ECE1778 Final Report
MRI Visualizer

David Qixiang Chen
Alex Rodionov

Word Count: 2408

Introduction

We aim to develop a mobile phone/tablet based neurosurgical MRI visualization application with the goal of assisting doctors to visualize and diagnose brain related pathologies without being present at a desktop workstation.

Magnetic Resonance Imaging (MRI) has been at the forefront of medical field as an important diagnostic tool. It has become a staple of modern surgical planning procedures in all aspect of the surgical field. Traditional MRI techniques present patient information in cross-sectional 2D planes of the subject. The brain possesses arguably the most complex gross anatomy in the human body, visualize and diagnose brain pathologies from 2D MRI representations is difficult to all but the most experienced doctors. New advancement in MRI technology allows 3D reconstruction of sub-cortical structures of the brain by means of diffusion tensor imaging (DTI). However these processes often require time consuming preprocessing and subsequently are limited to academic research.

Existing software such as 3D Slicer (http://www.slicer.org/) has advanced far in the field of work-station level 3D MRI visualization, and we have previously established the feasibility of visualizing cranial nerve and pre-surgical visualization of brain tumors such as vestibular schwannoma (Chen et al., 2011; Hodaie, Quan, & Chen, 2010). However training surgeons to utilize complex visualization software to manage and diagnose pathologies in clinical environment is infeasible. We aim to popularize the clinical use of such pre-surgical planning visualization in the hands of surgeons with minimal technical barrier.

Proposed Design

There are 3 primary areas of focus for our application: visualization of MRI scans, draw and visualize pre-generated 3D models, and manipulation of the 3D scene.

MRI Scans

The ability to import MRI volume files is without question a very important. There are many MRI formats from different MRI scanner manufacturers. We aim to import pre-computed 3D volume files from 3D Slicer, therefore we choose to support the NRRD format from the TEEM library that 3D Slicer uses extensively (http://teem.sourceforge.net/nrrd/).

There are more than two major types of MRI volumes that we plan to support.

Scalar Volumes

The most common type of MRI files is in the form of scalar volumes. These are commonly represented as header file with descriptions of scanning parameters and spatial alignments, and a binary block containing a 3 dimensional float array of voxel intensities.

DTI Volumes

Diffusion Tensor Imaging (DTI) is a novel neuroimaging modality that differs from conventional MRI in its sensitivity to changes in white matter microstructure. It is founded on the theory that random Brownian motion of water molecules can become constrained by the structure of its diffusion medium. MR images are obtained by scanning the brain as it is weighted against planes of different angles called gradient directions. The resulting MR image is used to calculate tensors containing the principal water diffusion directions as vectors and their respective diffusion magnitudes, or Eigenvalues referred to as λ_1, λ_2, λ_3. The resulting DTI Tensor volume is represented as a 3 dimensional array of vectors, and drawn by assigning a color to each of voxel depending on its vector orientation and Eigen magnitudes.

DTI volumes are loaded like regular MRI volumes, but will be visualized as color-by-orientation mode. Essentially a color is assigned for each orthogonal axis (left-right: red, anterior-posterior: green, superior-inferior:blue), a voxel is coloured based on its orientation towards these colors.

3D models

Often 3D models are generated from segmented anatomical landmarks. The models are to be drawn in the 3D viewport. They will be generated from 3D Slicer, and imported as .vtk format (www.vtk.org/VTK/img/file-formats.pdf).

General polygonal models

These are generic polygonal models defined as a list of vertices and polygons. Models of tumours, segmented structures such as the thalamus or hippocampus fall into this category. They will be assigned a color and drawn using OpenGL.

Tractography models

Tractography is a specialized method to trace the DTI tensor volume and generated a stream of polylines to visualize subcortical white matter tracts. It not only contains a list of vertices and lines, it also contains the tensor vector information at each vertex, allowing the software to visualize the tract by coloring each vertex by its tensor value.

GUI Interaction

We will provide the traditional 2D slice interface to view the MR volume, as well as a 3D view to allow free manipulation of the scene.

2D View

The 2D slice view represents a slice of the 3D MR volume along one of its primary axes (axial, sagittal and coronal). In accordance to the neuroradiology conventions, it will be represented in radiological order. Meaning that in the axial view, left and right side of the brain is the opposite to that of the viewer. The slice direction can be chosen by buttons at the edge of the screen.

View Navigation

Scrolling: Slices can be scrolled along its chosen axis by two fingers gliding across the touch screen.

Zoom: The current image slice can be zoomed by pinching the viewport with two fingers, or along a zoom bar widget.

Panning: If the image slice is zoomed to be larger than the screen, then a one finger sliding gesture will pan the image.

3D View

In the 3D View, all three primary orthogonal slices of the MR volume will be visible. User can select an orthogonal slice to manipulate by clicking on it in the viewport. A two-finger scrolling motion will scroll the selected slice along its axis. The visibility of the orthogonal slices can be toggled by widgets by the side of the screen. 3D mesh can also be selected and with their visibility toggled.

View Navigation

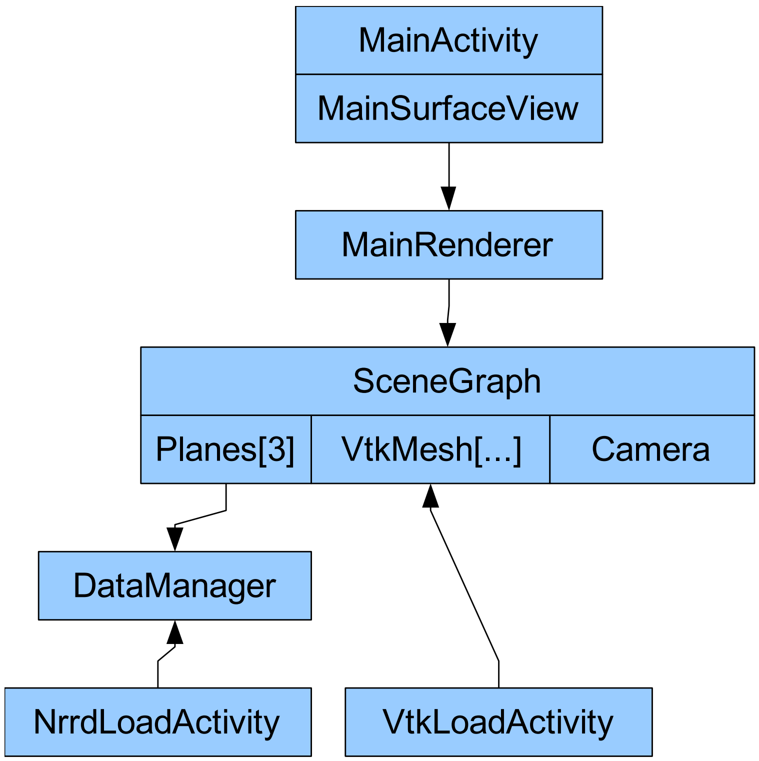
Rotation: Gliding of a single finger will rotate the 3D camera around the 3D scene.

Zoom: Pinching the two fingers on screen will zoom in and out of the 3D scene above the center of the screen.

Pan: A two finger pan gesture, where one finger is static, while another finger moves will pan the view along the perspective of the camera.

Roll: A two finger rolling gesture, where the two fingers are moving in opposite directions, will roll the view along the perspective of the camera.

Design Implementation



The above figure is a block diagram showing the main modules of our application. We describe each in more detail:

MainActivity: Presents the main user interface for viewing MRI data. It instantiates **MainSurfaceView** which is the component that all 3D rendering is output to.

MainRenderer: This module handles all drawing operations and maintains the state of which objects should be visible at any given time. The function that gets called when a frame is drawn is located here.

SceneGraph: This is a container for the various objects in the 3D world, including the three planes that show slices of an MRI volume, and any number of VtkMeshes which are polygonal models that also live in the 3d space. The Camera object maintains the state of the current rotation and location of the user's view in the 3D world.

DataManager: Encapsulates the 3D volume data to be drawn. When the Planes are rendered, they get their slice images from here.

NrrdLoadActivity: Provides the user interface to select 3D volume data (.nrrd) files for loading. Passes the filename to the DataManager which does the loading.

VtkLoadActivity: Provides the user interface to select polygonal object (.vtk) files for loading. It constructs a VtkMesh object and adds it to the SceneGraph, allowing it to be rendered along with all the other world objects.

Implemented Functionalities

2D View

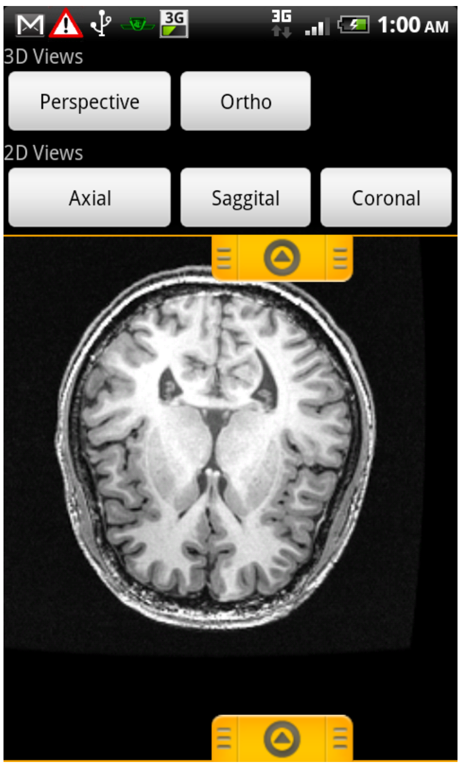


Figure 1

The 2D slice view represents a slice of the 3D MR volume along one of its primary axes (axial, sagittal and coronal). In accordance to the neuroradiology conventions, it is represented in radiological order. Meaning that in the axial view, left and right side of the brain is the opposite to that of the viewer (Figure 1).

Implemented:

- **View along primary axes:** Axial, sagittal and coronal views are toggled through button on the top sliding panel.
- **Zoom view:** Pinching the two fingers on screen will zoom in and out of the 3D scene.
- **Pan view:** Mode selected by a button in the button control panel. Gliding of a single finger will pan the view about the current view direction.

Not implemented

- **Scrolling**: Scrolling by two finger multi-touch is not implemented because of the lack of support for multiple two finger gestures in the Android API. There is significant difficulty in discerning the difference between the native two-finger pinch zoom gesture support in the Android 2.2 SDK, and a custom two-finger parallel movement gesture.

3D View

In the 3D View, all three primary orthogonal slices of the MR volume are visible (Figure 2).

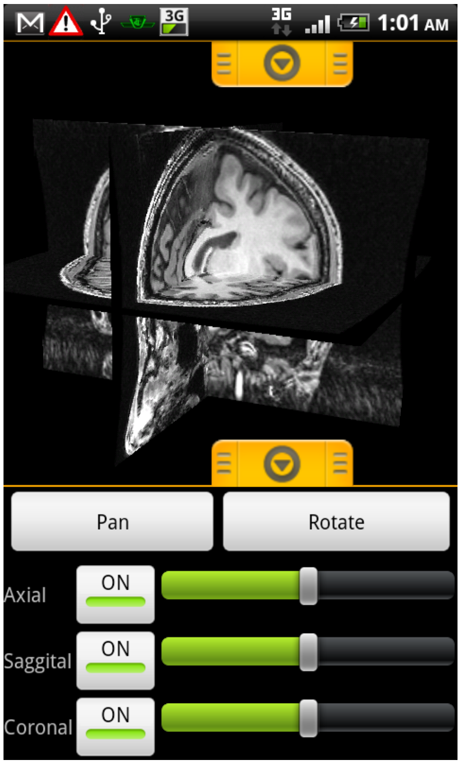


Figure 2

Implemented

- **Rotation view:** Mode selected by a button in the bottom sliding panel. Gliding of a single finger will rotate the 3D camera around the 3D scene.
- **Zoom view:** Pinching the two fingers on screen will zoom in and out of the 3D scene.
- **Pan view:** Mode selected by a button in the bottom sliding panel. Gliding of a single finger will pan the view about the current view direction.

Not implemented

- **Two finger pan gesture:** Not implemented because of lack of multi-touch gestures support in the API. See scrolling in 2D views.
- **Two finger roll gesture**: Rolling of the view is possible by touch and drag along the edge of the screen under view rotation by the implementation of a track-ball camera. It's redundant to have a separate rolling function.

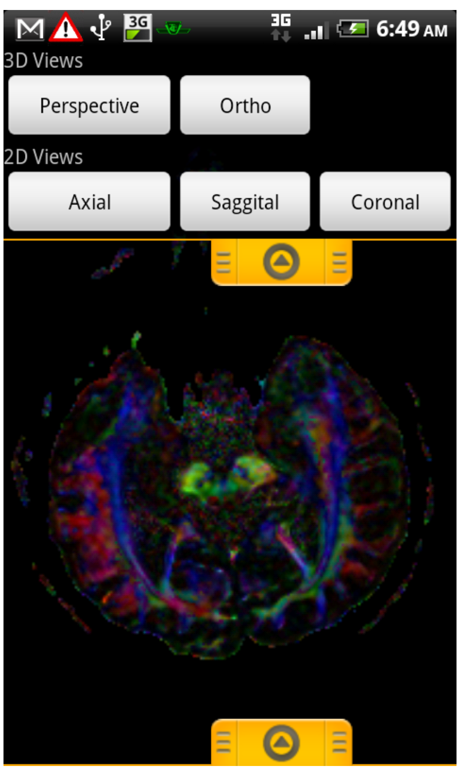


Figure 3

File format support

Scalar Volumes

Scalar volume in NRRD format has been successfully implemented. Scalar volumes specify one brightness value for every point in the volume.

DTI Volumes

DTI tensor volumes format has been successfully implemented. At each voxel the tensor is represented by a 3x3 matrix, and is visualize by assigning a color to the voxel depending on the orientation of the largest Eigen vector of the tensor matrix (Figure 3).

3D polygonal models

Polygonal models

Figure 4

Polygonal model visualization has been implemented. The model is assigned a random color for ease of visual recognition. Polygonal models allow visualization of 3D structures within the MRI volume in their full (non-sliced) form. These models are extracted off-line by a segmentation algorithm and imported into our application via standard .vtk files (Figure 4).

Tractography models

Because of unforeseen difficulties in implementing the DTI tensor volume reader that resulted in delay, the support for tractography models is not implemented because of time constraints.

What have we learned

Physical memory

Smartphones do not have very much RAM to work with. The volumes that we are trying to load can be many tens or even hundreds of megabytes in size. This was initially a concern for us, as we thought we would have to implement some kind of paging algorithm to only keep relevant parts of the input file in memory at any given time. It turns out that Android is capable of memory-mapping large files, thus allowing the entire file to appear logically as one random-access array.

Performance Bottlenecks

Surprisingly, the slowest part of our application is not the graphics or the slicing (in software) of large volumes, but the parsing of the 3d polygonal object files, which are text files. Performing regular expression parsing in Java takes a long time, even if the file is less than a megabyte.

GUI performance

We are pleasantly surprised by the GPU performance of our test phone (HTC Desire). 3D drawing was very responsive. In cases of a complex polygonal model, with 1,000,000+ vertices, the drawing performance was still very responsive.

A concern we had was the limited screen real-estate. The usage of top and bottom sliding panels allowed us to minimize screen clutter. However the Android API lacks supports for generic sliding panels, and only offers the ability to implement a sliding panel that extends from bottom to top of the screen. A third-party library was eventually used.

Multi-touch implementation was surprisingly difficult. It was a surprise to us that as one of the most marketed feature of Android smart-phones, multi-touch has rather limited touch development support. SDK 2.2 only supports pinch-to-zoom gesture, and it's difficult to implement other multi-touch gestures at the same time.

We have learnt that, when designing the UI, a more comprehensive research of the API's capacities should be done beforehand to properly assess the difficulties and plan accordingly.

Contributions

David Qixiang Chen

- Constructed the initial code framework.
- Track-ball camera
- UI and Sliding panels
- 2D and 3D views
- Aligned polygonal model to the MRI volume.
- General code fixes

Alex Rodionov

- File loaders for Scalar MRI volume (NRRD)
- File loaders for DTI MRI volume (NRRD)
- File loader for polygonal model (VTK)
- Data manager to correct for volume spatial details.
- UI slider for slice viewing.
- Volume slicing code and texture generation
- General code fixes

Future Directions

Additional features

As a MRI viewer, the application in its current form is very functional. However there is much to be improved upon.

- There are a number of popular MRI file formats (DICOM, NIFTI, etc.) that should be supported for this application to have any serious usage within research and healthcare.
- The ability to overlay multiple MRI volumes.
- Ability to select and modify attributes of 3D models
- Support for annotation and notes.
- Support for remote file access through existing MRI databases such as DICOM servers.
- Ability to seed and generate tractography models from DTI tensor data.

Field of application

The application is target at hospitals or private clinics where a portable MRI viewer can be used. It can be used on tablets or phones by the doctors for pre-surgical diagnosis and planning, as well as patient communication.

Commercialization Potential

We believe that the application has strong commercializing potential. The healthcare field is experiencing tremendous growth, and a lot of capital is going into neurosurgical related fields. We believe it’s a good resource for us to tap into. The application can be sold by bulk commercial license, and profits can also be made on maintenance contracts with hospitals and clinics. Partnerships with existing MRI manufacturers can also be made to function as a mobile extension to preexisting MRI databases.

References

Chen, D. Q., Quan, J., Guha, A., Tymianski, M., Mikulis, D., & Hodaie, M. (2011). Three dimensional in vivo modelling of vestibular schwannomas and surrounding cranial nerves using diffusion imaging tractography. *Neurosurgery*. doi: 10.1227/NEU.0b013e31820c6cbe.

Hodaie, M., Quan, J., & Chen, D. Q. (2010). In vivo visualization of cranial nerve pathways in humans using diffusion-based tractography. *Neurosurgery*, *66*(4), 788-95; discussion 795-6. doi: 10.1227/01.NEU.0000367613.09324.DA.