.g. deformation based morphometry).

**Diffusion Tensor Magnetic Resonance Imaging:** Diffusion tensor MR imaging (DTI), allows the quantification of oriented tissue in terms of its diffusion properties as well as with fiber tracking. While typically applied to white matter in the central nervous system, DTI is also used for characterizing muscle (cardiac, skeletal) as well as other structures. For example, the structural changes associated with cancer can be measured with DTI. The key processing steps are diffusion tensor estimation at each location, diffusion property calculation and fiber tracking. In addition, multisubject registration and segmentation may also be needed for quantification.

**Functional Neuroimaging:** Functional magnetic resonance imaging (fMRI) measures blood flow changes associated with brain function. It has been widely used since its development in 1990 for localizing functional regions of the brain, characterizing normal variation and understanding pathological differences. fMRI is also used clinically as an aid for neurosurgical planning. fMRI activation detection and comparison involves a variety of processing steps including registration for motion and distortion correction and the formation of group composite functional maps for the analysis and comparison of subject groups. Cortical Blood flow measurements when combined with more conventional blood oxygenation dependent contrast provide a more quantitative measure of

brain activity through calculation of cerebral metabolic rate of oxygen consumption ( $CMRO_2$ ). Positron emission tomography (PET) requires many of these same processing steps, in addition to methods for the quantification of blood perfusion and receptor kinetics [15]. The latter requires extensive processing.

**Cardiac Deformation Analysis:** The quantification of cardiac deformation patterns from sequences of 3D MR and 3D echocardiographic images is a key step toward both understanding normal heart function and for evaluating the effects of acute coronary occlusions and post cardiac insult remodeling. The key image analysis methods here are segmentation and non-rigid deformation estimation.

**Angiography:** The detection and quantification of different parts of the vasculature has important applications in the understanding of many diseases such as cancer and diabetes.

**Image Guided Neurosurgery:** Image guided neurosurgery can leverage virtually the entire spectrum of medical imaging and image analysis techniques including segmentation, registration, fMRI and MRS analysis etc. In epilepsy neurosurgery, a particular focus application, localization of intra-cranial electrodes used to detect seizure sites is an important capability.

**Image Guided Prostate Radiotherapy:** The key image analysis problem in prostate radiotherapy is the localization of the prostate and nearby soft tissue structures such as the bladder and rectum via the registration of pre-therapy CT images and intra-therapy CT acquisitions. Such registrations may also require the use of articulated transformation models to handle the movement of different bony structures in the hip and pelvic regions. The registration can then be used to update the radiation plan (which is based on pre-therapy images).

**Abdominal Fat Quantification:** The quantification of intra-abdominal visceral fat is becoming a key step in the understanding of the relationship between obesity, insulin resistance and the development of Type II diabetes. These methods currently require image segmentation (in terms of voxel classification) as well as the use of magnetic resonance spectroscopic methods (MRS) for the quantification of intra-organ and intra-muscle fat which is dispersed in the tissue and not easily detected using magnetic resonance imaging methods.

**Metabolic Imaging:** Magnetic resonance spectroscopy (MRS) is used to quantify biochemicals (metabolites) *in vivo* by measuring the magnetic resonance chemical shift spectrum and quantifying the characteristic peaks corresponding to chemicals of interest. (a.) *Proton MRS* can be used to measure the concentrations of cerebral metabolites, such as NAA (N-acetyl aspartate) and creatine, and with improved MR methodology also glutamate, glutamine and GABA ( $\gamma$ -aminobutyric acid). Proton MRS is also applicable in the rest of the body for water/fat imaging which is important for the study of diabetes [122] (b.) *POCE (Proton Observed Carbon Edited) MRS* can be used to look at metabolic fluxes, like the tricarboxylic acid (TCA) cycle flux. (c.)  $^{31}P$  *CSI* is a key technique in understanding energy metabolism in a wide range of disorders, including epilepsy. MRS is typically performed in a small number of voxels. Our goal is to implement efficient reconstruction methods that will make true spectroscopic chemical shift imaging (CSI) a practical modality. In addition, accurate segmentation (e.g. gray/white matter in the brain [25]) is essential for quantifying metabolite concentrations and registration methods are required to map the metabolic images to the underlying anatomy.

## 2.2 Medical Image Processing & Analysis

A large number of image processing and analysis methods has been developed over the last 20-30 years (a good but slightly dated critical review of the field can be found in Duncan & Ayache [29].) The following is a categorization of most of the key tasks in Medical Image Analysis. BioImage Suite aims to be a comprehensive image analysis software platform and does/will include representative algorithms in *all* of these categories.

**Pre-processing:** These initial image processing algorithms are often used in preparation for subsequent image analysis algorithms. They may also be used in their own right for the purpose of visualization. A good example is skull stripping [102] in neuroimaging which allows for the direct display of the cortex using volume rendering techniques. This visualization can be used for neurosurgical planning applications. Another example of this type of processing is MR bias inhomogeneity correction [100, 8, 115, 111, 43] for both brain and abdominal images which enables accurate tissue classification. We have developed innovative methods for bias field correction involving direct measurement of B1-fields [120].

**Voxel Classification:** These techniques label each voxel in the image according to tissue classes. Typical applications of these methods are the classification of brain tissue into gray matter, white matter and cerebrospinal fluid (CSF) [124, 129, 95, 85, 128, 30] in neuroimaging and fat/lean muscle classification in abdominal imaging [80].

**Deformable Surface Segmentation:** Deformable surfaces either using explicit parameterizations [55, 103] or levelset formulations [88, 94, 58, 16, 127] are applicable to all parts of the body as a method for measuring the shape and volume of different organs and sub-cortical brain structures [127]. In addition to automated methods, there are also applications of manual tracing and semi-automated methods in many practical applications both for complete segmentation or for the review/correction by an expert user of the results of automated segmentation such as in the BioImage Suite surface editor tool [79, 73, 12].

**Rigid and Articulated Registration:** Rigid registration allows for the alignment of different images of the same patient in a common coordinate space [10, 89, 109, 126], that subsequently permits combined analysis and visualization of the data. Such methods can be used both for intra-modality registration (e.g. serial MR, serial CT) or for multimodality registration (e.g. MR/CT, MR/PET etc.) An extension of these methods is our recent articulated piecewise rigid work for mapping serial lower limb mouse CT images [75] while accounting for differences in joint angles.

**Non-rigid Registration:** Non-rigid registration methods generate transformations that allow the mapping of images of different patients to a common coordinate space [17, 35, 113, 91, 114, 20, 24, 22]. In addition, such methods are also used to map post-surgical images to pre-surgical images of the same patient [110]. Once images from different patients are mapped into a common coordinate space, they can be used to generate voxel-wise group statistics for the study of within-group variability (both structure and function) as well as the quantification of statistical between-group differences (e.g. normal controls vs patients).

**Diffusion Tensor MR Imaging (DTI):** This novel imaging technique allows for the quantification and characterization of local brain tissue diffusion properties using a tensor model [9] and

the construction of potential white matter axonal pathways connecting different brain regions (e.g. [54, 67, 60, 50, 51]).

**Vessel Extraction and Tracking:** The localization and quantification of vascular structure has important applications in the evaluation of peripheral and coronary artery disease, as well, as in the study of vascular growth and remodeling [31].

**Cardiac Strain Analysis:** The quantification of cardiac motion and strain is crucial for the understanding of cardiac function, injury and remodeling [23, 63]. Image analysis in conjunction with biomechanical modeling [42, 45] allows for the measurement of regional strain from cardiac cine MRI [5, 97] and echocardiography[82].

**fMRI Processing:** Functional magnetic resonance imaging analysis involves a series of processing steps including: (i) motion correction of the timeseries raw T2\*-weighted images (e.g. [40, 39, 36, 66]), (ii) smoothing and denoising (iii) detection of activated regions [37, 84, 38, 65, 121, 99, 69, 21] and (iv) distortion correction for mapping the functional images to anatomical MRI images [53, 106].

**Magnetic Resonance Spectroscopy and Spectroscopic Image Analysis:** Typical MRS spectra are quantified by first approximating the spectrum with a linear combination of *in vitro* spectra (after shifting and broadening) measured with the exact same sequence: this is known as the LC-model [87]. Typical implementations of the LC-model, such as the one by Dr. Robin de Graaf of the Yale MRRC, require approximately 10 minutes for each voxel. While the computational time is feasible if each study consists of only a handful of voxels, the problem becomes intractable for larger studies such as a technically feasible CSI acquisition consisting of  $32 \times 32 \times 4 = 4096$  voxels. Current acquisitions are commonly restricted to low resolution measurements [18]. While simpler techniques can be used the full power of CSI will not be realized until the same proven procedures that have been applied at the voxel level become available at the image level.

**Compartmental Modeling: PET and MR Perfusion** The dynamic images that PET provides can be used to model the PET tracer kinetics to produce quantitative estimates of physiological parameters. Estimation of these parameters, including blood flow, metabolic rates and receptor concentrations, requires weighted nonlinear least squares fits of the time-activity curve of each voxel to the appropriate compartmental model equations. These same techniques can be adapted for dynamic contrast enhanced magnetic resonance images (DCE-MRI) and MR perfusion imaging.

## 2.3 Software Development Related to Medical Image Analysis

**Please Note:** This section represents work in progress and is by no means complete.

In this section we review related work in software grouped into two categories (i) development libraries for numerical computation and visualization, as well as, libraries specific to image analysis and (ii) complete software packages (both commercial and free) that are currently available to support biomedical research.

**a. General Development Libraries:** Medical image analysis in general, and medical image

analysis software in particular, rely on many core algorithmic techniques from a number of fields such as numerical analysis, graphical user interface design and computer graphics and visualization. Key examples include:

*Numerical Processing:* Much of the available code for numerical processing was originally developed in Fortran 77 often converted to C using the freely available *f2c* converter. A good example is the LAPACK linear algebra library [32] which is also available for C as CLAPACK. Other freely available libraries include the GNU Scientific Library [33], FITPACK (a spline fitting toolbox [26]), and the vnl:numerics library.

*Computer Graphics and Visualization:* The cornerstone of most scientific 3D Graphics and Visualization is the Open GL library [96, 70] originally developed by Silicon Graphics. Open GL is a low level library and is often used within a higher level toolkit such as the Open Inventor toolkit [125], or the Visualization Toolkit (VTK) [92]. VTK has gained substantial popularity for medical image analysis as it also provides common image/surface processing algorithms, and functionality for some numerical processing in addition to a complete set of graphics routines. BioImage Suite relies on VTK for most of its visualization tasks.

*Graphical User Interfaces:* There is a large number of possibilities for graphical user interface design, although the list shrinks dramatically in the case of cross-platform (i.e. Windows/Unix/Mac OS) support. Possible choices include WxWindows, FLTK, and the TK toolkit which comes together with the Tcl scripting language. BioImage Suite uses the Tcl/Tk combination together with a custom C++ wrapper around Tcl to enable the creation of user interfaces in both a scripting language (Tcl) and C++ (as needed for complex components).

*Medical Image Analysis Libraries* The key development in this area over the last 3-4 years has been the development of the NIH/NLM Insight Toolkit (ITK). ITK focuses exclusively on medical image segmentation and registration, and provides implementation of many commonly used algorithms. BioImage Suite uses some aspects of ITK (e.g. the levelset libraries) as a basis for implementation of some methods and will make accessible from the user interface some of the algorithms implemented in ITK that complement what is currently available.

An alternative is the Matlab programming environment (MathWorks, Inc. [62]) which provides a programming language, a large number of numerical methods as well as basic graphical interface capabilities. Some complex software packages are developed entirely in Matlab (e.g. SPM). However, while Matlab is an excellent prototyping tool, algorithms that do not naturally fall into a matrix manipulation paradigm can be highly inefficient.

**b. Complete Software Packages** There are a number of software packages available for medical image analysis that provide some overlapping capabilities; some are commercial but most are research-based. While a number of these packages offer strong features in specific areas, there is currently no software package that provides state-of-the-art methods in the array of techniques that are included in BioImage Suite.

The neuroimaging packages available are probably the most well developed. The Slicer [118] package has a similar design to BioImage Suite (built upon VTK/Tcl) and provides many valuable methods.

Slicer focuses on registration and segmentation techniques geared for surgical planning, particularly in the brain. Brainsuite [2] focuses primarily on segmentation and visualization of anatomical brain MR images. SPM [104] is a Matlab package for fMRI and PET activation analysis. FSL [41] also focuses on image analysis and statistical tools for brain imaging, including fMRI and DTI. AFNI [3] was originally specific to fMRI image analysis, although has been recently expanded for other tasks such as DTI analysis. Freesurfer [112] is a set of command-line neuroimaging utilities for the semi-automated reconstruction of the cortical surface and overlay of functional data onto the reconstructed surface. Map3D and BioPSE [11] are packages primarily for bioelectric field modeling program but they have many useful visualization and analysis methods including FEM analysis, DTI analysis and basic image processing capabilities. AIR [4] is limited to image registration. NIHImage/Scion [47] provides primarily basic image processing functions.

Some commercial packages are available, both general purpose and application focused. Commercial packages can be expensive, may be tied to a specific niche market and their development cycles are often too slow for researchers to influence. They also typically do not have the benefits of open source. MEdX [64] is software for visualization, processing, and analysis. It primarily serves as an interface to functionality integrated from AIR, SPM and FSL. VIDA [117] is software primarily for 3D visualization with an emphasis on cardiac applications including image segmentation and cardiac mechanics. The general purpose commercial software package Analyze [7] has strengths in visualization, segmentation, registration and measurement. BrainVoyager [119] is, like SPM, specific to brain activation analysis.

## 2.4 3D Graphics and Volume Rendering

### 2.4.1 3D Graphics

Most researchers' first experience with image display consists of simply displaying a two-dimensional image slice (most likely in Matlab). While there are some issues, such as the relative orientation (row/column vs column/row) and the position of the origin, it is for the most part a fairly straightforward and intuitive procedure. The same applies to rendering curves and point landmarks on the image. The key behind this simplicity is the fact that, unsurprisingly, two-dimensional computer monitors are well suited to displaying two-dimensional content!

The move to 3D graphics and visualization requires, in some way, getting around the fact that for the most part our display units are two-dimensional. The most common "illusion" used in 3D graphics – and the one that is employed by the Visualization Toolkit (VTK) and hence by BioImage Suite – is that the program first generates a 3D world/scene consisting of three-dimensional entities/models – actors/props in VTK parlance – which have various appearance properties. The world is illuminated by a set of lights and the "user" looks at the world through the eyes of a virtual camera – See Figure 2.1 for an example. The image that gets displayed on the computer monitor is precisely the output of such a virtual camera. To recap, a world consists of (i) Actors, (ii) Lights and (iii) a camera. The exact output naturally depends critically, in addition to the Actors themselves, on the position and orientation of the camera. If the camera is looking in the wrong direction, even if all the actors have been generated by your program, it will not show anything on your computer monitor. Just as in the real world, when you have to take a picture of a scene, you point the camera

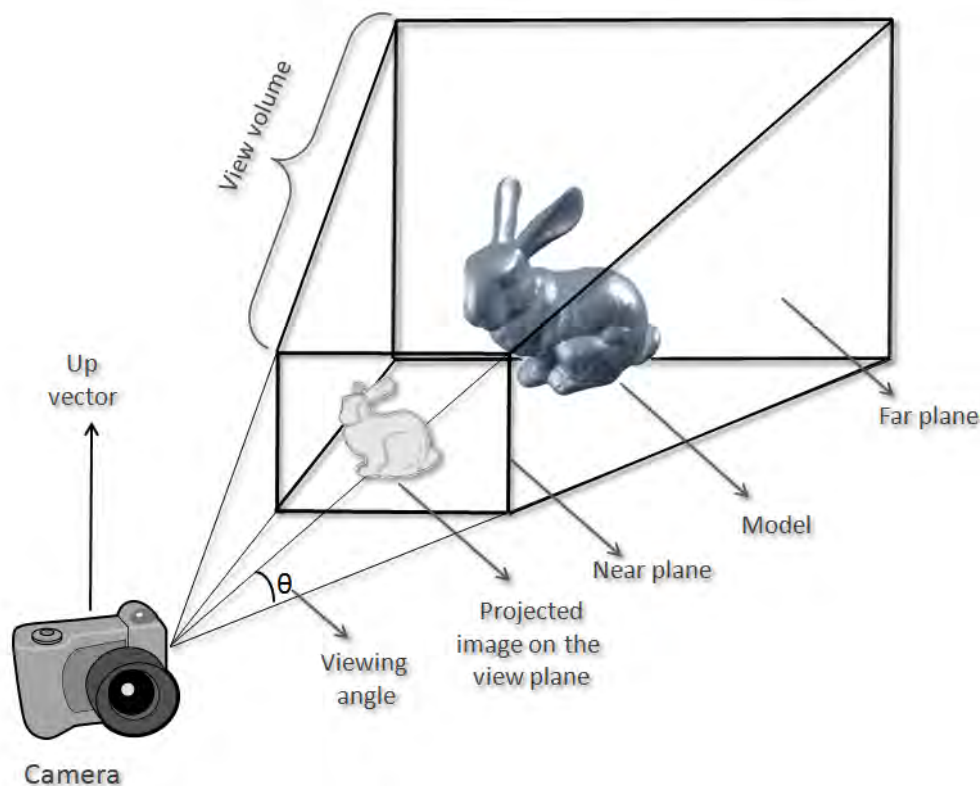


Figure 2.1: The Camera Model for 3D Graphics. A camera looks at the synthetic world (e.g. the bunny) and takes a picture of it – this is what is displayed on the screen/view plane. An additional advantage of this virtual world over the real world is that we can restrict the viewing volume to exclude objects that are “too close” or “too far” – hence visually cropping the scene.

towards that scene, in this 3D-world the camera has to be position correctly located (far enough from the object) and directed towards the scene.

Figure 2.1 shows a schematic of a camera and its properties. The frustum defined by the near and far planes forms the “view volume. ” Everything inside the view volume is shown on the screen whereas everything outside the view volume is “clipped” out from the scene and is not shown on the screen. The projection of the scene/actors on the view plane is shown on the screen. The view plane can be the same as the near plane.

The up-vector defines the “up” direction for the camera. Based on the up-vector, the image shown on the screen will be shown. Figure 2.2 shows a schematic of the result of having a up vector and an inverted up-vector. The inverted up-vector causes the projected image to be inverted. As can be seen, the camera in 3D graphics is very similar to a camera in the real world.

All display operations in BioImage Suite are done using 3D Rendering, even when a single image slice is involved. This allows, for example, for fast zooming in and out as this operation simply involves changing the zoom on the “virtual” camera as opposed to redrawing the image bitmap scaled. This type of renderer also allows the seamless mixing of images and polygonal data (e.g. surfaces, curves) in the display windows.





Figure 2.2: The Up-vector for the Camera. The left image shows the result of the camera vector pointing upwards. The right image shows the result of the up-vector pointing downwards. The projected image on the view plane is inverted.

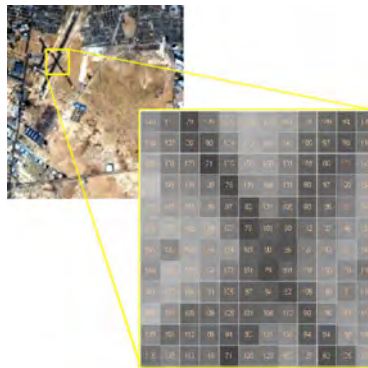


Figure 2.3: An example of an image (top) and how a computer looks at it (bottom).

## 2.4.2 Displaying 3D Images

Most people begin to think of images as essentially a matrix of numbers – see for example Figure 2.3. The value of these numbers may be arbitrary (e.g. anatomical MR acquisitions) or have physical meaning (e.g. a fractional anisotropy map). To display these image we somehow must convert each number to a color. The only complication, at this stage, is the relationship between the row/column indices of the matrix and the  $i$ - and  $j$ -axes of the image, respectively.<sup>1</sup>

**Notation:** We will use  $i, j, k$  to refer to the internal axis of the image in voxels and  $x, y, z$  to be the coordinates in mm.

This type of radical simplicity disappears, unfortunately, once we have issues with mapping images. Examples include, image registration applications when we are looking to estimate a transformation between two images, statistical shape model building etc. In general, in medical imaging, we need to keep track not only the image intensities but additional attributes such as voxel dimensions (voxels need not be isotropic), image orientation (the relationship of the  $i$ -,  $j$ - and  $k$ -axis orientation to the human body – see Figure 2.4), the position of the voxels etc.

<sup>1</sup>Often a problem in MATLAB, when often image matrices need to be transposed before display!

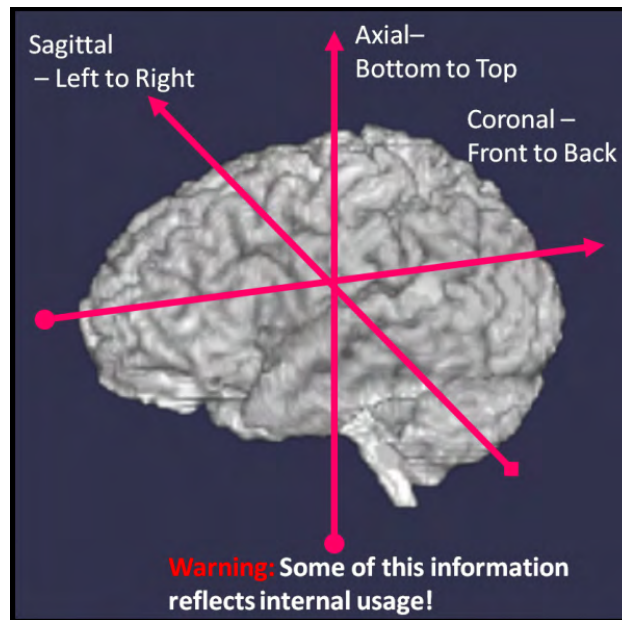


Figure 2.4: Standard Image Acquisition Orientations, Axial or Transverse, Coronal and Sagittal. The arrows indicate the z-axis direction (which may be inverted depending on the acquisition protocol), the x-axis and the y-axis are perpendicular to this. **Right:** Axial, coronal and sagittal slices. Oblique acquisitions are also sometimes used, in which the z-axis is rotated, or *obliqued* away from one of the standard acquisitions, e.g. coronal oblique.

### 2.4.3 Volume Rendering

Volume rendering is defined as the process of generating a 2D image **directly** from three-dimensional data. The basic idea is to simulate the absorption and emission of light as it goes through matter.

To simulate the passage of light through data, rays are cast from the image plane into the space containing the volume, as shown in Figure 2.5. For every pixel in the image plane, a ray is cast into the volume space that traverses the volume. At each point along the ray, the data is sampled to identify the density/intensity at that point. This value is different for different materials such as bone, tissue, fat and so on. Based on the intensity at that point along the ray, the color and opacity is identified using a lookup table called a *colormap* or *transfer function*.

### 2.4.4 Types of compositing functions

Color and opacity are accumulated along the ray and various compositing operators are used to get different effects. An *X-ray* image can be simulated by averaging the intensity values sampled along the ray. A *MIP* (Maximum intensity projection) image is obtained by preserving the maximum intensity value along the ray. MIP images are frequently used to visualize vascular structures as they can be clearly seen using this technique. Figure 2.6 shows an example of the two techniques.

The compositing technique used for direct volume rendering is based on accumulating the color

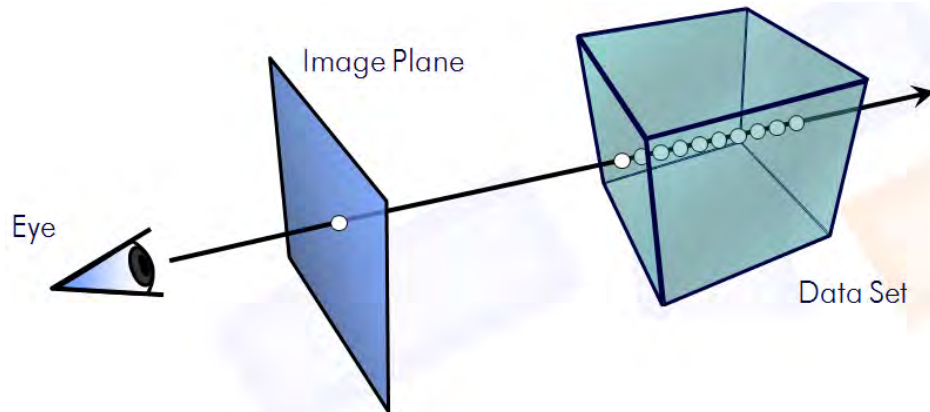


Figure 2.5: This image depicts a schematic of the volume rendering process. For every pixel in the image plane, a ray is cast into the volume space. Along the ray, the volume rendering integral is evaluated to simulate the passage of light through matter. At each position along the ray, the volume/data is sampled to identify the value at that location. The location identifies the material properties such as absorption and emission. A color and opacity is obtained for every density by performing a lookup into a table, called the *colormap* or *transfer function*. This allows users to color different regions of the volume differently.

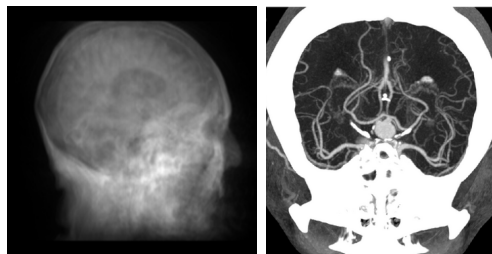


Figure 2.6: The left image shows an X-ray style image that can be generated by average the intensity values along the ray. The right image shows a MIP image that can be obtained by picking the maximum intensity observed along that ray. Such MIP images are widely used to visualize vessels.

based on the color and opacity of the current voxel under consideration. For example, during ray traversal if in a CT scan, bone is encountered, it absorbs more energy than tissue or a thin vessel.

### 2.4.5 Types of Volume Rendering

There are four types of volume rendering techniques: raycasting, texture mapping-based, shear warp and splatting. Raycasting and texture mapping-based techniques are the most widely used of the four. *Raycasting* is the technique in which an image is generated by casting rays into the volume space.

As graphics hardware got better with time, researchers devised a way to perform volume rendering using graphics hardware. This technique loads the data into graphics memory and utilizes 2D (and more recently 3D) texture maps to perform interpolation and blending. Since graphics hardware



Figure 2.7: This figure shows a schematic of the axis-aligned polygons on the left, that are texture-mapped and blended to get a volume rendered image of the data.

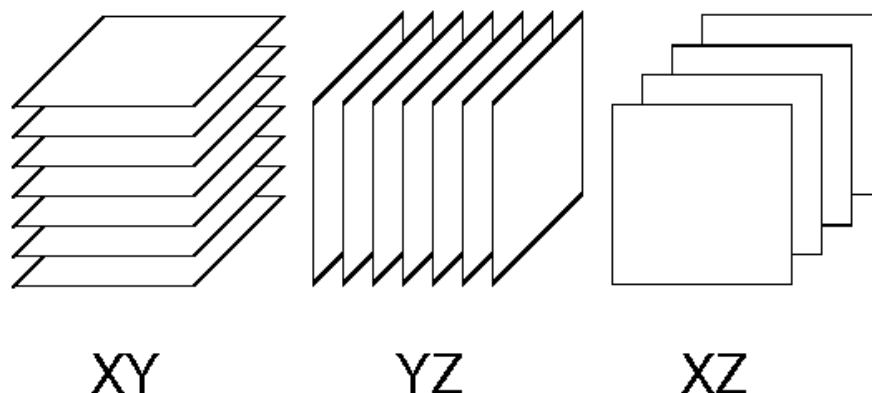


Figure 2.8: Three axis-aligned copies of the data have to be stored in memory to allow 2D texture mapped volume rendering. The XY-aligned slices are used when the view vector is aligned to the Z-axis, the YZ-aligned slices are used when the view vector is aligned to the X-axis and similarly, the XZ-aligned slices are used for the Y-axis aligned view vector.

is extremely fast, this technique is much faster than the raycasting technique described earlier.

### 2D texture mapping based volume rendering

2D texture mapping based volume rendering uses the ability of graphics hardware for fast rendering of polygons and textures. Figure 2.7 shows a schematic of the process. The leftmost image shows polygons that are aligned to an axis. These polygons, which are collectively known as the proxy geometry, are texture-mapped based on the values in the data. A blending step is performed that blends the result of each texture-mapped polygon in a back-to-front manner to get a volume rendered image, as shown in the rightmost image in Figure 2.7.

There are a few disadvantages with using such 2D-textures though. For this type of volume rendering, three copies of the data have to be stored in axis-aligned slices (one for each X, Y, Z-axes), as shown in Figure 2.8. Based on the view vector of the camera, the sets of data that is most perpendicular to the view vector is selected. The biggest drawback of using three sets of axis-aligned 2D textures is that as the viewer rotates the camera around the data, there is a visual popping artifact, when the algorithm switches from one set of axis-aligned slices to another.

### 3D texture mapping based volume rendering

The advent of 3D textures helped solve the problems of maintaining 3 copies of the same data in memory as well as the visual popping artifacts. The data can now be stored in a single 3D texture and instead of using axis-aligned slices, a new algorithm was developed to draw “view-aligned” slices, as shown in Figure 2.9. View-aligned slices ensured that slices were drawn orthogonal to the view vector at all times and ensured that there would be no visual artifacts.

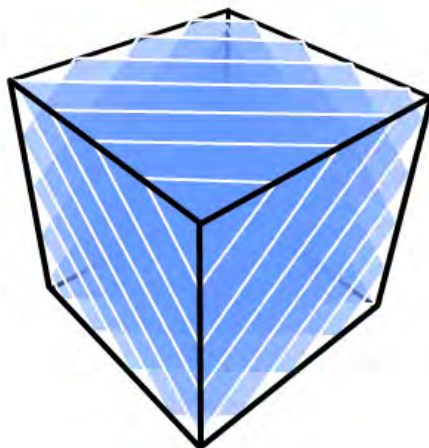


Figure 2.9: View aligned slices eliminate the need for storing three sets of data. The view aligned slices are interactively drawn perpendicular to the view vector, as the user rotates the camera around the data.

Figure 2.10 shows a schematic of the 3D texture mapping-based volume rendering approach. The render geometry consists of polygons that are drawn perpendicular to the view vector as the viewer rotates around the data. The rendering of proxy geometry, texture mapping the data from the 3D texture onto it and the blending, all happen at run time and provide a volume rendered image of the data to the viewer.

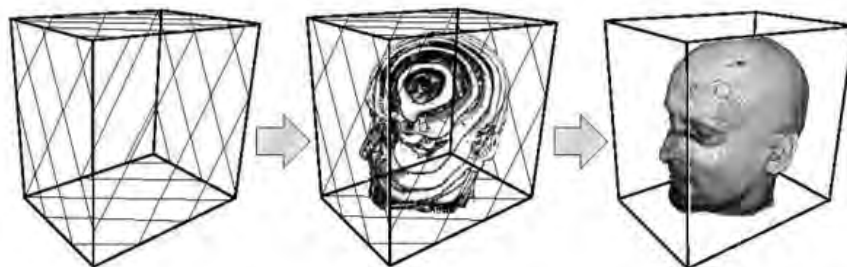


Figure 2.10: A schematic of the 3D textures based volume rendering process. The view aligned slices are textured using the 3D textures and blended to provide a volume rendered representation of the 3D data.

## Chapter 3

# Starting and Running BioImage Suite

### 3.1 Installation Overview

Installing BioImage Suite is a relatively simple process. On Microsoft Windows, we provide a self-contained installer file that completely automates the procedure. On UNIX derivatives (this includes Linux and the Apple Macintosh Operating System) the procedure involves uncompressing four files in the `/usr/local` directory or another location of your choice (in which some script files may need to be edited.)

**Overview: What is included:** All files required for BioImage Suite are available from the BioImage Suite webpage – [www.bioimagesuite.org](http://www.bioimagesuite.org). Special effort is made to package almost all required packages/libraries with the system <sup>1</sup>

BioImage Suite consists of four parts (which are collapsed into one file in the case of the Windows Installer):

1. A pre-compiled binary `itk241_yale` distribution containing the Insight Toolkit (ITK) v 2.4.1.
2. A combo binary `vtk44_yale` distribution which contains the Tcl/Tk scripting language, various Tcl extensions and the Visualization Toolkit. This includes binary versions of:
  - (a) The Tcl/Tk scripting language version 8.4.11
  - (b) The following Tcl/Tk Extensions:
    - Incr Tcl 3.2.1
    - tcllib 1.8.1
    - IWidgets 4.0.1
  - (c) The CLapack numerical library v3.

---

<sup>1</sup>The one exception is that the .NET framework needs to be separately installed on older versions of Windows (2000 and earlier).



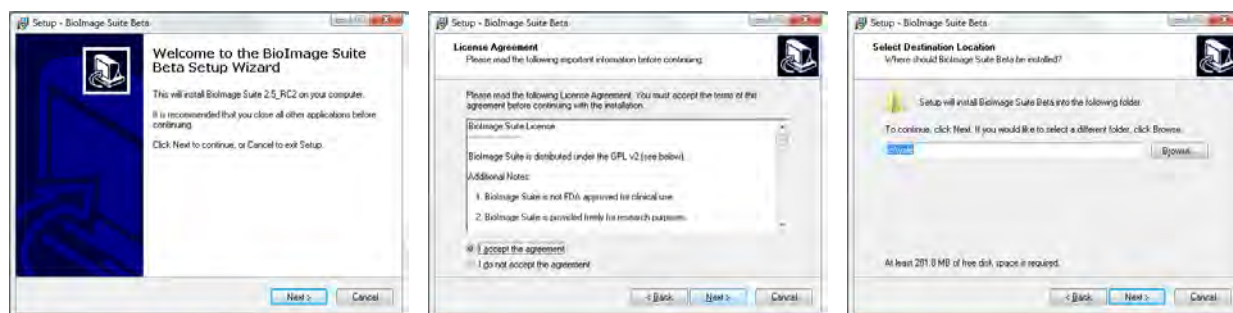


Figure 3.1: The Windows Installer in Action. The default location is `C:/yale`. Please avoid locations with spaces in the directory names – e.g. `C:/Program Files` !

- (d) A slightly patched version of the Visualization Toolkit with TCL Wrapping (v 4.4.2)
3. A pre-compiled `bioimagesuite_extra` distribution<sup>2</sup> consisting of the MINC 2.0.11 distribution and Xercesc 5.7.
4. Installation of the BioImage Suite software package itself.

## 3.2 Installation Instructions

For BioImage Suite 2.6, we have two types of installers: Complete and Update. The “Complete” installer is a self-contained installer that installs BioImage Suite and all the required software in the specified directory. The “Update” installer assumes that you have previously installed BioImage Suite 2.5 and updates the files in the “bioimagesuite” directory. NOTE: If upgrading, it is crucial that you install BioImage Suite in the same directory where you have previously installed BioImage Suite 2.5. In the case on Windows, the directory will most like be `C:\yale` and in the case of Linux/Mac, the directory will mostly like be `\usr\local`. Please confirm the existence of BioImage Suite in these directories before updating it.

### 3.2.1 Microsoft Windows

For Windows, there are two distributions: “vs2003” and “vs2005”. The “vs2003” version is compiled for users who use an older version of Windows such as Windows 2000 etc. The “vs2005” version is compiled for users using Windows Vista/XP. Once you have selected the distribution and the kind of installers (Complete/Update), simple download and execute the all-in-one installer to install BioImage Suite. – this is shown in Figure 3.1. The installer will ask a couple of questions and then perform the installation. The default and recommended installation directory is `c:/yale`. Please avoid installing BioImage Suite or storing data in directories containing spaces in their names – including unfortunately defaults such as “Program Files” and “Documents and Settings” – mercifully this later convention is eliminated in Microsoft VISTA.

<sup>2</sup>Which despite the name is required!

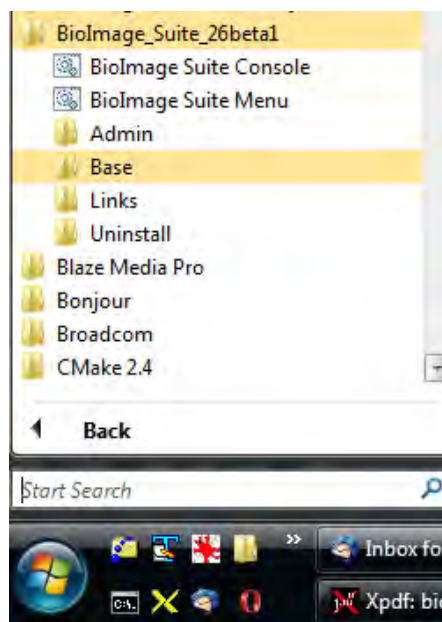


Figure 3.2: The BioImage Suite entry in the “Start” menu in Windows Vista. Identify the BioImage Suite entry in the menu and click on “BioImage Suite Console” or “BioImage Suite Menu” to start BioImage Suite.

Once the installation is complete an entry in the Start Menu appears (see Figure 3.2). The key entries are the `BioImage Suite Menu` which launches the main application, as in Figure 3.3 and the `BioImage Suite Console` which is a special command prompt that allows for the running of commandline applications etc.

### 3.2.2 UNIX: Linux and Mac OS X 10.4

The installation for both of these systems is now as simple as running a single “.sh” file.

**Downloading the appropriate installer** The user should identify the type of installer that the user wishes to install. For example, if the user prefers to download a “complete” installer compiled using “g++-4.0” for “Linux” on a 32-bit machine, the user would download the following file:

```
bioimagesuite-26_beta1_15_Jul_2008-Linux-g++-4.0-i386-complete.sh
```

. In case, the user were downloading an “update” installer compiled with “g++34” for a 64-bit “Linux” machine, the file to be downloaded would be



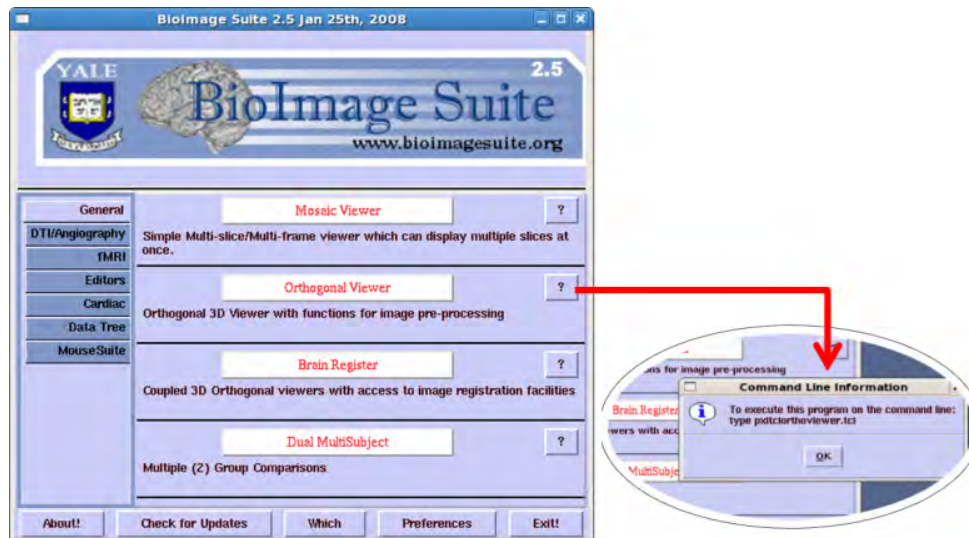


Figure 3.3: The BioImage Suite Start Menu entry. Pressing the “?” button next to each application shows how to directly invoke it on the commandline.

```
bioimagesuite-26_beta1_15_Jul_2008-Linux-g++34-x86_64.sh
```

**Installing** To install BioImage Suite, type

- `sh ./bioimagesuite_[filename].sh`  
where [filename] should be replaced with the kind of file that you downloaded as described above. On Mac, you may have to type  
`sudo sh ./bioimagesuite_[filename].sh --prefix=/usr/local`
- Answer “Yes” to the question asked.
- On Linux, Menu Entries for GNOME and KDE can be created by typing `bioimagesuite26/createmenuentries.sh`.

To start BioImage Suite (on Linux/Mac), type

```
path/bioimagesuite26/start_bioimagesuite
```

NOTE: Here “path” should be replaced with `/usr/local` or the specific path that you may have specified during installation.

Alternatively, you could start the BioImage Suite console by typing:

path/bioimagesuite26/bis\_console

or use the Menu Icons in Applications/Other (which are created if `createmenuentries.sh` has been executed as specified in the “Installing” section above).

**Note:** BioImage Suite, unlike many other software packages, uses accelerated 3D Graphics. This has two implications: (i) The quality of the graphics card is important – ideally you will need a decent video card (much like for games!) from somebody with like an NVIDIA chipset (recommended) or ATI – with properly configured drivers (on Linux you can use the ‘glxgears’ program to test this. (ii) Remote display of BioImage Suite, i.e. running on one machine and using the X-window protocol to display on another, will result in a significant performance hit – especially if volume rendering is involved. A BioImage Suite installation needs about 300 MB of disk space.

### 3.3 The Main BioImage Suite Menu

There is no such thing as the BioImage Suite application program. Rather, BioImage Suite consists of a number of different utilities/applications which share a number of common controls/elements. Depending on the task at hand, one needs to select the appropriate application. The following snapshots are all taken on a Windows Vista computer, however the appearance on other platforms is very similar.

The BioImage Suite main menu application, which is shown in Figure 3.3, consists of a tab-notebook which contains launch buttons for the various applications. We detail the applications next. In addition, the BioImage Suite “Preferences Editor” can be invoked using the “Preferences” button on the bottom button bar.

The BioImage Suite applications are grouped into seven groups namely:

- General – contains the core applications which are primarily geared towards neuroimage analysis, including Registration tools and tools for the formation of Multisubject Composite Functional Maps.
- DTI/Angiography – includes tools for diffusion weighted image analysis and early versions of our angiography tools.
- fMRI – includes the main functional MRI analysis tool and a GUI-editor for creating .xml files needed by this.
- Editors – contains the SurfaceEditor tool which allows for interactive segmentation of 3D images and surfaces. Two new applications the Mosaic Objectmap Editor and the Orthogonal Objectmap Editor provide functionality for “painting” objectmaps (i.e. manual segmentation) in multiple slices simultaneously. In addition, our specialized Electrode Editor – used for locating intracranial electrodes from CT images, is also available from this tab.

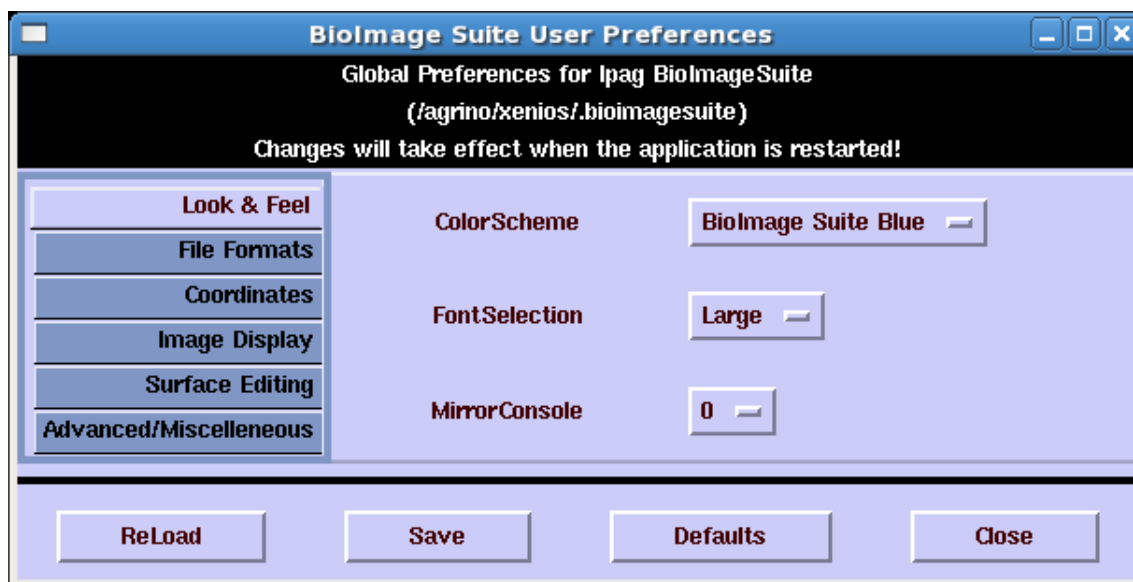


Figure 3.4: The BioImage Suite Preferences Editor.

- Cardiac – has two tools for viewing and surface editing (variations on other tools) specifically adapted to 4D Image Analysis.
- Data Tree – accesses the new datatree tool which serves as a database-like front-end for BioImage Suite. A lot of the future growth of BioImage Suite will involve this functionality.
- Mousesuite – includes early versions of our segmentation/registration Mouse Tools specifically adapted for small animal imaging.

### 3.4 Preferences Editor

The User Preferences Editor enables the setting of global parameters used by the BioImage Suite Applications. The preferences are stored in a file called `.bioimagesuite` in the user’s home directory (in this case `/agrino/xenios/.bioimagesuite` as the dialog itself informs the user). Changes made in the “Preferences Editor” – which is also accessible under the Help menu in many applications only take effect once the application is restarted.

There are six tabs in the editor each containing different sets of options described below:

#### 1. Look & Feel

- **Color Scheme** – chose between System Default, BioImage Suite Blue, Bisque and High Contrast color schemes. See Figure 3.5 for an example.
- **Font Selection** – sets the font for the menus/dialog boxes
- **Mirror Console** – by default in most GUI applications most “print-outs” go to the BioImage Suite Console (accessible under Help/Console in most applications). Setting Mirror

to 1 also sends any printouts to the native console (e.g. the DOS Window or the xterm window from which the application was started).

## 2. File Formats

- Force Output Format – BioImage Suite automatically saves images either as Analyze or NIFTI depending on the input image. Set this to force the output image type.
- Default Import Mode – sets the default import format for the Image Import Tool
- Minc Image Format enables the use of the .mnc format (This is MINC2 which is not to be confused with the more commonly used MINC1 .mnc format).

## 3. Coordinates

- ManualTalairach – set this to OFF unless you really know what you are doing!
- YaleAtlasAutoInitialize – this setting can allow the Yale Atlas Tool to be automatically initialized, when needed.
- WFUAtlasAutoInitialize – this setting can allow the WFU Atlas Tool to be automatically initialized, when needed.

## 4. Image Display

- NormalizeAnatomical – If enabled, the default windowing used for image display is automatically adjusted to improve the contrast (Also available using the “Nr” Colormap in the Viewers)
- Image Colormaps – if enabled, the current colormap is saved upon saving the image and reloaded afterwards. This is also work in progress.
- Interpolation – default interpolation mode for image reslicing.
- Volume Texture Mode – by default, the software will use hardware accelerated volume rendering. Disable on systems with older graphics cards (>5 years old – any recent ATI/Nvidia card should work fine with this on)
- FMRI Mode – selection of default colormap for overlay of functional images (these three choices are also available from the Overlay Tool).

## 5. Surface Editing

- MaxSurfaces – maximum number of surfaces available. Set this to the minimum requirement to speed up the application.
- ControlPointSize – sets the default size of the control points in the spline editor.

## 6. Advanced/Miscellaneous

- RigidRegistration – do not touch this unless specifically instructed to do so.
- VectorVisionLink – enables the VectorVisionLink interface to the BrainLAB Vector Vision Cranial Image Guided navigation system.
- Enable Beta Features – if on, additional options may be available which are considered experimental for now. Keep this off.

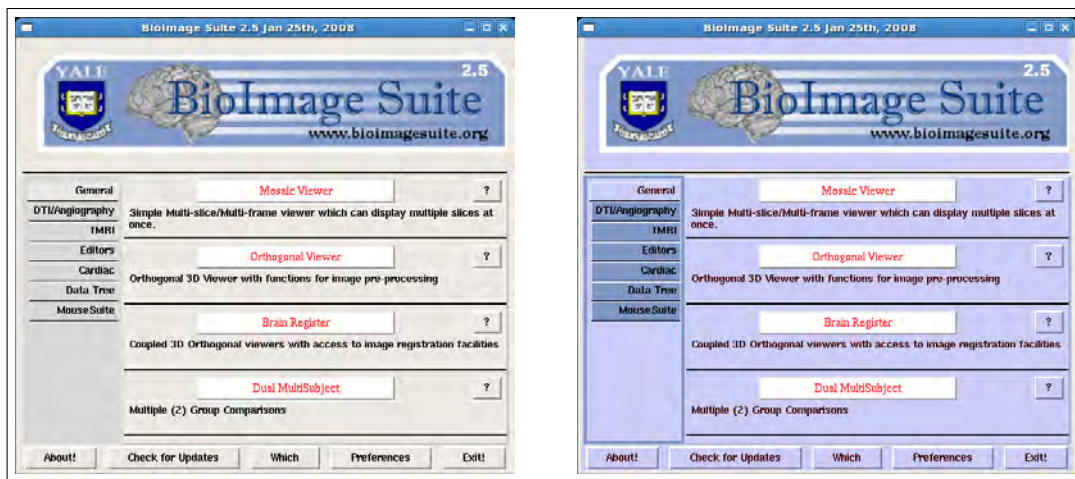


Figure 3.5: Changing the Color Scheme. Left: System (Default) Colors. Right: BioImage Suite “Blue” Color Scheme.

## Chapter 4

# Application Structure

### 4.1 Application Structure Overview

All BioImage Suite applications (e.g. the Brainsegment tool shown above) consist of typically three parts (a) a menu bar at the top, (b) a viewer in the middle and (c) a status bar on the bottom. The exception to this rule are the registration tools which have a main registration application consisting of a menu bar and a status bar and a pair of viewers (Reference Viewer and Transform Viewer) each containing the two images.

There are two basic viewers in BioImage Suite: the Orthogonal Viewer and the Mosaic Viewer. The Orthogonal Viewer also has two extended versions: the 4D Orthogonal Viewer which adds Movie/Cine mode for Cardiac applications and the objectmap viewer (used in Brain Segment, shown above) which allows for the transparent overlay of a presegmented image map (an objectmap) on the original image.

The menu for each application consists of a subset of common options and additional options specific to the application. The following common submenus are available in most applications:

- File Menu – this provides options for Loading/Saving Images as well as looking at Image Properties – described below.
- Display Menu – options for displaying different images – described below
- Image Processing Menu – options for image histogram display/manipulation and image processing tools. Described in more detail here.
- Segmentation Menu – Options for image segmentation, including thresholding/mathematical morphology, histogram clustering, and Markov Random Field (MRF) smoothing. In addition, facilities are included for Levelset segmentation and Bias Field Correction, as well as integration to the FSL software package for use of its Brain Extractor and Gray/White Segmentation Tools. Described in more detail here

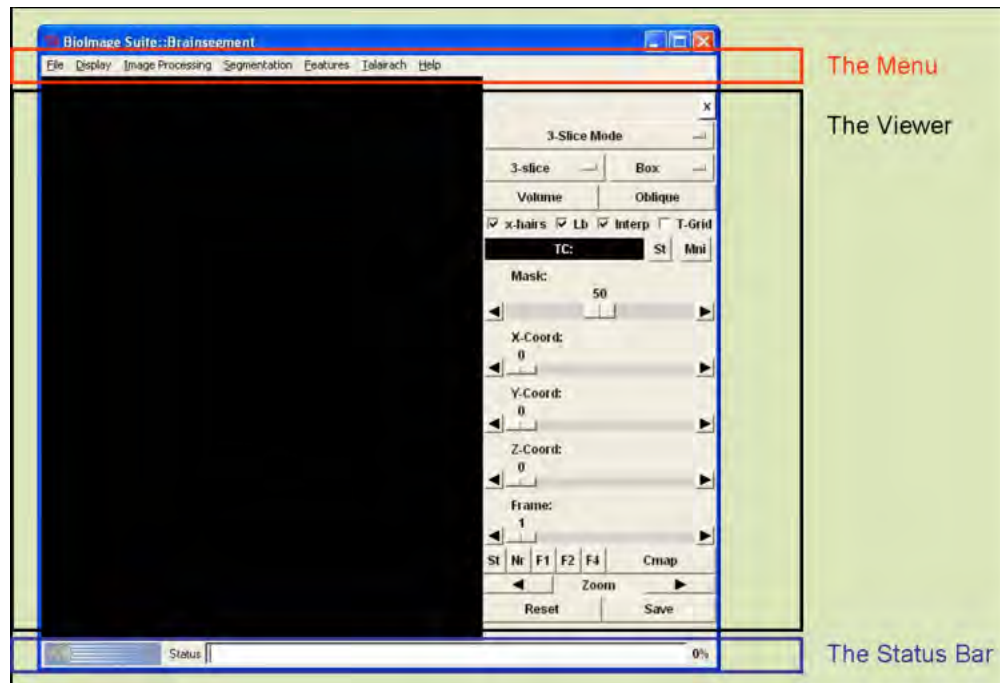


Figure 4.1: BioImage Suite Application Graphical Interface Structure.

- Features Menu – Facilities for clicking and manipulating landmarks, surfaces and surface objectmaps. Additional options may be available here depending on the application
- Talairach Menu – Options for setting the custom transformation for mapping image coordinates to stereotactic space.
- Additional application specific menus may also be present here.
- Help Menu – provides access to the “Preferences Dialog”

## 4.2 The File Menu

### Loading an Image:

To load an image into the viewer’s display, simply choose (File — Load) and select an image header file from the dialog box.

### Saving an Image:

Similarly, to save the image currently in the “Image” display, choose (File — Save), choose a directory with the dialog box, and input a file name. Images are saved in the NIFTI file type. A header is written based on the image dimensions, as well as any information you specified upon

import, if the image was imported from another file type, as described in the next section. This command saves the currently displayed image regardless of whether this is the “Image” or “Results” as described in the Display Menu below.

**Byte swapping** Depending on what platform the data will be used, you may wish to check the “Swap Bytes on Save” option when saving your images. A 16-bit integer requires two bytes to save; one that contains the value for the lower portion (0-255) and one for the higher portion (256-65535). Intel-based machines store the least significant (little) byte first, followed by the more significant byte. This is referred to as “little-endian” storage. Other computers, based on the Motorola 68000 family of processors, however, store integers in the opposite order (“big-endian” storage). The situation is the same when dealing with floating-point and double-precision numbers, which require 4 or more bytes to save. BioImage Suite deals with all these data types, and when reading files, checks to ensure that the byte order is correct (by ensuring that data is within a reasonable numerical range when read for the first time). However, if you are planning on using data processed with BioImage Suite in Windows in other software running on an older SGI machine, for example, you may need to swap bytes on saving. In most cases, you should not enable byte-swapping on save.

### Switch Directory:

This is a very simple convenience feature. It simply switches the base directory currently used by the viewers. Thus, when you go to open a file, the file selection dialog box will begin in this directory. This way, you do not have to navigate back to the same directory repeatedly. Select (File — Switch Directory) and choose a directory. That’s it. Now any other file open dialog box will begin at this directory. It is the equivalent of the Unix “cd” command!

### Standard Images

The (File — Standard Images) flyout contains links to a number of standard images and masks in Analyze format that may be useful for comparisons, as well as experimentation with various program features.

### Custom Images and Directories

A user can add links to their frequently used images or directories using the (File — Custom Images) or (File — Custom Directories) menu item. This allows for faster access and eliminates lengthy navigations to a certain data directory. The files can be added by going to the (File — Custom Images — Add) option. The left image in Figure 4.2 shows the location of the “Custom Images” menu item and the right image shows the result of adding two images.

The list of images can be edit by clicking on (File — Custom Images — Edit) which brings up a window that lets you select/deselect images. Figure 4.3 shows an example where the two added



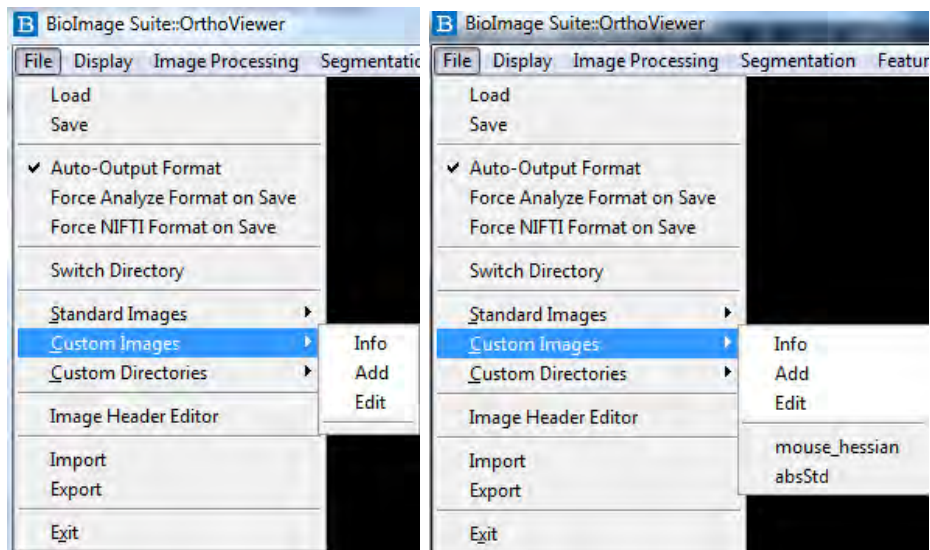


Figure 4.2: Custom Images option in the File menu. The right image shows the result of adding two images.

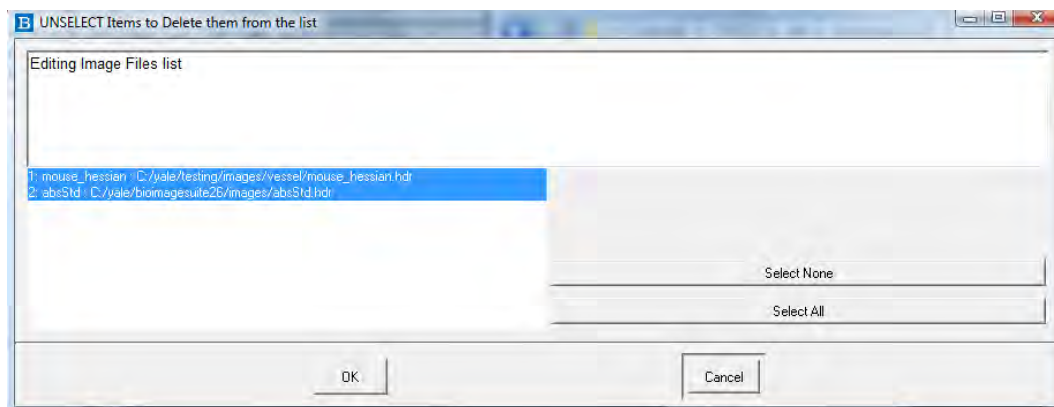


Figure 4.3: Custom Images “Edit” option in the File menu. The image shows the edit custom images window. Images can be selected/deselected by clicking on their listing.

images can be seen. By clicking on any image in the list, it gets deselected and is shown with a white background. All the selected images are shown with a blue background.

The list of these images is read from a file called `.pximagelist` in each users’ home directory. Clicking on (File — Custom Images — Info) brings up a box that informs the user of the location of the file, as shown in Figure 4.4. The format is simply a set of lines (all lines beginning with `#` are treated as comments and are ignored) as shown below for the two files:

```
#BioImage Suite File
# Format
# Image Files : Full Filename
# e.g
```

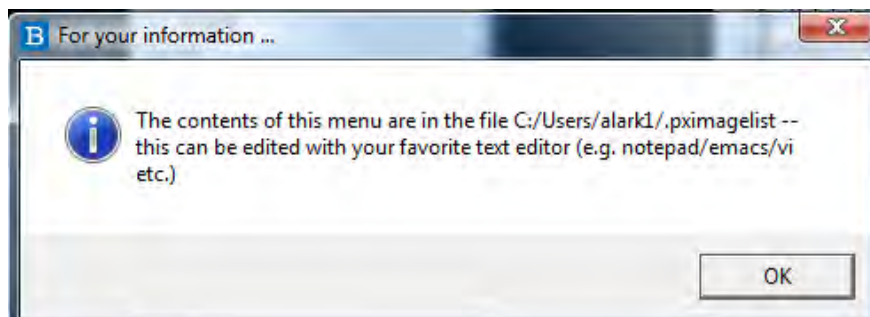


Figure 4.4: Custom Images “Info” option that shows the user the location of the .pximagerlist file which contains the list of all the custom images.

```
# MyFile : myfile .hdr .hdr.gz .nii .nii.gz
mouse_hessian : C:/yale/testing/images/vessel/mouse_hessian.hdr
absStd : C:/yale/bioimagesuite26/images/absStd.hdr
```

Similarly, Custom Directories can be added by clicking (File — Custom Directories — Add). The Add, Info, Edit operations are the same as “Custom Images,” except that the name of the file containing the list of directories is called “.biodirlist” instead of “.pximagerlist”.

### Image Header Editor:

The (File — Image Header Editor) menu choice brings up a tool that allows you to explicitly view and/or edit the header file for an NIFTI/Analyze format image loaded into the viewer. Any changes to the values put into these fields will not take effect until the image is saved and reloaded.

### Importing and Exporting:

The (File — Export) Command allows you to save the image in the viewers “Image” display as a set of JPEG images, corresponding to each of the native image slices. Invoking the (File — Import) command brings up the Import dialog box, which lets you open images of various file types. The “Filename” tab contains a file selection box with possible file types for import. Currently supported file types are: NIFTI, Analyze, Signa LX, Prism (SPECT), Binary, Raw MR, Signa SPR, Nmr47, TIFF, PNM, BMP, and JPEG. The “Properties” tab lets you edit the information contained in the Image Header of the file you are importing. This will be saved with the image when you save it as an NIFTI/Analyze file. If the image for import is not in the Analyze format, the header info will be mostly empty, but will contain information about the dimensions of the image selected. You can also fill in values, which will be written into the header when you save your Analyze file. A more detailed description of this complex control will be made available soon.

### 4.3 The Display Menu

**“Images” and “Results”:** Most applications incorporating a viewer in BioImage Suite, maintain two images simultaneously in memory. One image is stored as the “Image” display, and the other is stored as the “Results” display. When you perform an operation on an image, the output is sent to the “Results” display, which then becomes active (and visible). In order to do more calculations on the result, it must be copied into the “Image” display. To check which display is active, click the Display menu; the radio button associated with the active image will be highlighted.

Under the Display menu, you will find two options: “Image” and “Results” When an operation is performed on an image, the results are saved in the “Results” display of the viewer. This allows you to revert back to the original image, which still resides in the “Image” display. In order to perform further operations on a result image, you must choose (In order to perform further operations on a result image, you must choose (Display — Copy Results to Image). This will overwrite the Image display with the Results display. This process can be undone, uncommitting any changes that have been made and reverting back to the image in memory before copying results, by using the (Display — Undo Copy Results) command. The top two commands in the “Display” menu, “Image” and “Results” simply select which display image is shown. Choosing one does not delete the other – the opposite display image is simply hidden. It can be accessed by switching back. It is important to note that most of the operations provided by the Image Processing toolbox and other analysis tools take the Image Display as input and send their output to the Results display. In order to work with them further, the results must be copied to the Image display.

**Mask Image:** When the application makes use of the ObjectmapOrthogonal Viewer (which can be identified by the presence of a “Mask” Slider above the “X-Coord” slider) there is an additional image in memory - the mask which can be transparently overlaid on the underlying image. In addition, in such cases, two additional options appear under the Display Menu, namely “Copy Mask to Image” and “Copy Mask to Results” which allow for better access to the mask image.

## Chapter 5

# Looking At Images

### 5.1 Image Formats

BioImage Suite uses as its default file format the Mayo/Analyze 7.5 (and since 2.5 the NIFTI) file format. *BioImage Suite supports more or less the complete specification of the NIFTI format. We recommend using this in preference to the Analyze format if possible.*

**Analyze 7.5 format:** In this format images are stored as a pair of files .hdr/.img, for example brain.hdr and brain.img. The header file (.hdr) is 348 bytes long and stores information about the image dimensions (e.g. width x height x depth), voxel dimensions (how large the voxels are), and the orientation (e.g. coronal, axial, sagittal). (In the cases of images with gaps between slices, the voxel dimension is really the voxel to voxel spacing, including the gap!)

Analyze 7.5 is a “non-standard” standard. There are some extensions to it implemented by SPM (e.g. origin and axis direction) which BioImage Suite does not support. Also most implementations of this format do not correctly use the orientation field (e.g. some versions of AFNI).

The assumption made in BioImage Suite for the purpose of displaying the images is that images in Analyze format are stored as follows:

- Axial: x=right to left, y=anterior to posterior, z=inferior to superior
- Coronal: x=right to left, y=superior to inferior, z=anterior to posterior
- Sagittal: x=anterior to posterior, y=inferior to superior, z=left to right (this is for compatibility with the most common local acquisition which results in a non right-handed coordinate system – we suggest that Sagittal images be resliced to Axial before any serious processing)

**NIFTI Format:** This is a modernized version of the Analyze format. It has the advantage that the axis orientation is explicitly specified in the image header (unlike the old Analyze standard in

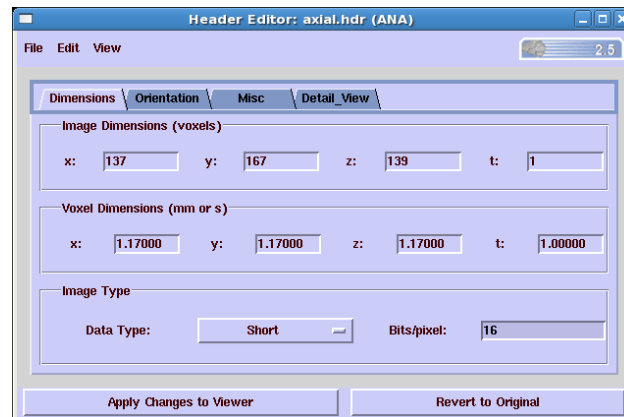


Figure 5.1: The BioImage Suite Image Header Editor. This is accessible under File/Image Header Editor.

which this was implicit). NIFTI images are either stored as single .nii (or .nii.gz compressed) files or as pairs of (.hdr/.img) files as in the Analyze format.

BioImage Suite supports almost the entire NIFTI standard with the exception of images with non-orthogonal axis. When an image is loaded into BioImage Suite it may be flipped to conform to our internal convention (which is implicitly assumed in the case of Analyze images – see above). For example, if an image comes in left to right, it will be flipped to be right-to-left. The header is also changed to reflect this – the image is still a valid, appropriately labeled NIFTI image!

**Other Formats:** Most scanners produce DICOM images, you will need to use external software to convert these to NIFTI or Analyze. The .mnc format is also popular – it was developed at the Montreal Neurological Institute. For arcane reasons, BioImage Suite supports v2 of the .mnc format, unfortunately v1 is more commonly used. Support for .mnc v1 will come with BioImage Suite 3 later this year.

**Image Header Editor:** The image header can be examined and modified (i.e. corrected) using the Image Header Editor (shown in Figure 5.1.)

## 5.2 The Viewers

### 5.2.1 The Orthogonal Viewer

The orthogonal viewer is probably the most commonly used viewer in BioImage Suite. It is shown in Figure 5.2. The images are displayed in the black area on the left side of the viewer window. This viewer can display either (i) single 2D slices, (ii) linked cursor orthogonal slices, (iii) 3D renderings and (iv) combinations of (ii) and (iii). There are a variety of ways to manipulate the views and move through them. For the most efficient control, the use of a three-button mouse is recommended.

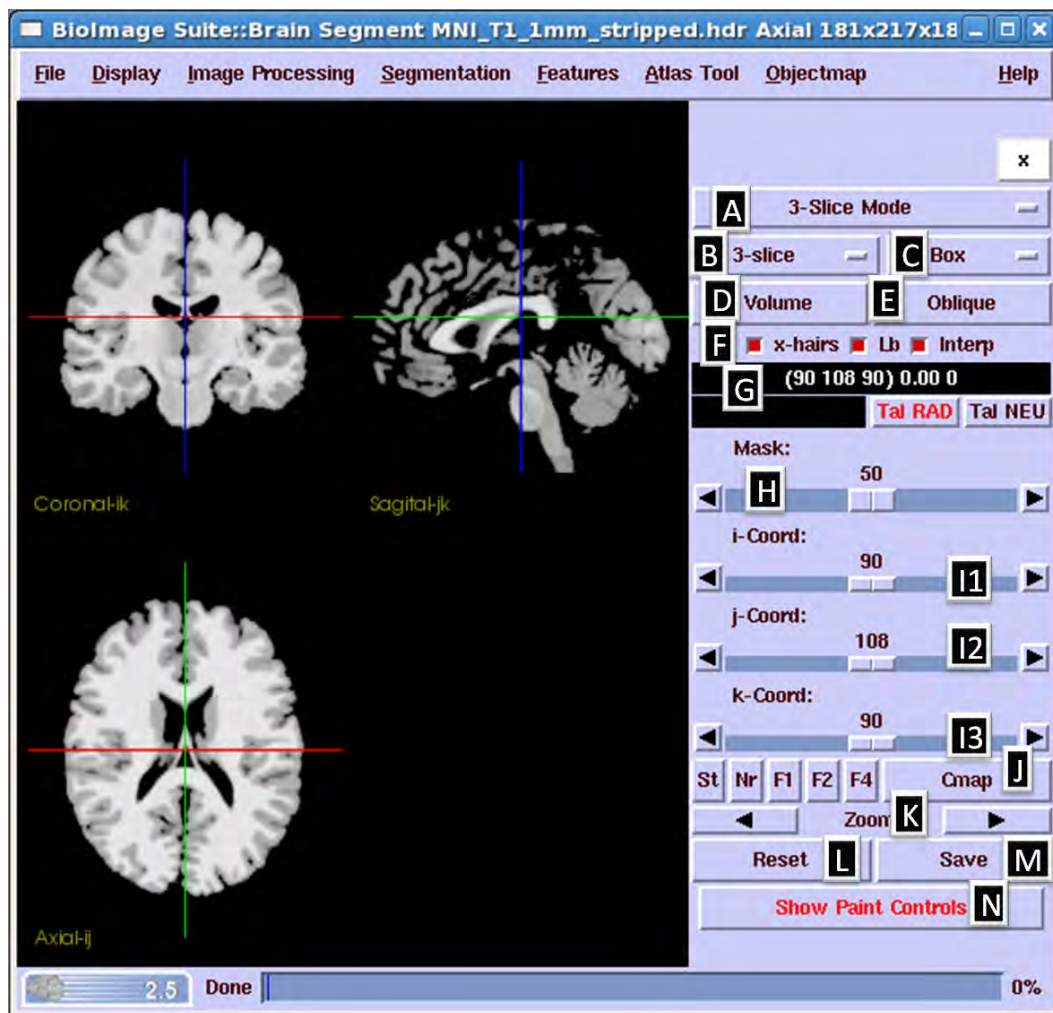


Figure 5.2: The BioImage Suite Orthogonal Viewers.

**A:** The Mode Selector – sets the viewer in 2D, 3D or combination modes. **B,C:** The 3D Mode Selectors – select which of many combinations of elements will appear in 3D. **D,E:** Buttons to access the Volume and Oblique sub-Controls. **F:** “x-hairs” – disables the viewer changing positions when the left mouse button is clicked in the viewer. This essentially locks/unlocks the viewer. “Lb” – this turns on/off the label/axis display in the viewer. “Interp” – this turns on/off Open GL blending interpolation of the image. If this is off and the image is zoomed sufficiently then individual voxels can be seen! **G:** The Talairach control – see later for more on this. **H,N:** These two controls interface with objectmap display and editing (not always present). **I1,I2,I3:** The navigation controls to select the slice display. **J:** Colormap Controls. The “Cmap” button brings up the colormap control, whereas the others set preset colormaps (see below). **K:** – the zoom control which allows the user to zoom in or/out. **L:** – the reset control which restores the view to a good default. **M:** – the save button which can be use to save screenshots of the viewer display as .jpeg or .tif images.



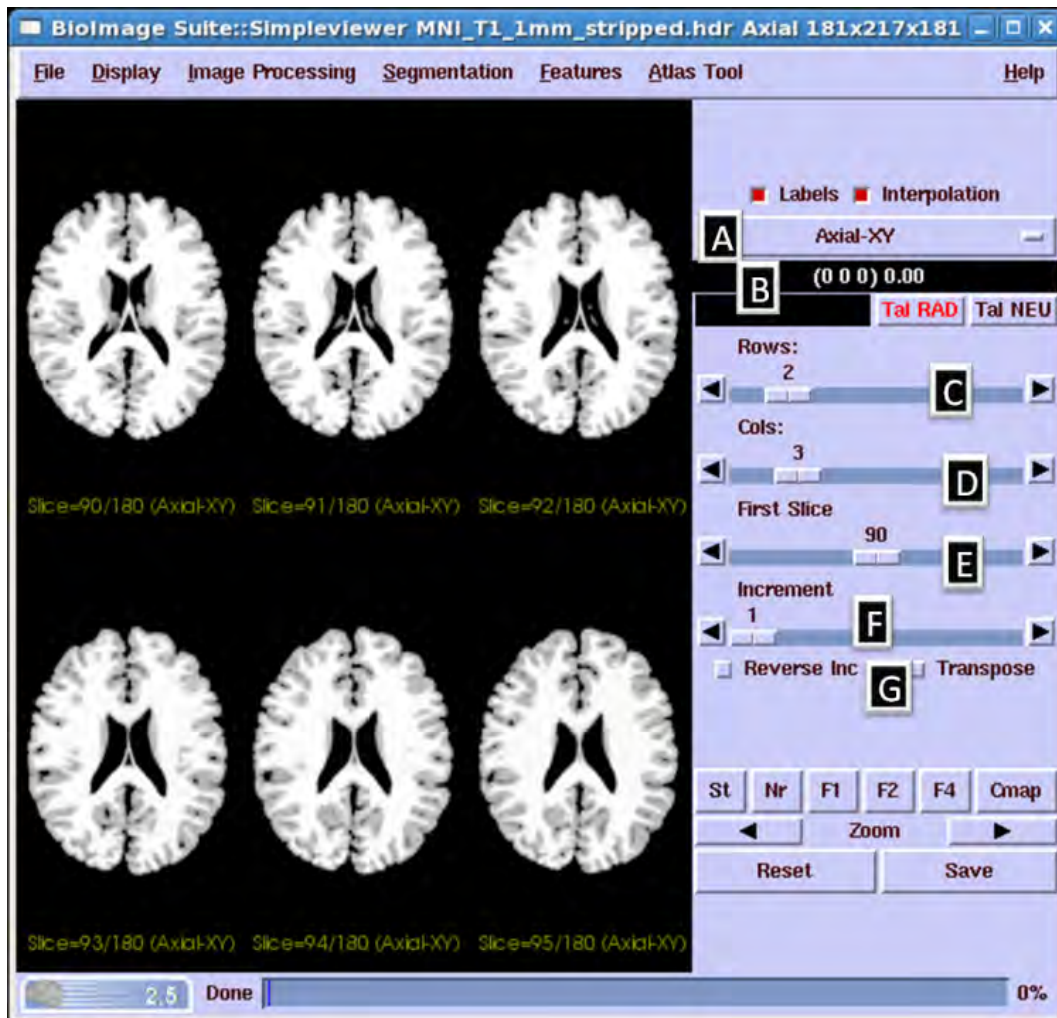


Figure 5.3: The BioImage Suite Mosaic Viewer.

We will only describe here functionality that is different from the Orthogonal Viewer. **A:** The Mode Selector – selects the orientation of the displayed slices. **B:** The Talairach control – see later for more on this. **C,D:** Controls how many slices to display in Rows  $\times$  Columns. **E:** Selects the slice for the top left view. **F:** Selects the increment, if this is greater than 1, then the viewer skips slices. **G:** Additional controls to reverse the order of the slices and to transpose the order in which they are displayed.

**Mouse Control in the viewing area** *Button 1 (left)* Click the left mouse button (button 1) anywhere on any of the 2-D slice views to set the cross-hairs to this point. Click and hold button 1 as you drag through the view space to move the cross-hairs cursor when it is active, and to rotate the 3-D display, when it is shown. (Unchecking the “x-hairs” box in the viewer control panel disables mouse-in-viewer movement of the cross-hairs. In this case, use the slider bars as described below.)

*Button 2 (middle/wheel)* Click and drag with button 2 to scale the display. Drag down to zoom in, and drag up to zoom out.

*Button 3 (right)* Clicking and dragging with mouse button 3 translates the display. Use it to slide the display up, down, and side to side. Pressing the shift key and the right mouse button brings up the Renderer Control.

## 5.2.2 The Mosaic Viewer

The mosaic or “simple” viewer can be used to display sequences of slices in the same orientation. It is shown in Figure 5.3, where the functionality different to that of the orthogonal viewer is highlighted.

## 5.2.3 Colormaps

Colormaps (or lookup tables) are essentially functions that map image intensity to display color. The most common colormap (which is often implicitly used) simply maps the lowest intensity value in the image to black and the highest intensity value to white. For example, in the case of an image with range 0:255, 0 is mapped to black, 255 to white and everything in between to progressively lighter shades of gray. The most common medical image colormap is the so called Level/Window colormap, illustrated in Figure 5.4(right). This colormap is defined by the variables, the level  $l$  and the window size  $w$ . The mapping  $x \mapsto y$ , where  $x$  is the input intensity and  $y$  the output color (from *black* = 0 to *white* = 1) is then specified as:

$$y = \begin{cases} 0 & \text{if } x \leq l - \frac{w}{2} \\ 1 & \text{if } x \geq l + \frac{w}{2} \\ \frac{x - (l - \frac{w}{2})}{w} & \text{otherwise} \end{cases}$$

More complex colormaps can be defined by using the full spectrum of RGBA (Red, Green, Blue, Alpha=Opacity). This allow us to map different values to different colors to highlight certain effects, e.g. functional activations overlaid on anatomical data. Colormaps become a lot more interesting when volume rendering is involved.

BioImage Suite has a colormap editor (Section 5.3) for manipulating the colormap, as five preset maps labeled **St**, **Nr**, **F1**, **F2**, **F4** – the later being directly accessible from the viewer controls. **F1**, **F2** and **F4** are overlay maps used for displaying functional overlay activations. **St** is the standard map where the darkest voxel is mapped to black and the brightest to white. **Nr** is a normalized



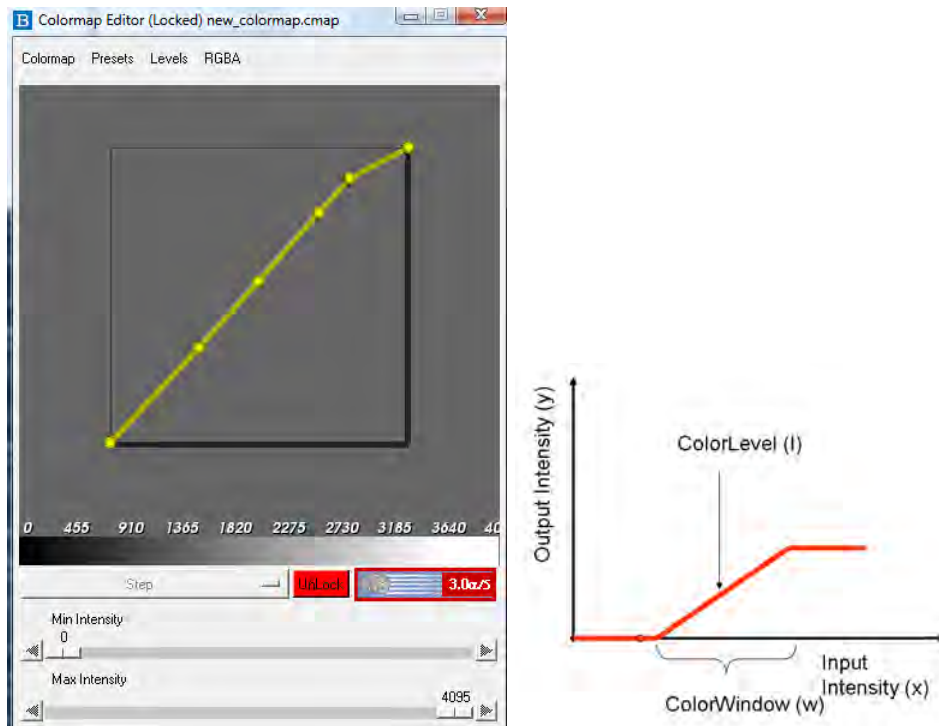


Figure 5.4: **Left:** The BioImage Suite Colormap Control. This is accessed using the “Cmap” button on the viewers. **Right:** The Window/Level Colormap.

colormap, where the window and level are set automatically to map the color spectrum from 1% to 99% of the cumulative histogram. This usually saturates the brightest voxels and results in better contrast in anatomical MRI scans.

### 5.3 The Colormap Editor

The Colormap editor consists of a menu bar that allows the selection of Presets (for fMRI) as well as controlling the maximum number of colors in the colormap, by clicking on 'Levels'. The 'RGBA' button in the menubar allows for specifically editing one of the curves that represent Red, Green, Blue, Alpha (Opacity) and RGB (All three colors at the same time).

In the editor, the points of the curve can be edited with the mouse, which in turn ends up editing the colormap. At a time, only a single curve can be edited. The curve to be edited is picked by using the menu button, as described above, or the right mouse button cycles through all the possible choices of curves that can be edited.

The current curve that is being edited obtains the spherical handles that the user can move around. The other curves are shown in the background for reference.

Based on the changes made to the colormap, a scalar bar gets updated automatically and is shown at the bottom. Figure 5.5 shows a grayscale color bar. More complex color bars are discussed later.

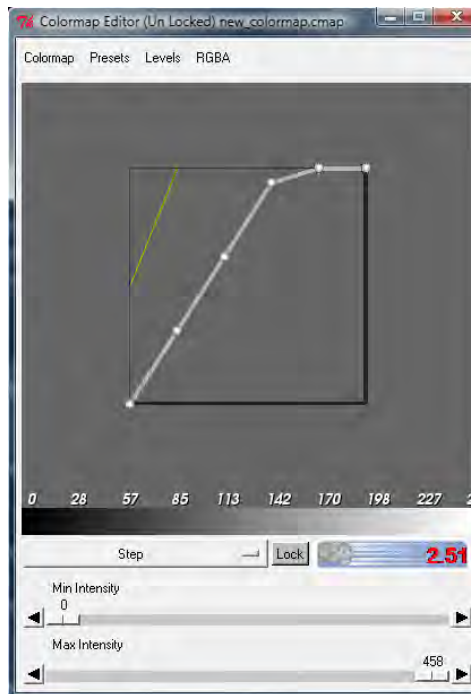


Figure 5.5: The colormap editor.

The user is also able to manually specify the minimum and maximum intensity using the slider bars. The colormap gets rescaled accordingly and is reflected in the color bar immediately.

In our colormap editor, we provide a number of preset controls. The default colormap is set to 'Step' which is a greyscale colormap that varies from black to white. Greyscale colormaps are proven to be more accurately perceivable than rainbow colormaps. Rainbow colormaps, though widely used, are considered to lead to misinterpretation of data values in an image [13].

In BioImageSuite, we allow the user to select one of

- Step - The Step option sets a greyscale colormap that varies from black to white with shades of grey in between.
- Gamma
- Constant-Hue
- Overlay
- Complex Overlay
- Temperature - Black body radiation-based colormap
- Rainbow - The colormap in which the Hue of the color is changed based on the intensity.

Figure 5.6 shows a screenshot of the preset colormaps in BioImageSuite. The drop down menu is invoked by clicking on 'Step' in Figure 5.5.

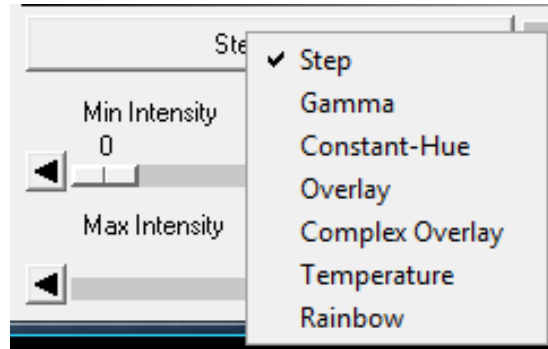


Figure 5.6: Preset colormaps built into BioimageSuite.

### 5.3.1 User-controlled colormaps

The user can define their own colormap as needed. To facilitate interaction with the colormap, we allow the user to modify the colormap using the mouse. At a time only one of curves for Red, Green, Blue, Alpha or RGB can be edited. The curve that is being edited is highlighted by a thicker line and control points in the form of spheres that allow manipulation of the curve. In the user defined colormap shown in Figure 5.8, the red curve is being edited and is therefore shown by a thicker linewidth. The spherical control points for the red curve too can be seen. The other curves can be seen behind it. The left image in Figure 5.8 shows a volume rendering of CT data using the colormap on the right. The bottom image shows another example resulting image of a user-defined colormap that highlights the vessels in a CT scan of a patient.

The colormap can also be controlled by using the 'Complex Controls' which can be accessed by clicking on 'RGBA' in the menubar and then clicking on 'Complex Controls'. The Complex controls allow the user to fine tune the controls as per the needs. Based on the changes made to the controls using the slider bars, the curves are updated in the main colormap editor. Figure 5.9 shows an example where the red value of the colormap has been modified and the changes can be seen in the images as well as the colormap editor.

## 5.4 Coordinates for NeuroImaging

The most popular Neuroimaging coordinate sets are (i) the Talairach coordinates – as defined by the Talairach atlas and (ii) the MNI coordinates as defined on the MNI template. In both cases the origin (0,0,0) is the AC (Anterior Commissure) and the axis are oriented as:

- X: Left to Right
- Y: Posterior to Anterior (i.e. back to front)
- Z: Inferior To Superior (i.e. bottom to top)

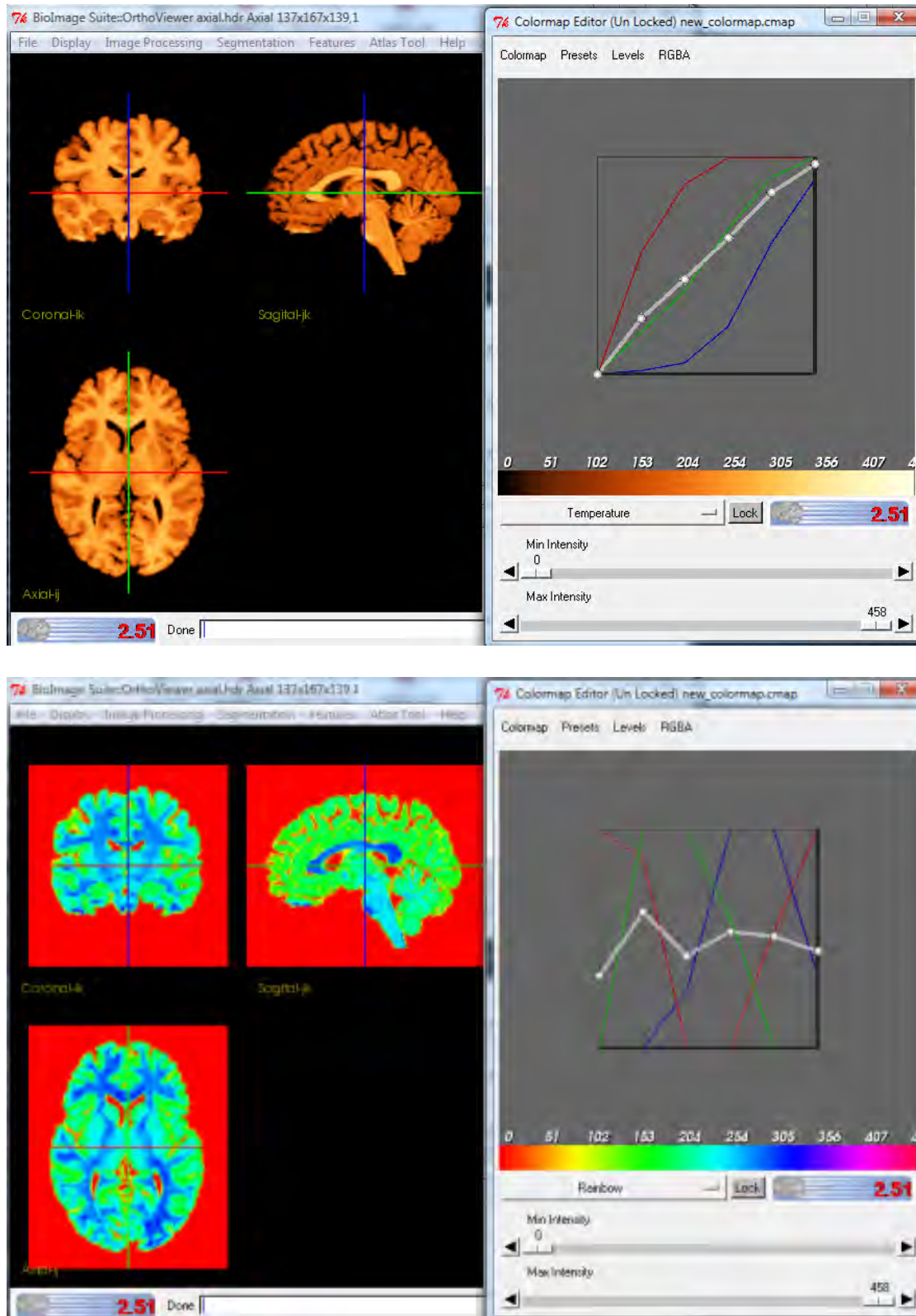


Figure 5.7: These images show preset colormaps that are built into BioImageSuite. The top image shows the temperature colormap and the bottom image shows the rainbow colormap.

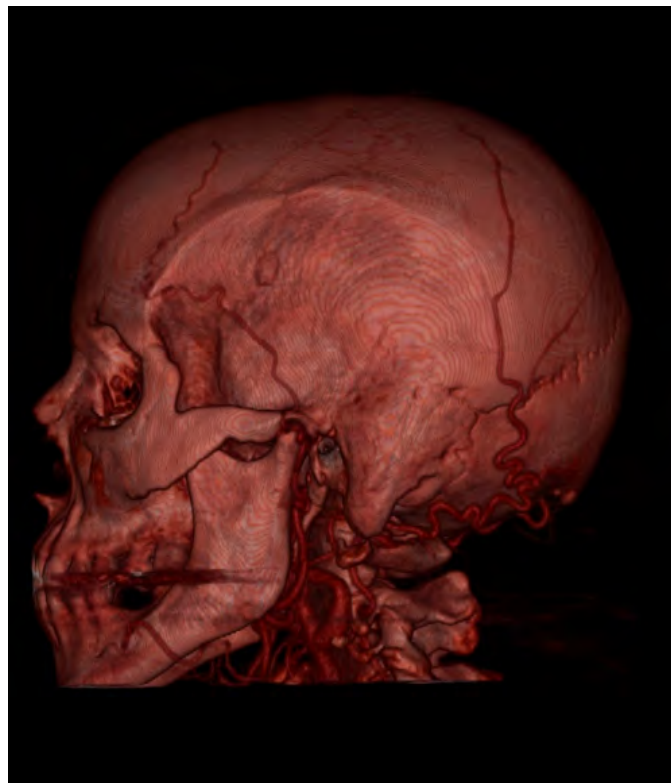
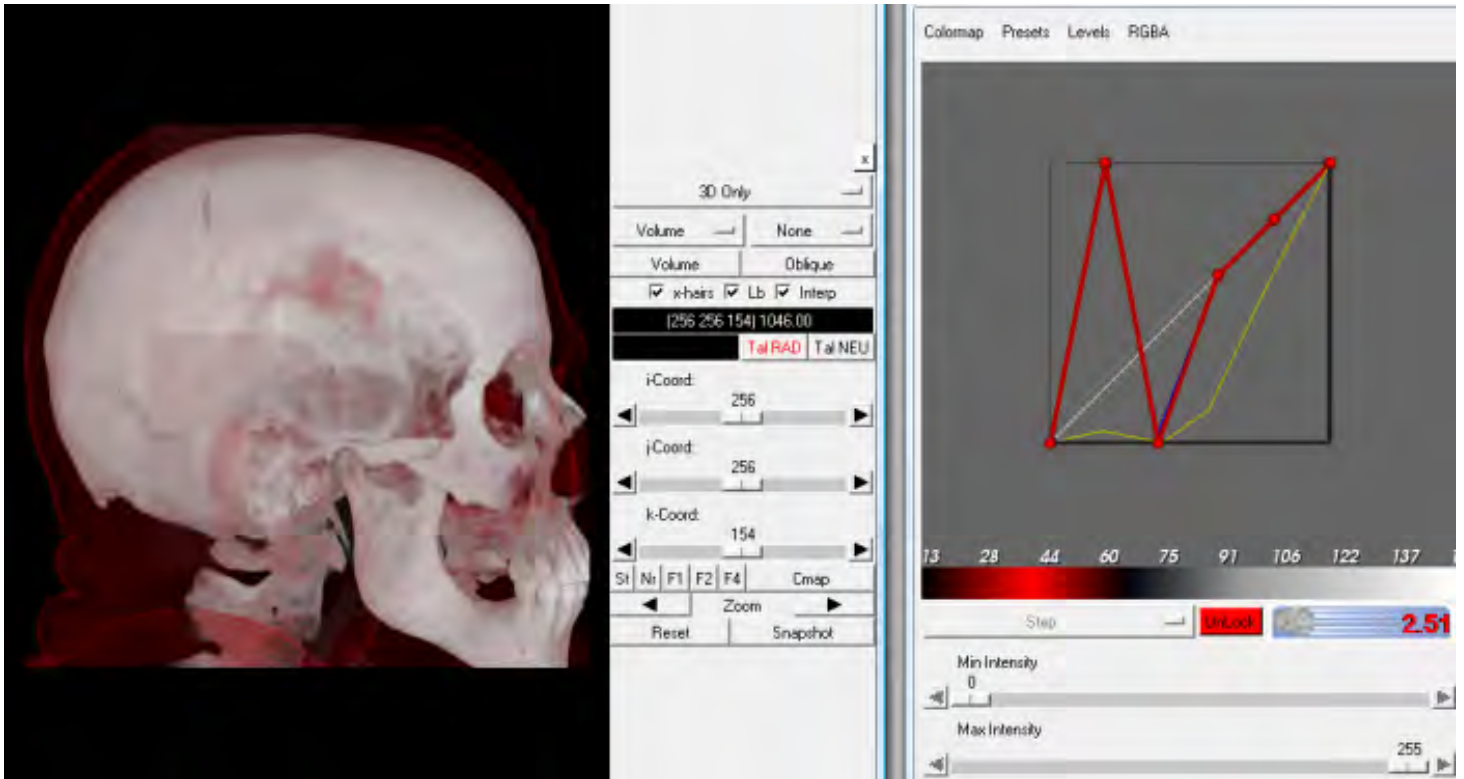


Figure 5.8: A user defined colormap highlighting the skull in the CT data. At the same time the skin around the skull is shown in red. The bottom image shows a volume rendered image generated by a different user defined colormap. The vessels are clearly highlighted here and can be easily seen.



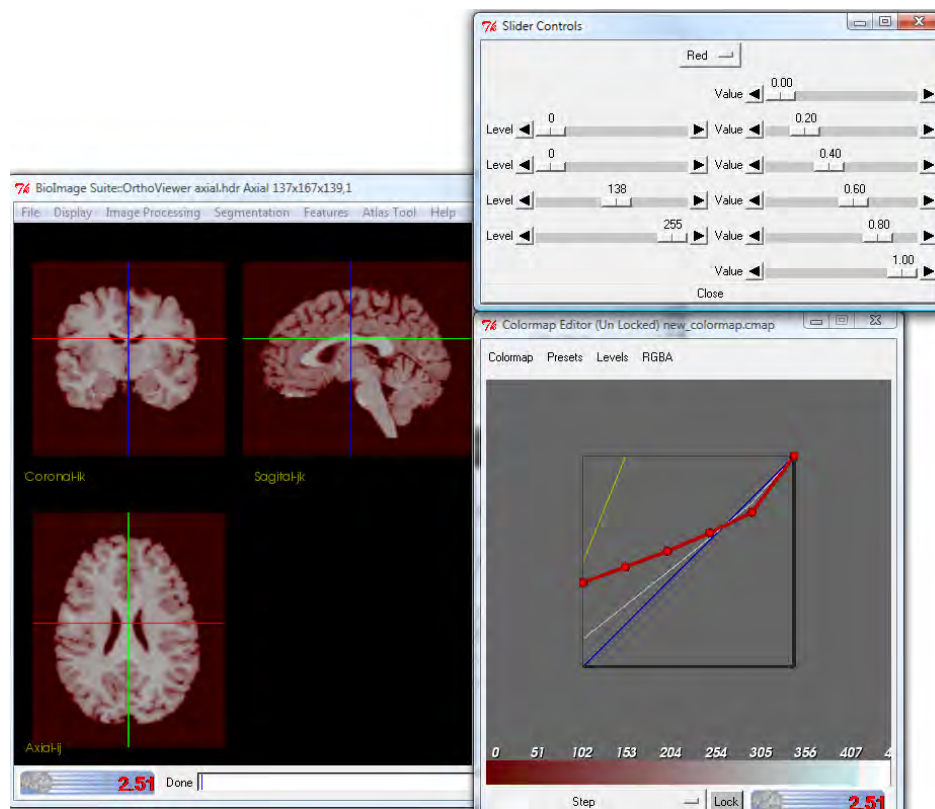


Figure 5.9: This image shows a screenshot of the Complex Controls being used to edit the colormap. The changes in the red value made using the slider bar are updated in the color bar as well the curves in the colormap editor. Accordingly, the three images on the left too have been updated.

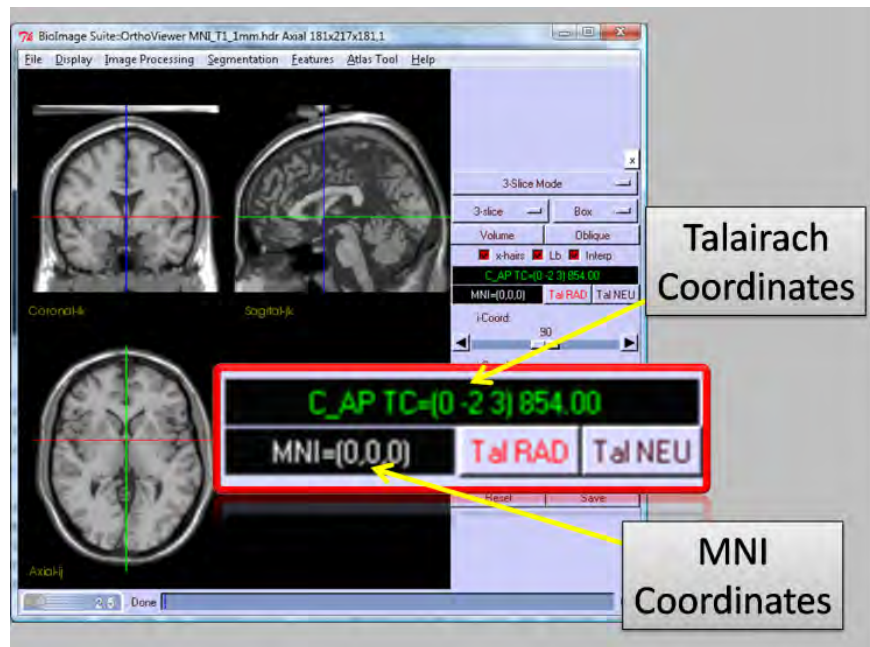


Figure 5.10: Talairach Coordinate facilities in BioImage Suite.

This is also known as RAS (Right, Anterior, Superior) setup. Both of these systems use millimeters as units, so a coordinate of (1,0,0) implies that the point is located 1mm to the right of the AC. For comparison, BioImage Suite internally and the DICOM standard tend to use an LPS coordinate system (i.e. X: Right to Left etc.)

The MNI coordinate system is relatively straightforward to use as there is an actual MRI image of the “MNI template” that defines the space. The Talairach system is based on a paper atlas and is harder to map to an actual MRI image. BioImage Suite uses a custom nonlinear mapping [56] to map MNI coordinates to Talairach space.

**Obtaining Coordinates** Once a brain image is registered to the MNI Template brain, and resliced to have the same voxel dimensions (Axial 1x1x1 mm resolution, 181x217x181) as this, it is trivial to obtain Talairach/MNI coordinates.

First, examine whether the Y-coordinate in the image (i.e. the actual pixel ordering as represented by the Y-Coord in the viewer) is increasing from Anterior to Posterior (BioImage Suite default) or Posterior to Anterior.

Next, press either the “Tal Rad” button in the “coordinate bar” (see Figure 1) if the y-coordinate increase from anterior to posterior or the “Tal Neuro” button otherwise. Pressing either of these buttons for the first time may result in a slight delay (5-10 seconds) as the lookup table from MNI Template image coordinates to Talairach space is loaded.

Now, simply navigate with the left mouse button in the viewer. The coordinates in the black label box, displayed in red, inside the “coordinate bar” are the Talairach coordinates under the mouse. (There are four numbers in the text box: the first three are the coordinates, the fourth number is

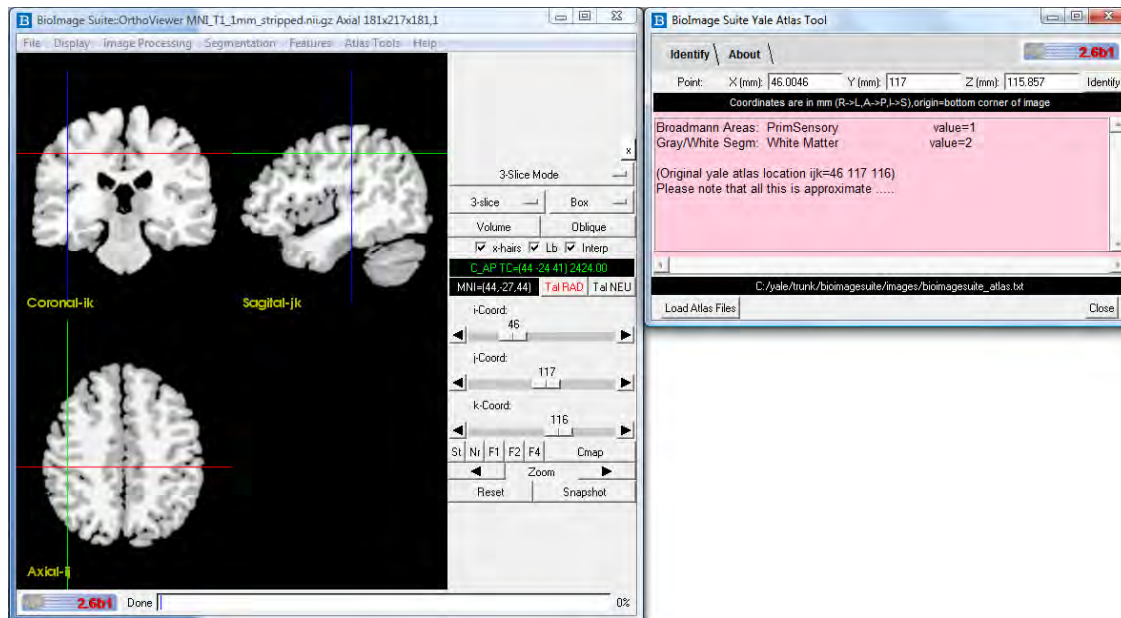


Figure 5.11: The Yale Broadmann Areas Atlas Tool

the image intensity at this point).

If the internal nonlinear map is used then the Talairach coordinates are printed in green with the prefix C-AP or C-PA (depending on whether the brain y-axis is anterior-to-posterior – Rad, or posterior-to-anterior – Neuro). The box below this (with the prefix MNI) shows the MNI Coordinates. See Figure 5.10.

*Note:* BioImage Suite can not actually tell whether an image is in MNI space or not. However, if it has the right dimensions, it assumes that it is in MNI space and that the user “knows what s/he is doing”.

## 5.5 Atlas Tools

### 5.5.1 Yale Broadmann Areas Atlas Tool

BioImage Suite provides a recently generated atlas of Broadmann areas defined on the MNI T1 template (the “Colin27” brain) at 1mm resolution [56]. To use the Yale Atlas tool:

1. Load an image into the viewer that is in MNI space (either 1mm, or 2mm resolution is fine).
2. Select the “Yale Broadmann Areas Atlas Tool” from the Atlas Tools Menu.
3. Load the atlas files using the Load Atlas Files button.
4. Confirm that the image is an MNI space by clicking the Tal RAD button.



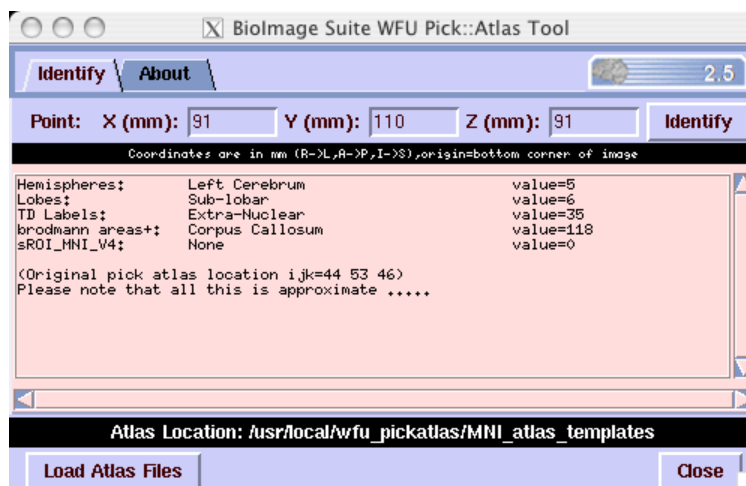


Figure 5.12: The WFU Pick::Atlas Tool.

5. You can now browse the image with your viewer, and read out your location in the brain and other atlas information from the Atlas Viewer’s Identify tab – see Figure 5.12.

Figure 5.11 shows a screenshot of the Yale Atlas tool that shows the cross hairs identifying a 3D location in space. Based on the Yale Brodmann Areas Atlas, a region is identified on the right as shown in this case to be the “PrimSensory” which is the “Primary Sensory Area” of the brain. The 3D location of the point are shown for reference. The atlas file’s location is also shown below for reference.

In User Preferences, the “YaleAtlasAutoInitialize” can be enabled /disabled to automatically initialize the Yale Atlas tool, see Section 3.4 for more details about the Preferences Editor.

## 5.5.2 WFU Atlas Tool

BioImage Suite can also interface to the WFU Pick Atlas [61] (which derives from the Talairach Daemon work [57]) to perform atlas lookups for specific brain locations. Note that these labels are at best approximate, but can be useful nonetheless. To use the WFU Pick atlas tool, you will need to first install the WFU pick atlas somewhere on your disk – see instructions below. To use the atlas do:

1. Load an image into the viewer that is in MNI space (either 1mm, or 2mm resolution is fine).
2. Select the WFU pick tool from the Atlas Tools Menu.
3. Load the atlas files using the Load Atlas Files button.
4. Confirm that the image is an MNI space by clicking the Tal RAD button.
5. You can now browse the image with your viewer, and read out your location in the brain and other atlas information from the Atlas Viewer’s Identify tab – see Figure 5.12.

---

In User Preferences, the “WFUAtlasAutoInitialize” can be enabled /disabled to automatically initialize the WFU Atlas tool, see Section 3.4 for more details about the Preferences Editor.

See Section A.2 for instructions as to how to obtain and install the WFU Pick Atlas.

## Chapter 6

# Advanced Image Visualization

### 6.1 4D Images

Often medical images are acquired in temporal sequences over time. Common examples of this are fMRI T2\* time series and cardiac imaging.

BioImage Suite can happily handle 4D image display and manipulation. In fact, BioImage Suite

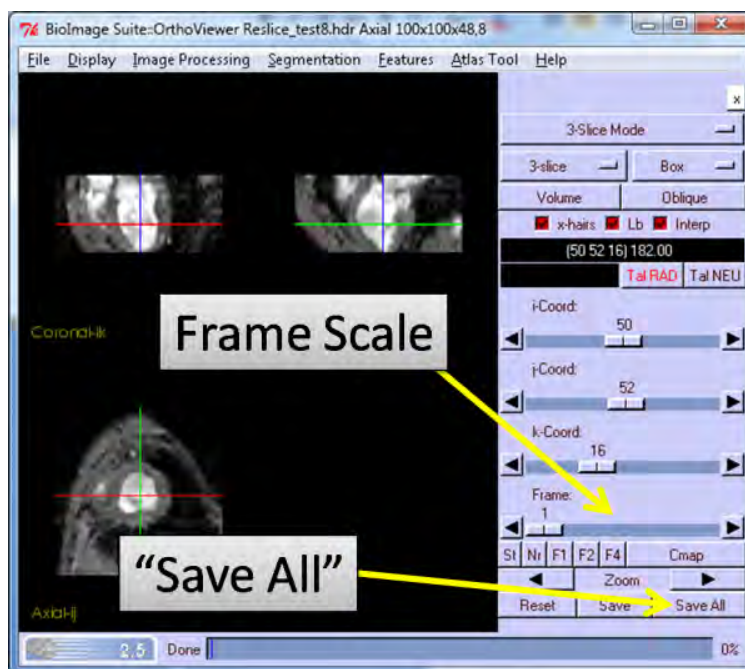


Figure 6.1: Displaying Image Sequences. When a 4D image is loaded, two additional controls appear in all viewers. (i) The Frame Scale which allows the user to change frames and (ii) A “Save All” button which takes a series of snapshot of the current viewer – one for each frame. ( This is equivalent to acquiring a series of snapshots manually using the “Save” button next to it).

started out as a package for cardiac image analysis, so the functionality for 4D images was there from the beginning (e.g. the 4D version of the SurfaceEditor predates the 3D version by about 7 years!). When a 4D Image is loaded in a BioImage Suite viewer, a couple of extra controls appear to facilitate manipulation of this type of images, as shown in figure 6.1.

In addition, BioImage Suite has a specialized viewer that can be used to “play” movies (or cine-loops) of 4D images. Applications that use this viewer can be found under the “Cardiac” tab of the BioImage Suite main menu. Figure 6.2 shows a snapshot of the “VolumeViewer” application. While this application, has some specialized functionality for Cardiac Image processing (to be found under the “Cardiac” menu), it can also be used to play movies of fMRI timeseries data. This can be useful for checking motion correction, intensity stability etc.

## 6.2 3D Rendering Controls

The orthogonal viewer has four sub-viewers. Three of these mimic 2D displays and are used to show one of the three orthogonal views “from above” e.g. Axial slice, and have stringent controls on where the camera can move (i.e. it is forced to stay in the same plane). The fourth viewer (the 3D Renderer) allows the user to completely control the camera position and to interact with the images (and surfaces, electrodes, landmarks etc) in 3D. There are two aspects of this renderer that we will discuss: (i) how the images are displayed – see Figure 6.3 – e.g. enabling/disabling volume rendering etc and (ii) the camera controls.

**The Camera Controls:** In 3D Mode the user is free to move the camera as they please around the viewer. In particular, they can manipulate the camera position, rotation, zoom and clip planes. In addition, six preset camera positions are available and the user can also program (and save) other presets as needed. The camera controls or “3D Renderer Controls” can be accessed by pressing “Shift + Right Mouse Button” anywhere in the 3D Viewer. The available controls are explained in more detail in Figure 6.4. In addition, the left mouse button is used to rotate the camera, the middle mouse button (press the wheel!) is used to zoom in and out and right mouse button is used to translate the camera.

## 6.3 Volume Rendering

### 6.3.1 Direct volume rendering

Volume rendering is defined as the process of generating a 2D image **directly** from three-dimensional data. The basic idea is to simulate the absorption and emission of light as it goes through matter.

To simulate the passage of light through data, rays are cast from the image plane into the space containing the volume, as shown in Figure 6.5. For every pixel in the image plane, a ray is cast into the volume space that traverses the volume. At each point along the ray, the data is sampled

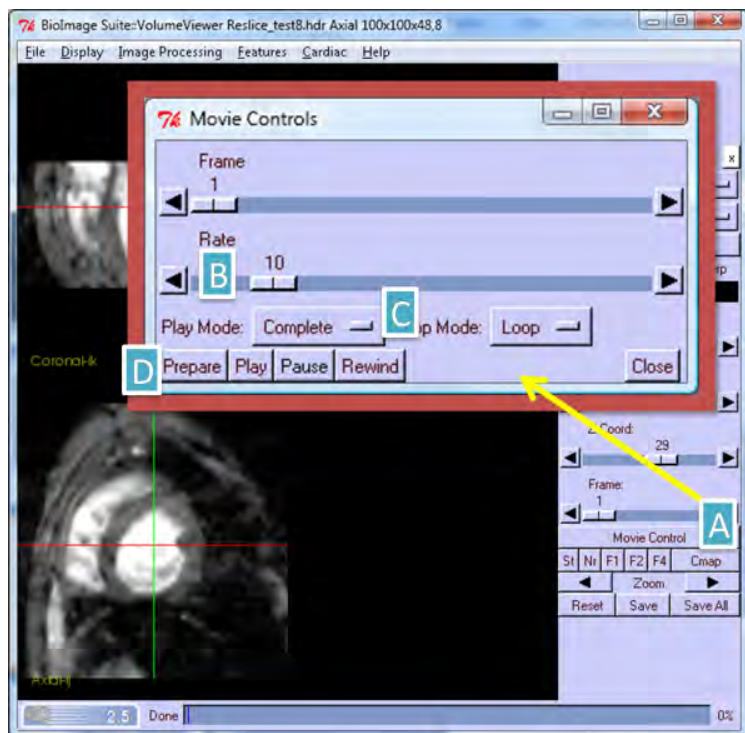


Figure 6.2: The specialized 4D viewer. One additional button (marked as **A**) called “Movie Control” is present. Pressing this shows the “Movie Controls” dialog which is shown on top of the main viewer. Within this, there are standard facilities for playing movies. The speed of the movie (in frames/second) is set using the “Rate” scale (**B**). This is maximum speed to be used and may not be achievable depending on the hardware. There are two modes of playing movies (a relic of older, slower graphic cards). The mode is selected by the “Play Mode” option menu (marked as **C**) and can be either “Complete” or “Fast”. Using “Complete” mode will result in slower performance – but the full viewer facilities (e.g. changing slice etc.) are available during the movie playing. Using “Fast” (as the name implies) will result in faster performance as BioImage Suite will cache all display prior to playing the movie. **Before** playing a movie in “Fast” mode use the “Prepare” button (marked as **D**) to cache all frames.

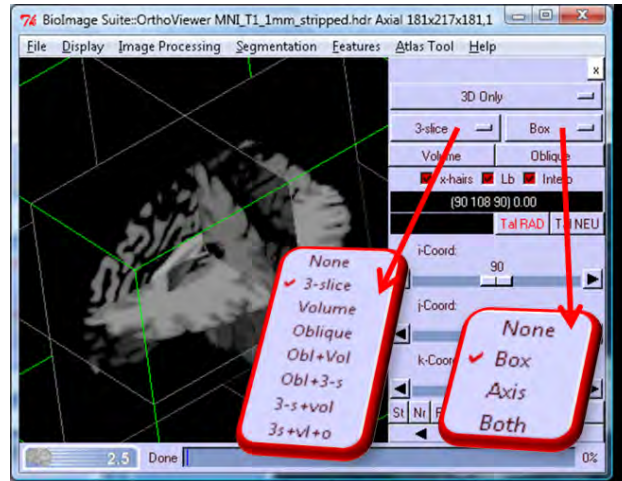


Figure 6.3: The orthogonal viewer has two option menus for controlling what gets displayed in the 3D Renderer. These are highlighted in this figure. The first control selects which of the combinations of possible 3D image displays are used (none, 3-slice display, volume rendering, oblique slice). The second selects the “decoration” around the image. “Box” refers to a cube placed around to show the outline (extent) of the image, whereas “axis” shows the native axes of the image (Red for the i-axis, Green for the j-axis and Blue for the k-axis – the slice selection).

to identify the density/intensity at that point. This value is different for different materials such as bone, tissue, fat and so on. Based on the intensity at that point along the ray, the color and opacity is identified using a lookup table called a *colormap* or *transfer function*.

### 6.3.2 Types of compositing functions

Color and opacity are accumulated along the ray and various compositing operators are used to get different effects. An *X-ray* image can be simulated by averaging the intensity values sampled along the ray. A *MIP* (Maximum intensity projection) image is obtained by preserving the maximum intensity value along the ray. MIP images are frequently used to visualize vascular structures as they can be clearly seen using this technique. Figure 6.6 shows an example of the two techniques.

The compositing technique used for direct volume rendering is based on accumulating the color based on the color and opacity of the current voxel under consideration. For example, during ray traversal if in a CT scan, bone is encountered, it absorbs more energy than tissue or a thin vessel.

### 6.3.3 Types of Volume Rendering

There are four types of volume rendering techniques: raycasting, texture mapping-based, shear warp and splatting. Raycasting and texture mapping-based techniques are the most widely used of the four. *Raycasting* is the technique in which an image is generated by casting rays into the volume space.

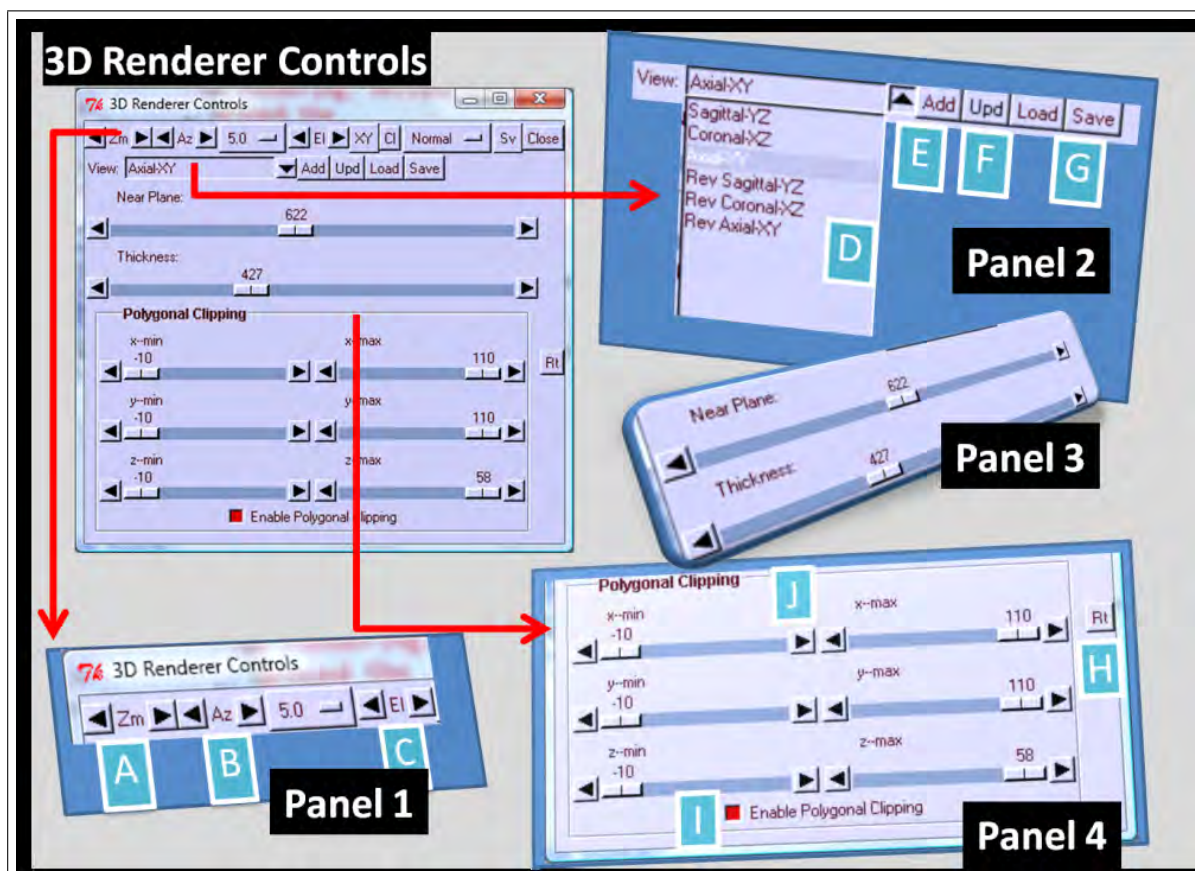


Figure 6.4: **The 3D Renderer Controls.** **Top Left:** The 3D Renderer Control Windows. This has 4 panels shown in zoom-ups below.

**Panel 1: Camera Zoom and Rotation.** There are three controls for setting camera zoom and rotation. **A:** The Zoom Control. **B:** The azimuth rotation control and **C:** The elevation rotation control. The amount of rotation is set by the drop menu between the two rotation controls.

**Panel 2: Camera Position Presets:** The drop menu (**D**) allows the user to select the preset camera position (think of these as camera “bookmarks”). There are six positions by default. Additional positions can be set using the “Add” button (**E**). A current preset may be updated with the camera position using the “Upd” button (**F**). Camera presets can be loaded and saved to file using the “Load” and “Save” buttons respectively. (**G**).

**Panel 3: Camera Clipping:** There are two controls here for setting the near clip plane and the clip thickness. Objects nearer to the camera than “Near Plane” or further away than “Near Plane+Thickness” are eliminated from the scene. These are restored to their default values using the “Rt” (Reset button) highlighted as **H** in Panel 4.

**Panel 4: Polygonal Clipping:** In addition to the general camera clipping controls (Panel 3), Bioimage Suite has additional clipping controls to “cut” either volumes or surfaces separately in planes that are not perpendicular to the camera. The volume clipping controls are on the Volume Control. The controls in this panel can be used to clip surface objects. The clipping controls are enabled using the “Enable Polygonal Clipping” checkbox (**I**). Six scales are provided to crop in each axis (**J**).



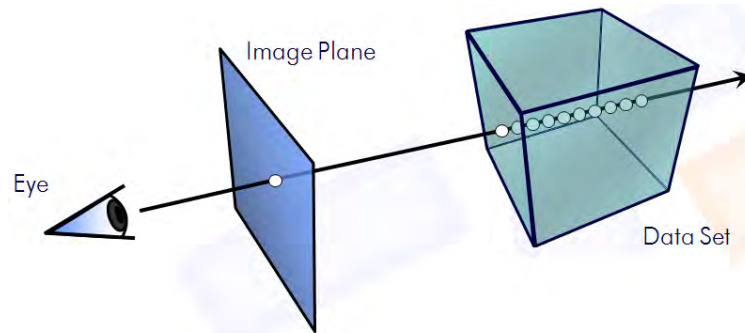


Figure 6.5: This image depicts a schematic of the volume rendering process. For every pixel in the image plane, a ray is cast into the volume space. Along the ray, the volume rendering integral is evaluated to simulate the passage of light through matter. At each position along the ray, the volume/data is sampled to identify the value at that location. The location identifies the material properties such as absorption and emission. A color and opacity is obtained for every density by performing a lookup into a table, called the *colormap* or *transfer function*. This allows users to color different regions of the volume differently.

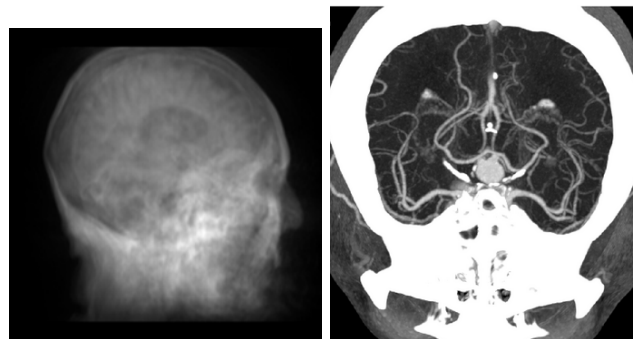


Figure 6.6: The left image shows an X-ray style image that can be generated by average the intensity values along the ray. The right image shows a MIP image that can be obtained by picking the maximum intensity observed along that ray. Such MIP images are widely used to visualize vessels.

As graphics hardware got better with time, researchers devised a way to perform volume rendering using graphics hardware. This technique loads the data into graphics memory and utilizes 2D (and more recently 3D) texture maps to perform interpolation and blending. Since graphics hardware is extremely fast, this technique is much faster than the raycasting technique described earlier.

### 6.3.4 Volume Rendering Facilities in BioImage Suite

BioImage Suite has support for three types of volume rendering namely (a) software based ray-casting, (b) texture mapping accelerated rendering and (c) software maximum intensity projection. Each has their own applications. For most studies using reasonable hardware option (b) is optimal (and hence the default).

In the case of texture mapped volumes, it is worth noting that the underlying graphics hardware



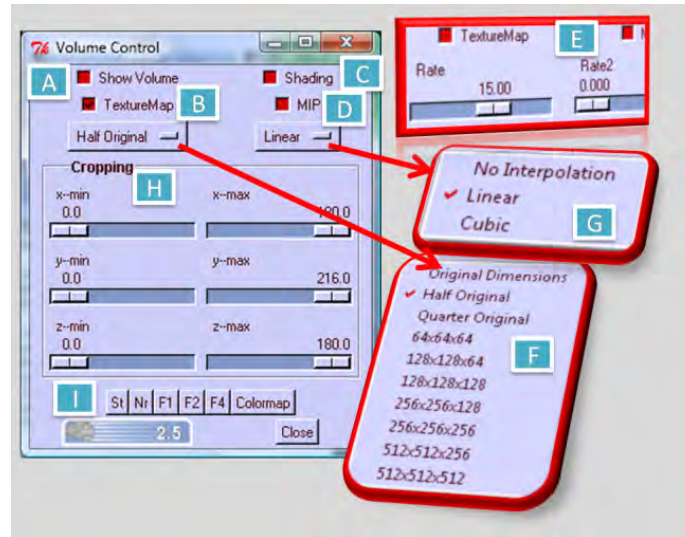


Figure 6.7: **Volume Rendering Controls..** **A:** Show volume, if unchecked the volume is not shown. **B:** Enable hardware accelerated texture mapping rendering. **C:** Enable shading, this emphasizes image edges.**D:** Enable MIP mode. When not using texture mapping the rendering is done “in software”. In this case the “Rate Controls” (**E:**) appear in the control and can be used to control the frame-rate for rendering (and hence the quality). “Rate 1” controls the rendering rate when moving the volume and “Rate 2” when the display is static. (0.0 is the best quality). The Reslicing controls **F:** and **G:** specify the image resolution to be used for rendering (and the interpolation mode to achieve this). A higher resolution results in better rendering quality at the expense of slower rendering. The Cropping controls **H:** can be used to shrink the volume. Finally the Colormap controls **I:** are used to control color mapping.

processes images only in dimensions of powers of 2, e.g. 64, 128, 256 etc. This means that a 65x65 image is as computationally expensive to render as an 128x128 image, since at each stage the rendering engine zero-pads the image size to the next power of 2! BioImage Suite reslices images prior to rendering to specific sizes to optimize this process. In particular note that cropping the volume (using the “Volume” control, shown in Figure 6.7 – **H**) can dramatically improve the quality of the rendering, as all the rendering pixels (e.g. 64x64) are allocated for the cropped portion as opposed to the whole image.

One more key point is to note that the volume rendering facilities have a separate colormap from the one used for slice display. Some synchronization is present though. For example, when preset colormaps are selected in the main viewer (e.g. St or Nr) their equivalents are also automatically applied for volume rendering.

## 6.4 Oblique Slices

While images are often looked at in orthogonal slices, sometimes there is great benefit in slicing the image in a direction that is not aligned with the acquisition axis (e.g. along the hippocampus). BioImage Suite has an oblique slice tool for doing this. The slice can be either manually positioned

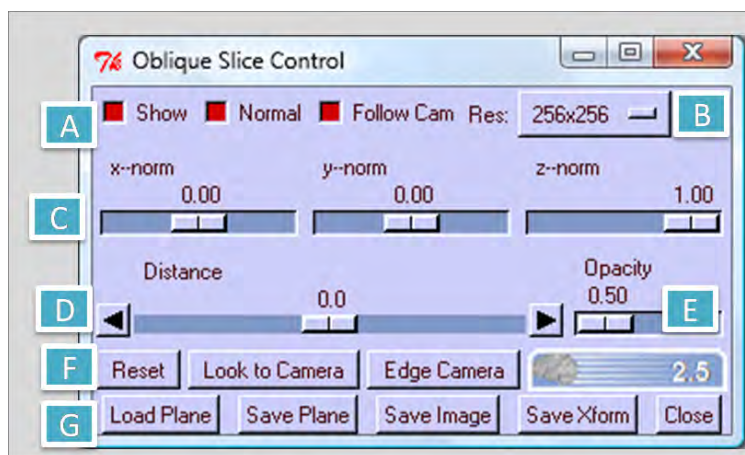


Figure 6.8: **The Oblique Slice Control.** Top Row (A): The “Show” checkbox determines whether the oblique slice is displayed. If the “Normal” checkbox is enabled the plane normal is also displayed. If the “Follow Cam” checkbox enables automatic positioning perpendicular to the camera. The “Res” option menu (B) determines the image quality. The manual positioning controls (C,D) can be used to position the oblique slice “by hand”. The opacity control (E) can be used to make this more or less transparent. The bottom row (G) has buttons for loading/saving the plane, the current 2D image slice (as an image) and the transformation (Save Xform). The later one can be used in BrainRegister to reslice the whole 3D image into this orientation. Additional facilities for manipulating the camera can be found in the row marked as (F) – these are discussed in the main text.

by specifying the normal direction and offset or by making it follow the camera position.

“Automatic” positioning works by placing the slice perpendicular to the current viewer angle, this is enabled/disabled using the **Follow Camera** checkbox, which adjusts the plane in real time to follow the viewer camera. Alternatively the **Look to Camera** button may be used to perform this once. The **Edge Camera** button places the slice at the front of the camera clipping range (see the *Renderer Controls* for an explanation.)

The image display is controlled by (i) the main colormap this is the same as for the main viewer (see Figure 2F), (ii) the resolution at which the slice samples the underlying image (this is set by the **Res:** drop menu 256x256 is the default), and (iii) the opacity which can make the oblique slice more or less transparent this is controlled by the **Opacity** slider.

## 6.5 The Animation Tool

The animation tool is divided into two tabs. The first part, “Main” is a simple automated JPEG grabber. Just as you can save a single snapshot of everything in the black box viewing area using File → Export, you can use the Animation Tool’s main function to save multiple JPEGs over a long amount of time with a given interval. To do this, select your delay time (or interval between snapshots) using the pulldown menu. Select your path using the “browse” button. You can edit the base name “grabbedname.jpg” to alter the resultant filenames (results will be e.g.

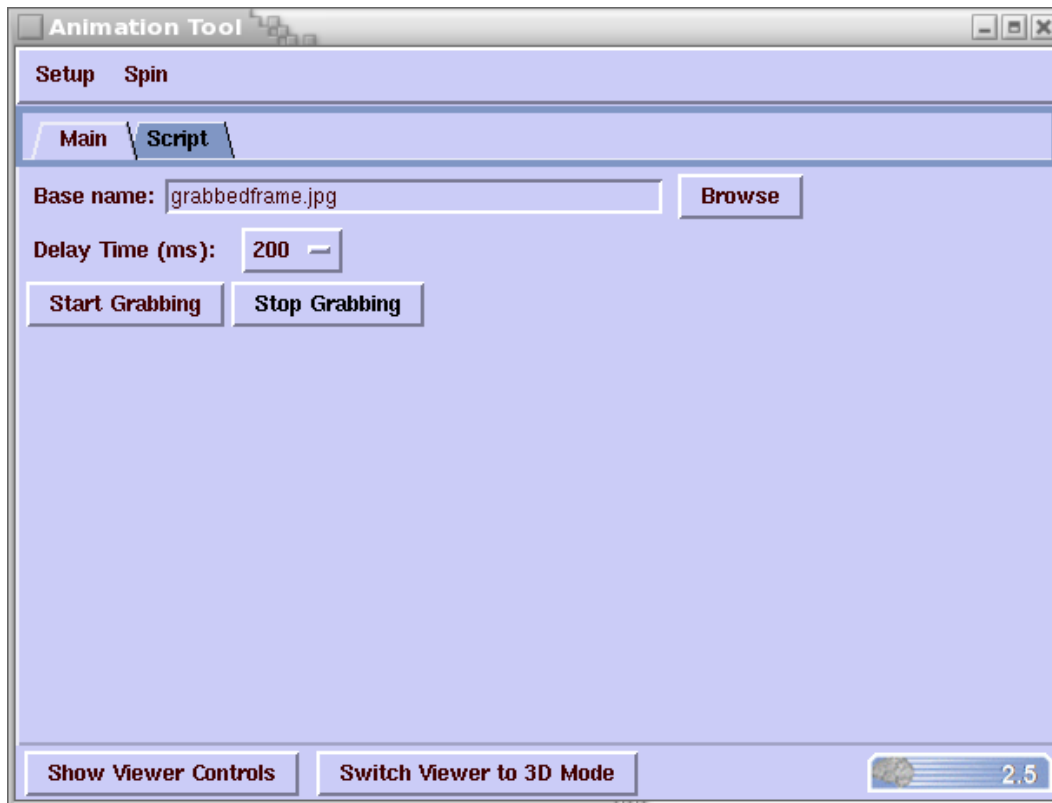


Figure 6.9: This image shows a snapshot of the animation tool which allows for automatic grabbing of frames from the viewer or manual grabbing of frames.

grabbedframe000.jpg as shown in Figure 6.9). After pressing “Start Grabbing,” you can manipulate the image as you desire. Then press “Stop Grabbing” to terminate the process.

(note that in order to get the 3D animations you may be interested in seeing, you must set the viewer to “3D only mode,” which can be done in the viewer or using the convenient button at the bottom of the Animation Tool)

### 6.5.1 The Script tab

Scripting is a more advanced method allowing for interpolation of animation between frames. This will save you the difficulty and time of having to drag the mouse at certain speeds and directions to obtain an adequate animation. Figure 6.10 shows a step by step description of the “Script” tab which allows for the creation of detailed animations.

1. The script. This is comprised of several reference frames provided by the user. The tool will create a sequential animation from the first frame through each subsequent frame based on the parameters provided by the user. You can select frames using the up/down buttons or by clicking on them, and delete them individually or completely as necessary.
2. A readout of coordinates and information about the current frame selected, including the

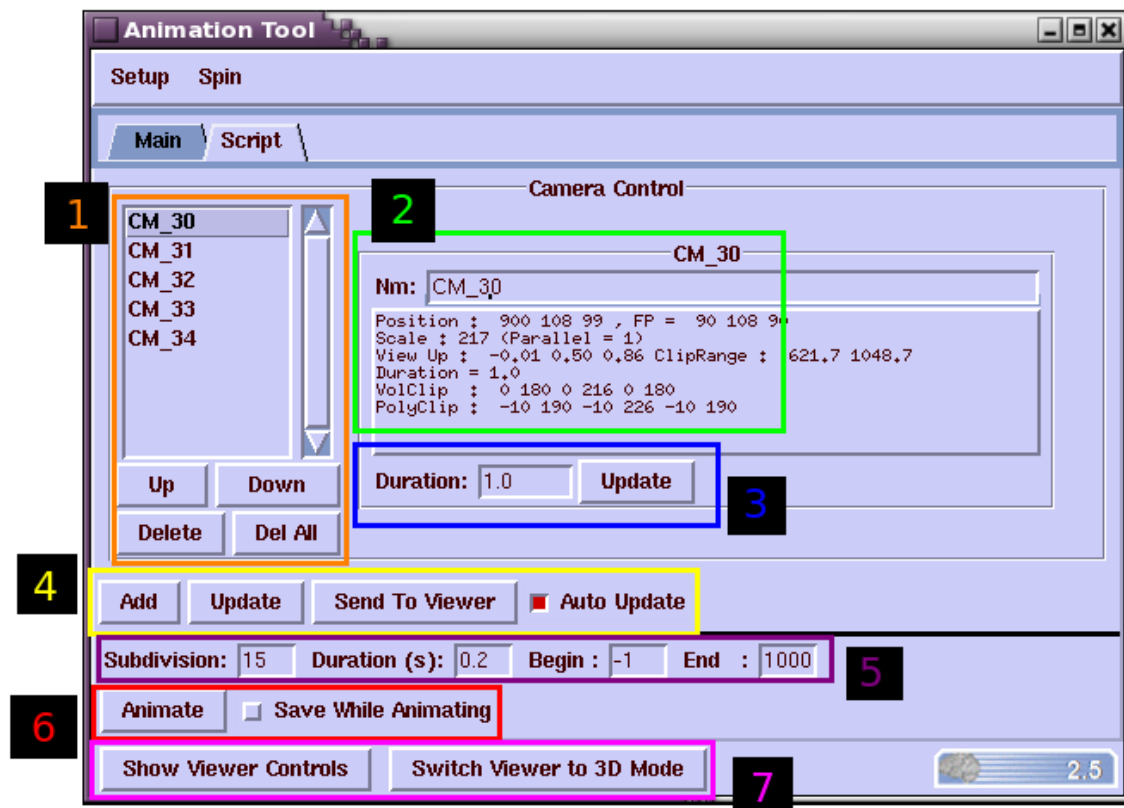


Figure 6.10: The Script tab. This tab allows for fine tuned controls for creating animations. The camera positions, duration at each camera position and the smoothness of the animation desired are some of the things that can be controlled using this tab. Section 6.5.1 has a more detail description for this image.



Figure 6.11: The Spin menu automatically computes and inserts camera positions that are 30 degrees apart. This prevents the need to accurately specify each and every position. Spin 10 inserts more camera positions which allows for smoother transitions between camera positions.

camera position and view.

3. A relative duration for the animation from the current to the subsequent frame for example, changing this value from 1.0 to 2.0 will make the animation take 2x the time, or operate at  $\frac{1}{2}$  speed.
4. Interface between the viewer and the script. “Add” takes the current view in the viewer and adds it at a frame (in this case the next frame would be CM.35). “Update” refreshes the currently selected frame in your script to represent whatever is currently in the viewer. “Send to viewer” refreshes the viewer to show the currently selected frame. “Auto update,” when selected, sends the frame to the viewer as soon as it is clicked (eliminating the need to click “send to viewer”).
5. Duration and subdivision factors for the entire animation (as opposed to the control in 3, which is for one step only). The subdivision controls the number of steps the animation takes to go from one frame to the next. Duration controls the length of the entire animation. Entering a low number of subdivisions and a length duration creates a “slide show” effect. Increasing the number of subdivisions creates a smoother, more detailed animation.
6. Animate shows what your current animation will look like based on the script and the parameters you have entered.
7. Calls viewer controls, or switches the viewer to 3D mode (most of the time the animation tool will be used to create 3D animations.). You may also want to use the “volume” drop-down menu in the viewer’s right pane to flesh out the image in 3D.

**Setup Menu** The setup menu allows you to load and save scripts for your convenience and exit the animation tool console when necessary.

**Spin Menu** The spin menu is located within the Animation Tool and can be accessed as a drop-down next to the Setup Menu.

- Spin 10 - automatically spins the model, horizontally, inserting 35 steps

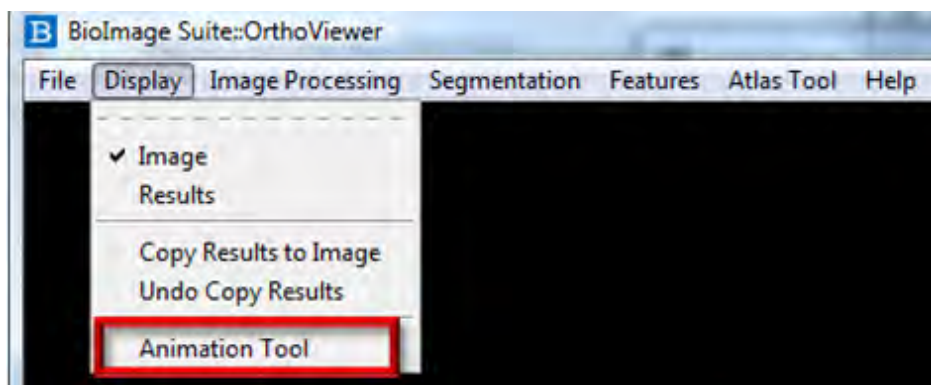


Figure 6.12: The animation tool can be invoked by clicking on the “Animation tool” button in the “Display” dropdown menu.

- Spin 20 - 18 steps
- Spin 30 - 12 steps
- Positive Rotation Adjusts between clockwise and counterclockwise spinning. The default, when positive rotation is checked, is counterclockwise.
- Azimuth adjusts between rotation about the z-axis and the xy-plane. The default is azimuthal spinning on the z-axis.

### 6.5.2 Making a simple animation

Start by loading the Orthogonal Viewer from the main menu. Once the viewer has loaded, choose an image from File–Sample Images. In this case, we’ll use the MNI T1 1mm stripped dataset.

Locate the Animation Tool from the Display dropdown menu as shown in Figure 6.12. The animation tool defaults to the “main” tab. Ensure that the save path is directed to an appropriate folder so that you can find the output files later. Experiment with this tab by setting the Delay time to 500 ms, and grabbing a few pictures with the “Start Grabbing” button while you rotate the image in the viewer. Click “Stop Grabbing” and examine the images in your file path. Move to the script tool using the tabs at the top of the Animation Tool window. Try making a simple script by clicking Add, then rotating the image slightly and clicking Add again. You may want to delete the original frame which will be called “none” as well. Once you have two or three points, click “animate” and view your animation in the window. Add a few more frames. Try doubling the default duration value for one of the frames to 2.0 and run the animation again.

Examine the Spin tab. You can either delete all of your first frames or add on to the animation. Use Spin 30 as it goes the quickest, and re-run your animation. Try changing the rotation direction and axis with the Azimuth and “Positive Rotation” options in the menu.

Once you have familiarized yourself with the options, you can choose to save your animation by clicking the “Save while Animating” tab next to the Animate button and run the animation again.

## Part II

# B. Anatomical Image Analysis

## Chapter 7

# The Image Processing and Histogram Tools

### 7.1 Introduction

Image Manipulation or Image Processing is the process via which we take one image as an input, perform some operation on it and then output a new image as an output. For our purposes an image is a large three-dimensional matrix of numbers where each cell in the matrix (or voxel element or voxel) also has “size” corresponding to the voxel size. There are a number of different categories of procedures – the following is an attempt at a crude taxonomy:

1. Intensity Manipulation – this type of operations take an image and processes each voxel separately based on simply its intensity. A simple example of this is image thresholding.
2. Regional Intensity Manipulation – this type of operations replaces the intensity at a voxel by some function of the intensities of a (mostly) small neighborhood of voxels around this. An example of this type of operation is Gaussian smoothing.
3. Image Resampling/Resizing – here the goal is to generate an image whose dimensions are different from the input image. For example, in image resampling our goal might be to produce an image of resolution 2x2x2 mm from an input image of resolution 1x1x1 mm.

Most BioImage Suite applications include an image processing option on their menu bars. This enables access to two tools, the Histogram Control and the Image Processing Utility Control which are described in this handout.

**Image Type – Not All Images are created equal:** While in memory, each image is stored as a large array of numbers. These arrays have a ‘type’ such as integer/float etc. The type of the image imposes some restrictions as to what operations can be done to it successfully. For example



if an image is of type ‘short’ it can only store integer values in the range -32768:32767. Consider the following two examples:

**Example 1:** Our “short” image has minimum intensity 0 and maximum intensity 4000. We cannot multiply this by a factor of 10 (e.g. under Math Operations) as the upper value will exceed 32767 ( $10 \times 4000 = 40000!$ ) and will result in weird and incorrect results.

**Example 2:** Our “short” image has minimum intensity 0 and maximum intensity 1 – this could be generated by a thresholding operation etc. This image can not be *usefully* smoothed with a Gaussian filter as the desired output would have values like 0.5 which can not be stored in a short image – as this can store only integers and decimals are ignored.

The most common types are:

1. Unsigned Char or Uchar – one byte per pixel, integers only, range 0:255.
2. Signed Char or char – one byte per pixel, integers only, range -128:127.
3. Unsigned Short or Ushort – two bytes per pixel, integers only, range 0:65535.
4. Signed Short or Short – two bytes per pixel, integers only, range -32768:32767.
5. Float – four bytes per pixel unlimited range, can store decimal points with a finite precision.
6. Double – eight bytes per pixel, same as float but with “double” precision.

## 7.2 “Image” and “Results”

All BioImage Suite viewer applications store at least two images in memory at the same time, labeled as “*Image*” and “*Results*”. This can be accessed under the Display menu. When an operation is performed on an image, the results are saved in the “*Results*” image of the viewer. This allows you to revert back to the original image, which still resides in “*Image*”. In order to perform further operations on a result image, you must choose (Display — Copy Results to Image) – see Figure 7.1 (left) . This will overwrite “*Image*” with “*Results*” display. This process can be undone, uncommitting any changes that have been made and reverting back to the image in memory before copying results, by using the (Display — Undo Copy Results) command. The top two commands in the “Display” menu, “*Image*” and “*Results*” simply select which image is shown. Choosing one does not delete the other – the other image is still there. It can be accessed by switching back. It is important to note that most of the operations provided by the Image Processing toolbox and other analysis tools take the “*Image*” as an input and send their output to the “*Results*”. In order to work with them further, the results must be copied to “*Image*”.

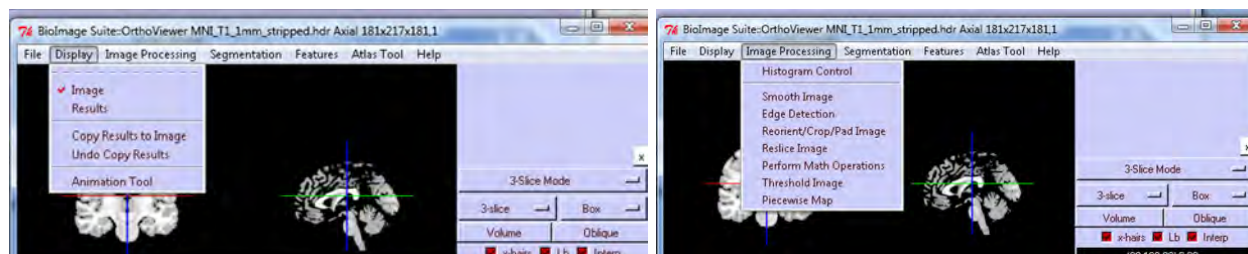


Figure 7.1: **Left:** The Display menu. **Right:** The Image Processing Menu.

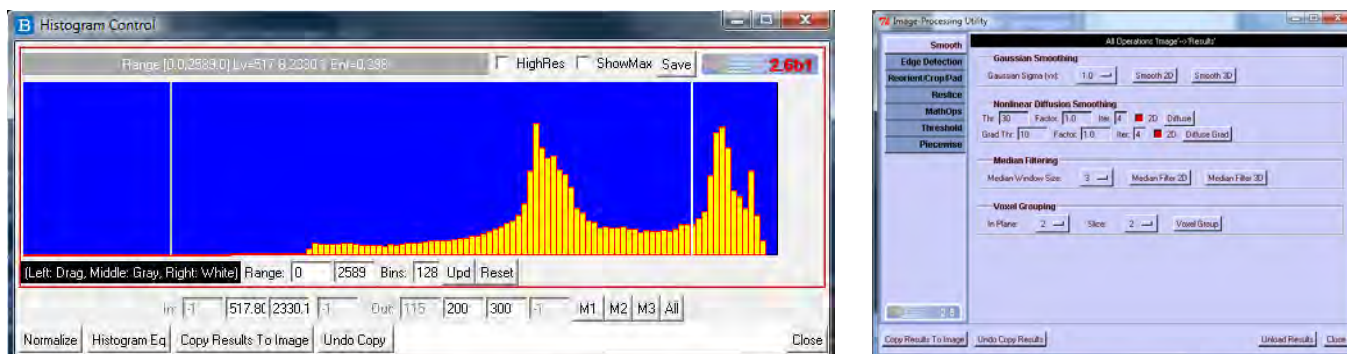


Figure 7.2: The Histogram Control Window (left), and the Image Processing Control Window (right).

### 7.3 Histogram Control

The Image Histogram control (shown in Figure 7.2) allows you to see how intensity values are distributed in your image. It can be accessed by choosing the (Image Processing — Histogram Control) menu option in the viewer window (Figure 1 - Right image). This brings up the histogram control window, which shows the intensities in the image distributed among a set of discrete bins.

**Intensity Range:** You can edit the range of intensities displayed in the image (and thus, displayed in the histogram control). Simply edit the values in the “Range” fields, and click **Update**. Hitting **Reset** resets the range to show the entire range of intensity bins in the image.

**Normalize** This performs a specialized intensity normalization function – this functionality is mostly obsolete now, see the Image Processing/Piecewise Map section for a replacement.

**Histogram Equalize:** Histogram Equalization attempts to flatten the histogram (i.e., distribute intensities evenly across a range), thus providing good contrast in the image. Clicking the **Histogram Eq** button redistributes the intensities such that all bins below the grey cursor become zero, and all bins above the white cursor saturate to the value at the white cursor. The intensity distribution for the region between the cursors is flattened.



Figure 7.3: The “Show Max” box is checked which shows the tallest bin. In this case, the tallest bin is the background. Figure 7.2 shows a histogram for the same image with the Show Max checkbox unchecked.

**High Res** If the “High Res” checkbox is disabled, disabled the histogram is obtained by sampling the image at a lower resolution to make the process faster (a reduction of a factor of 4 in each dimension). The shape of the histogram is approximately correct but the bin counts are roughly a factor of 64 off).

**Show Max** The “Show Max” checkbox allows the user to toggle the functionality of showing or hiding the ‘tallest’ bin. In some cases, the tallest bin is the background and ends up obscuring the interesting part. Figure 7.3 shows an example of the “Show Max” checkbox being checked for the same histogram as Figure 7.2.

**Save** The “Save” button saves the histogram as a text file.

### Mouse Control in the Histogram Control Window

**Left button:** Use the left mouse button to click on and drag either the grey or white bar to the desired location. Also, clicking and holding with the left button anywhere on the histogram changes the top display bar to show that bin and its value.

**Middle button:** Clicking on the histogram with the middle button places the grey bar at the clicked location.

**Right button:** The right mouse button places the white bar. Right-click at any location in the histogram control window to make the white bar jump there.

## 7.4 The Image Processing Control

This is a complex multi-tab control which encompasses a variety of image processing functionality. This control, which is shown in Figure 7.2 allows the user to quickly navigate to the various image processing functions available using the tabs on the left. These match the choices in the Image Processing menu. At the bottom of this window are a few buttons that commit the results of the operations to the image:

**Copy Results to Image:** This button is equivalent to the (Display — Copy Results to Image) menu command (see Displaying and Working with Results). You must click it after any operation performed in the Image Processing Utility window if you want the results to be permanent and able to be operated on by subsequent processing steps. **Undo Copy Results:** Undoes the copying of the “Results” to “Image” – uncommits results. **Unload Results** - Removes all images from memory (from this control) to save memory. **Close** - Closes the Image Processing Utility window. We describe each of the tabs next.

### 7.4.1 The “Smooth” Tab

**Gaussian Smoothing:** Image smoothing operations typically replace the intensity value at a voxel with a new value that is a function of the intensity of this voxel and its neighboring voxels. The most common operation which is “Gaussian smoothing” replaces the intensity at a voxel with a weighted sum of it and its neighbors where the weighting is defined using a Gaussian function (or kernel). The degree of the smoothing is directly dependent on the shape of this kernel, which in turn is function of the standard deviation or ( $\sigma$ ) used.

The value of sigma is often specified either in voxels (which is what is used in the BioImage Suite GUI), or using the so-called “Full-Width at Half-Maximum” (FWHM) specification. As stated in a Wikipedia page “A full width at half maximum (FWHM) is an expression of the extent of a function, given by the difference between the two extreme values of the independent variable at which the dependent variable is equal to half of its maximum value.” Given a FWHM of  $w$  we can obtain the equivalent Gaussian  $\sigma$  using the relationship:  $\sigma = w/2.355$ . Additional scaling may be needed if one is in  $mm$  and the other in voxels.

In BioImage suite one can perform Gaussian smoothing by (i) setting the filter size  $\sigma$  in voxels and pressing either “Smooth 2D” for 2D smoothing or “Smooth 3D” for full 3D smoothing of the image. Use “Smooth 3D” unless you have a really good reason not to.

**Diffusion and Diffusion Gradient Smoothing** Gaussian smoothing blurs across edges. Non-linear smoothing techniques were developed to avoid this. The algorithms available in BioImage Suite (based on original work by Perona and Malik) attempt to perform smoothing only within relatively homogeneous areas – and suspend smoothing if the intensities (or intensity gradients) of two adjacent pixels differ by an amount greater than the threshold.

**Diffuse** This algorithm takes into account the raw intensities of pixels in evaluating the threshold

requirement for smoothing. Thus, only the **Thr** field and **Factor** field must be specified. Click the **Diffuse** button to make the magic happen. **DiffuseG** To use the Diffusion Gradient smoothing algorithm, set the **Grad Thr** field along with the **Factor** and click the **DiffuseG** button. This will only smooth across pixels that *do not* have an intensity gradient difference greater than the threshold.

**Median Filtering:** Median filtering operates by a similar mechanism to Gaussian smoothing, but instead of using an *average* of the surrounding voxels to generate the new smoothed voxel value, it uses the *median* value of all the pixels in the median window radius. Thus, to use the Median Filter tool, simply set the window size (“Median Window Radius”) and click either **Median Filter 2D** to compute medians using only in-plane voxels, or **Median Filter 3D** to compute medians using a 3-dimensional window that includes voxels in adjacent slices. Median filtering is useful for removing what is known as “salt and pepper” noise which is uncorrelated one/of noise specs on the image.

**Voxel Grouping:** Voxel Grouping combines groups of voxels together, yielding a larger voxel with a single value. This simulates a lower acquisition resolution for the image, and is useful for comparing high-res images with lower-res images. The “In Plane” option sets the number of pixels to group into a single pixel within a slice, while the “Slice” option dictates the number of slices across which grouping will be performed. After setting these values, click **Voxel Group!** to do the grouping.

## 7.4.2 Edge Detection

The Edge detection tab has facilities for computing edgemaps of the image. We use a simplified form (single-scale) of the famous Canny edge detector. In essence this first computes the gradient of the image at a certain scale (set by the sigma variable) and additionally performs non-maximal suppression to thin the edges.

## 7.4.3 Cropping and Reorienting

**Reorienting an Image: The Axes:** Reorienting an image is, essentially, choosing along which pair of axes the planes of the slices will lie. Thus, in BioImage Suite, if you define your image as “Axial”, the ij plane will be in the axial slices. Likewise, choosing “Coronal” defines the coronal plane as the ij plane. This applies to sagittal as well. Thus, if your Image has the i and j axes (red and green lines) in the coronal plane, and you would like them to be in the axial plane, simply go to the “Reorient/Crop” tab or choose the (Image Processing — Reorient/Crop) choice from the menu, and click the “Axial” radio button. The image axes will be redefined.

**Relabel:** Clicking the **Relabel!** button does not affect the image, except to alter the image header so that the views are labeled correctly. Thus, if you have an image in which, for whatever

reason, the view that is actually coronal is labeled as “Axial” in the viewer, simply change the label settings using this feature.

**Cropping an Image:** Cropping an image comprises removal of peripheral pixels along any of the three axes. This reduces the size of the image, significantly reducing computing time for subsequent image calculations, especially registration. Thus, it is advantageous to remove any excessive blank space from the edges of your image before performing other operations.

To crop an image to your specifications, first determine the boundaries of the region you want to retain, using the cursors in the viewer (See “Navigation controls in the viewer control panel” in Working with the Orthogonal Viewer Display). Then, input the max and min values for X, Y, and Z (you may choose to use all of these, two, or just one) using the “Range” fields. The t parameter for cropping is used to crop images with a time dimension. Thus, if you choose a t range, only the images from within this set of the time series will be included in the result. Now, click the **Crop** button, and the image dimensions will change to fit the ranges you have specified. The lower bound in each dimension will become zero, and all data below it will be cut out, while the upper bound will become the maximum in that dimension of the image (the new maximum will be equal to the old maximum minus the summed amount cropped from both sides).

The “Rate” field specifies a sampling rate that can be applied to the image simultaneously to the crop operation. Thus, if you put two into the Rate field for X, the resulting image will be resampled using only every other voxel in the X direction, lowering its resolution in this dimension, but leaving the others unchanged.

**Blank** operates almost the same as **Crop**, except it does not change the image dimensions. Instead of deleting the voxels outside the range altogether, it simply sets their value to zero.

The **AutoCrop** button can be used to automatically perform a crop of the image where all the region to be cropped has intensity equal to zero.

#### 7.4.4 Reslice

Reslicing reevaluates the image as individual voxels and redefines the slices, effectively applying an affine transformation to the image. Thus, with this tool you can flip and rotate your image in all three dimensions.

**Transformation:** Applying transformations to the image is very easy. Simply choose the flip or rotation operation you would like to apply from the “Transform:” menu:

**Identity** – the identity transformation does not change the image. **Flip Slice Order(Z)** – the slice order is reversed, flips in the Z-axis. **FlipX** – flip image in the X-axis. **FlipY** – flip image in the Y-axis. **Transpose** – performs matrix transposition on the image; that is, permutes X and Y:  $\{X = Y \text{ and } Y = X\}$ . Essentially switches the X and Y dimensions. **Flip Transpose** – Transposes as above, but inverts as well:  $\{X = -Y \text{ and } Y = -X\}$ . Essentially switches X and

Y as well as performing a flip. **Custom Rotation** – Enables the RotX, RotY, and RotZ slider bars, allowing you to rotate the image any specified amount in any dimension. After selecting a transformation (and setting the rotation sliders if “Custom Rotation” is selected, hit the **Reslice!** button to perform the operation. The output will be shown in the viewer (stored in the results display).

**Sampling:** The sampling factor determines the voxel size on output. Thus, a factor of x2.0 results in an image that has voxels twice the size of the original. The overall image dimensions are unchanged, the resolution is simply changed by the sampling factor. A sampling factor < 1.0 will result in an image that has higher data density than the original, leading to increased computing times, while sampling factors > 1.0 will lower density, lowering subsequent computing time requirements.

Use the “Sample factor:” menu to choose a rate at which to sample the image. The entries preceded by “x” (e.g. “x0.5”) use a scaled voxel size based on the current voxel size, while those without the “x” (e.g. “1.0”) are absolute sampling rates. These are isotropic sampling parameters; if you want to sample at different rates in different directions, select the menu entry “Custom”, and set the sampling rate for each dimension individually using the X,Y, and Z sliders.

### 7.4.5 Thresholding Images

Thresholding images is essentially a process where the value of the intensity outside a certain allowed range is mapped to zero. The value in the allowed range is either left untouched or set to a fixed value (binary output). BioImage Suite allows two types of thresholding: (i) by range (setting the upper and lower value of the allowed range) and (ii) by list – which specifies a list of allowed values. For a description of all the functionality see Figures 7.4 and 7.5.

### 7.4.6 Piecewise Map

The piecewise map utility allows you to specify a piecewise linear function that maps input voxel intensities to new output intensity values. Thus, you can create a custom mapping function to generate many different types of output images. This essentially takes a set of input/output pairs and performs linear interpolation between them. In order to view the equation as it is being generated, click the **Show Equation** button, which will bring up a console showing the piecewise mapping function as defined by the input and output fields.

An example is shown in figure 7.6.

**Output Type:** The output type is the data type to be used to store the output values. The default is “Same as Input”, which is self-explanatory. The other choices are arranged in order of increasing precision.

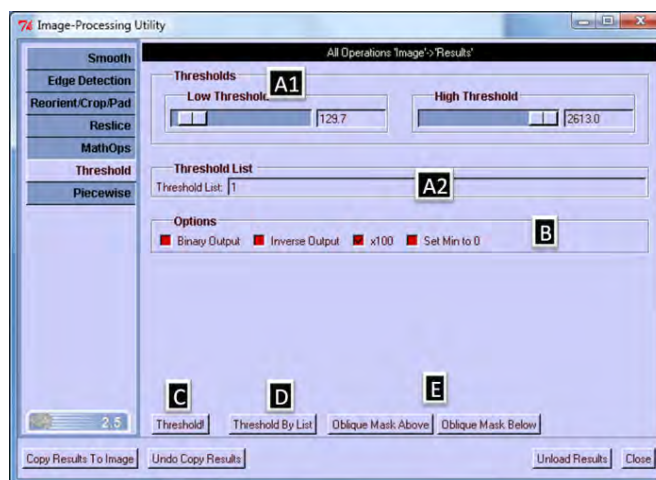


Figure 7.4: The Threshold Image Control. The thresholding range can be set using the two sliders (**A1**) or the entry box (**A2**). The row marked **B** has controls for specifying whether the output is binary or not (and whether to use 1 or 100 as the output of a binary thresholding – this is controlled by the “x100” checkbox). The “Inverse Output” checkbox determines whether the allowed range is the once specified or its compliment. Thresholding via range is done by pressing the “Threshold” (**C**) button, whereas thresholding via a list (specified in **A2**) is performed by pressing the “Threshold By List” (**D**) button.



Figure 7.5: BioImage Suite can be used to blank out regions of the image as shown above. These are specified using the current position of the oblique slice. This is accomplished by (i) positioning the oblique slice and (ii) pressing one of the two buttons “Oblique mask Above” or “Oblique Mask Below” in the threshold control – these are marked as **E** in Figure 7.4..



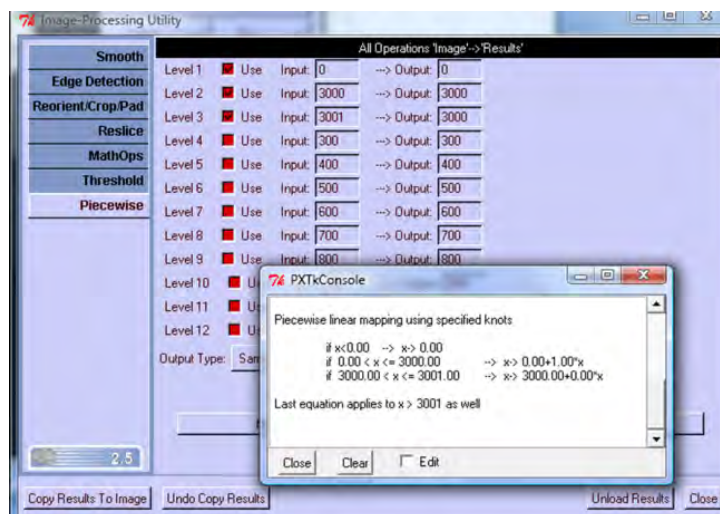


Figure 7.6: An example of using the Piecewise linear mapping tool to adjust image intensities. In this case, an image is remapped such that voxels with intensity below 0 are kept at 0, voxels between 0 and 3000 are unchanged and any voxel with intensity greater than 3000 is mapped to 3000. The popup (PXTkConsole) shows the result of pressing the “Show Equation” button, which describes the function used to remap the image intensities.

Click the **Multi Linear Operation** button to perform the piecewise mapping operation on the image. As usual, the results will be placed in the “Results” image of the viewer, and must be copied to the “Image” image before any further processing can be done.

## 7.5 EXAMPLE: Reorientation of Images

Many 3D whole brain acquisitions are taken in the sagittal orientation. If your reference brain has a different orientation (i.e. the MNI Brain is in the axial orientation) than your single subject brain, you must Reorient the single subject brain to match the orientation of your reference brain. The non-linear registration described later cannot be run between two different orientation types.

The step-by step procedure, with reference to Figure 7.7, is as follows:

1. Check the header of your window, it will list the following information about your image: Viewer being used, the name of the file that was loaded, the orientation and the image size. In this case the orientation begins as Sagittal.
2. Choose (Image Processing — Reorient/Crop Image)
3. A new Image Processing Utility window will appear. Choose proper orientation (axial if using the Colin Brain). Click Reorient
4. Click Copy Results to Image
5. Save File (you cannot continue onto the Brain Extraction unless you save the current image). Click (File — Save)

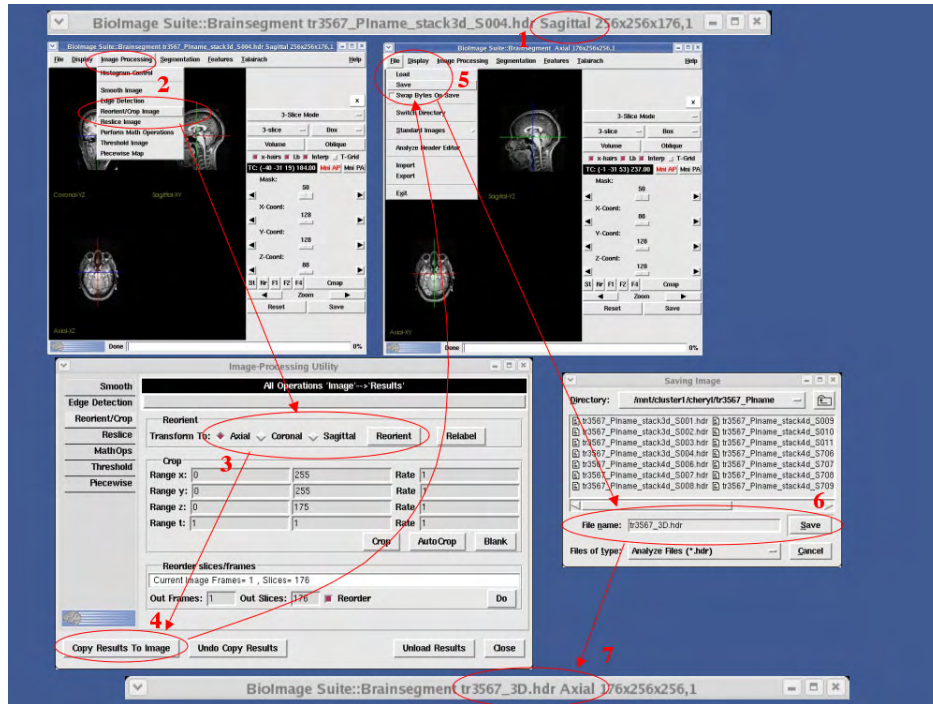


Figure 7.7: Methods used to reorient an image using the Image Processing Utility.

6. A new window will appear where you can type in your new filename and click Save.
7. Once saved, the header of your window will display the new filename as well as the new orientation.

## Chapter 8

# The Interactive Segmentation Tools

### 8.1 Introduction

Image Analysis is the process of extracting quantitative information from images. Unlike image processing, this information is not an image per se, but rather a description of some aspect of the image. Examples of such information include the volume of gray matter in the brain, the volume of the left hippocampus, the volume of fat in the abdomen, the surface area of a given sulcus etc. Segmentation of extracting a structure or group of voxels from an image. It usually means one of two things:

**Voxel Classification:** The main applications of these techniques is the labeling of each voxel in the image according to tissue classes. Typical applications of these methods are the classification of brain tissue into gray matter, white matter and cerebrospinal fluid (CSF) in neuroimaging and fat/lean muscle classification in abdominal imaging. The output of this type of method is an image, or an objectmap, with labels (e.g. 0,1,2,3) each representing a different type of tissue.

**Deformable Surface Segmentation:** Deformable surfaces either using explicit parameterizations or levelset formulations are applicable to all parts of the body as a method for measuring the shape and volume of different organs and sub-cortical brain structures. In addition to automated methods, there are also applications of manual tracing and semi-automated methods in many practical applications both for complete segmentation or for the review/correction by an expert user of the results of automated segmentation such as in the BioImage Suite surface editor tool. The output of these types of methods are surfaces which bound structures of interest, e.g. the surface of the hippocampus.

There is a large degree of overlap between these two methods. For example, structures such as the hippocampus can be segmented either by painting all voxels belonging to the interior of the hippocampus with a certain color – this is the voxel classification approach, or by defining the outer surface that encompasses this structure. We can also often, with some loss of resolution, convert from one description (e.g. the surface) to another (e.g. the objectmap image).

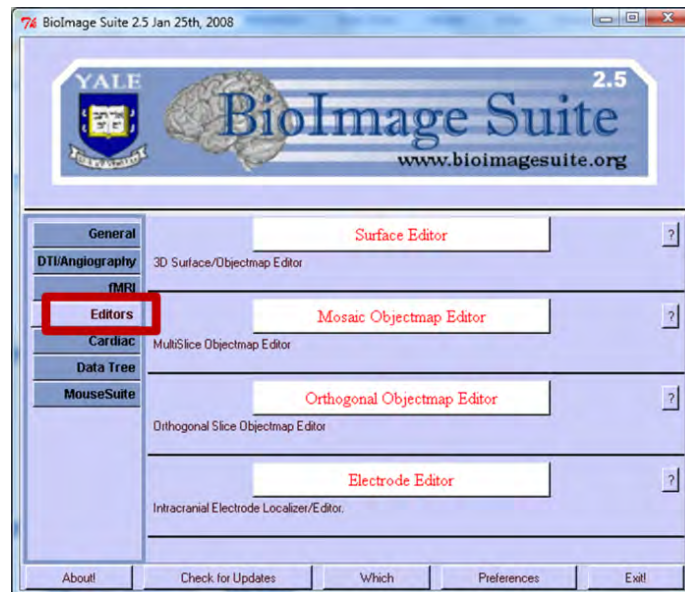


Figure 8.1: The Editor Tools.

**Objectmaps:** In BioImage Suite we will use the word objectmap to mean an image which has a finite number of intensities (e.g. 1-10) where each intensity value represents a unique object or tissue class. In other words, Objectmaps are essentially images (and are stored in the same file format as regular images) whose intensities have special meaning.

**Surfaces:** BioImage Suite stores surfaces using two major file formats. The most basic one is the polygonal surface (stored in files with a .vtk extension) which consists of a set of points and a set of connections between the points resulting, most commonly, in a set of connected triangular patches that form the surface. While this convention is very useful for visualization and post-processing, it is less convenient for editing. For this reason, the surface editor tools store the surfaces in .sur files where each surface consists of a set of planar curves, and each curve is represented by a b-spline. The nodes of the spline are essentially interpolated using piecewise cubic polynomials to yield a smooth curve.

The editor applications provide powerful functionality for interactive segmentation of different anatomical structures. There are two distinct modes of operation. The surface tools allow the interactive outlining of surfaces which are represented as “stacks” of curves, one curve per image slice. The “Editor” menu is shown in figure 8.1.

BioImage Suite has four editor tools. The Mosaic Objectmap Editor and the Orthogonal Objectmap Editor allow only for the editing of objectmaps. The Surface Editor allows for both objectmap editing as well as surface editing. The 4D Surface Editor (under the cardiac menu) allows for only the editing of surfaces on temporal sequences of images.

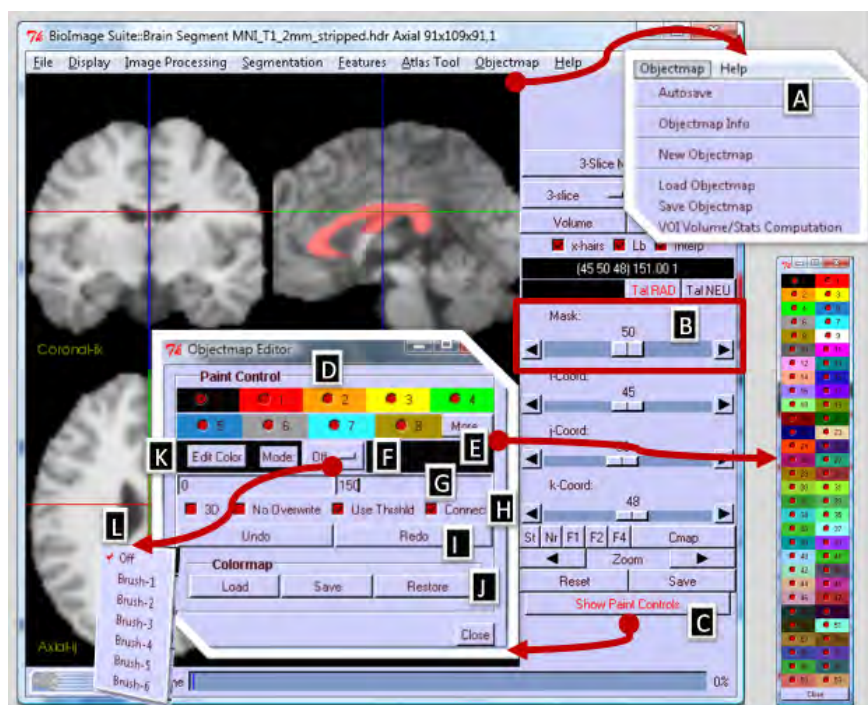


Figure 8.2: . The Orthogonal Objectmap Editor Tool.

## 8.2 The Objectmap Editor Tools

There are two tools here, one which allows for painting on orthogonal slices and the other on multiple parallel slices (this is slow!). We describe here the functionality of the Orthogonal Objectmap Editor (shown in Figure 8.2 – the use of the Mosaic Objectmap Editor is similar).

**The Objectmap Menu (A):** The objectmap (which is an image) can be loaded/saved/reinitialized using functionality under the Objectmap Menu. In particular note that the option “VOI Volume/Stats Computation” can be used to perform VOI analysis of the underlying image using the objectmap to define the VOIs! If Autosave is enabled, the objectmap is automatically saved periodically.

**The Mask Slider (B):** The objectmap is shown as a transparent overlay over the original image. The degree of the transparency is controlled using the mask slider (0=completely transparent, 100=opaque).

**Objectmap Editor:** The Objectmap Editor popup is invoked using the “Show Paint Controls” button (C). This has functionality for painting over the image.

**D: Paint Control:** Use the buttons 0-8 to select the color to paint with. If more colors are needed using the button marked “More” (E) to bring up an auxiliary control with 60 possible choices. The color definitions are controlled via the colormap which can be Loaded/Saved using functionality in the box marked as J. Individual colors can be edited using the “Edit Color” button (K).

The Painting mode is set via a number of controls. the Mode drop menu (**G**  $\leftrightarrow$  **L**) which selects the brush size (Off disables painting). If the Mode menu is set to “Off” then the left mouse button is use to navigate, otherwise it is used to both paint and navigate.

The four checkbuttons in the row marked as **H** control for (i) whether the brush is two or three dimensional (3D), whether the paintbrush will overwrite any other existing non-zero label (No-Overwrite) and whether the painting will respect thresholding and connectivity constraints. If the ”Use Thrshld” check button is enabled only voxels in the range specified in the threshold entries (marked as **G**) will be painted over. The “Connect” checkbox can be used to restrict the painting to voxels which are connected with the center voxel.

The Undo/Redo buttons **I** can be used to correct for mistakes during the painting. The unit of undo/redo is each mouse paint operation (from mouse press to mouse release).

**Note:** The objectmaps are not displayed in Oblique or Volume Rendering modes.

**ObjectMap Regularization** The objectmap regularization/smoothing in BioImage Suite can be manually performed using the functionality in the objectmap editor. Figure 8.3 shows a screenshot of the objectmap editor and the parameters required to smooth an objectmap. The “Factor” and “Conv” settings are parameters that control the smoothing kernel. On completion, a popup dialog box informs the user about the percent of voxel that were changed in the smoothing process.

## 8.3 The Surface Editor

The Surface Editor is the original BioImage Suite application. It was originally developed beginning in 1995 specifically for the segmentation of the bounding surfaces of the left ventricle of the heart (endocardium and epicardium) using then Silicon Graphics exclusive software libraries (Open Inventor). It has since been used for a number of applications including segmentation of neuroanatomical structure, region of interest definition of functional neuroimaging studies, cardiac segmentation, abdominal structure segmentation – including fat quantification – etc. The rest of this handout describes the core functionality of the surface editor tools. There are two related tools: the surface editor tool which includes both surface and objectmap tools for 3D images, and the 4D surfaceeditor tool which has only the surface editing tools but includes additional functionality for cardiac image analysis. Most of this handout is concerned with the 3D version of the tools.

### 8.3.1 Key Concepts

The SurfaceEditor tools operate and edit in 2D display in 3D paradigm. Editing is performed in a slice-by-slice mode using the slice editor tool and the 3D overall picture (i.e. 3D surface/3D objectmap) is displayed in the main Surface Editor window. The main window is updated either manually using the “Update” button in the editor or automatically when the slice is changed if the “Auto Upd” checkbutton is in the “on” state. The Surface Editor window consists of the same core



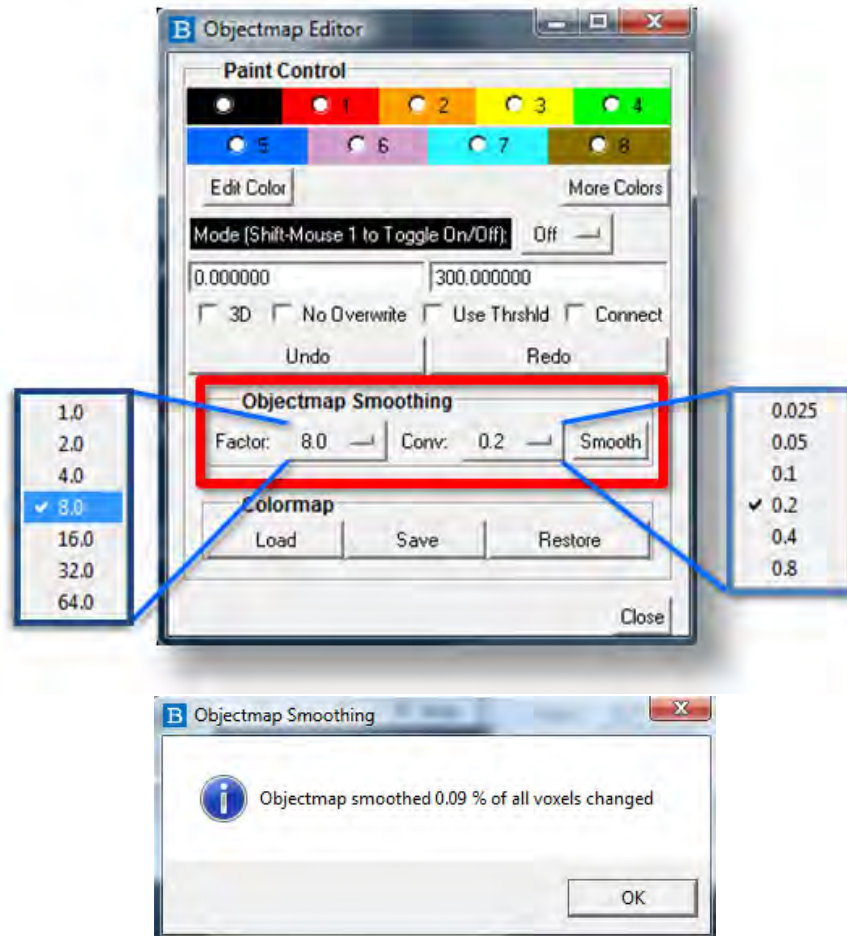


Figure 8.3: The Objectmap smoothing functionality in the objectmap editor. The bottom image shows a screenshot of the dialog box that pops up after the objectmap smoothing is complete. .

BioImage Suite orthogonal viewer described in the viewer pages, but has additional functionality related to the interactive segmentation tasks that will be described next.

**Note that editing in the slice editor is performed on the XY slice** of the underlying image, i.e. the axial slice if the image was acquired in an axial orientation, or the coronal slice if it was acquired in a coronal orientation. If a different 2D slice orientation is preferred, the image will need to be re-oriented or resliced (if oblique slices are required) using the Image Processing tool. The same applies to painting.

The Surface Editor has the ability to create surfaces from binary images. In this way segmentations generated using other tools such as itk-SNAP, which are typically stored as binary images, can be imported into the Surface Editor for more detailed manipulation. See the “Creating Surfaces from Binary Images” for more details.

### 8.3.2 The Main Surface Editor Window

The main Surface Editor window is shown to left in Figure 8.4. There are 5 parts to the user interface.

**A: The Menu Bar:** This contains the usual options of loading/saving the underlying images, image processing for smoothing, reslicing/cropping images, etc. In addition, the *objectmap* menu has options specific to the surface editor. *Please note that the File Menu only contains operations for loading/saving the underlying image and not the surfaces/objectmaps.* Separate functionality is provided to do this.

**B. The Tab Selection:** This selects which of three sets of controls appears on the right most pane of the window. These are the **Image** tab which has the usual orthogonal viewer options for manipulating how the image is displayed (e.g. single slice, orthogonal slices, volume rendering etc.), the **Surface** tab which contains most core surface operation and the **Surface+** tab contains additional surface operations. The “Surface” tab (shown) is divided into two parts, namely:

**C: Current-stack properties:** Options for manipulating the current surface (how to display this e.g. open surface and fast surface) and the all-important definition of surface extent, i.e. on which slices does the structure appear. This is the first thing that needs to be set when creating a new surface.

**D: Current-Stack Options:** These are options for resampling the current stack, flipping the x-axis, changing the color and loading and saving. On the bottom of the window

**E: Editor Slice:** contains functionality for selecting the current slice on which the spline/objectmap editor operates and a “button” for opening the Editor.

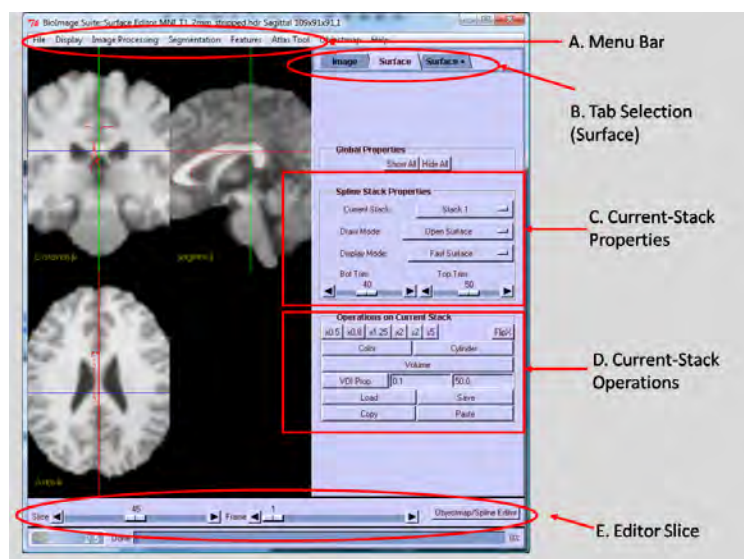


Figure 8.4: The Surface Editor. The main surface editor application.



## The “Image” “Surface” and “Surface+” tabs in more detail

Much of the functionality of the surface editor is bound within the notebook tabs in the rightmost pane. The **Image** tab is identical to the standard orthogonal viewer.

The **Surface** tab contains most of the functionality needed for manipulating surfaces in a global sense. The surface editor can be used to outline a number of surfaces at a time (typically 4; but can be set to as high as 50 using the user preferences editor). These can all be shown or hid using the Show All/Hide All buttons in the “Global Properties” frame at the top.

The “Spline Stack Properties” frames has the following options. First the “Current Stack” option menu enables the selection of the current surface (e.g. 1 to 4) for editing/manipulation. All subsequent operations operate on this surface. The “Draw Mode” option has mostly cosmetic effect. Typically the surfaces are cylinder-like with open tops and bottoms, i.e. “Open Surface”. The other options in this drop-menu enable the visual closing of either the top or the bottom or both ends of the surface. This adds computational cost to the surface updating and is best used sparingly. The “Display Mode” option has three choices “Fast Surface” which produces a fast but relatively low quality surface rendering, “Full Surface” which produces a slower but higher quality rendering and “Hide” which completely hides the surface. The “Bot Trim:” and “Top Trim:” controls enable the user to set *the critical bottom slice and top slice extends of the surface*. If the structure of interest appears only in e.g. slices 24-32, then bottom trim should be set to 24 and top trim to 32. Failure to do this will result in either additional or missing curves in the surface with erroneous results.

The “Operations on Current Stack Frame” contains options for global manipulation of the surface. The first line consists of a row of buttons that resample the surface. For example, “x0.5” resamples all curves to have half the number of control points (as discussed later each curve is parameterized as a b-spline), and similarly x0.8, x1.25 and x2. The “s2” and “s5” buttons re-parameterize the surface to have different degrees of smoothness and result in non-uniform b-spline. Finally the “FlipX” button performs a left-right mirroring of the surface which is useful in some applications. The “Color” button brings up a dialog box for selecting the color of the surface. Clicking the “Cylinder” button resets the surface to a cylinder of default location and size. The “Volume” button computes the volume of the surface by adding up the areas of the individual curves and multiplying by the image slice thickness. The VOI Prop tab performs VOI (volume-of-interest) analysis of the area inside the surface. The two thresholds can be set to restrict the analysis to only voxels in that range. The “Load” and “Save” buttons enable the loading and saving of surfaces in the custom “.sur” format. Finally the Copy and Paste options enable the copying of the surface to the clipboards and its pasting (presumably to another surface).

The **Surface+** tab contains some additional functionality. The “Current Stack” drop menu mirrors the one in the “Spline Stack Properties” frame and can be used to set the current surface. The **Load Set** and **Save Set** options can be used to load/save all the surfaces in the editor in a composite “.multisur” format file. (Note that the .multisur file simply contains pointers to an additional number of individual surface files. E.g. if one saved a group of 4 surfaces to mysurfaces.multisur, the following files will be created: mysurfaces.multisur, mysurfaces.multisur\_1.sur, mysurfaces.multisur\_2.sur, mysurfaces.multisur\_3.sur and mysurfaces.multisur\_4.sur). The **Autosave** and **Backup** checkbuttons if enabled turn on the autosave features of the software (leave them on!). The **Shift and**

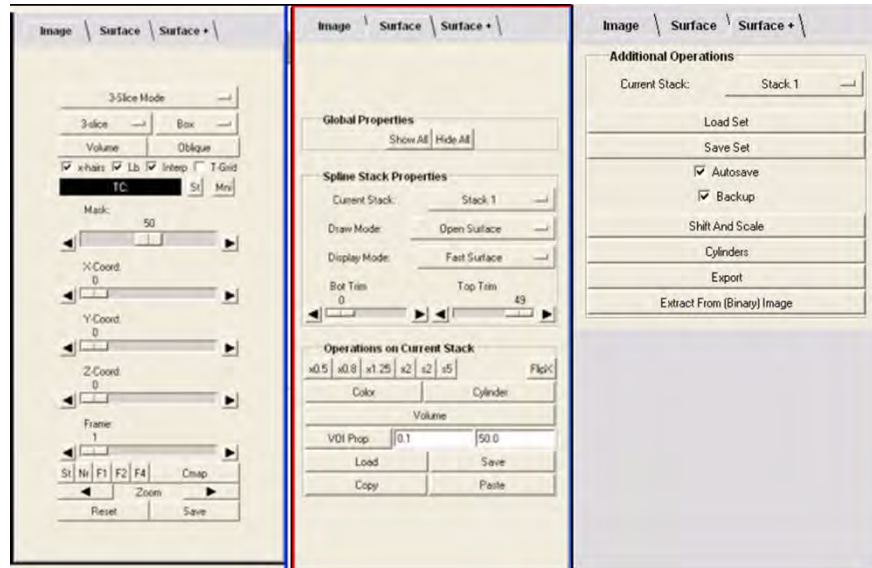


Figure 8.5: The three tab views available in the surface editor. These are all displayed one at a time on the right hand side of the main window, using the tabs at the top to toggle between them.

**Scale** button brings up a legacy control for re-scaling surfaces generated with older versions of Surface Editor which did not store proper slice thickness information. The **Cylinders** button brings up a control for creating a custom cylinder by specifying both radii, slice extents, curve centroids and number of control points. The **Export** buttons brings up a control for exporting surfaces either in the custom “.tstack” format or in the more common “.vtk” format (as defined by vtkPolyDataReader/Writer). The user can specify the sampling of the spline surfaces in mm as well as the format and a number of smoothing iterations to be applied prior to exporting.

**Creating Surfaces from Binary Images** A recent addition is the **Extract From Binary Image** button which allows for the creating of a surface (.sur) from a binary pre-segmented image. In this way segmentations generated using other tools such as itk-SNAP, which are typically stored as binary images, can be imported into the Surface Editor for more detailed manipulation. Pressing the **Extract From Binary Image** button, brings up a helper dialog which allows the user to specify four options: (i) Surface Smoothing: this determines how many control points will be used to parameterize the extracted surface, (ii) Image smoothing: which is the standard deviation of the kernel that is used to smooth the binary image prior to surface extraction and (iii) and (iv) bottom and top slice to limit the extraction range. The extraction routine automatically performs connectivity analysis on the underlying image to eliminate small holes etc. and will return only those consecutive slices for which the surface extraction has succeeded.

### 8.3.3 Objectmaps and the Objectmap Menu

*Most of the objectmap editing functionality is identical to that shown in Figure 8.2 and will not be repeated here. The only limitation is that all editing is done with a 2D brush, the “3D” checkbox is ignored.*

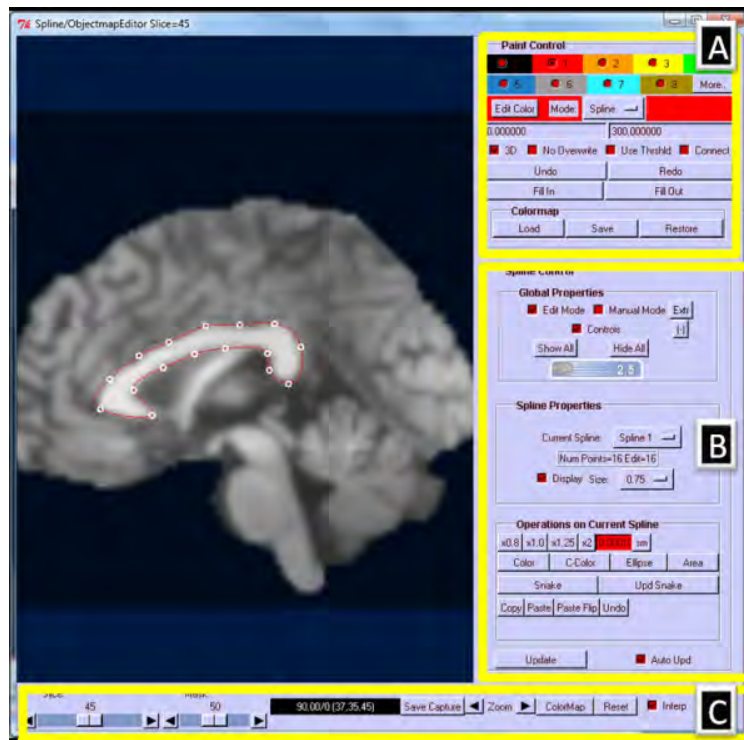


Figure 8.6: The Spline/Objectmap control – A tool for manual editing of surfaces. Here, a manually defined boundary for the corpus callosum that uses 16 points on one sagittal slice of an image is shown.

Objectmaps can be created either from the current displayed image in the viewer using the “Create From Data/Grab from Current Displayed Image” button or from the surfaces (i.e. convert surfaces to objectmaps) using the “Surface Objectmap Tool” control shown above that is accessed using the “Create from Data/Create from Surfaces” option. The default setup for this control (using the “Create!”) button enables the user to specify which surfaces to use, and value inside each surface as well as the background. (The Distance Map function creates a “2D Distance Map” based on the surfaces which is useful in some applications.) The “Mask” button creates a mask where only the parts of the image inside the selected surface are left, the rest are blanked. The opposite (i.e. only parts of the image outside the selected surfaces) is accomplished using the “Inverse” button.

The (Objectmap — Fat Objectmap Operations) menu choice brings up two different customized options for application specific needs not described here.

### 8.3.4 The Spline/Objectmap Control

The local editing of surfaces/objectmaps is done within the Spline/Objectmap tool shown at right in Figure 8.6. This is accessed using the “Objectmap/Spline Editor” button in the bottom left corner of the main SurfaceEditor window. This extremely powerful control enables the manipulation of individual curves by moving b-spline control points, the drawing of new curves by simply clicking points which are then converted to b-splines for further editing, as well as the manipulation of objectmaps using paint-brush like operations.

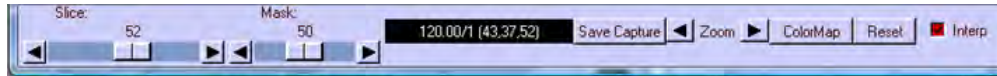


Figure 8.7: The bottom row tools in the Spline/Objectmap Control. The tools in this space let you choose which slice to work with, and how it is visualized during spline editing.

## Editing Splines

In the viewer portion (i.e. the left part of the image in Figure 8.6), we show a single spline. If the viewer “edit” mode is turned on, manual mode off and the mode in the Fill Control is set to “Spline” (these are all the default settings), then the spline can be manipulated by moving individual control points by clicking and dragging them using the left mouse button. Holding down the shift-key while and left mouse button allows you to move the entire spline. Clicking on the curve with the left mouse button while holding down the “Control” key adds a new control point half way between the closest two control points. In this way, a fully customizable spline curve can be generated on each slice of an image, thus creating a user-defined surface.

### The Bottom Row – selecting slice and objectmap transparency.

Given the absence of a menu in this control, most core functionality is present in the bottom row “buttonbar” shown above. The “Slice” control determines which slice is currently being edited. This is mirrored in the slice control in the bottom row of the main Surface Editor window. The “Mask” control sets the relative opacity of the objectmap with respect to the image. If set to 0 the objectmap is completely transparent (effectively hidden) whereas if set to 100 the objectmap becomes completely opaque and hides the underlying image. The next label “200(56,83,62)” shows information about the last clicked voxel in the image with, in this case, image coordinates (56,83,62) and intensity=200.00. The “SaveTiff” button enables the saving of the contents of the viewer (i.e. a screen grab) as a tiff/jpeg file (depending on the suffix).

The “Zoom” arrow buttons enabling zooming in and out (this is also accomplished by dragging with the middle mouse button). The “ColorMap” button brings up the standard colormap editor (as described in the viewers documentation) and the “Reset” button resets the viewer to default zoom/location levels. The “Interp” check button turns on/off standard anti-aliasing interpolation in the image display.

### The Spline Controls – manipulating splines

This control has functionality for manipulating individual curves. We again note that in the SurfaceEditor each surface is made up of a single curve on each XY slice (within the bounds set by the bottom trim and top trim controls). These curves are manipulated using the spline control shown here. Most of the basic functionality of the spline control is described in Figure 8.8.

The “Global Properties” frame has options for turning “Edit Mode” on and off, enabling and disabling “Manual Mode” described in detail below – showing and hiding the actual control points

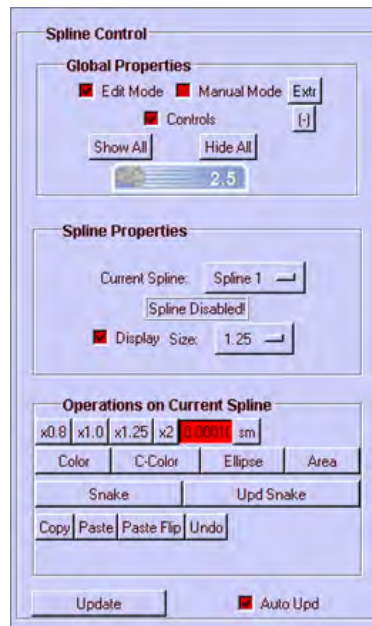


Figure 8.8: The Spline Control tool panel.

and showing and hiding all splines.

The (-) button is used to flip the relative depth positioning of the splines and the image. It is primarily needed when tracing on Sagittal images. If the splines are not visible a single click of the (-) button should fix the problem,

Individual spline properties are set using the options in the “Spline Properties” frame. The “Current Spline:” drop menu selects the current spline to edit/manipulate. Spline 1 is the spline corresponding to “Surface 1” in the main Surface Editor window, “Spline 2” is part of “Surface 2” etc. The spline can be hidden by unchecking the display button. The “Size:” option menu controls the size of the control points.

The “Operations On Current Spline” frame mirrors functionality in the “Operations on Current Stack” frame in the “Surface” tab of the main surface editor. The “Snake” and “Update Snake” options enable the use of interactive snake-like segmentation methods to be described in more detail in upcoming lectures.

The Copy/Paste options are the real workhorse options of this program. Given the general smoothness of most anatomical structures, the shape of a surface on slice 51 is very similar to that in slice 50. Hence the curve can be copied from slice 50, the slice number incremented to 51 and then the curve pasted giving an excellent initial position for further editing. The “Undo” button reverts to the version of the curve that is currently stored in the surface editor (i.e. prior to editing). Past flip performs an x-flip prior to pasting.

**The Update and “Auto Upd”** buttons control the communication between the spline/objectmap editor and the surface editor. On pressing update the versions of the curves and the objectmap slice (see below) as edited in the spline/objectmap editor are copied over to the main SurfaceEditor,

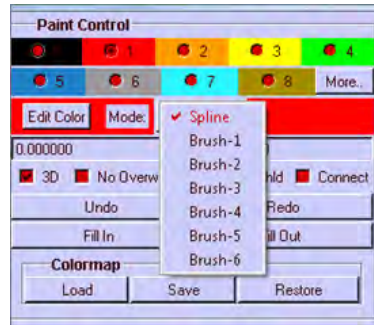


Figure 8.9: The Paint Controls. This control is mostly identical to the one described as part of the Objectmap Orthogonal Viewer – see Figure 8.2. It has a couple of additional buttons (Fill In, Fill Out) to use the current as a template for filling in regions. In addition, the first entry of the mode button is now “Spline” as opposed to “Off”. When spline is selected the first mouse button is used to edit the spline, otherwise it is used for painting.

and the 3D display in the Surface Editor is updated to reflect this. The “Auto Upd” checkbox enables automatic updating upon slice change. The “Update” button automatically copies the current spline and places it in the clipboard so that it can be pasted directly.

**Manual Mode:** . In certain cases the initial shape to be traced is very complex and cannot easily be captured by moving the control points of an elliptical spline to fit this. The use of “Manual mode” enables the use of the more traditional (and less flexible for subsequent editing purposes) method of clicking individual points on the boundary to create a polygonal curve which is then converted to a b-spline for editing. To use manual mode first enable this by selecting the “Manual Mode” checkbox towards the top right corner of the “Global Properties” frame.

Next, click individual points using the left mouse button to form a polygon (there is no need to close this, the fitting process automatically does this for you – fact the polygon is best left open). Once the polygon is mostly closed, turn off “Manual Mode” by clicking on the “Manual mode” checkbox as before. Once this is done a pop-up dialog asks whether you want to replace the current spline with one fitted to the current set of manually clicked points. If you answer “no”, the current manually clicked points are deleted. If you answer “yes” a spline is fitted to the polygonal points which can then be further edited.

### The Paint Controls - manipulating objectmaps

The “Paint control” – see Figure 8.9 – enables manipulation of object-maps by appropriately coloring in image voxels (by using the left mouse button). The “Fill In” and “Fill Out” buttons use the threshold (but not connect) settings together with the current spline to paint the image. Fill In will fill the inside of a spline with a certain value, whereas “Fill Out” will fill the region outside the spline with the selected value. If threshold settings are set, these are also used.

## 8.4 Delineating a surface: a step-by-step recipe.

1. Load the Image using the File/Load Option in the main Surface Editor window
2. Adjust the orientation of the image to ensure that the slice orientation on which you want to draw curves on is the same as the orientation of the image. For example, if the structure you are interested in is best outlined on coronal slices, the image will need to be re-oriented in a coronal orientation (Use the tool under Image Processing/Reorient-Crop Image). Remember to make the change permanent using “Copy Results to Image” and to save the reoriented image for later use.
3. Identify the extent of the structure of interest in the image, i.e. the bottom slice and the top slice that bound the surface. Set the bottom trim and the top trim in the main Surface Editor window, in the “Surface” tab to reflect this
4. Save the surface – will enable the autosave functionality to be used. The autosave function automatically saves a surface each time it is updated in a file “surfacename\_auto.sur”. Should there be an unexpected (and these days rare) software crash, the autosave file can be used to recover most lost work.
5. Select a slice roughly at the middle of the surface (i.e. half way between top trim and bottom trim), using the slice control in (**E. Editor Slice** ) on the main SurfaceEditor window
6. Open the spline/objectmap editor
7. Use the manual mode method to roughly outline the curve on this slice.
8. Resample the spline as necessary to have the appropriate number of control points.
9. Refine the curve by moving these control points.
10. Update the 3D Window by clicking “Update” in the Spline Editor
11. Change the slice.
12. Paste the last traced curve. (The “Update” button automatically does the “copy” operation).
13. Edit the curve
14. Go back to step 9 until all slices have been traced.



## Chapter 9

# Tissue Classification

### 9.1 Accessing the Segmentation Tool

Image Segmentation is the process of identifying features in images and marking them as distinct from one another. These features can be things like grey and white matter in an MRI of the brain or individual organs in an MR or CT image. We describe here the segmentation tools in BioImage Suite, some of which leverage the functionality of the FSL library of image analysis tools (See <http://www.fmrib.ox.ac.uk/fsl/>). See also Appendix B for instructions as to how to install FSL.

The Segmentation tools are accessed under the Segmentation Menu in most viewers – see Figure 9.1. The entries in the menu change depending on whether BioImage Suite successfully finds an installation of FSL on the computer. If FSL is found, two additional options appear for calling the Brain Extraction Tool (Skull Stripping) and the FSL - Gray/White Segmentation methodology respectively.

Many of the graphical user interfaces described below make use of a simple embedded image GUI control shown in figure 9.2. The appearance of this is somewhat variable but the basic concept is that it stores an image (usually an algorithm output image) and enables easy manipulation of this.

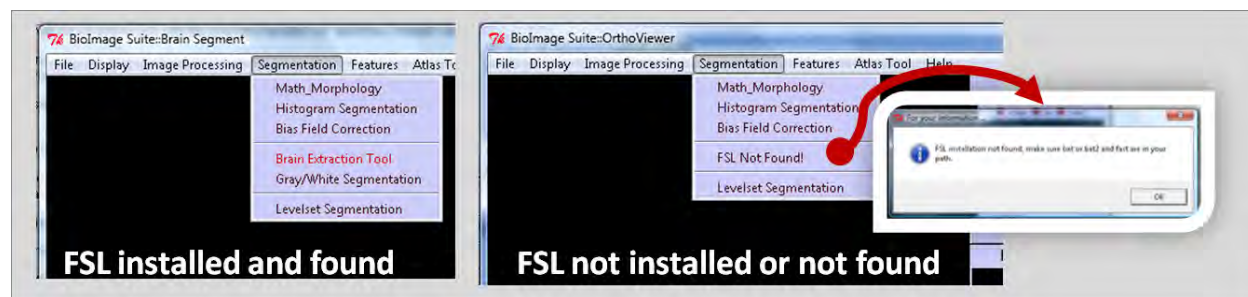


Figure 9.1: The Segmentation Menu. Left: FSL found, Right: FSL not found – see Section A.1 for details on installing and configuring FSL.





Figure 9.2: The image GUI control for manipulating an image. The filename is at the top (empty) with the dimensions (1 1 1) at the top right. The textbox below gives a longer description of the image. Common functionality (Load, Save, Info, Display) is available from the buttonbar below. Sometimes additional buttons appear here depending on the viewer. For example, in the Brain Register applications there are two buttons “Display Ref” and “Display Trn” instead of the “Display” button for displaying the image in either of the two viewers.

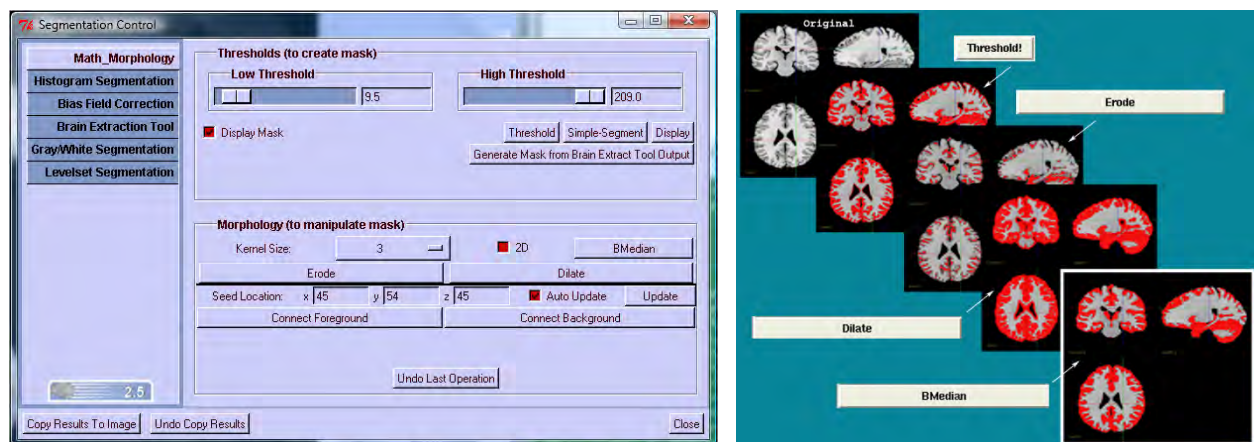


Figure 9.3: **Left:** The Segmentation Control with the Morphology Tab selected. **Right:** Some examples of the results of the operations in the Math Morphology Tab. The original is at the top left. The next image in the cascade is the mask image generated by thresholding. This mask is then eroded, yielding the next image in the series. The threshold result mask was then dilated to generate the fourth image. The threshold result mask was also passed through the BMedian function repeatedly ( 4x) to remove outlying “islands” (note the midbrain). These provide rough examples of the results possible using these tools.

## 9.2 Math Morphology

This section of the Image Segmentation toolbox provides some functions for simple mathematical morphology. These functions require you to threshold the image first, yielding either a mask in addition to the original image (in the objectmap viewer) or a binary image in the 3-component viewers. They are accessed via the (Segmentation — Math.Morphology) menu selection, as shown in Figure 9.3 above.

The top section of the Math\_Morphology section of the dialog box contains two sliders for setting upper and lower thresholds. Once they are set, simply press the **Threshold** button, and if the **Display Mask** checkbox is checked, the thresholded image will be displayed in the viewer. If not, simply hit the **Display** button to send the result of the threshold operation to the viewer. If, instead of thresholding, you want to generate a mask based on the output of the brain extraction tool, which is useful for generating whole-brain masks, hit the large **Generate Mask from Brain**

**Extract Tool Output** button.

The **Simple Segment** button is just a macro for segmenting that calls the threshold function followed by a couple other functions described below (The order is: Threshold; Erode; Connectivity; Dilate)...more on this function and its constituents below. See Fig 9.3 for some examples of a rough segmentation of an MRI brain image into grey and white matter regions. The red regions are the mask as shown in the 4-component viewer.

## Math Morphology Operations

In the lower half of the Math\_Morphology tab, there are a group of buttons that operate on the mask generated either by the thresholding or brain extraction tool import done in the upper half of the tab. If the “2D” checkbox is enabled all operations, apart from Connectivity, are performed in 2D (i.e. for each slice), otherwise the operations (e.g. Dilate/Erode/Median) are performed in 3D which is the default. Their functions are summarized below:

*Note:* The erode and dilate operations utilize as a parameter a kernel size, which sets how many of each voxels neighbors in each dimension should be examined. This is set with the “Kernel Size” menu. If the 2D option is selected, only the within-slice dimensions are considered. Essentially, the kernel size parameter controls how aggressive the erosion or dilation is.

**Erode:** Erode shrinks the mask by looking at the surrounding neighborhood of each voxel. If any of the neighbors have a value of zero, then the voxel in question is set to zero.

**Dilate:** Dilate grows the mask in the exact opposite manner as erode shrinks it: if any neighboring voxel has value 1, then the voxel in question is set to 1.

**BMedian:** BMedian polls the surrounding voxels and if the majority are non-zero, then the voxel in question is set to 1; if most of the neighbors are zero, then the voxel is set to 0.

**Connect Foreground:** This operation is dependent on the location of the cursor. It sets to zero any voxels in the mask that are not contiguous with the voxel indicated by the seed. The position of the seed can either be obtained using the viewer cursor or specified manually. (This voxel must be equal to one, otherwise the whole image goes to zero!) This is useful for separating out regions of interest.

**Connect Background:** This operation is dependent on the location of the cursor. It sets to one any voxels in the background that are not contiguous with the voxel indicated by the seed (This voxel must be equal to zero!) This is useful for eliminating small holes inside foreground regions.

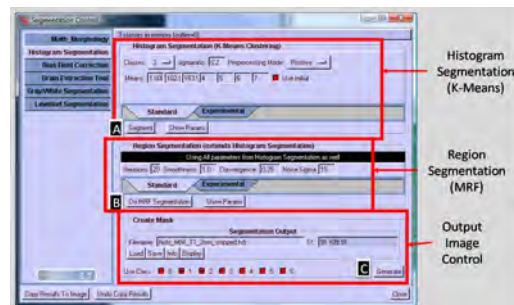


Figure 9.4: The Histogram Segmentation Tab.

### 9.3 Histogram Segmentation

This tab provides access to two algorithms, Histogram Segmentation and Region-based Segmentation. Both algorithms take an input image and generate an objectmap of values 0 to  $N-1$ , where  $N$  is the number of classes. Histogram Segmentation operates exclusively on the image histogram which makes it very fast, the downside is that there are no regional constraints on the segmentation. It is essentially a customized implementation of k-means clustering optimized for this application. The region segmentation extends Histogram segmentation (which it uses for initialization) to allow for regional homogeneity constraints using a Markov Random Field (MRF) smoothness model.

**Histogram Segmentation** This algorithm divides the image into  $N$  normally distributed intensity classes using a k-means clustering algorithm. It has three standard parameters. (Additional parameters can be set in the “experimental” tab which are not discussed here). The first parameter is “Classes” which sets the number of classes ( $N$ ) in which the image is to be divided. The second parameter is *sigma ratio* which sets the maximum value of the ratio of the maximum standard deviation to the minimum standard deviation. Setting this to 1 ensures that all classes have the same standard deviation. A small non-zero value is useful to ensure that small classes do not disappear. The last option is the preprocessing mode which describes the pre-processing applied to the image prior to generating a histogram. It has three settings: None – no preprocessing, Positive – all non-positive voxels are ignored and Autorange which eliminates voxel outside the 1%-99% span of the cumulative histogram.

The class means for upto seven classes are also displayed. This can be initialized manually – set the “Use Initial” checkbox to use the initial parameters. On completion of the segmentation – invoked by the “Segment” button the values in the mean text boxes reflect the results. The “Show Params” button provides additional details.

**Region Segmentation** This algorithm essentially adds a Markov Random Field smoothness model to the previous algorithm. This has the result that the segmentation can no longer be performed on the intensity histogram alone hence this algorithm is significantly slower, as each voxel needs to be examined individually. It uses all the parameters from the Histogram Segmentation as well as the following additional four parameters (plus additional ones in the experimental tab which are not described here). (a) Iterations – this is the maximum number of iterations, (b) Smoothness – this sets the weight of the spatial smoothness component, setting this to zero reverts to histogram segmentation, (c) Convergence – sets the maximum percentage of voxels that are

allowed to change in a given iteration and the algorithm considered to have converged and (d) Noise sigma – which is an estimate of the image noise standard deviation. This algorithm is very similar to the FSL/FAST algorithm with one key difference. FSL/FAST attempts to perform intensity-inhomogeneity correction at the same time as segmentation (which works OK at 1.5T, less at higher field strengths) whereas “Region Segmentation” is a pure segmentation method and assumes that the Bias field has been corrected previous to its use.

**Segmentation Output** This control enables the output of the histogram/region segmentations to be used to generate a mask for either bias field correction (see below) or math morphology operations above. By default the output of the last histogram or region segmentation run is stored in the “Segmentation Output” control which has options for Loading/Saving and Displaying this objectmap.

Below this, one can select which classes to use to create a mask. Clicking “Generate” takes the ROI defined by these classes and generates a binary mask for use in the Math Morphology Tool. Clicking “Do Bias Field” executes the polynomial bias field correction algorithm using the mask defined in this control and the parameters defined in the Bias Field Correction tab.

## 9.4 FSL Brain Extraction Tool

The brain extraction tool is used to remove the skull from an image, leaving only the region occupied by actual brain tissue. It segments these by using the dark space between the skull and brain, occupied by the Cerebro-spinal Fluid (CSF). This tool comes from the external program FSL’s toolkit (see the FSL webpage. Thus, it will only appear if FSL is installed. (See also Section A.1 for instructions as to how to install FSL.)

### 9.4.1 The BioImage Suite Interface to BET

In the “Brain Extraction Tool” tab in the Segmentation Control window, or via the ( Segmentation — Brain Extraction Tool ) menu selection, you have just a couple parameters to potentially set, and a button to execute the tool. The options are fairly straightforward.

To make the algorithm more aggressive, removing a larger portion of the image, set the Fractional Intensity Threshold slider to a higher value. To skew the stripping towards either the top or bottom of the image, use the Gradient Threshold slider. A positive value here yields a brain that has been stripped more aggressively at the top. If you check the “Use Current Crosshairs” checkbox, the location of the crosshairs in the viewer will be taken into account, and a sphere centered on that point will be the starting point for the brain extraction. This can help if there is a feature in the image that you want to be sure to remove. Set the crosshairs to a location away from this spot before executing the extraction, and check this box. The “Initial Radius” field only operates with the “Use Current Crosshairs” option selected, and sets the initial size of the sphere that i used to initialize the stripping. To execute simply press the **Execute Brain Extraction Tool** button. An output dialog box will pop up and inform you of the progress.

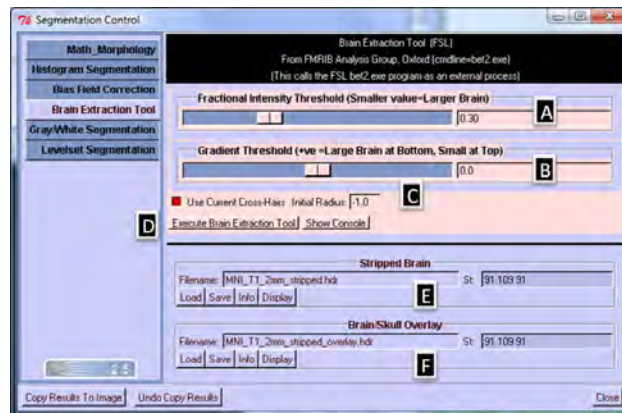


Figure 9.5: The Brain Extraction Tool Tab – this calls the external program BET from FSL.



Figure 9.6: Brain Skull Overlay.

When the brain extraction is done, the results will be automatically saved, with “\_stripped” appended to the original filename. The filename is placed into the “Stripped Brain” box at the bottom of the window, where you can save it somewhere new (**Save**), load another image (**Load**), get a popup with some information about the image dimensions and orientation (**Info**), and display it in the viewer (**Display**). The image is displayed in the Results view of whichever viewer contained the original. A brain/skull overlay image is also generated (example at right), which shows where the boundary between brain and non-brain has been set. This lets you quickly see if any regions in your image are being erroneously included or excluded from the brain, and change parameters accordingly. Here too, you can save and load the image, display its properties, and send it to the viewer.

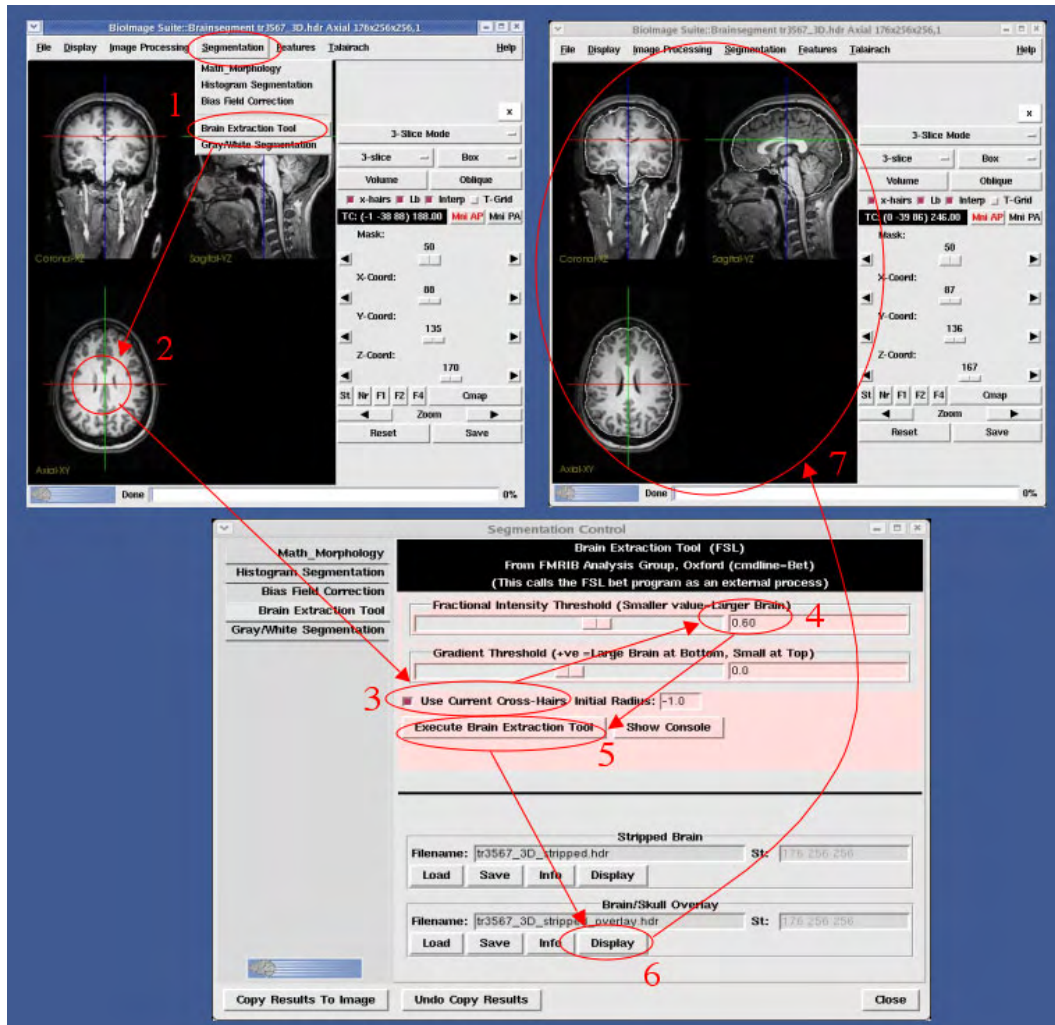


Figure 9.7: Methods used for the First Brain Extraction.

1. Choose (Segmentation — Brain Extraction Tool) – See Figure 9.7.
2. Put crosshairs in center of brain above corpus callosum along the midline
3. Click use current crosshairs (make the box to the left, red in color)
4. Set Fractional Intensity Threshold to 0.6 and the Gradient Threshold to 0.0
5. Click execute brain extraction tool - BioImageSuite will automatically save new stripped brain as [filename].stripped.hdr
6. Click Display Under Brain/Skull Overlay to check if any tissue is erroneously included or excluded
7. Examine outline of stripping by clicking and holding the left mouse button and dragging it back and forth over one view space (coronal) while watching the other (sagittal).
  - If part of the brain is excluded: Decrease Fractional Intensity Threshold (in increments of 0.1 or 0.5) and repeat steps 2-7.
  - If too much skull/throat/neck is included: Increase Fraction Intensity Threshold (in increments of 0.1 or 0.5) and repeat steps 2-7.



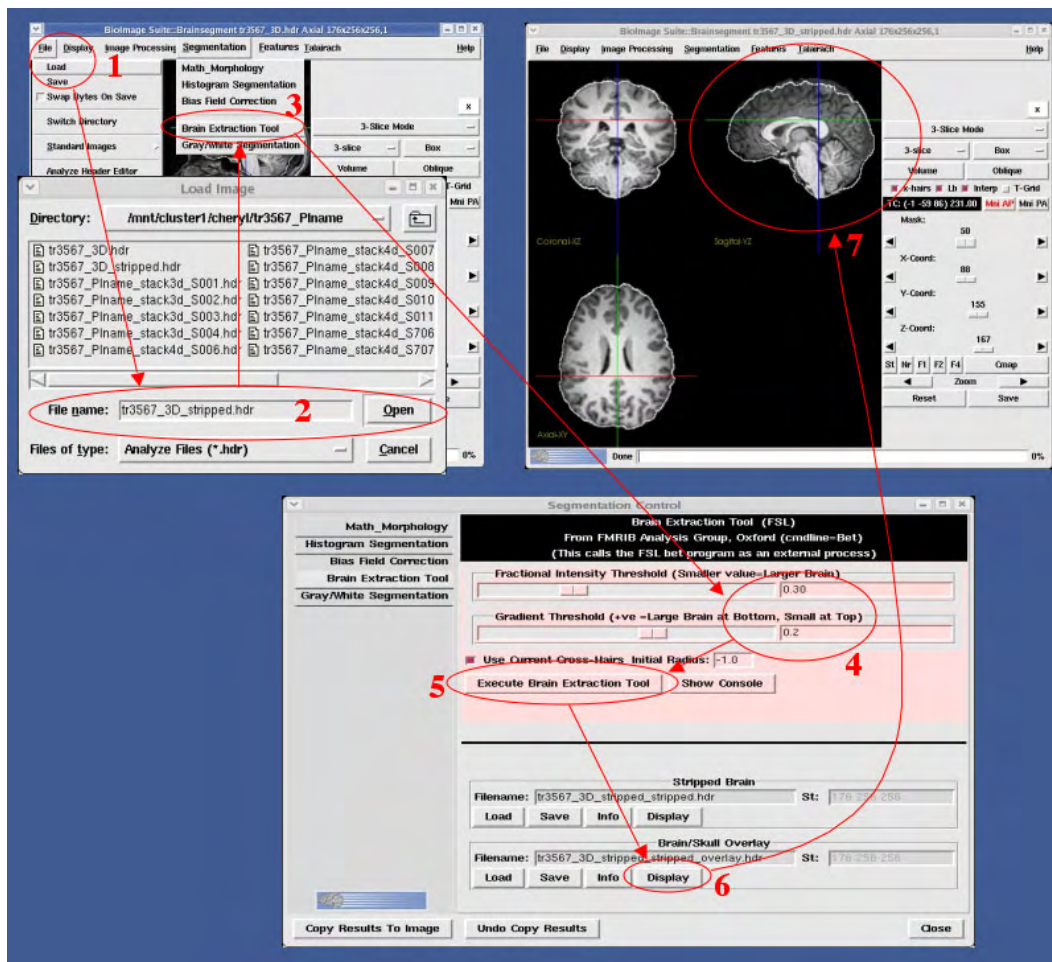


Figure 9.8: Methods used for the Second Brain Extraction.

1. Choose (File — Load), then (see Figure 9.8)
2. Choose [filename]\_stripped.hdr and click Open
3. Choose (Segmentation — Brain Extraction Tool). As above in steps 2 and 3, put the crosshairs in the center of the brain and make sure that the box to the left of use current crosshairs is red.
4. Change Fractional Intensity Threshold to 0.4 or 0.3 and the Gradient Threshold to 0.2
5. Click Execute Brain Extraction Tool. BioImageSuite will automatically save second stripped brain as [filename]\_stripped\_stripped.hdr.
6. Click Display under Brain/Skull Overlay to check if any brain is erroneously included or excluded.
7. Examine outline of stripping as stated above in Step number 7.

Sometimes doing multiple rounds of brain extraction on the same image yields better results than a single extremely aggressive operation. To do this, simply execute the brain extraction tool, and copy the contents of the results view to the image view (see Image Display vs. Results Display), then repeat, with slightly more aggressive parameters.

### 9.4.2 A Recipe for Brain Extraction - Removing the Skull from a 3D image

The brain extraction tool is used to remove the skull from an image, leaving only the region occupied by actual brain tissue. It segments these by using the dark space between the skull and brain, occupied by the CSF. This tool comes from the external program FSL's toolkit (see the FSL webpage). Thus, it will only appear if FSL is installed. (Such integration is easier on Unix systems at this point).

**First Brain Extraction:** The goal of the first brain extraction process is to remove most of the skull/throat/neck without losing any brain tissue. This extraction will not be extremely accurate and may include more skull than desired in order to keep all of the brain tissue. The remainder of the skull can be removed in the second brain extraction. This process is described in Figure 9.7.

**Second Brain Extraction:** The goal of the second brain extraction process is to be more accurate around the brain and remove the meninges and any remaining skull. This process is described in Figure 9.8

## 9.5 Grey/White Segmentation – using FSL Fast

*Note:* The Grey/White Segmentation control assumes that you have already performed a brain extraction (Described previously), yielding a brain with no skull.

**Also note:** The algorithm is optimized for a 1.5T scanner. Results may be different for other field strengths.

Upon execution of the Grey/White segmentation tool, by default the image is segmented into three classes: Grey matter, White matter, and Cerebro-spinal Fluid (CSF). You may specify more or less classes with “Number of Classes” menu on the right side of the window. When you click the **Execute Automated Segmentation Tool** button, a console pops up, showing the output of FSL program that performs this segmentation function. When the brain extraction is finished, the console will show a table of volumes for each class in the image.

The volumes are reported in  $mm^3$ , except for “brain percentage”. These values can also be examined in more detail by clicking the **Statistics** button in the “Post Process Output” box. The image output is a classification map (filename: \*\_segm\_restore.hdr), which has  $n + 1$  frames, where  $n$  is the number of classes selected. The first frame of the image shows all classes, with a value between 0 and 1 for each class. The remaining frames are binary images that correspond to each class.



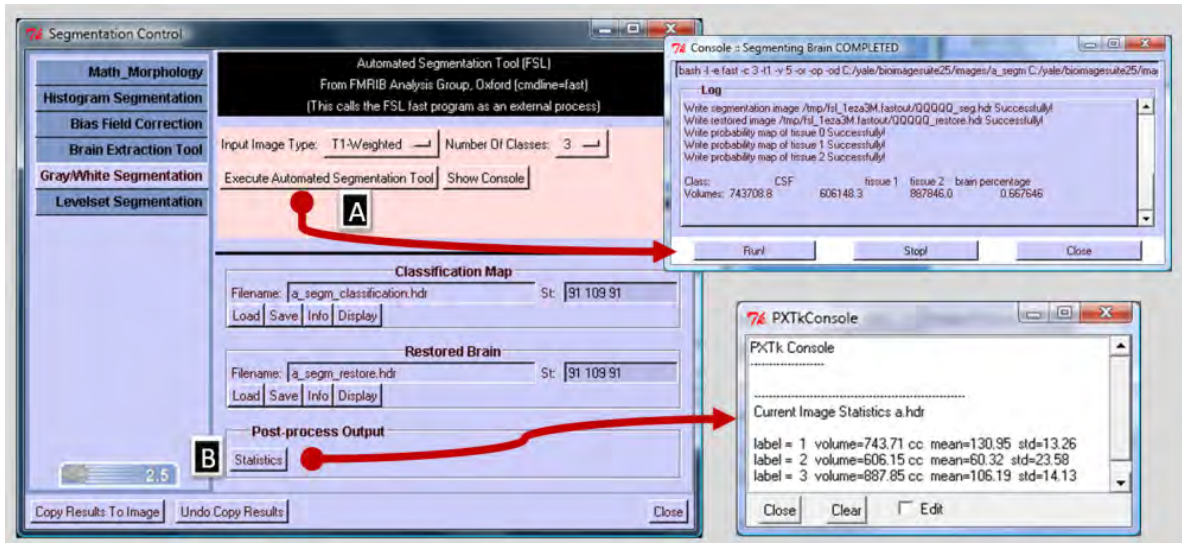


Figure 9.9: Grey/White Segmentation Control.

This algorithm assumes that the Brain is already skull stripped, use the Brain Extraction Tool to do so. The algorithm is very similar to the “Region Segmentation” algorithm described above with the addition of an integrated Bias Field Correction step. This works well for 1.5T brain images, its performance is worse at higher field strengths.

The outputs generated are a classification map which is binary objectmap with labeled tissue classes and a restored brain which is the result of applying the FSL/FAST bias field correction algorithm to the original image.

## 9.6 Bias Field Correction

Often MRI images are corrupted by intensity inhomogeneity or shading artifacts. This tab, shown in Figure 9.10, provides access to two algorithms, Slice Homogeneity Correction and Volumetric Bias Field Correction. Both aim to remove image intensity inhomogeneities caused by MR Scanner receiver coil sensitivity variations. Both algorithms may use a Mask Image to set the Region of Interest (ROI) in which the optimization is performed – this is stored in Mask Frame (labeled as C in Figure 9.10). The output of both algorithms is a Bias Corrected Image and optionally an estimate of the Bias Field which go to the Outputs Frame (D in Figure 9.10) in the bottom.

### 9.6.1 Slice Homogeneity Correction

This algorithm aims to eliminate slice-by-slice intensity variations. This is particularly useful in multi breathhold multislice acquisitions such as in the case of abdominal imaging. This very simple but surprisingly effective algorithm performs a best straight line fit between adjacent slices (using neighboring voxels as data point pairs – but excluding edge points) to estimate the scaling and (optionally if the “Pure Scaling” checkbutton is off) the offset between the two slices. If the “Use

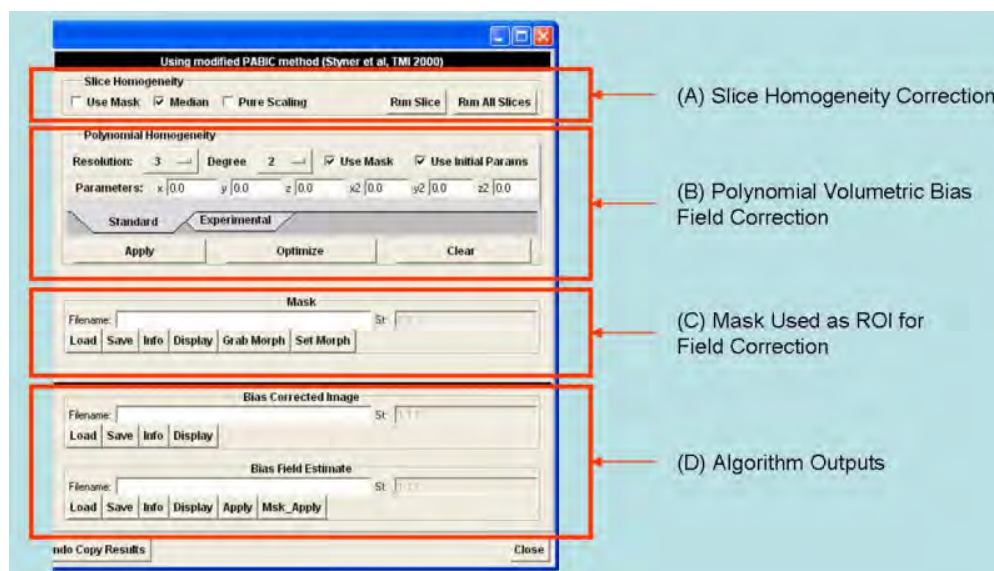


Figure 9.10: The Bias Field Correction Tab.

Mask” checkbutton is set then the Mask defined in the Mask Frame (C) is set. If the “Median” checkbutton is set then the line fitting is done using a least absolute value criterion (median – which can be more robust) whereas if it is off a standard least squares fit is performed. There are two buttons to invoke the algorithms. Run Slice – performs homogeneity correction only in the Z-direction (slice acquisition) Run All Slices – performs homogeneity correction in all three orthogonal axis directions (X,Y,Z). This is very useful as a pre-processing step for the polynomial (volumetric) bias field correction algorithm, described next. It is especially useful in cases of large inhomogeneities. See Frame A of Figure 9.10.

### 9.6.2 Polynomial Bias Field Correction

This algorithm is a custom implementation of the PABIC algorithm (Styner et al, TMI 2000 [111]). It aims to improve image inhomogeneity by estimating a polynomial model of the inhomogeneity that will result in the image intensities being more clustered with respect to their class means – the numbers of classes and the means invoke the Histogram Segmentation algorithm and its current parameter settings. The algorithm has 4 key parameters: (a) Resolution: 1 = Native, 2=Half Native etc. Since the bias field is a low frequency effect using a reduced resolution can be just as accurate and results in significant computational savings. (b) Degree: 1-3, determines the maximum degree of the polynomial used 1=linear, 2=quadratic, 3=cubic. (c) “Use Mask” – if on restricts the optimization to the ROI defined by the Mask Image (in frame C of Figure 9.10). (d) “Use Initial Parameters” – if on the user can specify initial values for the linear and some of the quadratic parameters to manually optimize the field estimate. Trial and error can be used to set this, clicking “Apply” simply applies these values with no optimization. Manual initialization may be necessary in the case of large inhomogeneities such as those present at high fields (e.g. 4T as in the example shown in Figure 4). An additional three buttons (Print, Load, Save) enable direct access to the polynomial coefficients underlying the bias field estimate.

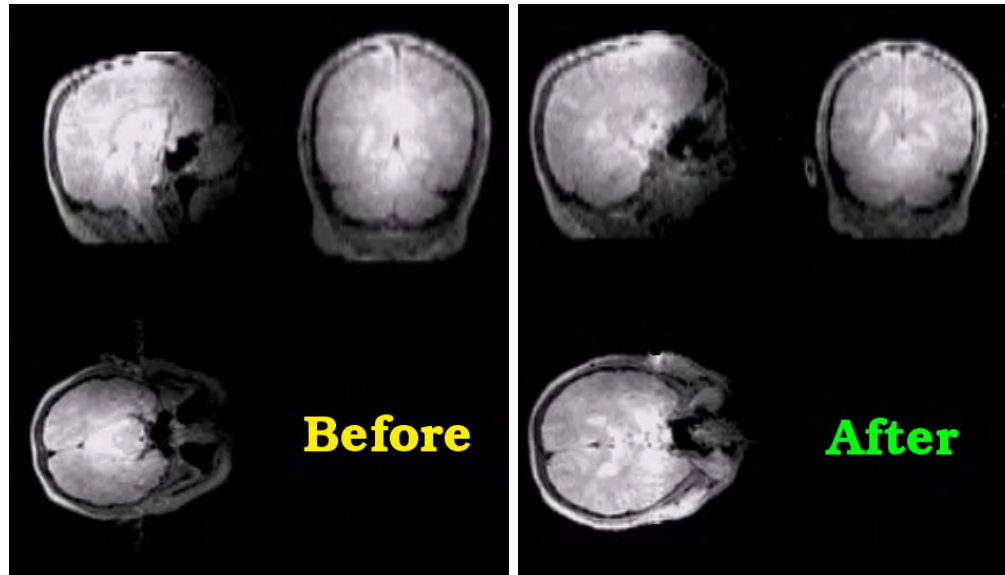


Figure 9.11: Bias Field Correction Example. **Left:** The left panel shows an image before bias field correction. Note the inhomogeneity in the intensities of the image (the frontal and temporal lobes are darker than the rest of the brain). **Right** In the image shown in this panel, this artifact has been corrected. The bias field in this example was particularly difficult to parameterize, and the data was of poor quality, but the result image is significantly more uniform in intensity. Often, when the bias field is linear or parabolic in nature, the result is even better.

The algorithm is invoked with the Optimize button. The Clear button resets the parameters to their default values (all zeros)

The Mask frame is used to manipulate the mask restricting the optimization of the algorithm. The Grab Morph button copies the current mask used in the Math Morphology tab to this control, whereas the Set Morph does the opposite, i.e. it transfers the current mask to the Math Morphology control for further editing.

**Outputs:** The corrected image from the last slice homogeneity or bias field correction operation is stored in the Bias Corrected Image control in frame D of the tab as shown in Figure 9.10. The estimated bias field is stored in the Bias Field control which has similar functionality.

### 9.6.3 A Recipe for Bias Field Correction for a Brain T1-Weighted Image

At higher field strengths, sometimes structural images acquire an intensity gradient across the image making some parts of the image brighter than others. This intensity gradient can influence segmentation algorithms erroneously, therefore a method has been developed to remove this intensity gradient from the image, it is known as bias field correction.

1. Choose (Segmentation — Brain Extraction Tool). A new Segmentation Control window will

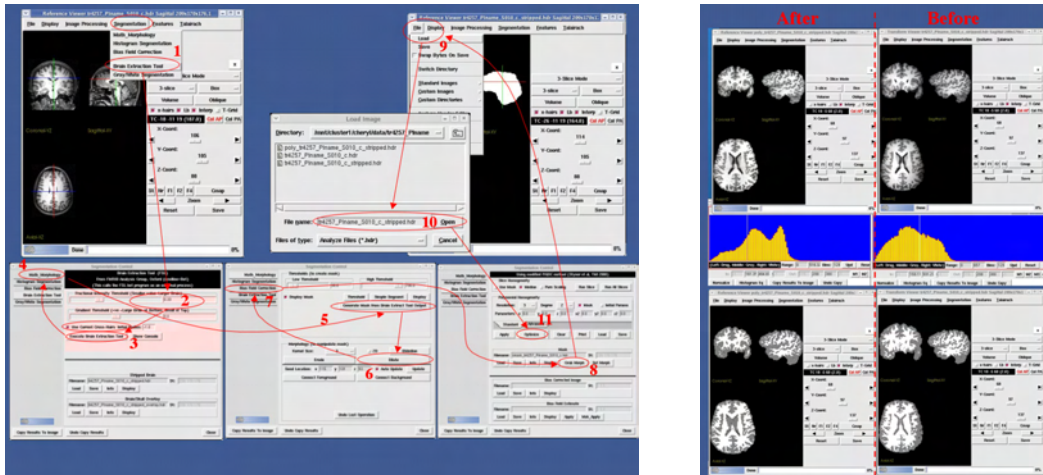


Figure 9.12: Methods used to correct the bias field of a structural image.

appear. Put the crosshairs in the center of the brain above the corpus callosum along the midline.

2. On the Segmentation Control Window, set the Fractional Intensity Threshold to 0.6 and the Gradient Threshold to 0.0
3. On the Segmentation Control Window, click use current crosshairs (make the box to the left, red in color). Click execute brain extraction tool - BioImageSuite will automatically save new stripped brain as [filename]\_stripped.hdr. Click Display Under Brain/Skull Overlay to check if any tissue is erroneously included or excluded.
4. Once a satisfactory stripping of the brain has been achieved, click the Math Morphology tab of the Segmentation Control Window.
5. On the (Segmentation Control — Math Morphology) tab click Generate Mask from Brain Extract Tool Output. This will create a white mask of the stripped brain.
6. On the (Segmentation Control — Math Morphology) tab click Dilate to increase the size of the mask slightly.
7. On the Segmentation Control Window click the Bias Field Correction tab.
8. On the (Segmentation Control — Bias Field Correction) tab click Grab Morph to input the previous mask into the bias field correction algorithm.
9. On the Viewer window choose (File — Load)
10. Choose the stripped version of the brain image.
11. On the (Segmentation Control — Bias Field Correction) tab click Optimize. This will generate the bias filed correction and apply it to the image.
12. The results shown in (Figure 10, image II), show that before the correction the histogram did not differentiate between gray and white matter, while after there are two clear peaks in the histogram. Also, a simple segmentation is more precise after the correction rather than before.

## 9.7 Appendix: A Little bit of Theory

### 9.7.1 The K-means Clustering Algorithm

The K-means clustering algorithm (see Duda & Hart) can be used to essentially estimate the thresholds above. In our implementation we assume that the intensity in each tissue class can be modeled as having a normal distribution with mean  $\mu$  and standard deviation  $\sigma$ . We will use the notation  $p(i|l = c) = \mathcal{N}(\mu_c, \sigma_c)$  to define this, where  $l$  is the label and  $c$  is the class number. The goal of the version of the k-means algorithm that we will describe is to estimate the class parameters  $(\mu, \sigma)$  for all classes and then to assign each voxel to the class that maximizes the function:

$$\begin{aligned} \hat{l}(\mathbf{x}) &= \arg \max_l p(i(\mathbf{x})|l(\mathbf{x}) = c) \\ &= \arg \max_l \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(i(\mathbf{x})-\mu_c)^2}{2\sigma_c^2}} \end{aligned} \quad (9.1)$$

A simpler form of the algorithm assumes that all  $\sigma_c$ 's have the same value. This reduces the problem to estimating the means only. The procedure can be described in recipe form as:

1. Decide on the number of classes  $M$
2. Assign class centroids  $\mu_c$  and optionally standard deviations  $\sigma_c$  for each class. The most common way to do this is to equally space the  $\mu_i$ 's and set all  $\sigma_i$ 's to some constant value.
3. Estimate the labels  $l(\mathbf{x})$  using equation 9.1. This is an exhaustive optimization – compute  $p(i|l = c)$  for all  $l$ 's and pick the  $l$  that maximizes this probability.
4. Estimate a new set of  $\mu_c$ 's and  $\sigma_c$ 's by computing the mean and standard deviation of the intensities of the voxels labeled as having class  $c$ .
5. Repeat steps 3-4 until the parameters  $\mu_c$  and  $\sigma_c$  converge.

Note that, since the spatial position of the voxels  $\mathbf{x}$  does not enter into the calculation, a quick way to implement this method is by first forming the image histogram and performing all operations on the histogram itself. This can speed up the iterations by a factor of 15000 or so in the case of an  $128 \times 128 \times 128$  image whose histogram can be described by 128 bins.

### 9.7.2 Imposing Local Smoothness using Markov Random Fields

The key weakness of the previous method is that, as noted, the spatial position of each voxel is not used during the segmentation. Spatial homogeneity is often a powerful constraint that can be used to improve the segmentation in the presence of image noise. This can be loosely thought of as finding the label at each voxel that is an optimal constraint between (i) the intensity at that voxel and (ii) the labels assigned to its neighboring voxels.

**Markov Random Fields:** The probabilistic structure used, most frequently, to capture such homogeneity is to model the label (classification) image as a Markov Random Field. This (and we are skipping a lot of math here) basically reduces to describing the probability of each voxel belonging to class  $l$ , as having a Gibbs distribution of the form:

$$p(l(\mathbf{x})) = k_1 \exp(-W(L(\mathcal{R}_{\mathbf{x}}), l)) \quad (9.2)$$

where  $k_1$  is a normalization constant,  $L(\mathcal{R}_{\mathbf{x}})$  is the set of labels of the neighboring voxels and  $W$  is a positive semi-definite function. This is a way of converting an “energy-function” like smoothness term into a probability distribution for integration into a probabilistic framework. The function  $W$  can take many forms, the one we will use here is from Zhang et al TMI 2001 [129] (the method in FSL):

$$W(l(\mathbf{X}_n)) = \sum_{\mathbf{x}' \in \mathcal{R}_{\mathbf{x}}} \delta(l(\mathbf{x}') - l(\mathbf{x})) \quad (9.3)$$

This essentially counts the number of voxels in the neighborhood of the voxel at location  $\mathbf{x}$  that have labels different from it.

**Overall Formulation using the Expectation-Minimization Framework:** We first define the vector  $\Theta$  as the collection of the means and standard deviations of all  $C$  classes, i.e.  $\Theta = [\mu_0, \sigma_0, \dots, \mu_{c-1}, \sigma_{c-1}]$ . The goal of the segmentation is to estimate both the set of labels  $L$  and the class parameters  $\Theta$ , given the image  $I$ . We can express this mathematically as:

$$\hat{L}, \hat{\Theta} = \arg \max_{L, \Theta} p(L, \Theta | I) \quad (9.4)$$

As is commonly done, this can be solved iteratively in the same spirit as the EM-framework as:

$$\mathbf{E}\text{-Step: } \Theta^k = \arg \max_{\Theta} p(\Theta | I, L^{k-1}), \quad \mathbf{M}\text{-Step: } L^k = \arg \max_L p(L | I, \Theta^k) \quad (9.5)$$

where  $k$  is the iteration number. In the E-Step we estimate a new set of parameters  $\Theta^k$  given the current classification  $L^{k-1}$ . In the M-Step, using the newly estimated  $\Theta^k$  we estimate a new classification  $L^k$ .

*E-Step:* This is straightforward. For each class  $i$  we estimate the mean and standard deviation of the intensities  $I$  of all the voxels where  $M = i$ . This is identical to the procedure used in the k-means algorithm above.

*M-Step:* This takes the form of a Bayesian a-posterior maximization. First we express

$$\hat{l}(\mathbf{x}) = \arg \max_l \log p(l(\mathbf{x}) | i(\mathbf{x}), \Theta^k, L(\mathcal{R}_{\mathbf{x}})) + \underbrace{k_1 + \log p(i(\mathbf{x}), \Theta^k | l(\mathbf{x}))}_{\text{Data Adherence Term}} + \underbrace{\log p(l | L(\mathcal{R}_{\mathbf{x}}))}_{\text{MRF Smoothness Term}} \quad (9.6)$$

where  $k_1$  is a constant. This equation is easily maximized by a greedy search strategy as  $M$  can only take values of  $1 \dots C$ . The prior term on the classification,  $p(L)$ , can be defined by modeling  $L$  as a Markov random field (see discussion above and equation 9.3). We express the likelihood (or data-adherence) term for each possible value of  $l(\mathbf{x}) = c$  as:

$$p(i(\mathbf{x}), \Theta^k | l(\mathbf{x}) = c) = p(i(\mathbf{x}) | \Theta^k, l(\mathbf{x}) = c) \quad (9.7)$$

which is similar to the model previously used in equation 9.1.



## Chapter 10

# Linear Registration

**What is Registration?** Image registration is the process of calculating realigning and warping factors that transform one image into the space of another. This allows for comparisons of corresponding regions of different images, as well as the creation of overlays. This function is important since acquisitions via different methods (MRI, SPECT, CT, etc.) result in differently sized, spaced, and oriented images. Furthermore, the comparison of images from different subjects, or a single subject before and after a morphological change, requires realignment of corresponding structures in order to extract meaningful information. The BioImage Suite software provides a variety of user-directed functions to perform these calculations, allowing you to register images in the manner most efficient and relevant to the information you are trying to obtain.

**Transformation Types** A transformation is the result of image registration; it is simply some function that maps a point in one image to a point in another. There are many different types of functions that can be used to specify this mapping, with varying degrees of associated flexibility, reliability and computational load.

*Rigid Transformations:* Rigid transformations are used for the registration of images from the same patient that have no distortion, and need to be realigned to the same orientation for meaningful comparisons to be made. A rigid transformation comprises 3 rotations and 3 translations. Therefore, it is a linear operation, and can be fully expressed in a 4x4 matrix. This type of registration is most often the fastest to compute.

*Affine Transformations:* Affine transformations are a broader class of linear transformations than rigid transformations, in that they include parameters for stretches and shears as well as rotations and translations. Nonetheless, they can still be represented by a 4x4 matrix, and are relatively quick to compute. Thus, affine transformations are typically used as a crude approximation to nonrigid transformations, either for rough estimates of location, or as a preliminary step.

*Nonrigid or Elastic Transformations:* Nonrigid transformations are used for registrations between images of different subjects, or images with distortions or actual physical differences (images of a patient's brain before and after an operation, for example). These transformations are non-linear,

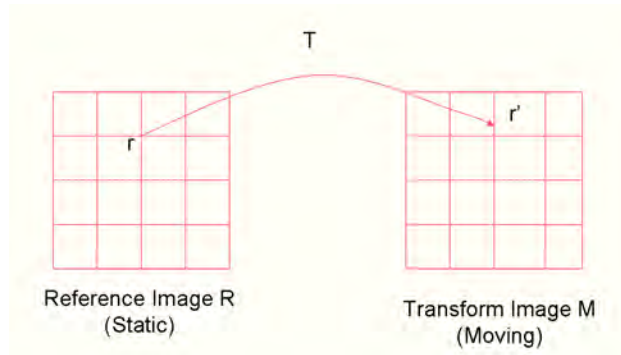


Figure 10.1: The Image Reslicing Process as implemented in `vtkImageReslice`

and thus have no matrix representation. In fact, they have a great many different parameterizations, the results of which get saved in a large grid file.

**Reslicing Images** Reslicing images is at the heart of most image registration procedures. While transforming surfaces is intuitive, and can be summarized in the three steps (i) take point, (ii) transform point and (iii) store point, image reslicing is somewhat counter-intuitive.

We will explain the process with reference to figure 10.1. In a typical registration process we have a Reference image and a Target image. The registration estimates the transformation *FROM* the Reference image *TO* the target image. This transformation can then be used in an image-reslicing operation to warp the Target image *BACK* to the Reference image, i.e. make the target look like the reference. In this way, while the transformation is “forward” the image moves **BACKWARDS**.

The process can be described by the following recipe, once the transformation exists:

- Create an empty image (often having the same geometry as a Reference Image).
- For each point (voxel)  $r$  in the empty image:
  1. Compute its corresponding point in the Target image  $r'$ ,  $T : r \mapsto r'$ .
  2. Interpolate the target image to find the image intensity  $I$  at position  $r'$  – which rarely corresponds to an exact voxel.
  3. Set the voxel  $r$  in the empty reference image to have intensity  $I$ .
- Repeat for all voxels.

## 10.1 Accessing the Registration Tools

Image registration functionality in BioImage Suite is available through the **Brain Register** or **Mouse Register** applications, as well as through the Data Tree Manager Tool. In both cases, the





Figure 10.2: **The Brain Register application.** This is a dual-viewer application, where the viewer on the left is termed the “Reference Viewer” and image on the right is called the “Transform Viewer”. All estimated registrations go from the “Image” in the Reference Viewer to the “Image” in the Transform Viewer.

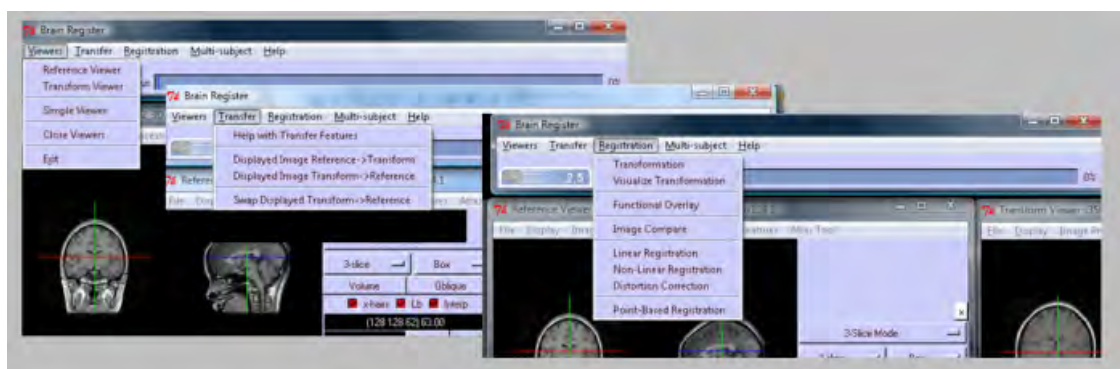


Figure 10.3: **Menu options for the main Brain Register application.** The *Viewers* menu (left) provides access to the three viewers, the Reference, Transform and Simple Viewer. The Simple Viewer is a Mosaic Viewer and can be used to view images from either the other two viewers. The *Transfer* menu can be used to move images from the Reference to the Transform viewer and vice-versa. The *Registration* menu brings up the options for performing linear and non-linear registrations as well as examining the registrations and other useful functions

GUI consists of two viewers (the “Reference Viewer” and the “Transform Viewer”) and a floating menu bar (Figure 10.2) that is linked to both viewers and mediates registrations across them.

The idea behind the two viewer style of display is to show the relationship between two images, as defined by the transformation between the two. Thus, when the “x-hairs” box is checked in both viewers, if you move the crosshairs in any view style in either viewer, the crosshairs in the other viewer move as well, to reflect the corresponding point in its image, according to the current transformation matrix that has been loaded, as described here. (Unchecking the “x-hairs” box in either viewer disables the movement of the crosshairs in that viewer, allowing you to lock them in place, while you navigate in the other viewer. This will be useful in manually defining crude initial transformations; more on this below.) The main menu of these applications has four submenus, as shown in Figure 10.3.

- Viewers – opens either the reference or transform viewer, both of which are orthogonal viewers by default but provides access to the Simple Viewer (an instance of the Mosaic Viewer) which can display multiple identically oriented slices. See the viewers page for more details on individual viewer functionality. Both the Reference and the Transform Viewer also have the full complement of Image Processing and Segmentation functionality.
- Transfer – commands for moving images from the reference to the transform viewer and vice versa or to swap them.
- Registration – provides access to the Registration/Overlay tool and the Point Based Registration controls.
- Multisubject – a dedicated Multisubject control that has functionality for generating composite functional maps. (The `pxitclDualMultisubject` app has a second multisubject control that facilitates group comparisons).

## 10.2 Registration — Transformation

**Loading and Saving Transformations:** In the menu bar shown in Figure 10.2, choose (Registration — Transformation) to bring up the Registration/Overlay Tool. The Registration/Overlay Tool window contains the tools needed to register and compare a pair of images. All transformation and registration operations use the Reference/Transform convention described in the box above. Transformations are generated FROM the Reference image TO the Transform image, and when loading a transformation, you should remember this scheme. The top section of the Transformation window in Figure 10.4, above, is a simple file load/save mechanism which allows you to save the transformation that is currently defined between the reference image and the transformation image. The 4x4 matrix below the filename reflects the transformation, if it’s linear. The **Load** button brings up a file selection dialog box that lets you load a previously saved transformation file of types Matrix (*\*.matr*), Grid/Combo (*\*.grd*), Thin Plate Spline Transforms (*\*.tps*), and Old Matrix Files (*\*.mat*).<sup>1</sup> When you load a transformation, the info box showing the transformation

---

<sup>1</sup>This last type, *\*.mat*, is the same format as *\*.matr*, but with a different suffix. This has been phased out due to the obvious potential confusion with MATLAB *.mat* files, which are not used by BioImage Suite.

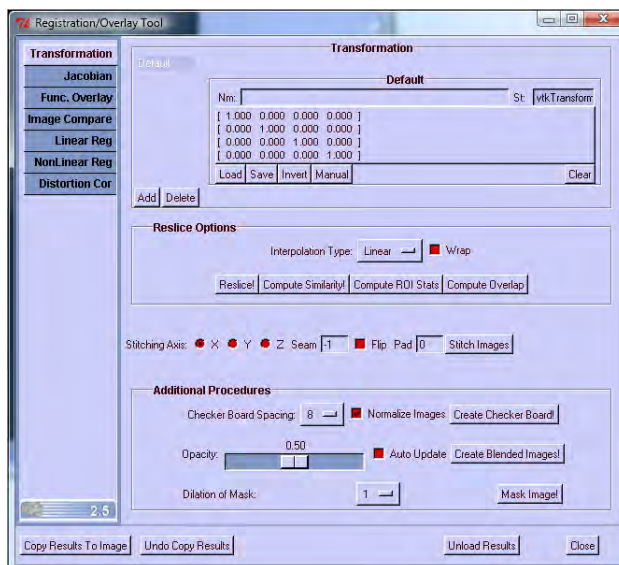


Figure 10.4: The Transformation Control Window. This window, a tab in the Registration/Overlay Tool, contains functions related to the loading, saving, and verification of transformations, as well as reslicing and stitching functions. It can be accessed directly from the (Registration — Transformation) menu command in the `pxitclbrainregister` menu bar.

matrix changes to reflect the parameters of the transformation. If the loaded file contains a linear transformation matrix, this is shown; if the loaded file is a Grid or Combo file, specifying a non-linear transformation (`*.grd`), then the box shows the Grid Dimensions and Spacing, as well as the linear transform matrix dimensions for Combo transformations.

## Transformation List

The listbox at the left side of the “Transformation” tool space is a holding space for multiple transformations, which can be loaded into memory simultaneously, but applied one at a time to the images in the viewers. When a new transformation is calculated, it will appear here. You can manually add a transformation (using the **Add** button), or delete a transformation (using the **Delete** button) to/from memory.

## Clear

Hitting the **Clear** button removes whatever transformation is currently selected in the Transformation List, and replaces it with the identity transform. This is reflected in the matrix display.

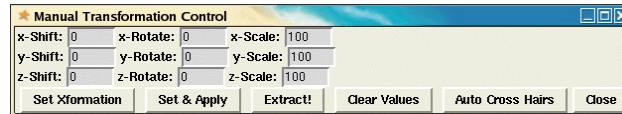


Figure 10.5: The Manual Transformation Control Popup. This box contains tools for manually altering the translation, rotation, and scale parameters of a transformation matrix.

## Inverse

With a transformation in memory (i.e. selected in the Transformation List), hitting the **Inverse** button computes the inverse transformation, and replaces the current transformation with it.

## Manual Transformation Control

You can use this tool to manually change translation, rotation, and scaling factors in all three dimensions. The tool can be accessed by clicking the **Manual** button underneath the matrix display in the transformation control window (See Fig 10.4) which will bring up the small manual control popup (Fig 10.5) containing input fields for the above parameters in each dimension. The fields will contain the current values for these parameters, which you can edit freely. The X-, Y- and Z- shift parameters are in the native scale of the reference image; the rotation parameters are in degrees; the scaling parameters are percentages. After inputting the desired values, hit the **Set Xformation** button to compute a 4x4 linear transformation matrix from your values, and send it to the transformation control. The transformation matrix will update to show your changes. This will not affect the images, however. To apply the transformation, you must hit the **Copy Results to Image** in the transformation control window. Equivalently, and as a shortcut, you can hit the **Set & Apply** button in the manual transformation control popup, which will directly apply your changes to the transformation and apply the result to the image.

The **Extract!** button lets you grab the shift, rotation, and scale parameters from the transformation matrix currently loaded in the transformation control. This way, you can load a transformation as described above, and alter it manually. To reset all values to their defaults (0 translation, 0 rotation, 100% scale), press the **Clear Values** button.

**Auto Crosshairs:** Sometimes it is easier to define a translation by placing cursors than by estimating numbers. The **Auto Crosshairs** button gives you this capability. If you can locate the same point in both of your images, you can use it to direct the manual setting of the translation parameter. In the viewers, simply place the cross hairs over the point in one image, uncheck the “x-hairs” checkbox in that viewer to lock them in place at that point, and place the crosshairs over the same point in the other image. Then, back in the manual transformation control popup, click the **Auto Cross Hairs** button, and the X-, Y- and Z- translation parameters will be updated to specify a translation that maps the two cross hair locations together (i.e. maps the crosshair point in the *Transformation Viewer* onto the point in the *Reference Viewer*). This technique is often easier than guessing at translation parameters and then navigating in the image to confirm them.

The manual transformation control popup is most often used as a rough alignment tool, as a

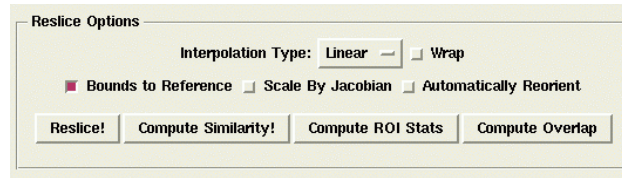


Figure 10.6: The Reslice Options Section in the Transformation Control.

preprocessing step to automatic registration. Besides reducing subsequent registration processing time, it also makes the registration more reliable, since corresponding regions are in closer relative proximity, increasing the likelihood of their overlap, and helping the registration algorithm to converge properly. It is strongly advised that you make an attempt to line your images up using the manual control if they are in radically different orientations or scales.

**Reslicing:** Once you have registered two images, you can put the resulting transformation information to work by “Reslicing”. Reslicing takes the loaded transformation matrix or grid and applies it to the Transformation Image, so that it “looks like” the Reference image. That is, it places the Transformation Image into the space of the Reference Image. This means that the Transformation Image gets diced up, analyzed, and rebuilt using voxels that are the same size as those of the Reference Image, and occupy an equally sized and shaped image volume. This process is wholly dependent on the transformation matrix that links the two images. To do it, simply hit the **Reslice!** button in the transformation control window. The image in the transformation viewer will be resliced into the space of the image in the reference viewer, and the output will be displayed (saved in the “Results” display of the transformation viewer).

This section contains the functions for reslicing images into other images’ spaces, and various options that relate to this operation. See also Figure 10.6.

**Reslice Options:** There are a few options that apply to the Reslicing function, which you can control in the “Reslice Options” section of the transformation control: **Interpolation** Choose the method for calculating values between those values that coincide in spatial location with the original values. Since the new image comprises differently sized voxels that occupy a new overall space, some method of interpolation must be employed. You may choose **Nearest Neighbor**, **Linear**, or **Cubic**.

**Wrapping:** If the “Wrap” checkbox is checked, the image will be allowed to wrap around the volume (i.e. anything that falls off the left side will appear back on the right side, etc.).

**Computing Measures of Registration Quality:** The other three buttons in the “Reslice Options” section are functions that compute measures of quality after a registration has been computed. Thus, after a transformation is loaded in the transformation control (either by clicking the **Load** button, or by having just completed a registration), these functions will yield a number of different numerical parameters that describe your registration. All operations are comparisons between two images. **Similarity** Clicking the **Compute Similarity!** button brings up a console

window that contains values for a number of measures of registration quality: Joint Entropy, Correlation, Normalized Mutual Information, Sum of Squared Differences, and Difference Entropy. These provide a quick diagnostic of how reliable the transformation loaded is.

**ROI Stats:** ROI stats give information about mean and standard deviations within discrete regions defined by a *Region Of Interest* image, which is a mask image that is registered to the reference image. Thus, you should load your ROI definition image into the Transform Viewer, your reference image in the Reference Viewer, and register them. Then click the **Compute ROI stats** button to get statistical measures out for each region in the ROI definition image.

**Overlap:** The **Compute Overlap** button thresholds both loaded images at 50% of their maximum value, and then computes the percentage of overlap between the resulting regions. This is very useful with binary valued images, but can also be a good diagnostic of registration quality in other images.

**Stitching:** The stitching function allows you to combine two adjacent images, provided that you are able to register them first. Thus you need a region of overlap between the two images. The result is a single image. The x, y, and z radio buttons let you choose which dimension to stitch in. The “Flip” checkbutton option lets you toggle which image comes first (i.e. lies at the top, left, or front, depending on the dimension of stitching). The “Pad” value is editable, and lets you specify how much to grow the image by before stitching (since you need to make the image bigger to fit both pieces into it).

**Additional Procedures:** This section contains a few extra functions that are associated with reslicing, and may be useful for evaluating the quality of registration operations. Often, the quickest way to check your registration is to create a blended or checkerboard image, and be sure that structures that cross image boundaries remain reasonably continuous. All of these operations apply the current transformation loaded into the transformation control.

**Checkerboard** Creating a checkerboard does just this: creates a combined image, in which alternating cube-shaped sections are contributed by each of the two images registered together. The “Spacing” option lets you control the size of the checkerboard spaces. The “Normalize Images” box should usually be left checked, since it normalizes the intensity scales of the two images, so that one set of checkerboard spaces does not appear much brighter than the other. Figure 10.7 left image shows an example of the checkerboard image.

**Blended Images** A blended image has contributions from both images, but they are complementary at every point. The effect is that both images become semi-transparent and get merged. Adjusting the “Opacity” slider button lets you adjust which image dominates in the result. A value of 0.30 means that the image is contributed to 30% by the image in the transformation viewer and 70% by the image in the reference viewer. A value of 0 would result in an image identical to that in the reference viewer. Check the “Auto Update” box to have the blended image result update as you move the opacity slider. Figure 10.7 right image shows an example of a blended image being used to combine a whole brain MRI with a scout slice MRI.

**Masking** Masking is the process of creating a binary image, i.e. an image that contains only



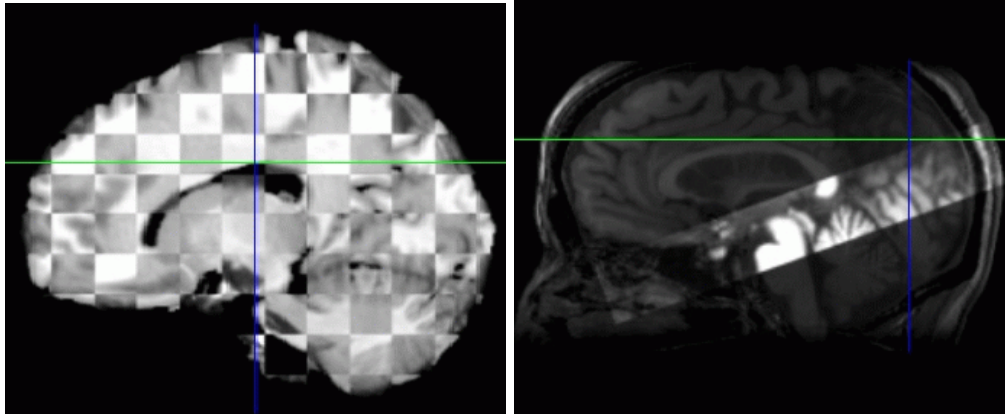


Figure 10.7: **Combining Images as a test of Registration Quality.** The combined images provide a quick diagnostic of the quality of the registration between two images. **In the left:** shows a checkerboard image, created by merging alternating cubic sections of two images. Areas discrepancy can be easily picked out, and areas that line up are also readily apparent (Trace the contours of the corpus callosum, for example: They are fairly continuous, even as you cross from one square in one image to the next in the other image). **On the right:** shows a blended image, created from a whole brain MRI and a scout slice MRI that has been resliced into the whole brain space. The image comprises 90% scout slice image and 10% whole brain image (the scout slice has no data outside the narrow band seen, thus outside this region only the whole brain image is visible). Again, by looking at the interface between the two regions, continuities and discrepancies are easily seen.

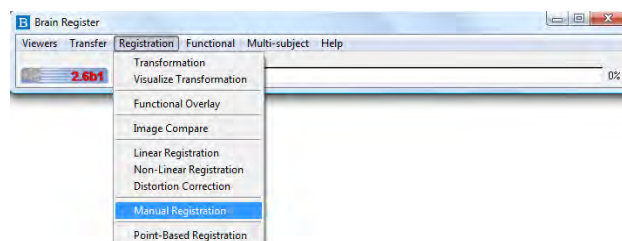


Figure 10.8: Manual registration tool choice in the menu.

two distinct values. Thus separate regions are created: those which have the value zero, and those which have the non-zero value. The **Mask Image!** button thresholds the image in the Transformation viewer at 50% of its maximum value, and then sets any voxels above the threshold to 100. This creates distinct regions in the image, creating a mask that can be used for region of interest definitions. The “Dilation of Mask” option allows you to specify how much to grow the mask image by after thresholding.

### 10.3 Manual Registration

Other than allowing users to manually control the parameters for registration, we also provide some visual metrics to help users pick parameters that can then be used for registration.



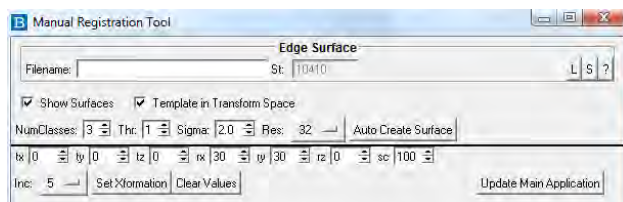


Figure 10.9: Manual registration tool choice in the menu.

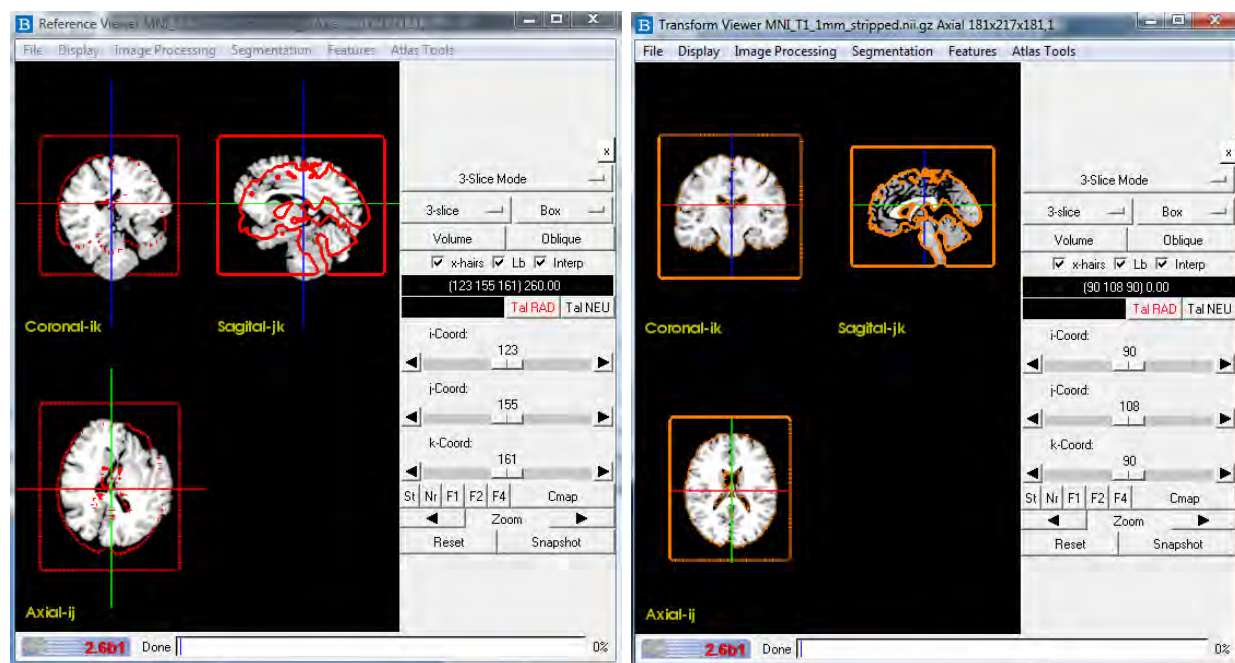


Figure 10.10: The Reference viewer and the Transform viewer with the surfaces overlaid to allow for control of registration parameters.

On invoking this tool, the Transform Viewer and the Reference Viewer windows are shown to the user. On loading the appropriate images into the transform viewer and the reference viewer, the user can then select the “Manual Registration” option in the menu as shown in Figure 10.8. On selecting manual registration, a “Manual Registration Tool” window pops up that allows the user to control parameters for manual registration. Figure 10.9 shows a screenshot of the tool. The “Show Surfaces” checkbox toggles the display of surfaces overlaid on the images. When the “Template in Transform Space” checkbox is checked/selected, the transformations by changing parameters such as  $tx$ ,  $ty$ ,  $tz$ ,  $rx$ ,  $ry$ ,  $rz$  and  $sc$  (scaling) are applied to the surface in the “Reference viewer”. If the “Template in Transform Space” checkbox is unchecked, the transformations are applied to the surface in the “Transform Viewer.” The “L” and “S” buttons in the tool refer to Load and Save which allow loading and saving of transforms. The “?” button provides more details about the generated surface such as the Number of points and number of cells.

To use this tool, the user should first click the “Auto Create Surface” button which creates a surface around the images in both the Transform viewer as well as the Reference viewer, as shown in Figure 10.10. On obtaining surface overlays, the user can then tweak the parameters using the controls in the Manual Registration tool. The “Update Main Application” button will take the user to the

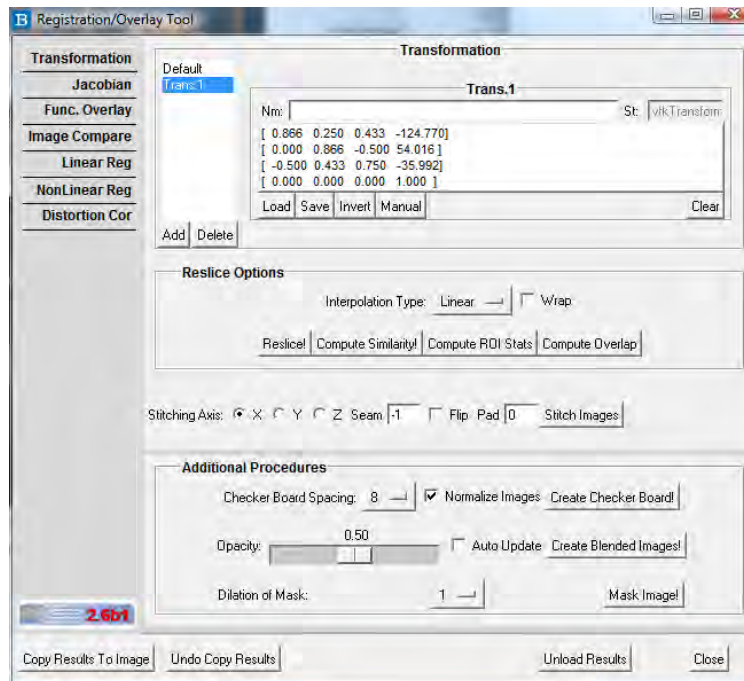


Figure 10.11: The transforms from the Manual Registration Tool such as translation, rotation and scaling are transmitted to the Registration/Overlay tool, which generates the transformation matrix based on the user selected transforms. Here we can see “Trans.1” shows the transformation matrix as per the selected transformations in Figure 10.9.

main registration tab and fills in the transformation matrix based on the user’s choices of rotation, translation and scaling transforms. Figure 10.11 shows a screenshot of the main registration tab with the transformation matrix filled in based on the transformations selected by the user in the Manual Registration Tool in Figure 10.9.

## 10.4 Linear Registration (Intensity Based)

Linear registration enables the computation of a transformation between two images that can be captured in terms of a 4x4 matrix (hence the term linear). The most common types are “Rigid” – simple translation and rotation which is useful in mapping images of the same subject that have no underlying distortion (both inter- and intra-modality, i.e. MRI to MRI or MRI to CT) and “Affine” which adds additional flexibility in terms of scaling/shearing of the image and is useful both as a crude distortion correction registration or for crude inter-subject registrations.

To perform a linear registration, first load the reference image in the “Reference” Viewer and then the target image in the “Transform Viewer”. The resulting transformation will map coordinates FROM the reference TO the target, and when it is applied for image reslicing will move the target image to the space of the reference – the overall goal of all registration being to move the target to look like the reference.

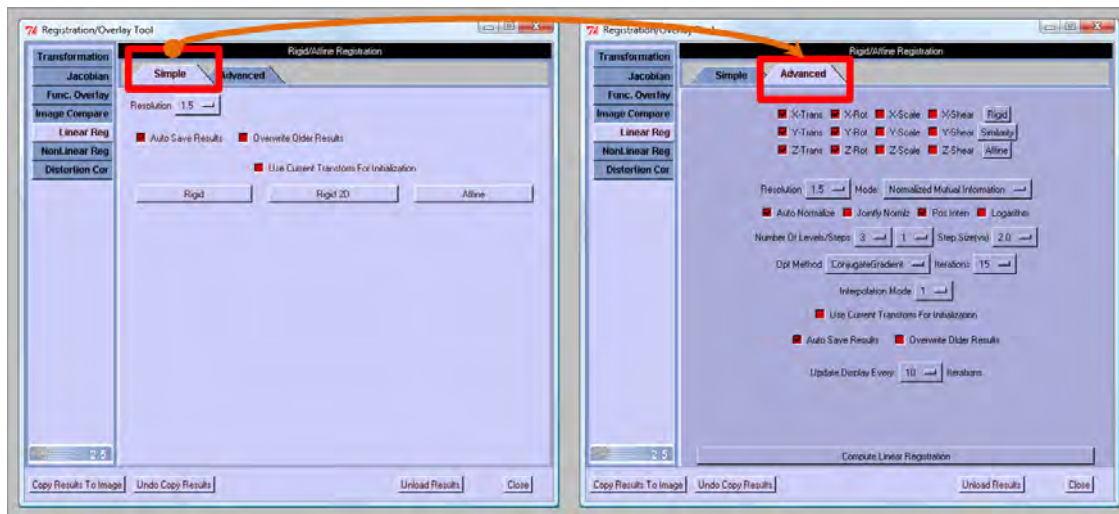


Figure 10.12: The Linear Registration Controls. Left: the most common options under the Simple tab. Right: advanced options.

There are two sets of controls. In the Simple Controls (Figure 10.12 top), the user specifies the resolution (as a multiple of the native resolution of the reference image), as well as whether to automatically save and overwrite the resulting transformation. If the “Use Current” transformation checkbox is enabled then the currently selected transformation in the Transformation List is used to initialize the registration parameters. If not then the transformation is initialized by mapping the “centers” of the two images. Three default operations are provided:

**Rigid** – compute a full 3D rigid mapping.

**Rigid2D** – compute a 2D rigid mapping.

**Affine** – compute a 12-parameter affine mapping.

Under the Advanced tab (Figure 8 bottom) the user can set additional parameters such as the similarity metric (default=Normalized Mutual Information), the optimization method etc. Key parameters are the number of levels (for multiresolution optimization = 3) and the number of steps (different step sizes for computing the gradient etc. = 1). The “Old Style” optimization method is that used in the original paper by Studholme et al [108]. If this is selected it is advisable to set the number of steps to 4 (default = 1).

## 10.5 Functional Overlay

The functional overlay control enables the overlaying of functional images onto anatomical images, to enable the combined display of structure and function. The anatomical image in this case is the Image (i.e. not results) of the Reference Viewer and the functional image is the Image of the Transform Viewer – the later is first resliced to match the anatomical image using the current transformation.

The basic principle used for the overlay is that the users sets a threshold for what constitutes



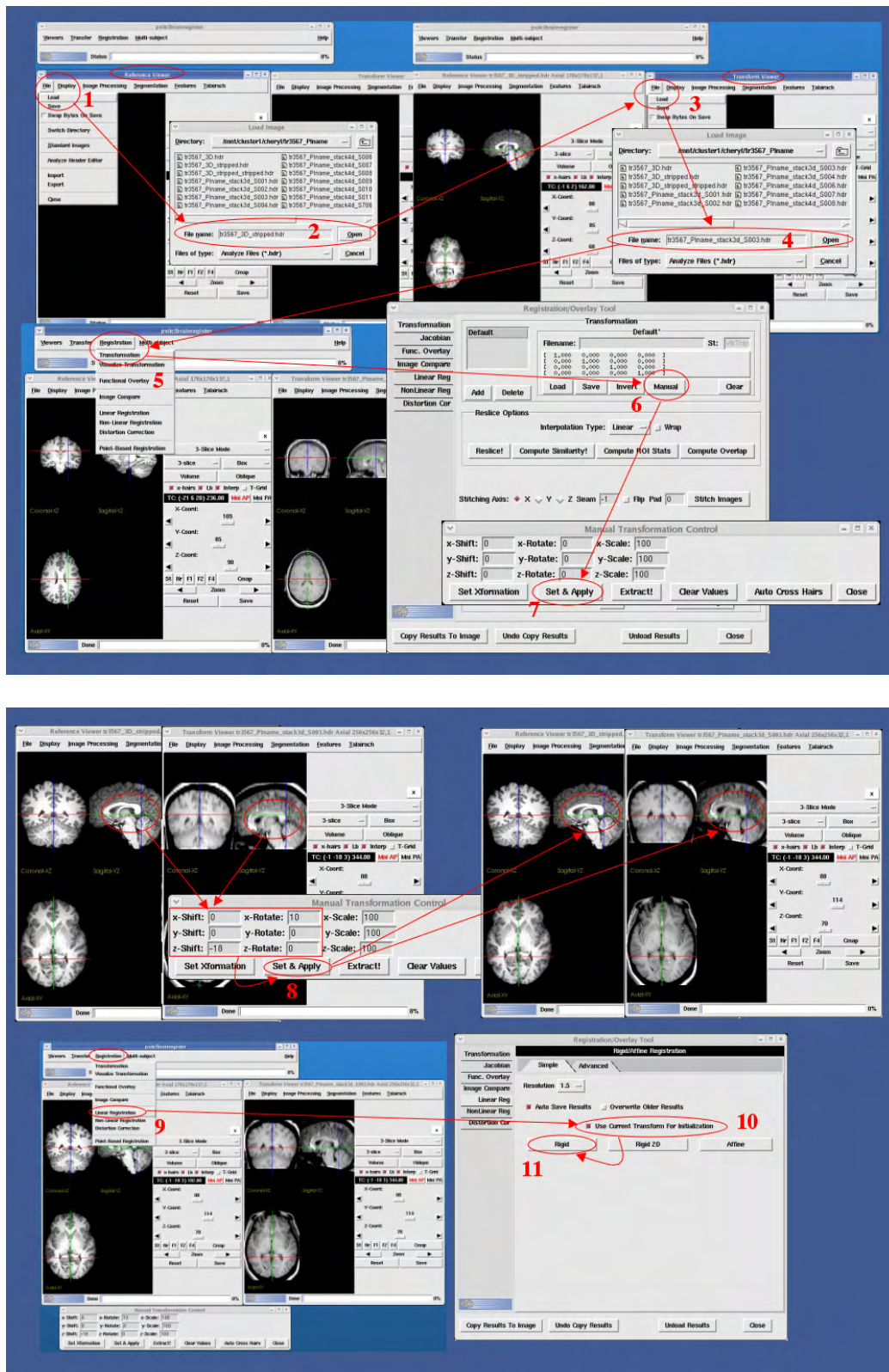


Figure 10.13: Methods used to manually transform an image to create a new starting point for the registration.

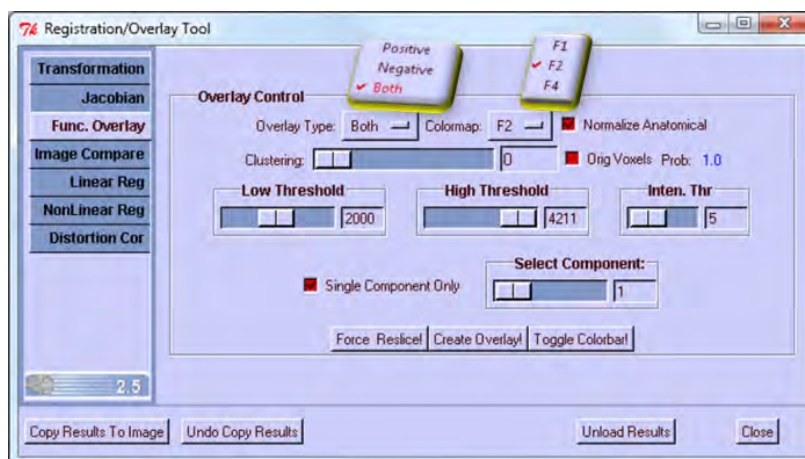


Figure 10.14: The Functional Overlay Control.

significant function using the Low Threshold slider and then saturates the functional data at the level set by the High Threshold slider.

Consider, for example, the case where the functional map has range -3000 to 3000 the anatomical image has range 0 to 256 and the F1 colormap is used. The F1 Colormap maps the anatomical image in the range 0 to 55, the negative functional data (if selected) in the range 56-59 (56 is the most significant) and the positive functional data in the range 60-63 (with 63 being the most significant). If the thresholds are set to 2000 and 2500 respectively, then the output of the overlay tool will be:

1. The anatomical image if functional image  $< 2000$  and functional image  $> -2000$  (i.e. insignificant activations), or if the anatomical image intensity (scaled in the range 0.55) is less than Inten Threshold. This last step masks spurious activations outside the brain or in the ventricles.
2. Otherwise
  - If function is positive and over 2000 then  $2000 \mapsto 60$ ,  $2500 \mapsto 63$  and anything over 2500 will saturate to 63.
  - If function is negative and under 2000 then  $-2000 \mapsto 59$ ,  $-2500 \mapsto 56$  and anything less than -2500 will saturate to -56.

*Additional Options:* The Overlay Type drop menu selects whether positive, negative or both positive and negative function is overlaid. The colormap is selected using the Colormap drop menu. Colormaps F2 and F4 perform similar mappings but use greater numbers of colors giving a higher fidelity overlay. The default colormap can be set in the User Preferences dialog. The Normalize Anatomical checkbox automatically windows the anatomical image prior to creating the overlay to improve its contrast.

The Clustering slider performs a cluster filter operation on the functional image. Only clusters greater than the selected size in voxels will be overlaid on the anatomy. By default, the number of

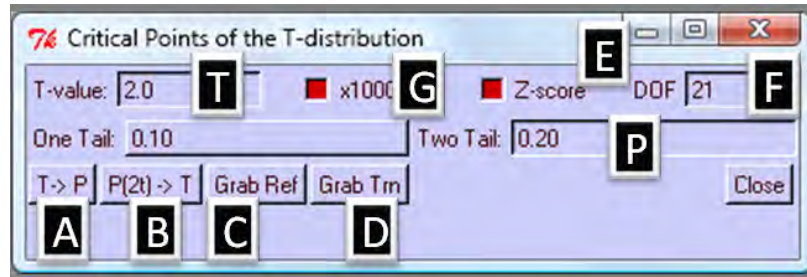


Figure 10.15: **The T-distribution Tool.** This is accessed under the Help menu of the main Brain Register application and can be used to convert from “t-values” to “p-values” and vice-versa. The T-value is entered into the box marked as **T**. This may be an actual t-value or one scaled by 1000 (if the checkbox **G** is enabled). It can also be set from the current position of the viewer cross hairs using the **C** or **D** buttons respectively. The number of degrees of freedom is specified in the entry marked as **F**. To convert to P-value press the button marked as **A**. The reverse conversion is accomplished by the button marked as **B**. (The control can also do z to p conversions – in this case enable the checkbox marked as **E**).

voxels is in the space (and resolution of the reference image. Enable “Orig Voxels” to scale this to the original resolution of the functional data.

If the input functional image is four-dimensional, i.e. there are multiple frames/tasks in it, the user can select to overlay only one frame by enabling the Single Component checkbox. The frame (component) is selected using the Select Component slider.

If all goes well the overlay is created using the CreateOverlay button. The Reslice button can be used to reslice and display the functional image only, if this is desired. The “Toggle Colorbar” button can be used to show a colorscale at the bottom of the viewer.

The t-dist distribution tool – see Figure 10.15 can be used to help set thresholds etc.

## 10.6 Image Compare

The Image Compare control can be used to either perform a straight image addition, subtraction and %change operations or compute a two-group t-test. It has four image controls labeled Mean1, Mean2, Standard Deviation 1 and Standard Deviation 2, where the images to be used for the comparison must be loaded.

There are three standard operations that only utilize the mean images namely:

- Mean1+Mean2
- Mean1-Mean2
- Intersection Threshold.

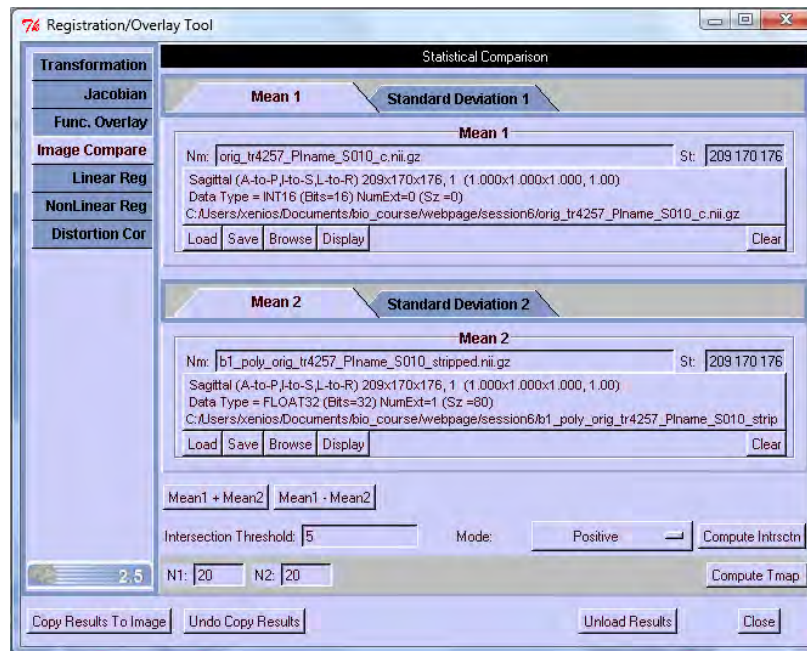


Figure 10.16: The Image Compare Control.

which result in an image placed in the Transform Viewer. Naturally, this operations can also be used to compare standard images (i.e. not means) provided that they have the same size!

Intersection Threshold performs a logical AND of the two images after first thresholding their values using the Intersection Threshold. When both images are above this, the output images has value equal to the average of the images, otherwise it is set to zero.

For the t-test computation, the standard deviation images and the size of the two groups must be also defined. The size of the two groups can be specified in the textboxes labeled N1 and N2 respectively at the bottom of the control. Invoking the Compute Tmap button will result in the computation of a voxelwise t-test between the groups represented by their mean and standard deviation images. The output image, which is equal to the t-score \* 1000 is displayed in the Transform Viewer. Conversion of these t-scores to p-values can be performed using the T-distribution table tool that is accessible under the Help menu of the main Brain Register application – Figure 10.15.

## 10.7 EXAMPLE: Interactive Registration Tools

The following instructions are for using the interactive graphical interface (GUI) to accomplish registrations between two brain images.

### 10.7.1 Co-register 2D conventional thick slice anatomicals to individual 3D

The step-by-step instructions, with reference to Figure 10.17 are:



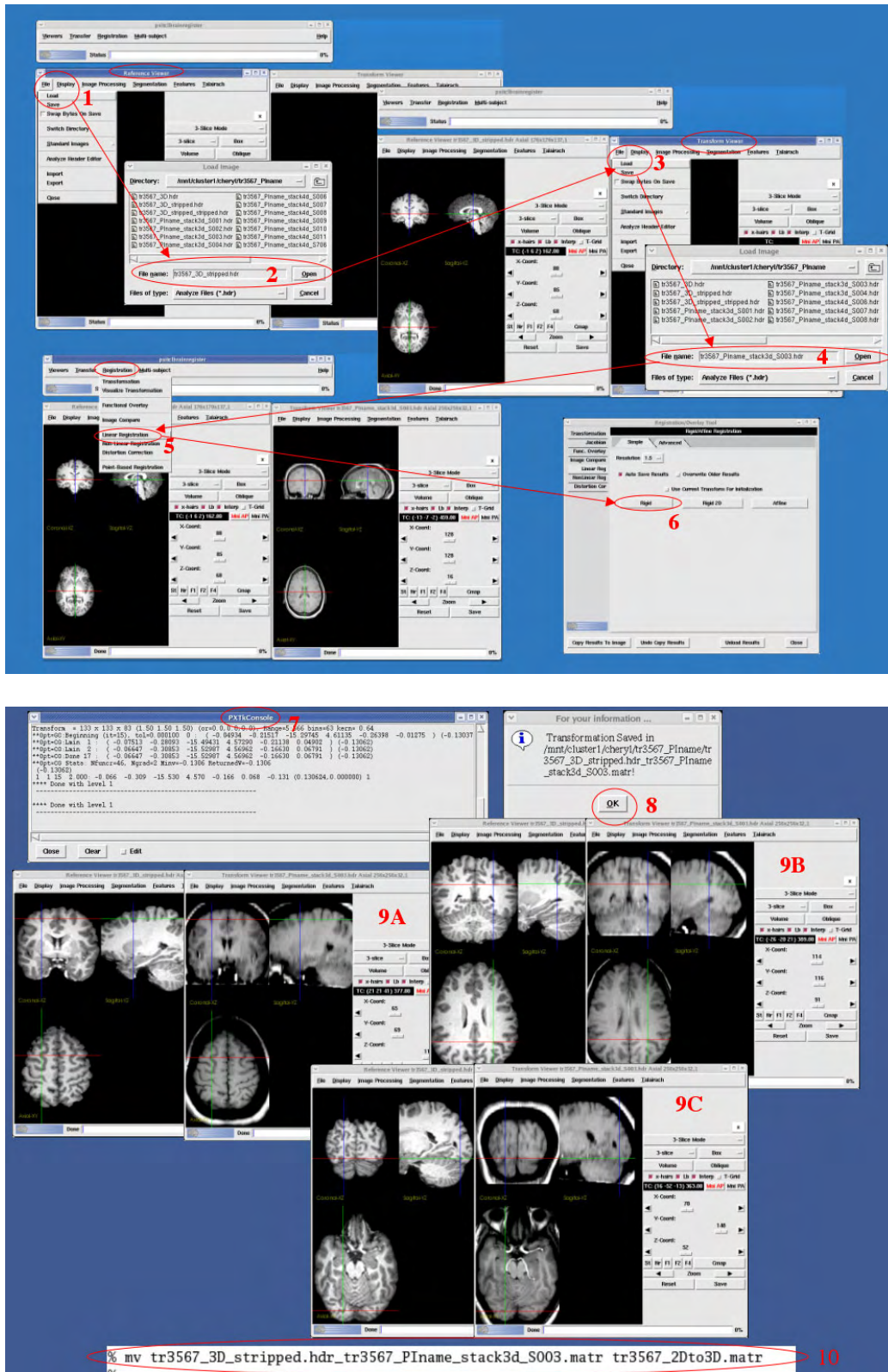


Figure 10.17: Methods used to Co-register a 2D conventional thick slice anatomical with a 3D wholebrain anatomical.

1. Choose brainregister from the BioImageSuite main menu. Three windows will appear: a transform viewer, a reference window and a brainregister menu bar. In the Reference Window choose (File — Load)
2. Choose the filename that refers to the stripped whole brain acquisition from the brain extraction steps explained above and click Open.
3. In the Transform Window choose (File — Load)
4. Choose the conventional 2D thick slice anatomical image and click Open
5. On the BrainRegister menu bar, choose (Registration — Linear Registration). A new Registration/OverlayTool window will appear.
6. In the new Registration/OverlayTool window click Rigid. This will compute a Linear Registration between the two loaded images. If the Auto Save Results box is red, BioImageSuite will automatically save the resulting .matr transformation file in the current working directory.
7. While the registration is running a PXTkConsole window will appear showing the status of the registration.
8. When the registration is complete a “For Your Information” window will appear telling you the name of the saved registration file. If the filename already exists the window will ask if you want to overwrite the existing file. Click OK.
9. Check the registration by clicking around in the 2D brain image (Transform Viewer) and seeing if the crosshairs match the same anatomical position in the 3D brain image (Reference Viewer). Three examples are shown - the gyral patterns should match on both images.
  - (a) Superior Frontal Gyrus
  - (b) Supramarginal Gyrus
  - (c) Middle Occipital Gyrus
10. Optionally in a unix window rename the newly generated .matr file to [studynum]\_2Dto3D.matr e.g.: `% mv tr3567_3D_stripped.hdr_tr3567_PIname_stack3d_S003.matr tr3567_2Dto3D.matr`

### 10.7.2 Co-register 4D echoplanar images to 2D conventional thick slice anatomicals

The step by step instructions with reference to Figures 10.18 and 10.19 are:

1. Choose brainregister from the BioImageSuite main menu. Three windows will appear: a transform viewer, a reference window and a brainregister menu bar. In the Reference Window choose (File — Load)
2. Choose the filename that refers to the conventional 2D thick slice anatomical image and click Open.
3. In the Transform Window choose (File — Load)

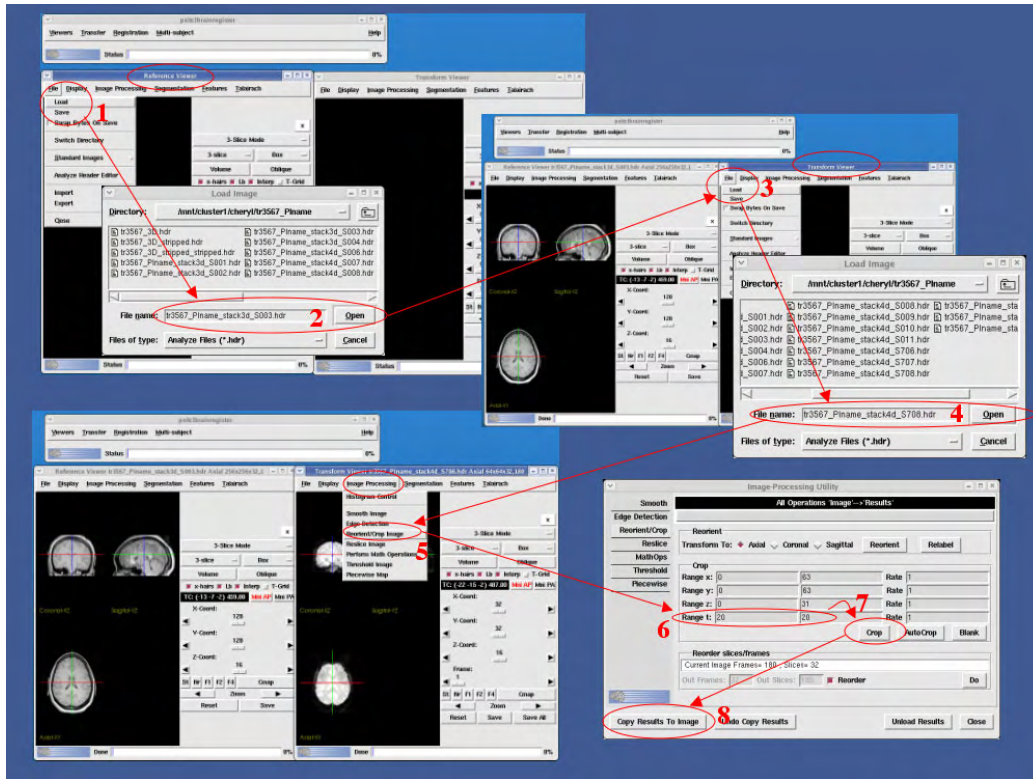


Figure 10.18: Methods used to Co-register a 4D echoplanar image with a 2D conventional thick slice anatomical. Part I.

4. Choose the middle trial of all echoplanar image files and click Open
5. In the Transform Window choose (Image Processing — ReOrient/Crop Image)
6. Choose a range t that will only be one image - this will increase the registration speed.
7. Click Crop
8. Click Copy Results to Image
9. There is a choice to be made as to which registration method should be used for the echoplanar to 2D registration. One can either choose to do a Linear Registration (LR), or because of the distortion found in echoplanar (i.e. functional) images once can choose to do a Linear Registration with Distortion Correction (LRDC). Both methods are described here.
  - (a) LR: On the BrainRegister menu bar, choose (Registration — Linear Registration).
  - (b) LRDC: On the BrainRegister menu bar, choose (Registration — Distortion Correction).
10. A new Registration/OverlayTool window will appear. If the Auto Save Results box is red, BioImageSuite will automatically save the resulting .matr or .grd transformation file in the current working directory
  - (a) LR: Click Rigid to compute the linear registration (a .matr file will be generated).
  - (b) LRDC: Click Compute Linear and Distortion Correction to compute the registration (a .grd file will be generated).



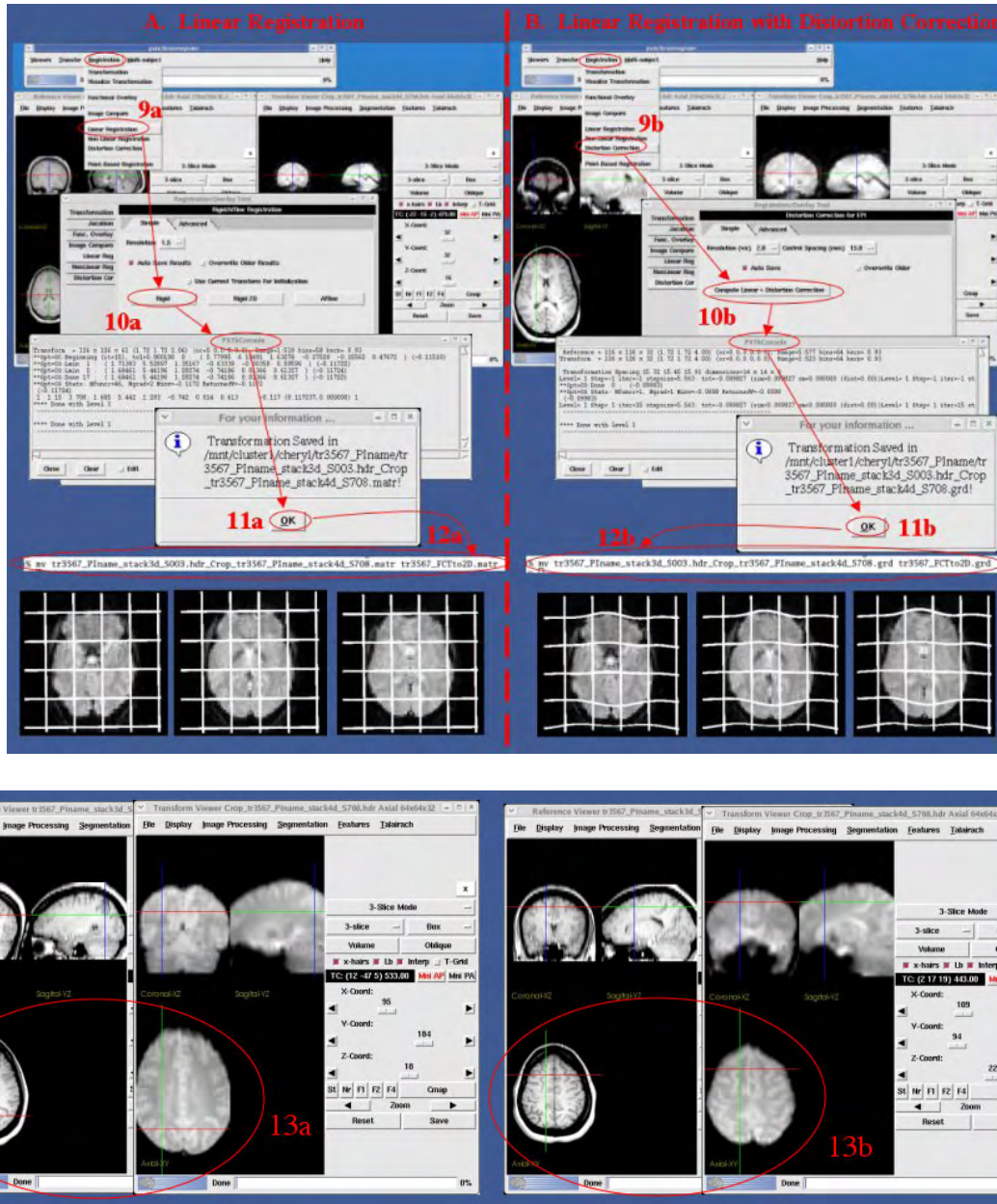


Figure 10.19: Methods used to Co-register a 4D echoplanar image with a 2D conventional thick slice anatomical. Parts II and III.

11. While the registration is running a PXTkConsole window will appear showing the status of the registration. When the registration is complete a “For Your Information” window will appear telling you the name of the saved registration file. If the filename already exists the window will ask if you want to overwrite the existing file.
  - (a) LR: Click OK
  - (b) LRDC: Click OK
12. In a unix window, rename the newly generated registration file. The six images at the bottom of Figure 7B were created using (Registration — Jacobian) and clicking Create. Comparing

these two sets of images illustrates the difference between the two transformation methods.

- (a) LR: ex: `% mv tr3567_PIname_stack3d_S003.hdr_Crop_tr3567_PIname_stack4d_S708.matr tr3567_FCTto2D.matr`
  - (b) LRDC: ex: `% mv tr3567_PIname_stack3d_S003.hdr_Crop_tr3567_PIname_stack4d_S708.grd tr3567_FCTto2D.grd`
13. Check the registration by clicking around in the echoplanar brain image (Transform Viewer) and seeing if the crosshairs match the same anatomical position in the 2D anatomical image (Reference Viewer).
- (a) LR: click around on different gyri to check if gyral patterns match on both images.
  - (b) LRDC: click around on different gyri to check if gyral patterns match on both images.

### 10.7.3 More Advanced Technique when Registrations are not Accurate

Sometimes if an image is quite rotated from the target, the automatic registration will not be accurate. The user can manipulate the transform image to approximate the target image using a Manual Transformation Control box. This approximation can then be used as an input starting point for the registration.

The step-by-step instructions, with reference to Figure 10.13, are:

1. Choose `brainregister` from the `BioImageSuite` main menu. Three windows will appear: a transform viewer, a reference window and a `brainregister` menu bar. In the Reference Window choose (`File — Load`)
2. Choose the filename that refers to the stripped whole brain acquisition from the brain extraction steps explained above and click `Open`.
3. In the Transform Window choose (`File — Load`)
4. Choose the conventional 2D thick slice anatomical image and click `Open`
5. On the `BrainRegister` menu bar Choose (`Registration — Transformation`). A new `Registration/OverlayTool` window will appear.
6. Under the Transformation block Click `Manual`. A separate `Manual Transformation Control` window will appear.
7. On the `Manual Transformation Control` window, with all values set to zero, click the `Set&Apply` button. This will center both images. Check the crosshairs in one window to see how different they are on the other.
8. Manually input shifts and rotations into the `Manual Transformation Control` window and click `Set&Apply` button to manipulate Transform window image. Repeat until the image in the Transform window approximates the image in the Reference window.

9. On the BrainRegister menu bar, choose (Registration — Linear Registration). A new Registration/OverlayTool window will appear.
10. Click Use Current Transform for Initialization, the box will turn red.
11. In the new Registration/OverlayTool window click Rigid. This will compute a Linear Registration between the two loaded images. If the Auto Save Results box is red, BioImageSuite will automatically save the resulting .matr transformation file in the current working directory.
12. While the registration is running a PXTkConsole window will appear showing the status of the registration.
13. When the registration is complete a “For Your Information” window will appear telling you the name of the saved registration file. If the filename already exists the window will ask if you want to overwrite the existing file.
14. Check the registration by clicking around in the 2D brain image (Transform Viewer) and seeing if the crosshairs match the same anatomical position in the 3D brain image (Reference Viewer).
15. Optionally, in a unix window rename the newly generated .matr file to [studynum]\_2Dto3D.matr

```
ex: \% mv tr3567\_3D\_stripped.hdr\_tr3567\_PIname\_stack3d\_S003.matr tr3567\_2Dto3D.matr
```

## 10.8 Linear Transformations Theory:

Transformations are maps which implement the following equation:  $x \mapsto y$ , or  $y = T(x)$ , where  $x$  is the input point and  $y$  is the output point. Linear transformations are represented as  $4 \times 4$  matrices. This enables the use of a single operation to capture both a translation as well as a combination of rotation/shear/scale. Ordinarily, we would write such a transformation in two parts as:

$$y = Ax + b \quad (10.1)$$

where  $A$  is a  $3 \times 3$  matrix that performs a combinations of rotation, scale and shear and  $b$  is a  $3 \times 1$  vector specifying the translation. A more compact representation is to use homogeneous coordinates. To accomplish this, we write each point as a 4-vector  $(x_1, x_2, x_3, 1)$ , and apply the transformation as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & b_1 \\ A_{21} & A_{22} & A_{23} & b_2 \\ A_{31} & A_{32} & A_{33} & b_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{bmatrix} \quad (10.2)$$

This method can be used to transform all linear transformations into linear algebra operations on  $4 \times 4$  matrices. This enables easy concatenation (matrix multiplication) and inversion (matrix inversion). Note also that a linear transformation can have at most 12-free parameters. There are 3 general types of linear transformations as follows:

1. Rigid – these have six parameters (3 rotations and 3 translations)

- 
2. Similarity – these have seven parameters, rigid + overall scale factor
  3. Affine – this is the general linear transformation group and has 12 parameters.



## Chapter 11

# Non Linear Registration

### 11.1 Introduction

A nonlinear registration ends up being is a displacement field between two images. In particular it takes a voxel in image 1 to a position in image 2. Some non-linear registration methods simply output directly the actual displacement field. Other methods (such as the one used in BioImage Suite) use a parametric representation for the displacement field based on spline models. These later type of methods enable a more compact representation.

BioImage Suite typically uses a tensor cubic-spline representation. These transformations are stored in a custom *.grd* file format and can be manipulated using both GUI tools (Brain Register) and command line tools – see Section 11.5 for more details on these.

### 11.2 Visualizing NonLinear Transformations

Visualizing and quantifying the effect of a nonlinear transformation is a difficult problem. Unlike linear transformations which are global in nature, nonlinear maps can have varying accuracy over the whole image. For example, in the case of a linear registration between two brains, if the map is accurate at a few dispersed points in the brain, we can be confident that it is equally accurate in the rest of the image. Nonlinear registrations on the other hand, can be highly accurate in some parts of the image and less accurate in others.

**Jacobian Maps:** A key property of nonlinear transformations is the determinant of the Jacobian of the transformation – often referred to as “Jacobian map”.<sup>1</sup> This is a measure of the local volume

---

<sup>1</sup>For those of you who remember your multivariate calculus: In multivariable integration we often have problems which are better solved by changing the variables. The classic case is mapping a problem from Cartesian  $(x, y)$  to polar  $(r, \theta)$  coordinates. In those integrations at the end of the conversion there is a scale factor  $|J|$  ( $=r$  in this case) which is the determinant of the Jacobian matrix between the two pairs of variables that quantifies the area (volume)

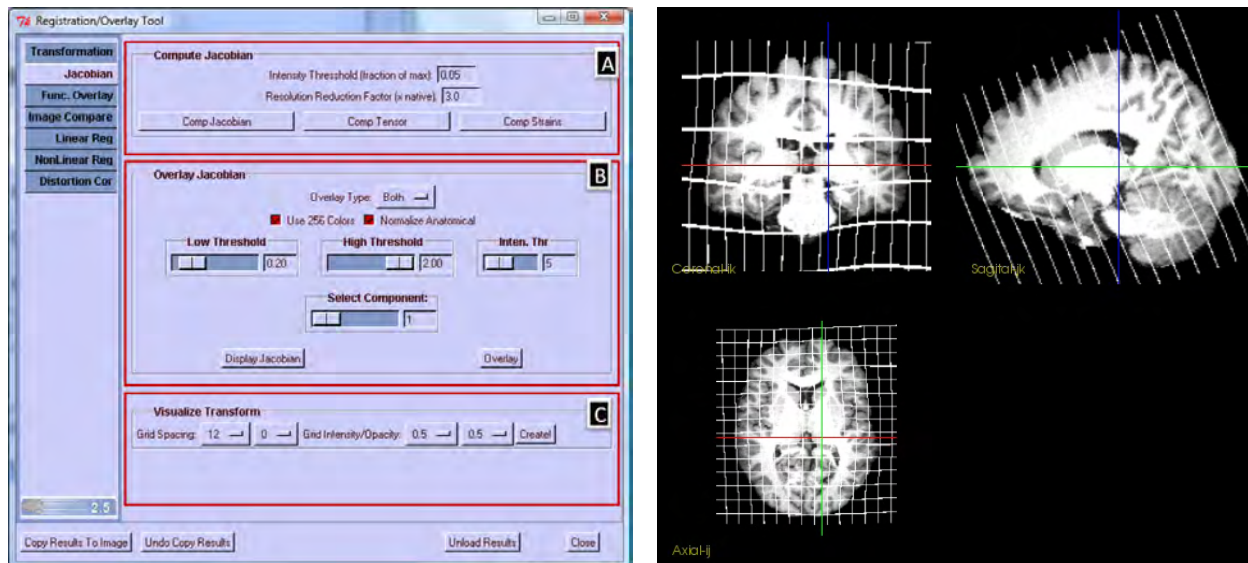


Figure 11.1: **Left:** The Visualize Transformations (Jacobian) Tab. **Right:** Superposition of a grid showing the effect of the transformation on the warped image. This is achieved using controls in the bottom box (C) of the tab shown on the left.

expansion required to map the reference image to the target image. If this number is equal to 1 then there is no volume change, if it is greater than 1 it implies that the target image is bigger than the reference image, whereas if it is less than 1 it implies that the reference image had to be shrunk to match the target. If this value goes below zero (or even close to zero), it is a sign of trouble. It implies that the transformation map has gone singular. In such cases, one needs to discard the transformation and repeat the registration with higher smoothness settings.

To quickly compute a Jacobian map for a transformation use the controls in Figure 11.1(left), Panel A. The two entry-boxes allow one to specify the resolution reduction factor (3.0 is the default) from the original image – this is useful for speeding up the process, the Jacobian maps are usually very smooth, so computing them at full resolution is not really needed, and the Intensity threshold factor (0.0 = do this everywhere). Both of these options can be used to speed up the process.

Three different combinations of results can be computed. The *Comp Jacobian* button simply computes the Jacobian map (as a percentage). The other two buttons *Comp Tensor* and *Comp Strain* store the whole tensor and eigenvalues of the Jacobian tensor respectively.

**Showing Grids over the Warped Image:** The Visualize Transform Box, marked as C in Figure 11.1(left), can be used to overlay a grid on the original image showing the effect of the transformation. The spacing of the grid and the intensity of the grid are controlled by the four option menus in the box. They are strictly used for visualization, so trial and error will often reveal the best settings. The new image which consists of a combination of the warped grid and the warped transform image is placed as “Results” in the transform viewer. See Figure 11.1(right) for an example.

ratio between the original small element ( $\delta x, \delta y$ ) and the mapped small element ( $\delta r, \delta \theta$ ). This is the same identical term, where in this case the two small elements are the original image voxels and the transformed image voxels.

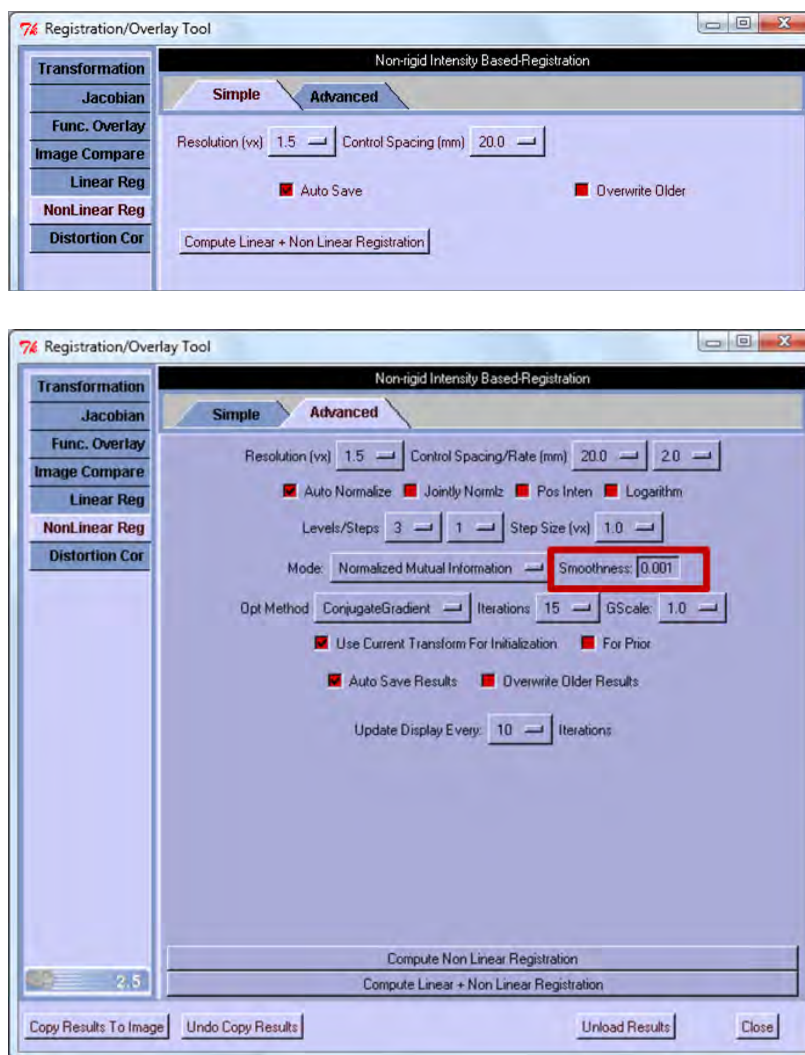


Figure 11.2: The Non-Linear Registration Controls. Above: the most common options under the Simple tab. Below: advanced options.

### 11.3 Nonrigid Registration (Intensity Based)

Non-Linear Registration allows for additional degrees of freedom and permits the more accurate mapping of images from different subjects into a common space. The methodology in this control derives from the work of Papademetris et al. [49] which in turn is closely related to work by Rueckert et al. [91]. The computed transformation is parameterized in terms of a tensor b-spline grid with uniform control point spacing (e.g. 20.0 mm in the example on the left).

To perform a nonlinear registration, just as in the case of the linear registration above, first load the reference image in the “Reference” Viewer and then the target image in the “Transform Viewer”. Again, similarly to the linear registration case, there are two sets of controls.

In the Simple Controls (Figure 11.2 top), the user specifies the resolution (as a multiple of the native resolution of the reference image), the control point spacing and whether to automatically save and

overwrite the resulting transformation. The registration is invoked by pressing the “Compute Linear + Non Linear Registration” which first computes an affine mapping followed by a nonrigid mapping. The resulting transformation when saved in the .grd file format includes both the affine and the non-linear components.

Under the Advanced tab (Figure 11.2 bottom) the user can set additional parameters such as the similarity metric (default=Normalized Mutual Information), the optimization method etc. Perhaps the most important of these parameters is the smoothness (the control for this is highlighted in figure 11.2(bottom)), setting this to a higher value ensures a more rigid mapping at the expense of less accuracy in the registration. Here the registration can be invoked in two ways. The first is as a straight non-linear registration using the “Compute Non Linear Registration” button – in this case either the current transformation is used for initialization or the linear component of the registration is assumed to be the identity matrix. The second is as combo Linear+Non Linear Registration in the same way as for the simple controls.

## 11.4 Distortion Correction (Single Axis Distortion)

The Distortion Correction functionality is essentially a constrained form of the non-linear registration in which the warping is assumed to be only in one direction. This reflects the case of distortion in echoplanar magnetic resonance acquisitions – such as those used for DTI and fMRI. The distortion direction is assumed to be along the “Y-axis” of the image (which is typical of the phase encode direction), if this is not the case for the specific images, the axis can be changed using the Phase-encode direction drop menu under the advanced controls (Figure 11.3 bottom). See Figure 11.4 for an example.

Most of the functionality of the distortion correction algorithm is closely related to the nonrigid registration methods and is shown in Figure 11.3. There are two key differences: (a) The resolution can be set to be lower i.e. 2.0 or 4.0 since typically the reference image (anatomical) has voxel size of  $\approx 1$ mm and the target (echoplanar) has voxel size  $\approx 4$  mm, in which case running the algorithm at 1 or 1.5 mm resolution is overkill. (b) if the Echo-planar image used is a spin-echo image with no signal loss the adjust drop menu can be used to enable the signal-loss conservation constraint (originally from the work of Studholme et al [107]) – this *may result* in better registrations.

## 11.5 Batch Mode Registration

### 11.5.1 A quick overview of GNU Make

All Batch Job Generators in BioImage Suite result in a makefile which needs to be processed with the Unix Make Utility (most commonly GNU Make). While “makefiles” are primarily used to compile large programs they are a great mechanism for general batch mode processing because they enable:

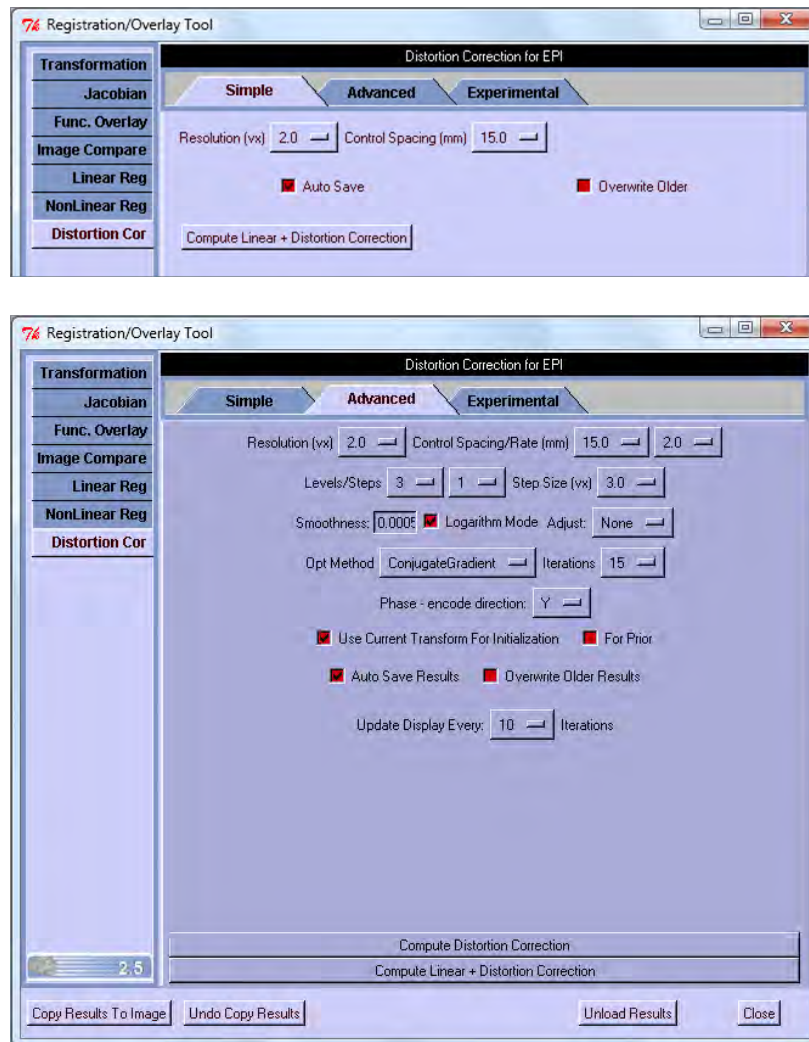


Figure 11.3: The Distortion Correction Controls. Above: the most common options under the Simple tab. Below: advanced options.

1. the definition of job dependencies – i.e. job A must be completed before job B
2. the use of multiple processors at the same time – i.e. run two or more jobs at once.
3. the ability to automatically recover from a system crash and to restart at the point that the job failed – i.e. no repeat computations.

The standard name for a makefile is unsurprisingly “makefile”, although in this context I recommend the use of more descriptive names e.g. “makefile.controls” etc. Given a makefile, the batch job is executed using

```
make -f makefile.controls
```

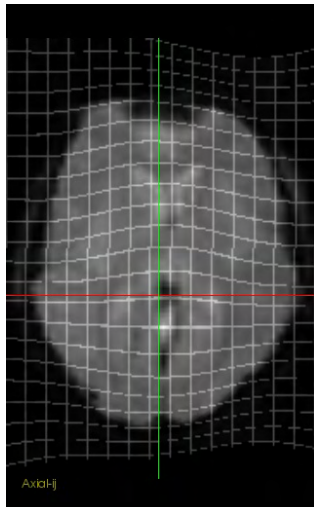


Figure 11.4: Visualization of a distortion correction transformation. This shows displacements only along the phase-encode (y) axis of the echoplanar image.

On Linux systems `make=gmake` i.e. typing `gmake` is equivalent to typing `make` – this may appear in some examples. Additional useful flags include

1. “-n” - do a dry run i.e. simply print a list of commands to be executed without doing anything

```
make -n -f makefile.controls
```

2. “-j” - specify how many jobs to run at once – typically equal to the number of processors available e.g. to use 2 processors type

```
make -j2 -f makefile.controls
```

In addition makefiles contain a number of “jobs” which may be explicitly specified. Typically the first job defined in BioImage Suite batch jobs is the equivalent of “do everything” so you need not worry about this - in some cases other specifications will need to be given . Note: If after starting a batch job – it is for some reason terminated (either by a system crash or a reboot or ..) it may be restarted by typing exactly the same command as the one used to originally start the batch job. The job dependency mechanism in `make` will ensure that no processing that was previously completed is re-run.

**Microsoft Windows Info:** Whereas `make` is a standard feature on most unix systems (including Mac OS X), to run batch jobs in MS-Windows you will to download and install GNU Make. A binary (from the UnixUtils distribution ) is included with BioImage Suite.

### 11.5.2 Single Reference Image Registrations

To compute batch-mode registrations to a single reference (typically inter-subject nonlinear brain mappings for the purpose of computing fMRI composite maps etc.) BioImage Suite provides the

pxmultiregister\_int.tcl tool. This is a batch-mode generator which will generate a makefile - the batch job itself is executed using the standard make program - see detailed description above. pxmultiregister\_int.tcl has a large number of options - with more to be added in the next release. Simply typing pxmultiregister\_int.tcl sample lists all options, which might be intimidating for the average user. A basic setup file has the form: (Lines beginning with # are treated as comments and may be added liberally to document the processing.

```
#Example Setup File
# all lines beginning with # are ignored
#
# Mode of Operation
set intensities_only 1
#
# List all images here
set imglist {
  Az2_2_sm0.8.hdr
  Az2_3_sm0.8.hdr
  Az2_4_sm0.8.hdr
}
# Put reference brain here
#
set refimg Az2_5_sm0.8.hdr
# Linear mode -- choices are "rigid", "similarity", "affine" --
# this is used to generate an initial transformation to be refined later.
set linearmode "affine"
# Tweak parameters for intensity based part
# (the ones below are typical for fMRI composite maps)
# Resolution is a scale factor x native
#(i.e. if reference image is 1.1x1.1x1.2 setting this to 1.5
# will resample the images to
# 1.65x1.65x1.7 mm). Integer values are not recommended due to
# artifacts in joint histogram computation, 1.5 or 1.2 are good values
set resolution 1.5

# Spacing defines the flexibility of the transformation --
# the gap between control points in the tensor b-spline model
# used for the non-linear transformation
# 15.0 mm is a good choice for composite functional maps
# For structural morphometric studies this should be reduced to
# 10 or 8 mm (with a corresponding increase in computational time)
# If the data is a fairly low resolution this
# can be increased to 20mm or so.
set spacing 15.0

# This is the maximum number of iterations for the
# Conjugate Gradient Minimizer (15 is usually a safe number)
set iterations 15

# Leave this to zero unless other instructed
# Regarding filetype: filetype = 1 includes directory name
# in output filename 0 does not
```



```
set filenamestype 0

# Set this higher if data quality is low
set smoothness 0.001

# If linearonly=1 then the resulting transformation will be linear
#           i.e. no warping will be attempted
set linearonly 0
```

Once the setup file is completed, the next step is to decide where the results (and a host of log files) will be stored. Typically this is in a subdirectory (e.g. results). If the setupfile is called controls\_setup and we want the results to go to controls\_results type:

```
pxmultiregister_int.tcl controls_setup control_results
```

This will check for the existence of all files in the setup file, if a file (image) is missing an error will be given. Once this completes OK, the next step is to generate the makefile using

```
pxmultiregister_int.tcl controls_setup control_results go > makefile.controls
```

At this point the batch-job is ready and can be started using the make utility described above.

### 11.5.3 Pairwise Image Registrations

The case of interest here is where one set of images (e.g. thick-slice “2D” conventional images) are to be mapped to another set of images (e.g. high resolution 3D anatomical images). To accomplish this use the pxpairwiseregister.tcl script. This is a batch-mode generator which will generate a makefile – the batch job itself is executed using the standard make program – see above for instructions as to how to use GNU make. pxpairwiseregister.tcl has a large number of options. Simply typing pxpairwiseregister.tcl sample lists all options, which might be intimidating for the average user. A basic setup file has the form: (Lines beginning with # are treated as comments and may be added liberally to document the processing.

```
#Example Setup File
# all lines beginning with # are ignored
#
```

```
#
#
# Put reference list here
#
set reflist {
  /data1/brains/1001/refA.hdr
  /data1/brains/1002/refB.hdr
}
#
# Put the target list here
#
set trglist {
  /data1/brains/1001/targA.hdr
  /data1/brains/1002/targB.hdr
}
#
#
# Type of registration
# mode = rigid,affine,nonlinear
set mode "nonlinear"
#
# Tweak parameters
# filetype = 1 includes directory name in output filename 0 does not
# defaults are for rigid/affine/nonlinear
set resolution 1.5
set spacing 15.0
set iterations 15
set stepsize 1
set smoothness auto
```

See the previous section for more description of the parameters. Executing `pxpairwiseregister.tcl` is also identical to `pxmultiregister_int.tcl`, i.e. first

```
pxpairwiseregister.tcl setup.txt results
```

then

```
pxpairwiseregister.tcl setup.txt results go > makefile.results
```

and then use the make utility to start the makefile.

### 11.5.4 Reslicing Images and other command line tool using transformations

The BioImage Suite registration tools (both command line and using the graphical user interface), produce transformation files as outputs, typically with a “.matr” or a “.grd” extensions. (See the File Formats Chapter for more details.) The command line tools described here can be used to apply these transformations to images (reslicing) or to manipulate the transformations in different ways.

**Image Reslicing** Given a reference image, a target image and an appropriate transformation (or set of transformations) the target image can be resliced in the space of the reference image using the command

```
pxmat_resliceimage.tcl reference_image target_image output_image
    interpolation_mode xform [xform2] [ xform3]

interpolation_mode = 0,1,3 none,linear,cubic
    (Avoid linear if the images are in the range 0.1)
xform = the .matr or .grd file
```

**Note:** The direction of the transformations is a common source of confusion. When computing registrations, the estimated transformations is **FROM** the reference **TO** the target. This transformation can be used in pxmat\_reslice.tcl to move the target image to the reference image. A good rule of thumb is to remember that images move in the opposite direction as transformations.

If multiple transformations are specified (upto 3) then the concatenation of these transformations will be used. For example, consider the case where we have:

- A transformation from a 3D reference brain to a 3D individual anatomical. **xf1.grd**
- A transformation from a 3D individual anatomical to a scout image. **xf2.matr**
- A transformation from the scout image to a functional acquisition. **xf3.matr**

The following command will reslice the functional image to the space of the 3D reference brain:

```
pxmat_resliceimage.tcl ref_image func_image resliced 3 xf1.grd xf2.matr xf3.matr
```

**Inverting Transformations** Often the inverse transformation is required. This can be accomplished using the pxmat\_inverttransform.tcl script. The syntax is:

```
pxmat_inverttransform.tcl reference_image output_transformation  xform1  
[ xform2 ] [ xform3 ]
```

If multiple transformations are specified (upto 3) then the inverse will be the inverse of the concatenation of these transformations.

**Computing Displacement Fields** Sometimes it is desired to get the displacement field for each voxel. This can be accomplished using the `pxmat_displacementfield.tcl` command. The syntax is:

```
pxmat_displacementfield.tcl reference_image output_field  
xform1 [ xform2 ] [ xform3 ]
```

The `output_field` will be a 4D analyze image file with three frames, storing the x, y and z displacement of each voxel in mm (or nominal units). If more than one transformation is specified the final displacement field will be a result of the concatenation of the transformations specified.

## 11.6 Example: Co-register reference 3D brain with individual 3D brain

The step-by-step instructions, with reference to Figure 11.5, are:

1. Choose `brainregister` from the BioImageSuite main menu. Three windows will appear: a Transform Viewer, a Reference Viewer and a BrainRegister menu bar. In the Reference Viewer choose (File — Standard Images — Colin\_1mm\_stripped), or the filename of your chosen reference brain.
2. In the Transform Window choose (File — Load)
3. Choose the filename that refers to the stripped whole brain acquisition from the brain extraction steps explained above and click Open.
4. On the BrainRegister menu bar, choose (Registration — Non-Linear Registration). A new Registration/OverlayTool window will appear.
5. In the new Registration/OverlayTool window click “Compute Linear + Non Linear Registration”. This will compute a Non-Linear Registration between the two loaded images. If the Auto Save Results box is red, BioImageSuite will automatically save the resulting .grd transformation file in the current working directory.

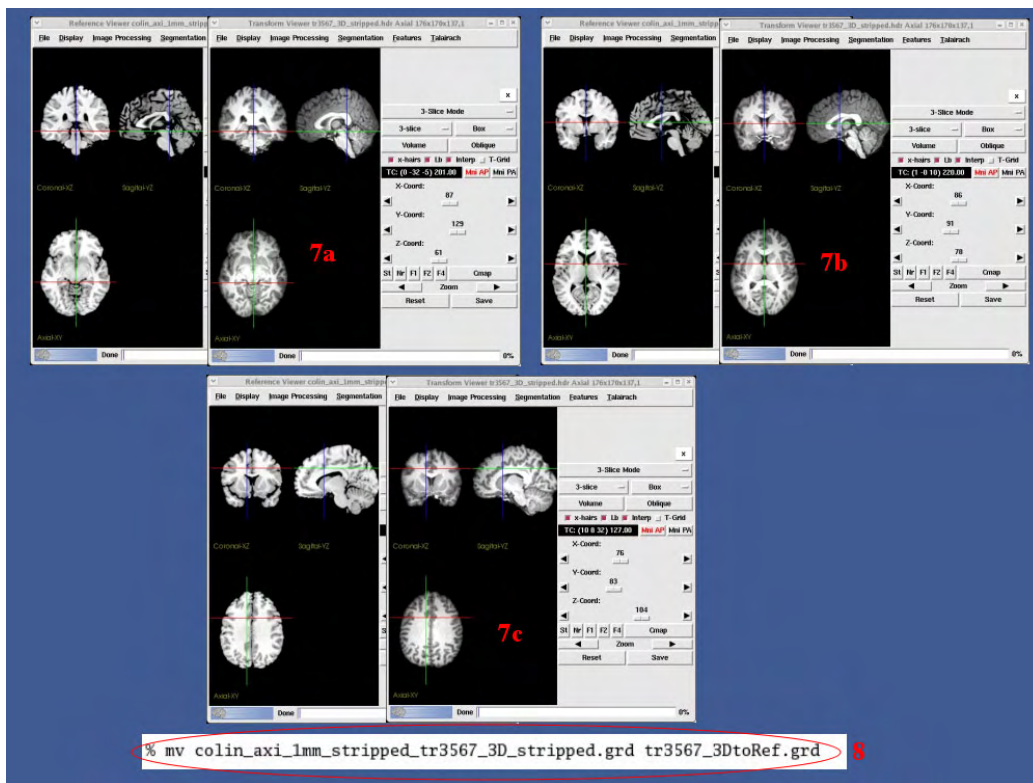
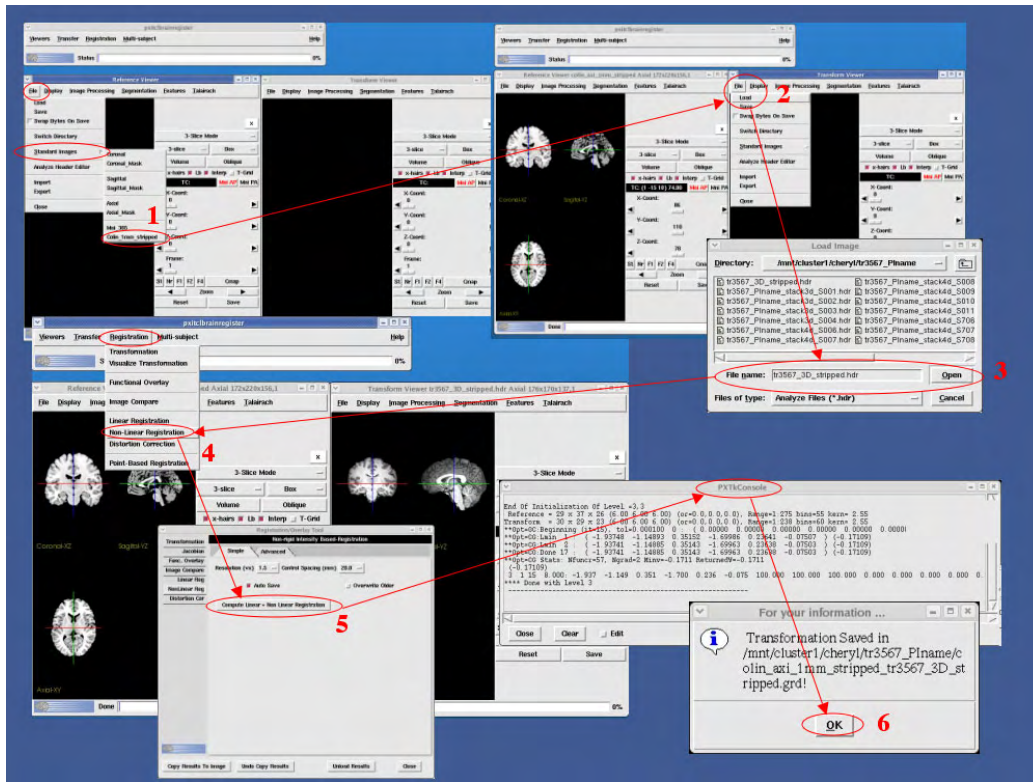


Figure 11.5: Methods used to Co-register an individual 3D wholebrain anatomical with a reference 3D brain image.



Figure 11.6: Methods used to load a previously saved non-linear transformation.

6. While the registration is running a PXTkConsole window will appear showing the status of the registration. When the registration is complete a “For Your Information” window will appear telling you the name of the saved registration file. If the filename already exists the window will ask if you want to overwrite the existing file.
7. Check the registration by clicking around in the individual 3D brain image (Transform Viewer) and seeing if the crosshairs match a similar anatomical position in the 3D reference image (Reference Viewer).
  - (a) Brainstem
  - (b) Anterior Commissure
  - (c) Cingulate Gyrus
8. In a unix window, rename the newly generated .grd file to [studynum]\_3DtoRef.grd
  - ex: `% mv colin_axi_1mm_stripped_tr3567_3d_stripped.grd tr3567_3DtoRef.grd`

## 11.7 Checking 3D to Reference non-linear registrations

The step-by-step instructions, with reference to Figure 11.6, are:

1. Choose brainregister from the BioImageSuite main menu. Three windows will appear: a Transform Viewer, a Reference Viewer and a BrainRegister menu bar. In the Reference Viewer choose (File — Standard Images — Colin\_1mm\_stripped), or the filename of your chosen reference brain.

2. In the Transform Window choose (File — Load)
3. Choose the filename that refers to the stripped whole brain acquisition from the brain extraction steps explained above and click Open.
4. On the BrainRegister menu bar, choose (Registration — Transformation). A new Registration/OverlayTool window will appear.
5. Under the Transformation block Click Load.
6. Choose the non-linear transformation for the 3D to reference registration (i.e. [studynumber]\_3DtoRef.matr) and Click Open.
7. Under Reslice Options Click Reslice.
8. Check the registration by clicking around in the individual 3D brain image (Transform Viewer) and seeing if the crosshairs match a similar anatomical position in the 3D reference image (Reference Viewer).

## 11.8 Remarks

In the next chapter, we discuss the use of surface based registration methods. These often a useful alternative if one is only interested in making specific structures with high accuracy (higher than what is possible with intensity based methods).

In addition to being useful for computing fMRI composite activations, nonlinear registration methods can be used for morphometric studies by examining the properties of the transformation map itself.



## Chapter 12

# Landmarks, Surfaces and Point-based Registration

### 12.1 Introduction

BioImage Suite can handle several types of objects. While images are by far the most important object type in BioImage Suite, point sets and surfaces also play significant roles in many applications. BioImage Suite has two specialized controls for manipulating these namely (i) The Landmark Control which is used to store and manipulate sets of points and (ii) The Surface Control which is used to manipulate surfaces. Surfaces and point-sets can be used as reference/target datasets for the Robust Point Matching registration algorithm – the Surface Objectmap Control can be used to define complex combinations of labeled surfaces. This session concludes with a look at the Point-based registration tools.

### 12.2 Acquiring Landmarks

Many operations in BioImage Suite require the use of landmark points. These include generating “Talairach Transformations” in neuroimaging as well as seeding of levelset-type segmentation algorithms etc. All landmark editing/storage etc. in BioImage Suite is handled by the Landmark Control. This is a multifunctional control that has capabilities for both landmark editing as well as curve tracing/extraction.

#### 12.2.1 Invoking and Interaction with the Viewers

When supported, the landmark control lives under the “Features” menu as shown in Figure 12.1. There may be up to three option boxes in the top of this menu marked:

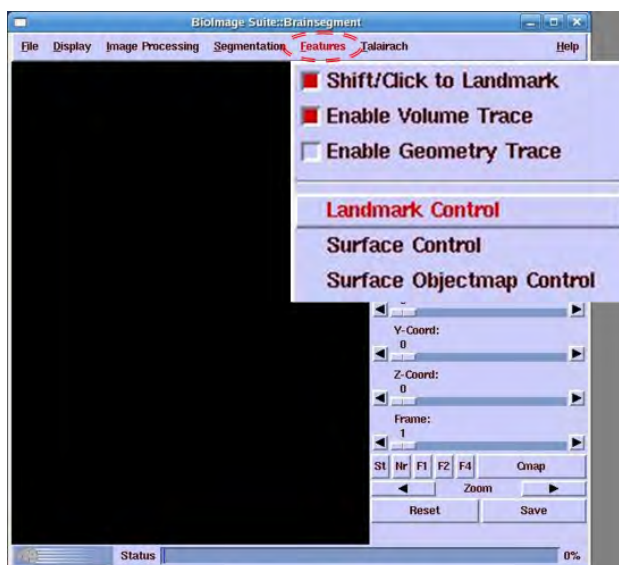


Figure 12.1: A viewer with the Features Menu.

- Shift/Click to Landmark
- Enable Volume Trace
- Enable Geometry Trace

If “Shift/Click to Landmark” is enabled (default), then to manipulate a landmark the user must hold the “Shift” key down and click with the left mouse button. This enables the use of the left mouse button alone for navigating in the images as usual. If desired, this option may be turned off, in which case navigation via the left mouse button is disabled and all left mouse button events are captured by the landmark control. (An exception to this is in some applications when the Electrode Editor is present; this is really a very specialized version of the landmark control and replaces its functionality). BioImage Suite also enables tracing on volume rendering or polygonal surfaces. These options are enabled using the “Enable Volume Trace” or “Enable Geometry Trace” menu selections. Enabling one of these disables the other. Volume tracing works by shooting a ray from the location of the virtual image camera and places a landmark at the first “non-zero” voxel in the image. It is most suitable for outlining cortical sulci from skulled-stripped brain images.

## 12.2.2 The Landmark Control

The Landmark Control is a complex tool for acquiring, storing, and manipulating “point-sets” which are simply collections of ordered points – the control can keep 9 different point sets in memory. Figure 12.2 shows a snapshot of the control, which is divided visually into four parts, namely: (i) the menu bar, (ii) the “point set properties” frame, (iii) the list of current points in the currently selected pointset and (iv) the “global properties” frame. All operations (either point acquisition or from the Landmark Control menu) are applied to the current pointset as selected using the option menu marked as (D) in Figure 12.2 (see also expanded version in Figure 12.3).

## Handling Mouse Input

Mouse operations (typically shift/click with the left button) are interpreted depending on the current mode, which is selected using the left option menu in the “General Properties” frame (this is marked as (C) in Figure 12.2 and shown expanded in Figure 12.3). There are 5 different options:

- **disabled** - mouse clicks are ignored.
- **pick mode** - selects a current landmark for editing.
- **auto mode** - this is used in conjunction with setup files (more later).
- **add mode** - a new landmark is added each time the mouse button is released.
- **continuous add mode** - this can be used to add points continuously as the mouse is moved. (Use with care!)

## PointSet Properties

In the pointset properties frame (see Figure 12.2) there are three items which control how a point-set is displayed.

**The mode option menu** ((A) in figure 12.2 and shown also in figure 12.3), lets you choose the display format for the set of points:

- **Landmarks** (i.e. individual points)
- **Open or Closed Curve** - A curve is drawn which connects the points in order and, if closed, - connects the last and first points.
- **Bounding box** simply draws the bounding box enclosing all points in 3D (this is useful for generating Talairach transformations).

**The point size menu** ((B) in figure 12.2) controls the size of the points in mm, ((B) in figure 12.2). **On the far right, a checkbox determines whether the points are displayed or not.** At the bottom of the pointset properties frame, a status textbox displays information about the current pointset, namely its name and the number of points in the set.

The *List of Current Points* simply lists the coordinates of all points in the current point set in mm.

## The Landmark Control Menu

**The “File” Menu** The File Menu contains options for loading and saving the pointsets. Pointsets can be loaded/saved to file in the default .land format. In addition they can be exported to standard

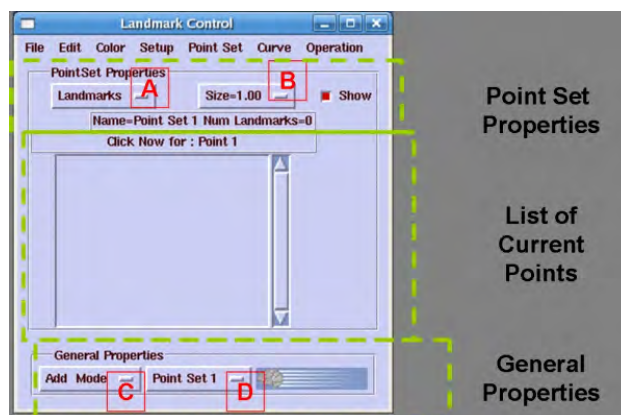


Figure 12.2: The Landmark Control.

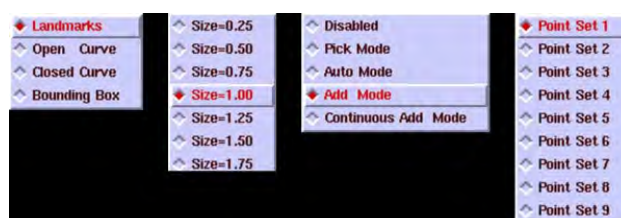


Figure 12.3: Expanded views of the option menus A-D of the Landmark Control, as defined in Figure 12.2 above.

vtk surface format (.vtk), as well as image objectmaps (where the voxel closed to each landmark is colored in).

**The “Edit” Menu** This contains self-explanatory options for copy/paste operations etc. The “Edit Point” option brings up a dialog box which enables direct manipulation of point coordinates by typing them into textboxes.

**The “Color” Menu** There are two options here: Landmark color, which enables setting the color of all the landmarks and Highlight color, which can be used to set the color of the “current” point – most commonly the last point.

**The “Setup” Menu** It is often desirable to guide the user to click a set of landmarks in a prescribed order. The Setup Tool has facilities to accomplish this. Pressing the Edit Setup File option under the setup menu brings up the Label Editor. This can be used to edit/create a set of labels. The setup prescription can be loaded/saved using the Load Setup File/Save Setup File options under the Setup menu. One preset mode is the Talairach Mode – an example of using this can be found in the Coordinates page.

**The “PointSet” Menu** This is the equivalent of the “Edit” menu for whole point sets. “Copy” and “Paste” copy or paste the current landmark to/from the clipboard. “Paste 2D” enables pasting only the x and y coordinates. Create Circle creates a circle in the XY plane.

**The “Curve” Menu** This contains operations for manipulating the pointset as a curve. Options include area and length computation, use of the curve to define a region of interest to perform ROI

analysis on the underlying image and options for smoothing and resampling the curve.

**The “Operations” Menu** There are two options here. (i) **Angle** returns the angle between the last two points and the x-axis. This is often useful in reslicing an image. (ii) **Extract** performs a local iso-contour extraction from the original image.

## 12.3 The Surface Control and the Surface Objectmap Control

**Note:** The Surface Control is a tool for manipulating and displaying surfaces parameterized as polygonal meshes. It can be used to create surfaces from images using the marching cubes algorithm – see the “Extract Iso-Contour” option below. However, for more detailed surface editing/creation see the Surface Editor webpage. Surfaces created using the Surface Editor can be exported to either .vtk or .tstack formats which can then be imported into the Surface Control. The Surface Control does not support direct loading of the native “.sur” format output of the Surface Editor as this captures the surface as a stack of parametric splines and NOT as a polyhedral surface.

The Surface Objectmap control is a specialized tool for creating surface objectmaps. These are used as inputs to the point-based registration tools. For more information see below. In the applications that they are available in, the surface and the surface objectmap controls appear under the Features Menu, as shown in Figure 1.

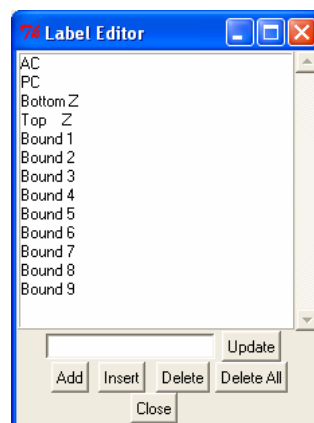


Figure 12.4: The Landmark Control Setup Tool. Landmark names can be added by typing them in the text box next to the update button and pressing the **Add** button.

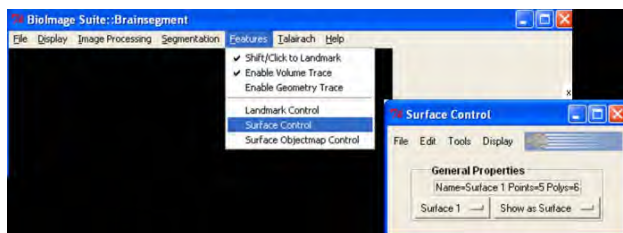


Figure 12.5: The Surface Control Menu.

### 12.3.1 The Surface Control

#### The General Properties Frame

The Surface Control essentially consists of a menubar and the general properties frame below it. On the general properties frame there are three items: (i) The status label which provides information about the current surface. (ii) The Surface Selector which allows the user to select which surface to manipulate – *all operations are performed on the current surface*. (iii) The Display menu selects how the current surface is displayed. There are four options. “Show as Surface” results in an optimal rendering for 3D viewers. “Show as Wireframe” results in the surface being displayed as a wireframe - this is often useful for showing the intersection of the surface with image slices. “Show as Points” eliminates the lines joining the points and just shows the individual surface points. Finally the “Do not show” option is used to turn off the display of the individual surface.

The display of each surface can be further manipulated using options in the display menu, which is described below.

### 12.3.2 The Menus

**The File & Edit Menus** There are four options under the File Menu:

- Load – for loading surfaces in the .sur or .tstack formats (the .tstack format is an internal format that originated in some of the cardiac applications).
- Save – for saving the surface in the .vtk format.
- “Binary Save” toggle. If selected save operations result in binary files which are more compact but less compatible across different platforms. Turning this off results in the surfaces being saved in text format which is both human readable and more cross-platform compatible at the expense of additional file size.
- Close – closes the control.

The Edit Menu also has four options:

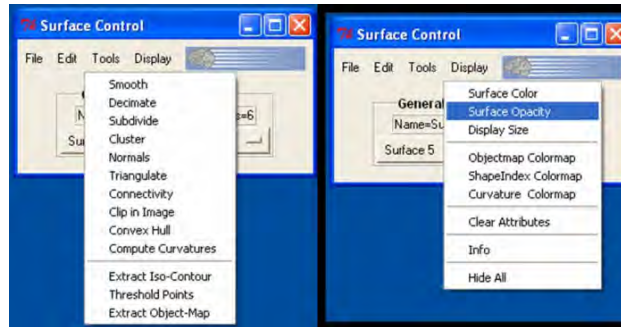


Figure 12.6: Surface Control Details. The Tools & Display Menus.

- The Copy and Paste options can be used to copy and/or paste in the current surface from the surface clipboard.
- The Undo option restores the previous state of the surface after manipulation with one of the tools described in the tools menu.
- The Append option is a poor man’s version of the Surface Objectmap control. It can be used to append different surfaces into a single surface.

**The Tools Menu:** The Tools Menu contains a number of options for surface manipulation. They are briefly described below. Most of these tools invoke directly VTK classes which are listed in parenthesis in the description.

- **Smooth:** surface smoothing. Two algorithms are available “Windowed Sinc” – (`vtkWindowedSincPolyDataFilter`) which is slower but does not shrink the – surface and “Laplacian” (`vtkSmoothPolyDataFilter`) which is faster but – may result in surface shrinkage.
- **Decimate** . This enables the reduction in the number of points/faces in the surface. (`vtkDecimatePro`).
- **Subdivide** . This is the opposite of decimation. It subdivides the surface to add additional points. (`vtkLinearSubdivisionFilter`).
- **Cluster** . This is a more radical version of the decimation algorithms which shrink the size of the surface by eliminating closely spaced points. Two algorithms can be invoked, the clean algorithm (`vtkCleanPolyData`) which performs simple distance sampling and the Quadric Clustering which performs local path fitting ( `vtkQuadricClustering`).
- **Normals** . This can be used to compute surface normals which can enormously improve the appearance of a surface (`vtkPolyDataNormals`).
- **Triangulate** . This tool ensures that all polygonal faces of a surface are triangles. (`vtkTriangleFilter`).
- **Connectivity** . The connectivity filter can return a connected surface (to eliminate small unconnected regions). (`vtkPolyDataConnectivityFilter`).



- **Clip In Image** . This function can clip a surface so that all its points lie within the current image.
- **Convex Hull** . This option generates the convex hull of a surface (vtkDelaunay3D).
- **Compute Curvatures** . This option can be used to compute the curvature of a surface. Five different options exist namely (i) The Shapeindex, (ii) the Mean Curvature, (iii) The Gaussian curvature, (iv) and (v) the first and second principal curvatures. These are computed by fitting local quadratic patches to the surface of size determined by the scale option. The range option can be used to saturate the values of the curvature to improve the display.
- **Extract Iso-Contour** - this applies an iso-contour extraction to extract a surface representing an iso-level in the image. (vtkContourFilter). The iso-level is set using “Iso-Contour Level”. The image may also be thresholded, smoothed and resampled to improve the extraction process. **Note:** Smoothing the image will result in smoother surfaces, it is often better to smooth the image rather than try to smooth the extracted surface afterwards. If the image is really an objectmap where each value corresponds to a pre-segmented structure, then use the Extract Object-Map option below instead.
- **Threshold Points** - this enables the surface to be thresholded based on the value of its - attribute or scalar vector. (vtkThresholdPoints)
- **Extract Object-Map** . This is a specialized version of Extract Iso-Contour for objectmap images, i.e. images which contain a small number of values, each of which corresponds to a pre-segmented structure. Such images can be generated using the Surface Editor tool.

**The Display Menu:** The display menu provides additional options for manipulating how the surface is displayed.

- **Surface Color** - allows the user to set the surface color.
- **Surface Opacity** - allows the user to set the surface opacity. (0=completely transparent, 1=completely opaque).
- **Display Size** - this is useful for setting the size of points and lines when the surface - is displayed as a wireframe or as points only.
- **Objectmap Colormap, ShapeIndex Colormap, Curvature Colormap** - these all allow for setting colormaps such that the surface color is determined by its attribute vector. Objectmap colormap is particularly suited for surfaces extracted using the “Extract Object-map” option, or generated using the Surface Objectmap control below.
- **Clear Attributes** - this clears the attribute vector of the surface.
- **Info** - this provides information about the current surface.
- **Hide All** - this hides all the surfaces!

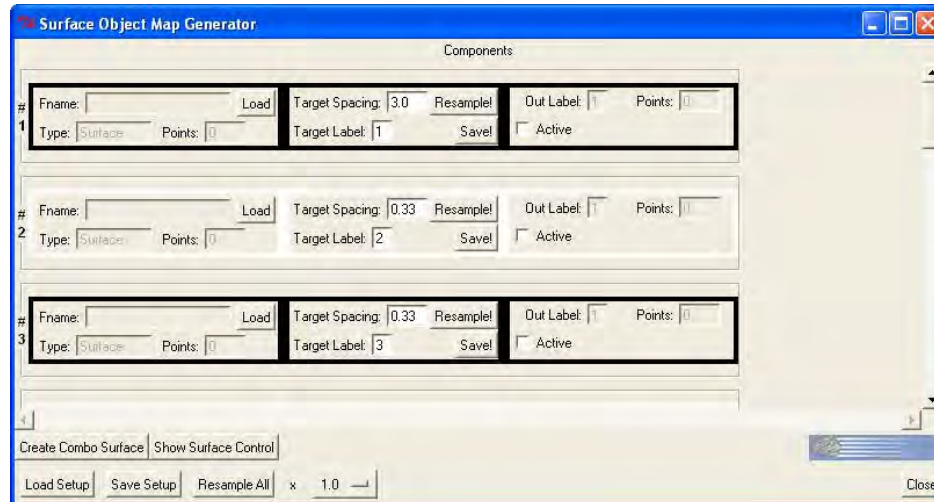


Figure 12.7: The Surface Objectmap Control.



### 12.3.3 The Surface Objectmap Control

The Surface Objectmap Control is designed to allow the combination of different surfaces and point sets into surface objectmaps which are used as inputs to the point-based registration tools. It can take as input up to 15 surfaces (in .vtk format) or point sets in (.land format, as saved by the Landmark Control.)

At the bottom of the surface objectmap control there are a number of buttons that can be used to Load/Save/Create the Objectmap. The **Load Setup** and **Save Setup** buttons enable loading/saving a list containing the filenames and attributes of the current set of surfaces.

The **Resample All** button causes all input surfaces to be resampled to have spacing equal to the product of the Target Spacing set in each surface (see below) multiplied by the scale factor set in the option menu next to the Resample All button.

The **Create Combo Surface** button appends all resampled surfaces into a single surface and places its output into the clipboard of the associated Surface Control, which can be invoked using the **Show Surface Control** button. The resulting surface objectmap can then be pasted into one of the surfaces in the Surface Control and saved for later use.

Each surface or pointset can then be resampled to have approximate point spacing set by the “Target Spacing” field and assigned a unique label for inclusion into the surface objectmap. Each point set has its own set of controls shown above. The control is divided into three panes (left to right). The leftmost pane provides information about the surface/pointset (filename, type=Surface/Landmarks, Number of Points.). The middle pane provides options for resampling the surface (spacing) and for setting its label for the surface objectmap to be outputted from the control. The rightmost

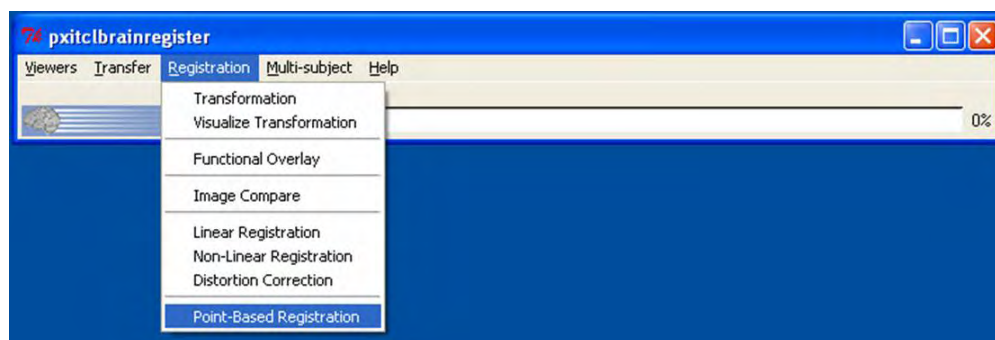


Figure 12.8: Invoking the Point-Based Registration Tool.

pane has one toggle button that enables the inclusion or exclusion of the surface from the final output (the “Active” toggle) as well as some information textboxes that display the actual number of points the surface is being resampled to.

## 12.4 Point-based Registration Tools

Point-based registration brings images into alignment based on feature points extracted from the images. The optimal transformation and correspondences must be determined.

### 12.4.1 Invoking and Interaction with the Viewers

The point-based registration control is available in `pxitclbrainregister` and `pxitclmouregister`. It is invoked using the point-based registration option under the Registration menu, as shown in Figure 12.8. The registration takes as inputs two sets of points saved as surfaces stored in the Surface Controls of the Reference and Transform viewers respectively. The output transformation is stored in the Transformation control of the Registration/Overlay tool.

In Robust Point Matching (RPM), the correspondence and transformation are determined together iteratively in a robust manner [19] accounting for outlier points which do not have a corresponding point in the other image. For more details see [30, 76].

### 12.4.2 The Main Point-Based Registration Control

The point based-registration control is divided into five parts:

1. The “Common” controls frame (**A** in Figure 12.9) which defines the two surfaces to be registered, and some global parameters.
2. The “parameters” controls frame (**B1/B2** in Figure 12.9) which defines the specific parameters for the linear registration (**B1**) and the non-linear registration (**B2**).

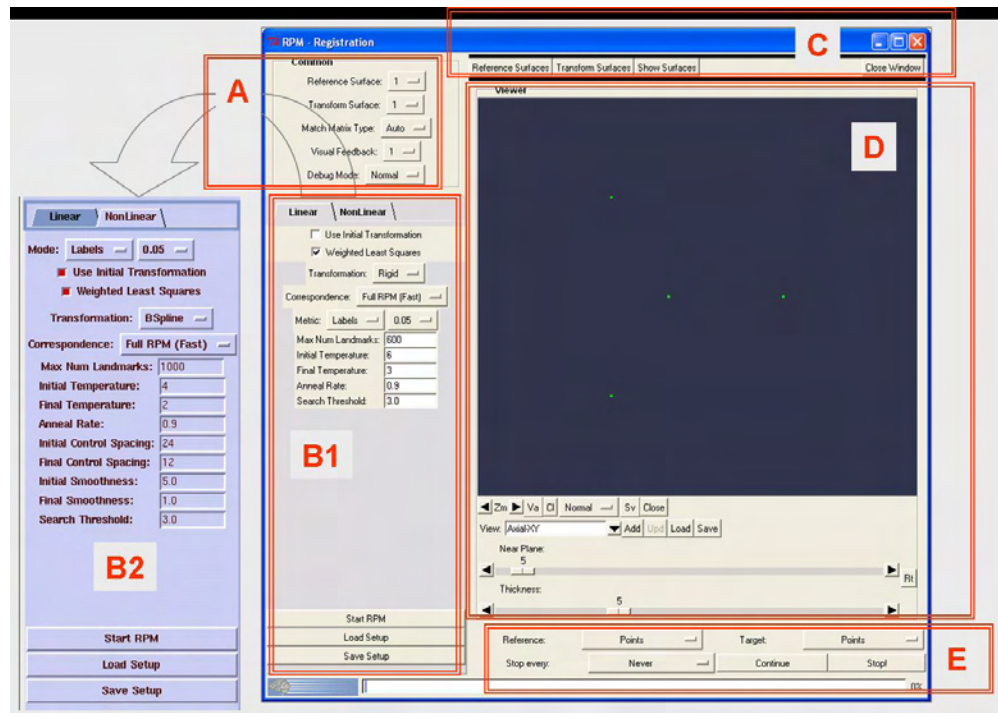


Figure 12.9: The Point-Based Registration Tool.

3. The “top-frame” controls frame (C in Figure 12.9) which has shortcuts to the two surface controls, and for closing the window.
4. The “Viewer” frame containing a separate viewer for monitoring the progress of the registration (C in Figure 12.9).
5. Finally, the update frame, at the bottom right (E in Figure 12.9) which contains settings for how often to update the display and how to display the surfaces.

### An example: Computing a Linear (rigid registration)

The procedure to compute a linear registration between two surfaces is as follows:

1. Load the reference surface in the surface control of the Reference Viewer. Note the index of the surface it is loaded in is 1 by default.
2. Similarly, load the target surface in the surface control of the Target Viewer, and note its index.
3. Open the point-based registration controls.
4. In the “Common Controls” (A), select the indices (most likely 1 and 1) for the reference and transform (target) surface.
5. Press “Show Surfaces” in (C) to show the surfaces and adjust the viewer to your liking. If no surfaces are visible, press the “Va” button in frame (D) to reset the display.

6. In Frame (**B1**), set the appropriate parameters. In particular, select the Linear tab and the “Rigid” transformation mode (other choices include Similarity and Affine). Set the desired number of points to use, in *Max Landmarks*. The only other parameters to touch are the “Initial Temperature” and “Final Temperature” which should be set to reflect (i) the maximum distance between the two surfaces prior to registration (initial temperature) and (ii) the point sampling distance of the surfaces, e.g. how closely sampled the points are (final temperature).
7. Press the **Start RPM** button at the bottom of frame **B1** to start the registration.

### A second example: Computing a Non-Linear (nonrigid registration)

This is similar to the Linear case, with the following changes:

1. A non-linear registration is often initialized by a linear registration. Run the linear registration first and verify that it is successful.
2. In step 6 above, there are a few more parameters to adjust. In particular:
  - First, select the Nonlinear tab. Enable **Use Initial Transformation** to use the last computed linear transformation (or the currently selected transformation in the transformation control of the Registration/Overlay tool.
  - The **Initial Temperature** needs to be set to account for the distance at the end of the linear step.
  - The **Initial Control Spacing** and **Final Control Spacing** reflect the values for the spacing of the tensor b-spline grid transformation that is to be computed. A rule of thumb is to select the final value to give the desired accuracy/computational cost and multiply this by 2 to set the initial value.
  - The **Initial Smoothness** and **Final Smoothness** determine the value of the regularization weight at the start and end of the process. Most registrations will start with a relatively high smoothness to avoid local minima and progressively relax this to get improved accuracy.
3. The computational cost is significantly higher!

## 12.5 Appendix: An Overview of Robust Point Matching

We present here a slightly modified form of the standard RPM methodology as can be found in Chui et al and Papademetris et al. This consists of two alternative steps: (i) the correspondence estimation step and (ii) the transformation estimation step. In the following discussion we will label the reference point set as  $X$  and the transform point set as  $Y$ . The goal of the registration is to estimate the transformation  $G : X \mapsto Y$ . We will label  $G^k$  the estimate of  $G$  at the end of iteration  $k$ .  $G^0$  is the starting transformation which can be the identity transformation.

### 12.5.1 Correspondence Estimation:

Given the point sets  $X$  and  $Y$  we estimate the match matrix  $M$ , where  $M_{ij}$  is the distance metric between points  $G^k(X_i)$  and  $Y_j$ . The standard distance metric is defined as:

$$M_{ij} = \frac{1}{\sqrt{2\pi T^2}} e^{-\frac{|G^k(X_i) - Y_j|^2}{2T^2}}, \quad \forall i \sum_j M_{ij} + C_i = 1, \quad \forall j \sum_i M_{ij} + R_j = 1 \quad (12.1)$$

where  $|X_i - Y_j|$  is the Euclidean distance between points  $X_i$  and  $Y_j$  and  $T$  is the temperature that controls the fuzziness of the correspondence. If the correspondence problem is to be thought of as a linear assignment problem, the rows and columns of  $M$  must sum to 1. The framework is further extended to handle outlier points by introducing an outlier column  $C$  and an outlier row  $R$ .  $C_i$  is a measure of the degree of ‘outlierness’ of a point in the reference point set  $X_i$  and  $R_j$  is the same for a point in the transform point set  $Y_j$ .  $C$  and  $R$  are initialized with constant values. The ability to model outliers allows this method to robustly match features of high variability such as cortical sulci, and to incorporate robustly manually outlined structures where the ends of the structures are user defined and somewhat arbitrary (e.g. in the case of the kidney users typically trace on certain slices and the kidney may partially extend into one more slice either side). Once the normalization is completed we can compute the correspondence as follows. Let  $V_i$  be the corresponding point to  $X_i$  and  $w_i$  the confidence in the match. Then  $V_i$  is defined as a normalized weighted sum of the points  $Y_j$  where the weights are the elements of the match matrix  $M$ .

$$V_i = \frac{\sum_j M_{ij} Y_j}{\sum_j M_{ij}}, \quad \text{and } w_i = (\sum_j M_{ij}) = 1 - C_i \quad (12.2)$$

Note that a point that has a high value in the outlier column  $C$  will have low confidence and vice-versa. We note that in our integrated method (Section 12.5.4) we simply use the correspondence piece of RPM.

### 12.5.2 Transformation Estimation

This is simply achieved by a regularized weighted least squares fit between  $X_i$  and  $V_i$  as follows:

$$G^k = \arg \min_g \sum_i w_i (g(X_i) - V_i)^2 + f(T) \mathcal{S}(g) \quad (12.3)$$

where  $\mathcal{S}(g)$  is a regularization functional (e.g. bending energy function) weighted by a function of the temperature  $f(T)$ . This last weighting term is used to decrease the regularization as we approach convergence.

### 12.5.3 Deterministic Annealing Framework

The alternating estimation of  $M$  and  $G$  is performed in a deterministic annealing framework. Starting with a high value of  $T$  corresponding to a rough estimate of the maximum mis-alignment distance we first estimate  $M$ , and then  $G$ . Then  $T$  is decreased by multiplying it with an annealing factor and the process is repeated until  $T$  becomes sufficiently small.

### 12.5.4 Integrated Points + Intensity-based Registration

**Intensity Module:** We use a slightly modified form of the intensity-based non rigid registration method first described by Rueckert et al. This method utilizes a free-form deformation transformation model based on tensor b-spline, and the normalized mutual information similarity metric. This metric can be expressed as:

$$NMI(A, B) = H(A, B)/(H(A) + H(B)) \quad (12.4)$$

where  $A$  and  $B$  are the two images, and  $H()$  is the image intensity entropy. This similarity metric is combined with a regularizing term to yield an optimization functional which is maximized in a multi-resolution framework. Our own implementation of this method first estimates a linear affine registration and then uses this as an input to estimate a full non-linear FFD transformation in a multiresolution manner.

**Integration Method:** We first estimate an initial registration using the RPM method alone, to ensure that landmarks are correctly aligned. We then proceed to refine the estimate of the transformation by minimizing the following optimization functional which is a trade-off between intensity similarity and adherence to point correspondence.

$$\hat{g} = \arg \max_g \left( \underbrace{NMI(g(A), B)}_{\text{Intensity Similarity}} - \underbrace{\frac{\lambda}{N} \left( \sum_i w_i |g(X_i) - V_i|^2 \right)}_{\text{Adherence to Point Correspondences}} \right) \quad (12.5)$$

where the first term is the intensity similarity distance (Equation 12.4), the second is a measure of adherence to the corresponding points as estimated by RPM (Equation 12.3) weighted by the constant  $\lambda$  ( $N$  is the number of points in the reference point-set). We note that during the optimization of this functional the correspondences maybe re-evaluated at each iteration at a constant temperature which is equal to the minimum temperature used in the initial registration. In practice, however, keeping the correspondences fixed produced marginally better results. The transformation estimated has the exact same parameterization as the FFD transformation estimated by the RPM algorithm.

This method is available only as part of the batch-mode command line tools.



## Part III

# C. Functional MRI Analysis

## Chapter 13

# The Single Subject fMRI Tool

### 13.1 Introduction

The Single Subject fMRI Tool can be used to analyze fMRI data using the General Linear Model [38]. It has a user friendly graphical interface for defining the study (e.g. image filenames, tasks, blocks etc) and the ability to either (i) perform the GLM computations internally or (ii) to create a shell script file that can be used to perform those computations in AFNI [3]. The internal computation and AFNI-based results agree to a few decimal points, however some users may prefer to use BioImage Suite to create the scripts and then use additional options in AFNI as they see fit.

### 13.2 The fMRI Tool User Interface

The interface has 5 tabs: “Study Definition”, “Anatomical Data”, “Session Information”, “Single Subject GLM” and “AGNI GLM”. It can be accessed either as a standalone application from the main BioImage Suite menu under the fMRI tab (in which case it has a viewer in a sixth tab) or as part of the BrainRegister application.

The “File” menu has choices is designed to load/save setup files, as well as to import setup files from an older fMRI Processing package in use at Yale ([98].

NOTE: the New/Load options clear all current parameters, and then loads the new project. Be sure to save all work before loading or clicking the “New” button.

The “Tools” menu is designed to make changes to the entire project. It has a single option – “Replace..” – which brings up a dialogue which allows the user to search and replace a specified string in all parameters in the current fMRI Tool project. The user types text into the search field and specifies what to replace with. The two strings do not have to be the same length. “Replace!” prompts the user with what it is going to do, and then proceeds to replace the string.

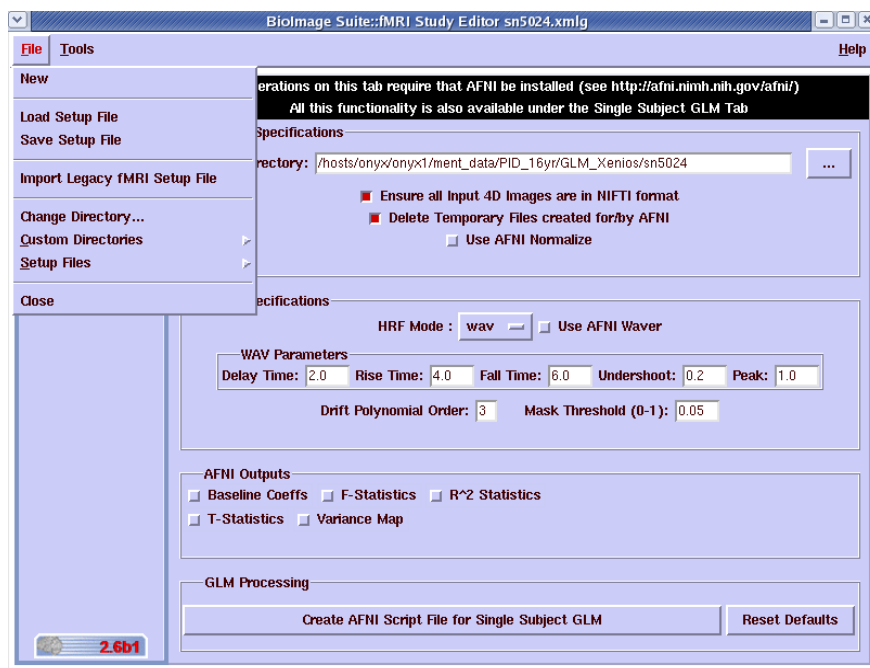


Figure 13.1: The FMRI Tool Menu Bar

The “Help” menu has one option which gives information about the XML Generator and BioImage Suite.

### 13.2.1 The Study Definition Tab

The Study Definition tab has inputs for the file header and reference brain.

**Study Title:** distinguishing title of the current study. This is a completely text input and there are no restrictions stopping the user from inputting irrelevant information. The information input into such text dialogues in throughout the fMRI Tool are simply written to the setup file without any processing.

**Subject ID:** non-identifying subject identification. All inputs are values here

**Reference Brain:** brain for common space and comparison to other patients. The user can specify a reference brain one of two ways: either by clicking the down arrow or by browsing using the ellipses button. In the former case, the name of the corresponding reference brain is for display purposes. The file path is automatically generated using the pre-set paths in BioImage Suite. If the user browses for the reference brain manually using the ellipses button, the filename is displayed in the dialogue. Once the reference brain is identified, the user can then choose to view the current image in the selection by pressing the “View” button. A high resolution reference brain that can be

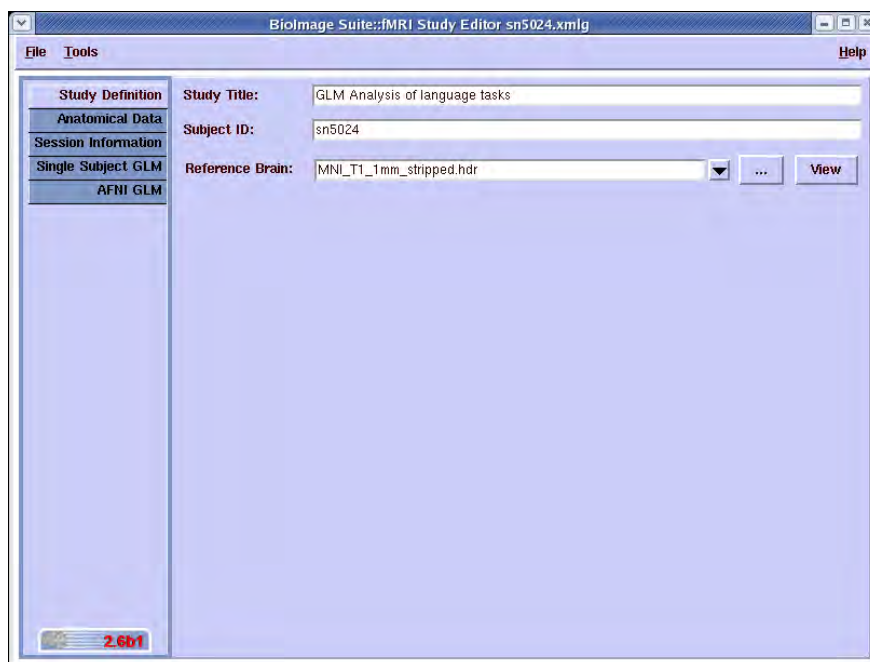


Figure 13.2: The fMRI Tool Study Definition Tab



Figure 13.3: Throughout the program there are a total of three ways to specify input: direct typing, selection from a list or browsing. Additionally, image input fields such as this reference brain input contain a view option.

used is the MNI-T1 1mm template (“Colin Brain” [44]). It has better resolution than the MNI-305 or ICBM-152 templates which allows for better non-linear registrations.

### 13.2.2 The Anatomical Data Tab

The anatomical data tab has input fields for five images and transforms necessary for fMRI processing. Each input is selected using the ellipses button to browse. The image load function is pre-filtered to look for “.hdr”/“.nii.gz” files, while the transformations are pre-filtered for “.matr” and “.grd” files. The images can also be viewed using the “view” button. If the image is identified and the user selects to view the image, it is then sent to the Viewer tab. Note that in this tab, the fields are disabled and the only way to select the appropriate file is by browsing for it. Also, although only the filenames are displayed, the entire filepath is saved when the setup file is saved.

Definitions:

- Conventional Image - the anatomical image acquired with the same slice-specification as the underlying fMRI study.
- Anatomical Image - high resolution 3D image of the whole brain.

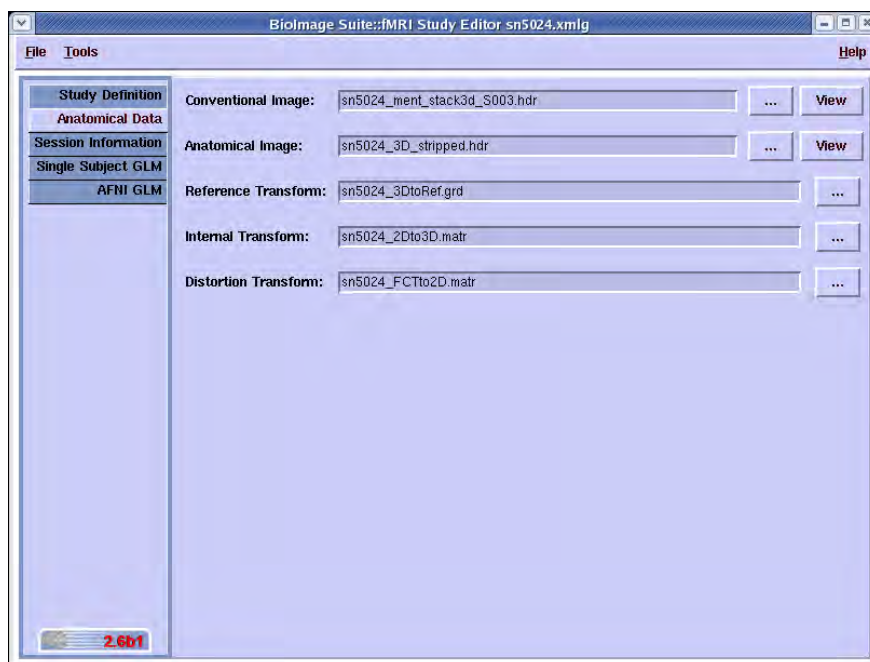


Figure 13.4: The FMRI Tool Anatomical Data Tab. This is where the anatomical images and the transformations needed to map them to each other and to the reference brain (common space) are specified.

- Reference Transform - maps the Reference/Template Brain to 3D Individual Anatomical Image. This is a non-linear registration that performs inter-brain warping.
- Internal Transform - maps the 3D Individual Anatomical Image to the “2D” Conventional Image. This is a rigid (linear) transformation which accounts for the differences in position and orientation between the two scans.
- Distortion Transform - maps the “2D” Conventional Image to the Functional/Echoplanar Image. This aims to capture the distortion in the echoplanar acquisitions used for fMRI

### 13.2.3 Session Information Tab

**Top Half: General Session Information** The session information tab contains fields for seven session specific parameters in the upper part of the window. These include the session ID, description, repetition time, number of slices, number of frames, a global flag as to whether all block definitions are in seconds or frames and a global skip frames which applies to all runs. These fields are text fields in which the user is free to type in any information desired. If the skip frames checkbox is not checked, the field is disabled and this information will not be used when performing the analysis.

**Bottom Half: Task and Run Information** The user needs to specify task names (descriptions) and runs. The two list windows are similar except that the run image files can be viewed. To add

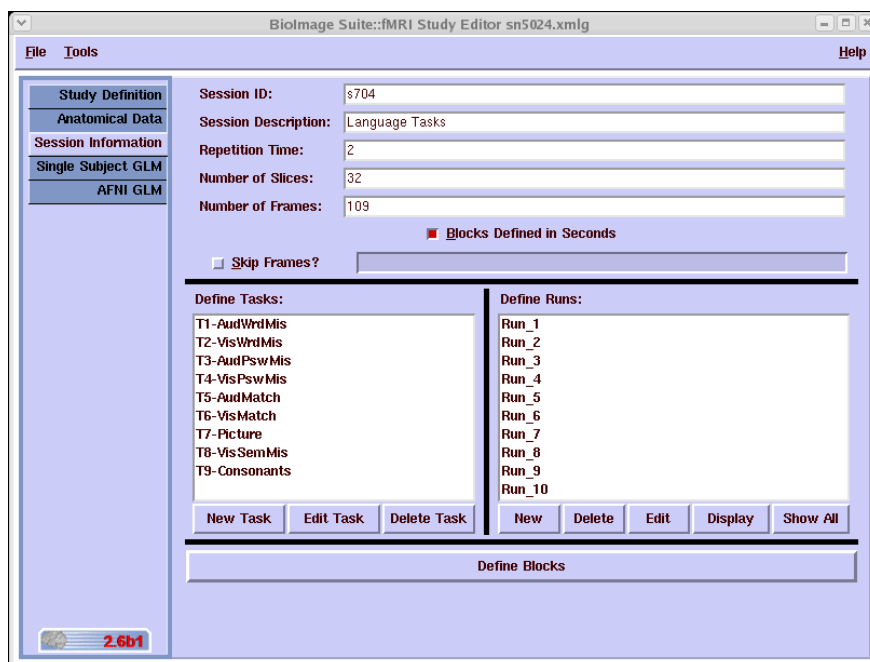


Figure 13.5: The FMRI Tool Session Information Tab.

a task, the user presses the “New Task” button to bring up a separate window. The task window contains an entry field that is open to all text. Once the user specifies the task name, and presses “Add Task”, the task is appended to the list with its identifying number, i.e. task 1 is labeled T1, and the user entered task identifier is listed. Double clicking on the task brings up the same task window and the task name can be changed. A task can be edited using the “Edit Task” button and can be deleted by clicking once on the desired task to highlight it and pressing the “Delete Task” button. If this happens the tasks are re-numbered to reflect the change.

The functions of the “Define Runs” list window are similar. The “New” button brings up a frame to define a new run. Here, browse for a filename to specify it in the entry field. If skip frames is desired within that run, check the box and list the frames to be skipped. Once this is all completed, press the “Add Run” button in the window. Double clicking on a run allows the user to change the run specifications, highlighting the run and pressing “Delete” deletes the run, highlighting and pressing “View” sends the image file to the viewer. The “Add Run” and “Edit Run” dialog boxes are shown in Figure 13.6.

Definitions:

- Session ID - Name of session for subject (not currently used)
- Session Description - Description of session for subject (not currently used)
- Repetition Time - Repetition Time of the Functional/Echoplanar trials in seconds
- Number of Slices - Number of Slices of the Functional/Echoplanar images
- Number of Frames - Number of Images per slice of the Functional/Echoplanar images



Figure 13.6: The Add (top)/Edit(bottom) Run dialog boxes.

- Skip frames - A block of images to skip within the Functional/Echoplanar trail. This block will be skipped for all trials unless a Skip Frames within run is defined.
- Task name - Short name to be used to define task that will be analyzed
- Run file - Functional/Echoplanar trial filename
- Skip Frames within run - To specify a different block of images to be skipped for each run separately.

**The Block Definition Control:** Once tasks and runs can be defined, the user can then press the “Define Blocks” button to identify the corresponding tframes for every combination of tasks and runs. This pops up the Block Definition dialog shown in Figure 13.7. The user needs to simply list the tframes and press “Done”. “Clear All” clears all tframes in the current window. The “Validate” button checks the block definitions for basic errors (e.g. frames outside the valid ranges, insufficient entry etc.)

The block definitions can be examined and verified graphically by pressing the “Plot” button shown in Figure 13.8. This by default shows the block design for the first run – the option menu in the bottom left corner can be used to select the other runs.

### 13.2.4 The Single Subject GLM Tab

The FMRI Tool Single Subject GLM Tab has options for performing the GLM analysis.

The following options can be set:

1. The HRF (Hemodynamic Response Function) Mode. This is one of:
  - “Wav” – the Cox “special” function which is the AFNI [3] default. The parameters for this can be set in the “Wav parameters” frame below.



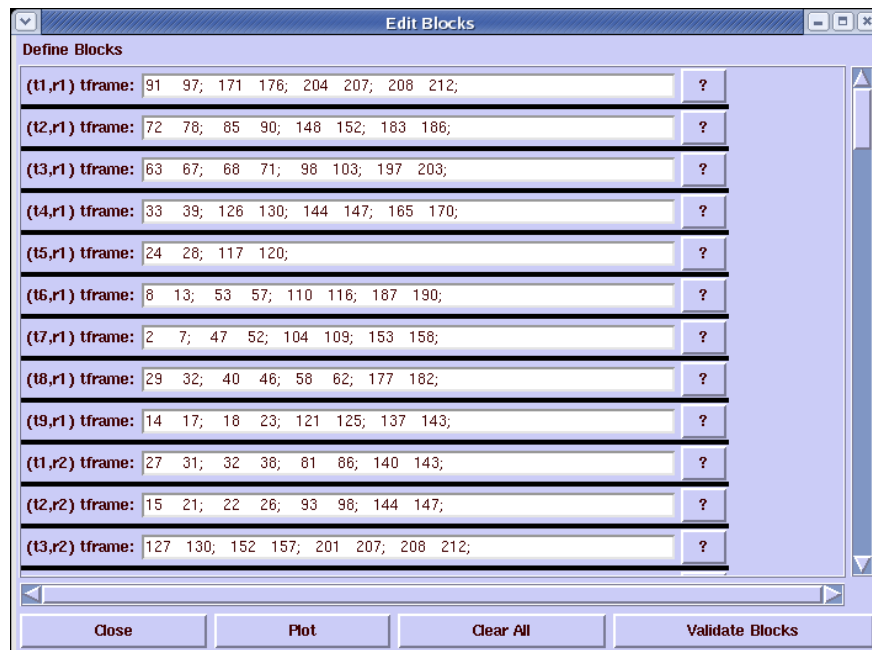


Figure 13.7: The Block Definition Dialog.

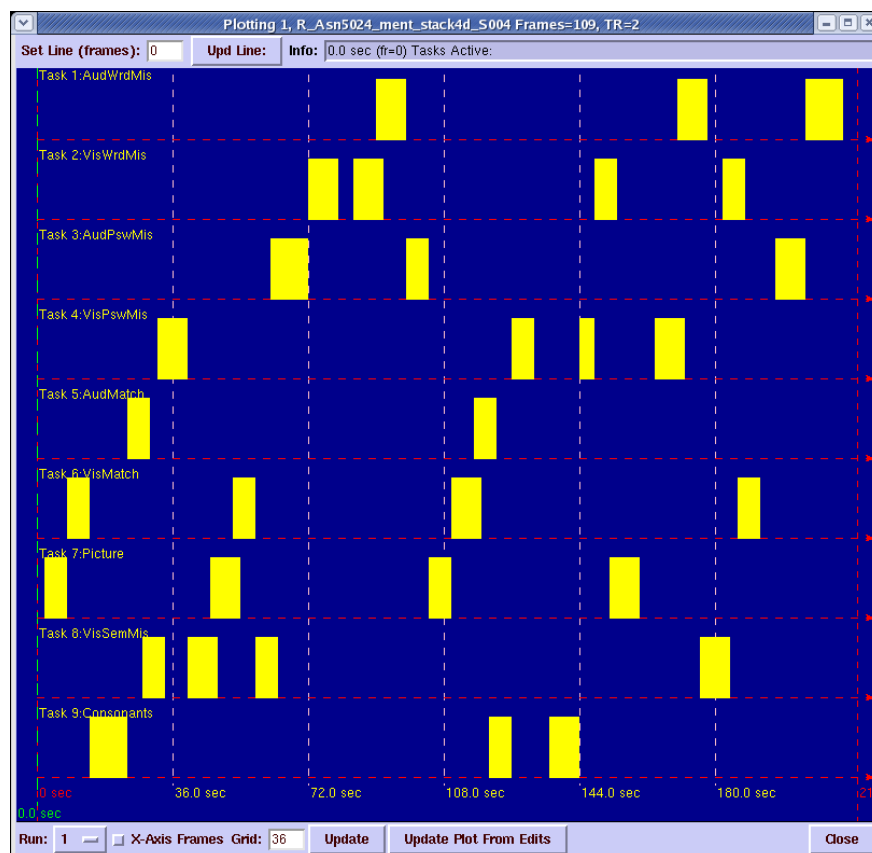


Figure 13.8: The Block Plotter Dialog.

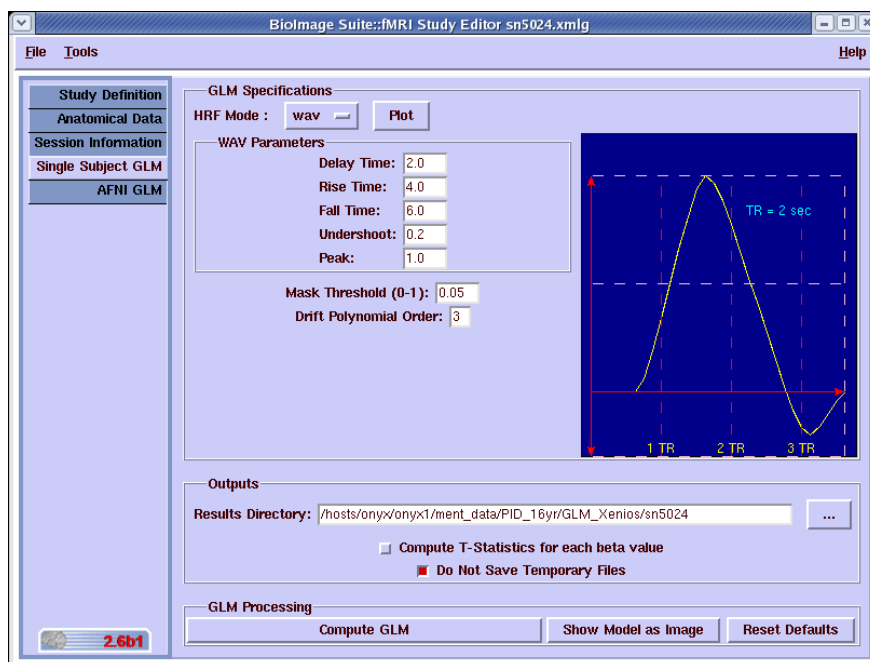


Figure 13.9: The FMRI Tool Single Subject GLM Tab.

- “Gamma” – a single gamma variate function.
- “Double Gamma” – a double gamma variate function.
- “Triple Gamma” – a triple gamma variate function.
- “SPM” – a double gamma variate function as specified in SPM [104].

Selecting an HRF from the option menu results in it being plotted in the plot window in the right of the dialog box.

2. Mask Threshold (0-1): This masks all data whose intensity is less than a certain percentage of the peak value in the fMRI time series. A value of 0.05 is the default which essentially masks all background voxels and speeds up the process.
3. Drift Polynomial Order (0-3): This is used to capture the image signal drift. The default is 3 (cubic Legendre polynomials)

Once the basic parameters are set, there are three more options in the Outputs frame:

- Results Directory : this defines the location of all output files.
- Compute T-statistics for each beta value: if on the tool will also save the t-test output for each regression coefficient.
- Do not Save Temporary Files: If disabled then some temporary files are not deleted – this enables debugging.

Finally there are three buttons on the bottom:

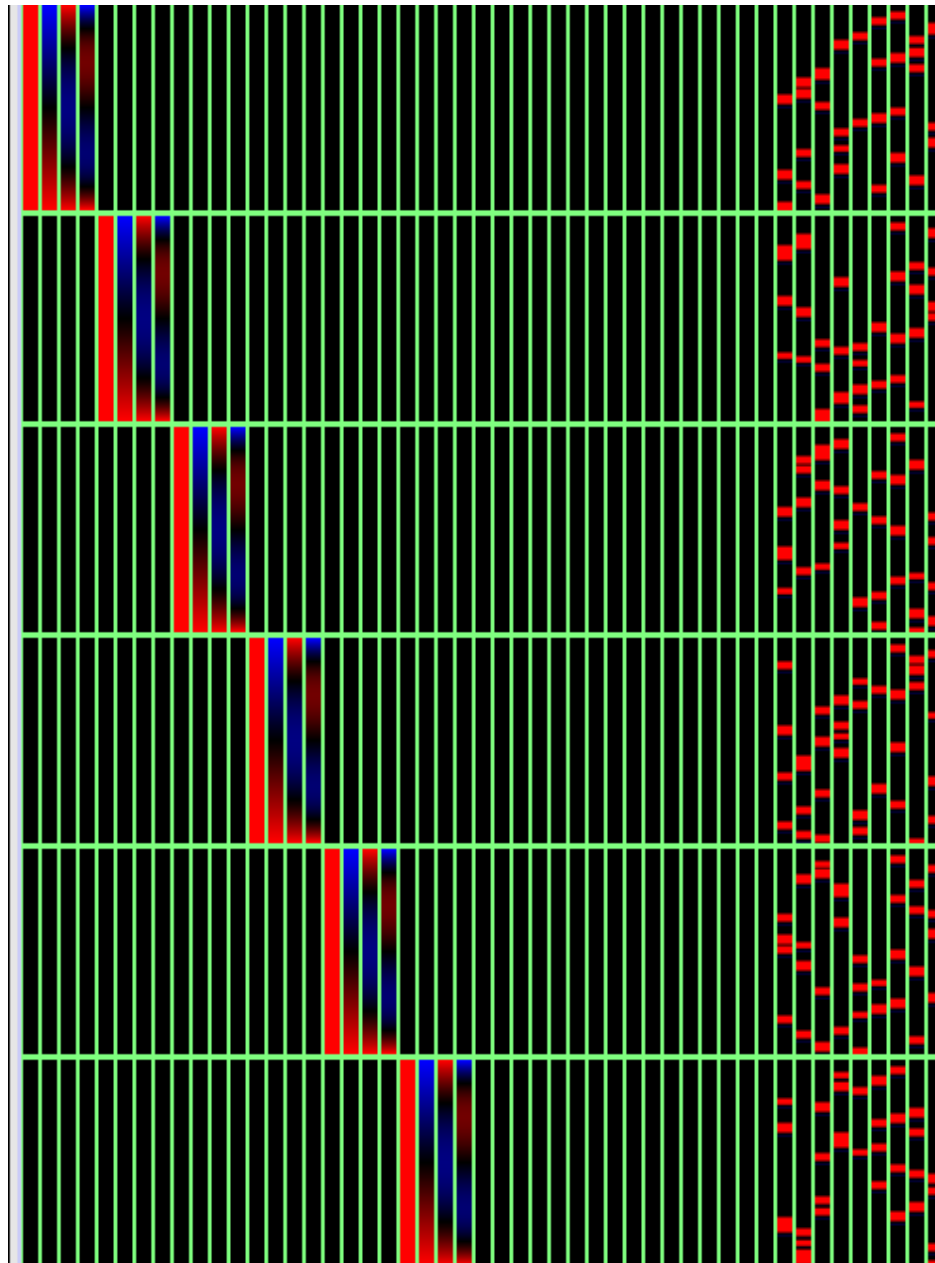


Figure 13.10: The output of the “Show Model as Image” operation. Here the GLM design matrix is plotted as an image (red=positive, blue=negative). For space reasons the image has been cropped to only show only the first 6 runs (of a total of 9). The vertical axis is time, whereas the horizontal axis indicates the regressor. The rightmost columns show the actual block design whereas the leftmost ( $9 \times 4 = 36$ ) columns are the drift terms. In this case we use a cubic drift correction (4 columns) which is separately done for each run.

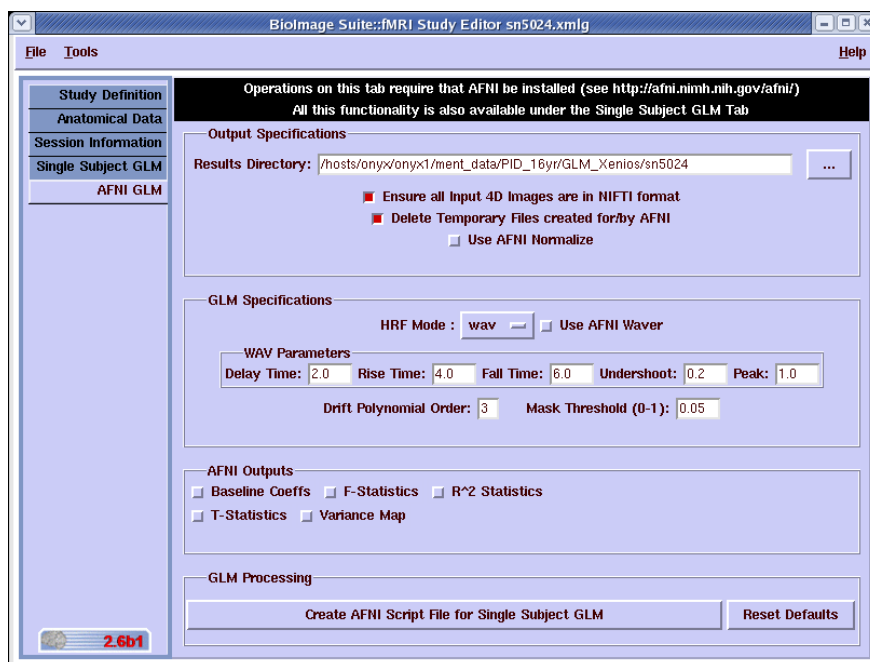


Figure 13.11: The FMRI Tool AFNI Tab.

- “Compute GLM” – this performs the computation which can take a couple of minutes or more depending on the speed of your computer.
- “Show Model as Image” – displays the GLM design matrix model as an image – see Figure 13.10.
- “Reset Defaults” – this resets all options on this tab to their default values.

If the fMRI Tool is used within the Brain Register tool (as opposed to a standalone application) then once GLM Computation is completed, the results are automatically loaded into the MultiSubject Tool for further display – see chapter 14 for more details on the MultiSubject Tool.

### 13.2.5 The AFNI GLM Tab

This tab contains duplicate functionality to that present in the Single Subject GLM Tab. The only difference is that instead of a “Compute GLM” button the corresponding option is “Create AFNI Script File”. This will create a text file containing a set of AFNI commands that can be used to compute the GLM. We use this extensively as a validation tool for our own GLM implementation.

Users familiar with AFNI can examine the script file and modify it as they please.

The two interesting options are:

- “Use AFNI Waver” – if off then BioImage Suite code will be used to generate the block design, if on then the AFNI program waver will be used to do this.

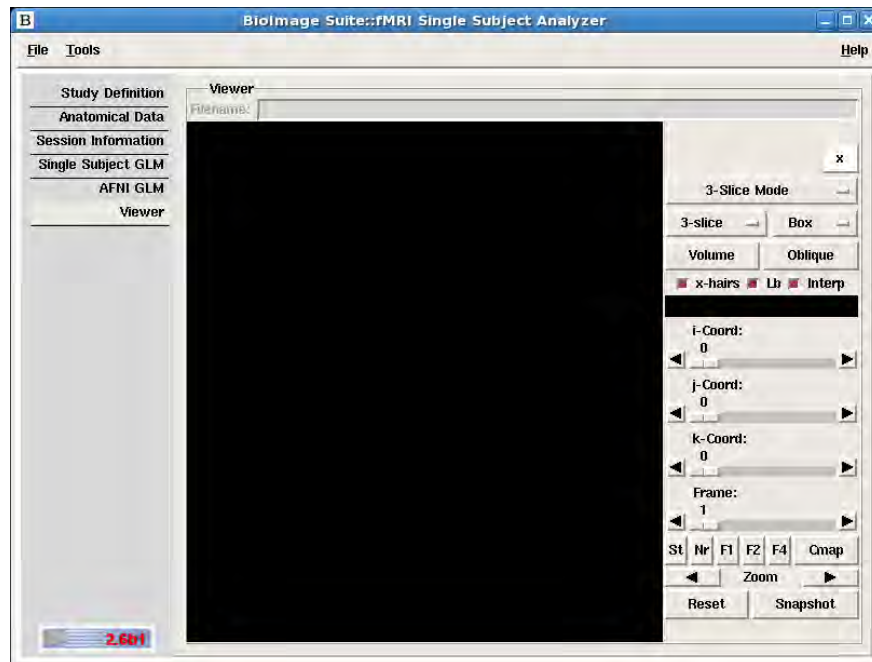


Figure 13.12: The FMRI Tool Viewer Tab.

- “Use AFNI Normalize” – as part of the script all runs are normalized to have mean 100 and concatenated. If the “Use AFNI Normalize” option is selected this will be done with a series of ANFI commands, otherwise the normalized and concatenated file is generated using internal BioImage Suite code.

### 13.2.6 Viewer Tab

If the fMRI Tool is run as a standalone application, then there is an extra tab – the “Viewer” tab. The viewer tab contains one of the BioImage Suite viewers. More information can on these can be found in the “The Viewers” Chapter of the manual – see Chapter 5. Throughout the fMRI Tool, certain fields have a “View” option, which automatically sends the image in that field to the viewer. If the image does not exist, the user will be notified.

## Part IV

# D. Multi Subject/Multi Image Analysis

## Chapter 14

# The Multi-Subject Control

### 14.1 Introduction

In the BrainRegister program, the main menu (see Figure 14.1) contains options that direct the many of the software's registration, transfer, and multisubject operations. The multisubject control is a specialized tool for computing multisubject composite fMRI activation maps. It can be activate through the BrainRegister and DualMultiSubject applications – the latter has two multisubject controls to enable easy group comparisons. The multisubject control interfaces extensively with the registration tools for computing registrations and performing image comparisons etc. The BrainRegister and DualMultiSubject applications are dual-viewer applications for, among other things, computing and testing registrations. The two viewers are labeled as the Reference viewer and the Transform viewer from the role they play during registrations. The reference image is taken from the Reference viewer whereas the transform image is taken from the Transform viewer. The multisubject control leverages both these viewers and the Registration tools (accessible under the Registration menu) for its operations.

The **Data Tree** tool is in some respects a more generalized and updated version of this control; however for computing fMRI composite activations using the MultiSubject tool is probably more convenient.

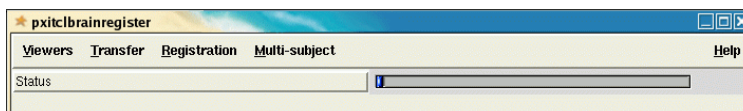


Figure 14.1: The BrainRegister program main menu bar.



## 14.2 Setup File Format

At the heart of the multisubject control is the setup file, which is stored with a “.msb” extension. While the setup file can be generated entirely using the graphical user interface, experienced users prefer using a text editor to edit this file directly.

A complete setup file for a group having 3 subjects and 2 tasks (or contrasts) is presented below. The text in typewriter-like font represents the setup file, whereas normal text represents comments.

**Critical Note:** There **MUST** be no spaces “ ” in any of the filenames in the setup file. This is generally a bad idea. Use underscores (“\_”) instead.

### 1. File Header

The first line of the setup file is its header which must contain exactly the text below. Any variations to this will lead to the setup file being rejected by the application.

```
#Multisubject Description File v3
```

### 2. Task Definition

fMRI experiments consists of a variety of tasks/contrasts. For example, a hand motor experiment, could have tasks “left v rest” and “right v rest” respectively . The multisubject control can handle a large number of such tasks provided that they are stored in analyze-format images with filenames of the form:

```
Common\_PrefixTask\_SuffixCommon\_Suffix
```

For example for the tasks above the three tasks could be called:

1. /data1/study/subject22\_leftvrest\_tmap.hdr
2. /data1/study/subject22\_rightvrest\_tmap.hdr

where the common prefix is “/data1/study/subject22\_“, the common suffix is “\_tmap.hdr“ and the task suffices are **leftvrest** , and **rightvrest** respectively. Such a set of tasks is prescribed in the setup file as follows:

## Tasks 2

---

 Task Name : Left vs Rest Hand Motor

 Task Suffix : leftvrest
 

---

Task Name : Right vs Rest Hand Motor

 Task Suffix : rightvrest
 

---

where after the word “Tasks” the number 2 signifies that there are two tasks, each of which is described with a descriptive “Task Name” and the all-important **Task Suffix**.

### 3. Individual Subject Data

Each subject is defined using upto nine pieces of information as listed below. All images must be in Analyze/NIFTI format (.hdr,.img pair or .nii.gz). The setup file stores the filenames of the images/transformations.

1. The **Subject Id** – this is the “name” of the subject.
2. The **3D Anatomical Image** – the filename of a 3D image of the whole brain.
3. The **“2D” Conventional (Scout) Image** – a filename of the conventional anatomical image – i.e. the anatomical image acquired with the same slice-specification as the underlying fMRI study.
4. The **Functional Image** which is the filename for the first task – in this case the “Left vs Rest Hand Motor” task. The filename must be in the same format as that described in the Task Definition section above.
5. The Reference Transformation which maps the Reference/Template Brain to **3D Individual Anatomical Image** . This is a non-linear registration that performs inter-brain warping.
6. The Internal Transformation which maps the **3D Individual Anatomical Image** to the **“2D” Conventional (Scout) Image**. This is a rigid (linear) transformation which accounts for the differences in position and orientation between the two scans.
7. Optional: The **Echoplanar Image** – is either a spin echo-anatomical image which is perfectly registered with the fMRI time series, or one of the fMRI time-series images (or perhaps the mean or median T2\* image).
8. Optional: The Distortion Transformation which maps the **“2D” Conventional Scout Image** to the **Functional/Echoplanar Image** . This aims to capture the distortion in the echoplanar acquisitions used for fMRI. (This is often left blank, especially if the software generating the statistical maps – the task files performs some alignment between the echoplanar and conventional images. Ideally, this should be a non-linear transformation which captures the distortion, although affine linear transformations are also often used here.

9. Optional: The **Field Map** image which is a direct measure of the distortion. This is currently not used directly, although it might be used in batch-distortion computation in the future.

**Note:** As shown in the example below, the appropriate line must still exist even if an optional field is blank. See for example all “Fieldmap Image:” lines below.

Subjects 3

```

-----
Subject Id : Subject 1
Anatomical Image : /data1/study/1256/1256.hdr
Conventional Image : /data1/study/1256/fmri_data/conv1256_05.hdr
Functional Image : /data1/study/1256/fmri_data/1256_leftvrest_m1.hdr
Reference Transformation : /data1/study/registrations/template_1256_rpm.grd
Internal Transformation : /data1/study/1256/1256_conv_1256_05.matr
Echoplanar Image :
Distortion Transformation :
Fieldmap Image :
-----

Subject Id : Subject 2
Anatomical Image : /data1/study/1268/1268.hdr
Conventional Image : /data1/study/1268/fmri_data/conv1256_05.hdr
Functional Image : /data1/study/1268/fmri_data/1256_leftvrest_m1.hdr
Reference Transformation : /data1/study/registrations/template_1268_rpm.grd
Internal Transformation : /data1/study/1268/1268_conv_1256_05.matr
Echoplanar Image :
Distortion Transformation :
Fieldmap Image :
-----

Subject Id : Subject 3
Anatomical Image : /data1/study/1285/1285.hdr
Conventional Image : /data1/study/1285/fmri_data/conv1285_05.hdr
Functional Image : /data1/study/1285/fmri_data/1285_leftvrest_m1.hdr
Reference Transformation : /data1/study/registrations/template_1285_rpm.grd
Internal Transformation : /data1/study/1285/1285_conv_1285_05.matr
Echoplanar Image :
Distortion Transformation :
Fieldmap Image :
```

#### 4. Reference and Output Images

The final section in the setup file defines the Reference/Template Brain and filenames for storing the composite anatomical/functional maps.

The Reference Image (often the MNI T1 Template) defines the space for the composite maps. An additional input (VOI image) can be used to define volumes-of-interest (perhaps created using the Surface Editor application) for VOI analysis. The VOI Image has each region labeled with a unique label (e.g. 1=Left Amygdala, 2=Right Amygdala etc.)

All outputs are in the space of the reference image. The outputs are:

- *Average Anatomical* – if desired the program can compute the average anatomical image using the transformations and the individual 3D Anatomical images.
- *Std Anatomical* – the standard deviation of the anatomical images.
- *Average Functional* – the average functional map image filename for the first task.
- *Std Functional* – the standard deviation of the first task.
- *Tscore Functional* – a t-test of the first task against zero.
- *Three lines marked Legacy which exist for compatibility with older versions of the multisubject control.*

The filenames for the three task-dependent outputs (Average Functional, Std Functional, Tscore Functional) **must be** in the same format as that described in the Task Definition section above.

```
-----
Reference/Output Images
-----
```

```
Reference Image : /data1/study/template/template.hdr
VOI Image      : /data1/study/template/template_vois.hdr
Average Anatomical : /data1/study/results5/study_average_anatomical.hdr
Std Anatomical  : /data1/study/results5/study_std_anatomical.hdr
Average Functional : /data1/study/results5/study_average_functional_leftvrest.hdr
Std Functional  : /data1/study/results5/study_std_functional_leftvrest.hdr
Tscore Functional : /data1/study/results5/study_tmap_functional_leftvrest.hdr
Legacy         :
Legacy         :
Legacy         :
```

### 14.3 The Multisubject Tool Graphical User Interface

This control has three tabs: “Base”, “Subjects” and “Results”. The user defines the reference brain and the tasks in the “Base”-tab. Individual subject images/transformations can be accessed in the “Subjects”-tab. All composite result computation/display is performed using controls in the “Results”-tab.

The menubar contains facilities for loading/saving the setup file, loading all images, performing batch computations (best avoided, use the command line batch mode tools instead), and for interfacing to the Image Compare tool in the Registration/Overlay Control.

**Important:** Loading a setup file does not load the images into memory – this is to allow manipulation of the setup file without the memory/disk-access overhead involved in loading all the images

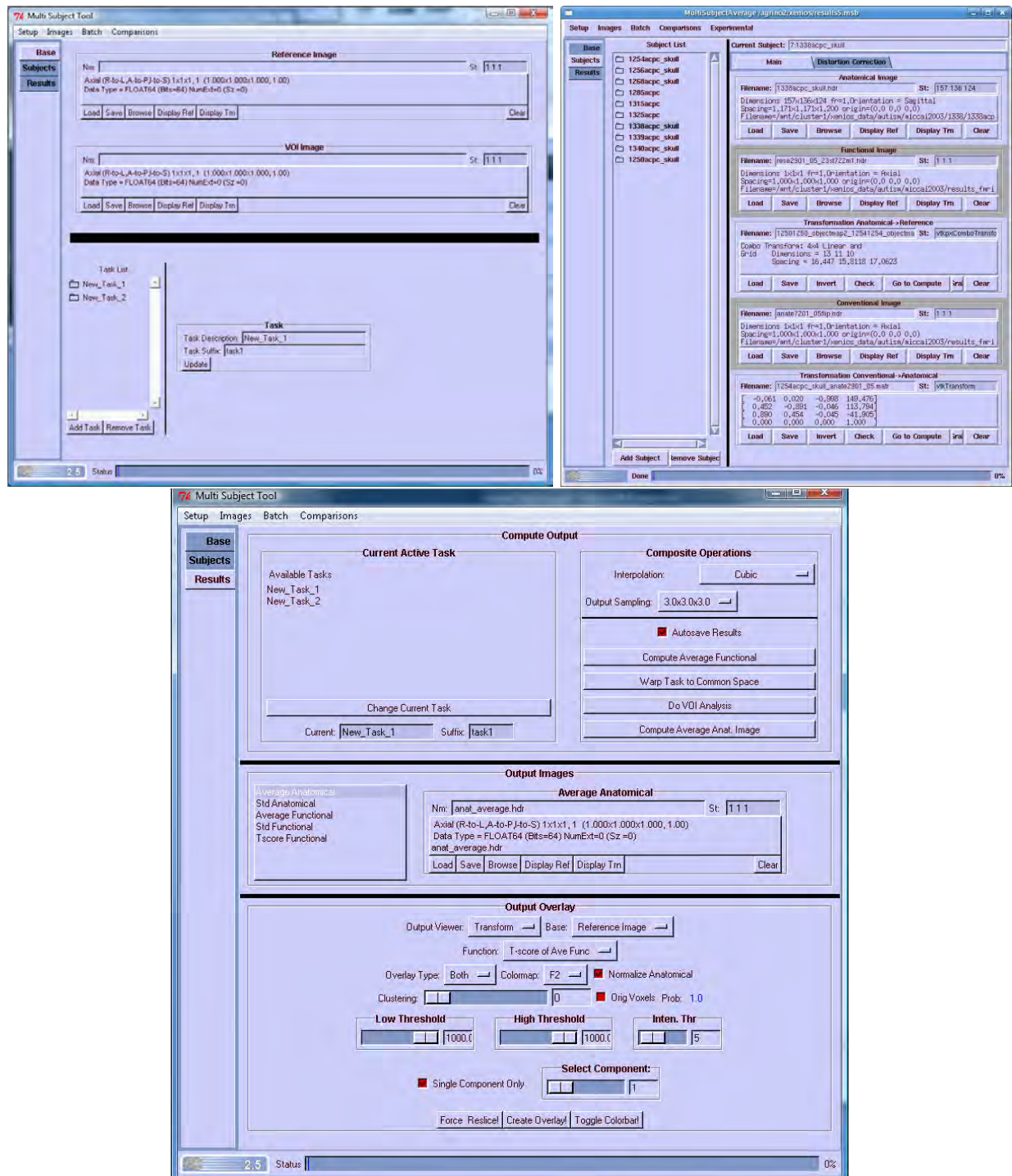


Figure 14.2: The three tabs in the multisubject tool, the base (top left), the subjects (top right) and the results tab (bottom). These are described in more detail below.

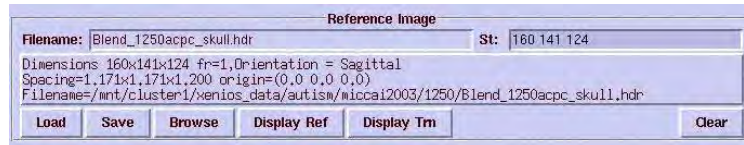


Figure 14.3: The Reference Image Control. A similar control (not shown) is used to manipulate the VOI Image.

for a large study. To load the images use the **Load All Images** menu option under the “Images” menu.

### 14.3.1 The Base Tab

This essentially consists of the Reference and VOI Image Controls and the Task definition Control, both of which are described next.

#### The Reference Image Control

This is a common control in BioImage Suite for handling images. It consists of two textboxes, which display information about the current image and a set of buttons for manipulating the image (See Figure 14.3.)

The three textboxes display (i) the filename, (ii) the dimensions – if these are “1 1 1” **this means that the image is not in memory, just its filename**, and (iii) a detailed description of the image.

The **Load** button is used to load the image into memory. The **Save** button can be used to save the image, perhaps in a different location. The **Browse** button is used to define the image filename *without actually loading* the image. The **Display Ref** and **Display Trn** buttons display the image in the reference and transform viewers respectively. Finally the **Clear** button unloads and clears the image.

The setup file must be saved for changes to the image filename to be made permanent.

*The VOI Image* is manipulated using an identical control.

#### The Task Definition Control

The task definition control enables the addition/removal and editing of tasks. This relates to the Task Definition section of the setup file described earlier. As shown in Figure 14.4, this control consists of a list of tasks (left) with two associated buttons for adding a new task and removing the selected task, and a task window right for editing the task description and task suffix for the current task. Edits are acted upon when the **Update** button is pressed.



Figure 14.4: The Task Definition Control.

You **may not** delete the task that is currently active – this is the first task when the setup file is loaded, as this makes the setup file invalid. The current task is selected in the Results tab, described below.

### 14.3.2 The Subject Tab

The subjects tab, shown in the middle of Figure 14.2, is where information for each subject can be manipulated. This relates directly to the Subject Definition section of the setup file described earlier. The graphical user interface for the Subject tab is divided into a left column which contains the list of subjects and two buttons for adding a new subject and removing the currently selected subject, and the right column which displays the information for the current subject. The currently selected subject is also used by the Overlay Controls in the Results Tab (below) to determine which subject’s data to overlay when the user requests “Individual Subject” images.

The right column of the Surface Tab contains the subject properties. The properties frame has a textbox at the top which has the name of the current subject (extracted automatically from the Anatomical Image), and then below this a nested tab control with two panes, “Main” and “Distortion Correction”. Between them, these two panes contain controls for manipulating the 8 elements of the individual subject definition described above.

#### 1. In the Main Tab

- Subject Id which contains the “name” of the Subject.
- Anatomical Image which stores the 3D Anatomical Image.
- Functional Image which stores the Functional Image for the current task.
- Transformation Anatomical → Reference which stores the Reference Transformation.
- Conventional Image which stores the “2D” Conventional (Scout) Image.
- Transformation Conventional → Anatomical which stores the Internal Transformation.

#### 2. In the Distortion Correction Tab

- Spin-echo Echoplanar image which stores the Echoplanar image.



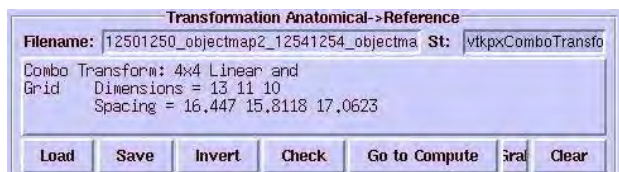


Figure 14.5: A Transformation Control.

- Transformation Echoplanar  $\rightarrow$  Conventional which stores the Distortion Transformation.
- Fieldmap Image which stores the Field Map.

Of these controls, 5 are image controls. These are similar to the reference image control shown in Figure 3. They are labeled as: The other 3 controls are transformation controls and are used to store the Reference, Internal and Distortion Transformations respectively.

One of these is shown in Figure 14.5. It consists of (much like the image control shown previously) of three textboxes and a set of buttons. The three textboxes display (i) the filename, (ii) the class of the transformation (`vtkTransform=linear`, `vtkpxComboTransform=nonlinear`), and (iii) details about the transformation. If the transformation is a linear one, then the 4x4 matrix is shown in the details box. The **Load** button is used to load the transformation and similarly the **Save** button can be used to save the transformation, perhaps in a different location. The **Invert** button can be used to invert the transformation – use with care in the case of nonlinear transformations, and the **Clear** button sets the transformation to identity.

The **Check** button can be used to confirm the quality of the transformation. It places the original reference image (e.g. in the case of the Anatomical  $\mapsto$  Reference transformation, this is the 3D Reference Image Template) in the left, or Reference Viewer, and places a resliced (warped) version of the Transform image (in this case the Anatomical image) in the Transform Viewer. The linked cursors of the two viewers can then be used to navigate to important structures and visually inspect the quality of the mapping.

The *Go to Compute* button sets the program for computing the transformation. It places the appropriate reference image in the Reference viewer, the appropriate Target in the Transform viewer and opens the Registration control at the appropriate tab (either linear or nonlinear). Next the user needs to click on a button in the Registration control (e.g. in the linear case either “Rigid” or “Affine”, in the non-linear case “Compute Linear + Non Linear Registration”) to start the registration. Once this is completed it can be migrated back to the multisubject tool using the *Grab* button. Until the *Grab* button is pushed the computed transformation is not stored anywhere and could be lost if the program is closed.

All modifications to the Individual subject information are saved when the Setup File is saved. Use Setup/Save Setup File button in the multisubject control to do this. It is suggested that you save the setup file periodically.

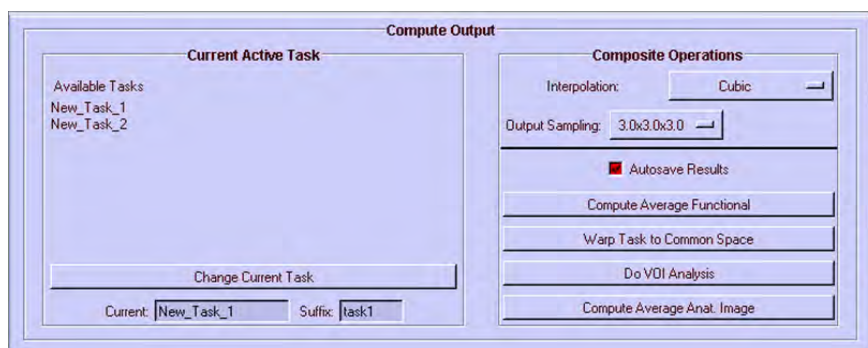


Figure 14.6: The Compute Output Frame.

### 14.3.3 The Results Tab

A user, once she has finished defining the input data is defined and the registrations check, most of the time (while using the Multisubject Control!) in the Results-tab. This is divided vertically into three large units, the Compute Output frame (shown in Figure 14.6, the Output Results frame (shown in Figure 14.7) and the Output Overlay pane (see Figure 14.8). Briefly, the Compute Output frame is used to select the current task and to compute average maps. The Output Results frame can be used to directly visualize these results as well as Load/Save them to disk. The Output Overlay frame has functionality for generating anatomical/functional overlays for visualization.

#### The Compute Output Frame

The Compute Output frame is shown in Figure 14.6. On the left hand side there is a frame titled “Current Active Task” which lets the user select the current task. The multisubject control only keeps one task in memory. To change the current task select it in the list and press the Change Current Task button. Pressing this will load the specific task files for all the subjects as well as the average, std and t-score functional maps for this task if they exist.

Composite functional maps, for the current task, are computed using the controls on the right hand side. The user may set the resolution at which these are computed (lower resolution=faster computation) as well as the interpolation mode for image reslicing. The Compute Average Functional button generates the composite maps (i.e. average functional, std functional and tscore functional). These are automatically saved if the Autosave results checkbox is enabled. Alternatively if a user desires to warp all functional data to a common frame and process them using external software, he can use the Warp Tasks to Common Space button. Both these buttons will perform their respective operations for the current task. If multiple task computations or exporting is desired, it can be accomplished using similarly named options from the Batch menu, in the multisubject control menubar. Finally, the Compute Average Image button computes the average anatomical image – this is useful as a quick visual check on the transformations.

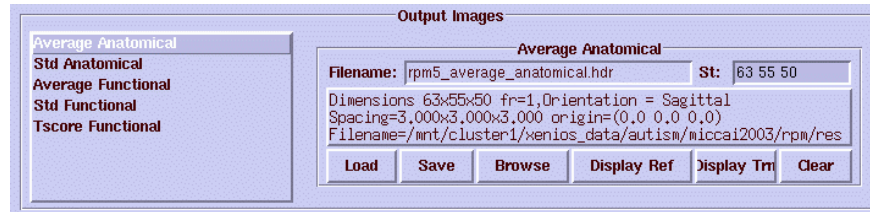


Figure 14.7: The Output Results Frame.

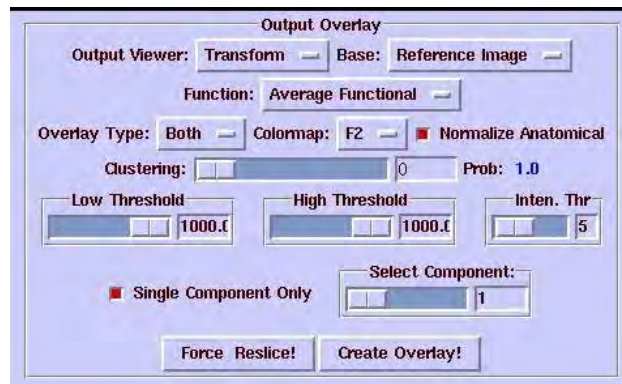


Figure 14.8: The Overlay Controls Frame.

### The Output Results Frame

The Output Results frame, shown in Figure 14.7, consists of a single control, which is a multi-image control. This is a multiple image version of the standard image control used in the other tabs, e.g. see the description of the reference image control (see also shown in Figure 14.3). The only additional element is that a set of images share this control. The current image is selected from the list on the left hand side – see Figure 14.7.

### The Overlay Controls Frame

The overlay controls frame is a common control, also found in the Registration and DataTree tools, for creating overlays of functional data on anatomical data. The basic principle used for the overlay is that the users sets a threshold for what constitutes significant function using the Low Threshold slider and then saturates the functional data at the level set by the High Threshold slider. For more details see the description in the Registration handout.

This overlay control has three additional options over the standard overlay control.

1. The Output Viewer can be used to direct the overlay to either the Reference or the Transform Viewer.
2. The Base drop menu can be used to select which image will be used as the anatomical image on which the function is overlaid.

3. The Function drop menu is used to select which functional image will be used in the overlay.

If in either the Base or the Function drop menus an option marked “Individual” is selected, this refers to the individual anatomical/functional image from the current subject – the subject that is selected in the Subject Tab.

## 14.4 Examples

### 14.4.1 How to Overlay Activation Maps onto Anatomical Images

#### (a) Overlaying Common Space Tmaps onto Reference Image

The step-by-step instructions with reference to Figures 14.9 and 14.10 are:

1. Choose Brain Register from the BioImageSuite main menu. Three windows will appear: a Transform Viewer, a Reference Viewer and a BrainRegister menu bar. In the Reference Viewer choose (File — Standard Images — MNI\_T1\_1mm\_stripped), or the filename of your chosen reference brain.
2. In the Transform Window choose (File — Load).
3. Choose the result map you wish to overlay (this map must be already saved in reference space).
4. On the BrainRegister menu bar choose (Help — T-distribution Table). A new window called Critical Points of the T-distribution will appear.
5. On the new Critical Points of the T-distribution window enter the degrees of freedom for your chosen result map. Also, enter the desired t-tailed pvalue. Use the following formulas to determine your degrees of freedom.
  - Single group composite:  $df = \#Subj - 1$
  - Difference between two groups of subjects:  $df = (\#SubjGrp1 + \#SubjGrp2) - 2$
6. On the Critical Points of the T-distribution window click the “P(2t)->T” button. This will calculate the T-value needed (shown in green box) for your degrees of freedom and chosen p-value.
7. On the BrainRegister menu bar choose (Registration — Functional Overlay). A new Registration/Overlay Tool window will appear.
8. On the new Registration/Overlay Tool window enter the t-value displayed on the Critical Points of the T-distribution window.
9. On the Registration/Overlay Tool window click “Create Overlay”.

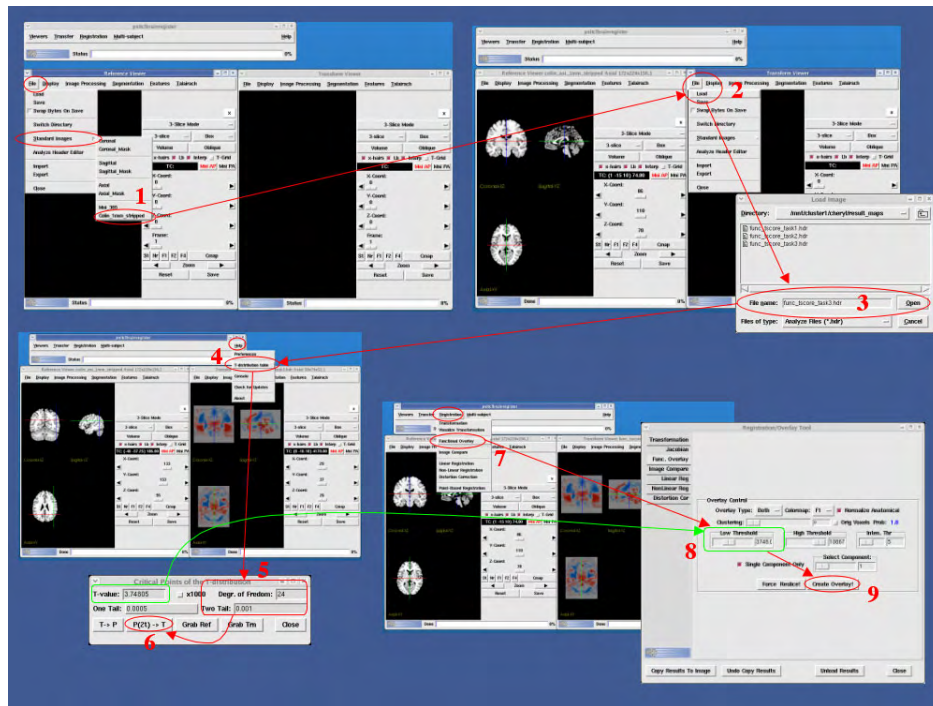


Figure 14.9: Methods used to overlay a result map in reference space onto the reference anatomical image.

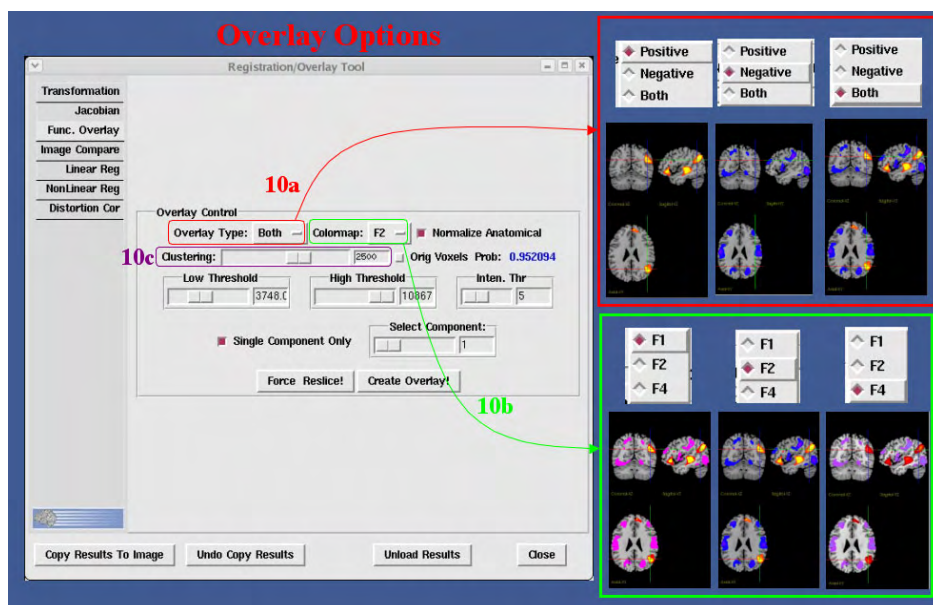


Figure 14.10: Methods used to overlay a result map in reference space onto the reference anatomical image – Part II.



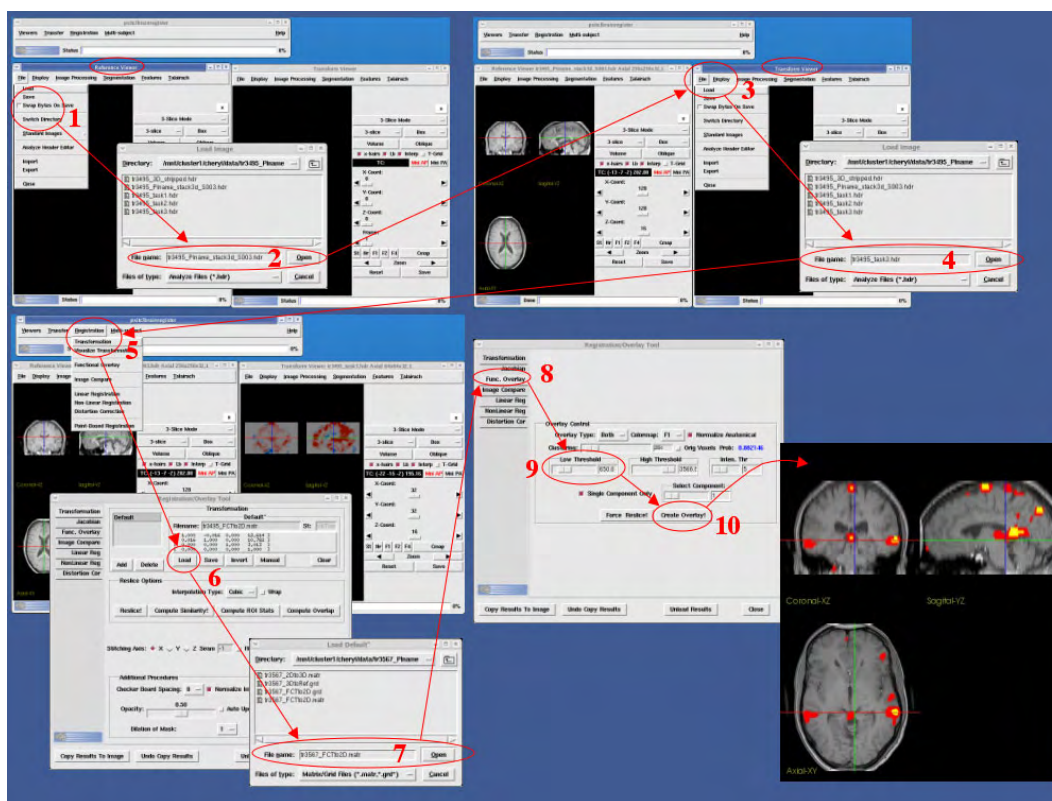


Figure 14.11: Methods used to overlay a result map in individual space onto the individual 2D anatomical image.

10. There are several options the user can manipulate to display different results other than the threshold. –See Figure 6a image II.
  - (a) Overlay Type (shown in red box): Positive, Negative or Both. This toggle button changes what values of the image are displayed.
  - (b) Colormap (shown in green box): F1, F2, F4. This toggle button changes which colormap is used to display the results. The user can change the default colormap by going back to the BioImage Suite main window and choose Preferences (as seen in Figure 1). A new BioSuite User Preference window will appear and the user can choose the desired default colormap. Then choose Save to save your default preferences.
  - (c) Clustering (shown in purple box): This slide bar can generate a cluster size and the displayed image will contain only clusters that are greater than the chosen cluster size.

## (b) Overlaying Individual Tmaps onto Individual 2D Anatomical

The step-by-step instructions, with reference to Figure 14.11 are:

1. Choose brainregister from the BioImageSuite main menu. Three windows will appear: a transform viewer, a reference window and a brainregister menu bar. In the Reference Window choose (File — Load)

2. Choose the filename that refers to the conventional 2D thick slice anatomical image and click Open.
3. In the Transform Window choose (File — Load)
4. Choose the task t-map or percent signal change map that is chosen to be overlaid onto the anatomical. Click Open.
5. On the BrainRegister menu bar Choose (Registration — Transformation). A new Registration/OverlayTool window will appear.
6. Under the Transformation block Click Load.
7. Choose the functional or echoplanar to 2D registration and Click Open. DO NOT CLICK RESLICE!
8. On the Registration/Overlay Tool window choose the Functional Overlay tab.
9. Choose your threshold
10. Click Create Overlay. This will reslice the functional map with the chosen registration as well as overlay the resliced image onto the anatomical and display the results in the Transform window.

## 14.4.2 Generating Across Subjects Composite Maps

### (a) Single Group Composite Maps.

First, generate a .msb setup file as described in an earlier section of this document.

The step-by-step instructions, with reference to Figure 14.12, are:

1. Choose brainregister or dualmultisubject from the BioImageSuite main menu. Three windows will appear: a Transform Viewer, a Reference Viewer and a BrainRegister menu bar. On the BrainRegister menu bar choose (Multisubject — Base). A new MultiSubject Tool window will appear.
2. On the new MultiSubject Tool window choose (Setup — Load Setup).
3. Choose your generated .msb file and click Open.
4. On the MultiSubject Tool window choose the Results Tab.
5. Choose your task of interest from the list of tasks then click Change Current Task. If your task of interest is the first task the program will warn you that the new task is the same as the old task. The status bar at the bottom of the window will create a blue bar increasing in length as each subject's filename is renamed to the new task. The word Done will appear next to the status bar when the renaming is complete. If you toggle back to the Subject Tab, you should see your new task in the Functional Image Filename.



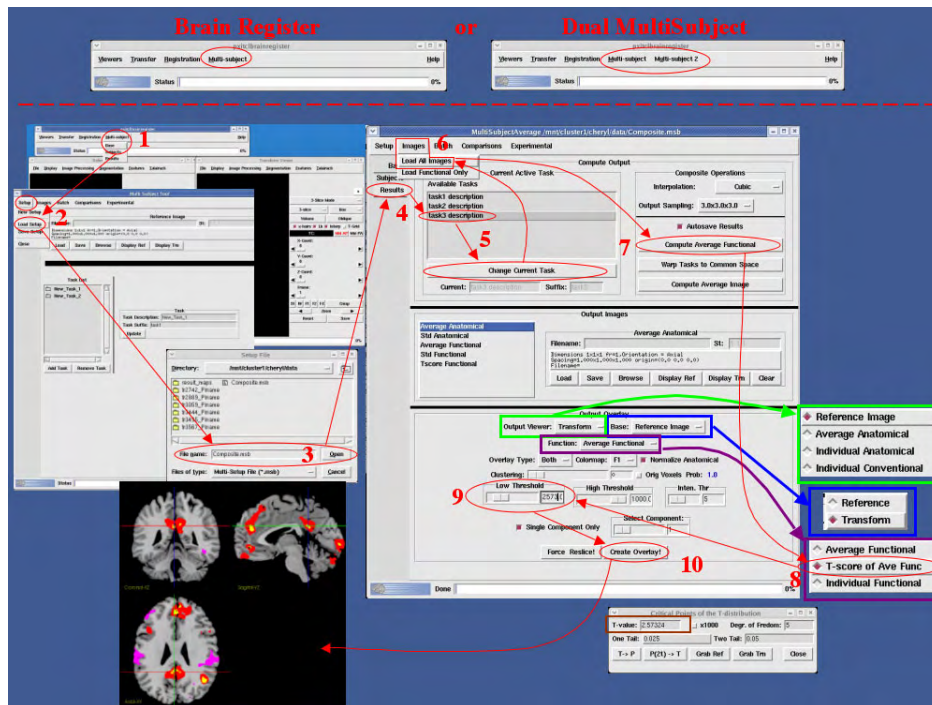


Figure 14.12: Methods used to generate a composite map from a single group of subjects.

6. On the MultiSubject Tool window click (Images — Load All Images). The status bar at the bottom of the window will create a blue bar increasing in length as the number of loaded subjects increases. The word Done will appear next to the status bar when the loading is complete.
7. On the MultiSubject Tool window click Compute Average Functional. The status bar at the bottom of the window will create a blue bar increasing in length as the number of subjects computed into the average increases. The word Done will appear next to the status bar when the computation is complete.
8. On the MultiSubject Tool window the bottom portion is labeled Output Overlay. There are several options to choose that will change the output image using three separate toggle buttons: Output Viewer, Base and Function. The Output Viewer button chooses which window (Reference or Transform) in which the output will be displayed. The Base button allows the user to choose the anatomical image on which to overlay to results. Finally, the Function button allows the user to choose between average function, t-score functional or individual functional. For purposes of this demonstration choose Output Viewer to be Transform, Base to be Reference Image and Function to be T-score of Average Functional.
9. Choose Threshold (See steps 4-6 on the Overlay Common Space Tmaps onto Reference Image Instructions for how to choose a threshold).
10. Click Create Overlay, the program will display the results in the chosen output viewer.

## (b) Two Group T-test Comparisons

Generate two separate .msb files for each group as described earlier in this handout.

With these available, the step-by-step instructions, with reference to Figure 14.13, are:

1. Choose dualmultisubject from the BioImageSuite main menu. Three windows will appear: a Transform Viewer, a Reference Viewer and a BrainRegister menu bar. On the BrainRegister menu bar choose the first (Multisubject — Base). A new MultiSubject Tool window will appear.
2. On the new MultiSubject Tool window choose (Setup — Load Setup).
3. Choose your generated .msb file for your first group and click Open.
4. On the MultiSubject Tool window choose theResults Tab. Choose your task of interest from the list of tasks then click Change Current Task.
5. On the MultiSubject Tool window click (Images — Load All Images).
6. On the MultiSubject Tool window click Compute Average Functional.
7. On the BrainRegister menu bar choose the second (Multisubject2 — Base). A second MultiSubject Tool window will appear.
8. On the second MultiSubject Tool window choose (Setup — Load Setup).
9. Choose your generated .msb file for your second group and click Open.
10. On the MultiSubject Tool window choose theResults Tab. Choose your task of interest from the list of tasks then click Change Current Task.
11. On the MultiSubject Tool window click (Images — Load All Images).
12. On the MultiSubject Tool window click Compute Average Functional.
13. On either Multisubject window click the Base tab and under the Reference window click Display Reference.
14. On either Multisubject window under Comparisons click Show Compare Tool. A new Registration/Overlay Tool window will appear and it will be on the Compare Image tab. The filename boxes will be empty.
15. From the first Multisubject window (Group 1) under Comparisons click Send Mean Functional to Compare Tool as Set1. This will fill in the first filename box on the Registration/Overlay Tool window.
16. From the second Multisubject window (Group 2) under Comparisons click Send Mean Functional to Compare Tool as Set2. This will fill in the second filename box on the Registration/Overlay Tool window.
17. On the Registration/Overly Tool window click Compute Tmap. The computed functional image will be displayed in the transform window.

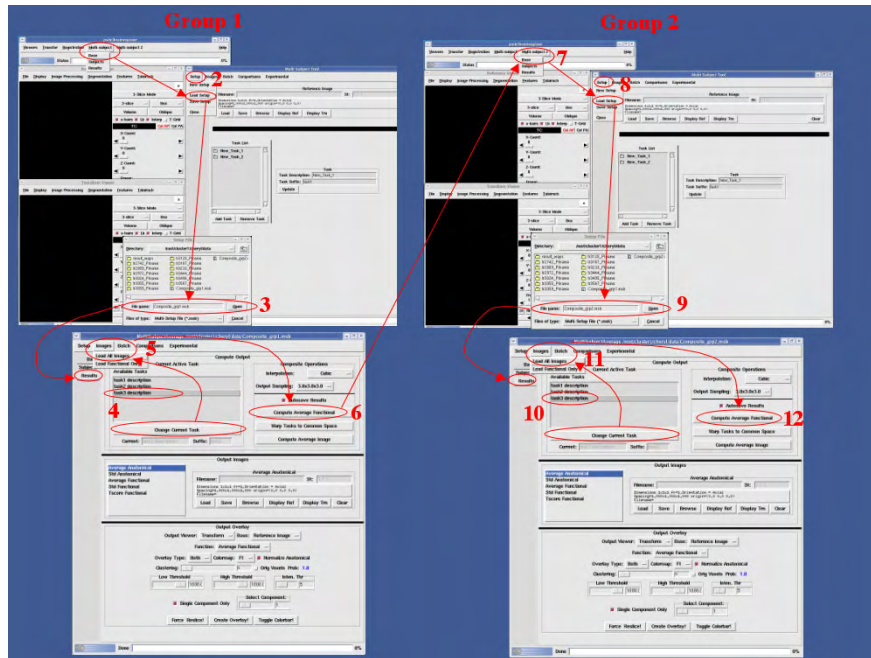


Figure 14.13: Methods used to generate a composite map from a single group of subjects – Part I.

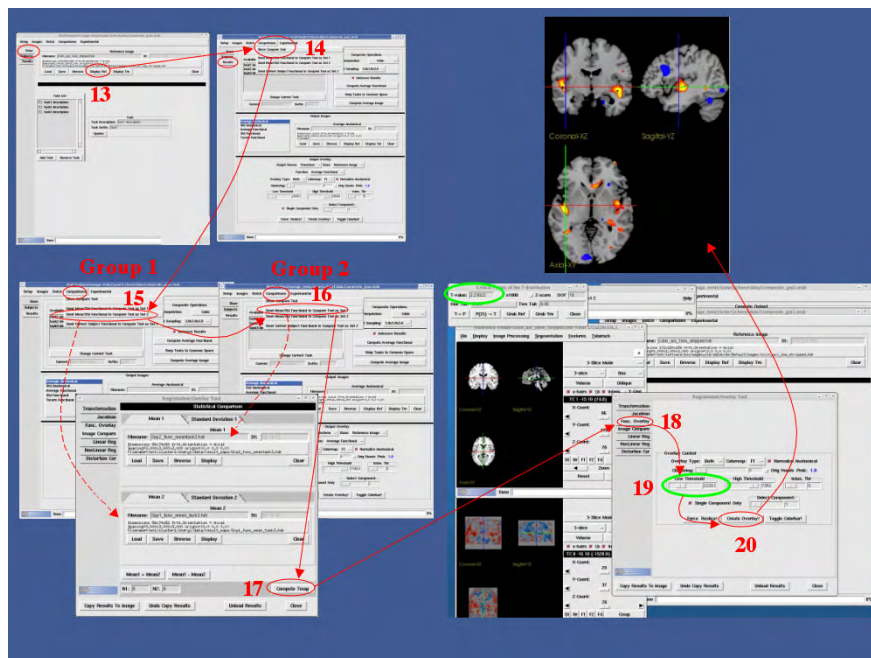


Figure 14.14: Methods used to generate a composite map from a single group of subjects – Part II.



Figure 14.15: Creating Multislice Displays. The snapshot shown is from an earlier version of the Simple Viewer. The newer version allows for multiple images to be displayed along different rows. See Section 14.5.

18. On the Registration/Overlay Tool window click the Functional Overlay tab.
19. Choose your threshold and enter it into the Low Threshold box. Refer to instructions for 5a (steps 4-8) for an explanation on how to choose a threshold.
20. On the Registration/Overlay Tool window click Create Overlay.

### 14.4.3 Creating Multi-Slice Image Displays

Once a Tmap has been overlaid onto an anatomical, it can be viewed in 3D space in the Transform window, and a separate window called the Simple Viewer can generate multi-slice images. You can accomplish this by either choosing `pxitclsimpleviewer` straight from the BioImageSuite menu or you can operate the same window through `pxitclbrainregister` or `pxitcldualmultisubject`.

The step-by-step instructions, with reference to Figure 14.15 are:

1. Choose `brainregister` or `dualmultisubject` from the BioImageSuite main menu. Three windows will appear: a Transform Viewer, a Reference Viewer and a BrainRegister menu bar. Generate your image of choice with any of the above described methods. On the BrainRegister menu bar choose (Viewers — Simple Viewer). A new Simple Viewer window will appear.



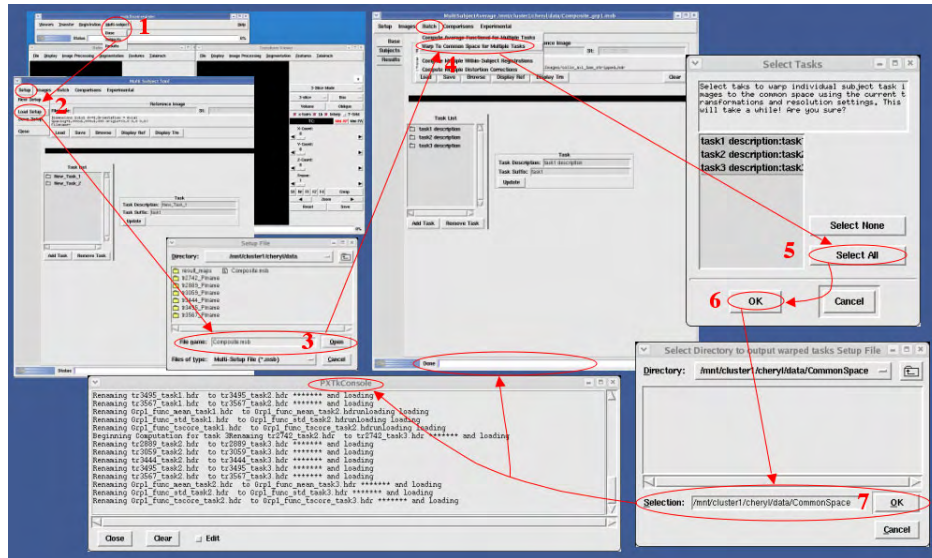


Figure 14.16: Warping single subject data to common space.

2. On the new Simple Viewer window click ( Display — grab image from transform viewer).
3. Choose orientation of images by using the toggle button.
  - (a) Axial - XY
  - (b) Coronal - XZ
  - (c) Sagittal - YZ
4. Choose how your images will be displayed.
  - (a) Choose number of viewers or how many slices will be displayed.
  - (b) Choose the First Slice to begin viewing your slices.
  - (c) Choose the Increment or how many images to skip in between viewed images.
5. Toggle the labels On (box is red) or Off (box is the color of the background). The labels will display the slice number in image space and the orientation of the slice.
6. The arrows around the Zoom will allow the user to increase or decrease the image size of all slices. The user can also reshape the size of the entire window to decrease the amount of black space around each slice.
7. Click Save to save the image displayed on the screen. Be careful not to have the Save window over your image or you will save a figure of the Save Window instead. The file type for the image being saved can be toggled between jpeg and tiff file formats.

#### 14.4.4 Warping single subject data to Common Space

Individual tmaps can be saved with all the transformations applied, thus being placed into the Common Reference Space.

The step-by-step instructions, with reference to Figure 14.16 are:

1. Choose `brainregister` or `dualmultisubject` from the `BioImageSuite` main menu. Three windows will appear: a `Transform Viewer`, a `Reference Viewer` and a `BrainRegister` menu bar. On the `BrainRegister` menu bar choose (`Multisubject — Base`). A new `MultiSubject Tool` window will appear.
2. On the new `MultiSubject Tool` window choose (`Setup — Load Setup`).
3. Choose your generated `.msb` file and click `Open`.
4. In the `MultiSubject Tool` window click (`Batch — Warp to Common Space for Multiple Tasks`). A new `Select Tasks` window will appear.
5. On the new `Select Tasks` window choose which tasks you want to warp to common space by highlighting them or just click `Select All`.
6. On the new `Select Tasks` window click `OK`. A new `Select Directory` window will appear.
7. On the new `Select Directory` window choose the directory for the output images. Click `OK`.
8. A new `PXTkConsole` will appear showing the status of the batch job. Also the `Status bar` at the bottom of the `MultiSubject Tool` window will lengthen and display what is being done. Upon completion the word `Done` will appear next to the `Status bar`.

## 14.5 The new SimpleViewer Tool

As of `BioImage Suite 2.5`, the simple viewer tool has been augmented to allow for the display of multiple images along different rows. This control “takes” images from the two other main viewers (`Reference` or `Transform`). The first row is always populated using the “`Grab Image from ..`” options. Additional images can be grabbed using the “`Grab Aux Image`” buttons as shown in Figure 14.17. Each auxiliary image corresponds to a row (beginning at 0 = main image) of the viewer. The “`Clear Aux Images`” option can be used to reset the viewer to single image mode. When multiple images are displayed, the number of rows is fixed to the number of images.

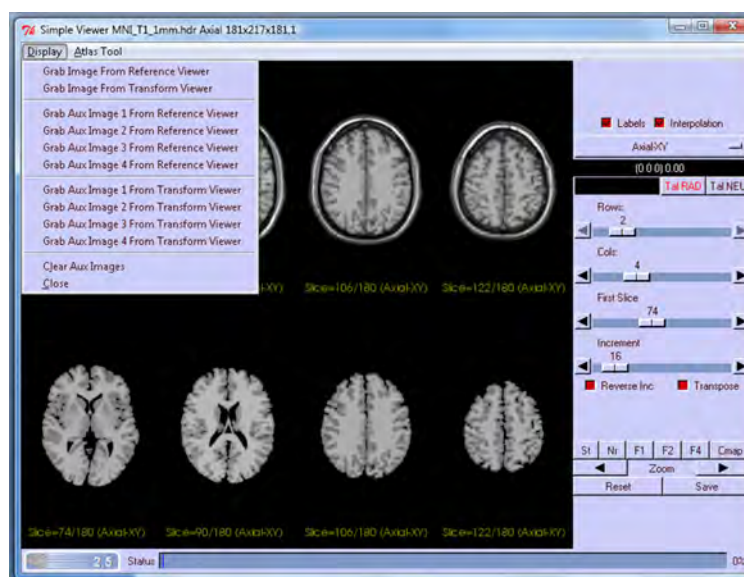


Figure 14.17: The new simple viewer tool. Two images are shown, one is the unstripped brain (top row) and the second is a stripped version (bottom row). Upto 5 different images can be displayed at once; they must have identical dimensions, e.g. everything must be in common space.



## Chapter 15

# The Data Tree Manager

### 15.1 Introduction

The Data Tree Manager Tool, shown in Figure 15.1 is a powerful utility for managing and working with whole sets of images, surfaces, transformations, and other data relevant to both clinical and research applications. All of the images and their relevant transformation files are stored in a configuration defined by a “tree” style hierarchy, allowing you to quickly visualize which transformations are available, and which need to be calculated in order to perform a given operation. Furthermore, multi-subject operations are easily automated, and associated data management facilities are included.

The top menubar of the Data Tree Manager contains menus that allow you to access the viewers directly (see Chapter 5), as well as the Registration tools which operate on the images contained in the viewers at any given time (see Chapter 10). Thus, the datatree is a convenient central organizing point for many of the operations that BioImage Suite performs.

### 15.2 The Tree

The hierarchy tree structure is the heart of the Datatree Manager. The following sections describe how to work with the tree structure and take advantage of its features.

#### Image Tree structure and function

The left hand panel of the Data Tree Manager, shown in Figure 15.2, displays the tree hierarchy. This panel is the most important part of the tool. It shows not only which files (images, surfaces, and transformations) have been loaded, but also these relationships between the images and surfaces. In this tree hierarchy, the images and surfaces are shown as nodes. A node located directly above

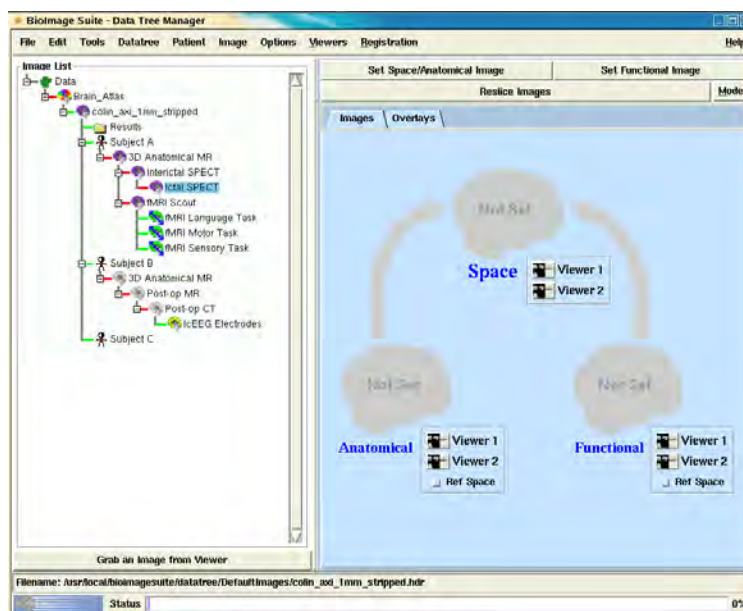


Figure 15.1: The Data Tree Manager

another node is known as the parent of the second node while the second node is known as the child of the first node. Transformations are shown as the lines in between the nodes and must be computed for each parent-child pair. Once the transformation between a child and parent is computed, the child will inherit all of the transformations that can be applied to the parent. This relation between parents and children makes calculating several permutations of transformations easier and less time consuming.

### Building a tree

The tree is built by adding children to already existing parents. As a result, the top parent in the tree, “Data,” cannot be deleted. As other nodes are added to the tree, sibling nodes are arranged in ASCII order. Thus, giving a node a name that begins with 0 will force that node to ahead of its siblings in the tree display. (A convenient naming convention is to begin all names with a number, allowing you to explicitly set display order).

### Adding a node

To add a node to the tree, right-click on the existing node where you would like to add the new node as a child. A pop-up menu, shown in Figure 15.3, will appear, from which you may select “Add an Image as Child,” “Add an ROI Definition,” or “Add a Surface as Child.” The Surface choice has a flyout which allows you to add either a Surface, Electrode Grid, or Landmark Set. If you choose “Add an Image as Child,” a dialog box will appear with a list of standard choices for image modality. If you choose one of these, it will become the default title of the node. A custom title can be created via the entry box below the list. The advantage to selecting a title from the list is that the image type will be automatically set to an internal number convention. These

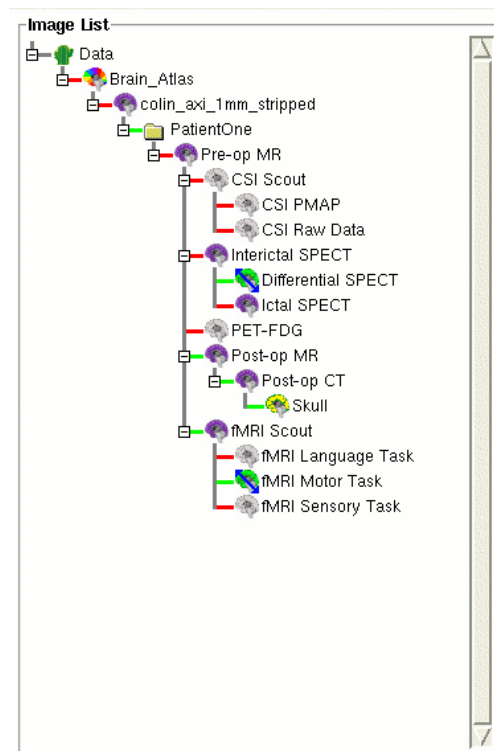


Figure 15.2: The tree hierarchy. The images with purple brain icons exist, while those with grey icons have not been loaded. Similarly, transformation with green lines exists, while those with red lines have not been loaded.

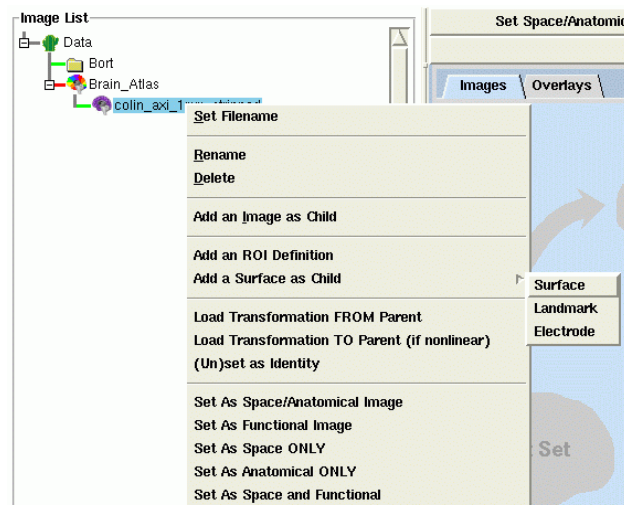


Figure 15.3: Adding a Node to the Tree. Right clicking on an image node brings up this menu, from which you can choose to add an Image, an ROI Definition, or any of the Surface Types in the flyout.

conventions can be edited from the Options menu (See the section on Image Types). The newly added node is just a placeholder until the file that holds that actual image is specified.

The same menu as described above is available under the “Image” menu heading in the menubar. The commands in the menu will be performed relative to the currently selected node in the tree.

### Specifying a Node’s Filename

In the right-click pop-up menu or the “Image” menu in the menubar, select “Set Filename.” A file selection dialog box will appear. Once the filename has been set, the node will turn from gray to purple. Also, the file path will appear in the status bar when the node is selected.

### Renaming and Deleting Nodes

To rename or delete a node, right-click on it. In the pop-up menu, select either “Rename” or “Delete.” If “Rename” is selected, a dialog box will appear asking for the new name for the node. The node will then either be renamed or deleted, and the tree will be rearranged to reflect the new order of nodes. Remember that the nodes are displayed in ASCII order. Numbers come first, capital letters come next, and lowercase letters come last.

### Node Types



**Image** Image nodes are the most important type of node in the tree. Each one contains a single image file.



**Folder** A Folder node functions as a grouping mechanism, but has no significance in the calculation of transformations. It is inherently an identity; essentially “passed over” when concatenation of transformations is taking place. It can sit in the tree at any location, since it functions as an identity when being concatenated into the result transformation, if it is in intermediate in the transformation path.



**Patient** A Patient node is actually a “folder” containing all data for a given patient. What makes them different from a folder node is the fact that they can have an arbitrary array of subject measures attached to them, which can be useful for statistical comparisons etc. These properties are set in the Node Info Box for the patient nodes.



**Surface** A surface is a \*.vtk file type that specifies a surface extracted from an image. Surfaces cannot be added as children of Patients or Folders (or the root).



**Electrode** An electrode grid is a special variety of surface that specifies electrode locations in the brain. These also can only be added as children of images.



**Landmark** A landmark definition is also a surface file that specifies any set of landmarks that you indicate. These also can only be added as children of images.



**Atlas** An atlas is a special type of image that specifies an anatomical atlas which has marked regions and a text file associated with it that identifies them. If an Atlas node is included in the tree and appropriately registered to its children, then the Atlas Tool can be used to navigate through any image registered to the atlas (directly or indirectly through the tree).



**Results** Results nodes are created by the program when multisubject calculations are performed.



**ROI Definitions** An ROI definition is a mask image that has discrete regions. These are used in multiple subject calculations. ROI definitions must be added explicitly as such, by using the ( Image — Add an ROI Definition ) menu option, or selecting that command from the right-click pop-up menu in the tree.

*Note:* The icons for some of the types can be changed in (View — Choose Icon Colors ). The flyouts have color choices for the brain icons.

## Transformations and Identities

A transformation is simply the file generated by the registration process. It defines the function for mapping a point in one image or surface to the corresponding point in another image or surface. Bioimage Suite uses transformations stored in either \*.matr files (linear transformations) or \*.grd files (non-linear transformations). It is easy to see that linking nodes together with transformations is useful, since it allows you to significantly decrease the number of individual registrations that need to be calculated. Once the relationships in a chain of nodes are defined, and the transformations between parent and children are identified, any image or surface in the datatree can be mapped into the space of any other image or surface in the datatree. The tree hierarchy display shows this explicitly with the use of dynamically colored connecting lines.

**Transformation Line Colors:** Each line represents two transformations; a transformation to the parent and a transformation from the parent. The color of the line that lies horizontally to the left of a node gives the information about the transformations to and from that node's parent. If the line is red, then neither transformation has been specified. If it is green, then both transformations have been specified. If the line is yellow, then only the transformation **FROM** the parent has been specified. If it is purple, then only the transformation **TO** the parent has been specified.

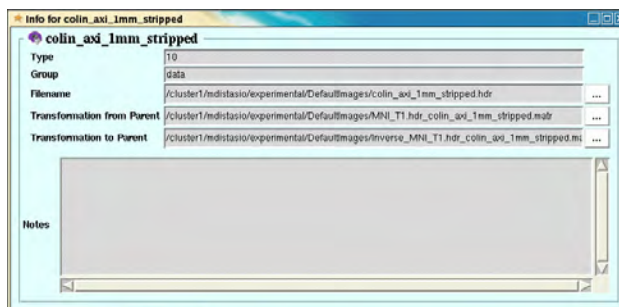


Figure 15.4: The Node Info Box. The name of the node is displayed as the title. Here you can get and set the filename of the image, as well as the transformations to and from its parent.

**Specifying Transformations:** The transformation between parent and child nodes is specified at the child node. Right-click on the node, and select “Load Transformation *FROM* Parent.” A file selection dialog box will appear allowing you to choose the transformation file. For linear transformation, the transformation to the parent or inverse transform is automatically calculated and does not need to load into the datatree. However, the inverse of a non-linear transformation must be loaded independently of the original transformation. To load the inverse of a non-linear transformation, right-click on the child node, and select “Load Transformation *TO* Parent.” A file selection dialog box will appear allowing you to choose the transformation file. See Chapter 10 for more details on inverting transformations.

### Node Info Dialog Box

Double-clicking on a node in the tree brings up the Node Info Dialog Box. The Node Info Dialog Box is shown in Figure 15.4.

**Viewing the Node Info Dialog Box:** The node’s icon and title are displayed at the top of the Node Info Dialog Box, and more detailed information about the node is available below. Here, you can view the node name, type, filename, and the filenames of the transformations that apply to the node. For the fields that specify a file associated with the node (“Filename,” “Transformation *FROM* Parent,” and “Transformation *TO* Parent,”), a button with an ellipsis (“...” ) is placed next to the field. Once clicked on, a file selection dialog box appears, allowing the file to be set directly. This change will be reflected immediately. The large box at the bottom of the Node Info Dialog Box is a space for notes. Anything that is typed here will be saved when the Node Info Dialog is closed. All of this data is saved when you save a tree.

**Extra Information for Patient Nodes:** The Node Info Dialog Box for patient nodes contains a set of customizable fields that can be set to contain any patient data that you would like to note. Currently there are no facilities for performing statistical calculations on images along with these values, but this will be implemented soon. To define which patient attributes you want to include, use ( Options — Edit Patient Property List ).

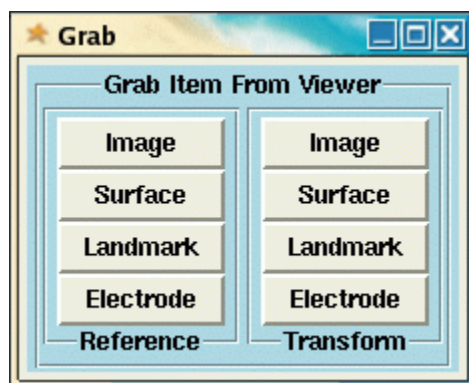


Figure 15.5: The Grab Box. This small dialog box allows you to grab any of the items in it from the corresponding data in either viewer and place it into the tree.

### Grabbing an Image from the Viewer

If you want to add a node to the tree based on the image data that is currently displayed in the viewer, select a node in the tree for the parent and click the **Grab Image From Viewer** button (located at the bottom of the tree display). The image to be grabbed must be in the “Image” Display of the viewer (see Chapter 5). The Grab Box, the small dialog box shown in Figure 15.5, will pop up and show the different types of objects that you can grab from either of the two viewers. Simply click “Image,” “Surface,” “Electrodes,” or “Landmarks,” for the viewer you wish to grab from. A file selection box with the appropriate file type will appear, asking for the newly grabbed file to be saved. Once saved, the file will appear in the tree, as a child of the node that was initially selected initially. It will be named “Grabbed\_Image\_\*” with \* = *some number* to differentiate it from other grabbed images. The name can be renamed by using the “Rename” feature describe above. (You will still be able to create a file even if there is no data of the correct type in the viewer - a node will be added to the tree, but it will contain nothing).

### Cutting and Pasting Parts of the Tree

If you want to move a part of the tree (subtree) to another part of the tree, or make a copy of the part, you can use the Cut, Copy, and Paste commands. These features are found in the Edit menu and work just like Cut, Copy, and Paste, in any text editor:

- When you select a node and choose the ( Edit — Cut ) menu item, the node is deleted, and it is placed, along with its subtree of children, onto the clipboard.
- When you select a node and choose the ( Edit — Copy ) menu item, a copy of the node and its children is placed onto to clipboard, but the node remains in place.
- When you select a node and choose the ( Edit — Paste ) menu item, the contents of the clipboard are added to the tree as descendants of the currently selected node.



This is useful if you have a section of the tree that needs to be repeated over and over. You can simply copy and paste a set of images multiple times, making tree building much quicker than defining one image at a time.

### 15.3 Space, Anatomical, and Functional Images

The right side of the Data Tree Tool display, shown in Figure 15.6, shows the images that are currently being worked with, and potentially sent to the viewers. The three brain space represent images, surfaces, electrodes, or landmark data, and the two arrows represent transformations. Images are resliced into the coordinates of the image in the “Space” location. Most often, you will set the “Space” and “Anatomical” images to be the same image. Thus, when an overlay is created, the image will appear in the coordinates of the image that is shown as the anatomy. The “Functional” image will be resliced into the space defined by the “Space” image, so that when displayed, the function data coincides with the correct anatomical loci, based on the registration.

**What is Reslicing?** Reslicing is the process of sampling an image and creating a new image from it that has slices that coincide with the slices of another “space defining” image. For example, if the space defining image has one slice for every 1mm in its Z-axis, then any image resliced using this space definition will end up with one slice every 1mm in its Z-axis, regardless of which orientation the slices are in the original. The process of reslicing yields a pair of images in which X, Y, and Z coordinates correspond to the same location in the image, provided that the registration file that maps points from one to the other is valid. See the section on Transformations.

#### Setting Images to be Resliced

To set the space defining image, simply select an image in the tree and click on the **Set Space/Anatomical Image** button at the top of the right side of the Datatree Manager. This will place the selected image into “Space” and “Anatomical” spaces. The display shows two colorful generic brain images above the space title, and the name of the node across it. This lets you know that these slots in the reslicer are filled. *textbfNote:* the arrow between the two spaces turns green.

Clicking the **Set Functional Image** button places the selected node in the tree into the brain space labeled “Functional”. Again, the colorful brain appears in the display, along with the name of the node that you chose.

If you want to set the Space and Anatomical Images separately, you should use the options in the bottom section of the Image menu on the menu bar (which mirrors the popup menu that appears when you right click on an image). Here you can choose **Set As Space/Anatomical Image**, **Set As Functional Image**, **Set As Space ONLY**, **Set As Anatomical ONLY**, or **Set As Space and Functional**. Thus, you can quickly set up any configuration of space, anatomical, and functional data from reslicing.

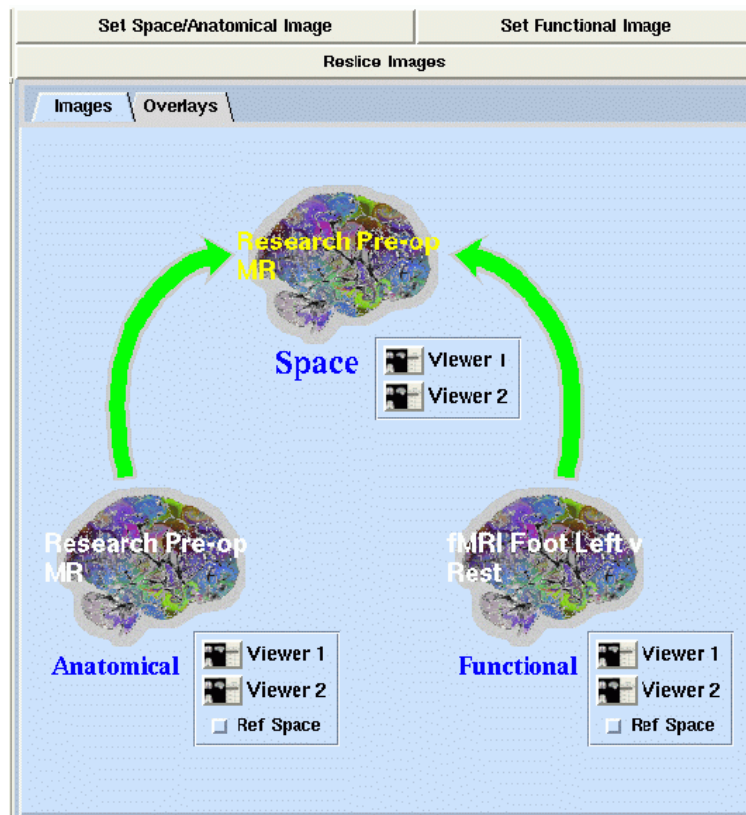


Figure 15.6: The Reslicer. The top “Space” section holds the image into whose space you want to reslice the other images. The other images are resliced to match its resolution and size.

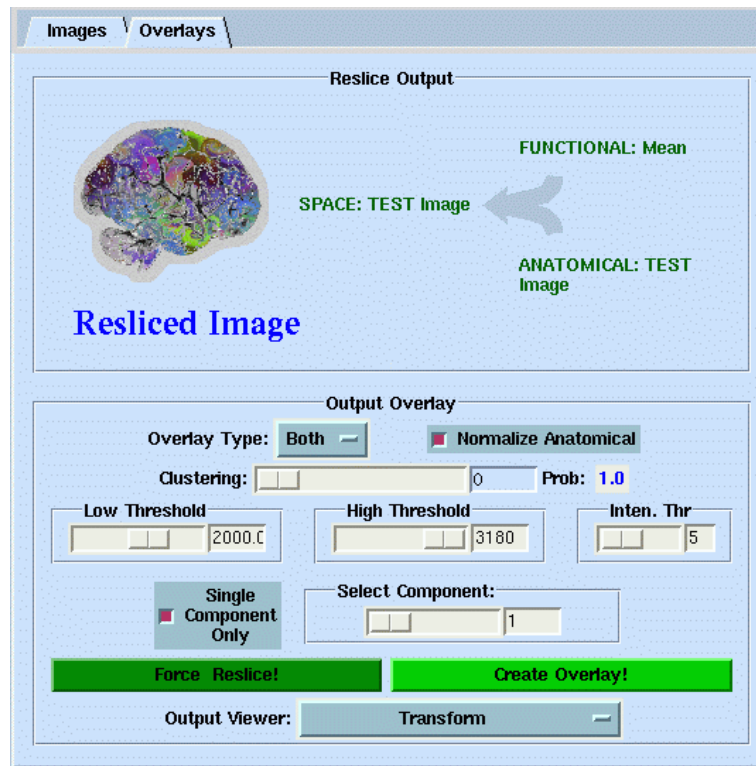


Figure 15.7: The Overlay Tab. The overlay tab in the right side window of the Datatree Manager allows you to create views of functional data ontop of anatomical data with a number of optional settings.

## The Big Arrows: Concatenating Transformations

The large arrows in the right side display that point to the Space Image represent the concatenation of all required transformations that form the path from the Functional and Anatomical images. This visualization quickly shows whether transformation into reference space is available. If all required transformations have been specified in the tree for either pair of images, then the arrow connecting them will be green. If any transformation is missing, the arrow will be red. If you try to reslice two images when they are connected by a red arrow, you will receive an error message which will indicate the first missing transformation in the queue.

## Reslicing - make it happen

In order to reslice images according to the configuration you specified by setting the “Space”, “Anatomical”, and “Functional” images, and connecting transformations, simply click the large **Reslice** button at the top of the right side of the Datatree Manager. The “Anatomical” image will be resliced into the coordinates of the “Space” image; then the “Functional” image will be resliced into the coordinates of the “Space” image. The results will not show up in a viewer immediately; they must be sent to a viewer by using the Overlay Tool. The Overlay Tool is located in the “Overlay” tab. (right under the **Reslice** button).

## 15.4 The Overlay Tab

The “Overlay” tab, shown in Figure 15.7, contains the tools that allow you to combine two images in a meaningful way, taking advantage of the registration file that links them, as specified in the tree. The functional data can be overlaid on the anatomical data, creating “hotspots” of color on a gray background (the color scheme, including the anatomical colors, can be changed with the Colormap Editor).

### 15.4.1 The Reslice Output Box

The output of the reslicing operation is sent directly to the “Overlay” tab. This operation is indicated by the update of the Reslice Output box to show the combination of two images into the space of a third (which often matches the anatomical - see Reslicing). The presence of the colored brain icon image in this box tells you that an overlay image is in memory is ready to be sent to a viewer. To send an overlay to a viewer, just choose a viewer from the “Output Viewer” menu at the bottom of the Overlay Tab (Transform Viewer is the default). Then click the **Create Overlay!** button, and the image will be shipped to the viewer for further analysis or display (see Chapter 5).

### Thresholding the Overlay

The next most commonly used tool in the “Overlay” Tab is the Threshold function. The “Low Threshold” and “High Threshold” sliders allow you to apply a threshold to the functional data before it is overlaid on the anatomical image. Simply set these slider bars, whose range will be automatically set by the range of intensities in your images, and click the **Create Overlay!** button.

## 15.5 Multiple Image Calculations

One of the advantages of the tree tool is that it brings all your data together in one place, and lets you work with all of your images and other data in the space of any other image in the tree. This functionality is very useful when it comes to doing group comparisons, averaging, and other multiple image operations.

### Pre-processing

Before doing calculations on a group of images, they must be registered and arranged in the tree, as described in the Registration page, and in The Tree, above. Then you must select which space you would like to perform the calculations in. The rest of the images will be transformed **TO** this space before comparisons and calculations are made. Set the image you want to use to define the

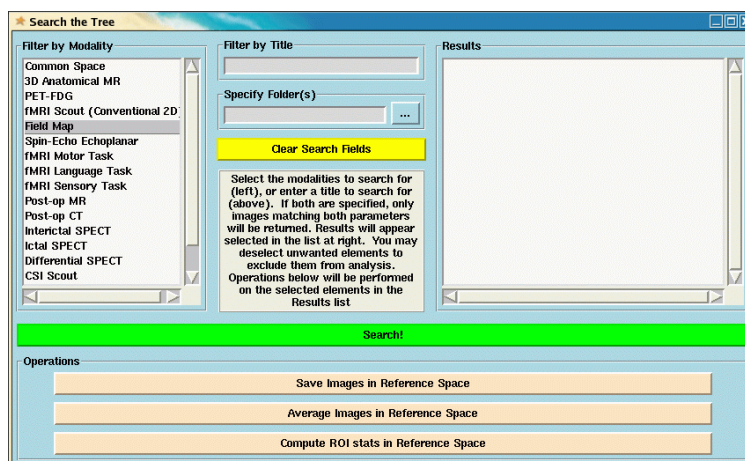


Figure 15.8: The Search Box. This dialog box is accessed via the ( Tools — Multiple Image Calculations ) menu option. It contains fields for searching the tree for images by name, type, or containing folder. The results can then be fed into image statistics tools.

space and set it as the “Space” image (using either the **Set Space/Anatomical Imag** button, the Image menu, or the pop-up menu in the tree.

## The Multiple Image Search Dialog

All multiple image calculations are performed from the “Multiple Image Search Dialog”, which is available in the Tools menu and shown in Figure 15.8. The steps required are simple: Define which Images should be operated upon, and click the button to perform the operation!

### Choosing your images

The top section of the dialog box (See Figure ) is devoted to searching the tree. The images that will be operated on are those images that are in the “Results” section *and are selected* . There are three ways you can define which images to include, and they are complementary. Thus, you can search by image type, image title, containing folder, or any combination of these.

- The “Filter by Modality” section contains a list of the image types currently defined (See Image Types). You can select one type by clicking on it in the list, or multiple types with Ctrl-clicking. Then click the **Search!** button. All images of that type will be placed in the “Results” list, and selected by default. If you do not want to select any image types, just Ctrl-click on the selected item in this list, and the selection will disappear, meaning that all image types will be included in the search.
- If you specify a search string in the “Filter by Title” entry field, only nodes that match this title will be placed in the “Results” list when you hit the **Search!** button. If you leave it blank, all titles will be included in the search.

- You can select a folder in the tree to search. Click the “...” button in the “Specify Folder(s)” section, which will bring up a list dialog box containing all the folder names present in the tree. Select one (or more with Ctrl-click) folder(s) in the list and click **OK**. Now, only nodes within the folders selected will be included in the search.

Once you have a list of nodes in the “Results” section, you can manually select those upon which you want to do some calculations. By default, all items in the “Results” list are selected, but you can deselect them one at a time by just left-clicking on them.

### Doing the Calculations

The functions that are available in the lower half of the dialog box are subject to change, since we often identify new multiple image calculations that are useful. To perform any of these calculations, just click the button!

**Save All Images in Reference Space:** This button queries the user for a directory, and then saves a copy of each result image, transformed to the reference space (as set in the main window).

**Average Images in Reference Space:** This button transforms all images to reference space, and then averages them, creating three result images, which are saved, and placed automatically into a folder in the tree: Average, STD Dev, and T-map. These three images are in the space of the “Space” Image, and thus the folder is created as a child of this image.

**Compute ROI Stats in Reference Space:** This function relies on there being an ROI definition image in the tree and linked to the other images by valid transformations. When you click this button, the program will search the tree for all ROI definition images (which must be added as this type specifically - see the bottom of the Node Types box). A list dialog box will appear, in which you should select just one ROI image. This ROI image, along with all the “Results” images will be transformed to the reference space, and statistics will be computed in each image for each region defined in the ROI image. The output is a tab-delimited text file, which you will be prompted automatically to save. This numerical output will also be displayed in a console window that pops up.

## 15.6 Functionality for Intracranial Electrode Attributes

An Electrode Node (see Node Types) is meant to have a \*.mgrid file loaded into it. These files specify the locations of a set of electrodes. Once you have loaded an \*.mgrid file, you can send it to the Electrode Control, which will allow you to display the electrodes over the image that is in the **Transform** viewer. The Electrode Control does not send electrodes to the Reference viewer, so if

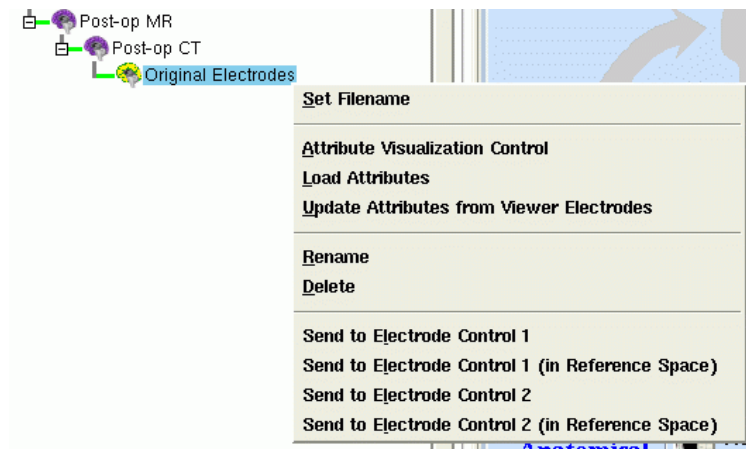


Figure 15.9: The Electrode Menu. This menu appears when you right-click on an electrode node. It contains features for working with electrode objects.

you want to work with an electrode file in relation to your image, send the image to the Transform viewer.

## The Electrode Menu

The following menu commands are available for electrode nodes:

- **Set Filename:** As for all nodes, this allows you to load a file, to which this node will become a pointer.
- **Attribute Visualization Control:** Sends the node's attributes to the visualization control, described below.
- **Load Attributes:** Load an electrode attribute file for the node (see next section).
- **Update Attributes from Viewer Electrodes:** Sets the node's attribute array to that of the electrode file currently displayed in the viewer.
- **Rename:** Changes the title of the node.
- **Delete:** Deletes the node from the tree.
- **Send to Electrode Control (1 or 2):** Sends the node's electrode data to one of the two electrode controls that are linked to the Transform Viewer (Viewer 2). *Both Electrode Control 1 and 2 show their electrodes in Viewer 2.* Thus, two sets of electrodes can be displayed in the Transform Viewer simultaneously.
- **Send to Electrode Control (1 or 2) in Reference Space:** Same as above, except the transformations from the electrode node to whatever image is set as space are applied to the electrode object first.



## Electrode Attributes

**What are they?:** Electrode attributes are simply a binary array that gets attached to an electrode node, and contains flexible information, as directed by an imported loadfile. Simply put, this function allows you to create “categorized” electrodes, and differentiate between them visually and via region of interest calculations.

An example would be instructive: The table below shows an example theoretical structure for an attribute definition for a given set of electrodes.

Elec. #	Elec. Name	On Surface	Depth	Gamma Increase	Gamma Decrease
1	A1	1	0	0	1
2	A3	1	0	1	0
3	A8	1	0	0	1
4	B1	0	1	0	1
5	B2	0	1	0	1
6	B3	1	0	1	0
7	B4	1	0	1	0
8	C6	1	0	1	0
9	D2	1	0	1	0
And so on..					

The electrode numbers and names in the set of electrode attributes to be loaded must correspond exactly with the electrode names of the \*.mgrid file it will be associated with. The rest of the fields are completely customizable (thus, you can name the various headings anything you like - simply by editing the electrode attribute load file).

## Loading Electrode Attributes

In order to load the attributes for the Electrode Attribute dialog box, you need to place them into a tab-delimited file (you should be able to cut and paste from a spreadsheet, for example) with a header that matches the “electrode attribute template”, which looks like this:

```
#Electrode Attribute Description File
#-----
# (space or tab delimited)
electrode# name brain exclude theta_up theta_down beta_up beta_down gamma_up
#-----
```

You can either create a new text file by cutting and pasting the above, or find the header in the template file **ElectrodeAttributeTemplate.ele** , in the same directory as the DataTree executables.

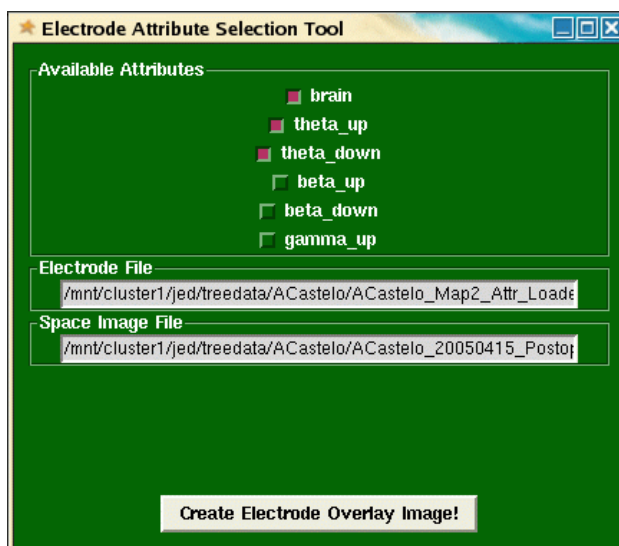


Figure 15.10: The Electrode Attribute Tool. Here we see the tool, accessed via the electrode menu, which allows users to create overlays showing electrodes with differing properties in different layers.

### The Attribute Visualization Control

The function of the attribute visualization control is very simple, it creates an overlay image out of a space image and an electrode file, based on the properties specified by the electrodes' attributes. Thus, a 4D image is created, with the regions on any given layer corresponding to all the electrodes that share the property in the corresponding column. The checkboxes present in the top part of the window represent the attributes that are set, and will change depending on the attribute file that you load. The utility of this is that you can create a different image for any combination of 1 or more attributes. Simply check the boxes of the attributes that you want included in the overlay image, and push the **Create Electrode Overlay Image!** button. The appropriate transformations will be applied, and small spots representing the electrodes will be overlaid on the Space Image (whose filename is displayed for reference in the bottom field of this dialog box). *Note:* As with many operations in the Data Tree tool, the space image must be set for the attribute visualization control to work,

The resulting image is appropriate for use as a ROI definition image, in order to quantify regions of overlap between electrodes and functional data.

## 15.7 Options

### Lock the Space Image

This disables changing of the “Space Image” (see Setting the Reslicer Images). This is useful if you know that you are going to want to always warp your images to the same space, and want to avoid accidentally changing it.

## Image Types

Each image type is set by a number (the default image type group ranges from 10 to 9200, but you can choose any whole number scheme you want). These numbers are arbitrary, and only serve as tags to help with searching. The numbers and associated list of image types are completely customizable, so that when you add images to the tree, the list of image types available can reflect your study's image types.

## Patient Properties

Choosing the ( Options — Edit Patient Property List ) menu option brings up a small list dialog box that has a simple list of attributes that will be shown in the node info dialog box for each patient node.

## Part V

# E. Diffusion Weighted Image Analysis

## Chapter 16

# Diffusion Tensor Image Analysis

### 16.1 Introduction

Diffusion Tensor Imaging (DTI) is a magnetic resonance imaging (MRI) modality that provides information about the movement of water molecules in tissue. When this movement is hindered by membranes and macromolecules, water diffusion becomes anisotropic. In highly structured tissues such as muscle and nerve fibers, this anisotropy can be used to characterize local tissue structure.

In the brain, water diffuses more along white matter fibers (axons) than across them. DTI allows the study of normal tissue as well as changes in development, aging, disease and degeneration. DTI also allows the study of anatomical connectivity in the brain. DTI is also used to study the structure of muscle, such as in the heart.

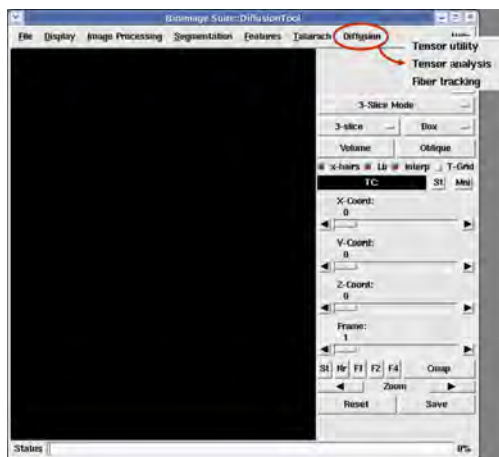


Figure 16.1: Accessing the Diffusion Tools.

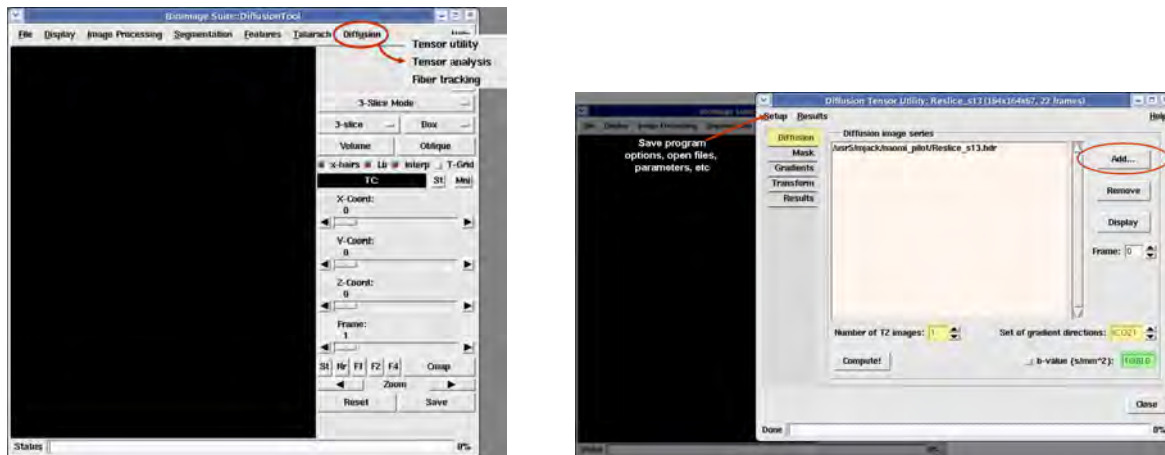


Figure 16.2: Left: Accessing the Diffusion Tools. Right: Loading DWI Image Series

## 16.2 Accessing the Diffusion Tool

In order to use these tools, first invoke the diffusion tool from the main BioImage Suite menu and select from the “Diffusion” menu options. The diffusion tool package is divided into three modules: Tensor Utility (Section 16.3), Tensor Analysis (Chapter 17) and Fiber Tracking (Chapter 18). The **tensor utility tool** reads in diffusion-weighted images and computes the diffusion tensor image. The **tensor analysis tool** reads in the tensor image and computes diffusion-derived maps and region statistics. The **fiber tracking tool** takes in the tensor image and generates fiber bundles and associated statistics.

## 16.3 Tensor Utility

The Tensor Utility tool computes the diffusion tensor (DT) from the diffusion weighted images. In order to invoke the Tensor Utility tool, simply choose the “Diffusion” menu within the diffusion tool, and select the “Tensor utility” option. This tool also generates the ADC (Apparent Diffusion Coefficient) map, the mean DW image (in the case of a multiple DW series acquisition), and an anatomical mask representative of the brain region (using the T2 image acquired along with the DWIs). It allows for a custom set of gradient directions and any number of diffusion-free images (T2). The tensor utility also provides a set of tensor transformation options.

## 16.4 Loading diffusion-weighted images (DWI)

The first step in computing the tensor is to load the DWI series into the program. The DWI series consists of one or more diffusion-free T2-weighted volumes followed by a set of diffusion weighted volumes, one for each diffusion gradient direction. The series must be in Analyze/NIFTI format, and it must be a 4D image (multiple 3D volumes or frames, as in fMRI acquisitions). The file should be comprised of  $n$  diffusion-free images and  $m$  diffusion-weighted images. For example,

consider a DW acquisition consisting of one T2-weighted frame and 21 diffusion-weighted frames corresponding to 21 diffusion gradient directions:

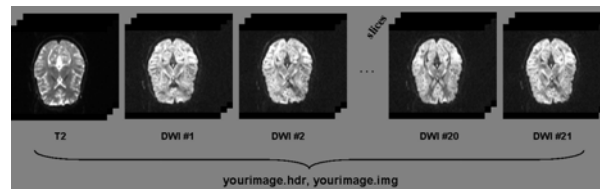


Figure 16.3: The Structure of the Input Image

The file must be ordered such that all T2-weighted frames come first (in this case, the first frame), followed by all frames of the diffusion-weighted data with the DW frame corresponding to the first gradient direction next, and so on (frames 2 through 22). You can add any number of series by clicking the **Add** button, but all of them must have the same size. After an image series is loaded, the program will attempt to guess the number of T2 images and gradient directions from the total number of frames. Always check if these numbers match your acquisition parameters.

**The b-Value:** The b-value is set by default to  $1000 \text{ s/mm}^2$ . If you would like your diffusion maps (ADCs, Trace and Mean Diffusivity) to contain absolute diffusivity values, set the checkbox to allow the entry of the specific b-value used in your acquisition. The b-value will have no effect on indices such as RA or FA, and will not impact fiber tracking.

**Note:** The `pxmat_create4dimage.tcl` script can be used to combine a number of 3D volume images into a single 4D image.

## 16.5 Specifying gradient directions

Once you have loaded your images, check whether the number of gradient directions corresponds to your acquisition's and if the set of directions matches the prescribed ones.

The tensor utility comes pre-loaded with different sets of common gradient directions. Select the set which corresponds to your acquisition protocol.

Alternatively, you can load in the `tensor.dat` file which contains a number of predefined gradient sets (This can be downloaded from the BioImage Suite webpage). If you would like to create your own set, you must first create a text file using your text editor and input your directions according to the following format:

```
n
x1 y1 z1
x2 y2 z2
.....
```



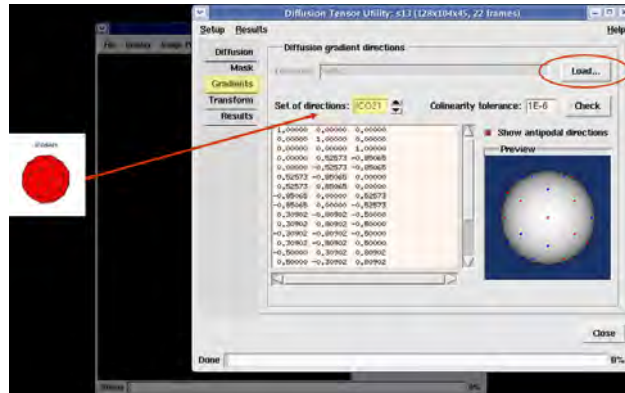


Figure 16.4: Specifying Gradient Directions.

xn yn zn

where  $n$  is the number of directions, and  $x_i, y_i, z_i$  are the  $x, y, z$  coordinates of the  $i$ th direction, separated by spaces. This file can contain multiple sets of directions: simply append each new set after the previous, obeying the format above. Save this file with a `.dat` extension, and load it via the **Load** button in the Gradients pane. The new set of directions should appear in the list, and it should also be depicted in the Preview window. There you will be able to interact with the sphere model and observe the direction distributions. To reset the Preview window, press the 'r' key.

For unbiased results, one should at least sample the diffusion space using noncollinear directions. To check whether your set contains collinear directions, press the **Check** button.

## 16.6 Loading a mask

The tensor tool by default creates a binary mask of the tissue region using the diffusion-free images (T2). The tensor will be calculated only for points that belong to this mask. Points outside the mask will be assigned 0-tensors.

The tensor tool uses simple histogram thresholding to create this mask. In case you would like to load your own, first uncheck the option “Compute from unweighted diffusion series”, then click Load to read your file. This image must be in Analyze/NIFTI format, with the same size as your volume and with values 1.0 representing the foreground (the mask), and 0 for the background.

Alternatively, you can threshold the image manually by disabling the Auto-threshold feature. The Trace connectivity option, when enabled, turns on a morphology-based algorithm that keeps only connected voxels in the masks. Remote “islands” are removed. The center point of the image is used as the center of the mask.

## 16.7 Computing the tensor

Once you have loaded the DWIs, checked the number of T2s and gradient directions and specified the appropriate anatomical mask, you are ready to compute the diffusion tensor by pressing the button **Compute!** in the Diffusion pane. The status bar will display the task progress and once the tensor computation is done, you will be automatically taken to the Results pane. You are then given the opportunity to save the diffusion tensor as well as the other results that were computed. To save a single result, select the desired item in the Results list box, and click the **Save** button.

By default, the diffusion tensor is the result displayed when the calculation is done. In order to display other results such as the mean diffusion-weighted image or the apparent diffusion coefficient, simply select the desired image and click the **Display** button.

**Prefix:** Each result is associated with a suffix that will be appended to the name given in the Prefix field. For example, if your prefix is “s13\_”, the resulting tensor will be saved under the name “s13\_dti\_tensor”. You can freely alter the prefix, but suffix convention is fixed. The prefix is normally defined by the name of the last DWI image series that was loaded (It will copy the name until an underscore symbol is found).

## 16.8 Tensor transformations

The magnetic diffusion gradient directions are specified using the physical coordinate system of the scanner. The z axis runs through the gantry while the x and y axes run orthogonal to it (see Figure 16.7; axes in yellow). Depending on the acquisition orientation, the origin of the gradient coordinate system and the image coordinate system (after it’s stored as a 4D image) may not coincide. As a result, the diffusion tensor, when displayed will look incorrect. This inconsistency may not be easy to spot, since it requires some knowledge of the anatomy being scanned.

Assuming the case of a transaxial acquisition, the gradient axes match the x, y, z axes of the image

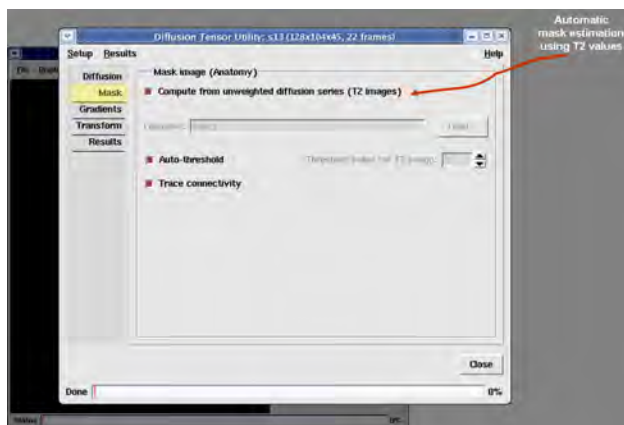


Figure 16.5: Loading a Mask Image

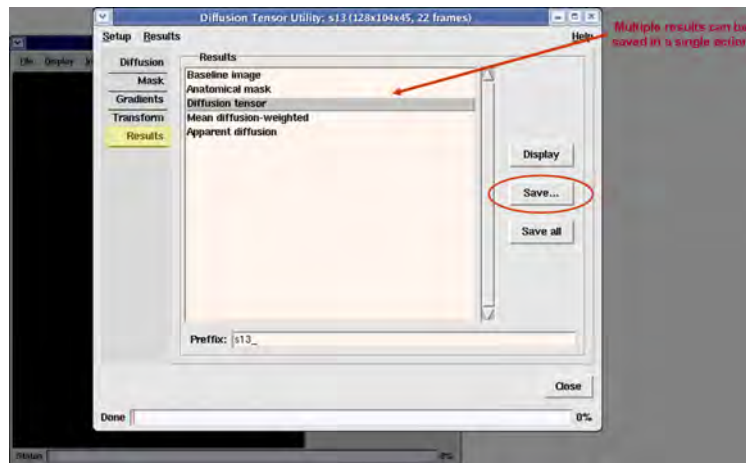


Figure 16.6: Computing the Tensor Image.

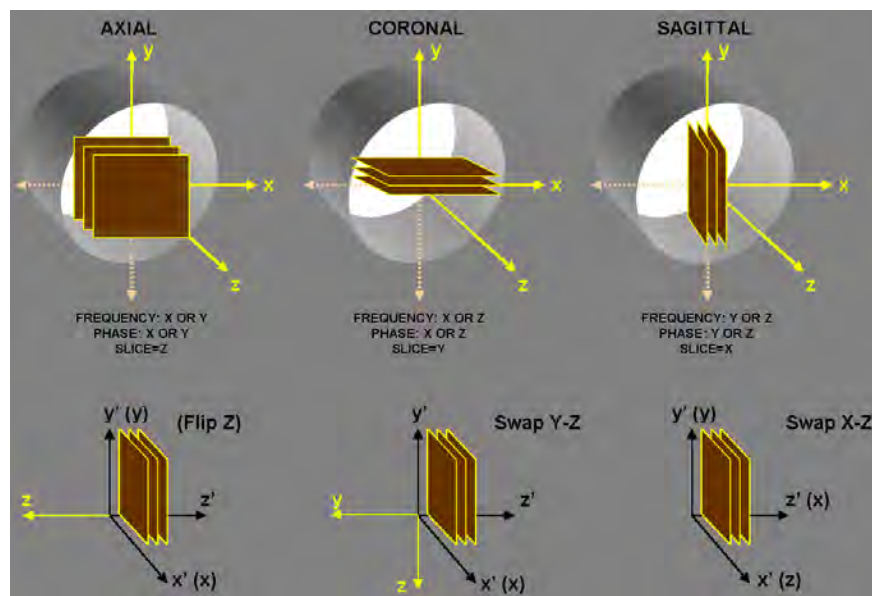


Figure 16.7: Tensor Orientations.

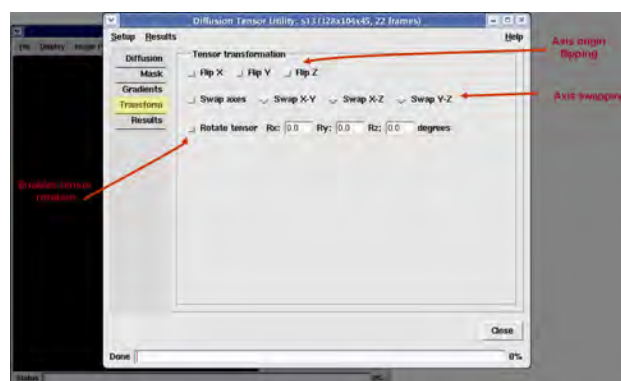


Figure 16.8: Tensor Transformations.

acquisition system (see axes in black, Figure 16.7). However, depending on the direction of the phase and frequency encoding gradients, the tensor may need to be flipped in sign. Also, if the image is stored last slice first, you will need to flip z. In the case of a coronal or sagittal acquisition, axes must be swapped (see Figure 16.8).

To perform such transformations, use the Transform pane, which will allow you to make the necessary flips and swaps. In addition, it will also allow the tensor field to be rotated, a step necessary when images are previously transformed by a rotation component. Once you select the necessary transformations, click on **Compute!** (in the Diffusion pane) to compute the diffusion tensor. If you previously computed the tensor, select the new transformations, compute it again, and then save the results.

If you are not sure about transformations to your image at this stage, proceed to the tensor analysis step. There you will be able to visualize the tensor field and determine if any transformation is necessary.

For further diffusion analysis, continue with Chapter 17 (Tensor Analysis) or Chapter 18 (Fiber Tracking).

## Chapter 17

# Diffusion Tensor Analysis

### 17.1 Introduction

The tensor analysis tool will take a previously computed diffusion tensor image (generated from the Tensor Utility), and create a series of diffusion-derived maps. It will also calculate region of interest (ROI) statistics of these maps, when an ROI mask is provided. Finally, the tensor analysis program also provides a set of visualization tools to depict tensors and vector fields. In order to invoke tensor analysis, choose the “Diffusion” menu, and select the “Tensor analysis” item.

### 17.2 Loading the diffusion tensor

To load the tensor in the tensor analysis program, click on **Load** in the Tensor pane, and select the tensor image. When loading is complete, the tensor image will be displayed in the main viewer. Note that because the tensor is symmetric, you will only see 6 components out of a total of 9 elements.

Normally, you will not need to change the ordering of the tensor components in the Components drop-down box. However, if the tensor image was created with third-party software, it may be possible that the ordering is different. Hence, in addition to the xx-xy-xz-yy-yz-zz ordering, xx-yy-zz-xy-xz-yz is also provided.

If you have saved an anatomical mask in the tensor utility tool, or have used the segmentation tools to create an ROI mask for analysis, you can load it in the Region of Interest mask. Simply disable the option Estimate from tensor, then click on **Load**. The program also allows you to have multiple regions of interest in the same mask. They have to be labeled consecutively from 1,2,...,n. After loading the multiple ROI mask, you should see the correct number of regions displayed in the option Number of regions in mask.

It is often the case that in the presence of noise and artifacts, a few tensors may yield negative

(likely very small magnitude) eigenvalues. Since this is not physically possible, you may choose to exclude these regions from the diffusion maps and statistics. You can enable that option by checking Mask out regions with negative eigenvalues. The default behavior is to include these regions, but to make their eigenvalues positive. Once the tensor image is loaded and, optionally, also the mask, click on **Compute!** to calculate the diffusion maps and generate statistics.

## 17.3 Results

Once the diffusion maps are computed, you will be taken to the Results pane, where a list of the results will be displayed. By default, the Fractional Anisotropy (FA) map is displayed in the main viewer. You may select different results and display them using the **Display** button.

Some of the result items are grayed out, which means they are not computed by default. To enable their calculation, double-click on them, or select the item and press “On/Off”. Then, go back to the Tensor pane, and click **Compute!** once more.

Note that diffusion maps are only shown at the regions of interest. At this moment, you may choose to save any of the computed results.

## 17.4 Statistics

Basic diffusion map statistics are calculated for each one of the regions in the ROI mask. The statistics are displayed in a report format, and can be saved as a text file. You can then import this file into statistical analysis programs or text editors.

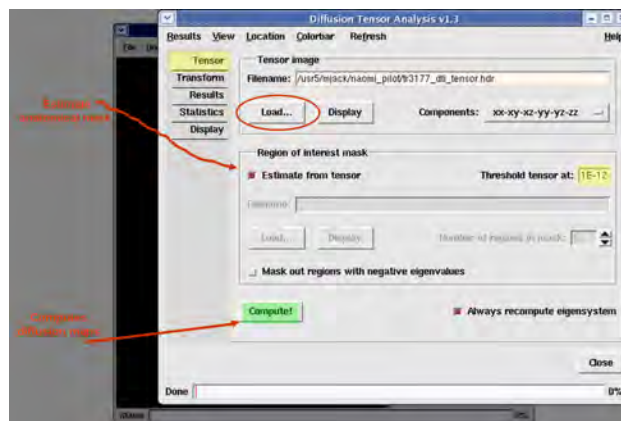


Figure 17.1: Loading The Diffusion Tensor.

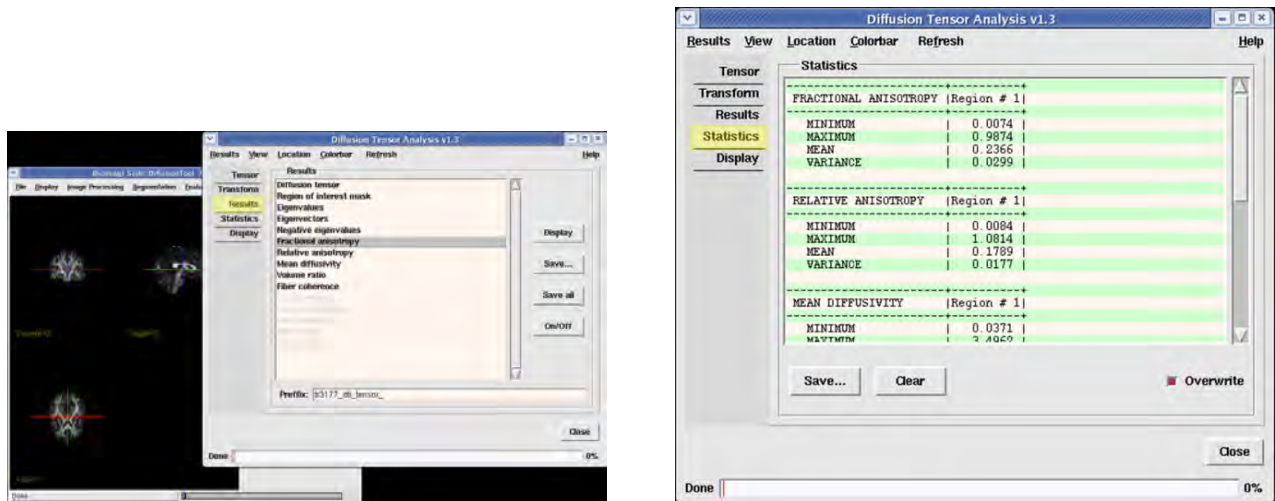


Figure 17.2: Left: The Results tab of the Tensor Analysis tool. Right: The Statistics tab of the Tensor Analysis tool.

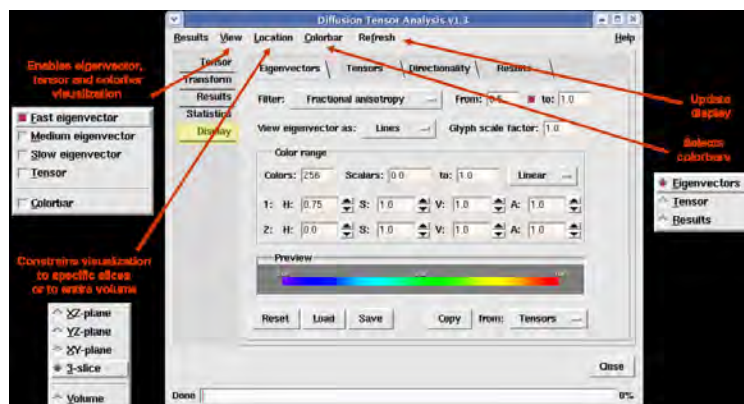


Figure 17.3: The Visualization tab of the Tensor Analysis Tool.

## 17.5 Visualization

The tensor analysis tool provides a series of visualization options, ranging from vector to tensor plots and the ability of managing their colormaps. See below for a summary of the display options.

To view vector fields and tensors, make sure the viewer is in 3D display mode.

### Plotting the principal eigenvector field (Fast Eigenvector)

In the Display pane, select the View menu, and enable the option “Fast eigenvector”. By default, the eigenvectors are displayed on the three orthogonal planes, however this can be changed in the Location menu. Make sure this matches what you are currently displaying. **Important:** as you move through the slices in the main viewer, you will need to click Refresh to replot the vector field on the newly selected slices.



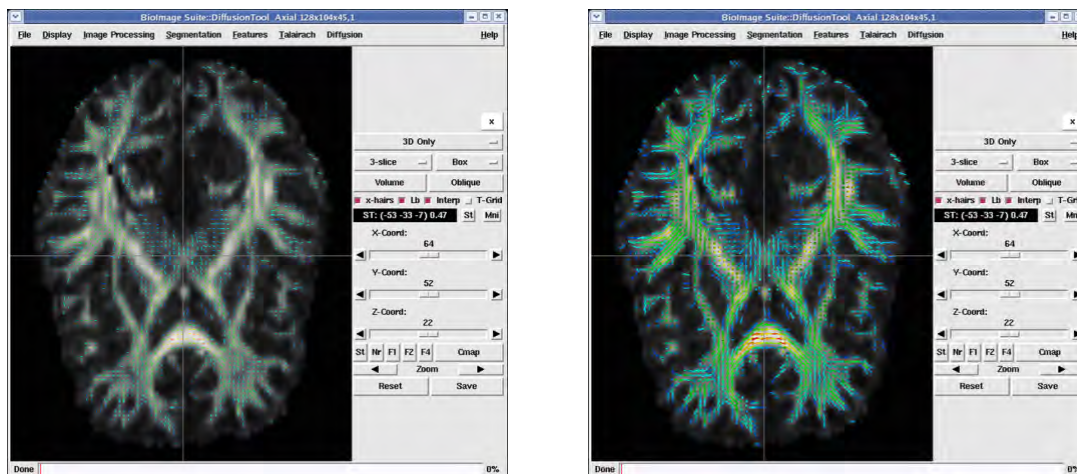


Figure 17.4: Left: Principal eigenvector displayed as lines. Right: Principal eigenvector displayed as tubes.

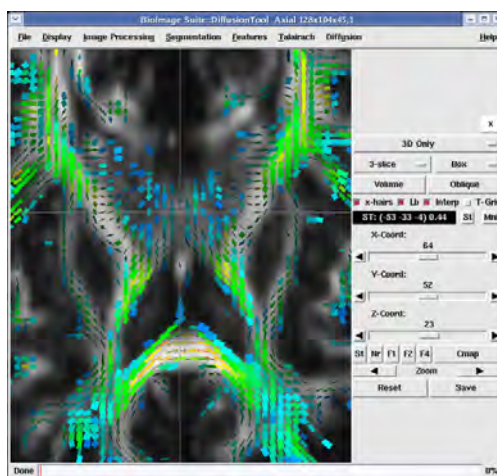


Figure 17.5: Tensors displayed as sheets (axis with smallest magnitude is discarded).

In the Eigenvectors tab, you can fine-tune the display of the eigenvector plot. You may select a diffusion map to filter the results and control the lower and upper bounds. Vectors will only be displayed if they are within the selected range of values. Eigenvectors can be viewed as Lines, Tubes, or Arrows, and can be custom-scaled (see Glyph scale factor option). Tensor analysis does not allow scaling vectors by their magnitude.

**Plotting the tensor field** Alternatively, you can also display the tensor ellipsoids by checking the option “Tensor“ in the View menu. The same type of filtering options are available for managing the tensor visualization, which is accessible by the Tensors tab. Tensors can be viewed as Ellipsoids, Cuboids or Sheets.

**Display the directionality map** The directionality map, also called the color tensor or directionally-encoded colormap, represents the principal eigenvector by means of a color representation. The

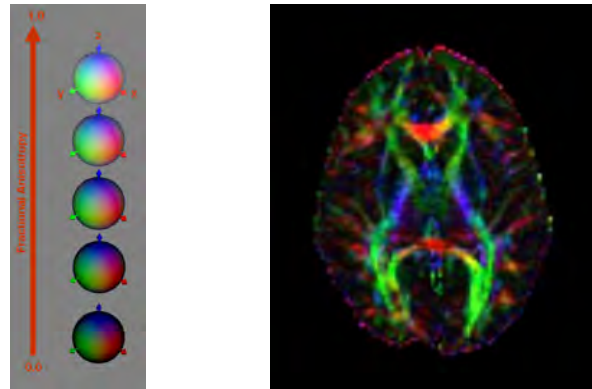


Figure 17.6: Transaxial cut of a directionality map.

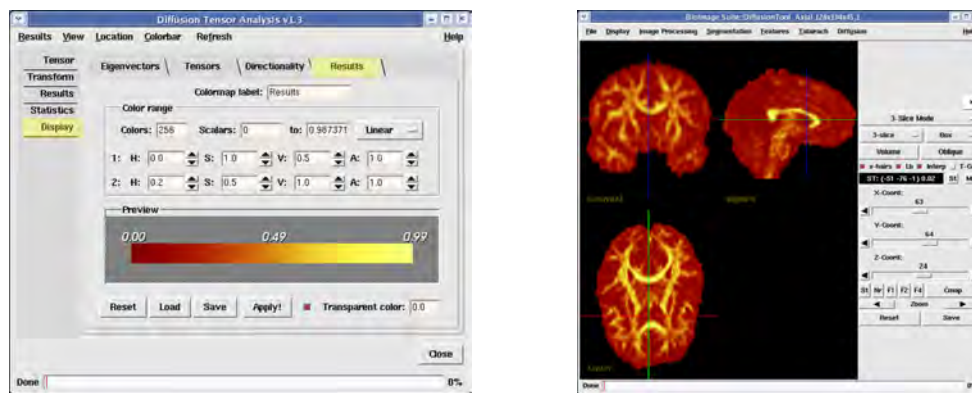


Figure 17.7: Coloring the Results.

most common scheme is to represent the x component as red, the y component as green and the z component as blue. Normally, the brightness is modulated by a second measure, such as fractional anisotropy.

To compute the directionality map, click on the “Directionality“ tab, select the intensity modulation method, the color scheme (absolute value is the default), and the number of colors for quantization. Then click the **Apply!** button. The resulting map will be shown in the main viewer.

**Coloring the results** The colormap editor is a straightforward way to color your diffusion images. You can define the number of colors to be used and the scalar range that it will represent. The resulting colormap will be a linear gradient between two colors, color 1: and 2:, both specified in HSVA (hue, saturation, value, alpha) coordinates. The alpha parameter corresponds to opacity.

You can apply a colormap to any of the diffusion maps that were calculated by the tensor analysis tool. First select the diffusion map to be displayed from the Results tab. Click on Display. Now in the Display tab, select Results and customize the colormap accordingly. Once you are ready, click on **Apply!**. Now the diffusion map should be displayed in the colormap you have selected. Below is an example of coloring the FA map with a hot color scheme.

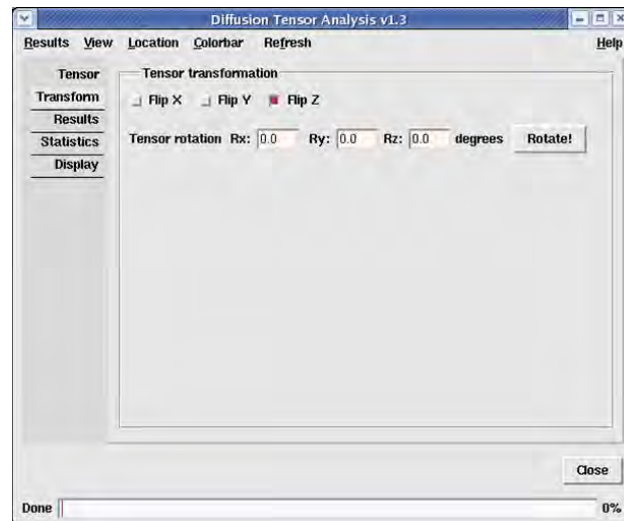


Figure 17.8: Applying Transformations.

## 17.6 Transformations

A subset of tensor transformations can be found in the Tensor Analysis tool. This is to allow convenient transformation in situations where the transformation is not known prior to the tensor analysis. Using the visualization tools described above, one can determine by looking at the orientation of the vector or tensor fields when a transformation is required.

To apply a transformation, choose the Transform tab and select the type of transformation. For flips, the result is instantaneous and will update any visualization glyphs present in the main viewer.

## Chapter 18

# Fiber Tracking

### 18.1 Introduction

The fiber tracking program allows you to reconstruct fiber pathways from the diffusion tensor information. This program uses the streamline technique to integrate fiber bundles through the vector field comprising the principal direction of diffusion. It handles seeding the integration at individual points, entire regions of interest (ROIs), or via the landmark editor in the main viewer. It allows for arbitrary seed density, and provides flexible stopping criteria, including maximum fiber angle, fractional anisotropy and mean diffusivity values. In addition, the program calculates basic fiber bundle statistics. In order to invoke the fiber tracking tool, simply choose “Diffusion” menu, and select “Fiber Tracking”. Below follows a sequence of steps explaining the use of the fiber tracking program.

### 18.2 Loading the input images

Under the **Input** tab, you will first need to load the tensor image, computed by the Tensor Utility tool (see Section 16), as shown below. Second, load either the anatomical mask you created earlier or the ROI you would like to use for tracking in the Region of interest mask. Finally, load the auxiliary image, which could be any image in DTI space, for which fiber statistics can also be computed, in the Map for analysis box. Once all these input images are loaded, make sure that you have an image in the viewer (e.g. mask, or map for analysis) before you start tracking.

### 18.3 Directionality

As above, the color scheme may be set using the **Directionality** tab.

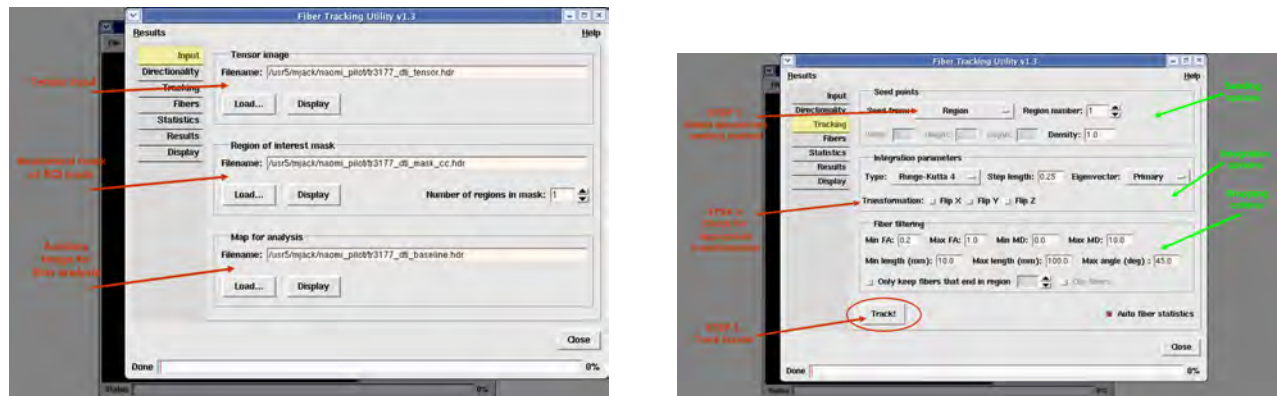


Figure 18.1: Left: Loading Input Images. Right: An Overview of the Fiber Tracking Process.

## 18.4 Tracking

Under the **Tracking** tab, you will find a series of options and parameters to control the fiber tracking procedure. To fiber track, you must 1) Select the appropriate seeding method; 2) Select parameters for tracking, such as transformation type, ranges for FA, or MD, and maximum angle; and finally 3) press **Track!**.

**Seed point selection:** You may choose to seed the tracking using a single point, a region of interest, a volume, or points from the landmark control. In the case of a single point, the position of the cross-hairs in the main viewer will be used as coordinates for the point. In the case of a region of interest, you must specify the region number to start tracking from. All points from the ROI will be used as seeds. You can also increase the point density. For volume, a 3D window (subvolume) of size Width  $\times$  Height  $\times$  Depth is taken to initialize the tracking.

**Integration parameters:** The integration method used by the Fiber Tracking program is the well known Runge-Kutta method. You may specify second-order integration or fourth-order integration, which provides better accuracy. The step length (h) for integration represents the percentage of a voxel diagonal's length. Higher percentages will yield coarser results while smaller values will yield smoother fibers.

By default, the eigenvector field to be followed is the Primary, also known as the principal direction of diffusion. Note that the integration is performed both ways from each seed, at  $+h$  and  $-h$ , and the final result will constitute a single fiber. The user is also allowed a set of simple transformations to the vector field when tracking: sign flips in the x, y and z directions. For more complex transformations, see Sections 16 and 17.

**Fiber filtering:** Here you will see a series of stopping criteria options, ranging from minimum and maximum fractional anisotropy (FA), mean diffusivity (MD), fiber length, and maximum angle between steps. In addition, you may choose to track only fibers that cross a specific region. For

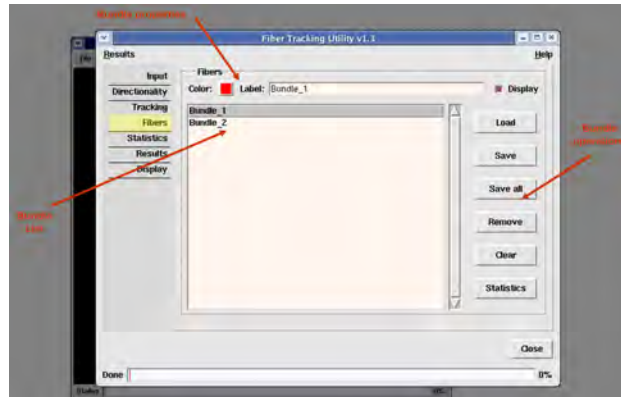


Figure 18.2: Fiber Bundles.

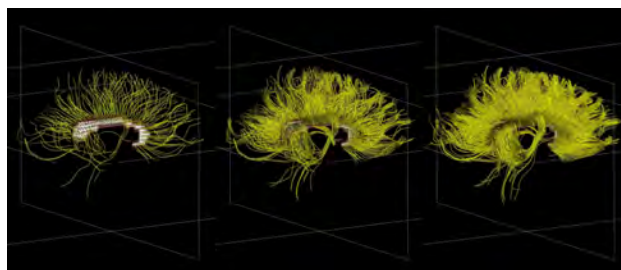


Figure 18.3: Examples of fiber tracking by seeding a region of region of interest in the corpus callosum with increasing fiber densities.

that, enable the aforementioned option and choose the corresponding region number. By enabling Clip fibers, only fiber segments that end in the specified region are kept.

Once the relevant parameters are set, simply press the **Track!** button to start tracing.

## 18.5 Fiber bundles

Once tracking is complete, you are taken to the **Fibers** tab, in which the tracing results will accumulate. Bundle names and colors can be changed: simply select a bundle from the list and either change the label or press the **color** button. They can also be made visible or hidden by clicking the **Display** checkbox.

You can save a bundle or a set of bundles by selecting them from the list and pressing the **Save** button. Note that you will be prompted for a directory name for saving. Bundles are saved according to their label names. To save a bundle with a different name, you must change its label first.

*Important:* Fiber bundles are not immediately visible in the viewer, until you switch the viewer to 3D mode, **and** select “View fibers” under the Display tab, in the Fiber Tracking program (See Section 18.8).



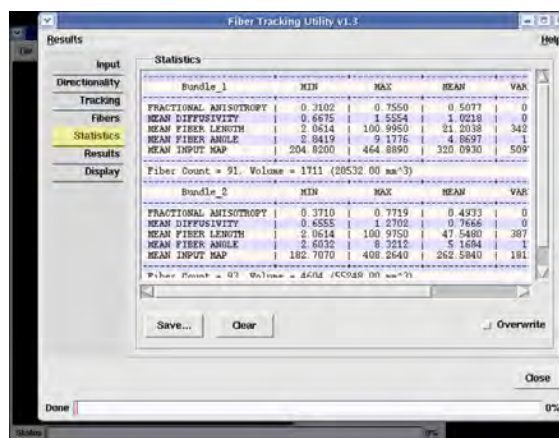


Figure 18.4: Bundle Statistics.

## 18.6 Statistics

Under the **Statistics** tab, you will find the basic statistics for every bundle that has been tracked. Both fractional anisotropy (FA) and mean diffusivity (MD) as well as fiber length and mean fiber angle measures are computed. In addition, statistics on the auxiliary map, or input map, are also provided. Press the **Save** button to save the report as a text (ASCII) file which can be imported by other software (spreadsheets, statistics, etc.).

## 18.7 Results

Under the **Results** tab, you will see a list of images generated by the tracking program. Currently, the only output from the tracking program is the fiber quantization mask, a binary mask representing the last tracked bundle in raster format. This mask can be saved as an Analyze image for use in BioImage Suite or other programs.

## 18.8 Display

As in the Tensor Analysis tool, under the **Display** tab you will find a set of visualization tools. Here you will be able to view the fibers, as well as color them according to various measures (FA, MD, etc.) and change their representation.

**View fibers:** As mentioned earlier, fiber bundles will not be visible in the viewer until you switch the viewer to 3D only and then check the option View fibers to display them.

**View colormap:** If you would like to display the colormap in the main viewer, select this option.



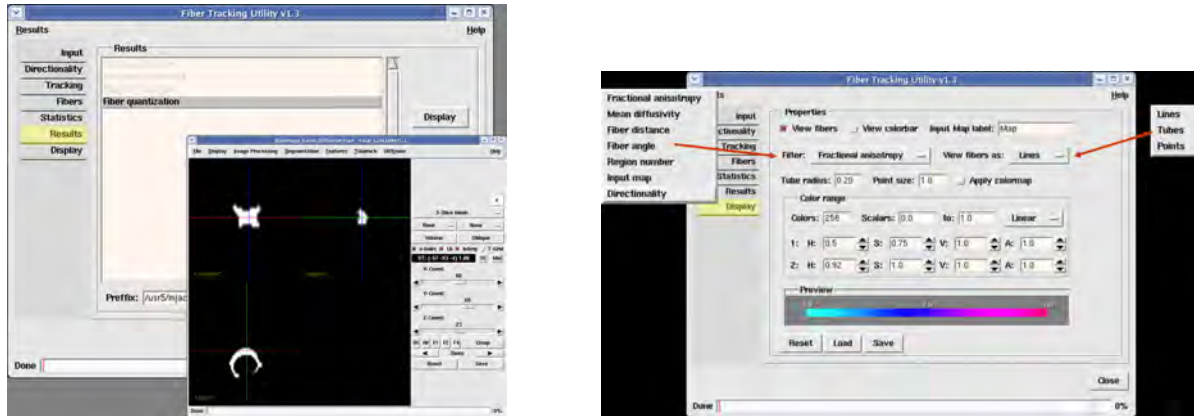


Figure 18.5: Left: Example of fiber quantization result, delineation of the corpus callosum. Right: Fiber Display Options.

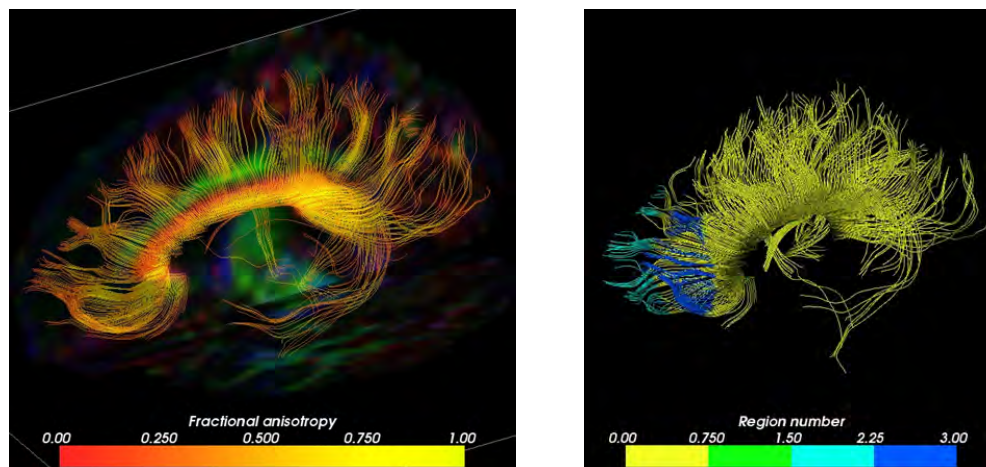


Figure 18.6: Examples of fiber tracings color according to fractional anisotropy (left) and region number (right).

**Filter:** Fibers are assigned to a solid color by default. Alternatively, you may choose to color them according to different indices, such as FA or MD. The Filter option lets you specify which measure should be used to color the fibers. To enable this option, check the Apply colormap option. Fibers can also be displayed as a set of points, lines or tubes. The tube radius and point size can also be selected.

## Part VI

# F. Neurosurgery Tools

## Chapter 19

# Intracranial Electrode Localization

### 19.1 Introduction

BioImage Suite provides a tool for localizing and labeling electrodes based on an image that contains visible electrodes, such as a CT image. This tool is the Electrode Editor, which consists of two windows: An Image Viewer (Fig. 19.2) and an Electrode Control Window (Fig. 19.3). With these tools, you can create a custom array of electrodes and place them into an image in any configuration you like. Later, you can visualize these electrodes in other images, and load attributes for the electrodes with the Data Tree Manager (see Chapter 15). The generation of electrode location data results in the creation of a \*.mgrid file, which holds the electrode properties and locations.

**Starting the Electrode Editor** The Electrode Editor is available either through the main BioImage Suite menu, or by typing `pxitlectrodeeditor` on the command line.

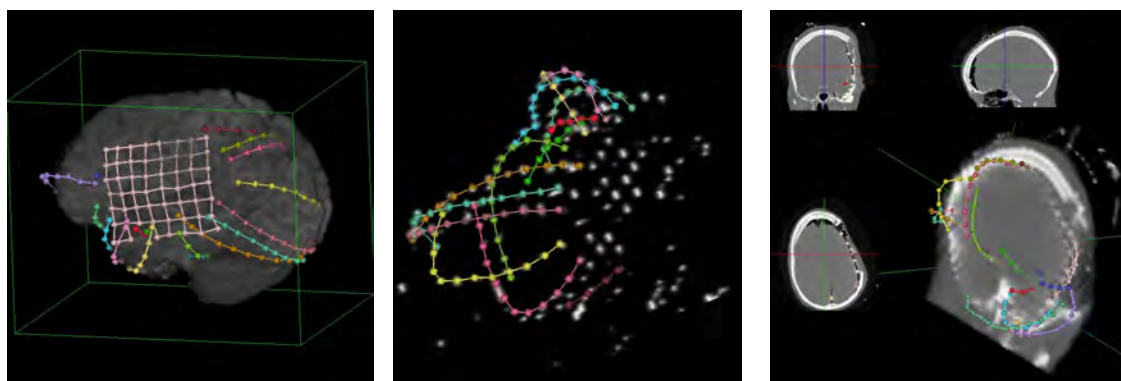


Figure 19.1: Different Views of the Electrode Editor in Action. **Left:** intracranial electrodes overlaid on an MRI Brain Image. **Middle:** Electrodes overlaid on a thresholded volume rendering of a CT image. The bright dots in the CT are the metal contacts of the electrodes. **Right:** A more complex rendering of the electrodes overlaid on a CT image.

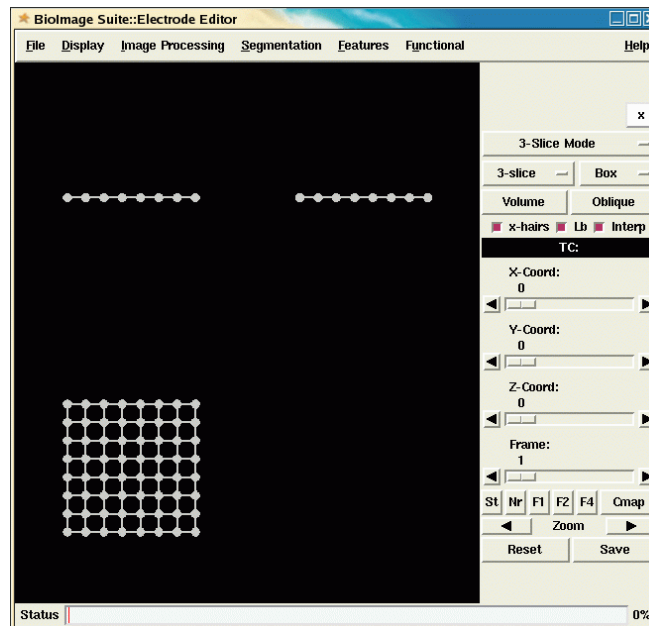


Figure 19.2: The Electrode Editor Image Viewer. This is a normal viewer in most respects, and operates just like the 3D viewer described in Viewers. The only difference is that it has electrodes displayed in it, which are linked to the Electrode Control Window.

**Note:** Much of the effort involved in localizing electrodes is based on the manipulating the image and the Viewer to better view the positions of the electrodes. Look at The Viewers (Chapter 5), and Image Processing (Chapter 7), for a more in-depth descriptions of these functions.

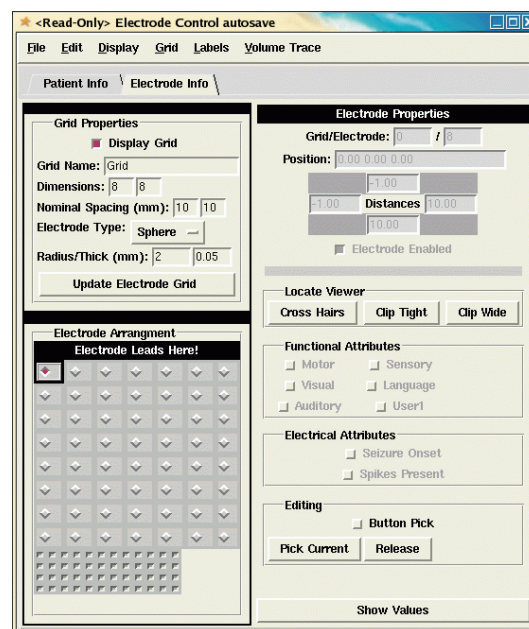


Figure 19.3: The Electrode Editor Image Viewer. This window is where electrodes are added and their properties are set. For a description of all the functions in this window, see the Electrode Editor Function Catalog, below.

## 19.2 Getting Started

(1) **Load the base image:** Once the Electrode Editor Tool has started, you need to load an image into the viewer. Just as with all viewers in BioImage Suite, choose the ( File — Load ) menu option and open your image (or import a non-analyze format file with ( File — Import ). This image must contain recognizable (significantly bright) locations of electrodes, since these will help the software position an electrode exactly as you click in their vicinity.

(2) **Set Editing Options:** A few options must be set for you to start editing and localizing electrodes. **In Image Viewer Window:** Toggle the ( Features — Enable Volume Trace ) option to ON. **In Electrode Control Window:** Choose the ( Edit — Full Edit Mode ) radio option, in order to allow you to edit all aspects of your electrodes. On the right side of the “Electrode Info tab” in the Electrode Control Window, check the “Button Pick” option.

## 19.3 Working with Electrode Grids

In the Electrode Control Window, the template electrodes start as (  $X \times Y$  ) grids. In practice, many electrode sets are actually arranged as strips of electrodes in a row. In this case, simply change the template electrode to a grid with dimensions (  $1 \times Y$  ). **Note:** In this manual, the term “grid” refers to both this strip configuration (  $1 \times Y$  ) and the grid configuration (  $X \times Y$  ).

### Adding Grids and Setting Their Properties

**Add/Delete Grids:** Electrode grids are added and their properties are manipulated in the Electrode Control Window. To add a new grid, or remove an unneeded grid, look in the “Patient Info” Tab. On the right side of the window are the **Add New Grid** and **Delete Grid** buttons. When added, a new electrode grid is placed in the “Grid Information” list.

To set an electrode grid’s properties (length, width, name, spacing, etc.), select it in the “Grid Information” list, and click on the “Electrode Info” tab. This tab now contains options for this individual grid. By default, newly added grids are given an X dimension of 1, making them strips.

**Grid Properties:** In this section of the “Electrode Info” tab, you can set the name of the grid (often the name includes some anatomical landmark). **Note:** Names should not contain spaces. Dimensions are set here as well. Nominal spacing sets the assumed spacing between neighboring electrode on the grid in both dimensions. You can choose what type of icon you would like to be used to display the electrode location (either sphere or disc), as well as the radius and thickness thereof. Once you have set these options correctly for the electrode grid in question, click the **Update Electrode Grid** button. After querying you to be sure you want to make the changes, the electrodes in the Viewer, as well as in the “Electrode Arrangement” box will update to show the appropriate dimensions.

**Electrode Arrangement:** In this box, you are able to select individual electrodes within a grid. Note that electrodes are numbered *from the end furthest away from the leads to the end closest to the leads*. **A Note on Electrode Numbers.** Electrodes are numbered from the end furthest away from the leads to the end closest to the leads. Thus, in a strip of electrode, electrode 1 is at the bottom of the Electrode Arrangement display, and the highest numbered electrode (whose number depends on the length of the strip) is at the top. If the electrode has been cut, so that the end distal from the leads is missing, you should still input the original length of the electrode strip in the “Dimensions” field, and then select the electrodes that are missing in the arrangement view, and uncheck the “Electrode Enabled” box on the right side of the tab window. The electrode will become grey in the dialog box and disappear in the viewer. This is important because it allows you to create shorter electrode strips that are numbered starting with numbers other than 1. (If you just create a shorter electrode, it will be numbered 1-6, for example, not 3-8). Thus, you can maintain consistency with the electrodes’ real numbering scheme.

### 19.3.1 Localizing Electrodes

**Note:** If you are using a volume visualization in the viewer (see the page on Viewers), before localizing electrodes you should go to the Features menu in the viewer window and make sure that the “Enable Volume Trace” checkbox is checked. This allows the viewer to interpret mouse clicks as being localizations on the volume as you see it.

Once all the electrodes have been added in the “Patient Info” tab, and their names, dimensions, and spacings specified, you can start placing them in their proper locations in the Viewer Window. The base image should already be loaded in the Viewer (see Chapter 3). The next step is to make the electrode locations in the image readily visible by thresholding it. We have had success thresholding CT images of platinum electrodes from 3000 to maximum, and steel electrodes from 1500 to maximum, if they are present. See Chapter 7 for how to do the thresholding operation. Leaving the Viewer in the Results view is fine in this case.

Once the loci of the electrodes are readily visually apparent in the base image, you can start selecting and localizing electrodes to their proper locations (Sometimes it may be helpful to use the volume control to crop the viewable image somewhat in order to temporarily remove confounding electrodes from the view).

Find and identify a specific electrode in the image. Then go back to the Electrode Control window (See Fig. 19.3), choose the “Patient Info” tab, and select the electrode grid that contains the electrode you have picked. Click on the “Electrode Info” tab, and in the “Electrode Arrangement” tab, click on the button in the grid that corresponds with this electrode. In the “Editing” Box, click the **Pick Current** button. Now go back to the Viewer, and *Hold down Shift while left-clicking* on the bright area in the image that represents the electrode. The marker for that electrode will move to where you clicked. Go back to the Electrode Control Window, select another electrode, click **Pick Current** and repeat.

**Volume Trace:** The electrode is placed in the center of the bright region that is close to the location that you Shift-click on. The volume of the space searched is under your control. To set

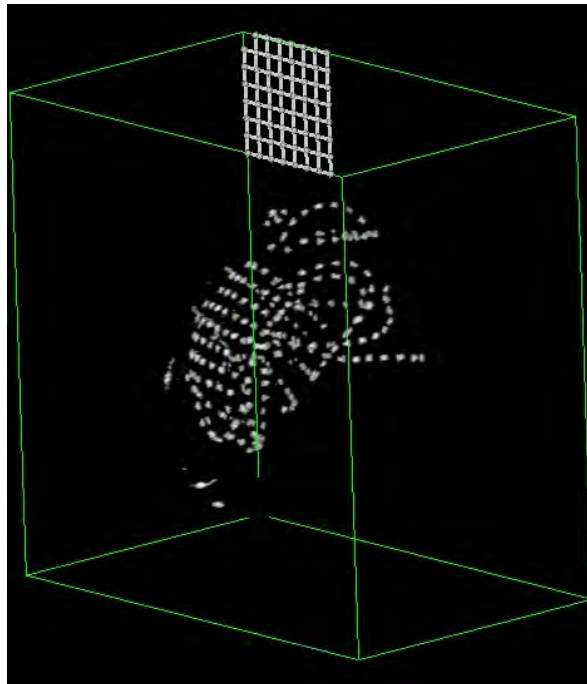


Figure 19.4: An example thresholded CT image. This image was thresholded at 3000 to maximum, and shows very clearly the locations of all the surface and depth electrodes that have been implanted. At the top of the image, you can see the electrode grid at its default location, ready to be localized based on the image.

it, simply go to the Volume Trace menu, and select one of the “VOI Size” options. The default is a radius of 10. If two electrodes are close together in the image, it may be useful to decrease the VOI size, so that they will not be localized together by a single click.

**Undo:** In the Edit menu there is an “Undo Move Electrodes” option. This can undo the last electrode placement operation performed. This operation is not cumulative, however.

### Grid Color

Once all electrodes in a grid have been placed or while you are working on them, you may wish to color the grid, so that it can be easily spotted and differentiated from the others. To do this, use the ( Grid — Grid Color ) menu option. A Color selection dialog box will appear, in which you can choose a unique color for the grid.

### Saving the whole scheme

The ( File — Save ) operation saves an \*.mgrid file that contains all the information you have entered: electrode grid dimensions and locations, names, numbers, color schemes, etc. This file can





Figure 19.5: Electrode Arrangement Box. Electrodes are numbered from the end farthest away from the leads to the end closest to the leads. Thus, in a strip of electrode, electrode 1 is at the bottom of the Electrode Arrangement display, and the highest numbered electrode is at the top.

then be edited later or loaded into the electrode control modules of other viewer sets. These \*.mgrid files are also able to be loaded into the Data Tree Manager, so that you can apply transformations to them and display the electrodes in the spaces of other images (on the pre-operative MRI, for example).

## 19.4 Electrode Editor Function Catalog

The functions in the viewer of the Electrode Editor program are common to all viewers, and are detailed in Chapter 3. This section explains the tools that are available in the Electrode Control Window (Figure 19.3)

### 19.4.1 Patient Info Tab

The Patient Info Tab contains a summary of which electrode grids are present in the current scheme, and what their numbers and names are. As described above, in Adding Electrodes, this is where you add new electrode grids and delete unneeded ones. Using the **Add New Grid** button will add a new grid at the bottom of the list, with a default name, which you can change in the Electrode Info Tab (See below). Clicking the **Delete Grid** button deletes the currently selected grid.

The Patient info tab also contains a couple fields for some info about the patient: a simple description, and a comment, in case you need them.

### 19.4.2 Electrode Info Tab

This tab allows the users to set various properties of the electrode grid.

**Grid Properties:** The options in this section set the properties of each grid as a whole.

**Grid Name** The name of the grid or electrode strip

**Grid Dimensions** The length and width (in number of electrodes) of the grid. If you need to represent an electrode strip, simply use a 1 in the first field.

**Nominal Spacing** The initial spacing between electrodes in the grid. This is flexible, as spacing will change for each electrode as you place it, but it should be close to the actual spacing on the electrode strips before implanting (Usually 10mm or 5mm).

**Electrode Type** This menu contains two options for how the electrodes should be drawn: either as spheres or as discs.

**Radius/Thick (mm)** This pair of options specifies the size of the rendered electrode representations. Just the radius is needed for spheres, while discs are drawn according to the radius size and a thickness as well. When you have made the changes you want, click the

**Update Electrode Grid** button. This will remove any placements you have made on the old electrode grid, and replace it with a new grid that has the properties you have specified.

**Electrode Arrangement:** This section of the Electrode Info Tab is just a grid of radio buttons that correspond with the electrodes that are available for you to edit and place. This is where you select an individual electrode to place, or enable/disable. See Localizing Electrodes.

### 19.4.3 Electrode Properties

The values in this section update as you select electrodes in the Electrode Arrangement section. The top field, “Grid/Electrode” specifies which electrode number in which grid you are working with. The “Position” field gives the X, Y, and Z coordinates of the electrodes location in the image. The four “Distances” fields show the distance from the electrode in question to each of its neighbors (If the electrode is a strip, the top and left fields will have the same value, as will the bottom and right pair of fields). The “Electrode Enabled” checkbox lets you disable an electrode that is not in use (see A Note About Electrode Numbers).

**Locate Viewers:** The buttons in this section let you quickly send the crosshairs or the view window in the viewer to the electrode selected in the Electrode Arrangement section.

**Attributes:** The Functional Attributes and Electrical Attributes sections are simply groups of pre-set checkboxes that you can use to mark individual electrodes as having certain attributes. If you need to apply a more complex set of attributes, use the electrode attributes function in the Data Tree Manager.

#### 19.4.4 File Menu

The file menu provides features for loading, saving, and exporting your data.

**Load** The electrode editor loads \*.mgrid files, which contain all information about an array of electrodes, including locations (based on the base image, which should be loaded into the viewer as well for visualization purposes).

**Save** Save an \*.mgrid file.

**Export** You can export a delimited text file that specifies all the values for the electrode array, you can export the electrode array as a surface file (see Chapter 12), or you can generate a new binary image file out of the electrodes, with a single pixel with non-zero value for each electrode. Finally, the “Export Picked Function” option saves the intensity value of the image under the electrodes. When electrodes are overlaid on a function image, this is useful in determining if an electrode is on a certain function area.

**Close** Exit the electrode editor.

#### 19.4.5 Edit Menu

The Edit menu changes what feature of a grid can be edited. This can prevent accidental changes to the grid. You can choose which sets of parameters you want to be able to change during editing by selecting the appropriate editing mode in the Edit menu.

**Display Only Mode** Disables all editing of electrode locations and attributes. Only for visualization and export.

**Only Edit Attributes** Disallows placement of electrodes, but allows you to set attributes and properties of individual electrodes.

**Full Edit Mode** All electrode locations and properties may be set and manipulated.

**Auto-Save:** Automatically saves your work periodically.

**Undo Move Electrode:** Moves the most recently placed electrode back to its previous location. This only works once - it is not cumulative, so you cannot undo multiple placements.

### 19.4.6 Display Menu

You may not wish to display all electrodes at once, even though they all exist in the file. Thus, there are a few options in the Display menu that allow you to choose which electrodes should be shown.

**Show Current:** - Only show the electrode grid selected in the “Grid Information” section.

**Show All** - Show all electrode grids.

**Show None** - Make all grids invisible.

**Show Some** - Bring up “Select Grids to Display” box. A simple dialog appears whenever you select the ( Display — Show Some ) menu option. It contains a list of all electrode grids in the file (from the “Grid Information” section of the Patient Info tab). As you select or deselect grids in the list, they will be displayed or made invisible. You can also choose all or none from here.

**Show Grid Lines:** This checkbox menu option allows you to turn the drawing of grid lines between the electrodes on and off.

**Pick and Display Image Function:** This menu option recolors the electrodes based on the color of the voxel on which each electrode is centered. This is useful when the electrodes are shown on a functional image, since different colored functional regions will yield different colored electrodes, allowing you to quickly see which electrodes are sitting in regions of which function.

**Show Attributes:** Brings up a dialog box similar to the “Select Grids to Display” dialog. In this case, the list is populated with the attribute categories that have been set for the electrode grids present. Clicking on one will highlight the electrodes that have that attribute. Note that this method of visualizing lets you only view one attribute at a time.

### 19.4.7 Grid Menu

The Grid Menu provides features for various grid manipulations.

**Grid Color:** This brings up a color selection dialog box that lets you choose the color of the current grid.

**Grid Flip X:** Performs a row flip on the selected electrode grid.

**Grid Flip Y:** Performs a column flip on the selected electrode grid.

**Grid Transpose:** Transposes the electrode grid, switching rows for columns.

**Transform Grid:** Brings up a dialog box in which you can load transformation files, warping the grid into a different space. This functionality has been deprecated by the Data Tree Tool.

**Auto Warp:** If the corners of a grid (ends for a strip electrode) have been placed, this function does a thin-spline transformation, interpolating the locations of the rest of the electrodes in the grid. This is helpful in that it brings the grid approximately into position, allowing you to make further refinements without confounding electrode connections to faraway unplaced electrodes. (In general Auto Warp automatically detects which electrodes have been moved from their default positions and then uses these to predict the location of the rest of the electrodes.)

**Undo Auto Warp:** Undoes the Auto Warp placing.

### 19.4.8 Labels Menu

The Labels Menu allows the user to show the numbers of the electrodes.

**Font:** The top section of the Font menu contains a range of options for the font size of the labels that are displayed next to the electrodes. The default is zero, and this is equivalent to turning labeling off.

**Skipping Labels:** If the display looks too crowded with all labels, choose one of the options in the skipping section of the Font menu, which will only label every *n*th electrode in a grid.

**Labels Color:** This menu option brings up a dialog box in which you can select the color in which to display the labels.

### 19.4.9 Volume Trace Menu

This menu contains a range of “VOI Size” options, from 0 to 25. These set the size of the volume searched for voxels above a threshold which define the region whose centroid is the actual location that the electrode gets placed on. So, when you click on a location in the image, a volume this size centered on this spot is searched for voxels that are above the threshold. The voxels that remain

then define a volume whose centroid is calculated. The electrode's final location is on this centroid. Thus, a VOI Size of zero causes electrodes to be placed exactly where you click. Larger sizes create a greater "snap-to" effect.

## Chapter 20

# The VVLink Tool

### 20.1 Introduction

In collaboration with BrainLAB AG we have performed work to integrate research image analysis methods and software with a commercial image guided surgery navigation system (the BrainLAB VectorVision Cranial System.) This work was described in a recent conference paper [83] – see

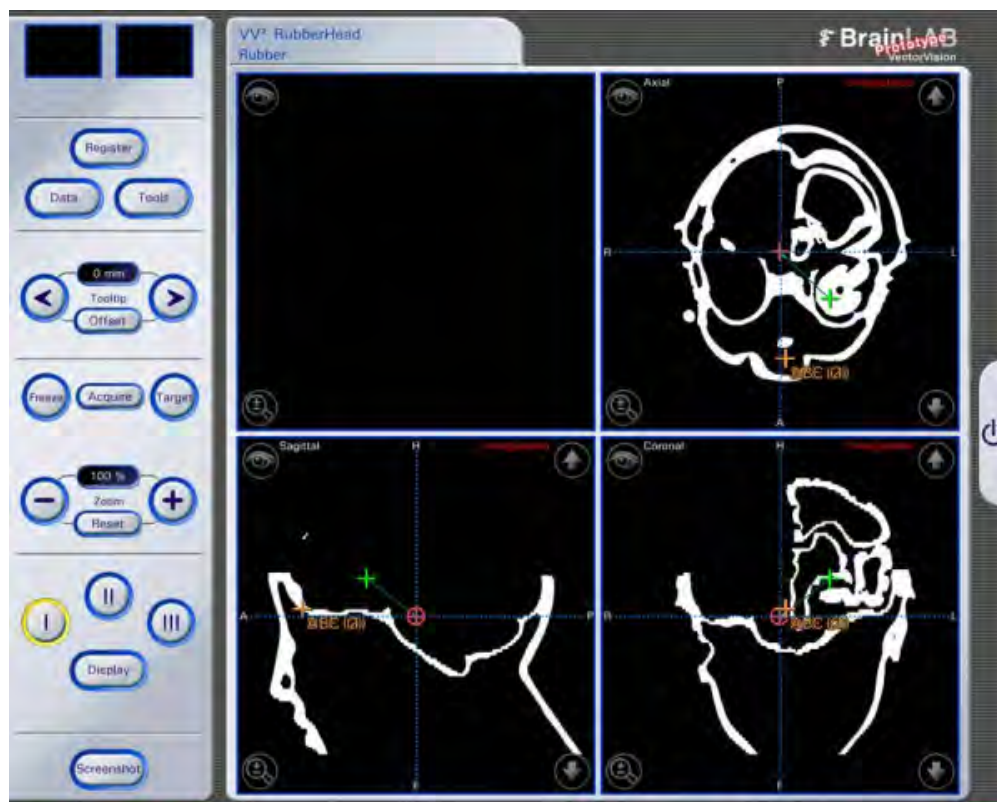


Figure 20.1: The BrainLAB VectorVision Cranial Image-Guided Surgery Application.



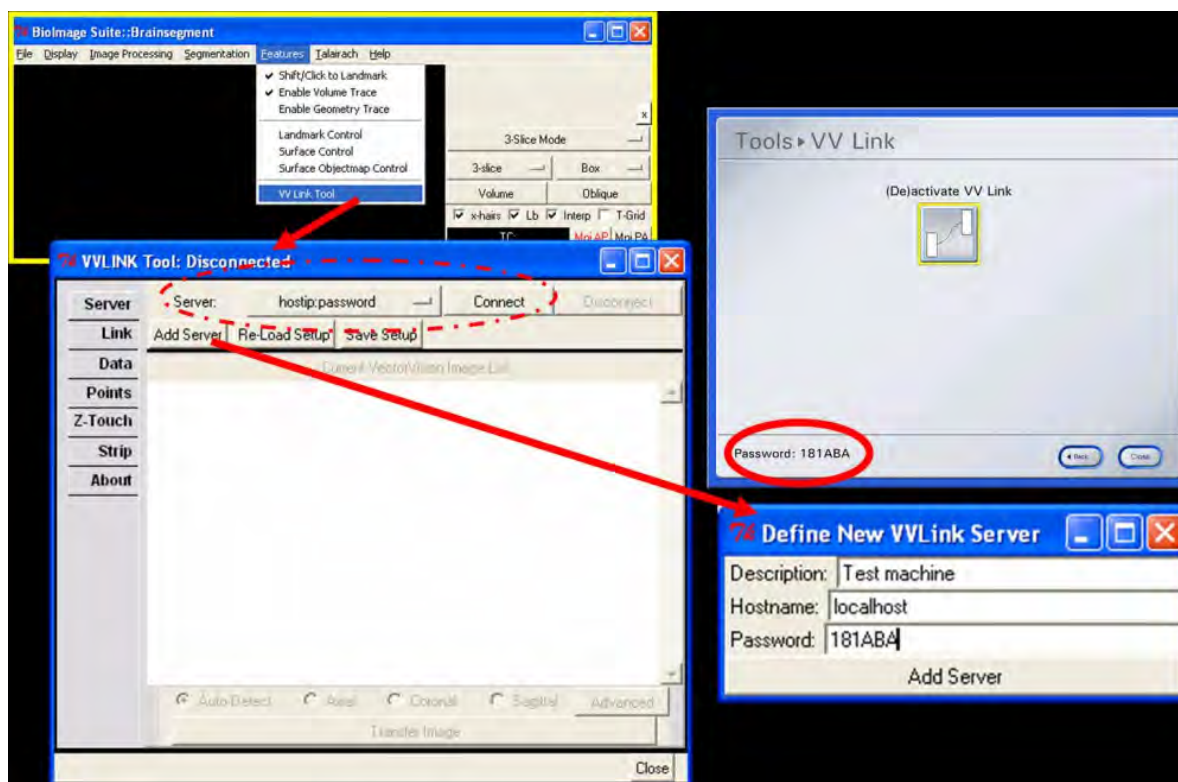


Figure 20.2: Accessing the BioImage Suite VVLink Tool (left) and defining VVC Servers (right).

also Markus Neff’s Master Thesis [68]. The integration was achieved using a custom designed client/server architecture termed VectorVision Link (VV Link) which extends functionality from the Visualization Toolkit. VV Link enables bi-directional data transfer such as image data sets, visualizations and tool positions in real time. This page describes the VVLink Tool of BioImage Suite which is the module that handles the communication with the VectorVision Cranial (VVC) system.

**Note:** The snapshots in this document were taken from a prototype version of VVC that was made available us during the design phase of VVLink. The released version of BioImage Suite (Windows and Linux platforms only) is, however, designed to communicate with the commercially available newly released VVC 7.8 – the actual user interface for VVC 7.8 is slightly different.

## 20.2 The BioImage Suite VVLink Interface

If available and enabled (using the Preferences Editor under Advanced/Miscellaneous), the VVLink tool is accessible under the Features menu as shown in Figure 20.2.

The first step in using the VVLink tool is to connect to a VVC host machine (typically an Image-Guided surgery system). Each host is defined by three values, namely a description, the hostname (or IP address), and its password. Server configurations are saved in a file called “.vvlinkservers” in the user’s home directory, and are displayed in the drop-menu titled “server” in the “Server”-tab

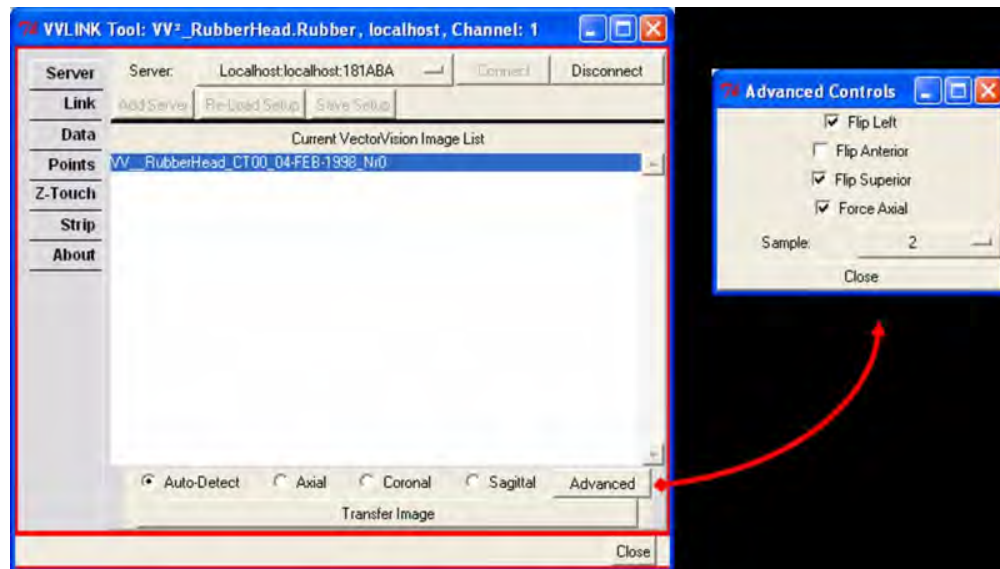


Figure 20.3: Transferring Images from the VVC system to the BioImage Suite VVLink Tool.

of the VVLink Tool. This tab is highlighted using a dotted ellipse in Figure 20.2 (left, bottom).

If no servers have been defined then click on the **Add New Server** button. This action brings up the dialog box shown in Figure 20.2 (right, bottom). The server password is obtained from the VVC system under Tools->VVLink as shown in the snapshot in Figure 20.2 (right, top). The Server configuration may be loaded and saved using the **Re-Load Setup** and **Save Setup** buttons respectively.

### 20.3 Connecting and Transferring Images

Once the server is defined, the next step is to connect to it. This is accomplished using the **Connect** button, next to the “Server” drop-menu. If the connection is successful, the VVLink Tool will download a list of available images in the VVC System as shown in Figure 20.3. Also the VVLink Tool will be redrawn with a red frame as opposed to a black frame. (Compare Figures 20.2 and 20.3).

**Note:** VV Cranial needs to (a) be configured to allow connections from the specific machine as it uses IP address filtering (change the variable AllowedSubnets in the configuration file vvcranial.ini, and (b) to enable the VVLink Interface. Consult the VV Cranial documentation for more details. In this particular example (Figure 20.3), there is only one image in the system, although more may appear. To enable proper communication between VVCranial and BioImage Suite, the user **must next transfer an image**. Transferring an image establishes the coordinate transformations between the two systems. To transfer an image, first select it in the list titled “Current VectorVision Image List” and then transfer it using the **Transfer** button. Most images transfer successfully using the “Auto-detect” settings. If the orientation (axial, coronal, sagittal) is wrong, the correct orientation may be manually set. Should the image need to be flipped about any of coordinate axis (**use this with care!**) use the **Advanced** button to bring up the advanced controls (Figure 20.3

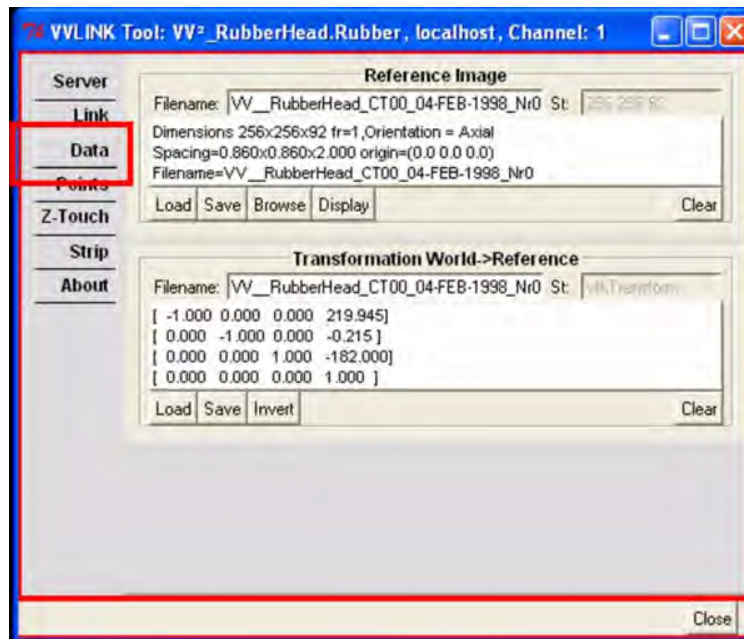


Figure 20.4: Saving Images and Transformations obtained from the VVC system.

right) and set parameters appropriately.

## 20.4 The Data “Tab”, Saving Images and Transformations

If the image transfer operation was successful, the information transferred (the image and its associated transformation to the world coordinate system) are placed in two controls on the “Data”-tab shown in Figure 20.4. The image is also displayed in the viewer associated with the VVLink Tool.

Both the image and the transformation can be manipulated (Load/Save etc.) using the options in their respective controls.

The transformation is used to map points in the world coordinate system (i.e. the operating room coordinate system) defined by the BrainLAB VVC system following patient to image registration to the image coordinate system, so that tools tracked in the VVC system can be appropriately displayed in BioImage Suite.

## 20.5 Real Time Communication

There are two aspects to the real-time communication functionality made available in the “Link”-tab (see Figure 20.5.), (i) streaming visualizations and (ii) linking the viewer cursor with the surgical navigation pointer.

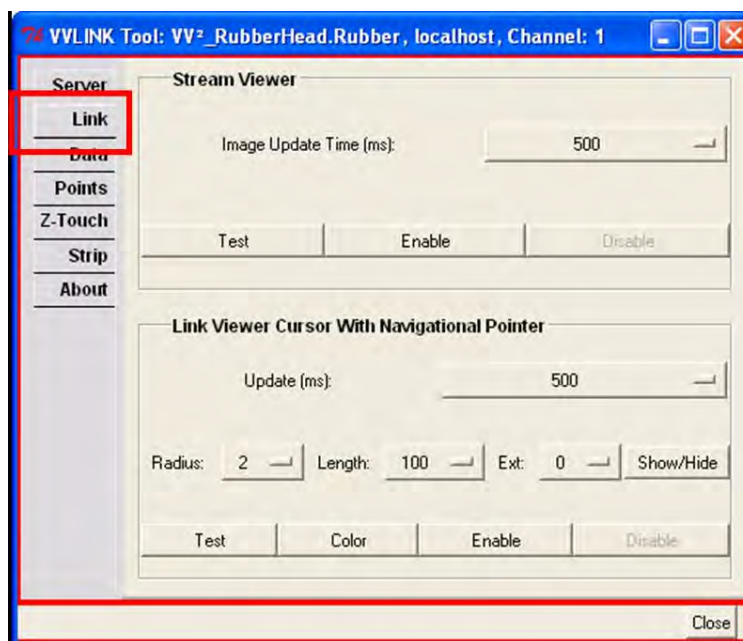


Figure 20.5: The “Link”-tab in VVLINK Tool, which facilitates both streaming visualizations and linking the BioImage Suite viewer cursor (cross-hairs) to the surgical navigation pointer.

**Streaming Visualizations** VV Cranial has the ability to take as an input a stream of bitmaps and display them within the VVC interface. (See the paper for examples) We have implemented functionality in BioImage Suite to capture the current viewer display and make it available to VV Cranial as such a bitmap stream. The user interface for this is located in the top half of the “Link”-tab.

The connection can be tested using the **Test** button and then, if desired, enabled using the **Enable** button. Once enabled BioImage Suite will send a stream of snapshots from the attached viewer at a rate determined by the setting of the **Input Update Time** drop-menu. This can be stopped using the **Disable** button. Please note, that streaming visualizations requires that the client machine (i.e. the computer running BioImage Suite) is computationally powerful enough to handle this operation.

The streaming visualizations are accessible in VV Cranial under the “Info View” tab of each VV Cranial Viewer.

## 20.6 Obtaining and Transferring the position of Tools and Landmark points

The VVLINK Tool can extract “labeled point” positions and tracked tool positions from the VVC System using VVLINK. These can be acquired and stored using one of three methods:

- Landmarks – pressing either the **Get Labeled Points As Landmarks** button or the **Get**

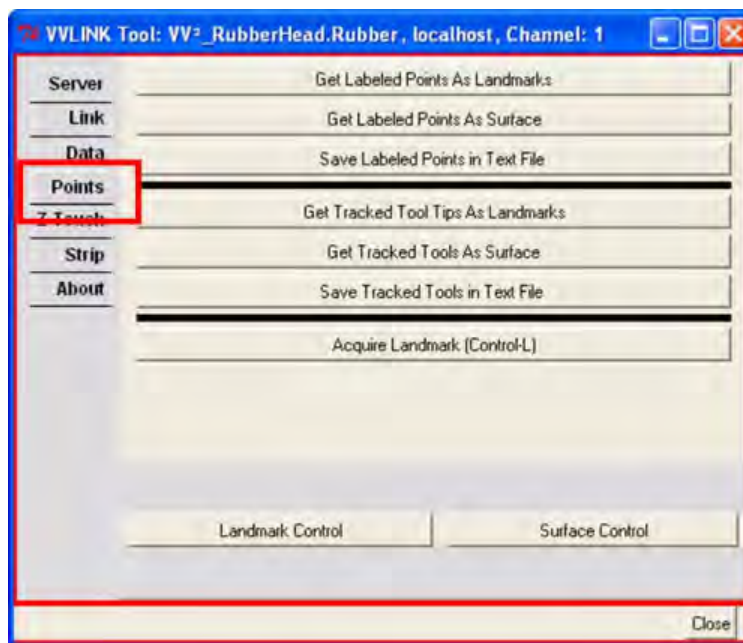


Figure 20.6: The “Points”-tab in VVLINK Tool, which has functions for saving the current positions of tracked tools, the locations of labeled landmarks, and for acquiring landmark points in the Landmark Control using the surgical navigation pointer.

**Tracked Tool Tips As Landmarks** button. This action places the landmarks in the clipboard of the Landmark Control. These landmarks can next be pasted into any pointset using the “PointSet->Paste” option from the Landmark Control menu. The Landmark Control is accessible using the **Landmark Control** button towards the bottom of the “Points”-tab.

- Surfaces – by pressing either the **Get Labeled Points As Surfaces** or the **Get Tracked Tool Tips As Surfaces** buttons. This action places the surfaces in the clipboard of the Surface Control. These surfaces can be pasted into any pointset using the “Edit->Paste” option from the Surface Control menu. The Surface Control is accessible using the **Surface Control** button towards the bottom of the “Points”-tab.
- Text Files – by pressing either the **Save Labeled Points in Text File** or the **Save Tracked Tools in Text File** buttons. The Save Tracked Tools option does not ask for a filename to eliminate any unnecessary delays. Instead, it simply saves the tools in a file titled tools\_time.txt in the current directory, where time is the current time, e.g. tools\_162339\_13Jul2006.txt. The current directory can be set using the **Switch Directory** option under the **File** menu of the current viewer.

In addition the **Add Landmark** button (or pressing Ctrl-L) can be used to capture a landmark from the current position of the surgical navigation pointer and place it in the Landmark Control, in a similar manner to clicking the mouse (shift-click) in the BioImage Suite viewer.

The Z-touch and Strip tabs which appear in the screenshots are Beta Features and may not be present in your version of BioImage Suite. They are not supported as they stand.

## Chapter 21

# The Differential SPECT Tool

### 21.1 Introduction

The SPECT tool implements several features for localizing focal epilepsy based on ictal and interictal SPECT subtraction methods. These features include two comparative methods to a healthy normal population, a straight subtract method for ictal and interictal SPECTs, a utility for combining activation blobs, and a cluster level statistics calculator base on random field theory. The two main methods are a reimplementaion of the ISAS (Ictal-Interictal Subtraction Analysis by SPM) algorithm and a new variant, the ISAB (Ictal-Interictal Subtraction Analysis by Bioimage Suite) algorithm.

### 21.2 Interfacing with the Data Tree Manager

The SPECT tool is design to be used with the Data Tree Manager and a datatree structure. The Image Tab of the SPECT tool, shown in Figure 21.1 provides the functions for interacting with the Data Tree Manager. The **Make Tree** button creates a template datatree with the necessary healthy normal images in place and blank nodes for the patient images. The **Grab Selection** button on the SPECT tool allows inputs to ISAS or ISAB to be set directly from the datatree. Any resulting outputs from the SPECT processing methods are automatically put into the current datatree at the appropriate node. This integration with the Data Tree Manager allows for easier organization of data and a better visualization of how the images relates to each other (see Chapter 15).

### 21.3 Ictal-Interictal Subtraction Analysis by SPM

The ISAS algorithm was introduced by Chang DJ et al. and McNally et al. It consists of several preprocessing steps followed by a comparison between the patient and a healthy normal population.



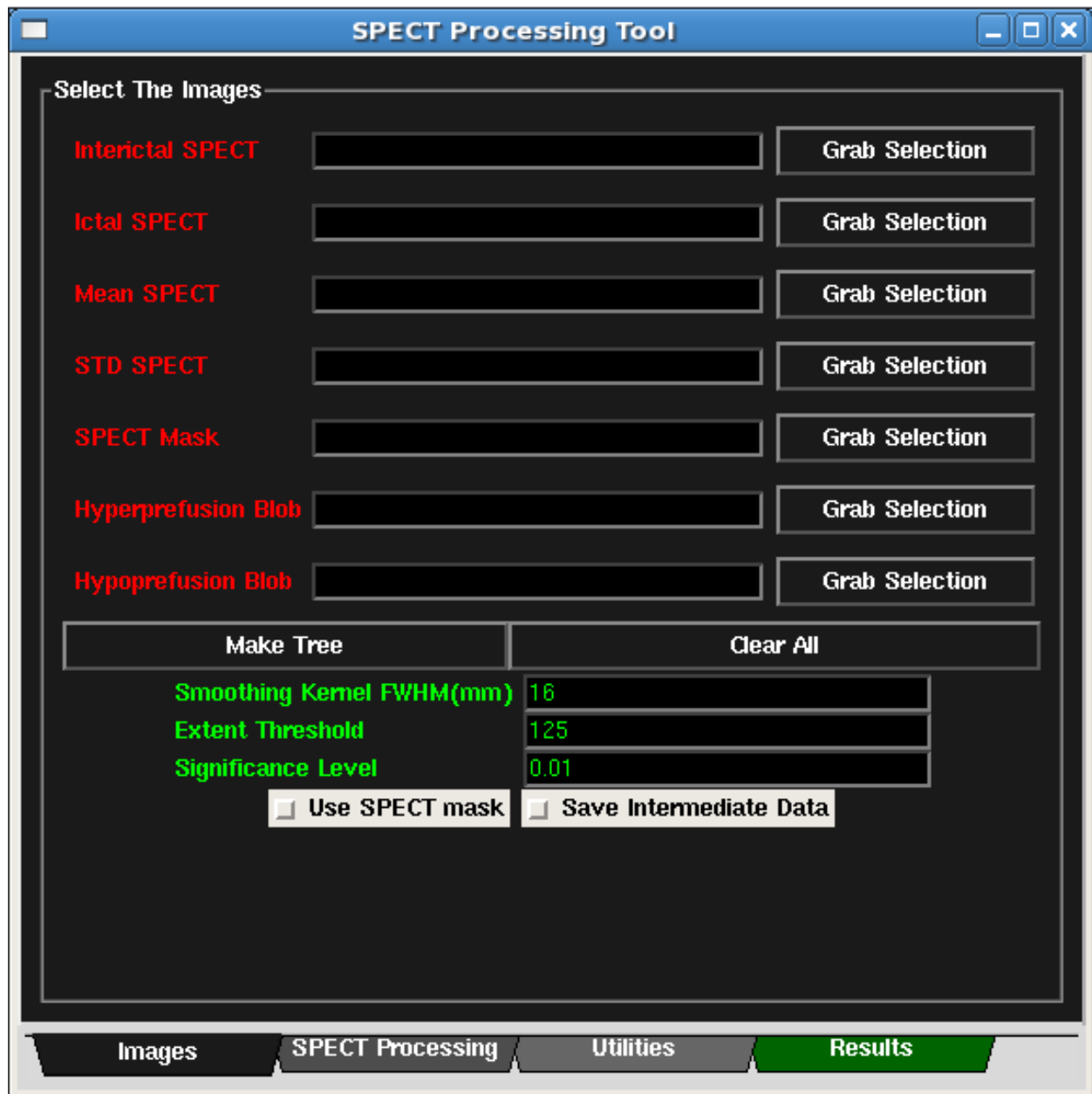


Figure 21.1: The Images Tab of the SPECT Tool. The Images Tab interacts with the Data Tree Manager, sets inputs for the processing methods, and changes various options.



First, the ictal and interictal SPECTs are non-linearly registered to MNI space. The warped SPECTs are then masked, smoothed and proportionally scaled to have a mean intensity value of 50. The proportional scaling scale factor is determined first by finding the full mean intensity (the mean intensity of every voxel in the image), second by finding the mean intensity of voxels greater than the full mean divided by 8, and third by dividing 50 by the mean voxel intensity. A voxel by voxel T-test or tmap is calculated by comparing the difference between the patient's ictal and interictal SPECT to the differences of between two SPECT images for healthy normals using:

$$t = \frac{x - \mu}{\sigma} \quad (21.1)$$

where

$$x = Ict_{Patient} - Int_{Patient} \quad (21.2)$$

$$\mu = \frac{1}{14} \sum_{k=1}^{14} Ict_{HNk} - Int_{HNk} \quad (21.3)$$

$$\sigma = \frac{1}{13} \sum_{k=1}^{14} (Ict_{HNk} - Int_{HNk} - \mu)^2 \quad (21.4)$$

The tmap can then be thresholded to find activation blobs and sent to the Results Tab for analysis. The **ISAS\_Calc** button is located on the SPECT Processing Tab in the SPECT Tool, shown in Figure 21.2. For step by step instructions for performing ISAS in BioImage Suite, please see example 21.8. For step by step instructions for performing ISAS in SPM, please see the ISAS website.

## 21.4 Ictal-Interictal Subtraction Analysis by Bioimage Suite

The ISAB algorithm uses the same preprocessing steps and healthy normal population as the ISAS algorithm. However ISAB assumes the mean noise between the healthy normal SPECT images is zero and estimates an unbiased standard deviation by using a half normal distribution. The tmap is calculated from the following:

$$t = \frac{x - \mu}{\sigma} \quad (21.5)$$

where

$$x = Ict_{Patient} - Int_{Patient} \quad (21.6)$$

$$\mu = 0 \quad (21.7)$$

$$\sigma = \sqrt{\frac{\pi}{2} \frac{\sum_{k=1}^{14} |Ict_{HNk} - Int_{HNk}|}{14}} \quad (21.8)$$

The **ISAB\_Calc** button is located on the SPECT Processing Tab in the SPECT Tool. For step by step instructions for performing ISAB, please see the example 21.8.

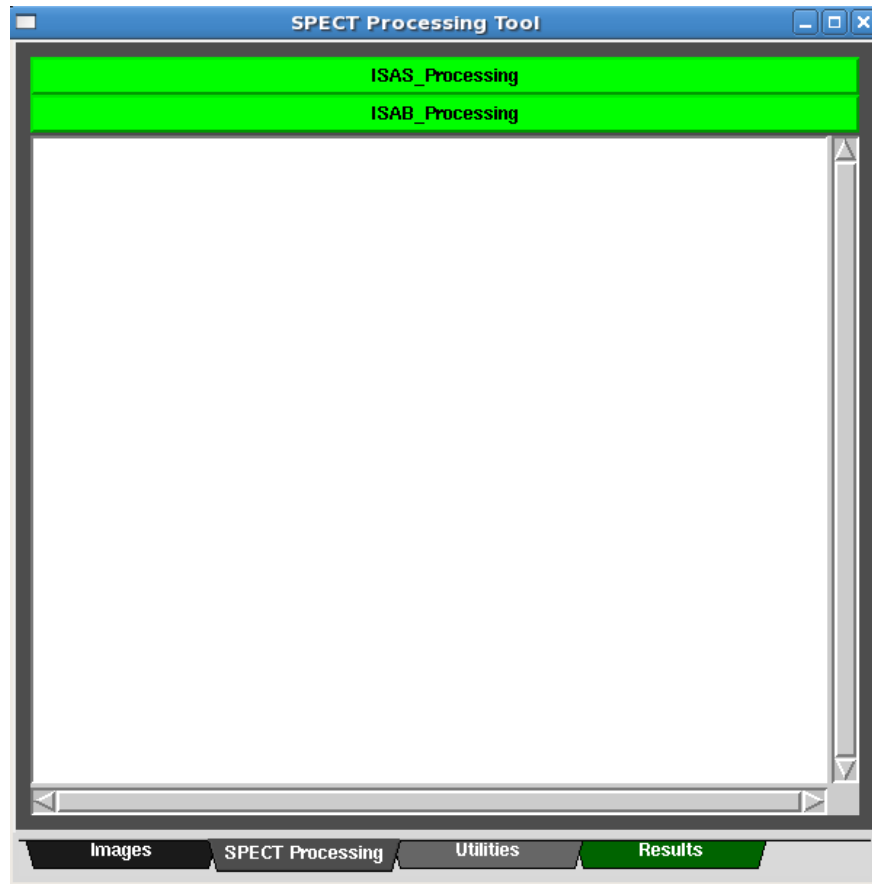


Figure 21.2: The SPECT Processing Tab of the SPECT Tool. The SPECT Processing Tab houses the ISAS and ISAB buttons and displays relevant information during calculations.

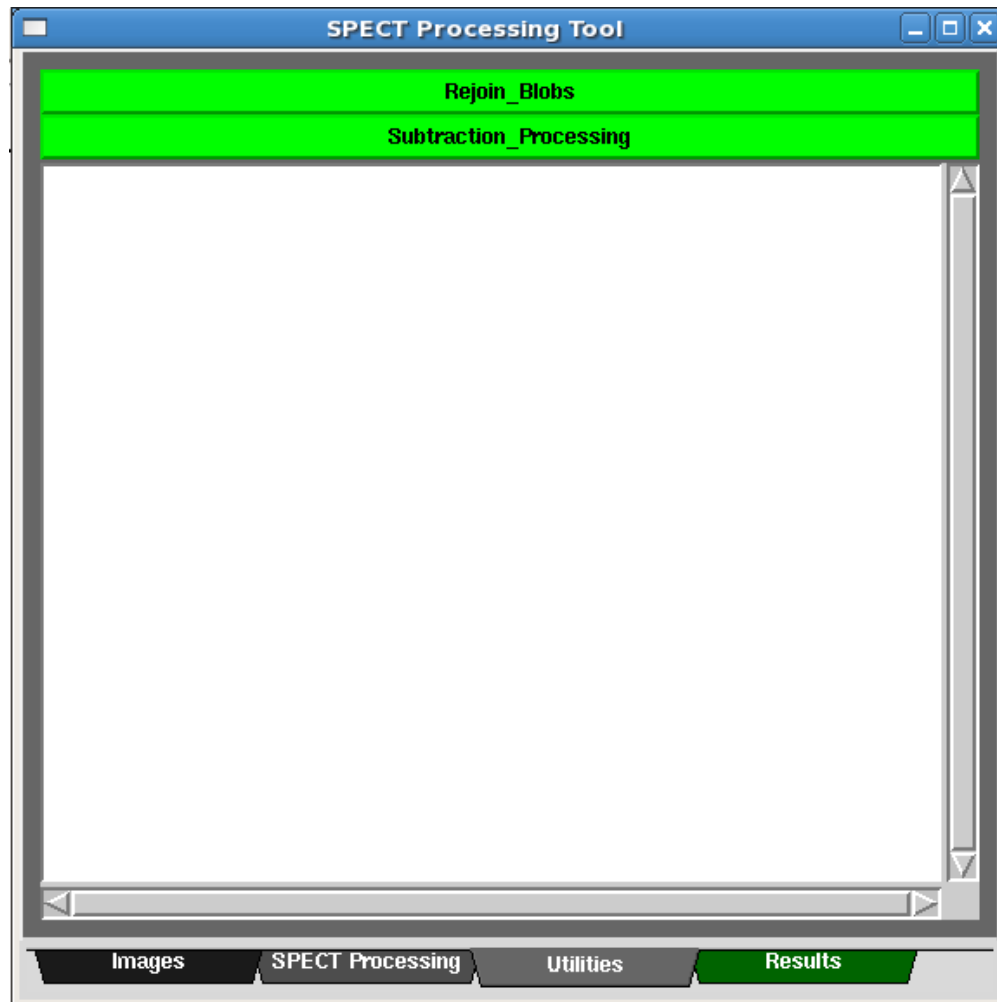


Figure 21.3: The Utilities Tab of the SPECT Tool. The Utilities Tab houses two buttons to assist in the SPECT process and displays relevant information during calculations.

## 21.5 Subtraction Processing

The **Subtraction\_Processing** button located in the Utilities Tab, shown in Figure 21.2, provides a double check against registration, masking, and smoothing errors that may occur with ISAS and ISAB. The ictal SPECT is first rigidly aligned and intensity normalized to the interictal SPECT. The interictal SPECT is then subtracted from aligned and normalized ictal SPECT. The resulting subtraction SPECT can be thresholded at an appropriate intensity to use as a check against ISAS or ISAB. This image is automatically calculated when either ISAS or ISAB is performed but can be calculated by itself. See Example 21.9 on using the Utilities Tab below for step by step instructions.

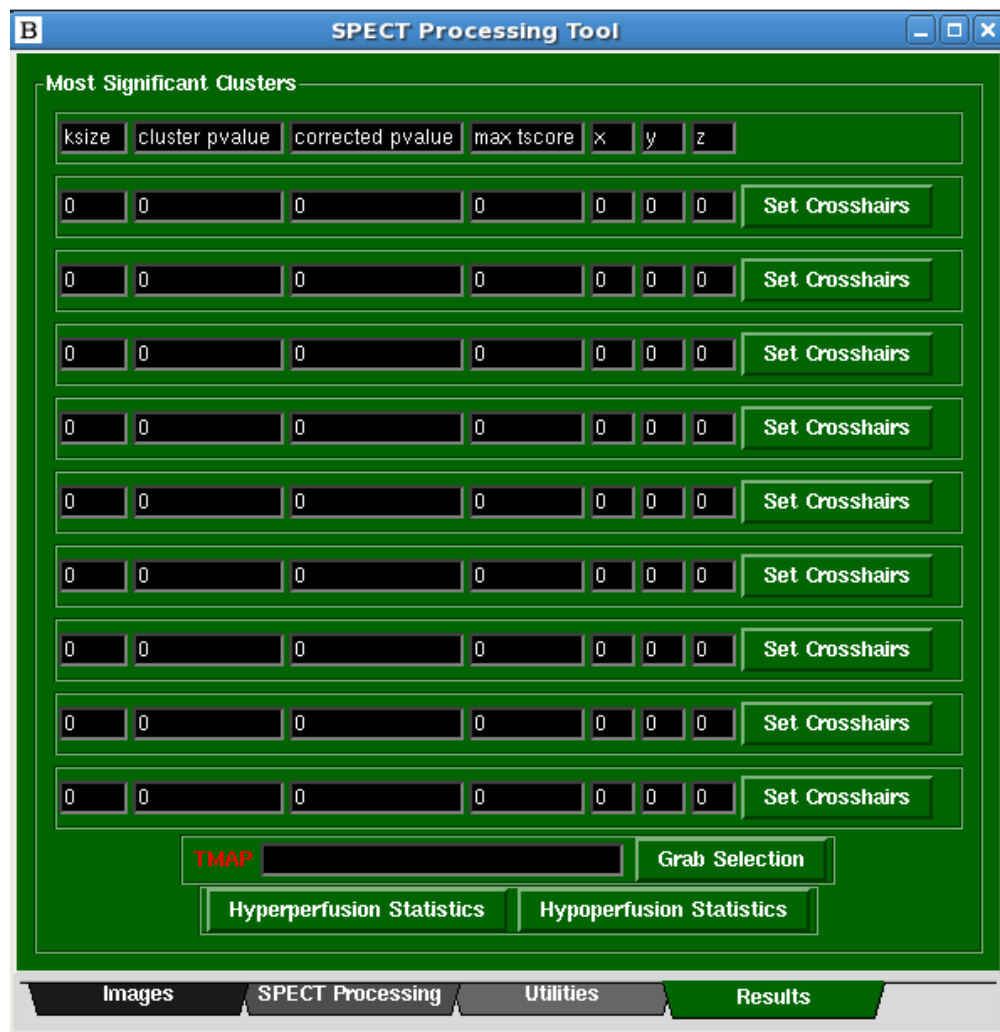


Figure 21.4: The Results Tab of the SPECT Tool. The Results Tab computes and displays cluster level statistics based on the tmap computed either from ISAS or ISAB.

## 21.6 Rejoining Blobs

The **Rejoin\_Blobs** button located in the Utilities Tab provides an interface for combining the activation blobs created in SPM. When ISAS is performed with SPM, two of the outputs are a hyperperfusion blob and a hypoperfusion blob. Combining these two blobs allows the blobs to be viewed together for better visualization of the changes between the ictal and the interictal SPECT. See Example 21.9 on using the Utilities Tab below for step by step instructing.

## 21.7 Cluster Level Statistics

The Results Tab provides an interface to compute cluster level statistics based on the tmap computed either from ISAS or ISAB. The **Hyperperfusion Statistics** button calculates the cluster level p-values based on increases between the ictal and interictal SPECTs while the **Hypoperfusion Statistic** button calculates p-values based on decreases between the ictal and interictal SPECTs. The uncorrected p-value is based on the smoothness of the healthy normal population SPECTS, the extend threshold, the significance level, and the size of the cluster. The corrected p-value also takes into account the shape of the tmap. The significance level and extend threshold can be set in the Images Tab of the SPECT Tool. The cluster level statistics are based off the work of Friston et al. and Worsley et al.

## 21.8 EXAMPLE: Running ISAS and ISAB

This section provides a step by step guide for running either ISAS or ISAB using the SPECT tool. It is assumed that the user is comfortable with the Data Tree Manager. Please see Chapter 15 for a description of the Data Tree Manager.

### 21.8.1 Setting Images

1. Open the Data Tree Manager in BioImage Suite. .
2. Click “Tools” in the menu of the Data Tree Manger, then “SPECT Processing Tool”. The SPECT Processing Tool will pop up in a new dialog box.
3. Click “File” in the menu of the Data Tree Manger, then “Switch Directories”. Select the appropriate working directory.
4. In the Images Tab of the SPECT Processing Tool, click the **Make Tree** button. See Figure 21.5.
  - (a) This creates a new datatree in the Data Tree Manager with all of the images preset for processing. The images and transformations between the MNI Template MRI, Mean SPECT, ISAS STD SPECT, and ISAB STD SPECT images are already specified.

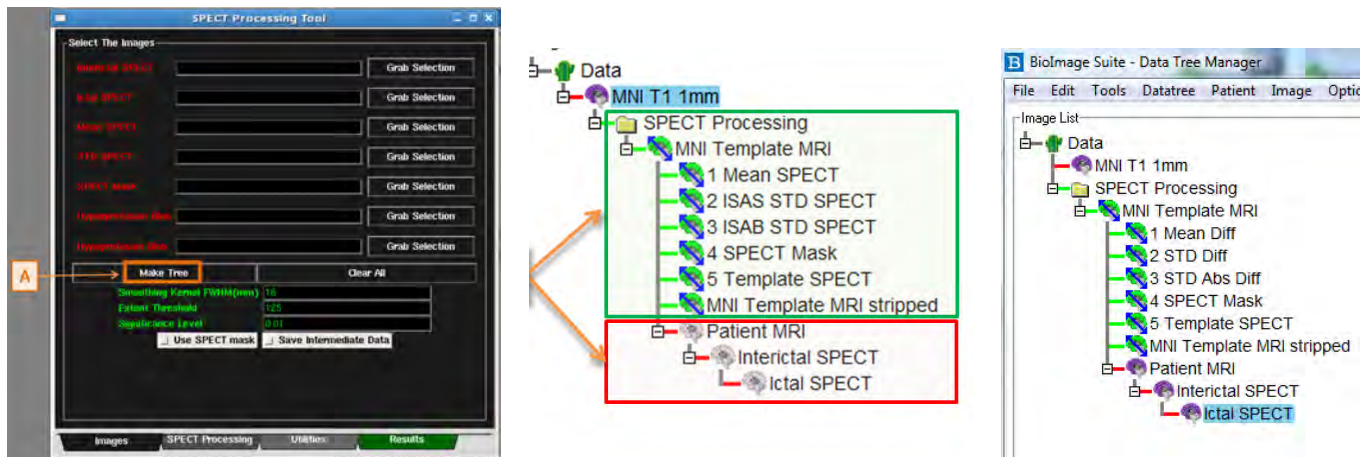


Figure 21.5: Making the template datatree using the **Make Tree** button. **Left:** The location of the **Make Tree** button, labeled A. **Middle:** The output of **Make Tree** button. The nodes in the green box are set while the nodes in the red box are not set. **Right:** The template datatree with all images loaded.

- (b) The required nodes that are not yet specified, i.e. Interictal SPECT, Ictal SPECT, Patient MRI, are gray and inserted into the tree as well.
5. In the Data Tree Manager, right click on the Patient MRI node and select “Set Filename”. Browse to find the appropriate filename that corresponds to the patient’s MRI. Here, the path of the image is being set; the image is not loaded to memory at this time.
6. Similarly, repeat for the Interictal SPECT node and Ictal SPECT node.

## 21.8.2 Registering Images

1. First register the Ictal SPECT to the Interictal SPECT using a rigid registration.
  - (a) On the Data Tree Manager, highlight the Interictal SPECT and click the **Set Space/Anatomical Image** button. See Figure 21.6.
  - (b) Highlight the Ictal SPECT on the Data Tree Manager and click the **Set Functional Image** button.
  - (c) Send each of these images to the viewers by clicking the **RF Viewer** button for Space/Anatomical image and the **TR Viewer** button for the functional image. This sends the Interictal to the “Reference” viewer and the Ictal to the “Transform” viewer.
  - (d) In the Data Tree Manager menu, click “Registration”, and then “Linear Registration.” The Registration/Overlay Tool should pop up in a new dialog box.
  - (e) To perform the registration of the Ictal SPECT to the Interictal SPECT, click the **Rigid** button.
  - (f) Once the registration is complete, you can check it in the viewers. Also, a message box should pop up displaying where the transformation was saved.

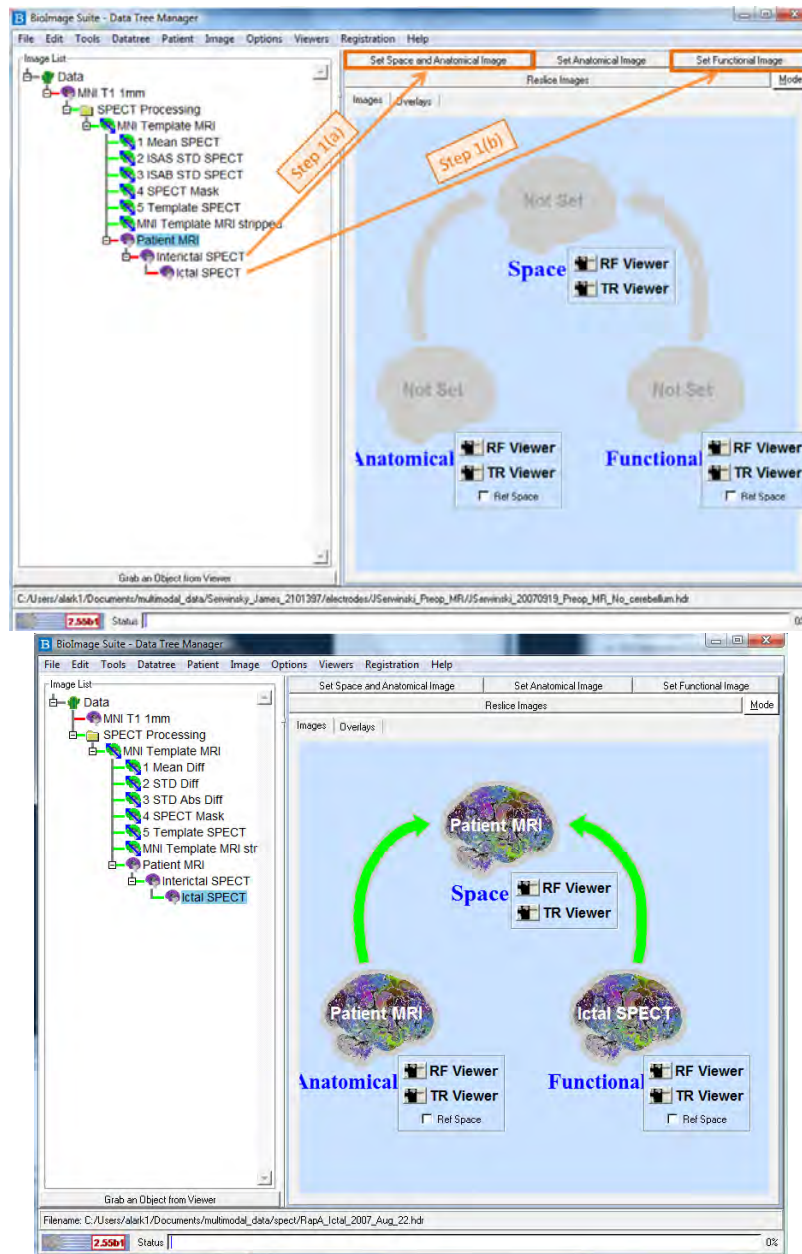


Figure 21.6: Registering Images for SPECT processing. **Top:** Setting the “Space/Anatomical” Image and the “Functional” Image. **Bottom:** A completed tree with all transformations loaded. Notice the lines between the nodes are green.



- (g) In the Data Tree Manager, right-click on the Ictal SPECT node, select “Load Transformation **From** Parent,” and load the automatically saved \*.matr file corresponding to the transformation of the Ictal SPECT to the Interictal SPECT.
2. Perform the same registration steps with the Interictal SPECT as the “Functional Image” and the Patient MRI as the “Space/Anatomical Image” to register the Interictal SPECT to the Patient MRI.
3. Perform the same registration steps with the Patient MRI as the “Functional Image” and the MNI Template MRI as the “Space/Anatomical Image” to register the Patient MRI to the MNI Template MRI. However, this time use the **Compute Linear+Non-Linear Registration** button found under the Non-Linear Registration Tab of the Registration/Overlay Tool instead of the **Rigid** button.
4. In the Data Tree Manger, right-click on the Patient MRI node, select “Load Transformation **From** Parent,” and load the automatically saved \*.grd file corresponding to the transformation of the Patient MRI to the MNI Template MRI image.
5. For non-linear registrations, the inverse transformation must be explicitly calculated and loaded to the tree.
  - (a) First click on the Transformation Tab of the Registration/Overlay Tool.
  - (b) In the list of transformations on the left had side of the tab, select the non-linear \*.grd file specifying the transformation between the Patient MRI and the MNI Template MRI.
  - (c) Click the **Invert** button located to the right. This will save the transformation as “Invert\_\*.grd”.
  - (d) Once the transformation is inverted, click the **Save** button to save the new \*.grd file.
  - (e) In the tree, right click on the Patient MRI node, select ”Load Transformation **To** Parent,” and load the saved \*.grd file corresponding to the inverted registration of the Patient MRI to the Template MRI image.

*Note:* All node lines between image should be green at this point as in Figure 21.6. The green arrows indicates that the transformations between the images are set. More information on performing registrations in BioImage Suite can be found in Chapter 10.

### 21.8.3 Performing ISAS and ISAB

1. First, select the Icterictal SPECT node on the Data Tree Manager. Then click the corresponding **Grab Selection** button on the Images Tab of the SPECT Tool. See Figure 21.7. The Interictal text should turn green once the image is set.
2. Repeat for the Ictal SPECT.
3. Set the population mean and standard deviation.
  - (a) For ISAS, set the Mean SPECT with the Mean SPECT image in the Data Tree Manager and set the STD SPECT with the ISAS STD image in the Data Tree Manager.

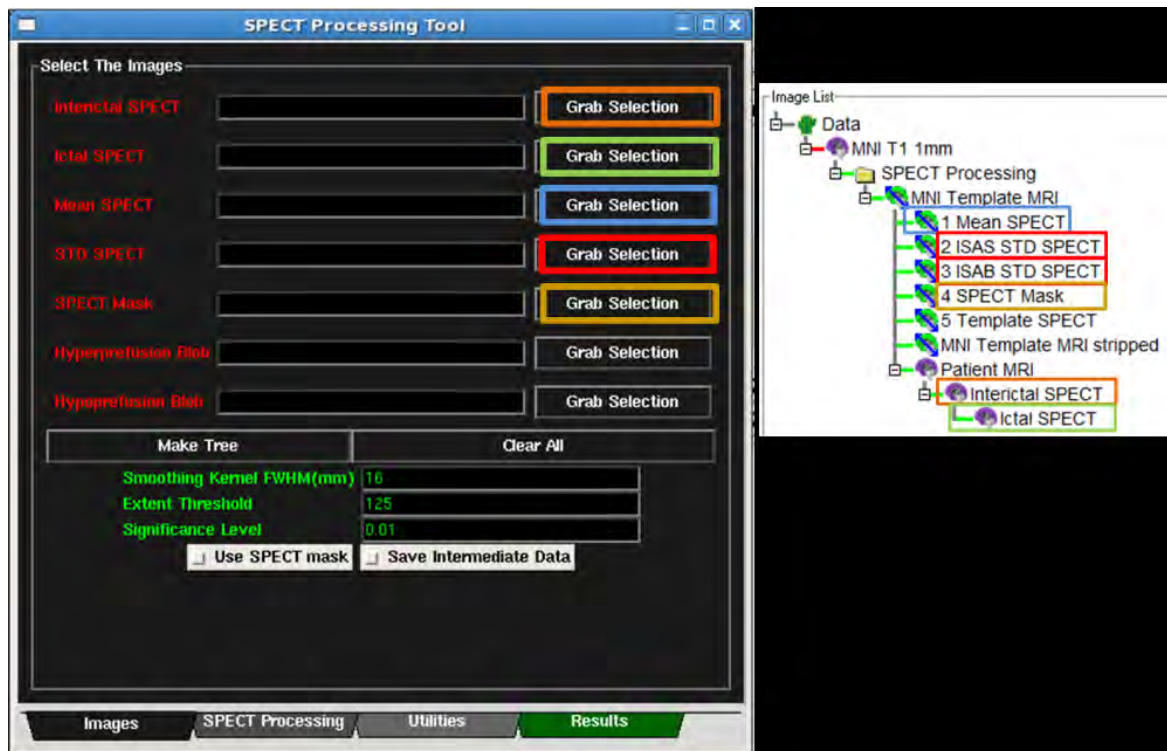


Figure 21.7: The data tree and the SPECT Processing Tool are color coded. The color coding shows which **Grab Selection** button corresponds to which node in the data tree.

- (b) For ISAB, set the STD SPECT with the ISAB STD image. *Note:* The mean image does not need to be set for the ISAB algorithm.
4. (Optional) Set the SPECT mask.
5. At this point, all the necessary images should be set and the text should turn green. See Figure 21.8
6. Set the options located at the bottom of the Images Tab to the correct values.
  - (a) The Smoothing Kernel, Extent Threshold, and Significance level are set to default ISAS and ISAB values.
  - (b) To include masking in ISAS or ISAB, check the “Use SPECT Mask” checkbox. *Note:* The SPECT mask must be set to use this option.
  - (c) To save all intermediate images, check the ”Save Intermediate Data”.
7. Click on the SPECT Processing Tab of the SPECT Processing Tool. Then click on either the **ISAS\_Processing** or **ISAB\_Processing** buttons to begin.
8. The outputs will be stored in the datatree under either the ISAS STD node or the ISAB STD node. The default outputs are the Preprocessed\_Ictal image, Preprocessed\_Interictal image, the TMAP image, and the Prefusion Blobs image.

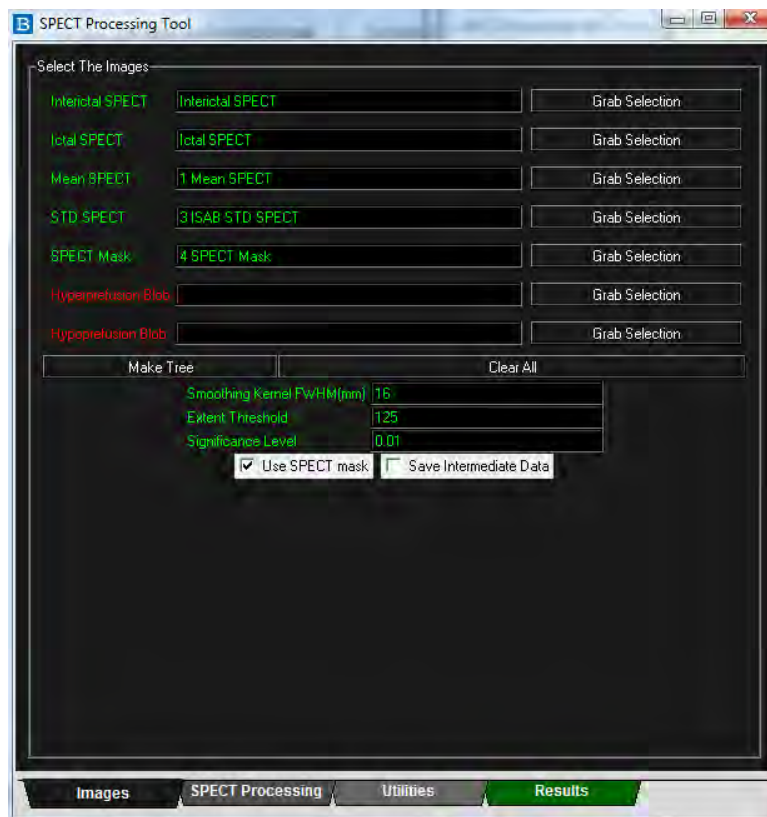


Figure 21.8: The Images tab with all images and options set.



Figure 21.9: The output of the creating an overlay of the Perfusion Blobs onto a template MRI.

### 21.8.4 Viewing Results

1. Create an overlay of the Perfusion Blobs onto the MNI Template MRI. Figure 21.9 shows the Perfusion Blobs overlaid on a template MRI.
  - (a) Highlight the MNI Template MRI node on the Data Tree Manager. Then, click the **Set Space/Anatomical Image** button located to the right.
  - (b) Highlight the Perfusion Blobs image on the datatree and click the **Set Functional Image** button.
  - (c) Click the **Reslice Images** button locate under the **Set Space/Anatomical Image** button.
  - (d) In the Overlays Tab of the Data Tree Manager (locate below the **Reslice Images** button), set the low threshold to zero and click the **Create Overlay** button.
  - (e) The transform viewer should now show the Prefusion Blobs overlaid on to the Template MRI.
2. On the SPECT Tool, click on the Results Tab and set the TMAP image with the output TMAP from the SPECT processing.
3. Click either the **Hyperperfusion Statistics** button or the **Hypoperfusion Statistics** button to calculate the cluster level statistics. The cluster size, cluster p-value, corrected p-value, the maximum tscore, and XYZ coordinates of the maximum tscore will be displayed on the Results Tab.

4. The **Set Crosshairs** buttons can be used to navigate the Transform Viewer to the XYZ coordinates of the maximum tscore.
5. To view the Subtraction\_Processing results, create an overlay using the instructions above with the Subtraction image as the “Functional Image” and the Patient MRI as the “Space/Anatomical Image”. *Note:* The Subtraction image is locate under the Interictal node on the datatree.
6. Create coronal slices of the Perfusion Blobs on the Patient MRI
  - (a) Create an overlay using the steps above, but use the Perfusion Blobs as the “Functional Image” and the Patient MRI “Space/Anatomical Image.”
  - (b) On the Data Tree Manager menu, select “Viewers” and then “Simple Viewer.” The Simple Viewer will now pop up.
  - (c) In the Simple Viewer main menu, select “Display” and then “Grab From Transform Viewer.”
  - (d) Adjust the number of rows and columns in the Simple Viewer to display desired number of coronal slices.

## 21.9 EXAMPLE: Using the Utilities Tab

This example illustrates how to use the Utilities Tab of the SPECT tool.

### 21.9.1 Running the Subtraction Processing

1. Start with a datatree with the Ictal and Interictal SPECT and the rigid transformation set. For detailed instructions on setting up the tree, refer to steps 21.8.1-21.8.2 in Example 21.8
2. First, select the Interictal SPECT node on the Data Tree Manager. Then click the corresponding **Grab Selection** button on the Images Tab of the SPECT Tool. The Interictal text should turn green once the image is set.
3. Repeat for the Ictal SPECT.
4. In the Utilities Tab of the SPECT Processing Tool, click on the **Subtraction\_Processing** button.
5. Once finished, the SPECT tool will place the result as a child of the Interictal SPECT node.
6. To view the results, please refer to Viewing Results on Example 21.8.

### 21.9.2 Running the Rejoin Blobs Utility

1. In a the Data Tree Manager under the MNI Template MRI, add a node, title it “Hyperperfusion Blob”, and set the hyperperfusion blob filename to the node. For more information on the Data Tree Manager see Chapter 15

2. Add a second node under the MNI brain. Title this node “Hypoperfusion Blob”, and set the hypoperfusion blob filename to the node
3. First, select the Hypoperfusion Blob on the datatree and then click the **Grab Selection** button on the Images Tab of the SPECT Tool. The Hyperperfusion text should turn green once the image is set.
4. Repeat for the Hypoperfusion Blob.
5. On the Utilities Tab of the SPECT tool, click the **Rejoin Blobs**.
6. The result, a single blob showing both hyperperfusion and hypoperfusion, will be placed in the datatree as a child of the Hyperperfusion Blob node. *Note:* Images created in SPM2 have a different initial orientation and will show up looking flipped in BioImage Suite. To correct for this problem, the image must be rotated  $180^\circ$  in the  $z$ -direction. For more information on rotating images, see Chapter 7.

## Part VII

# G. Cardiovascular Image Analysis



## Chapter 22

# 4D Surface Editor

### 22.1 Introduction

The 4D Surface Editor package (which predates the 3D version) is designed specifically for cardiac image analysis. As such it has additional functionality for this purpose, and it also provides graphical user interfaces to parts of the shape-based deformation analysis methods (see Papademetris et al. TMI 2002 [81].)

*There are five major differences in the user interface of the 4D surface editor as compared to the 3D version, namely:*

1. There is no objectmap functionality – there are only surface-based editing tools. The objectmap menu is replaced by a much simpler “Edit Surface” menu.
2. There is a “Cardiac” menu button in the menu bar which provides access to two additional controls.
3. There are four-tabs in the rightmost pane instead of three. The “Multi” tab corresponds mostly to the “Surface+” tab of the 3D editor, whereas the “Segment” tab has additional functionality for batch mode segmentation and curve interpolation
4. There is a cine-mode (if the loaded image has more than one time point) accessible using the “Movie Control” button in the “Image” tab.
5. The Spline Editor (which replaces the Spline/Objectmap Editor) has the ability to display what is termed “ghost curves”, i.e. the curves of the same surface at the previous and next time frames to enable the construction of temporally smooth sequences. This is shown in Figure 22.3.

The rest of this description assumes familiarity with the 3D editor and simply focuses on the above 5 points.

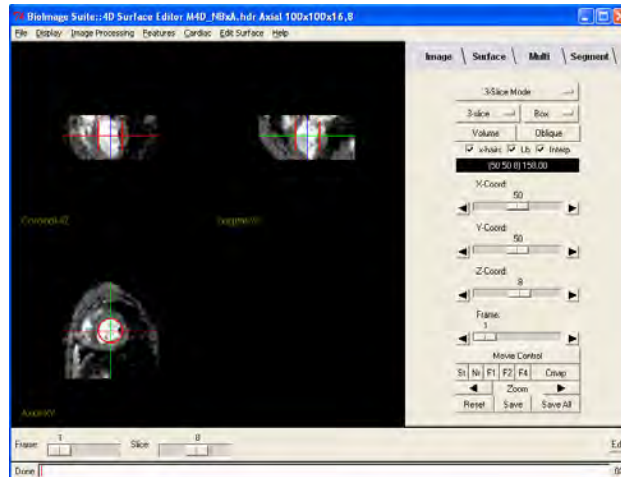


Figure 22.1: The 4D Surface Editor – Similar to the 3D surface editor, but with slightly different control features.

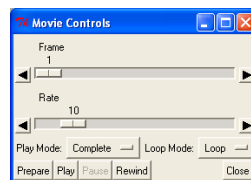


Figure 22.2: The Movie Controls Popup.

## 22.2 Movie Controls (Cine Mode)

The movie control (*shown in Figure 22.2*) is accessed using the “Movie Control” button in the “Image” tab. It has functionality for playing cine-loops of the contents of the viewer (e.g. image, surfaces, tstacks, strain maps etc.). It has two distinct play modes “Complete” which is equivalent to simply incrementing the frame and “Fast” which uses cached versions all frames prior to playing to achieve higher performance at the loss of interaction during movie mode. The caching is performed by clicking the “Prepare” button.

## 22.3 The “Multi” and “Segment” Tabs

The “Multi” tab is closely related to the “Surface+” tab in 3D Surface Editor. The key difference is that the Load All/Save All functionality refers to the loading/saving of temporal sequences of surfaces instead of surface sets. In addition the “Export” button provides access to a more advanced export facility, for exporting temporal sequences of surfaces.

The “Segment” tab provides facilities for spatial and temporal interpolation. In the “Spatial” tab the user specifies the top and the bottom slice and curves in between are filled in. In the temporal interpolation, the assumption is that the user will first trace the ED and ES surfaces and then the interpolation algorithm can be used to generate good candidate segmentations by simple interpolations.

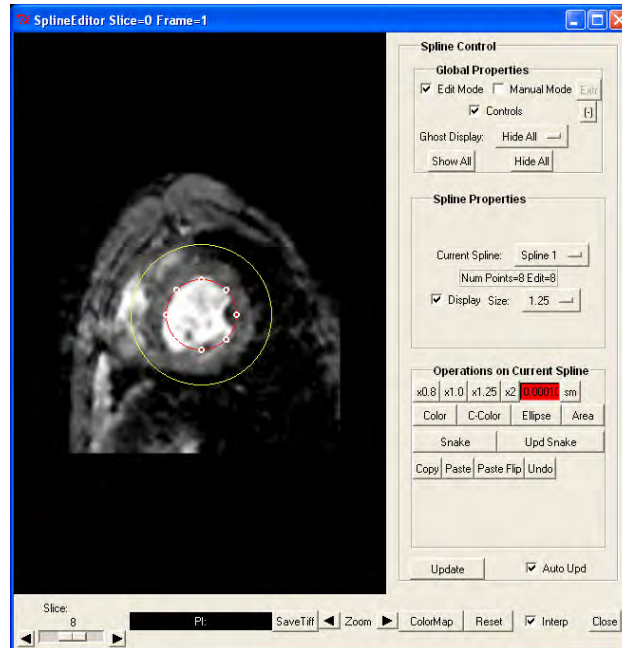


Figure 22.3: The Spline Editor is similar in spirit to the more advanced Spline/Objectmap editor used in the 3D Surface Editor. It does not have any “objectmap” functionality. The one additional feature is the “Ghost Display” option which enable the display of the previous and/or next curve(s) in the temporal sequence (the ghosts of the past and the future) to enable the easier construction of smooth temporal sequences.

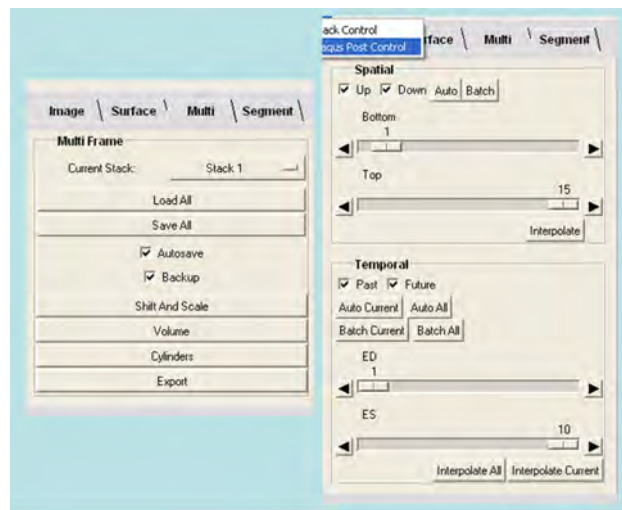


Figure 22.4: Additional Functionality of The 4D Surface Editor

The “Auto” and “Batch” modes refer to automated segmentation using the snake algorithm, This is work in progress and should be used sparingly.

## 22.4 Changes in the menu and controls

The 4D viewer has a number of different menu choices and associated control windows from the those of the 3D viewer.

**The Menu** The Cardiac menu shows how to access the “tstack” control and the “Abaqus post control”. The tstack control shown below has functionality for manipulating tstack (Triangulated Stack files). These are created from the surface .sur files and represent an explicit surface representation (in terms of points and triangles) as opposed to the implicit cubic polynomial representation of surfaces stored in the .sur files.

**The Tstack Control** The .tstack files can be generated using the “Export” button in the “Multi” tab of the main 4D surface editor window. These generates a series of files which can be Loaded/Saved using the Load/Save options in the T-stack control. Much of the rest of the functionality of the T-stack control mirrors closely that for the SplineStack control described earlier.

One key difference is the presence of the “Color Mode” menu. This selects the function used to create a color scale for the surface. These measures are based on the different combinations of the two principal curvatures (e.g. Gaussian, Mean, 1st principal, 2nd principal, bending energy and shape index). These colorscales are aids to visualization.

**Computing Curvatures** The curvatures can be computed using the “Computing Curvatures” control (shown in the top right) which is accessed using the “Curvatures” button. There are two parameters to be set. The “scale” parameter controls the size of the neighborhood used for curvature computation and the “smooth iterations” the number of smoothing iterations applied prior to curvature computation.

Once curvatures are computed they can be used as an input to the shape tracking algorithm. This is done in the “Shape-based .. tracking control” which is accessed from the “Shape-Tracking” menu on the T-stack control. The options involved are too technical for a brief overview of this kind. The same applies to the solid control which creates a finite-element mesh between selected endocardial and epi-cardial surfaces. The functionality captured in these controls (curvatures, shape-based tracking, create solid) represents most of the steps in the shape-based tracking deformation algorithm of [Papademetris et al, TMI 2002]. The one step missing is the finite element analysis step itself which requires the presence of the Abaqus Finite Element package.

The output of the Abaqus FEM package can be loaded into the “Abaqus Post Control” to visualize and quantify regional myocardial strains.



Figure 22.5: Additional menu choices in the 4D Surface Editor – The Cardiac menu is present here, containing two tools for working with cardiac images in the form of triangulated stack files, and output files from the Abaqus FEM package.

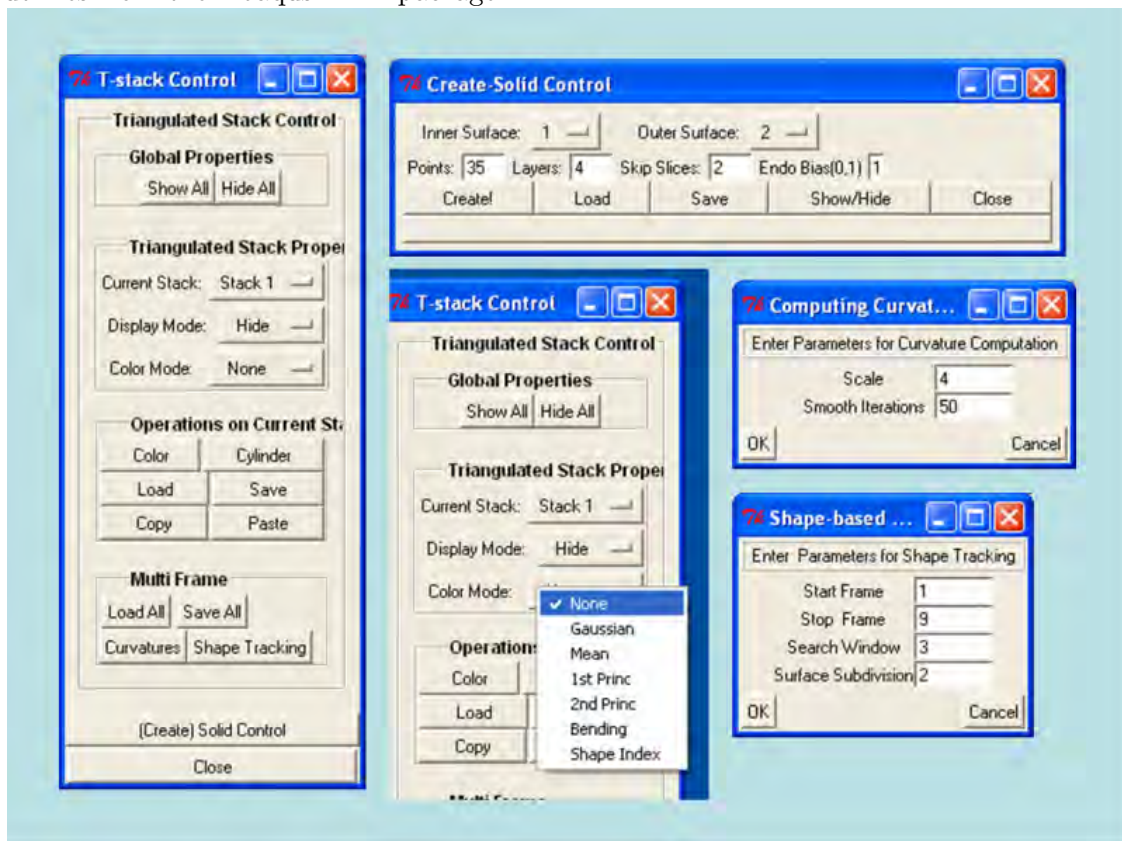


Figure 22.6: The various control windows that are accessible from the 4D Surface Editor.

## Chapter 23

# Estimation of LV Deformation

This document describes the commands that implement the cardiac strain computation methodology described in Papademetris et al TMI 2002[81]. The Surface Editor is used to generate the initial Surfaces in “.sur” format.

**Caveat:** One of the processing modules below (Step 5) requires the presence of the Abaqus Finite Element package and the local Yale extensions for this. Unfortunately these are not available for download at this point.

### Step 1: The Input Surfaces

The output of the tracing (manual segmentation) process consists of endo and epi cardial surfaces sometimes named endo.01.sur .. endo.02.sur .. endo.16.sur and epi.01.sur etc. Alternatively the study name is used in which case the files called \*N.01.sur are the endocardial surfaces and the files called \*P.01.sur are the epi-cardial surfaces.

**Note:** NEVER run the analysis in the same directory as the original files.

The first step is to determine the End-Diastole (ED) and the End-Systole(ES). We only run the analysis for these frames. In this case assume that ED is frame 2 and ES is frame 8. So long as ES ED things are ok. If not use the program pxabrenameframes to reorder the frames. (see step 2b)

### Step 2: Creating Triangulated Surfaces

The input surface .sur are in a complex b-spline stack format, and they need to be sampled to triangulated surfaces before they can be used for the analysis. (This effectively is step 2 of figure 1).

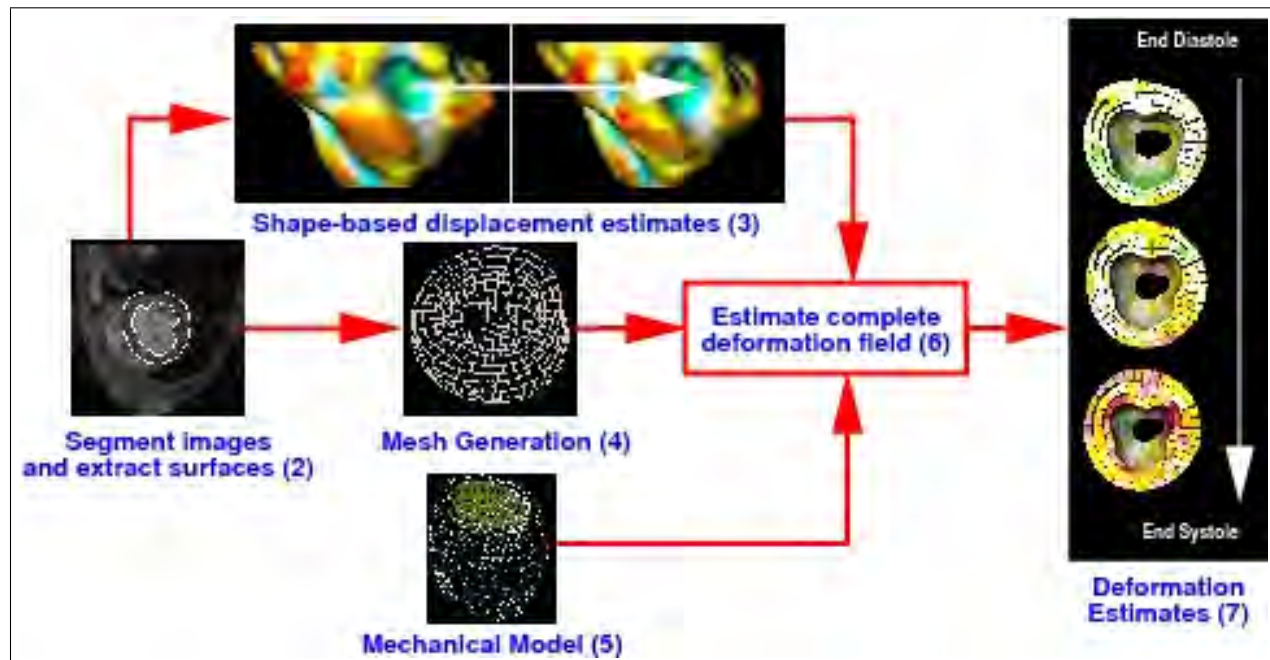


Figure 23.1: Outline of the complete process. The numbers in parenthesis (e.g. Mesh Generation (4)) refer to the section Papademetris et al 2002 where this component is described, i.e. mesh generation is described in section 4

**MRI:** The command to do this is `pxsur2tstack`

`pxsur2tstack` is surface to tstack converter

```
Usage : pxsur2tstack [ -IP 1.0 ] [ -OP 1.0 ] [ -CM 1 ] [ -DM=0 ] [ -PM ]
          [ -BZ -1 ] [ -TZ 10000.0 ] [ -IT 0 ] [ -SC 0 ]
          [ -IV 0 ] filename1 filename2
```

IP = Inplane spacing, OP = Out-of-plane spacing PM=Polar Middle

BZ=bottomz=-1 TZ=topz=10000.0, IT=Smoothing Iterations=0,

SC=Curvature Scale=0 CM=Mode (1=Linear,3=Cat-Spline)

DM=draw mode(-1=polar 0=open,1=closedbottom etc.) IV=Save Inventor file also

Typically this would be run as

```
pxsur2tstack -IP 1.0 -OP 1.0 -IT 100 -SC 3 -IV 1 endo.*.sur epi*sur
```

Sometimes the `-BZ` and `-TZ` flags are needed to crop the surfaces to a certain z-coordinate range. In this case if we wanted to crop the surfaces between 9.0 and 15.0 we could type:



```
pxsur2tstack -IP 1.0 -OP 1.0 -IT 100 -SC 3 -IV 1 -BZ 9.0 -TZ 15.0 endo.*.sur epi*sur
```

The output flag -IV 1 also produces inventor movie files. These can be viewed using ivview (if Open Inventor is installed on your machine)

```
ivview endo.01.iv
```

**Polar Ultrasound:** The command to do this is pxpolarsur2tstack

pxpolarsur2tstack is surface to tstack converter

```
Usage : pxpolarsur2tstack [ -IP 1.0 ] [-OP 1.0] [-CM 1] [-DM=0] [-PM]
        [-BZ -1] [-TZ 10000.0 ] [-IT 0 ] [-SC 0 ] [-IV 0 ]filename1
        filename2 .
```

```
IP = Inplane spacing, OP = Out-of-plane spacing PM=Polar Middle
BZ=bottomz=-1 TZ=topz=10000.0, IT=Smoothing Iterations=0,
SC=Curvature Scale=0 CM=Mode (1=Linear,3=Cat-Spline) DM=draw
mode(-1=polar 0=open,1=closedbottom etc.) IV=Save Inventor file
also
```

Typically this would be run as

```
pxpolarsur2tstack -IP 1.0 -OP 1.0 -IT 100 -SC 3 -IV 1 -BZ 80 -TZ 180 endo.*.sur epi*sur
```

Sometimes the -BZ and -TZ flags are needed to crop the surfaces to a certain z-coordinate range. In this case if we wanted to crop the surfaces between 90.0 and 175.0 we could type:

```
pxsur2tstack -IP 1.0 -OP 1.0 -IT 100 -SC 3 -IV 1 -BZ 90.0 -TZ 175.0 endo.*.sur epi*sur
```

The output flag -IV 1 also produces inventor movie files. These can be viewed using ivview

```
ivview endo.01.iv
```

## Step 2b: Renaming the frames

If ED is after ES (e.g. ED is frame 8, and ES is frame 1) use the program `pxabrenameframes` to reorder the frames.

```
Usage: pxabrenameframes first last period newfirst endostem epistem
```

To go from ED = 10 and ES=2 with a total of 16 frames we can type (assume endo epi name convention)

```
Usage: pxabrenameframes 10 2 16 1 endo epi
```

## Step 3: Running the Shape-Tracking Algorithm

This is the first real step in the analysis which generates the initial displacement estimates using the shape-tracking algorithm. This is done using `pxabshapetracking`. (This effectively is step 3 of figure 1.)

```
Usage : pxabshapetracking [-O outputstem] [ -F 1 ] [-L 16] [-LM 16] [-S
  1] [-IN 1] [-IT 5 ] [-SM 5] [-SW 3] [-CS 0] [-LA 0.5 ] [-G 0 ]
  inputname1 [inputname2] [inputname3] ..
```

1. Output Filename (same as input if not specified) O=name stem
2. Temporal Extent of Study default 16 frames, F=FirstFrame,  
L=LastFrame LM=Last Periodic Frame=16
3. Surface 2 Subdivision, S=Pixels
4. Shape Tracking Parameters IN - init mode, IT=init iterations  
SM=init smoothing iterations SW - search window,  
CS=shape-tracking smooth iterations -LA smoothing weight  
G - Global Alignment
5. Input filename: input name stem as is

To run the shape tracking algorithm for frames 2-8, endo and epi we type:

```
pxabshapetracking -F 2 -L 8 -SW 3 endo epi
```

#### Step 4: Generating the initial solid

To do this we need first two .tstack files with the same z-extent. First copy the ED surfaces i.e. in this case endo.02.sur and epi.02.sur to **a new directory** and look at the files to find their extent. If say endo.02.sur extends from 6.0 to 30.0 and epi.02.sur from 3.0 to 27.0 we have to use the common extent as:

```
pxsur2tstack -IP 1.0 -OP 1.0 -IT 100 -SC 3 -IV 1 -BZ 6.0 -TZ 30.0 endo.02.sur epi.03.sur
```

The solid can then be generated using the pxabcreatesolid command

```
pxabcreatesolid fname1 fname2 numstacks numnodes bias skipslices springs output
```

Typically we run this command as

```
pxabcreatesolid endo.02.tstack epi.02.tstack 4 35 1 2 0 solid.sld
```

A file called solid.iv will also be produced we can view this using ivview. (This effectively is step 4 of figure 1.)

## Step 5: Running the Finite Element Process

This is initiated using the `pxabcreatejobmulti` command.

```
Usage: pxabcreatejobmulti solidname endostem epistem enfMode
       springstiffness beginframe endframe resout materialmode
       incompress [orient=0 ] [run=1] [jobname][framename]
```

In this case we would type

```
pxabcreatejobmulti solid/solid.sld endo epi 2 4.0 2 8 0 1 0.4 2 2 outputFibers
```

This will generate the output data sets (after an hour or two) in the fiber specific directions. These need to be rotated in the cardiac coordinate frame using:

```
pxabcreatejobmulti solid/solid.sld endo epi 2 4.0 2 8 0 1 0.4 1 -2 outputRad outputFibersT
```

This should not take too long. (1-3 minutes).

The final results can be viewed using the Volume Viewer Control. This requires no parameters. The study can be loaded (this is the `.stat` file) in this case the important files will be called `outputRad.stat` and `outputFibers.stat`. (This effectively is step 6 of figure 1.)

## Step 6: Generating Average Strain Values (Pies) for analysis

This is done using `pxmakepies`.

```
usage pxmakepies studyname reppoint [numframes=12] [direction=1]
       [numslices=3] [numsectors=8] [numwedges=1] [ principal=0 ]
       [outputname]
```

To do this we also need a file usually ending in `.ref` which defines the reference point. This file is one line long (any other lines are ignored) consisting of the  $(x,y,z)$  coordinates of the reference point.

Typical usage for our case:

```
pxmakepies outputRad output.ref 20 1 3 8 1 0 cardiacpies
```

The important parameters are which produce a division in 3 slices x 8 sectors x 1 transmural.

### **Additional Visualization Tools**

You may use VolumeViewer to view the results. You will need the original images as a base. Further the image header has to have to correct information with respect to frames etc.

## Chapter 24

# Angiography Tools

**Note:** The angiography tools described in this document represent an implementation of the methodology of the work of Jackowski et al presented at MICCAI 2005 [52]. This is still an early implementation and should be used with care. The vessel tools are available in the *Vessel Tool* component of BioImage Suite.

The Angiography tools are closely related to the Diffusion tools as even a cursory look at the User Interfaces will make easily obvious. The key insight is that given an angiography image, once the Hessian matrix is computed (at multiple scales – this is performed using the Vessel Utility tool, tracking vessels is fairly similar to fiber tracking. This is accomplished using the Vessel Tracking Tool.

In general the procedure for extracting vessels is as follows:

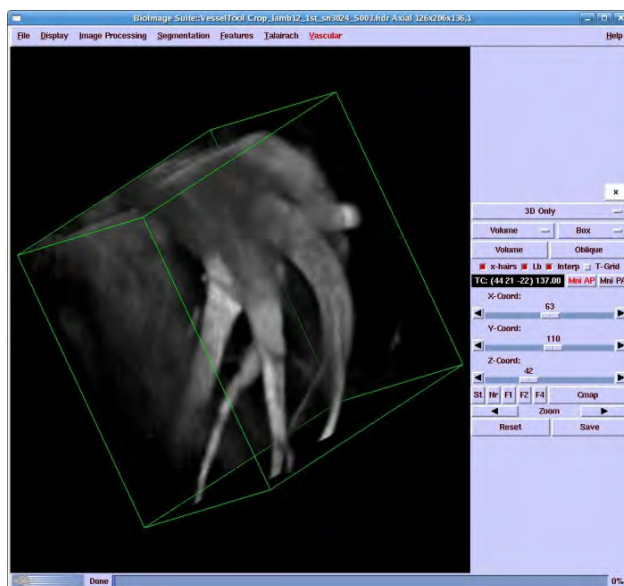


Figure 24.1: The Vessel Tool Program.

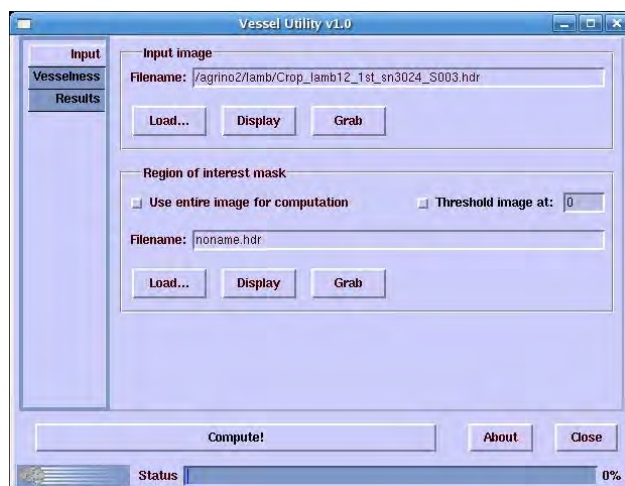


Figure 24.2: The Vessel Tool “Input Tab”.

- Compute the Hessian Matrix at different scales using the Vessel Utility Tool.
- Use the Hessian as an input to the Vessel Tracking tool to extract vessels.

## 24.1 The Vessel Utility Tool

The Vessel Utility Tool consists of three tabs, the “Input Tab”, the “Vesselness Tab” and the “Results Tab”, shown above. The general procedure for using this is as follows.

1. The user specifies the input image and an optional mask image in the *input* tab. The mask, if used, should be a binary image that encloses the region in which the vessel of interest lie. It is a computational speedup tool.
2. Next in the *vesselness tab* one needs to specify the maximum intensity inside the vessel (top left), and the size of the target vessels (in voxels), which is controlled by the range of scales set in the multi-scale analysis.
3. Once the setup is done, pressing the **Compute** button at the bottom of the image, will generate the output Hessian as well as other result files detailed below.

### The Input Tab

This essentially is used to specify the input image and the region of interest (mask) image.

The Input Image frame is used to specify the raw input image. The “Load” button loads an image from a file. The “Grab” button grabs the currently displayed image in the viewer, whereas the “Display” button sends the input image to the viewer.



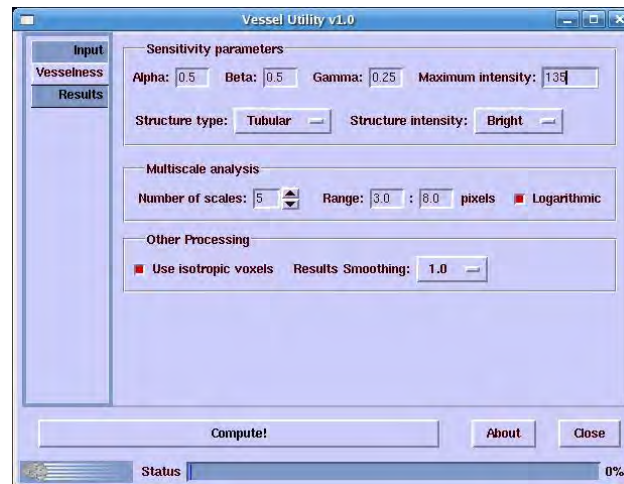


Figure 24.3: The Vessel Tool “Vesselness Tab”.

The Region of Interest frame is used to optionally specify a mask for the computation of the Hessian and associated parameters. There are three options:

- Use the entire image for computation. This is the default and is accomplished by checking on the “Use Entire Image for Computation” checkbox and leaving the “Threshold Image at” button to off.
- Use a thresholded version of the image. This is accomplished by checking on the “Use Entire Image for Computation” checkbox and leaving the “Threshold Image at” button to **on** and setting an appropriate threshold.
- Using an external mask. This can be generated using the Segmentation tools. A good way to obtain a mask is to threshold the image at some reasonable level and then dilate the mask somewhat to ensure that no vessels are lost due to locally dark regions. Then ensure that “Display Mask” in the Math\_Morphology tab (of the Segmentation Tool) is selected, display the mask and use the **Grab** button to grab it from the viewer into the vessel utility control.

## The Vesselness Tab

The vesselness tab is used to set the detailed parameters for the computation of the Hessian and the vesselness measure that is used to select the scale at which the Hessian is computed at each voxel (See Jackowski et al for details.)

The “sensitivity” parameters are used to evaluate the vesselness measure defined in equation (1) of Jackowski et al. The only parameter that needs to be modified is the **Maximum Intensity** parameter, which should be set to the approximate maximum intensity inside the vessels of interest. The other parameters should not be changed, unless one has specific reasons for changing them.

The “multiscale” analysis parameters should be set so as to reflect the range of vessel sizes one is interest in (in voxels). The Hessian matrix is computed at a number of scales (e.g. 5 in the example

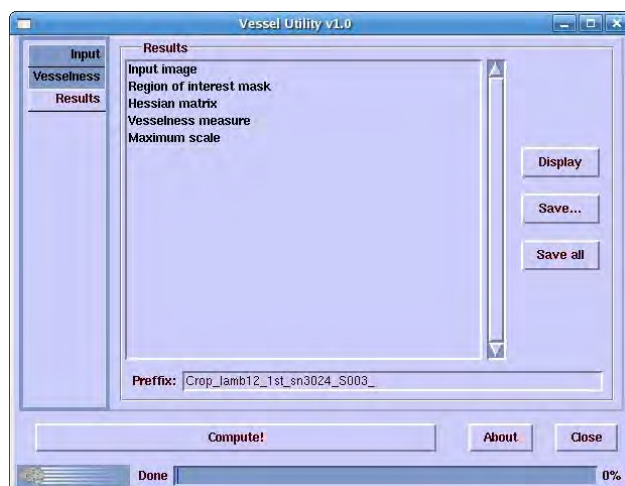


Figure 24.4: The Vessel Tool “Results Tab”.

shown) ranging from the minimum vessel size (e.g. 3.0 voxels in this size) to the maximum vessel size (e.g. 8.0 in the figure shown). If the “logarithmic” check button then the scales are concentrated more towards the lower end of the range, otherwise they are evenly distributed.

The “other processing” parameters tab has two parameters. If the “isotropic” checkbox is enabled, the input image is resliced to isotropic resolution (average of all three voxel dimensions) prior to any subsequent processing, otherwise the original image is used. The “Results Smoothing” option menu can be used to enable smoothing of the “Vesselness Measure” and “Maximum Scale Results” to make them more visually pleasing. This smoothing does not affect the computation of the Hessian in any way.

## The Results Tab

The results of the processing are stored in the results tab. The key ones are:

- The Hessian matrix, which is the key input to the vessel tracking tool.
- The Vesselness measure, which represents the likelihood of having a voxel *centerline* at any given voxel.
- The Maximum scale measure, which represents the most likely estimate of vessel radius (in voxels) at each location. This measure can be error prone and should be used with care.

The results can be displayed using the **Display** button and saved using the **Save** or **Save All** buttons.

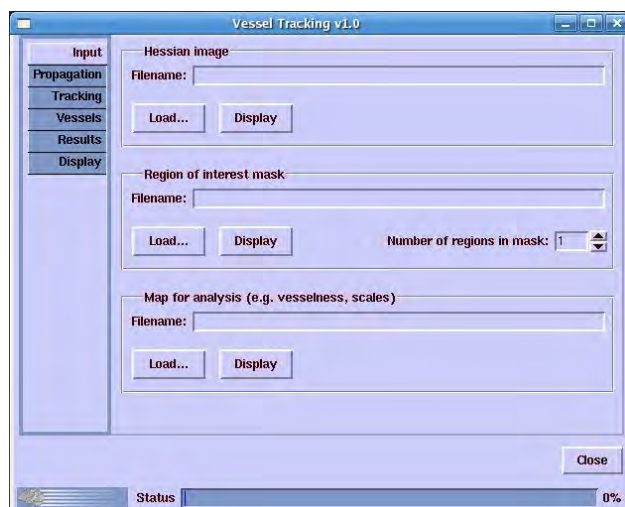


Figure 24.5: The Vessel Tracking Tool.

## 24.2 The Vessel Tracking Tool

The Vessel Tracking tool takes the Hessian Matrix computed from the Vessel Utility tool above as an input and uses fast marching techniques (see Jackowski et al for details.) to extract vessel centerlines. This tool is closely related to the Fiber Tracking tool and the documentation for the Fiber Tracking tool should also be consulted for the vessel tracking tool (see Chapter 18.) In this document, we mostly highlight issues specific to vessel tracking.

### Loading the Images

First, you will need to load the Hessian image, computed by the Vessel Utility (above). Secondly, load either the anatomical mask you have created earlier or the ROI you would like to use for tracking in the Region of interest mask. Finally load the auxiliary image, which could be either the vesselness or the maximum scale image computed by the Vessel Utility tool, in the Map for analysis box. Once all these input images are loaded, make sure that you have an image in the viewer (e.g. mask, or map for analysis) before you start tracking.

### Propagation

Once the data is loaded, the next step is to compute the fast marching solution. Switch the control to the “Propagation” tab, and **place the viewer crosshairs and the “center” of the region of interest, e.g. in a large vessel.** Then press the **Compute** button. Once the propagation is computed then vessel tracking can be performed.

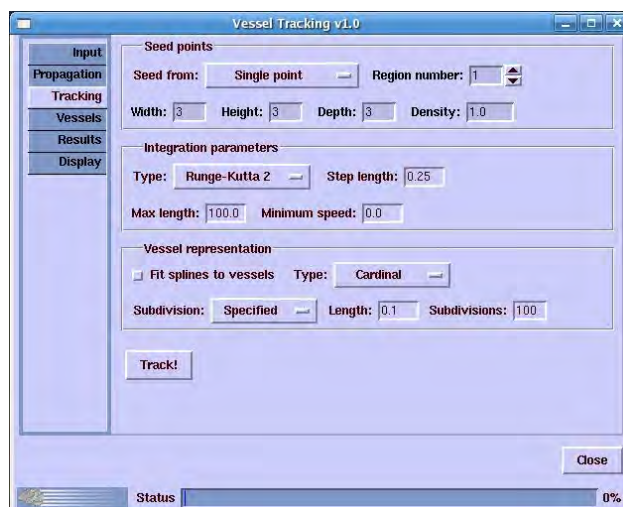


Figure 24.6: The Vessel Tracking Tool – the “Tracking” tab.

## Tracking

**Seed point selection:** You may choose to seed the tracking using a single point, a region of interest, a volume, or points from the landmark control. In case of a single point, the position of the cross-hairs in the main viewer will be used as coordinates for the point. In case of a region of interest, you must specify the region number to start tracking from. All points from the ROI will be used as seeds. You can also increase the point density. For volume, a 3D window of size Width x Height x Depth is taken to initialize the tracking.

**Integration Parameters:** The integration method used by the Fiber Tracking program is the well known Runge-Kutta. You may specify second-order or fourth order integration, which provides better accuracy. The step length (h) for integration represents the percentage of a voxel diagonal’s length. Higher percentages will yield coarser results while smaller values will yield smoother vessels. Once you made sure the relevant parameters were set, simply press the Track! button to start tracing.

## Vessels

Once tracking is complete, you are taken to the Vessels tab, in which the tracing results will accumulate. Vessel names and colors can be changed, simply select a vessel from the list and either change the label or press the color button. They can also be made visible or hidden by clicking the Display checkbox. You can save a vessel or a set of vessels by selecting them from the list and pressing the Save button. Note that you will be prompted for a directory name for saving. The vessels are saved according to their label names. To save a vessel with a different name, you must change its label first.

**Important!** The Vessels are not immediately visible in the viewer, until you switch the viewer to 3D mode, and select View Vessels under the Display tab. For more information on the “Display” tab see the documentation for the Fiber Tracking tool.

## Chapter 25

# Processing Mouse Hindlimb SPECT/CT Images

### 25.1 Introduction

BioImage Suite is an integrated image analysis software suite developed at Yale University to facilitate the processing of multiple image modalities. It has been used for CT and SPECT images from cardiac animal studies, but has applications to MRI as well as human studies.

This chapter derives from the “Mouse Suite” tutorial, originally written by Patti Cavaliere with help from Don and Donna Dione. It guides the user through the reorientation and cropping of mouse hind limb SPECT images, and co-registration with CT images, to result in segmentation of hind limb images and statistical evaluation of the region of interest (ROI).

The procedure involves the following 5 steps.

- Step I – Flip and crop images
- Step II – Remove table from CT images
- Step III – Draw planes on CT images
- Step IV – Segment soft tissue from CT images
- Step V – Compute ROI statistics

The processing is all done within the programs “Mouse Segment” and “Mouse Register” which can be found under the “MouseSuite” tab from the BioImage Suite main menu. To start invoke MouseRegister. Once this program opens, three windows will appear: BioImage Suite::MouseRegister, Reference Viewer, and Transform Viewer as shown in Figure 25.1.

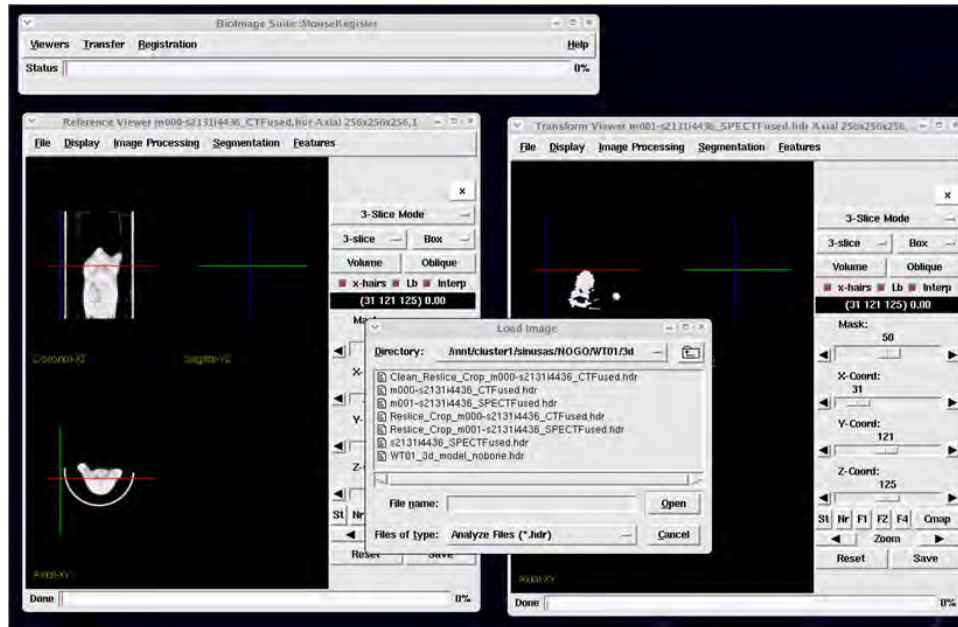


Figure 25.1: Loading Images into the Mouse Register application

## 25.2 Flip and Crop Images

### 25.2.1 Flip Images

Each Viewer can be designated for either CT or SPECT image processing. For this demonstration, we will use the Reference Viewer for CT, and the Transform Viewer for SPECT images. See step 6.

1. Starting with the Reference Viewer window, choose File, Switch Directory. The Select Current Directory box appears.
2. Select the directory that you want to open. Click OK.
3. Go back to the Reference Viewer window to load study images.
4. Choose File, Load. The Load Image box appears.
5. Double click the name of the study that you want to load.

Note: Use first CT study name when choosing which CT image to load.

6. In the Transform Viewer window, go to File, Load, then choose SPECT.

Note: Once the study is loaded, three images will appear in each of the two viewer windows. Each image will represent three orientations of the mouse designated as Coronal, Sagittal, and Axial. Hold down the left mouse button to move the color-coded-cursors. The cursor will move concurrently in both screens.

7. Make sure the CT and SPECT images are registered with each other in the x and y directions.

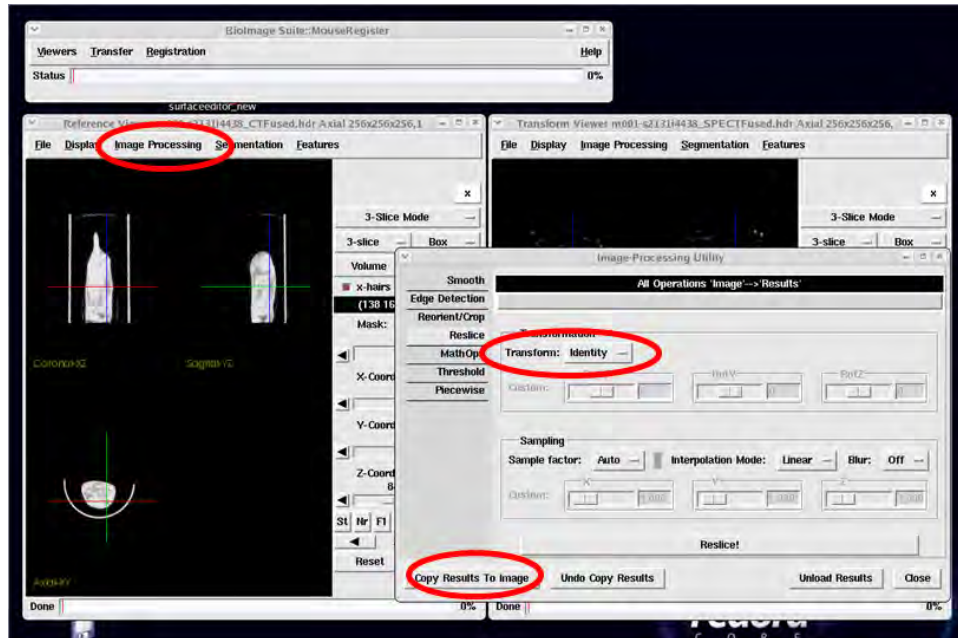


Figure 25.2: Flipping Image Orientation (if needed)

8. Follow steps to edit the header – this is often a problem with the specific datasets used in the preparation of this tutorial and may not apply in the general case.
  - (a) Go to File on the Reference Viewer window.
  - (b) Choose Image Header Editor.
  - (c) The Image Header Editor box opens.
  - (d) Under Voxel Dimensions, change the x-size, y-size, and z-size to 1.0, if needed.
  - (e) Next click Save, and then Close.
  - (f) If the header was changed, then repeat steps 4 and 5 to reload image.
9. Repeat steps 8 a-d to edit the header on the Transform Viewer window.
10. To flip images to the correct orientation, start with the Reference Viewer(CT images) window. Choose Image Processing, then Reslice Image. The Image Processing Utility box appears – see Figure 25.2
 

Click the Identity button, then hold down the left mouse button and choose Flip X. Next click Reslice! When reslicing is complete, then click Copy Results To Image. Repeat above for the Transform Viewer window(SPECT images). You are now ready to crop the images.

### 25.3 Cropping CT Images

Begin with either Reference Viewer(CT), or Transform Viewer(SPECT) window.



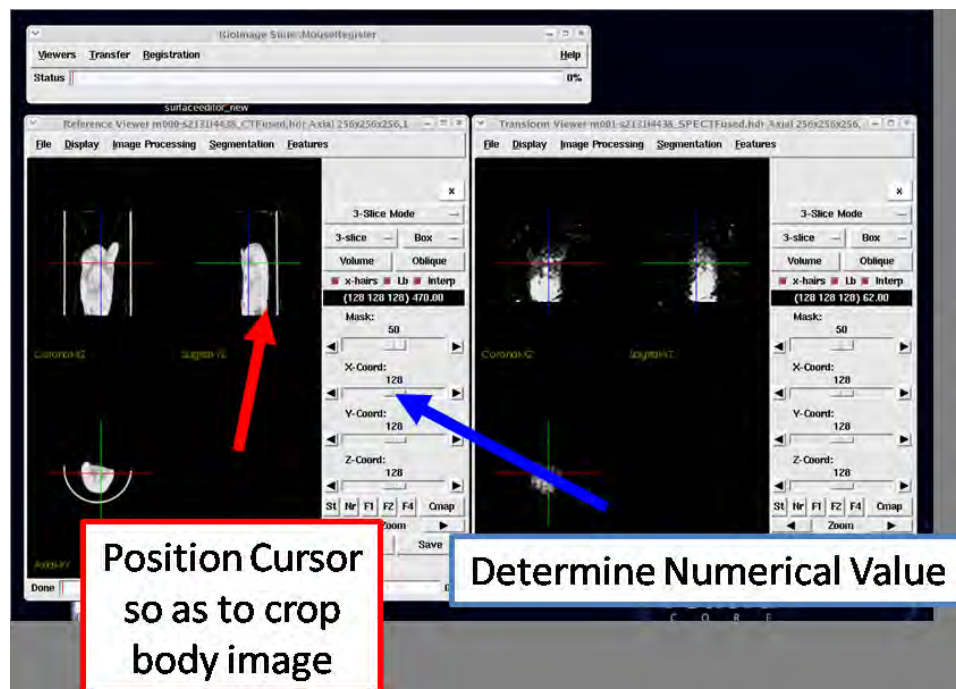


Figure 25.3: Identifying the Cropping Extent.

1. Position the cursor alongside the body so as to crop body image from background. Use care not to crop too close to body-opt for a larger image – see Figure 25.3.
2. Determine the numerical values of each coordinate visible using the scale to the right of the image. Repeat process for each side of image. Use Reference Viewer to determine x and y coordinates, and Transform Viewer to determine the z coordinates – again see Figure 25.3.
3. Record each coordinate on a piece of paper as follows:

	LOW Value	HIGH Value
x	( )	( )
y	( )	( )
z	( )	( )

4. Once coordinates are recorded, choose Image Processing on the Transform Viewer menu.
5. Next choose Reorient/Crop. A box will appear.
6. Type coordinates into table, then click Crop!. Note: New values will appear in table based on cropped size of object.
7. Click Copy Results To Image.
8. On the Transform Viewer window, Choose File, Save. A Saving Image box will appear with a prompt to rename the cropped image.
9. Rename image as CropRsliceName of study, then click the Save button.
10. Repeat steps 4 through 9 on the Reference Viewer window.



This completes cropping of CT and 1st SPECT images. Return to Transform Viewer window to begin cropping of 2nd SPECT images.

### 25.3.1 CROPPING 2nd SPECT IMAGES

1. Return to the Transform Viewer window. Repeat steps 3-7 of Step I, LOADING STUDY IMAGES starting on page 7, to load the 2nd SPECT image.
2. Once the study is loaded, three images will appear in the viewer window. Note: The bladder may be visible in Tc99m images in comparison to TL-201 images.
3. Choose Image Processing from the menu on the viewer window. Click Reslice Image. The Image Processing Utility box appears.
4. Click the Identity button, then hold down the left mouse button and choose Flip X. Next click Reslice!.
5. Next click Copy Results To Image.
6. Choose Reorient/Crop from the Image Processing menu.
7. Type x, y, and z coordinates into table.
8. Click Crop!, then click Copy Results To Image.
9. On the Transform Viewer window, Choose File, Save. A Saving Image box will appear with a prompt to rename the cropped image as CropResliceName of study.
10. . Minimize the Transform Viewer window and proceed to Step II REMOVING THE IMAGING TABLE FROM CT VIEWS.

## 25.4 Removing the Imaging Table from CT Views

1. Return to the Reference Viewer(CT) window.
2. Choose File, Load. The Load Image box appears again.
3. Check to make sure that you are in the correct Directory, then choose the CT image that has been resliced and cropped.
4. The Reference View window will appear with (3) images.
5. Choose Segmentation on the Reference Viewer.
6. Next click Math Morphology from the dropdown menu.
7. A Segmentation Control window opens – see Figure 25.4.
8. Enter a value of 200 in the Low Threshold box. Move the sliding arrow to the far right in the High Threshold box.

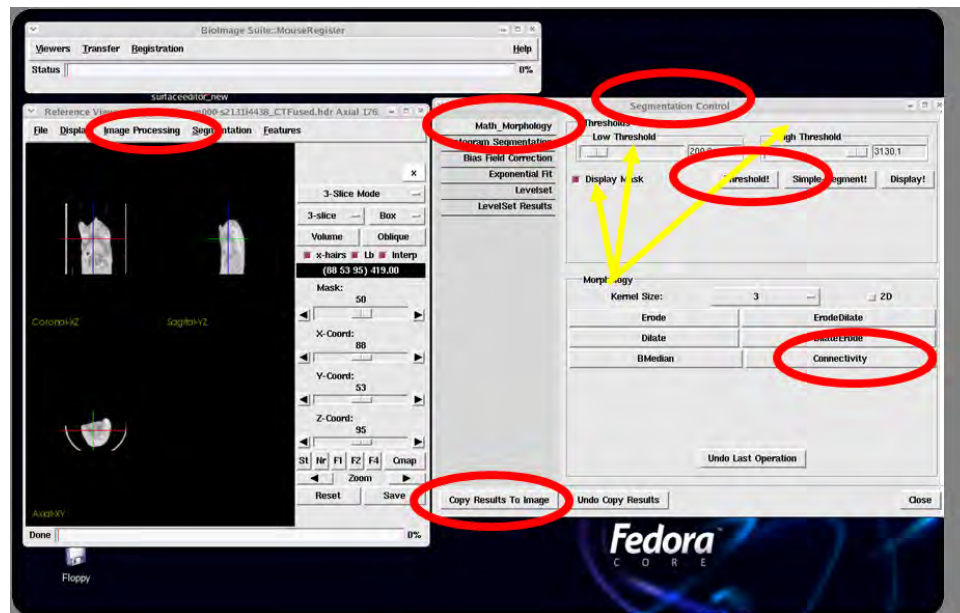


Figure 25.4: Using the Segmentation/Math Morphology Tool to mask out the imaging table from the mouse CT image.

9. Make sure the Display Mask is red (enabled).
10. Then click the Threshold! button. The mouse image will turn pink.
11. Position the cross-shaped cursor (+) on mouse body .
12. Click the Connectivity button. The table in the Reference Viewer window may change from pink to white.  
 Note: If the table does not turn white after clicking the Connectivity button, click Erode, then re-click the Connectivity button. This Connectivity/Erode process may be repeated, up to three times. When table disappears, then press Dilate.  
 NOTE: the Dilate button must be clicked the same number of times that the Erode button was clicked.
13. Uncheck the “Display Mask” checkbox. The mouse image will turn white again.
14. Click the Display! button. The table will disappear.
15. Go to File, Save. A Saving Image window will appear.
16. Select the name of the study, but rename it as CleanCropResliceName of study.

## 25.5 Draw Planes on CT Image

1. Return to the Reference Viewer(CT) window.

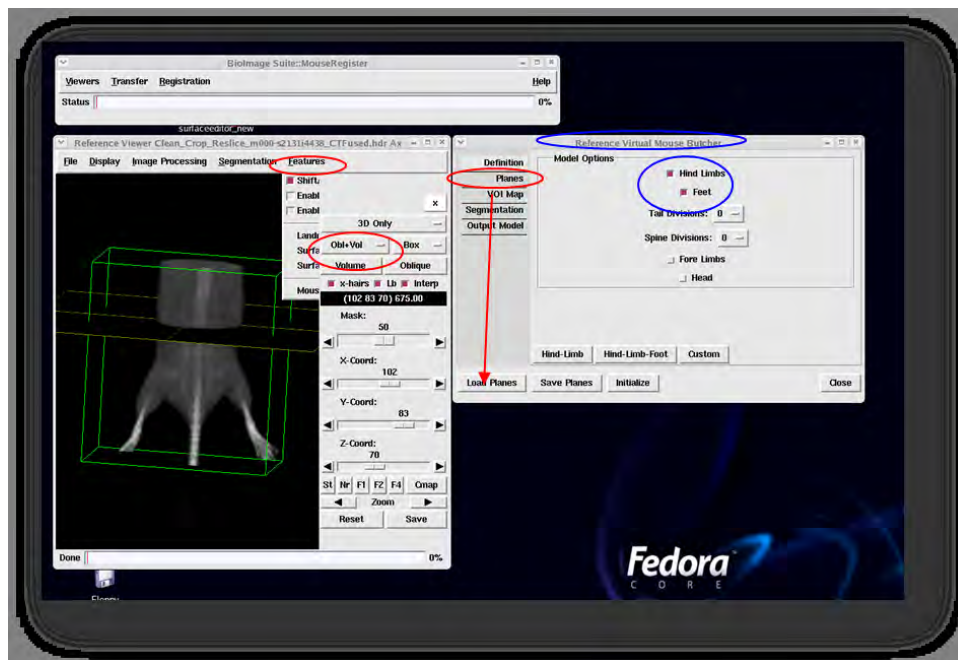


Figure 25.5: Bringing up the “Virtual Butcher” control which enables the parcellation of the CT Image into regions of interest using Cropping Planes.

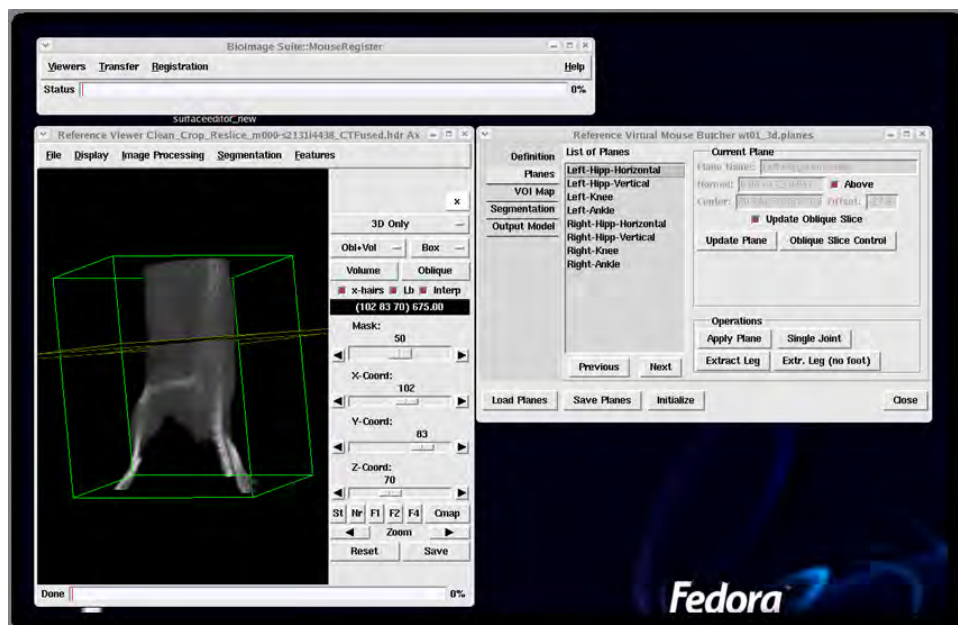


Figure 25.6: The “Planes” – tab of the “Virtual Butcher” control.

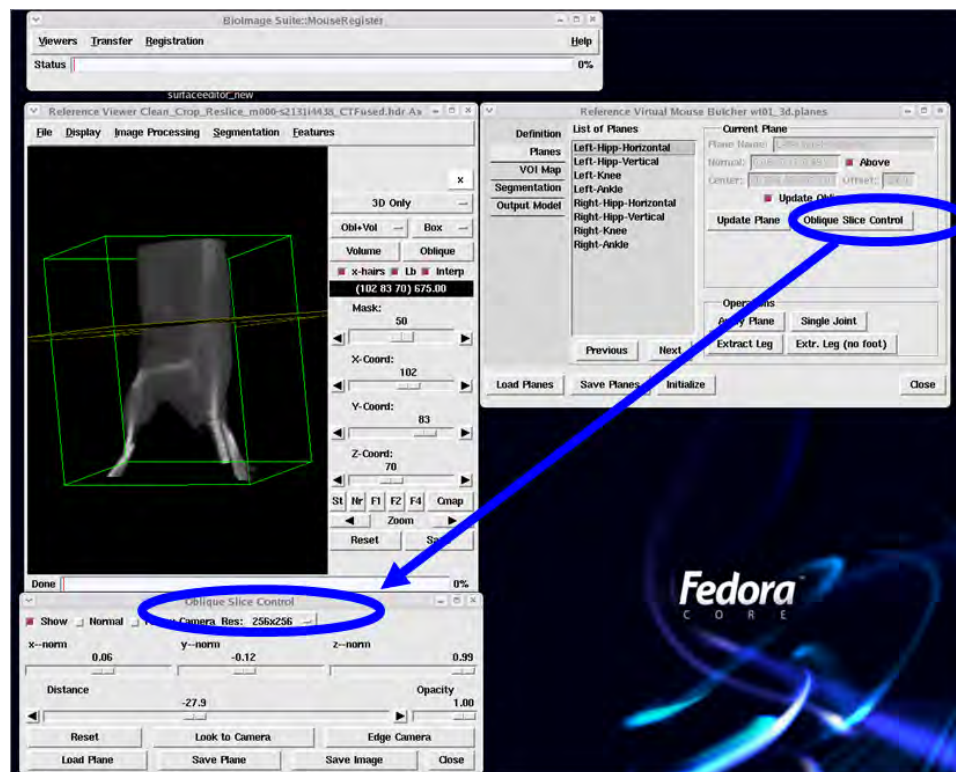


Figure 25.7: Parcellating the Mouse CT Image into Regions using Cropping Planes. This figure shows the three windows: (i) the viewer, (ii) the virtual butcher control and (iii) the oblique slice control

2. Click 3-Slice Mode from right hand menu by holding down the left mouse button, then choose the 3D Only button when the drop-down menu appears.
3. Click the 3-slice button, then choose the Obl + Vol option.
4. Next choose Features on the Reference Viewer menu. Select Mouse Chopper from the drop down menu. A Reference Virtual Mouse Butcher dialog appears – see Figure 25.5 The “Hind Limbs” and “Feet” checkboxes should be turned on by default.
5. On the “Virtual Mouse Butcher Dialog” select the Planes tab – see Figure 25.6 –, then Load Planes. A Load Planes box appears with a list of studies. Choose a study directory that has been previously segmented. Load planes from that study.  
Note: A menu of eight anatomical categories will be listed under the List of Planes.
6. Next click Save Planes, then rename file for this study. Be sure to save it in the proper directory.
7. Click Oblique Slice Control. An Oblique Slice Control box appears. Position this Oblique Slice Control box under the CT mouse image for better visualization of planes during this part of the segmentation process and stretch box to length of viewer window. Sliders located above the Distance will control the rotational and transitional position of the yellow transparent plane (x,y,z) – this is shown in Figure 25.7.

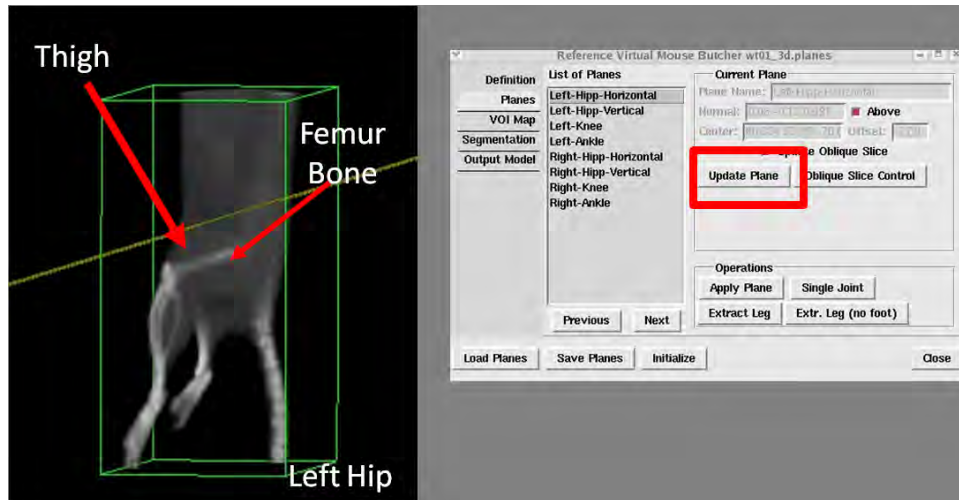


Figure 25.8: Placing the first plane to separate the hip. Once done press the “Update Plane” button on the butcher control to store the position and orientation of the plane.

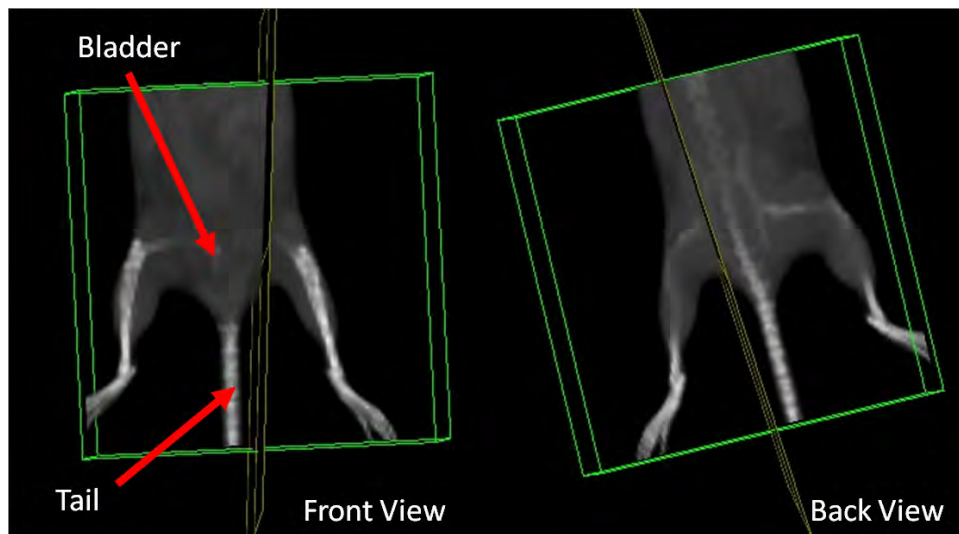


Figure 25.9: Placing the second plane to separate the hip.



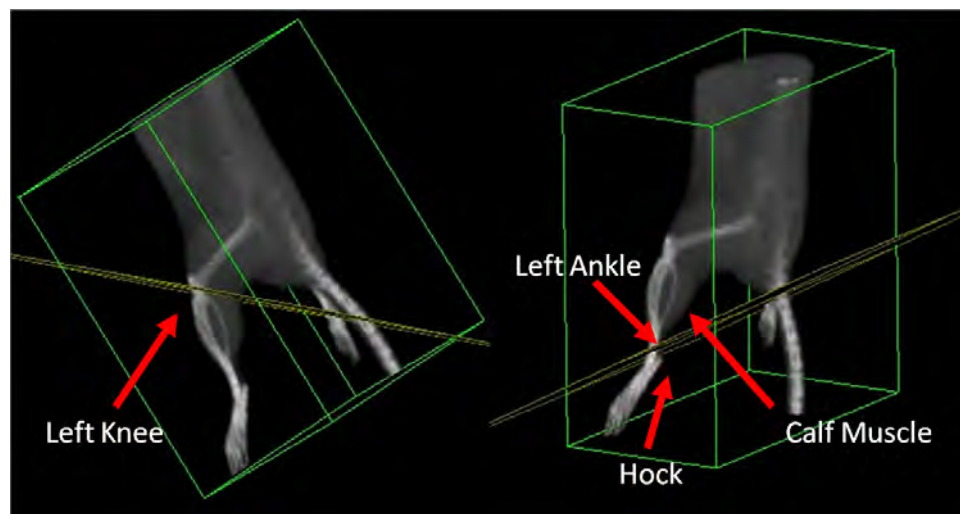


Figure 25.10: Placing single planes at the knee (left) and the ankle (right).

8. Start to adjust the first plane, Left Hipp-Horizontal, by moving the x,y,z sliders using your mouse. Position the yellow transparent plane parallel to the femur bone and above the thigh muscle – see Figure 25.8
9. Once optimal position is achieved, click Update Plane on the ReferenceVirtual Mouse Butcher window.
10. Select next anatomical category, Left Hipp-Vertical – see Figure 25.9. Align the yellow transparent plane parallel to pelvis in the space located between the head of the femur, and curve of pelvic bones. Check front and back views of the mouse by rotating the image using the left mouse button in the viewer window.  
 Note: The bladder and tail may introduce unwanted isotope activity into region of interest. Bladder may be visible on frontal view of mouse. Try to align plane so as to exclude bladder and tail, if possible. If not possible, see appendix on how to remove tail from CT image.
11. Once optimal position is achieved, click Update Plane on the Reference Virtual Mouse Butcher window.
12. Select next anatomical category, Left Knee, and align plane through point of kneecap on lateral view of mouse – see Figure 25.10 (left). Try to exclude body wall tissue before clicking Update Plane.
13. Select the next anatomical category, Left Ankle. Align the plane just above the protruding hock bone of foot – see Figure 25.10(right). Angle so as to exclude the calf muscle. Click Update Plane.
14. Repeat steps 8-13 for the Right side anatomy of mouse images. Click Update Plane after each category crop is finished.
15. Next click Save Planes, then click Save.
16. Prompt will ask if you want to overwrite. Choose OK.
17. Return to the ReferenceVirtual Mouse Butcher window.

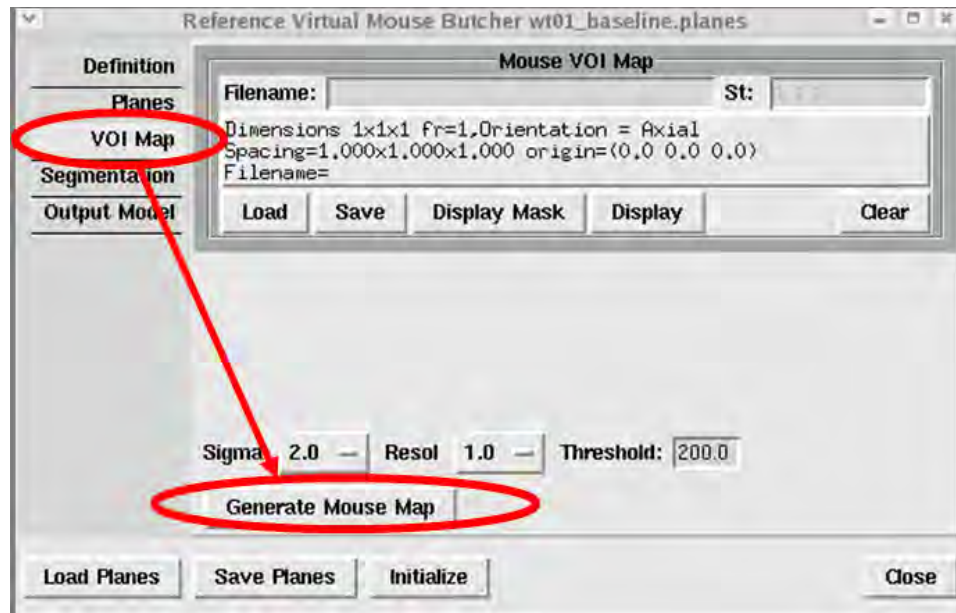


Figure 25.11: Generating the Mouse Map.

18. Choose VOI Map on the left menu by holding down the left mouse button.
19. Change the Resolution value to 1.0, then click on Generate Mouse Map – see Figure 25.11.  
Note: This step takes approximately 2-5 minutes.
20. When Generate Mouse Map is finished, a For Your Information box appears. Click OK.
21. A Reference Surfaces window will appear. Go to Edit, Paste – see Figure 25.12 (top).
22. The mouse image on the Reference Viewer window will be color coded into (7) anatomical regions (Torso, R & L foot, R & L lower leg, R & L upper leg/hip area) – see Figure 25.12.
23. Check that the Left and Right proportions are approximately equal, and that the colors do not repeat in any two regions, then click Do Not Show in the Reference Surfaces window.  
Note: The Do Not Show button alternates with the Show as Surface button, so you may need to click the Show as Surface button first.

## 25.6 Segment soft Tissue from CT Images

1. Return to the Reference Viewer(CT) window.
2. Click 3-D Only to change back to 3-slice Mode.
3. Choose Segmentation on the left menu of the Reference Virtual Mouse Butcher box using the left mouse button. Then choose Show Segmentation Control.
4. A Segmentation Control box appears. Choose Histogram Segmentation from the menu bar at left, if it does not come up as the default window.

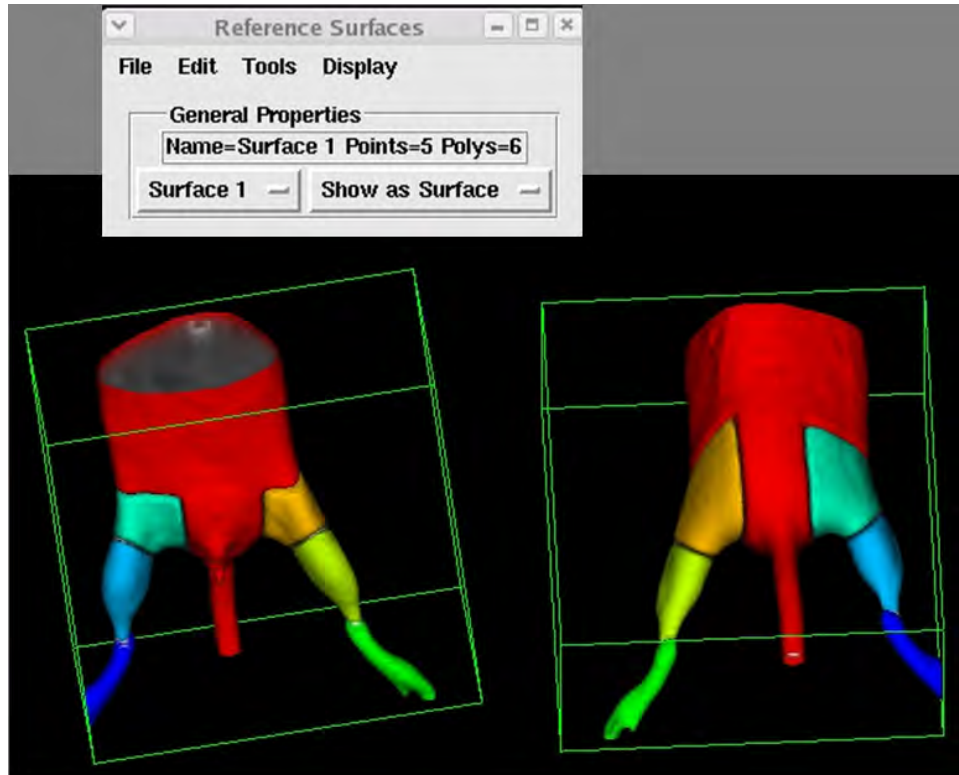


Figure 25.12: Surface Renderings of the Mouse Map.

5. Set the Classes field to 3. The Classes field is located at the top of the Segmentation Control box.
6. Click Segment, located toward the middle of this box.
7. Next choose Math Morphology located at the top of the menu bar at left.
8. Make sure the Display Mask is enabled (red).
9. Go back to Histogram Segmentation.
10. Set the Use Class to 2. The Use Class is located at the bottom of the Segmentation Control box.
11. Click Generate located to the right of the Use Class options.  
 Note: Mouse image on Reference Viewer will turn pink again. If pink areas do not Represent ONLY bone, then go back and change Classes to 4, then click Segment again, change Use Class to 3, click Generate again.
12. Go to the Mouse VOI Map box in the Reference Virtual Mouse Butcher and click Grab.
13. After using Grab, the Muscle Min and Muscle Max should be set to “1” and “1” and the Bone Class should be set to “2”.  
 Note: These numbers will be different if the Classes and Use Class numbers were increased.



14. Click the Soft Tissue VOI button to remove the bone from image.  
Note: If the image is not satisfactory, repeat process to change Class from 4 to 5 as described in step 11.
15. When image is acceptable, go to the Reference Viewer window and choose File, Save. Choose the correct Directory, and save the new file as ModelNoboneName of study.

## 25.7 Compute ROI Statistics

1. Close all the Mouse Butcher windows used.
2. Open the Transform Viewer window that was minimized in Step I.
3. Choose File, Load, then select ModelNoboneNname of study.
4. Go to the Reference Viewer window.
5. Choose File, Load, then select the 1st SPECT that was cropped in Step I. You will now have (3) windows open: The Reference Viewer, Transform Viewer, and a BioImage Suite:: Mouse Register window.
6. Choose Registration, then Transformation from the left drop down menu, located in the BioImage:: Suite Mouse Register window. A Registration/Overlay Tool box appears.
7. Click Compute ROI Stats.
8. A PXTkConsole window will appear. This box will list the ROI stats computed for the SPECT image. Highlight, cut, and paste these region values into a new text document and save them as: Animal name, Isotope, Day.
9. Repeat steps 5-8 for any other SPECT images.

Note: If regions 1 or 4 values are significantly greater than the other regions, this may indicate contamination due to inclusion of the bladder or tail.

## Part VIII

# Additional/Miscellaneous Topics

## Chapter 26

# File Formats

BioImage Suite uses a number of different file formats for storing different types of data such as images, surfaces, landmark collections, colormaps (palettes), setup files etc. While most users do not need to understand the internals of these formats, this page might be useful to advanced users who are trying to integrate BioImage Suite into their workflow and use it in conjunction with other software.

### 26.1 Images

BioImage Suite uses as its default file format the Mayo/Analyze 7.5 (and since 2.5 the NIFTI) file format. *BioImage Suite supports more or less the complete specification of the NIFTI format. We recommend using this in preference to the Analyze format if possible.*

#### 26.1.1 NIFTI Format

This is a modernized version of the Analyze format described below. It has the advantage that the axis orientation is explicitly specified in the image header (unlike the old Analyze standard in which this was implicit).

NIFTI images are either stored as single .nii (or .nii.gz compressed) files or as pairs of (.hdr/.img) files as in the Analyze format.

BioImage Suite supports almost the entire NIFTI standard with the exception of images with non-orthogonal axis. When an image is loaded into BioImage Suite it may be flipped to conform to our internal convention (which is implicitly assumed in the case of Analyze images – more below). For example, if an image comes in left to right, it will be flipped to be right-to-left. The header is also changed to reflect this – the image is still a valid, appropriately labeled NIFTI image!

### 26.1.2 Analyze 7.5 format

In the case of the Analyze 7.5 format, while there are many other fields in the header, BioImage Suite relies on the following:

- Image Dimensions: x,y,z,t
- Voxel Dimensions: x,y,z (all assumed to be positive!)
- Orientation: 0=axial, 1=coronal, 2=sagittal, 3=polar (ultrasound,local extension)
- Data type, unsigned char, short, int ,float

The assumption made in BioImage Suite for the purpose of displaying the images is that the images are stored as follows:

- Axial: x=right to left, y=anterior to posterior, z=inferior to superior
- Coronal: x=right to left, y=superior to inferior, z=anterior to posterior
- Sagittal: x=anterior to posterior, y=inferior to superior, z=left to right (this is for compatibility with the most common local acquisition which results in a non right-handed coordinate system – we suggest that Sagittal images be resliced to Axial before any serious processing)

Note: In particular, we do not use the origin field (SPM extension).

The File Header can be viewed/edited using Header Editor, under the File Menu. Images be reoriented using the Image Processing tools.

BioImage Suite can import a variety of other formats, under File/Image Import. However, these must be converted to Analyze/NIFTI format for use in the batch mode processing tools etc. DICOM import is limited to reading images in which the whole series is placed in its own directory in which there are no other files.

## 26.2 Surfaces

There a number of surface formats used in BioImage Suite. The most common are:

1. The b-spline stack surface format.sur – this is format used by the Surface Editor to store a surface as a stack of b-splines one per slice.
2. The triangulated stack surface format .tstack – this is primarily used by the cardiac tools to store triangulated surfaces and additional point attributes such as normals and curvature. It can also specify whether the surface is made up of a collection of planar contours and define these explicitly which is necessary for some of the cardiac mesh generation tools.

3. Standard VTK formats, in particular the one defined by its implementation in `vtkPolyDataReader` and `vtkPolyDataWriter`. BioImage Suite stores such surfaces using a `.vtk` extension. The BioImage Suite Surface Control use the `.vtk` file format is its primary format.

### The `.sur` format

The first part of the file is the header. The first line must be reproduced exactly. The next four lines define the overall extend of the spline stack, each number being proceeded by the “#” sign. The second line (139) represents the maximum number of splines which is defined by the number of slices in the underlying image. The third line (#0) can take the values 0 or 1 depending on whether the user has explicitly defined the apex in the case left ventricular surfaces. The fourth and fifth lines (#64 and #66) represent the actual extent of the surface, in this case there are splines for slices 64 to 66 (i.e. a total of 3 splines). The last line of the header defines the coordinates of the apex in mm (again for cardiac applications only)

```
#BFPSplineStackApex File , numsplines, has apex,trim_bottom,trim_top
#139
#0
#64
#66
74.40 93.60 -0.50
```

After the header, the file stores each spline in order. In this file there are 3 splines. Each spline consists of its own header beginning with the line that starts with “#FPBSpline”. The next line (#2) is the dimension of the spline, in this case the 2 signifies that this is a planar curve (which is always the case when the spline is part of a surface). The next number (#74.87) represents the z-coordinate of the spline in mm (or the same units as the voxel dimensions of the underlying image). The center of the bottom slice is at  $z=0.0$ . The next number is degree of the b-spline, in this case it is cubic (#3). Next there is the flag indicating whether the spline is closed (#1) or open, it is always closed in this format. Finally, the last item lists the number of unique control points which in this case is 6.

Following the divider line “#-----”, the next line lists the knot positions. BioImage Suite can use non-uniform splines. After that, the positions of the control points are listed. Since this is a closed spline, there are three control points which repeat, to yield the periodic constraint.

```
#-----
#FPBSpline File header= dimension,zlevel,degree,closed,numpoints
```

```

#2
#74.879997
#3
#1
#6
#-----
#-0.50000 -0.33333 -0.16667 0.00000 0.16667 0.33333 0.50000 0.66667 0.83333 1.00000 1.16667 1.33333
 86.86 68.37
 95.78 93.25
 87.11 119.31
 64.55 124.11
 53.84 96.50
 58.25 72.18
 86.86 68.37
 95.78 93.25
 87.11 119.31

```

Following the first spline, there are the rest of the splines. Two more are given here to complete this file.

```

#FPBSpline File header= dimension,zlevel,degree,closed,numpoints
#2
#76.049995
#3
#1
#12
#-----
#-0.44684 -0.29066 -0.13710 0.00000 0.04604 0.08276 0.16831 0.26755 0.33929 0.36480 0.37926 0.40480
#0.55316 0.70934 0.86290 1.00000 1.04604 1.08276 1.16831
 89.49 78.72
 95.52 91.12
 92.51 100.27
 85.26 102.16
 78.19 93.16
 68.64 93.45
 61.40 100.72
 57.08 101.01
 54.08 99.36
 55.15 90.30
 57.36 77.48
 73.49 64.66
 89.49 78.72

```

95.52 91.12  
92.51 100.27

**Note:** For those who are looking at the code, this file format can be read/saved using the `vtkpxSplineStackSource` class in the Legacy directory.

### The `.tstack` format

The first part of the file is again the header, with the usual caveat that the first line must be reproduced exactly. The next few lines list, in order, the number of planar contours in the stack (11 in this case), the number of points (1221) and the number of triangles (2220). The next item in the header is the maximum number of (first-order) neighbors that any-given point in this `tstack` has (6). The final two flags identify whether points in this file have associated valid normals (1) and curvatures (0). In this specific file the points do have normals, but the principal curvatures have not been computed, hence their values are set to zero.

```
#TriangulatedStack File,numcontours,numpoints,numtriangles,maxconnection,hasnormals,hascurvature
#3
#314
#416
#21
#1
#0
```

The next section contains the index of the first point and z-coordinate for each of the planar contours. For example, the third contour starts at point 222 and has z-coordinate 77.220. (This example is of a cylinder surface, hence the regularity!)

```
#-----Contour Indices-----
0      74.880
111    76.050
222    77.220
333    78.390
444    79.560
555    80.730
```



```

666    81.900
777    83.070
888    84.240
999    85.410
1110   86.580

```

Next, each point is listed (we list the first 4 here). Each point is represented by nine items, (i) the index, (ii) the x,y,z coordinates, (iii) the x,y and z components of the surface normal and (iv) the first and second principal curvature of the surface at this point.

```

#-----Points-----
  0  93.000  93.637  74.880  1.000  0.010  0.000  0.000000  0.000000
  1  92.977  94.808  74.880  0.999  0.044  0.000  0.000000  0.000000
  2  92.908  96.012  74.880  0.996  0.083  0.000  0.000000  0.000000
  3  92.792  97.206  74.880  0.992  0.123  0.000  0.000000  0.000000

```

Next, the triangles are listed. Each triangle is represented by four numbers, (i) the index, (ii) the indices of the three points that make up the triangle.

```

#-----Triangles-----
  0    0    112    111
  1    0     1    112
  2    1    113    112
  3    1     2    113

```

Finally, for the purpose of fast searching, the neighbors of each point are listed. Each line consists of (i) the index of the specific point, (ii) the number of neighbors it has, and (iii) the indices of its neighbors.

```

#-----Neighbors-----
0      4 112    111    1    110

```

---

```

1      4 0      112    113    2
2      4 1      113    114    3
...
424    6 312    423    313    425    535    536
425    6 313    424    314    426    536    537

```

**Note:** For those who are looking at the code, this file format can be read/saved using the `vtkpx-TriangulatedStack` class in the Legacy directory. The `vtkpxTStackReader` class can also be used to read this class.

### The .vtk format

The interested reader is referred to the documentation of the Visualization Toolkit for more details on this format.

## 26.3 Landmark Collections

This format stores collections of points and is primarily used by the Landmark Control. The points (landmarks) are stored in **voxel coordinates** with the information above them in the header giving the position of the (0,0,0) voxel and the dimensions of each voxel (1.2,1.17,1.17). The first line “#Landmark Data”, must be reproduced exactly. The other part of the header (line 3) lists the number of points (4 in this case) and the Mode. The Mode flag is 2 if this represents a closed curve or 1 otherwise.

```

#Landmark Data
#Number of Landmarks Mode
4 1
#Origin
  0.000  0.000  0.000
#Spacing
  1.200  1.170  1.170
#Landmarks
  62.00  48.77  97.34
  62.00  88.95  86.71
  62.65  68.87  87.00
  33.47  69.00  79.26

```

## 26.4 Transformations

There are a number of transformation file formats in BioImage Suite. The two most important ones are the linear transformations (.matr) and the non-linear transformations (.grd).

### The .matr file format

This is simply a 4x4 matrix stored in a text file with no additional information, e.g.

```
0.996  0.059  -0.061  -16.756
-0.054  0.994  0.094  -0.724
0.066  -0.091  0.994  -20.989
0.000  0.000  0.000  1.000
```

### The .grd nonlinear transformation file format

The .grd file contains a combined linear and non-linear transformation. The first line is “#vtkpxNewComboTransform File” and must be reproduced exactly. The third line defines whether in transforming a point, the non linear transformation is applied first, when combining the two transformations. Next is the linear transformation, which is the usual 4x4 matrix.

```
#vtkpxNewComboTransform File
#NonLinearFirst
1
#Linear Component
0.997  0.072  -0.064  -4.009
-0.070  0.980  -0.114  21.625
0.049  0.139  0.987  -63.856
0.000  0.000  0.000  1.000
```

After the specification of the linear transformation, the file contains the specification of the non-linear component. This is a tensor spline. The first term (Origin) is the position of the first control point (in the corner of the tensor grid). Next follows the spacing of the tensor grid in x,y and z directions respectively. Following this, the dimensions of the grid (15x15x12) are specified. The final header term is the interpolation mode. This can take values: 0=nearest neighbor, 1=trilinear, 3=tricubic, and 4=tensor b-spline – the last one is the most common. After the header, the displacements of each control point are given, each line list the index of the control point followed

by its displacement.

```
#vtkpxBaseGridTransform2 File
#Origin
  -15.4000 -16.0273 -63.8557
#Spacing
  19.8390 20.1219 20.2082
#Dimensions
  15 15 12
#Interpolation Mode
4
#Displacements
0 -0.0351 -0.0093 0.0486
1 -0.0394 -0.0048 0.0240
2 -0.0439 0.0023 -0.0098
```

**Note:** For those who are looking at the code, this file format can be read/saved using the `vtkpx-ComboTransform` class in the Registration directory. A generic transformation reader for all the file formats in BioImage Suite is available in the `vtkpxTransformationUtil` class in the same directory &ndash; look at the method `LoadAbstractTransform`.

## 26.5 Colormaps

Colormaps are used to map a scalar intensity value into RGBA space. They are typically implemented as lookup tables. The Colormap Editor supports two colormap formats, the native “**.cmap**” format, and the Analyze “**.lkup**” format. These are described next.

### The **.cmap** colormap file format

This consists of a short header that defines (i) the number of colors (256 in this case) and the input range (0.0 to 255.0 in this case). Much like the rest of the custom file formats, the first line must be reproduced exactly. If the intensity to be mapped has value below the lower range level (e.g. anything less than 0.000 in this case) then it gets mapped to the same intensity as the lower range level. Similarly, if it has value above the upper level (e.g. anything greater than 255.0), it will get mapped to the same intensity as the upper range level.

```
#IPAGRGBAColormap
#Number of Colors
```

```

256
#Range
0.000 255.000
#Colors
#Index R G B Alpha

```

Following the header, the file includes the RGBA mapping for each level. Each line has 5 elements: (i) the index, (ii) the RGBA value, each value being in the range of 0.0 to 1.0.

```

    0 0.000 0.000 0.000 0.502
    1 0.004 0.004 0.004 1.000
    2 0.008 0.008 0.008 1.000
    3 0.012 0.012 0.012 1.000
    4 0.016 0.016 0.016 1.000
....
251 1.000 1.000 0.529 1.000
252 1.000 1.000 0.557 1.000
253 1.000 1.000 0.584 1.000
254 1.000 1.000 0.612 1.000
255 1.000 1.000 0.639 1.000

```

### The Analyze .lkup colormap file format

This is an extremely simple format, retained for compatibility reasons. It consists of exactly 256 lines, each of which has an RGB (note: no Alpha/Transparency Channel) value, where the range is between 0 and 255. For example, for the same colormap above, this has the form:

```

    0  0  0
    1  1  1
    2  2  2
    3  3  3
    4  4  4
...
255 255 135
255 255 142
255 255 149
255 255 156
255 255 163

```

## 26.6 Setup Files

Different components of BioImage Suite have their own setup files for storing complex information, such as lists of filenames and associated transformations etc. While, some of these are easy to read, the only one that users should consider editing in a text editor is the Multisubject Control setup file which is described in detailed in the Multisubject manual page.

## Chapter 27

# Command Line/Batch Mode Tools

### 27.1 Introduction

BioImage Suite has a number of Command Line Utilities that are useful both for batch-mode computation and for the integration of BioImage Suite tools into other software. In particular these tools can be divided into:

- Inspecting analyze image headers and surface files.
- Image processing such as smoothing, thresholding, flipping etc.
- Segmentation (voxel classification), slice inhomogeneity correction & bias field correction tools.
- Image Reslicing and other utilities to manipulate transformations.
- Batch mode registration tools.
- Batch mode segmentation tools.

For all command line tools to function the path needs to be appropriately set at the command line. On Microsoft Windows this is accomplished by first calling `c:\yale\bioimagesuite\setpaths.bat`, where as on Unix by the sourcing of one of the following shell scripts, either `/usr/local/bioimagesuite/setpaths.csh` or `/usr/local/bioimagesuite/setpaths.sh` depending on the shell used. See the Running BioImage Suite page for more details.

Typing any of the `bioimagesuite` command names with no arguments prints a short description of its usage. Arguments in square brackets are optional.

**Note:** Most of this functionality (with the exception of the batch-mode tools) is available via the graphical user interface see: the Image Processing, (ii) Segmentation and Registration Chapters respectively.

## 27.2 Inspecting Analyze Header Files and Surface Files (.vtk)

The `pxmat_headerinfo.tcl` command can be used to print basic information about analyze header files (.hdr). The syntax is:

```
pxmat_headerinfo.tcl image1 image2 .. imagen
```

This can be used to quickly scan the contents of a directory e.g. `pxmat_headerinfo.tcl *.hdr` A similar script for examining surface headers `pxmat_surfaceinfo.tcl` is also available.

## 27.3 Image Processing Tasks

**Flipping Images** Sometimes images need to be flipped about one of the coordinate axes. This can be accomplished using the `pxmat_flip.tcl` script. The syntax is

```
pxmat_flip.tcl direction file1 file2 file3 ..  
                Direction 0=x, 1=y, 2=z
```

**Reorienting** Reorienting is defined here as switching the orientation of an image from, for example, axial to coronal. This is accomplished by the `pxmat_reorientimage.tcl` command. The syntax is

```
pxmat_reorientimage.tcl input_image output_image output_orientation  
                        output_orientation = 0 axial 1 coronal 2 sagittal
```

The input orientation is detected automatically.

**Smoothing** This is accomplished by the `pxmat_smoothimage.tcl` command. The syntax is



```
pxmat_smoothimage.tcl kernel_size input_image1 input_image2 ...
```

where the smoothing kernel\_size in mm represents FWHM filter size. Multiple images may be specified on the command line.

**Thresholding** Similarly this is performed by the pxmat\_thresholdimage.tcl command. The syntax is:

```
pxmat_thresholdimage.tcl threshold input_image1 input_image2 ...
```

The output is a set of binary images with value = 1 where the original images had values above “threshold” and zero elsewhere. Multiple images may be specified on the command line.

**Resampling Images:** This is accomplished using the pxmat\_resampleimage.tcl command. The syntax is:

```
pxmat_resampleimage.tcl input_image output_image blur_mode \  
voxelsize1 [ voxelsize2 ] [ voxelsize3 ]
```

where blurmode = 0 or 1, setting this to one blurs the image with an appropriate Gaussian filter. The output dimensions of the voxels in mm are set using the voxelsize1, voxelsize2 and voxelsize3 parameters (if only one specified then an isotropic image is generated). (voxelsize1 = voxel size x, voxelsize2=voxel size y, voxelsize3=voxel size z).

**Combining Multiple 3D Images into a single 4D Image** This is accomplished using the pxmat\_create4dimage.tcl command. The syntax is:

```
pxmat_create4dimage.tcl image1 image2 .. imagen  
output = image1_4D.hdr
```

This is particularly useful for generating appropriate input images for the Diffusion Tools.

**Extracting 3D Image volumes from a single 4D Image** This is accomplished using the `pxmat_split4dimage.tcl` command. The syntax is:

```
pxmat_split4dimage.tcl image [ begin =1 ] [ end = maximum number of frames ]
```

The output images are saved in filenames `image_001.hdr`, `image_002.hdr` etc.

## 27.4 Segmentation and Bias Field Correction Tools

**Segmentation Tools** A single script `pxmat_segment.tcl` captures some of the functionality from the < a tools (Histogram & Region Segmentation.) The reader is referred to that page for more details. The syntax for `pxmat_segment.tcl` is:

```
pxmat_segment.tcl input numclasses [ smoothness = 1.0 ] \  
    [ maxsigmaratio = 0.05 ] [ mrfiterations = 20 ] \  
    [ outputname = "" ]
```

```
Number of classes = number of tissue labels  
    (background is a tissue label so add this)  
Smoothness = spatial smoothness, is set to zero --  
    a faster algorithm is used  
MaxSigmaRatio = this constrains the ratio of the max  
    standard deviation to the min standard deviation  
If Output Name is blank defaults are either  
    input_quicksegm (smoothness=0.0) or input_segm
```

If smoothness is set to zero then the “histogram”-segmentation method is used, otherwise the region-segmentation method is applied. The `maxsigmaratio` parameter can be useful is one class has a relatively small number of voxels.

**Slice Inhomogeneity correction & Bias Field Correction tools.** `pxmat_biasfield.tcl` is a multifunctional script for performing both these tasks. The syntax is:

```
pxmat_biasfield.tcl input [ mode = 5 ] [ threshold =0.05 ] [ numclasses =
4 ]\
  [ res=3 ] [ maxsigmaratio = 0.5 ] [ minb1 =0.8 ] [ maxb1=1.25 ]
```

```
mode = 0 (slice homogeneity) 1 (triple slice homogeneity)
      2 (quadratic polynomial) 3 (cubic polynomial)
      4 = triple slice + quadratic, 5=triple slice + cubic.
Threshold = 0.05, normalized threshold (0.1) to exclude voxels from
      bias field estimation -- to eliminate background
Number of classes = number of tissue labels
      (background is a tissue label so add this) for polynomial
Resolution = 3, resolution sampling for polynomial estimation
MaxSigmaRatio = this constrains the ratio of the max standard
      deviation to the min standard deviation
Min B1 and Max B1 specify the range of B1 field allowed
```

**Slice Inhomogeneity correction** This is performed simply using `mode = 0` in `pxmat_biasfield.tcl`.

**Bias Field correction** There are 2 methods here, which can be used in combination:

1. Multiple (triple slice) orientation slice inhomogeneity correction – set `mode = 1` in `pxmat_biasfield.tcl`.
2. Polynomial bias field correction using the method of Styner et al (TMI 2000) – set `mode = 2` (quadratic ) or `3` (cubic) in `pxmat_biasfield.tcl`.

The second method requires a crude histogram segmentation hence the other parameters. This invokes the histogram segmentation (i.e. `smoothness=0`) method from the previous section.

The two bias field correction methods may be combined by setting the mode to 4 or 5 respectively. In general, the triple-slice method can be used to get a good initial starting point for the polynomial method in cases where the bias field change is large.

In addition to the segmentation parameters, the parameters **threshold** , **res** , **minb1** and **maxb1** or only used by the polynomial method.

## 27.5 Reslicing Images and other command line tool using transformations

The BioImage Suite registration tools (both command line and using the graphical user interface), produce transformation files as outputs, typically with a “.matr” or a “.grd” extensions. (See the File Formats Chapter for more details.) The command line tools described here can be used to apply these transformations to images (reslicing) or to manipulate the transformations in different ways.

**Image Reslicing** Given a reference image, a target image and an appropriate transformation (or set of transformations) the target image can be resliced in the space of the reference image using the command

```
pxmat_resliceimage.tcl reference_image target_image output_image
    interpolation_mode xform [xform2] [ xform3]

interpolation_mode = 0,1,3 none,linear,cubic
    (Avoid linear if the images are in the range 0.1)
xform = the .matr or .grd file
```

**Note:** The direction of the transformations is a common source of confusion. When computing registrations, the estimated transformations is **FROM** the reference **TO** the target. This transformation can be used in pmat\_reslice.tcl to move the target image to the reference image. A good rule of thumb is to remember that images move in the opposite direction as transformations.

If multiple transformations are specified (upto 3) then the concatenation of these transformations will be used. For example, consider the case where we have:

- A transformation from a 3D reference brain to a 3D individual anatomical. **xf1.grd**
- A transformation from a 3D individual anatomical to a scout image. **xf2.matr**
- A transformation from the scout image to a functional acquisition. **xf3.matr**

The following command will reslice the functional image to the space of the 3D reference brain:

```
pxmat_resliceimage.tcl ref_image func_image resliced 3 xf1.grd xf2.matr xf3.matr
```

**Inverting Transformations** Often the inverse transformation is required. This can be accomplished using the `pxmat_inverttransform.tcl` script. The syntax is:

```
pxmat_inverttransform.tcl reference_image output_transformation xform1  
[ xform2 ] [ xform3 ]
```

If multiple transformations are specified (upto 3) then the inverse will be the inverse of the concatenation of these transformations.

**Computing Displacement Fields** Sometimes it is desired to get the displacement field for each voxel. This can be accomplished using the `pxmat_displacementfield.tcl` command. The syntax is:

```
pxmat_displacementfield.tcl reference_image output_field  
xform1 [ xform2 ] [ xform3 ]
```

The `output_field` will be a 4D analyze image file with three frames, storing the x, y and z displacement of each voxel in mm (or nominal units). If more than one transformation is specified the final displacement field will be a result of the concatenation of the transformations specified.

## 27.6 Batch Mode Registration

**A quick overview of GNU Make.** All Batch Job Generators in BioImage Suite result in a makefile which needs to be processed with the Unix Make Utility (most commonly GNU Make). While “makefiles” are primarily used to compile large programs they are a great mechanism for general batch mode processing because they enable:

1. the definition of job dependencies – i.e. job A must be completed before job B
2. the use of multiple processors at the same time – i.e. run two or more jobs at once.
3. the ability to automatically recover from a system crash and to restart at the point that the job failed – i.e. no repeat computations.

The standard name for a makefile is unsurprisingly “makefile”, although in this context I recommend the use of more descriptive names e.g. “makefile.controls” etc. Given a makefile, the batch job is executed using

```
make -f makefile.controls
```

On Linux systems `make=gmake` i.e. typing `gmake` is equivalent to typing `make` – this may appear in some examples. Additional useful flags include

1. “-n” - do a dry run i.e. simply print a list of commands to be executed without doing anything

```
make -n -f makefile.controls
```

2. “-j” - specify how many jobs to run at once – typically equal to the number of processors available e.g. to use 2 processors type

```
make -j2 -f makefile.controls
```

In addition makefiles contain a number of “jobs” which may be explicitly specified. Typically the first job defined in BioImage Suite batch jobs is the equivalent of “do everything” so you need not worry about this - in some cases other specifications will need to be given . Note: If after starting a batch job – it is for some reason terminated (either by a system crash or a reboot or ..) it may be restarted by typing exactly the same command as the one used to originally start the batch job. The job dependency mechanism in `make` will ensure that no processing that was previously completed is re-run.

**Microsoft Windows Info** Whereas `make` is a standard feature on most Unix system, to run batch jobs in MS-Windows you will to download and install GNU Make. A binary (from the UnixUtils distribution ) is available for download at this location (source code here. Place `make.exe` in a directory in your path such as `C:\yale\vtk44.yale\bin`.

**Single Reference Image Registrations** To compute batch-mode registrations to a single reference (typically inter-subject nonlinear brain mappings for the purpose of computing fMRI composite maps etc.) BioImage Suite provides the `pxmultiregister_int.tcl` tool. This is a batch-mode generator which will generate a makefile - the batch job itself is executed using the standard `make` program - see detailed description above. `pxmultiregister_int.tcl` has a large number of options – with more to be added in the next release. Simply typing `pxmultiregister_int.tcl sample` lists all options, which might be intimidating for the average user. A basic setup file has the form: (Lines beginning with `#` are treated as comments and may be added liberally to document the processing.

```
#Example Setup File
# all lines beginning with # are ignored
#
#
# Mode of Operation
set intensities_only 1
```

```
#
#
#
# List all images here
#
set imglist {
  Az2_2_sm0.8.hdr
  Az2_3_sm0.8.hdr
  Az2_4_sm0.8.hdr
}
#
# Put reference brain here
#
set refimg Az2_5_sm0.8.hdr

# Linear mode -- choices are "rigid", "similarity", "affine" --
# this is used to generate an initial transformation to be refined later.
set linearmode "affine"

# Tweak parameters for intensity based part
# (the ones below are typical for fMRI composite maps)

# Resolution is a scale factor x native
#(i.e. if reference image is 1.1x1.1x1.2 setting this to 1.5
# will resample the images to
# 1.65x1.65x1.7 mm). Integer values are not recommended due to
# artifacts in joint histogram computation, 1.5 or 1.2 are good values
set resolution 1.5

# Spacing defines the flexibility of the transformation --
# the gap between control points in the tensor b-spline model
# used for the non-linear transformation
# 15.0 mm is a good choice for composite functional maps
# For structural morphometric studies this should be reduced to
# 10 or 8 mm (with a corresponding increase in computational time)
# If the data is a fairly low resolution this
# can be increased to 20mm or so.
set spacing 15.0

# This is the maximum number of iterations for the
# Conjugate Gradient Minimizer (15 is usually a safe number)
set iterations 15

# Leave this to zero unless other instructed
# Regarding filetype: filetype = 1 includes directory name
# in output filename 0 does not
set filenamestype 0
```

```
# Set this higher if data quality is low
set smoothness 0.001

# If linearonly=1 then the resulting transformation will be linear
#           i.e. no warping will be attempted
set linearonly 0
```

Once the setup file is completed, the next step is to decide where the results (and a host of log files) will be stored. Typically this is in a subdirectory (e.g. results). If the setupfile is called controls\_setup and we want the results to go to controls\_results type:

```
pxmultiregister_int.tcl controls_setup control_results
```

This will check for the existence of all files in the setup file, if a file (image) is missing an error will be given. Once this completes OK, the next step is to generate the makefile using

```
pxmultiregister_int.tcl controls_setup control_results go > makefile.controls
```

At this point the batch-job is ready and can be started using the make utility described above.

**Pairwise Image Registrations** The case of interest here is where one set of images (e.g. thick-slice “2D” conventional images) are to be mapped to another set of images (e.g. high resolution 3D anatomical images). To accomplish this use the pxpairwiseregister.tcl script. This is a batch-mode generator which will generate a makefile – the batch job itself is executed using the standard make program – see above for instructions as to how to use GNU make. pxpairwiseregister.tcl has a large number of options. Simply typing pxpairwiseregiste.tcl sample lists all options, which might be intimidating for the average user. A basic setup file has the form: (Lines beginning with # are treated as comments and may be added liberally to document the processing.

```
#Example Setup File
# all lines beginning with # are ignored
#
#
#
```



```
# Put reference list here
#
set refflist {
  /data1/brains/1001/refA.hdr
  /data1/brains/1002/refB.hdr
}
#
# Put the target list here
#
set trglislist {
  /data1/brains/1001/targA.hdr
  /data1/brains/1002/targB.hdr
}
#
#
# Type of registration
# mode = rigid,affine,nonlinear
set mode "nonlinear"
#
# Tweak parameters
# filetype = 1 includes directory name in output filename 0 does not
# defaults are for rigid/affine/nonlinear
set resolution 1.5
set spacing 15.0
set iterations 15
set stepsize 1
set smoothness auto
```

See the previous section for more description of the parameters. Executing `pxpairwiseregister.tcl` is also identical to `pxmultiregister_int.tcl`, i.e. first

```
pxpairwiseregister.tcl setup.txt results
```

then

```
pxpairwiseregister.tcl setup.txt results go > makefile.results
```

and then use the make utility to start the makefile.

**Batch Mode Registration Utilities** The following helper scripts are used to assist/compute the batch mode registrations. They are *not meant to be invoked directly*, so only a cursory description is given. Invoking each script with no arguments gives a brief description of its syntax.

1. pxmat\_register.tcl – computes intensity based registrations.
2. pxmat\_distortioncorrection.tcl – computes intensity based distortion correction for Echoplanar images.
3. pxmat\_pointregister.tcl – computes point based registrations.
4. pxmat\_integratedregistration.tcl – computes integrated intensity + point based registrations.
5. pxmat\_computeoverlap.tcl – computes the overlap of pre-segmented structures after registration – this is a measure of registration quality.
6. pxmat\_computesimilarity.tcl – computes the similarity of two images after registration.
7. pxmat\_computedistance.tcl – computes the distance of two surfaces after registration.

## 27.7 Batch Mode Segmentation

To compute batch-mode segmentation, BioImage Suite provides the pxmultisegment.tcl tool. This is a batch-mode generator which will generate a makefile - the batch job itself is executed using the standard make program - see detailed description above. Simply typing pxmultisegment.tcl sample lists all options. A basic setup file has the form: (Lines beginning with # are treated as comments and may be added liberally to document the processing.

```
#Example Setup File
#Example Setup File
# all lines beginning with # are ignored
#
#
# Mode of Operation
#
#
# List all images here
#
set imglist {
  4T.hdr
  1147acpc.hdr
}

#
#
```

```
# Mode
# Set this to 0 to use native BioImage Suite tools
set usefsl 0
set numclasses 3
set identifier segm
#
#
# FSL parameters
set imgtype 1
#
#
# BioImage Suite parameters
set smoothness 0
set maxsigmaratio 0.05
set mriterations 20
```

If `usefsl=1` then the batch mode segmentation will leverage the FSL/FAST Gray/White Segmentation tools, otherwise the native BioImage Suite tools are employed. Once the setup file is completed, the next step is to decide where the results (and a host of log files) will be stored. Typically this is in a subdirectory (e.g. results). If the setupfile is called `controls_setup` and we want the results to go to `controls_results` type:

```
pxmultisegment.tcl controls_setup control_results
```

This will check for the existence of all files in the setup file, if a file (image) is missing an error will be given. Once this completes OK, the next step is to generate the makefile using

```
pxmultisegment.tcl controls_setup control_results go > makefile.controls
```

At this point the batch-job is ready and can be started using the make utility described above.

**Part IX**

**Appendices**

## Appendix A

# Installing Other Software

BioImage Suite can take advantage of the presence of other software. If these are not installed some functionality will not be present (e.g. brain skull stripping which requires FSL) but the rest of BioImage Suite will function without any problems.

### A.1 Installing and Configuring FSL

FSL is a software package developed at the Oxford Centre for Functional Magnetic Resonance Imaging of the Brain, It has a lot of complementary functionality to BioImage Suite and as described above BioImage Suite can invoke cleanly two modules in FSL – the Brain Extractor Tool and FAST Gray/White Segmentation tools respectively. The appropriate graphical user interfaces will only appear in the Segmentation tools if FSL is detected in your path. This involves a two step process:

1. Downloading and installing FSL – follow the instructions at the FSL webpage.
  - Edit the configuration files `fslconf.sh` and `fslconf.csh` (located in `$FSLDIR/etc/fslconf`) to set the default file format to `ANALYZE`.
2. Configuring BioImage Suite to let it know where FSL is installed.
  - On Linux/Mac OS X/Unix operating systems, if FSL is installed in its default location `/usr/local/fsl` there should be no additional configuration needed. Otherwise edit the files `/usr/local/bioimagesuite/setpaths.sh` and `/usr/local/bioimagesuite/setpaths.csh` to set `FSLDIR` appropriately. The relevant lines are (in `setpaths.csh`)
 

```
setenv FSLDIR /usr/local/fsl setenv FSLOUTPUTTYPE ANALYZE
```
  - On Windows ...
    - FSL uses the cygwin emulation layer. You can either download this from the cygwin web page or use the version linked to from the FSL. webpage.

- Next install FSL as usual
- Then edit the file `c:\yale\bioimagesuite\setpaths.bat` to set both `FSLDIR` and `CYGWINDIR`. `FSLDIR` is the location of FSL in the unix-like path setup used by cygwin. `CYGWINDIR` is the location of your CYGWIN installation, typically `C:\CYGWIN`. The relevant portion of `setpaths.bat` is:

```
@set FSLDIR=/usr/local/fsl
@set CYGWINDIR=c:/cygwin
@set WFSLDIR=%CYGWINDIR%/FSLDIR%
@set FSLOUTPUTTYPE=ANALYZE
```

## A.2 Installing the WFU Pick Atlas

The BioImage Suite uses the atlases that come with the WFU Pick Atlas Tool - this needs to be downloaded separately from [www.fmri.wfumbc.edu](http://www.fmri.wfumbc.edu). The easiest way to make this work with BioImage Suite is to simply uncompress the Pick Atlas software in the parent directory of BioImage Suite (e.g. `c:\yale` or `/usr/local`). Alternatively, you may set the value of the environment variable `WFUATLASDIR` to point to the location of this software.

## Appendix B

# Compiling BioImage Suite

**BioImage Suite is an application not a toolkit. Compiling all of BioImage Suite including all its prerequisites is not for the inexperienced or faint-hearted. We suggest using the binary versions whenever possible.**

### B.1 Overview

Installing BioImage Suite consists of four parts.

1. Installation of a pre-compiled binary `itk241_distribution` containing the Insight Toolkit (ITK) v 2.4.1. For more information on ITK see <http://www.itk.org>
2. Installation of a combo binary `vtk44_yale` distribution containing the Tcl/Tk scripting language, various extensions and the Visualization Toolkit. This includes binary versions of:
  - The Tcl/Tk scripting language v 8.4.11 [34].
  - The following Tcl/Tk Extensions:
    - (a) Incr Tcl 3.2.1 [48] and Iwidgets 4.0.1.
    - (b) Tcllib 1.8.
    - (c) Metakit 2.4.93 [6] – this is not really required by BioImage Suite at this point.
    - (d) BWidget 1.6.1.
  - The CLapack numerical library v3.
  - A slightly patched version of the Visualization Toolkit v4.4.2 [92] with TCL Wrapping enabled. Two classes have been changed:
    - `vtkDataArray.cxx` – to eliminate a `GetRange()` bug – we simply used a later version of the file from the VTK CVS release.
    - `vtkImageReslice.cxx` – to set `Optimization` to `Off` by default as this gave us trouble with some of our nonlinear transformations in certain cases

3. Optionally the installation of a combo binary `bioimagesuite_extra` which primarily provides the MINC2 libraries and XERCESC – these are optional. (XERCESC is only really used at this point.)
4. Installation of the BioImage Suite software package itself.

All of the Open Source software listed in 1,2 and 3 essentially compiled without changes on *all platforms*.

BioImage Suite itself consists of a mixture of C++ libraries and .tcl code. The development process is described in the online book “Introduction to Programming for Medical Image Analysis with VTK” [74].

The BioImage Source code is released under the GNU General Public License (v2) [59].

## B.2 Compiling BioImage Suite

### Notes

Note that the binary `bioimagesuite_extra`, `itk241` and `vtk44` distributions distributed with BioImage Suite are needed for compiling this. While these are open source, it is easier to use the precompiled versions to avoid unnecessary problems. (See Section B.3 for some notes on building these distributions). A certain familiarity with the UNIX operating system and CMAKE is assumed.

**The installation assumes you are using CMake 2.4.x. CMake 2.6 might work.**

### Steps

1. Place the BioImage Suite source code into a directory we will refer to as `$MYSRC`.
2. Edit the `setpaths.csh` or `setpaths.sh` file to reflect their new location. i.e. in `setpaths.csh` change the line

```
setenv BASE /usr/local/bioimagesuite
```

to something like `$MYSRC/bioimagesuite ...` use **absolute paths**.

3. Source the appropriate `setpaths` file (much like when running the software) to set all environment variables i.e.

```
source \${MYSRC}/bioimagesuite/setpaths.csh
```

(or `setpaths.sh` if using one of the sh/ksh/bash shells. or modify `setpaths.bat` on MS-Windows).



## 4. Configure the build

```
cd \${MYSRC}/build;
ccmake ..
```

Set the following variables as:

```
LIBRARY_OUTPUT_PATH = \${MYSRC}/bioimagesuite/lib
EXECUTABLE_OUTPUT_PATH = \${MYSRC}/bioimagesuite/bin
```

5. Compile – Simply type `make`6. To run from the source tree build version type `bin.tcl`.

To verify that this is the source build click on the "Which" button on the bottom the main pxmenu application. If it says something like `mysrc/bioimagesuite` then all is good.

## B.3 Compiling the Prerequisites

These are some notes on building the binary distributions `vtk44_yale`, `itk241_yale` and `bioimage-suite.extra`. They are by no means complete. We strongly suggest using the binary versions provided if at all possible.

The exact sources used for these packages are available from the download page.

### B.3.1 The `vtk44_yale` distribution

#### Tcl/Tk and Extensions

\* Tcl and Tk – use `tcl 8.4.11/tk 8.4.11`

```
cd unix;
./configure --enable-shared --prefix=/usr/local/vtk44_yale
```

\* Mac OS X X11

```
Tcl: ./configure --prefix=/usr/local/vtk44_x11_yale \
      --disable-framework --enable-shared --enable-thread
Tk:  ./configure --prefix=/usr/local/vtk44_x11_yale \
```

```
--disable-framework --enable-shared --enable-threads --with-x \  
--disable-aqua  
Itcl: ./configure --prefix=/usr/local/vtk44_x11_yale \  
--disable-framework --enable-shared --enable-threads --with-x \  
--disable-aqua
```

### Incr Tcl (3.2.1)

```
./configure --enable-shared --prefix=/usr/local/vtk44_yale --with-gcc
```

### Iwidgets 4.0.1

```
./configure --enable-shared --prefix=/usr/local/vtk44_yale --with-itcl=./itcl3.2.1
```

### Tcllib - 1.8

```
./configure --enable-shared --prefix=/usr/local/vtk44_yale
```

### VTK 4.4

Use the patched source provided – to change

- `vtkDataArray.cxx` – to eliminate a `GetRange()` bug – we simply used a later version of the file from the VTK CVS release.
- `vtkImageReslice.cxx` – to set `Optimization` to `Off` by default as this gave us trouble with some of our nonlinear transformations in certain cases
- Enable all libraries – set `Java/Python OFF`, `Tcl ON`. Build `Shared`.

Before running CMAKE set

```
TCL_LIBRARY_PATH=/usr/local/vtk44_yale/lib:/usr/local/vtk44_yale/lib/vtk/tcl  
TK_LIBRARY_PATH=/usr/local/vtk44_yale/lib:/usr/local/vtk44_yale/lib/vtk/tcl
```

When configuring cmake

```
CMAKE_INSTALL_PREFIX=/usr/local/vtk44_yale
BUILD_SHARED_LIBS=ON
VTK_WRAP_TCL=ON
```

## Mac OS X X11

```
BUILD_EXAMPLES                OFF
BUILD_SHARED_LIBS             ON
CMAKE_BACKWARDS_COMPATIBILITY 2.2
CMAKE_BUILD_TYPE
CMAKE_INSTALL_PREFIX          /usr/local/vtk44_x11_yale
TCL_INCLUDE_PATH              /usr/local/vtk44_x11_yale/include
TCL_LIBRARY                   /usr/local/vtk44_x11_yale/lib/libtcl8.4.dylib
TK_INCLUDE_PATH               /usr/local/vtk44_x11_yale/include/;
                               /Users/xenios/x11vtk/tk8.4.11/xlib/;
                               /Users/xenios/x11vtk/tk8.4.11/generic
TK_INTERNAL_PATH              /Users/xenios/x11vtk/VTK/Rendering/tkInternals/tk84OSX
TK_LIBRARY                    /usr/local/vtk44_x11_yale/lib/libtk8.4.dylib
VTK_DATA_ROOT                 VTK_DATA_ROOT-NOTFOUND
VTK_USE_CARBON                OFF
VTK_USE_COCOA                 OFF
VTK_USE_HYBRID                ON
VTK_USE_PARALLEL              ON
VTK_USE_PATENTED              ON
VTK_USE_RENDERING             ON
VTK_USE_X                     ON
VTK_WRAP_JAVA                 OFF
VTK_WRAP_PYTHON               OFF
VTK_WRAP_TCL                  ON
CMAKE_X_CFLAGS                 -I/usr/X11R6/include
CMAKE_X_LIBS                  -lSM;-lICE;/usr/X11R6/lib/libX11.a;/usr/X11R6/lib/libXext.a
OPENGL_INCLUDE_DIR            /usr/X11R6/include
OPENGL_gl_LIBRARY             /usr/X11R6/lib/libGL.dylib
OPENGL_glu_LIBRARY            /usr/X11R6/lib/libGLU.dylib
```

## CLAPACK

Compiling CLAPACK This can be a pain, as Makefiles need to be edited by hand. Using the ”-fPIC” flag is helpful. This needs to be added to both ”make.inc” but also to all other Makefiles!

Since CLAPACK is in C, there is no good reason to not use the binary version pro-

vided. It will link just fine with libraries compiled with other compilers.

On linux, you can also use the lapack library that comes with the operating system.

Manually create `/usr/local/vtk44_yale/lib/lapack` and

```
cp F2CLIB/libf77.a /usr/local/vtk44_yale/lib/lapack
cp F2CLIB/libI77.a /usr/local/vtk44_yale/lib/lapack
cp blas.a /usr/local/vtk44_yale/lib/lapack/libblas.a
cp lapack.a /usr/local/vtk44_yale/lib/lapack/liblapack.a
```

## Config Files

The following two files need to be created in `/usr/local/vtk44_yale/lib/lapack`.

\* CLAPACKConfig.cmake

```
#-----
#
# CLAPACKConfig.cmake - CLAPACK CMake configuration file for external projects.
#
# This file is used by the UseCLAPACK.cmake module to load CLAPACK's settings f
or an external project.

# The CLAPACK include file directories.
SET(CLAPACK_INCLUDE_DIRS "/usr/local/vtk44_yale/lib/lapack/")

# The CLAPACK library directories.
SET(CLAPACK_LIBRARY_DIRS "/usr/local/vtk44_yale/lib/lapack/")

# The location of the UseCLAPACK.cmake file.
SET(CLAPACK_USE_FILE "/usr/local/vtk44_yale/lib/lapack/UseCLAPACK.cmake")

# The name of the CLAPACK project
SET(CMAKE_BUILD_SETTING_PROJECT_NAME "CLAPACK")
```

\* UseCLAPACK.cmake

```
#
# This module is provided as CLAPACK_USE_FILE by CLAPACKConfig.cmake. It can
# be included in a project to load the needed compiler and linker
# settings to use CLAPACK.
#

# Add include directories needed to use CLAPACK.
INCLUDE_DIRECTORIES(\${CLAPACK_INCLUDE_DIRS})

# Add link directories needed to use CLAPACK.
LINK_DIRECTORIES(\${CLAPACK_LIBRARY_DIRS})

SET(CLAPACK_LIBRARIES lapack blas I77 F77 )
```

### B.3.2 The itk241\_yale distribution

#### ITK 2.4.1

```
BUILD_SHARED_LIBS: ON
CMAKE_INSTALL_PREFIX: /usr/local/itk241_yale
```

### B.3.3 The bioimagesuite\_extra distribution

#### Xercesc

After you untar the source tree

```
cd xerces-c-src_2_7_0/
setenv XERCESROOT `pwd`
cd src
cd xercesc
./runConfigure -P /usr/local/bioimagesuite_extra -x g++ -c gcc
```

For 64-bit linux install add -b 64 to the line above (and perhaps -p linux)

### Minc2, Netcdf 3.6.1, HDF5 1.6.5

```
setenv FC ""
setenv CXX ""
```

On Solaris setenv CFLAGS "-fPIC" !

### Netcdf

```
cd src
./configure --prefix=/usr/local/bioimagesuite_extra/minc20_yale
```

### hdf5

```
./configure --prefix=/usr/local/bioimagesuite_extra/minc20_yale \
--disable-shared
```

### minc2

```
./configure --prefix=/usr/local/bioimagesuite_extra/minc20_yale \
CFLAGS=-I/usr/local/bioimagesuite_extra/minc20_yale/include/ \
LDFLAGS=-L/usr/local/bioimagesuite_extra/minc20_yale/lib/ \
--enable-minc2
```

On Solaris add -fPIC to CFLAFS.

## B.4 Miscellaneous

BioImage Suite is developed primarily on Linux workstations running CentOS 4.4/5.0 and the gcc 3.2 compiler. In addition the development team has access to a Sparc Ultra 10, an SGI Octane (2xR12K), a pair of Mac Minis (powerpc and intel) and a number of Windows XP/2000/VISTA

computers.

Some of the versions are compiled and/or tested inside virtual machines hosted using VMware Server. Such virtual machines include Linux versions: Red Hat 7.3, Red Hat 8.0, Fedora Core 3-5, Debian Sarge 3.1, Ubuntu 5.10,6.06 and OpenSUSE 10.0. In addition we have virtual machines for Solaris x86 v10, Darwin x86 v8.01 and Free BSD 6.0, as well as Windows 98.

BioImage Suite has been successfully compiled using gcc versions 2.95, 2.96, 3.2, 3.3, 3.4 and 4.01 on Linux and other operating systems, as well as Microsoft Visual Studio .NET 2003/2005 (and also using gcc3.3 under cygwin).

## Appendix C

# Bioimagesuite FAQ

This is a list of questions that are frequently asked on either the BioImageSuite Forum at <http://research.yale.edu/bioimagesuite/forum/index.php> or in person.

### C.1 Working with DICOM data

BioImage Suite uses as a default the "old" Mayo Analyze format in which each image is stored in a pair of files

.hdr = 348 byte header .img = raw image data

See <http://bioimagesuite.org/public/FileFormats.html>.

We can do some basic DICOM to Analyze conversion using the File Import tool, provide that as stated in the manual "DICOM import is limited to reading images in which the whole series is placed in its own directory in which there are no other files."

A quick Google search will reveal that there a number of tools out there for the DICOM-*i*Analyze Conversion, this web page

<http://www.sph.sc.edu/comd/rorden/dicom.html> is a good start. Internally we have a set of linux scripts to do this with, if you are using Linux, we can provide some of these tools as well.

There is a very nice DICOM viewer called ezDICOM which will be happy to do the conversion you need. You only need to ask it to convert to Analyze, point it to your DICOM file and presto, there is a .hdr and .img file sitting next to your .dcm file.

Find it here:



<http://www.sph.sc.edu/comd/rorden/ezdicom.html#intro>

Some other nice DICOM tools to be found at David Clunie's site <http://www.dclunie.com/> Have a look at [dicom3tools](http://www.dclunie.com/dicom3tools.html) <http://www.dclunie.com/dicom3tools.html> and also his PixelMed toolkit

<http://www.pixelmed.com/index.html#PixelMedJavaDICOMToolkit>

## C.2 How to install BIS on Debian

For running BioImage Suite 2.5 on Debian you first have to install VTK and ITK manually.

1. Prerequisites - You need to install following packages:

```
# apt-get install g++ make cmake xlibmesa-glx-dev tcl8.4-dev tk8.4-dev iwidgets4 tcllib \\  
dpkg-dev cdbshelp debhelper zlib1g-dev libpng12-dev libtiff4-dev lapack3-dev
```

2. Download and Install VTK-4.4.2 from source

```
$ wget --http-user=USERNAME --http-password=PASSWORD \\  
http://bioimagesuite.org/download/src/dependencies/VTK_4.4.2_patched.zip  
$ unzip VTK_4.4.2_patched.zip  
$ cd VTK  
$ cmake .
```

With

```
- VTK_USE_HYBRID           ON  
- VTK_WRAP_TCL            ON  
- VTK_USE_PARALLEL        ON  
- BUILD_TESTING           OFF
```

```
$ make  
# make install (as root, install path is /usr/local/)
```

3. Backport ITK 3.4

Add the Debian-Sid sources to your `/etc/apt/sources.list`

```
# pico /etc/apt/sources.list (deb-src http://ftp.de.debian.org/debian/ sid main)  
# apt-get update
```

To auto-build the package when it's been downloaded, just add `-b` to the command line, like this:

```
# apt-get -b source libinsighttoolkit-dev
```

If you decide not to create the .deb at the time of the download, you can create it later by running:

```
# dpkg-buildpackage -rfakeroot -uc -b
```

Install the packages

```
# dpkg -i libinsighttoolkit3.4_3.4.0-1_i386.deb
# dpkg -i libinsighttoolkit-dev_3.4.0-1_i386.deb
```

Remove the Debian-Sid sources from your /etc/apt/sources.list

```
# pico /etc/apt/sources.list (deb-src http://ftp.de.debian.org/debian/ sid main)
```

4. Download and Install BioImageSuite 2.5 (<http://bioimagesuite.org/download/src/>)

```
$ unzip bioimagesuite_2.5_01_29_2008_src_2366.zip
```

Change into BIS main directory

```
$ cd bioimagesuite_src/
```

uncomment clapack

```
$ pico CMakeOptions.cmake
```

```
$ cmake .
```

With

```
- BUILD_SHARED_LIBS ON
- BIOIMAGESUITE_USE_SYSTEM_LIBS ON
$ make
```

Set the BASE variable in file setpath.sh

```
$ pico bioimagesuite_src/bioimagesuite/setpaths.sh
```

(in my case: BASE=/home/guest/bioimagesuite/bioimagesuite-2.5.0/bioimagesuite\_src/bioimagesuite)

```
$ source bioimagesuite_src/bioimagesuite/setpaths.sh
```

Start the Program

```
$ bis.tcl
```

5. To “install” simply type “make install”.

6. A couple of additional comments

- Download and Install VTK-4.4.2 from source – please use the slightly patched version from the BioImage Suite webpage, this fixes a nasty bug – See: <http://bioimagesuite.org/download/src/dependencies/>. Otherwise the software will crash if the image you are using has more than 4 frames!
- You will also need to install “lapack”

Set BIOIMAGESUITE\_USE\_SYSTEM\_LIBS to ON to use this.

### C.3 Working with TIFF images

If these are ordinary tiffs. You should be able to import these into BioImage Suite more or less directly. Alternatively consider using one of the many conversion tools available to convert the tiffs to Analyze (7.5) format. Possible tools include

- LONI Debabeler:

[http://www.loni.ucla.edu/Software/Software\\_Detail.jsp?software\\_id=11](http://www.loni.ucla.edu/Software/Software_Detail.jsp?software_id=11)

- MRicro: <http://www.sph.sc.edu/comd/rorden/mricro.html>
- XMedcon: <http://xmedcon.sourceforge.net/>

While we can support a variety of formats internally, it might be better to use a tool specialized for format conversion prior to moving the images into BioImage Suite.

BioImage Suite prefers grayscale images, color would be most likely interpreted as time frame (i.e. multi-component images e.g. red, green and blue are typically dealt with as multiframe images). You are better off converting them to grayscale.

### C.4 How can I obtain Bioimagesuite?

Instructions for downloading BioImage Suite can be found at: <http://bioimagesuite.org/public/Downloads.html>.

Basically you need to

Register on this forum and log in. Follow this link <http://research.yale.edu/bioimagesuite/forum/index.php?topic=> to a message in the forum that specifies the username/password you will need to use to download the Software. This link is not accessible to non-members of the forum.

Please note: The username/password required to download BioImage Suite is not your forum username/password but rather the common username/password given in the thread pointed to by the previous link (<http://research.yale.edu/bioimagesuite/forum/index.php?topic=88.0>)

The only reason for this is so that we can keep track of who and how many are downloading the software. The process of registering on the forum is completely automated and mostly instantaneous.

### C.5 Is there any documentation for Bioimagesuite?

Other than this manual, which can also be found online at <http://www.bioimagesuite.org/public/Intro.html>, we have also taught a course on using Bioimagesuite for image analysis. The handouts for the course

and some test data can be found at <http://research.yale.edu/bioimagesuite/course/>

## C.6 How do I convert matlab format (.mat) into .hdr ?

Take a look at this toolkit

<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=8797&objectType=FILE>

(See also notes here <http://www.rotman-baycrest.on.ca/~jimmy/NIFTI/examples.txt>)

BioImage Suite can read both NIFTI and Analyze just fine.

# Bibliography

- [1] Cygwin: a Linux-like environment for Windows. <http://www.cygwin.com/>.
- [2] Brainsuite 2: a magnetic resonance (MR) image analysis tool designed for identifying tissue types and surfaces in MR images of the human head. <http://brainsuite.usc.edu/>.
- [3] AFNI. <http://afni.nimh.nih.gov/afni/>.
- [4] Automated Image Registration (AIR). <http://bishopw.loni.ucla.edu/air5/>.
- [5] A. A. Amini and J. L. Prince, editors. *Measurement of Cardiac Deformations from MRI: Physical and Mathematical Models*. Kluwer Academic Publishers, December 2001.
- [6] Metakit: an efficient embedded database library. <http://www.equi4.com/metakit.html>.
- [7] Analyze. <http://www.analyzedirect.com/>.
- [8] J. B. Arnold, J.-S. Liow, K. A. Schaper, J. J. Stern, J. G. Sled, D. W. Shattuck, A. J. Worth, M. S. Cohen, R. M. Leahy, J. C. Mazziotta, , and D. A. Rottenberg. Qualitative and quantitative evaluation of six algorithms for correcting intensity nonuniformity effects. *NeuroImage*, 13(5):931–943, 2001.
- [9] P. J. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor MRI. *J. Mag. Res., Series B*, 111(3):209–219, Jun 1996.
- [10] P. J. Besl and N. D. Mackay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.
- [11] BioPSE. <http://www.sci.utah.edu/ncrr/>.
- [12] H. P. Blumberg, C. Fredericks, F. Wang, J. Kalmar, L. Spencer, X. Papademetris, B. Pittman, A. Martin, B. S. Peterson, and J. H. Krystal R. Fulbright. Preliminary evidence for persistent abnormalities in amygdala volumes in adolescents with bipolar disorder. *Bipolar Disorders*, in-revision.
- [13] David Borland and Russell M. Taylor II. Rainbow color map (still) considered harmful. *IEEE Comput. Graph. Appl.*, 27(2):14–17, 2007.
- [14] BrainLAB, Heimstetten, Germany. <http://www.brainlab.com/>.
- [15] R. E. Carson. Tracer kinetic modeling. In PE Valk, DL Bailey, DW Townsend, and MN Maisey, editors, *Positron Emission Tomography: Basic Science and Clinical Practice*, pages 147–179. Springer-Verlag, London, 2003.

- [16] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. Journal of Computer Vision*, 22:61–79, 1997.
- [17] G. E. Christensen, M. I. Miller, and M. W. Vannier. Individualizing neuroanatomical atlases using a massively parallel computer. *Computer*, pages 32–38, 1996.
- [18] W. J. Chu, R. I. Kuzniecky, J. W. Hugg, B. A. Khalil, F. Gilliam, E. Faught, and H. P. Hetherington. Evaluation of temporal lobe epilepsy using 1H spectroscopic imaging segmentation at 4. 1T. *Mag. Res. Medicine*, 36:21–29, 2000.
- [19] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *Comp. Vision and Image Understanding*, 89(2-3):114–141, January 2003.
- [20] D. L. Collins, G. Le Goualher, and A. C. Evans. Non-linear cerebral registration with sulcal constraints. In *Med. Image Computing and Comp. Aided Inter. (MICCAI)*, pages 974–984. Springer, Berlin, 1998.
- [21] R. T. Constable, P. Skudlarski, and J. Gore. An ROC approach for evaluating functional brain MR imaging and post-processing protocols. *Mag. Res. Medicine*, 34:57–64, 1995.
- [22] I. Corouge, C. Barillot, P. Hellier, P. Toulouse, and B. Gibaud. Non-linear local registration of functional data. In M.-A. Viergever, T. Dohi, and M. Vannier, editors, *Med. Image Computing and Comp. Aided Inter. (MICCAI)*, pages 948–956, 2001.
- [23] P. Croissile, C. C. Moore, R. M. Judd., J. A. C. Lima, M. Arai, E. R. McVeigh, L. C. Becker, and E. A. Zerhouni. Differentiation of viable and nonviable myocardium by the use of three-dimensional tagged MRI in 2-day-old reperfused infarcts. *Circulation*, 99:284–291, 1999.
- [24] C. Davatzikos. Spatial transformation and registration of brain images using elastically deformable models. *Comp. Vision and Image Understanding*, 66(2):207–222, 1997.
- [25] RA de Graaf, JW Pan, F Telang, J-H Lee, P Brown, EJ Novotny, HP Hetherington, and DL Rothman. Differentiation of glucose transport in human brain gray and white matter. *J. Cereb. Blood Flow Metab.*, 21:483–492, 2001.
- [26] P. Dierckx. *Curve and Surface Fitting Splines*. Oxford, 1995.
- [27] M. DiStasio, K. Vives, and X. Papademetris. The BioImage Suite Datatree Tool: Enabling flexible realtime surgical visualizations. In *ISC/NA-MIC Workshop on Open Science at MICCAI 2006*, 2006. <http://hdl.handle.net/1926/208>.
- [28] Doxygen. <http://www.doxygen.org/>.
- [29] J. Duncan and N. Ayache. Medical image analysis: Progress over two decades and the challenges ahead. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):85–106, 2000.
- [30] J. Duncan, X. Papademetris, J. Yang, M. Jackowski, X. Zeng, and L. H. Staib. Geometric strategies for neuroanatomical analysis from MRI. *NeuroImage*, 23:S34–S45, 2004.
- [31] C. L. Duvall, W. R. Taylor, D. Weiss, and R. E. Guldberg. Quantitative microcomputed tomography analysis of collateral vessel development after ischemic injury. *Am. J. Physiol. Heart. Circ. Physiol.*, 287(1):H302–310, 2004.

- [32] E. Anderson et al. *Lapack User's Guide*. SIAM, 1999.
- [33] M. Galassi et al. *GNU Scientific Library Reference Manual*. Network Theory Ltd, 2003.
- [34] TCL Developer Exchange. <http://www.tcl.tk/>.
- [35] J. Feldmar, G. Malandain, J. Declerck, and N. Ayache. Extension of the ICP algorithm to non-rigid intensity-based registration of 3D volumes. In *Proc. Workshop Math. Meth. Biomed. Image Anal.*, pages 84–93, June 1996.
- [36] L. Freire and J-F Mangin. Motion correction algorithms may create spurious brain activities in the absence of subject motion. *NeuroImage*, 14:709–722, 2001.
- [37] K. Friston and W. Penny. Posterior probability maps and SPMs. *NeuroImage*, 19(3):1240–1249, 2003.
- [38] K. J. Friston, P. Jezzard, and R. Turner. The analysis of functional MRI time-series. *Human Brain Mapping*, 1:153–171, 1994.
- [39] K. J. Friston, S. R. Williams, R. Howard, R. S. J. Frackowiak, and R. Turner. Movement-related effects in fMRI time-series. *Mag. Res. Medicine*, 35:346–355, 1996.
- [40] K.J. Friston, J. Ashburner, J.B. Poline, C.D. Frith, J.D. Heather, and R.S.J. Frackowiak. Spatial registration and normalization of images. *Human Brain Mapping*, 2:165–189, 1995.
- [41] FSL. <http://www.fmrib.ox.ac.uk/fsl/fast/>.
- [42] J. M. Guccione, A. D. McCulloch, and L. K. Waldman. Passive material properties of intact ventricular myocardium determined from a cylindrical model. *Journal of Biomechanical Engineering*, 113:42–55, 1991.
- [43] R Guillemaud and M. Brady. Estimating the bias field of MR images. *IEEE Trans. Med. Imag.*, 16(3):238–251, 1997.
- [44] CJ Holmes, R Hoge, L Collins, R Woods, AW Toga, and AC Evans. Enhancement of MR images using registration for signal averaging. *J Comput Assist Tomogr*, 22(2):324–33, Mar-Apr 1998.
- [45] P. J. Hunter, B. H. Smaill, P. M. F. Nielsen, and I. J. LeGrice. A mathematical model of cardiac anatomy. In A. V. Panfilov and A. V. Holden, editors, *Computational Biology of the Heart*, pages 171–215. John Wiley & Sons, 1997.
- [46] L. Ibanez and W. Schroeder. *The ITK Software Guide: The Insight Segmentation and Registration Toolkit*. Kitware, Inc., Albany, NY, [www.itk.org](http://www.itk.org), 2003.
- [47] NIH Image. <http://rsb.info.nih.gov/nih-image/>.
- [48] [incr Tcl]. <http://incrtcl.sourceforge.net/itcl/>.
- [49] A. P. Jackowski, X. Papademetris, C. Klaiman, L. Win, B. Pober, and R. T. Schultz. A non-linear intensity-based brain morphometric analysis of williams syndrome. In *Human Brain Mapping*, 2004.

- [50] M. Jackowski, C. Kao, and L. Staib. Estimation of anatomical connectivity by anisotropic front propagation and diffusion tensor imaging. In *Medical Image Computing and Computer Aided Intervention (MICCAI)*, 2004.
- [51] M. Jackowski, C. Y. Kao, M. Qiu, R. T. Constable, and L. H. Staib. White matter tractography by anisotropic wavefront evolution and diffusion tensor imaging. *Med. Image Anal.*, in-press, 2005.
- [52] M. Jackowski, X. Papademetris, L. W. Dobrucki, A.J. Sinusas, and L.H. Staib. Characterizing vascular connectivity from microCT images. In *Medical Image Computing and Comp Aided Intervention (MICCAI)*, 2005.
- [53] P. Jezzard and S. Clare. Sources of distortion in functional MRI data. *Human Brain Mapping*, 8:80–85, 1999.
- [54] D. K. Jones, A. Simmons, S. C.R. Williams, and M. A. Horsfield. Noninvasive assessment of axonal fiber connectivity in the human brain via diffusion tensor MRI. *Mag. Res. Medicine*, 42(1):37–41, 1999.
- [55] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proc. Int. Conf. on Computer Vision*, pages 259–268, 1987.
- [56] C Lacadie, RK Fulbright, RT Constable, and Papademetris X. More accurate talairach coordinates for neuroimaging using nonlinear registration. *NeuroImage*, accepted, 2008.
- [57] JL Lancaster, PT Fox, S. Mikiten, and Rainey L. Talairach daemon, <http://ric.uthscsa.edu/new/resources/talairachdaemon/talairachdaemon.html>.
- [58] M. Leventon, E. Grimson, and O. Faugeras. Statistical shape influence in geodesic active contours. In *IEEE Conf on Computer Vision and Pattern Recognition*, pages 316–323, 2000.
- [59] GNU General Public License. <http://www.gnu.org/copyleft/gpl.htm>.
- [60] N. F. Lori, E. Akbudak, J. S. Shimony, T. S. Cull, A. Z. Snyder, R. K. Guillery, and T. E. Conturo. Diffusion tensor fiber tracking of human brain connectivity: acquisition methods, reliability analysis and biological results. *NMR in Biomedicine*, 15(7-8):494–515, 2002.
- [61] J.A. Maldjian and et al. An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fMRI data sets. *NeuroImage*, 19:1233–1239, 2003.
- [62] Matlab. <http://www.mathworks.com/products/matlab/>.
- [63] E. R. McVeigh. Regional myocardial function. *Cardiology Clinics*, 16(2):189–206, 1998.
- [64] MedX. <http://medx.sensor.com/products/medx/index.html>.
- [65] F. Meyer. Wavelet-based estimation of a semiparametric generalized linear model of fMRI time-series. *IEEE Trans. Med. Imag.*, 22(3):315–322, 2003.
- [66] V. L. Morgan, D. R. Pickens, S. L. Hartmann, and R. P. Price. Comparison of functional MRI image realignment tools using a computer generated phantom. *Mag. Res. Medicine*, 46:510–514, 2001.



- [67] S. Mori and P. C. M. van Zijl. Fiber tracking: principles and strategies - a technical review. *NMR in Biomedicine*, 15(7-8):468–480, 2002.
- [68] M. Neff. Design and implementation of an interface facilitating data exchange between an IGS system and external image processing software. Master’s thesis, Technical University of Munich, 2003. This project was jointly performed at BrainLAB AG (Munich, Germany) and Yale University (New Haven, CT U.S.A).
- [69] W. Ni, R. T. Constable, E. Mencl, K. R. Pugh, R. K. Fulbright, S. E. Shaywitz, B. A. Shaywitz, J. C. Gore, and D. Shankweiler. An event-related neuroimaging study distinguishing form and content in sentence processing. *J. Cognitive Neuroscience*, 12(1):120–133, 2000.
- [70] OpenGL. <http://www.opengl.org/>.
- [71] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley Professional, 1994.
- [72] X. Papademetris. Programming for medical image analysis using VTK, <http://noodle.med.yale.edu/papad/seminar/>.
- [73] X. Papademetris. *Estimation of 3D Left Ventricular Deformation from Medical Images using Biomechanical Models*. Ph. D. dissertation, Yale University, New Haven CT. (URL=<http://noodle.med.yale.edu/thesis>), May 2000.
- [74] X. Papademetris. *An introduction to programming for medical image analysis with the visualization toolkit*. A programming guide for the BioImage Suite project (<http://www.bioimagesuite.org/vtkbook/>), December 2006.
- [75] X. Papademetris, D.P. Dione, L. W. Dobrucki, L.H. Staib, and A.J. Sinusas. Articulated rigid registration for serial lower-limb mouse imaging. In *Medical Image Computing and Comp Aided Intervention (MICCAI)*, 2005.
- [76] X. Papademetris, A. Jackowski, R. T. Schultz, L. H. Staib, and J. S. Duncan. Computing 3D non-rigid brain registration using extended robust point matching for composite multisubject fMRI analysis. In *Med. Im Computing and Comp Aided Intervention (MICCAI) Part II LLNCS 2879*, pages 788–795. Springer-Verlag, 2003.
- [77] X. Papademetris, M. Jackowski, N. Rajeevan, R.T. Constable, and L.H Staib. BioImage Suite: An integrated medical image analysis suite, Section of Bioimaging Sciences, Dept. of Diagnostic Radiology, Yale School of Medicine, <http://www.bioimagesuite.org>.
- [78] X. Papademetris, M. Jackowski, N. Rajeevan, M. DiStasio, H. Okuda, R. T. Constable, and L. H. Staib. BioImage Suite: An integrated medical image analysis suite: An update, 2006. <http://hdl.handle.net/1926/209>.
- [79] X. Papademetris, J. V. Rambo, D. P. Dione, A. J. Sinusas, and J. S. Duncan. Visually interactive cine-3D segmentation of cardiac MR images. *Suppl. to the J. Am. Coll. of Cardiology Vol. 31, Number 2 (Suppl. A)*, February 1998.
- [80] X. Papademetris, P. Shkarin, L. H. Staib, and K. L. Behar. MRI-based whole body fat quantification in mice. In *Information Processing in Medical Imaging (IPMI)*, pages 369–380. Springer-Verlag, 2005.

- [81] X. Papademetris, A. J. Sinusas, D. P. Dione, R. T. Constable, and J. S. Duncan. Estimation of 3D left ventricular deformation from medical images using biomechanical models. *IEEE Trans. Med. Imag.*, 21(7), 2002.
- [82] X. Papademetris, A. J. Sinusas, D. P. Dione, and J. S. Duncan. Estimation of 3D left ventricular deformation from echocardiography. *Medical Image Analysis*, 5(1):17–28, March 2001.
- [83] X. Papademetris, K. P. Vives, M. DiStasio, L. H. Staib, M. Neff, S. Flossman, N. Frielinghaus, H. Zaveri, E. J. Novotny, H. Blumenfeld, R. T. Constable, H. P. Hetherington, R. B. Duckrow, S. S. Spencer, D. D. Spencer, and J. S. Duncan. Development of a research interface for image guided intervention: Initial application to epilepsy neurosurgery. In *International Symposium on Biomedical Imaging ISBI*, pages 490–493, 2006.
- [84] W. D. Penny and K. J. Friston. Mixtures of general linear models for functional neuroimaging. *IEEE Trans. Med. Imag.*, 22(4):504–514, 2003.
- [85] D. L. Pham and J. L. Prince. Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Trans. Med. Imag.*, 18(9):737–752, September 1999.
- [86] Apache HTTPD Server Project. <http://httpd.apache.org/>.
- [87] S. W. Provencher. Estimation of metabolite concentrations from localized in vivo proton NMR spectra. *Magn Reson Med.*, 30(6):672–9, Dec 1993.
- [88] J. A. Sethian R. Malladi and B. C. Vemuri. Shape modeling with front propagation: a level set approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):158–174, February 1995.
- [89] A. Rangarajan, H. Chui, and J. S. Duncan. Rigid point feature registration using mutual information. *Med. Image Anal.*, 3(4):425–440, 1999.
- [90] The Netlib repository. <http://www.netlib.org>.
- [91] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, and D. J. Hawkes. Non-rigid registration using free-form deformations: Application to breast MR images. *IEEE Trans. Med. Imag.*, 18(8):712–721, 1999.
- [92] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware, Inc., Albany, NY, [www.vtk.org](http://www.vtk.org), 2003.
- [93] VMware Server. <http://www.vmware.com/products/server/>.
- [94] J. A. Sethian. *Level set methods: Evolving interfaces in geometry, fluid mechanics, computer vision and materials science*. Cambridge University Press, 1996.
- [95] D.W. Shattuck, S. R. Sandor-Leahy, K.A. Schaper, D.A. Rottenberg, and R.M. Leahy. Magnetic resonance image tissue classification using a partial volume model. *NeuroImage*, 13(5):856–876, May 2001.
- [96] D. Shreiner, M. Woo, J. Neidera, and T. Davis. *OpenGL: Programming Guide: The official guide to learning OpenGL, Version 1.4*. Addison-Wesley Publishing, fourth edition, 2004.

- [97] A. J. Sinusas, X. Papademetris, R. T. Constable, D. P. Dione, M. D. Slade, P. Shi, and J. S. Duncan. Quantification of 3-D regional myocardial deformation: Shape-based analysis of magnetic resonance images. *American Journal of Physiology: Heart and Circulatory Physiology*, 281:H698–H714, August 2001.
- [98] P. Skudlarski. fMRI software package: [http://mri.med.yale.edu/fmri\\_software.htm](http://mri.med.yale.edu/fmri_software.htm).
- [99] P. Skudlarski, R. T. Constable, and J. C. Gore. ROC analysis of statistical methods used in functional MRI: Individual subjects. *Neuroimage*, 9(3):311–329, 1998.
- [100] JG Sled, AP Zijdenbos, and AC Evans. A nonparametric method for automatic correction of intensity nonuniformity in mri data. *IEEE Trans. Med. Imag.*, 17:87–97, 1998.
- [101] C. Smith. *[Incr-tcl/tk] from the Ground Up*. McGraw-Hill, 2000.
- [102] S.M. Smith. Fast robust automated brain extraction. *Human Brain Mapping*, 17(3):143–155, November 2002.
- [103] L. H. Staib and J. S. Duncan. Model-based deformable surface finding for medical images. *IEEE Trans. Med. Imag.*, 78(5):720–731, October 1996.
- [104] Statistical Parametric Mapping (SPM). <http://www.fil.ion.ucl.ac.uk/spm/>.
- [105] B. Stroustrup. *The C++ Programming Language: Second Edition*. Addison-Wesley, 1991.
- [106] C. Studholme, R. T. Constable, and J. S. Duncan. Accurate alignment of functional EPI data to anatomical MRI using a physics based distortion model. *IEEE Trans. Med. Imag.*, 19(11):1115–1127, November 2000.
- [107] C. Studholme, R. T. Constable, and J. S. Duncan. Non-rigid spin echo mri registration incorporating an image distortion model: Application to accurate alignment of fMRI to conventional MRI. *IEEE Trans. Med. Imag.*, 19(11), 2000.
- [108] C. Studholme, D. Hill, and D. Hawkes. Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimisation of voxel similarity measures. *Med. Phys.*, 24(1):25–35, 1997.
- [109] C. Studholme, D. Hill, and D. Hawkes. Automated three-dimensional registration of magnetic resonance and positron emission tomography brain images by multiresolution optimisation of voxel similarity measures. *Medical Physics*, 24(1):25–35, 1997.
- [110] C. Studholme, E. Novotny, I. G. Zubal, and J. Duncan. Estimating tissue deformation between functional images induced by intra-cranial electrode implantation using anatomical MRI. *NeuroImage*, 13(4):561–576, 2001.
- [111] M. Styner, C. Brechbühler, G. Székely, and G. Gerig. Parametric estimate of intensity inhomogeneities applied to mri. *IEEE Trans. Med. Imag.*, 19(3):153–165, 2000.
- [112] Free Surfer. <http://surfer.nmr.mgh.harvard.edu/>.
- [113] J. P. Thirion. Image matching as a diffusion process: An analogy with maxwell’s demons. *Med. Image Anal.*, 2(3):243–260, 1998.
- [114] A. W. Toga. *Brain Warping*. Academic Press, San Diego, 1999.

- [115] K. Van Leemput, F. Maes, D. Vandermeulen, and P. Suetens. Automated model-based bias field correction of mr images of the brain. *IEEE Trans. Med. Imag.*, 18(10):885–896, 1999.
- [116] Subversion: A version control system. <http://subversion.tigris.org/>.
- [117] VIDA. <http://dpi.radiology.uiowa.edu/vida/vidahome.html>.
- [118] 3D Slicer: Medical Visualization and Processing Environment for Research. <http://www.slicer.org>.
- [119] Brain Voyager. <http://www.brainvoyager.com>.
- [120] J. Wang, M. Qiu, Q. X. Yang, M. B. Smith, and R. T. Constable. Correction of transmission and reception induced signal intensity inhomogeneities *in vivo*. *Mag. Res. Medicine*, 2004. (submitted).
- [121] Y. Wang, R. Schultz, R. T. Constable, and L. H. Staib. Nonlinear estimation and modeling of fMRI data using spatio-temporal support vector regression. In C. Taylor and A. Noble, editors, *Information Processing in Medical Imaging*, pages 647–659. LNCS 2732, Springer, Berlin, 2003.
- [122] R. Weiss, S.E. Taksali, S. Dufour, C.W. Yeckel, X. Papademetris, G. Kline, W.V. Tamborlane, J. Dziura, G.I. Shulman, and S. Caprio. The “Obese Insulin Sensitive adolescent” – importance of adiponectin and lipid partitioning. *J. Clin Endocrinol. Metab.*, March 2005.
- [123] B. Welch, K. Jones, and J. Hobbs. *Practical Programming in Tcl and Tk: 4th Edition*. Prentice Hall, 2003.
- [124] W.M. Wells, R. Kikinis, W.E.L Grimson, and F. Jolesz. Adaptive segmentation of MRI data. *IEEE Trans. Med. Imag.*, 15:429–442, 1996.
- [125] J. Wernecke. *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor, Release 2*. Addison-Wesley, 1994.
- [126] R. P. Woods, S. R. Cherry, and J. C. Mazziotta. Rapid automated algorithm for aligning and reslicing PET images. *J. Comput. Assist. Tomogr.*, 16:620–633, 1992.
- [127] J. Yang, L. H. Staib, and J. S. Duncan. Neighbor-constrained segmentation with level set based 3D deformable models. *IEEE Trans. Med. Imag.*, 23(8):940 – 948, Aug 2004.
- [128] X. Zeng, L. H. Staib, R. T. Schultz, and J. S. Duncan. Segmentation and measurement of the cortex from 3D MR images using coupled surfaces propagation. *IEEE Trans. Med. Imag.*, 18(10), 1999.
- [129] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden markov random field model and the expectation maximization algorithm. *IEEE Trans. Med. Imag.*, 20(1):45–57, 2001.