FusionViewer: An Open Source Toolkit for Viewing Multimodality Images

Release 1.0

A. Lu[1], C. Lau[2], L. Ng[2,3], L. Gong[1], P. E. Kinahan[3], A. Alessio[3], J. Goldschneider[3],and S. D. Pathak[2,3]

February 18, 2008

[1]Insightful Corporation, Seattle WA
[2]Allen Institute, Seattle WA
[3]University of Washington, Seattle WA

Abstract

FusionViewer is an open source and platform independent viewer that has been specifically designed for PET/CT image display. The combination of PET and CT images offers complementary functional and anatomical information. The application (FusionViewer) facilitates efficient visualization and analysis of PET/CT studies via different viewing modes (linked cursor display, alpha-blend mode, checkerboard mode and split window mode). FusionViewer is implemented in Java and uses the Java OpenGL (JOGL) library and the Insight Segmentation and Registration Toolkit (ITK) library, which make it both a fast and cross-platform application. Its intuitive graphical user interface makes it easy to be used by physicians, radiologists, and research scientists. Several analysis and display tools are already available (navigator, zoom, pan, screen snapshot, ROI, and line measure tool; alpha-blending, checkerboard display, and split window display). Along with PET/CT, several other modalities where co-registered images are often visualized simultaneously have benefited from the use of this software.

Contents

1. Introduction 2
2. FusionViewer Design 2
2.1. Architecture Overview 2
2.2. Control Flow Exemplars 3
2.2.1. IO Control Flow 3
2.2.2. ROI Measurement Flow 5
2.2.3 Mouse Interaction Flow 6
3. FusionViewer Use Cases 6
3.1. Scenario 1: Physician or Researcher Using the Standard Release 7
3.2. Scenario 2: Researcher Integrating Their Application 7
3.3. Scenario 3: Other Research Uses 8
4. Conclusions 10

1. Introduction

Cancer management using positron emission tomography (PET) imaging is rapidly expanding its role in clinical practice [1][2]. The high sensitivity of PET to detect cancer is confounded by the minimal anatomical information it provides. Additional anatomical information would greatly benefit diagnosis, staging, therapy planning and treatment monitoring. Computed tomography (CT) provides detailed anatomical information but is less sensitive for cancer detection and/or characterization than PET. Combining PET and CT images enables accurate localization of the functional information with respect to detailed patient anatomy. In clinical settings, most PET/CT image review is currently performed using manufacturer specific software solutions associated with vendor specified application interfaces and are often distributed in packages tied to specific workstations. We believe an open source and platform independent visualization application will improve both researchers and practitioners' ability to review, interpret and analyze multimodality images. Furthermore, providing the ability to view PET/CT images directly on a physician's desktop/laptop computer (independent of the scanner manufacturer) along with improved display options should lead to greatly increased clinical efficacy of fusion technology. FusionViewer is designed with the aforementioned philosophy in mind, and it is also unique in these regards.

The two areas of emphasis in the application are to:

- Design a solution that is robust, fast and requires minimum user interactions, and
- Provide a variety of options for displaying and analyzing the PET and CT images. These images are independent volumetric data sets that are aligned (i.e. by acquisition on a dual-mode PET/CT scanner or by post-acquisition registration of data acquired on separate PET and CT scanners) but typically with different voxel sizes. It is worth noting that there are as yet no accepted standards (e.g. DICOM) for fused or linked image volumes.

FusionViewer is implemented in Java, linked to the Java OpenGL (JOGL) library and Insight Segmentation and Registration Toolkit (ITK) library, which make it both fast and a cross-platform application. The application details are documented and posted at sourceforge.net. In this article we provide a high level description of the FusionViewer for developers and end users.

2. FusionViewer Design

2.1. Architecture Overview

FusionViewer employs several classical design patterns [3] judiciously chosen for optimal display of multimodality images with emphasis on the viewer being lightweight (in terms of different libraries used and the algorithms used for different applications) and fast in performing a variety of image manipulation tasks on very large data sets. Given these constraints, advanced segmentation and registration algorithms have not been integrated into it, although one can easily customize the open source environment to meet application needs.

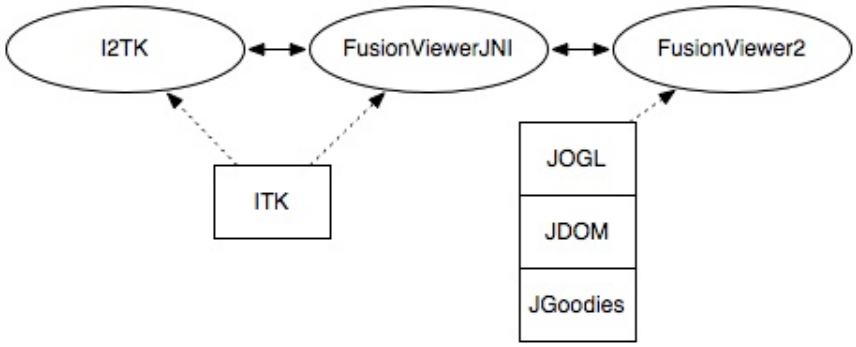

Figure 1. FusionViewer architecture.

The FusionViewer application, Figure 1, is divided into three modules: I2TK, FusionViewerJNI, and FusionViewer2. The I2TK module, implemented in C++, provides application specific extensions to ITK. The FusionViewer2 module, implemented in Java, is the graphical user interface (GUI). The FusionViewerJNI module, implemented in C++, provides the interface between the C++ native functionalities and the FusionViewer2 GUI module. The I2TK module and the FusionViewerJNI module must be compiled before the FusionViewer2 module for those who want to develop or extend this toolkit.

The FusionViewer2 GUI module is comprised of four packages: *display*, *input-output* (*IO*), *model* and the *user interface* (*UI*). The classes in each of the four packages are documented using Doxygen [4] and can be effectively navigated via the HTML interface.

The application involves user interactions with the image data; therefore the control path in the code can differ significantly as it depends on the context. In this paper, we briefly outline three different control flows to orient the developer with the different classes used in the context of achieving a specific goal.

2.2. Control Flow Exemplars

In this section we describe function control flows for three key tasks: reading images (IO control flow); performing ROI based measurements (ROI measurement flow); and tracking mouse control flows associated with different image manipulation tools (mouse interaction flows).

2.2.1. IO Control Flow

The FusionViewer application is aimed primarily for PET/CT image fusion; however, the design of the toolkit is done in a generic manner to facilitate a broad range of multimodality image viewing. We have chosen to leverage the IO factory classes in the ITK toolkit to read in a variety of different images, which enables us to take advantage of the different image meta storage classes associated with the image container along with additional interfaces. The interface of the ITK toolkit to Java is facilitated by a Java Native Interface [5] (FusionViewerJNI). The researcher who would like to integrate additional ITK algorithms into the FusionViewer application may use the JNI interfaces as a template for further customization.

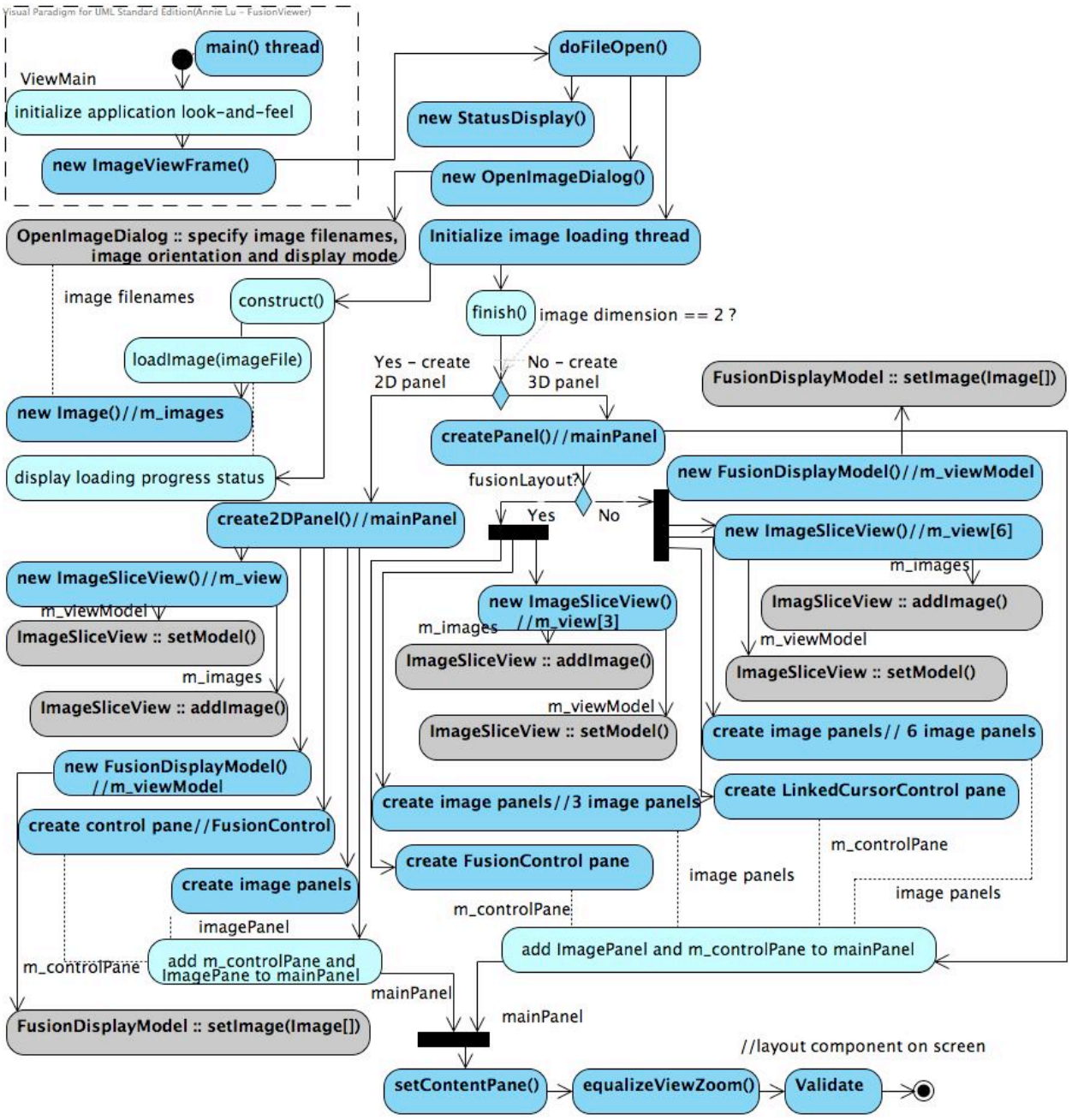

Figure 2. UML diagram for image loading control flow.

Figure 2 shows the control flow for importing images into the application. The entry point to the application is the `ViewMain` class. A dialog interface managed by class `OpenImageDialog` subsequently creates the function calls that load the images using the ITK IO factory. Following the load operation, which involves reading images into memory, the control shifts to the `FusionDisplayModel` class which is used to define the data structures that store image information associated with a user specified view layout (such as the linked cursor view or the fused image display view) and image orientation (axial, sagittal, or coronal). Once the model is successfully populated the *display* package has classes for rendering the buffered image data onto the screen.

2.2.2. ROI Measurement Flow

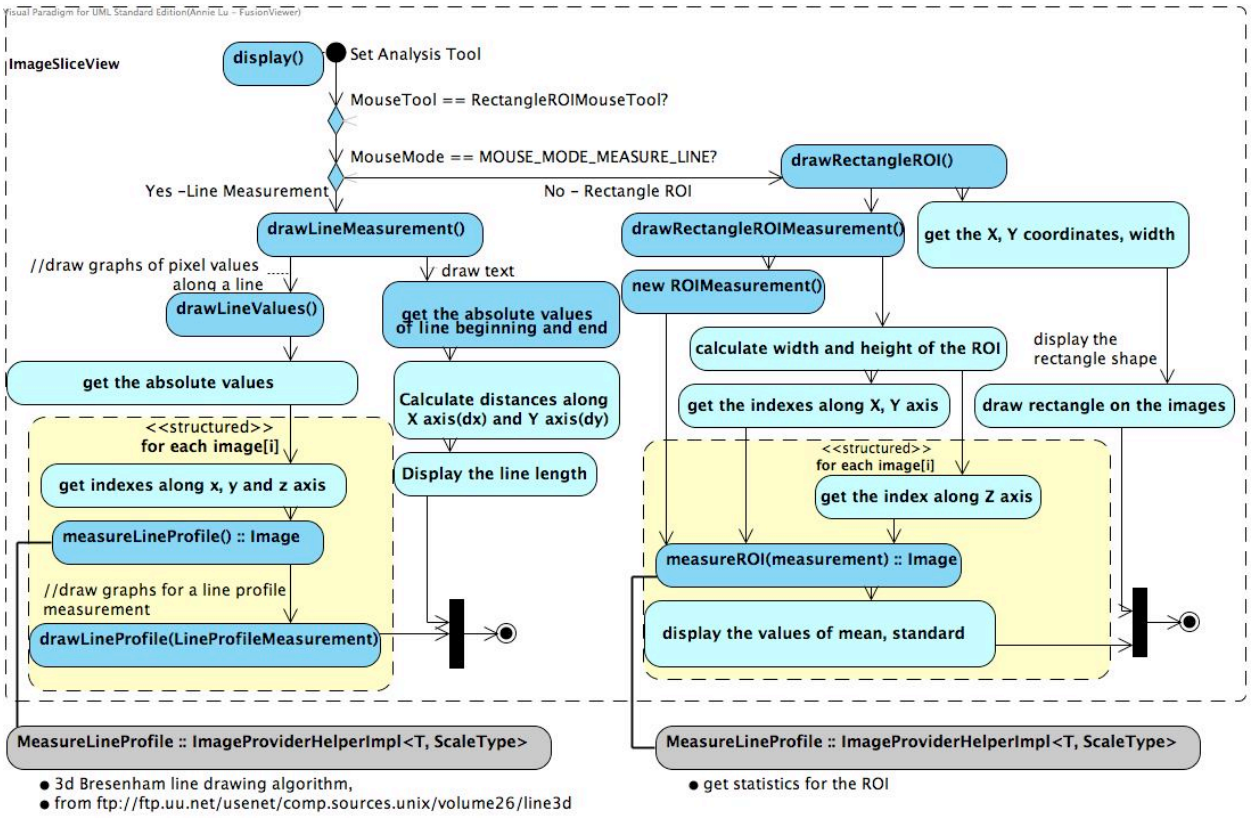

Figure 3. UML diagram for region of interest measurements.

The second control flow diagram, Figure 3, illustrates the functional layout for ROI based measurements. In this case, the `ImageSliceView` class forms the entry point the ROI based calculations. Two primary ROI tools are currently supported. The first is a rectangular region of interest, and the second is a line based measurement feature. The two ROI tools while functionally similar (i.e. each calculates a variety of statistics associated with the region), the implementation involves different interaction with the display interface. For instance, the rectangular ROI tool outlines the results as a text output in the FusionViewer display panel, while the line tool has to display, in addition to text output such as line length, a graphical display of the pixel intensity profiles for both foreground and background images.

2.2.3 Mouse Interaction Flow

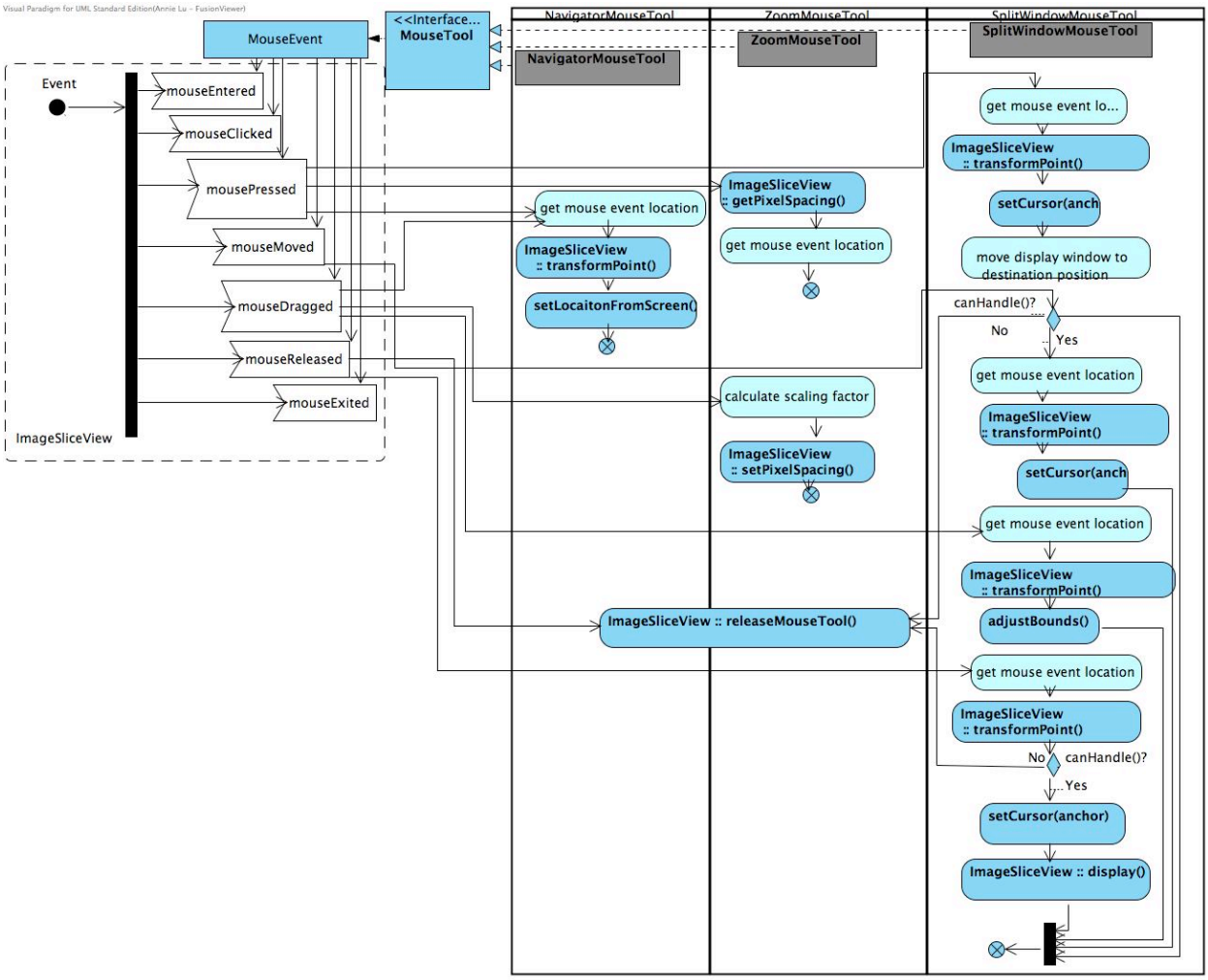

Figure 4. UML diagram for three mouse events.

The last control flow diagram, Figure 4, shows the intricate interaction of the different analysis tools with the mouse. To save space, only three different functions involving mouse interactions are illustrated in this figure: The *Navigator* tool allows the cursor to be moved through a 3D space, in both fusion or linked cursor display modes; the *Zoom* tool facilitates seamless zooming on a specific image panel; and the *SplitWindow* tool allows the user to display one image in a resizable floating window over the other image to enable fused viewing of the windowed region with variable opacity in the fusion display mode. These tools are implemented using a combination of standard UI based mouse interaction schemes involving the listener design pattern.

3. FusionViewer Use Cases

FusionViewer is intended to be of use in three different categories of application. In this section we present some of the use cases that we have anticipated in our design of the toolkit.

3.1. Scenario 1: Physician or Researcher Using the Standard Release

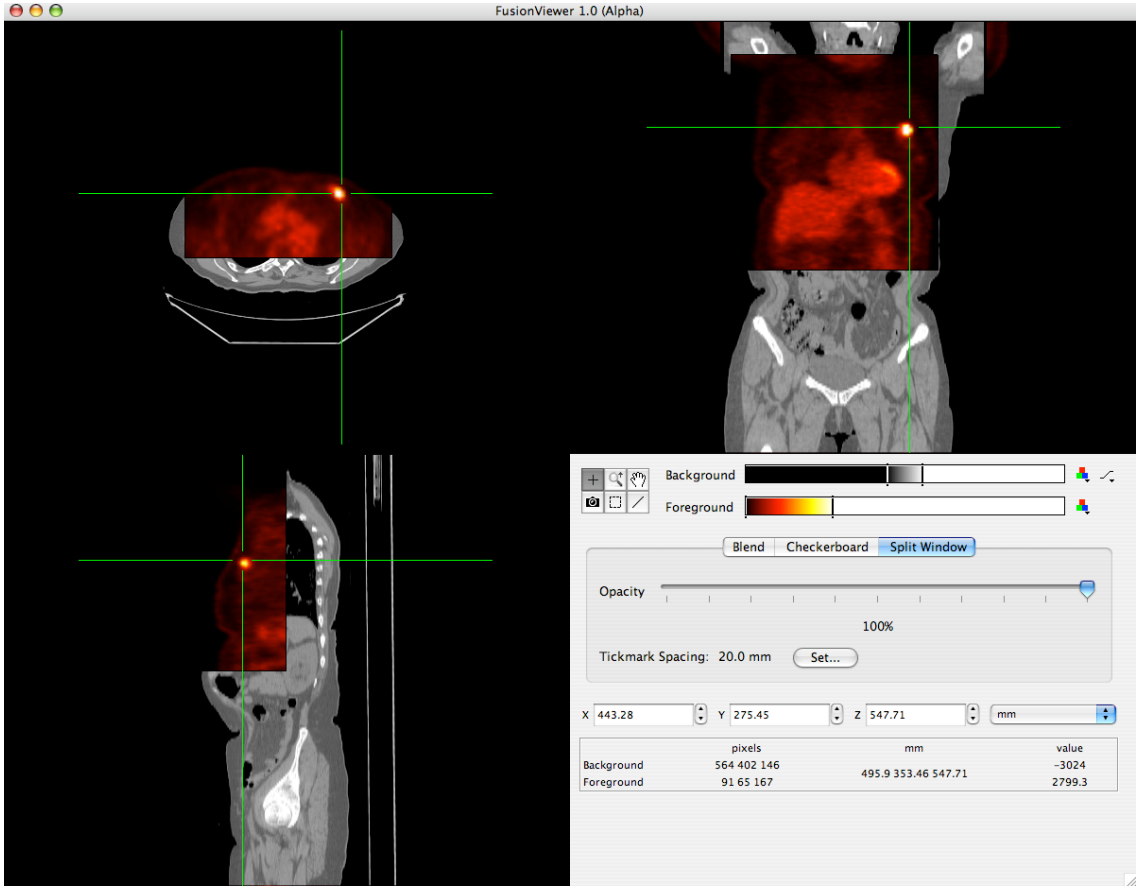

Figure 5. Split window display.

Physicians and researchers are often faced with the challenge of viewing multimodality or longitudinal data, such as PET/CT images, dynamic MRI images, before and after studies, and viewing a scan with respect to a standard atlas [6]. Many times the images are in different formats and may have been acquired using different scanners. We anticipate that FusionViewer will be able to successfully fill an unmet need for many of these users. The user in this case is simply going to use the FusionViewer application to load the images, perform multimodality image viewing, and record clinical observations. The user could also overlay multimodality images, and use for example, the split window display to improve visualization of pathology with normal anatomy as shown in Figure 5.

3.2. Scenario 2: Researcher Integrating Their Application

Researchers are continually seeking improvements to the discovery, diagnosis, and treatment processes. New imaging modalities and new algorithmic innovations for data processing present visualization challenges since there is no one particular viewing application designed to support these activities. As a result, researchers are often challenged in having to create their own display interfaces, resulting in a variety of software tools each having their own advantages and disadvantages. Often times for visualization of multimodality data, the data need to be displayed in a consistent manner but visualization

requires the flexibility of using different colormaps and window level settings. The FusionViewer application allows the user to customize these different display options using a XML based configuration interface. In addition, we believe the medical imaging research community has by and large accepted the ITK toolkit as a source of advanced algorithms. Based on the current framework that interfaces ITK with FusionViewer, we expect that a researcher will be able to customize this lightweight viewer to adapt different algorithms for specific application development.

3.3. Scenario 3: Other Research Uses

The capabilities of FusionViewer extend beyond PET/CT image viewing applications. In this section we present two examples where the application has been used for two- and three-dimensional histological tissue images and for three-dimensional magnetic resonance angiography.

The Allen Institute has successfully completed a genome-wide anatomical mouse brain atlas with cellular resolution (http://www.brain-map.org). An informatics pipeline was built to map in-situ hybridized tissue sections stained for mouse genome mRNA transcripts into a common anatomical three-dimension digital mouse atlas using custom registration techniques followed by segmentation of individual cellular regions of expression for further informatics data analysis and image search [7]. FusionViewer was found to be of great value for visualizing two separately aligned channels of both two- and three-dimensional image data at various stages of the project. For instance, to study the segmentation of cellular regions from tissue sections [7], on the foreground channel an ISH image and in the background channel segmentation results were loaded. By using the blending options the segmentation performance could easily be tested on a large number of two-dimensional images. Once the expression information from these images was mapped back to the three-dimension Reference Atlas, the 3D mode of FusionViewer was used for data visualization. The ability to load customized colormaps with ease facilitated rapid and effective illustrations for both interpretations and publications [7][8].

In magnetic resonance angiography (MRA), the use of blood pool contrast agents allows high resolution images of the peripheral vasculature to be acquired during the equilibrium phase of contrast-enhancement. However, a difficulty of equilibrium phase imaging is the simultaneous enhancement of arteries and veins. After artery-vein separation is performed, the segmented arteries (or veins) and original equilibrium phase datasets can be visualized simultaneously using the FusionViewer [9][10][11]. As shown in Figure 6, the blending between the segmented arteries (or veins) and original images can be controlled via a slider, allowing users to rapidly switch between the two datasets. Using the FusionViewer, vessel segments that were missed by the segmentation algorithm could easily be tracked in the original dataset, which users could refer to for clarification.

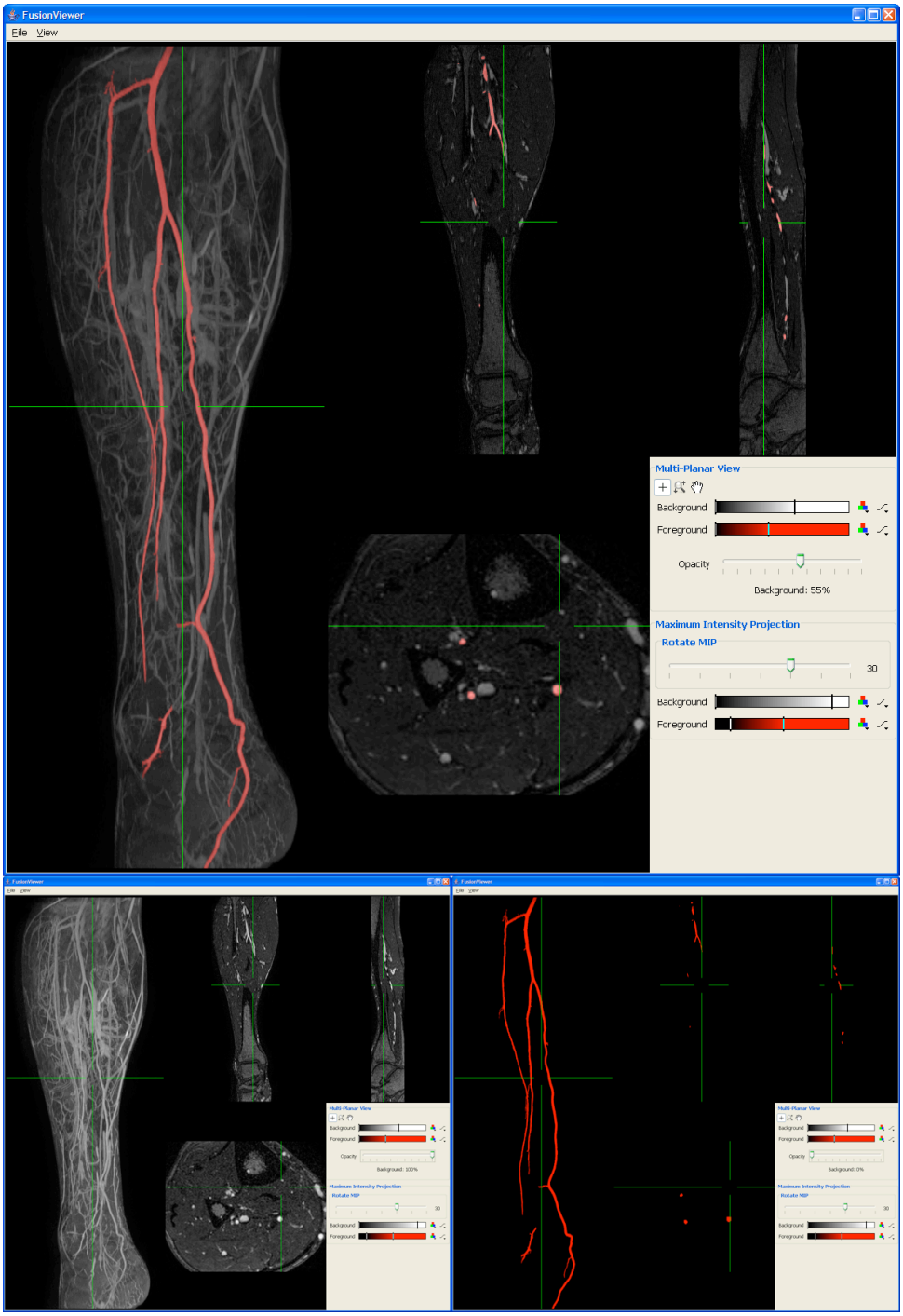

Figure 6: FusionViewer allowing visualization of both original and segmented equilibrium phase MRA datasets.

4. Conclusions

We have presented a high-level overview of the FusionViewer application and elaborated on its design. The different scenarios presented illustrate the scope of the toolkit and is aimed to promote its usage in both medical imaging community and the larger image processing community in general. We refer the reader to the on-line documentation for more details.

Acknowledgements

This work was supported by NIH Grants CA099329 and CA115870.

References

[1] I.R. Francis, R.K. Brown, A.M. Avram. "The clinical role of CT/PET in oncology: An update." Cancer Imaging 5,S68-S75, 2005.

[2] A. Alessio, P.E. Kinahan, P. Cheng, H. Vesselle, J.S. Karp. "PET/CT Scanner Instrumentation, Challenges, and Solutions." Radiologic Clinics of North America. 42(6):1017-32, 2004.

[3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, Massachusetts: Addison-Wesley, 1994.

[4] D. van Heesch. Doxygen – A Documentation System for C++, C, Java, Objective-C, Python, and IDL. Online at www.doxygen.org.

[5] S. Liang. The Java Native Interface Programmer's Guide and Specification. Sun Microsystems, 2002.

[6] J.B. Antione Maintz and Max A. Viergever. "A survey of medical image registration." Medical Image Analysis (2)1:1-36. Oxford University Press, 1998.

[7] L. Ng, S.D. Pathak, L. Kuan, C. Lau, H.W. Dong, A. Sodt, C. Dang, J. Gee, E. Lein, A. Jones, and M. Hawrylycz, "Genomics scale neuroinformatics for 3-D gene expression mapping." IEEE/ACM Tranactions on Computational Biology and Bioinformatics. 4:382-93, 2007.

[8] E.S. Lein, et al., "Refining cellular neuroanatomy through unbiased genome wide survey of gene expression in the brain." Nature, 11:168-76, 2007.

[9] M.S. Wang, D.R. Haynor, G.J. Wilson, and J.H. Maki, "Intravascular hematocrit layering in equilibrium phase contrast-enhanced MR angiography of the peripheral vasculature." Journal of Magnetic Resonance Imaging, 24(6):1393-400, 2006.

[10] M.S. Wang, D.R. Haynor, T. Leiner, G.J. Wilson, M. Shutske, W. Eubank, J.H. Maki, "Artery-vein separation of ultra-high resolution contrast-enhanced MRA using a blood pool agent," International Society for Magnetic Resonance in Medicine, #5930, 2007.

[11] M.S. Wang. "Ultra-high resolution imaging and artery-vein separation of blood pool contrast-enhanced MRA." Thesis (Ph. D.), University of Washington, 2007.