CMPUT 302

Project Part 1

Augmented Reality in Medicine

Project Design

Michael Feist
Michael Dardis
Ian Watts
Devon Waldon

Motivation:

Design an augmented reality system for the projection of internal systems onto a physical patient. The system would allow the user to safely interact with a patient and be able to view important internal structures while doing so.

Goals:

Current Scope:

Project an image onto a patient:

Since the system is designed to be a teaching aide, projecting an augmented image onto a real patient allows for multiple users to view the augmentation simultaneously. The system also provides both students and instructors with a tactile object to interact with, as opposed to a two dimensional image or chart. Furthermore, the projected image can be customized, allowing different views (perhaps representing the muscular system, skeletal system, etc) to be projected onto the patient, depending on the needs of the user.

Modify projection based on relative position of the projector:

Enabling the projector to move freely within the confines of the 3D space being monitored and adjusting the projected image accordingly will give us an accurate representation of the patient. This freedom allows the users to view the augmentation from any angle they choose and also permits the users to determine the physical area focused on by the projector.

Easy user operation:

The operators of this system will not necessarily be expert computer users. The setup, calibration, and operation of the system will be designed to be both easy and intuitive. This will hopefully minimize the learning curve needed to progress from a 'new user' of the system to a user capable of properly operating the augmentation.

Allow for expansion:

Since the system is a prototype, expansion of the system is important. This will allow future programmers to extend or utilize some of the work we do on the current scope of this project.

Optional Expansions:

Provide a GUI using the projector:

Note: Client has expressed strong interest in this option

Providing a set of menus and options using the projector would contain the UI to the single projector, without requiring the users to learn both the projector operation and a computer-based interface. It would also remove the need to move away from the patient to change settings during a presentation, teaching session, etc.

Allow for tracking of hand positions:

Note: Client has expressed strong interest in this option

Enabling the system to track a user's hands would enable more advanced interaction possibilities. These could include system feedback (mentioned below) as well as providing a basic framework that could be expanded to include gesture UI, multiple user identification (allowing both teacher and student to interact simultaneously), etc.

Provide system feedback:

Feedback would allow the system to monitor and/or guide user's interactions with the patient via hand placement guides, pressure feedback, or real time image feedback of manipulations done to the patient. For example, if a user applies pressure to particular places on the spine, the augmented image would shift to show the effects of the adjustment.

Interaction Design:

In order to calibrate the system, a user will first place markers on the patient, then indicate to the computer where these markers were placed using a point-and-click GUI interface. The user will also be able to use the computer to calibrate the projector using a series of numeric inputs and dials, allowing them to fine tune the projection. Any further adjustments or corrections to calibration or setup will be done using this computer GUI interface.

Once the system is calibrated, the user will be able to move freely around the room with the projector. As the user moves, the system will maintain the correct image projection, without skewing or warping the image. The portability of this design opens up several options regarding how the user 'carries' the projector. For instance, they may mount it to a shoulder or a head, without any additional software revisions.

Optional Expansions:

The next iteration will allow users to interact with a augmented-reality GUI system that will be displayed on the patient's back or on a flat surface.

System Design and Implementation:

The system will be using an Optitrack 3D Tracker system to track multiple markers in real time. This data will then be associated with points of particular objects using the ARToolKit we have been provided with. Once we know where the 3D markers are and what object they are associated with, we will be able to set up a 3D scene for rendering that accurately represents the real world.

For the initial build, we will only be focusing on two objects: the back of the patient, and the projector. The back will initially be treated as a solid object, while future iterations may involve spine motion and manipulation. The second object that will be tracked is the location and orientation of the projector. We need to know the location of both objects so we are able to accurately correct the image distortion caused by the possibility that the projector may be placed at any location in the room.

Once we have both objects properly tracked we will render the scene from the position of the projector using OpenSceneGraph. Initially, our system will require the patient's back to be scanned into a 3D model. This will be done either using a Kinect or some other professional grade scanner. Once the patient is scanned, we will use a triangulated 3D model of the patient in our rendering. Doing so will correct all image warping.

Even though we can correct the image warping, the projector will still have issues with focus. To fix this issue we will be using a laser projector that will keep the image in constant focus even when the surface is not flat.

After we have proper tracking of the projector and spine we will create a simple GUI that a user can use to calibrate the system. Initially, the GUI will be very simple and used mostly to calibrate the projector. The final goal is to have a user interface that allows users to place marks on a patient and then align the markers with a given image. This means that the user could, for example, place a marker on the fourth vertebra and would then be able to click on the fourth vertebra on the image.

After the initial build is working we will then start focusing on tracking the hands. At first we will only be tracking the finger tips but the end goal is to track how the fingers are bent. Once we have accurate finger tracking, we will be able to move the GUI from the laptop screen and place the menu right on the patient's back or the table that they lie on. This will allow the user to select and update information without having to leave to use a computer.

OptiTrack, ARToolkit System Development:

Step 1: Calibrate the OptiTrack system (OPT)
(1 Day)

Step 2: Track a point (OPT)
(1 - 3 Days)

Step 3: Track multiple points (OPT)
(1 - 7 Days)

Step 4: Group points as objects (ART)
(3 - 7 Days)

Step 5: Get transformation matrix from marker (ART)
(3 - 7 Days)

Testing Tracking:

To test tracking we will start tracking an object and monitor the coordinates coming from system. When we move or rotate the object we will verify that the coordinates change as expected.

OpenSceneGraph and Rendering System Development:

Step 1: Install proper libraries and set up a simple 3D scene in Visual Studios
(1 - 2 Weeks)

Step 2: Load custom 3D object (.obj)
(3 – 7 Days)

Step 3: Load textures and project them onto object
(3 – 7 Days)

Step 4: Able to manipulate Camera in 3D space to accurately represent the properties of the projector
(3 – 7 Days)

Step 5: Build a GUI HUD so users can calibrate the projector.
(1 – 2 Weeks)

Step 6: Build a GUI HUD so users can align markers.
(1 – 3 Weeks)

Step 7 (Optional): Build a GUI HUD to display on patient or on surface.
Note: Client has expressed strong interest in this option
(1 – 2 Weeks)

Testing The Projection:

To test the system and make sure the image is correctly projected onto the patient with minimal to no warping, we will project a grid onto the intended surface. We can then manually measure the lines of the grid to ensure that each square is indeed square. If warping occurs then our squares will become rectangles or trapezoids, and further corrections can be made.

User Study:

We plan to study the effectiveness and 'intuitiveness' of our user interface by asking volunteers to attempt to operate our system with minimal instructions (outside of the GUI), and noting the steps or areas in which they become confused. We will also allow them to become familiarized with the system, monitoring both their struggles and the areas they find relatively straightforward.

Following the test subject's time with the system, we will issue them with a survey, which will utilize a combination of numerical and comment based questions. This will to allow us to both plot quantitative data as well as receive qualitative user responses. Additional responses will be obtained with the comment sheet, where the participants can mention additional information they feel we would find important.

Language:

C/C++

Developer IDE:

Visual Studios 2010 on Windows 7

Libraries:

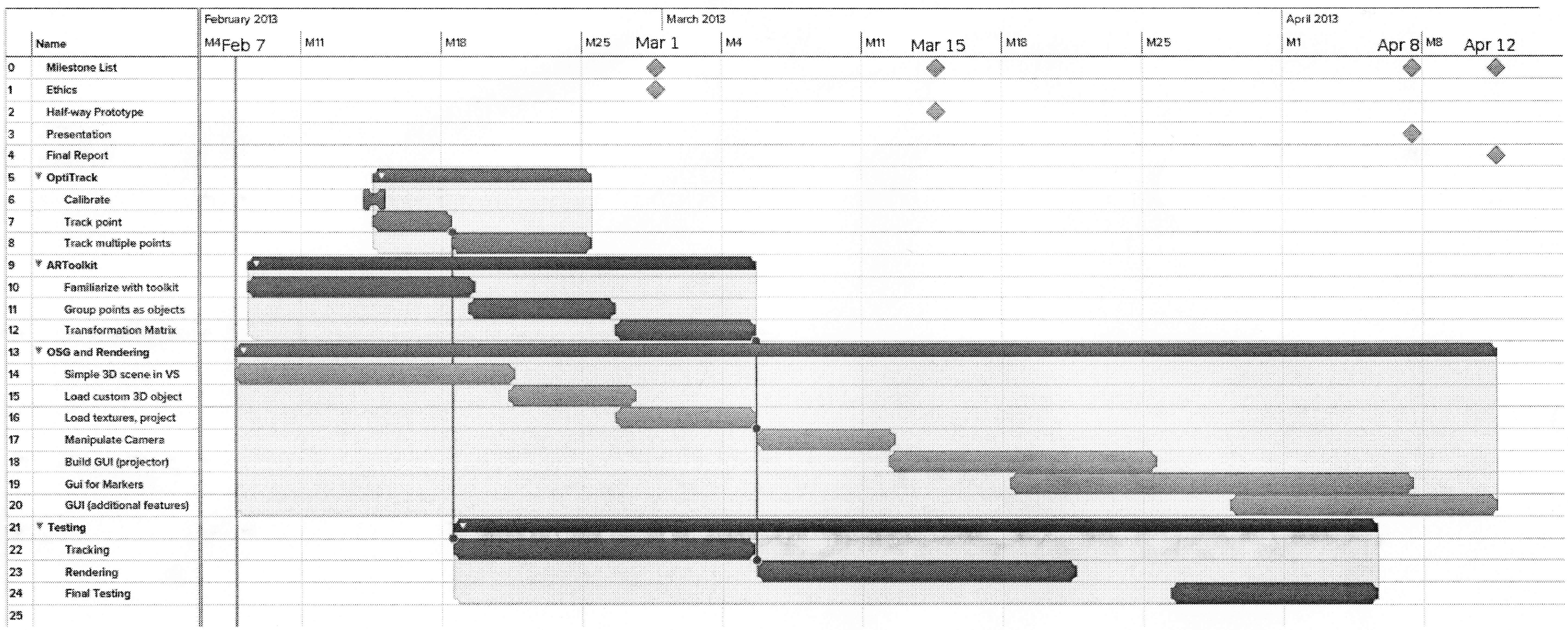
OpenSceneGraph
ARToolKit
osgART (requires ARToolKit and OpenSceneGraph)

Hardware:

Pico Projector
OptiTrack with markers
Laptop

Documentation:

http://www.stackedboxes.org/~lmb/en/computer-stuff/asittbpo-open-scene-graph/index
http://www.openscenegraph.org/projects/osg/wiki/Support
http://www.openscenegraph.org/projects/osg/wiki/Support/Tutorials
http://www.artoolworks.com/support/library/osgART_tutorial_1:_First_simple_osgART_scene
http://www.artoolworks.com/support/library/osgART


February 2013
March 2013
April 2013
Name
M4 Feb 7
M11
M18
M25
Mar 1
M4
M11
Mar 15
M18
M25
M1
Apr 8
M8
Apr 12
0 Milestone List
1 Ethics
2 Half-way Prototype
3 Presentation
4 Final Report
5 OptiTrack
6 Calibrate
7 Track point
8 Track multiple points
9 ARToolkit
10 Familiarize with toolkit
11 Group points as objects
12 Transformation Matrix
13 OSG and Rendering
14 Simple 3D scene in VS
15 Load custom 3D object
16 Load textures, project
17 Manipulate Camera
18 Build GUI (projector)
19 Gui for Markers
20 GUI (additional features)
21 Testing
22 Tracking
23 Rendering
24 Final Testing
25