

Improving User Experiences in Indoor Navigation with Augmented Reality

Andrew Zhong

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2014-74

<http://www.eecs.berkeley.edu/Pubs/TechRpts/2014/EECS-2014-74.html>

May 15, 2014



Copyright © 2014, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

University of California, Berkeley College of Engineering

MASTER OF ENGINEERING - SPRING 2014

Electrical Engineering & Computer Sciences

Visual Computing and Computer Graphics

Improving User Experiences in Indoor Navigation with Augmented Reality

Xuanyu Zhong

This **Masters Project Paper** fulfills the Master of Engineering degree requirement.

Approved by:

1. Capstone Project Advisor:

Signature: _____ Date _____

Print Name/Department: **Björn Hartmann / Electrical Engineering & Computer Sciences**

2. Faculty Committee Member #2:

Signature: _____ Date _____

Print Name/Department: **Donald Wroblewski / Fung Institute for Engineering Leadership**

Improving User Experiences in Indoor Navigation with Augmented Reality

“Andrew” Xuanyu Zhong

Univeristy of California, Berkeley

Abstract

In today’s world, we take advantage of positioning signals from the Global Positioning System (GPS) which navigates to our destinations through the best routes possible. However, this advantage is only valid outdoors. GPS does not work well indoors due to signal attenuation from reflections across indoor surfaces [1]. With regard to this, a number of alternative technologies have been developed for indoor navigation. They are mostly based on Wi-Fi, infrared, or Bluetooth, with an accuracy level of several meters which is suitable for basic indoor navigation [2].

On the other hand, the augmented reality technology (AR) enables applications to generate and align a supplemented view of the real world. It requires high positioning accuracy which could be achieved by object tracking combined with sensory data. In this project, we incorporated augmented reality into indoor positioning and designed a novel user interface with a compelling application that delivers an interactive indoor navigation experience through augmented graphical views aligned with indoor objects. It is demonstrated in the CITRIS Invention Lab at Berkeley and works by providing augmented navigation, interactive device instructions, demo products display and device reservation lookup for lab visitors and apprentices. It runs on the Android mobile platform powered by OpenGL graphics, orientation sensors and an indoor position simulator. The user study shows that 80% of users prefer our augmented reality based user interface.

Introduction

The augmented reality (AR) technology enables applications to generate and superimpose a supplemented view of the physical environment in graphics, based on sensory input such as camera video stream, orientation and location. Simply put, it computes and aligns the generated virtual space with the real world. The term augmented reality has existed for more than 20 years. Since then, prototypes of augmented display systems have been introduced by research institutions in technical conferences. It was not until the last decade that we saw a boom of AR-based applications and devices showing up on the market. They come with exciting developments in computer vision algorithms, scalable computing power as well as mobile graphics. These AR applications mainly run and perform better on large-screen smartphones and tablets, which according to IDC [3] will account for 87% of connected devices sales by 2017. In a world where computing devices are going mobile and having larger displays, it is reasonable to expect augmented reality to mature and thrive.

Indoor navigation serves as a perfect use case for augmented reality. The finer-grained indoor positioning will provide important location domain knowledge as an input to the AR module. At the same time, it will benefit from precise alignment of augmented graphics and provides more interactive user experiences. In this project, we partnered with Qualcomm and took advantage of its indoor navigation technology based on Wi-Fi access points. It will give us a positioning accuracy of 2 – 5 meters, surpassing positioning by GPS or mobile networks.

The prototype to be discussed in this report uses a generic simulated location API and will transfer to Qualcomm’s indoor navigation API once it becomes available.

Our prototype application is based on a use case of showing or navigating in an indoor area, such as a museum, shopping mall or laboratory. When up and running, it overlays an augmented graphics layer on top of the camera view. The graphics layer can be text, images or 3D models. We managed to deploy our application in the CITRIS Invention Lab at Berkeley, which provides students with an innovative environments for rapid prototyping. The goal of the application is to provide augmented indoor navigation to lab equipment, interactive step-by-step instructions of popular devices such as 3D printers and laser cutters as well as device reservation lookups. Figure 1 shows a screenshot of our application in navigation mode where an augmented arrow will navigate to your device of interest.

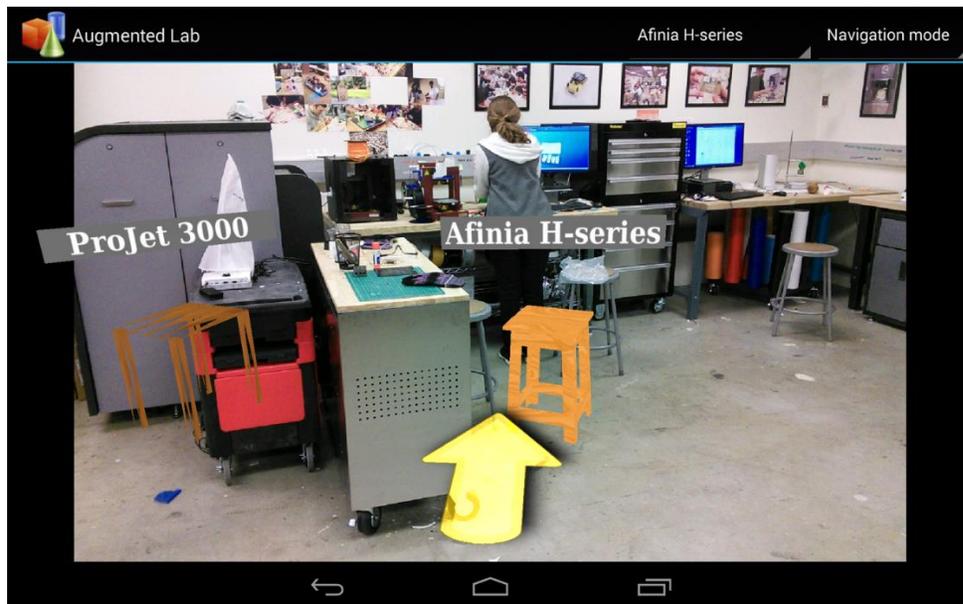


Figure 1. Our prototype application in navigation mode deployed in CITRIS Invention Lab

Users will aim their phone or tablet camera towards the indoor environment and receive augmented information overlaid on the camera view. Figure 2 shows how an Invention lab user interacts with our application.



Figure 2. An Invention lab user interacting with our application

Literature Review

Our application embraces both indoor navigation and augmented reality, which is rarely seen in existing navigation applications. Here we compared our application with several existing applications and theories.

Google Map has been a solid product in terms of both outdoor and indoor navigation. According to Google, indoor maps and navigation has been activated in more than 250 venues (e.g. airports, shopping malls, and universities) in the US and over 10,000 floor plans are available throughout the world [4]. For indoor positioning, it uses similar Wi-Fi technologies based on access points and therefore has comparable precision compared with our application. However, Google Map does not feature augmented reality and provides a plain 2D map for navigation. Its user interface is less interactive than our application.

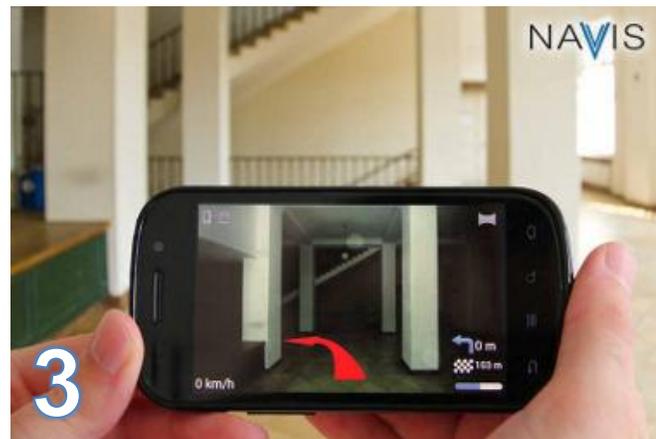
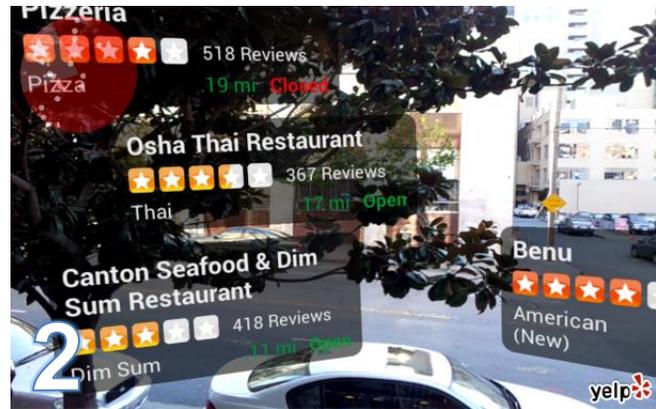
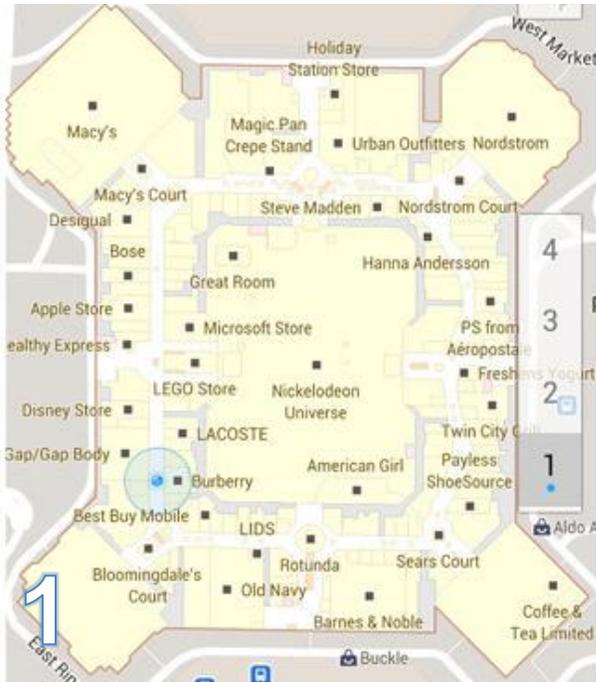


Figure 3. Existing navigation applications:
1 – Google indoor map (indoor navigation w/o AR)[5]
2 – Yelp Monocle (Outdoor Navigation w/ AR) [6]
3 – NavVis (Indoor Navigation w/ AR) [7]

Yelp Monocle integrated augmented reality and provided an interactive user interface on top of camera view. It uses GPS, compass and orientation sensors from mobile phones to

navigate. It is able to overlay the nearby entities such as restaurants and shops on the camera view. However, the GPS technology put a limit of 5-50 meters on its navigation precision and it does not work well indoors because of signal attenuation from reflections across indoor surfaces [1].

Unlike most positioning systems, NavVis does not rely on wireless signals to deliver positioning service. Instead, it computes the user's exact location by taking a photo of the surrounding environment and matching it with the image database [8]. Instead of tracking fiducial markers, it provides a generic approach of high precision indoor positioning through computer vision.

The augmented reality technology enables applications to generate and superimpose a supplemented view of the physical environment. In the area of indoor navigation, AR technology provides a new approach to improve accuracy of existing wireless navigation systems. By detecting objects in the video stream and computing the object's coordinates and orientation, the application is able to set up an augmented virtual coordinate system. Figure 4 shows an augmented reality application which is designed for searching for books in a library [9]. It detects optical markers on shelves, sets up a virtual coordinate system and computes the viewer's location coordinates. Although the location is relative, if combined with wireless positioning technologies, the virtual coordinates could be one-to-one aligned with the real world.

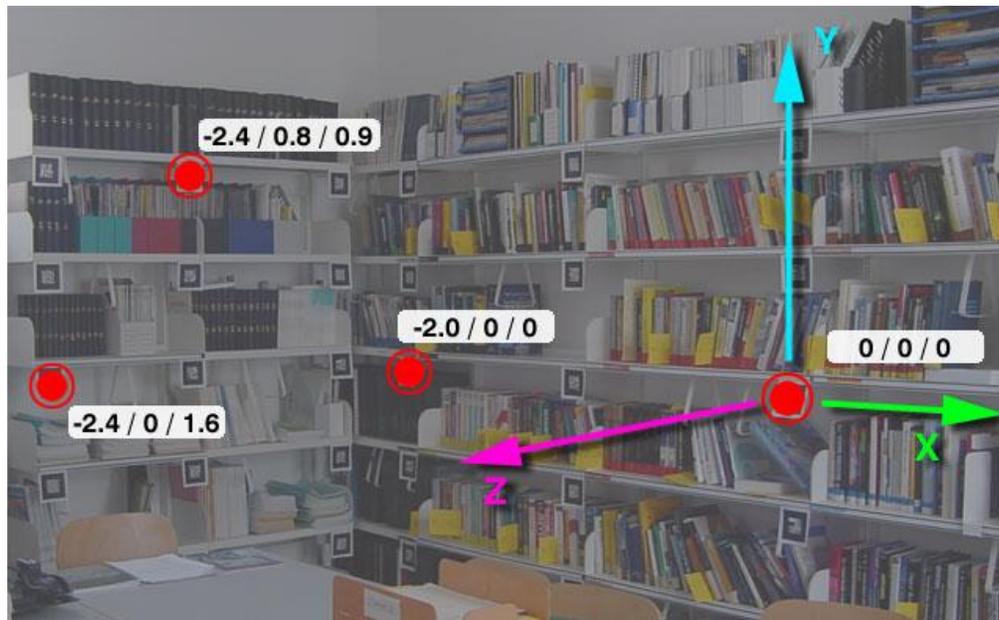


Figure 4. AR Library – aligning a virtual coordinate system in library for searching purpose [9]

Methodology

We use Android 4.3 as our development platform. The technical framework of our application is presented in Figure 5. It comprises of three main modules, graphics rendering, positioning and orientation, as well as user interface.

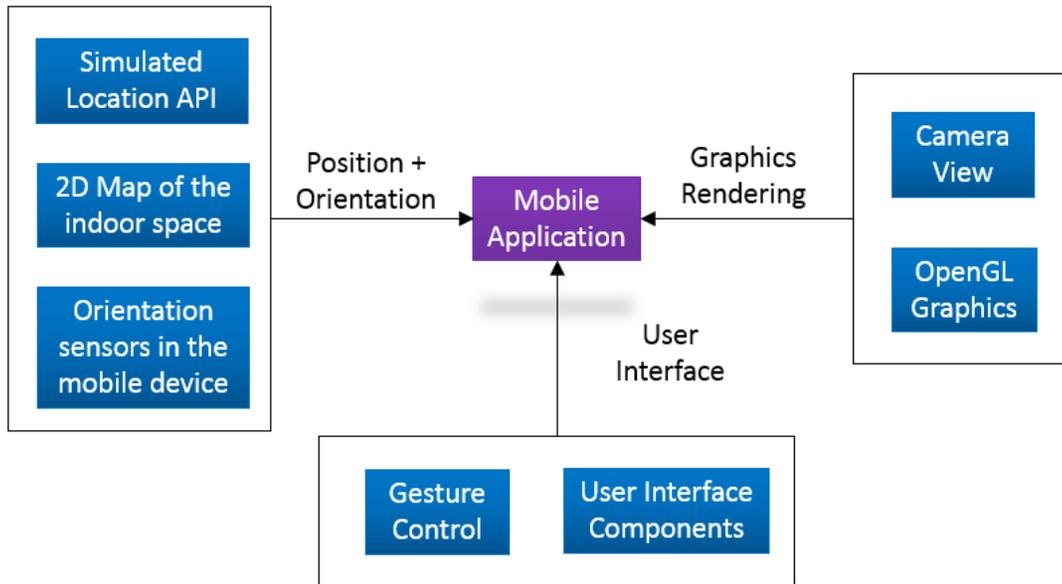


Figure 5. Technical framework of our application

Graphics Rendering

For the graphics rendering module, we want to render a graphics layer on top of live camera video streams. The graphics layer is rendered through the OpenGL ES 2.0 graphics library. Figure 6 gives an overview of the OpenGL rendering pipeline. We implemented each of the following stages for all objects to be shown on screen.

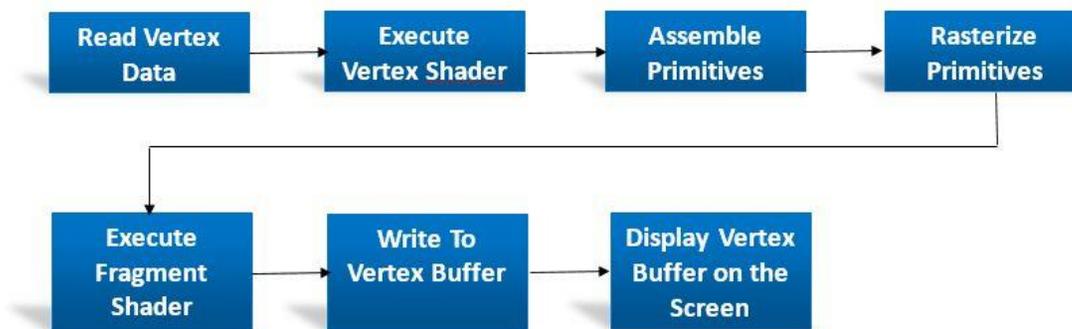


Figure 6. OpenGL rendering pipeline [10]

We either hand-created the 2D/3D models by creating vertices and mapping textures or imported them from existing Wavefront .obj models. We also programmed multiple vertex and fragment shaders in GLSL for non-photorealistic rendering that customizes on color, transparency and texture mapping. OpenGL renders all the graphics on a GLSurface in Android. When the GLSurface is created, vertex and fragment shaders are compiled individually and then linked together into a shader program. Textures to be mapped are loaded here. During every drawing cycle, all shader programs are activated, vertex data gets bound to their respective shader program and the modelViewProjection matrix (product of model, view and projection matrices) is passed on for vertex transformations. Texture coordinates are also mapped with vertex coordinates here.

Position and Orientation

In order to align the graphics layer with the real world, we need to obtain current position and orientation of the phone or tablet. We managed to extract real-time orientation from both accelerometer and magnetometer. The main program activity listens to sensor change events and during each event, it records the azimuth, pitch and roll angles of the phone or tablet and remap the rotation matrix to the most recent coordinate system. Furthermore, we added a low-pass filter to stabilize the sensor outputs. We then updated the view matrix (specifically lookat location, camera up and camera right vector) according to the three filtered angles and

recalculated the modelViewProjection matrix, so that only the region that the camera is pointing at gets rendered on the screen.

As mentioned before, since Qualcomm’s indoor positioning API was not available to us, we looked for alternative positioning service providers. At first we tried Google’s positioning API which is based on GPS and mobile networks. It turned out as expected, its accuracy level was not high enough for indoor use. Therefore we created a web application which lays out an indoor map and simulates user’s indoor position coordinates in latitude and longitude.

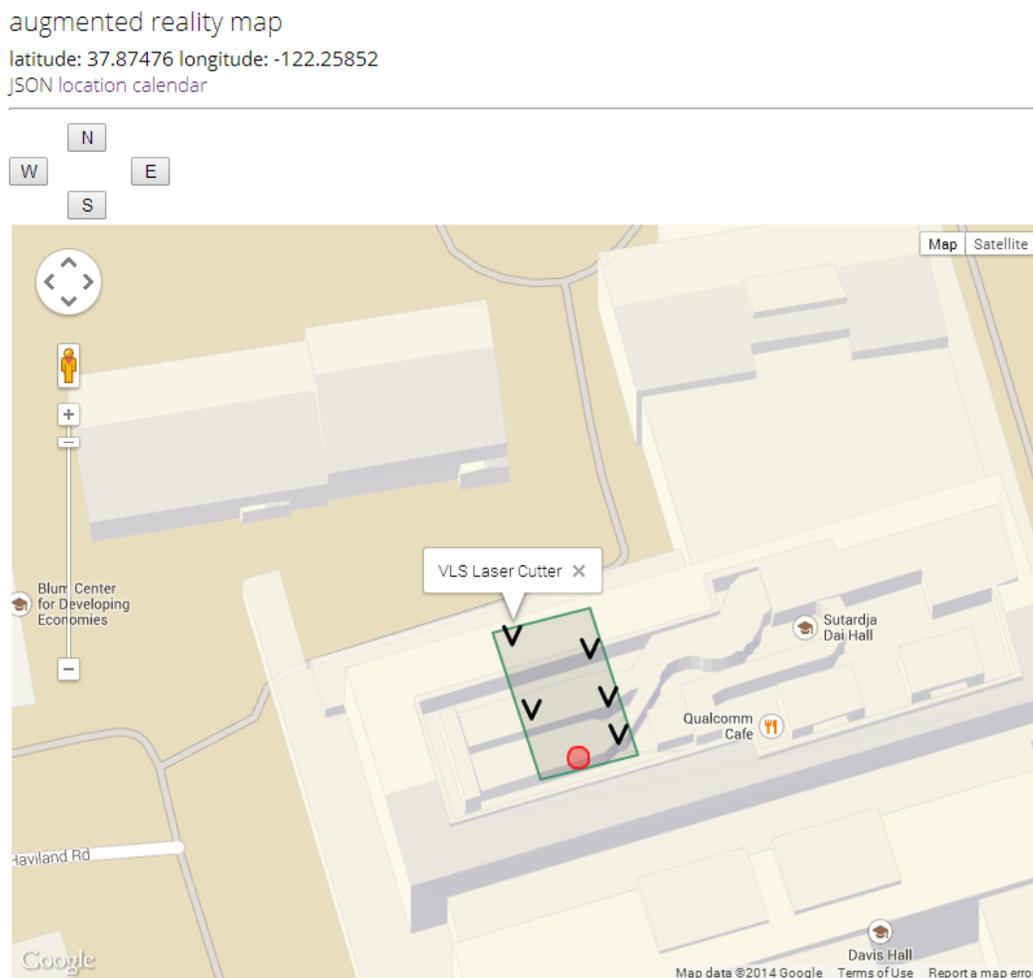


Figure 7. Simulated indoor positioning

The simulated map is shown in Figure 7. The green boundary marks the CITRIS Invention Lab at Berkeley, while the black arrows denote different devices in the lab. The red dot marks the current simulated indoor position. The user can select another location simply by clicking on the map or pressing the arrow keys on the keyboard. The front end, written in JavaScript and HTML, also uses Google Map API for the simulated map. The backend is built in Python on Google App Engine. Our Android application parses the latest coordinates from the simulated map and treats it as from Qualcomm’s indoor positioning API.

User Interface

We targeted a specific user scenario for this project, the CITRIS Invention Lab, where students practiced rapid prototyping on 3D printers, laser cutting tools and other devices. We conducted initial interviews with lab managers, students and visitors to recognize frequently used devices and relevant use cases. From these interviews, we identified two types of users: students who operate these devices and visitors who take a lab tour to check out lab devices and students’ works. Therefore we came up with four use cases, each of which serves as one mode in the application: apprentice mode where users learn to operate a selected device, calendar mode where users check the schedule of each device, visitor mode where visitors views previous students’ works and navigation mode where a virtual arrow navigates to a selected device.

Our initial user interface was designed through a rapid wire framing tool called “Balsamiq Mockups” [11]. We took photographs of the CITRIS Invention Lab and collected text instructions for all devices of interest to be shown in our application. With these, we came up with a clickable mockup user interface in pdf format. It demonstrates the control flow between the four modes of the prototype application: apprentice mode, calendar mode, visitor mode and navigation mode.

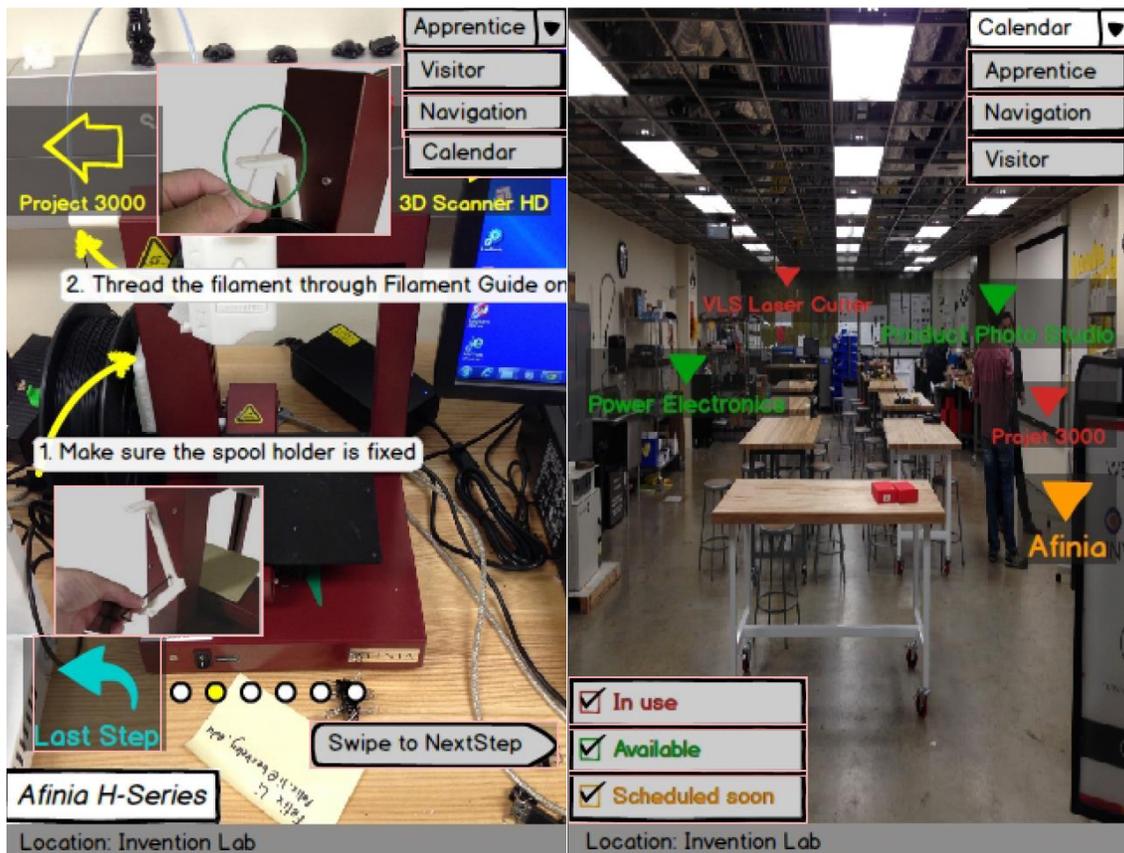


Figure 8. Mockup user interface: apprentice mode (left), calendar mode (right)

As shown in Figure 8, the apprentice mode in the mockup presents an interactive step-by-step instruction in how to operate a specific device. In the calendar mode, arrows are rendered and overlaid on devices, whose colors are determined by availability of the devices

they are pointing to. Figure 9 shows the mockup user interface for visitor mode and navigation mode. The visitor mode presents an interactive gallery of students’ works on the selected device while in the navigation mode virtual arrows are rendered for navigation purpose.

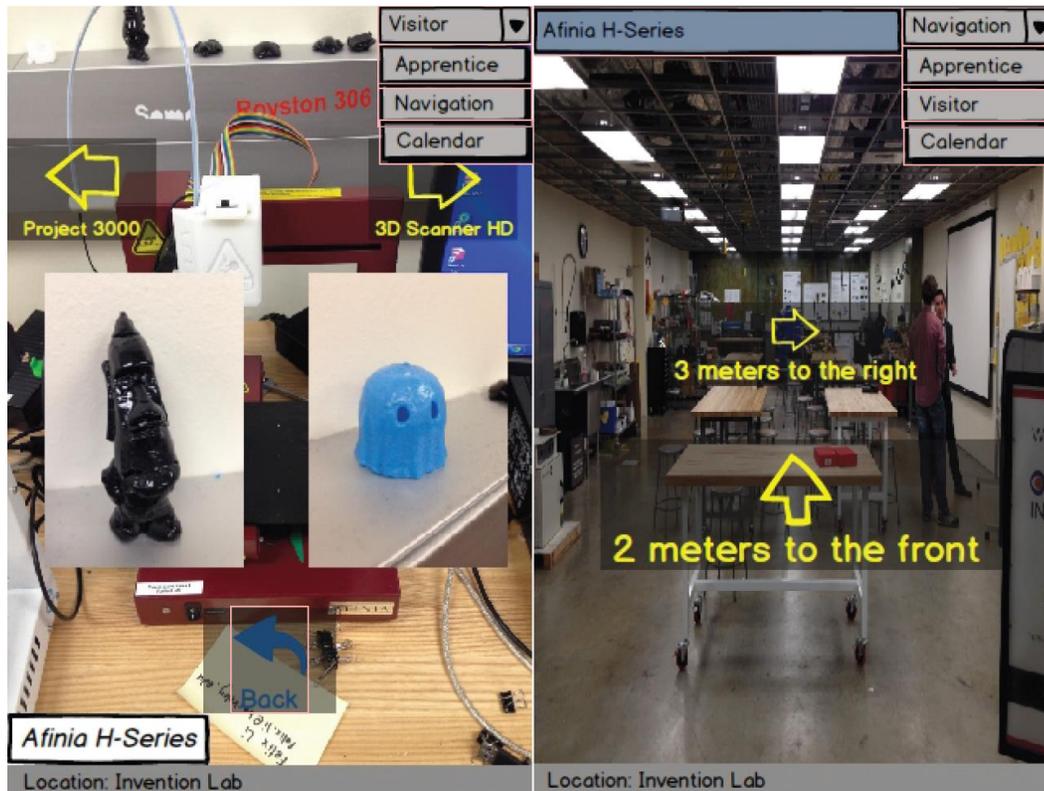


Figure 9. Mockup user interface: visitor mode (left), navigation mode (right)

We performed user testing on the mockup user interface to validate our understanding of the four use cases. The users were randomly recruited from current lab users and veterans. After having a hand-on experience with our mockup UI, they were mostly excited about the four modes and provided detailed feedback on laying out certain UI elements. Once the use cases were validated, we started to implement the user interface on the Android platform.

Results and Discussion

Graphics Rendering

We managed to create several modules for loading different graphical models. This includes rectangular textures, general 3D textured objects and non-textured simple geometries (circles, cylinders and cones). Objects of the same type could be easily grouped together, say an arrow is made of a cone on top of a cylinder. A screenshot of our application rendering the three types of objects is shown in Figure 10. In the screenshot, the small picture in middle top was rendered as a rectangular texture. The chair was a 3D textured object while the arrows were made from non-textured geometries.



Figure 10. Graphics layer made of rectangular textures, 3D textured objects (chair) and non-textured arrows

The graphics scene is pre-loaded into the application from a text input file. It specifies the number of all three kinds of objects, texture or color, displacement and rotation of each individual object. For 3D objects, we transfer each of their Wavefront .obj file to an individual header file. We simplify faces in one obj file to a group of triangles to draw and write their vertex coordinates in order into a header file. To load one 3D object, the android application will parse the header file corresponding to the object and draw its group of triangles in sequence. The loading time is around 5 – 10 seconds on Android 4.4 depending on the complexity of object files. Because the triangles mostly share their vertices with two or more other ones, the header file contains duplicate coordinates. This means that by reducing duplicates in the header file, the loading time could be further reduced in future.

Once loaded, the graphics layer will align with the corresponding real world objects. Changes on the current indoor position will update the augmented scene in a way that the graphics view always aligns with the real objects (Figure 11).



Figure 11. Augmented scene based on user's indoor position

Position and Orientation

The position and orientation module worked as intended. We implemented location fetching by having a separate thread parse the simulated location every 0.1 seconds. The orientation extracted from both accelerator and magnetometer tend to be unstable on both tablet and phone. Therefore we further adapted cutoff frequency of the low-pass filter to mitigate the instability.

User Interface

Our user interface consists of four modes: navigation, apprentice, visitor and calendar.

Figure 12 presents the navigation mode. A user can select his or her device of interest from a

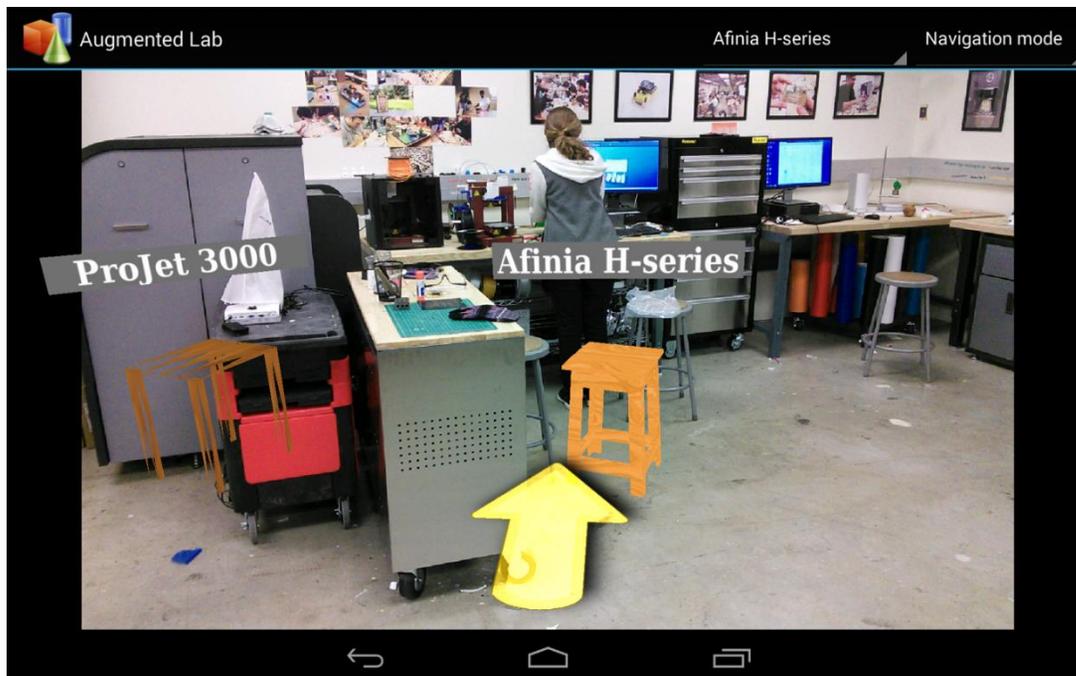


Figure 12. Navigation mode – navigate to “Afinia H-series”, a 3D printer

dropdown menu. An augmented arrow powered by OpenGL graphics will navigate to that specific device. The direction vector is computed by subtracting location coordinates of the phone or tablet from those of the device of interest.

When touching one device on the camera view, a user can select either apprentice mode or visitor mode for that device. Apprentice mode will start with a set of step-by-step instructions. The instructions are in swipeable slides and implemented with Android ViewPager. Figure 13 shows the first slide of instructions for the laser cutter in the Invention Lab.

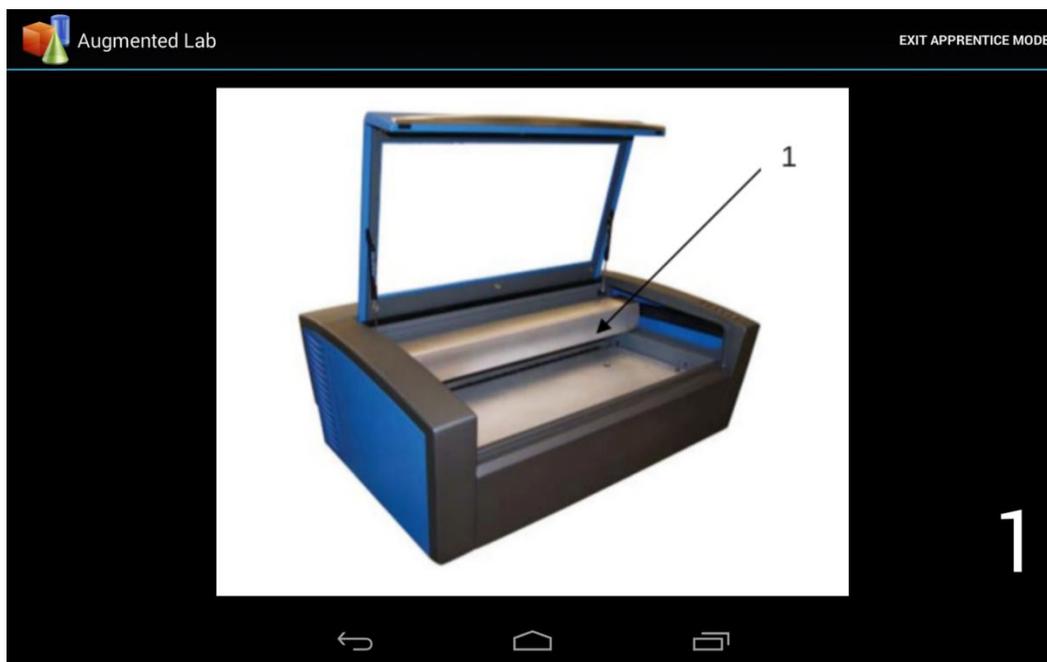


Figure 13. Apprentice mode - swipeable slides of instructions for the laser cutter

One can also select visitor mode by touching. This mode will render previous students' works on this device. In Figure 14, a 3D model of a dinosaur was rendered in visitor mode of

“Afinia H-series”, the 3D printer. Users can also rotate, zoom in and zoom out the model to get the best viewing angle.

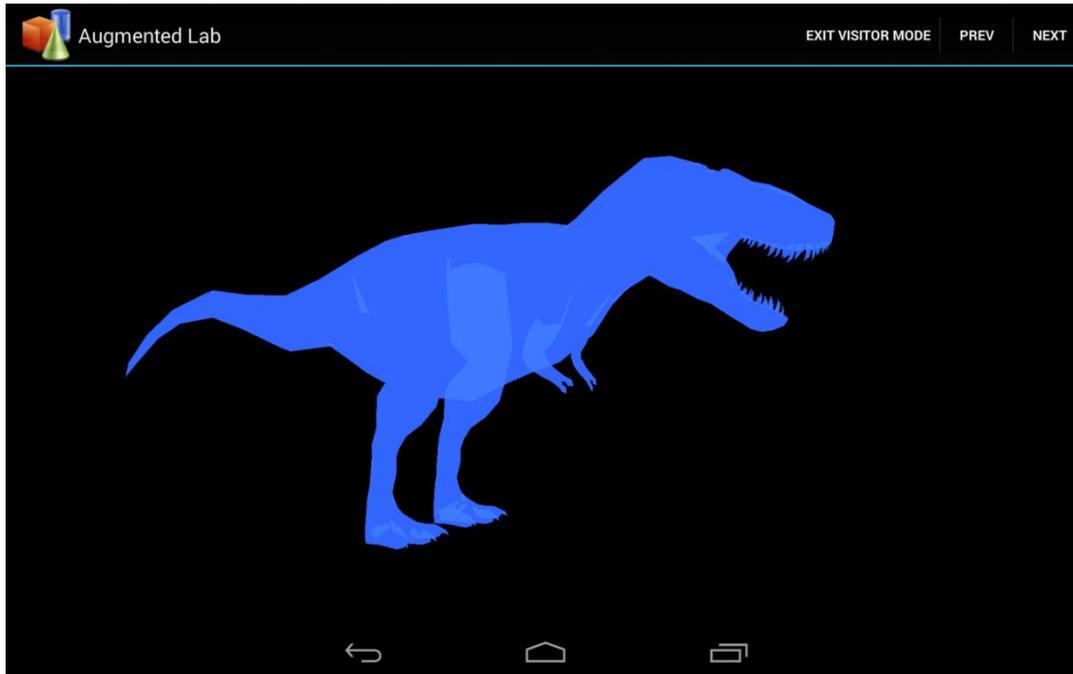


Figure 14. Visitor mode – 3D model of a dinosaur printed out on “Afinia H-series”, a 3D printer

There is also a calendar mode where users can check the reservation status of all devices. As shown in Figure 15, augmented arrows are aligned with each device and their colors notify the availability. By touching a device on the camera view, users will be directed to the calendar webpage for that device.

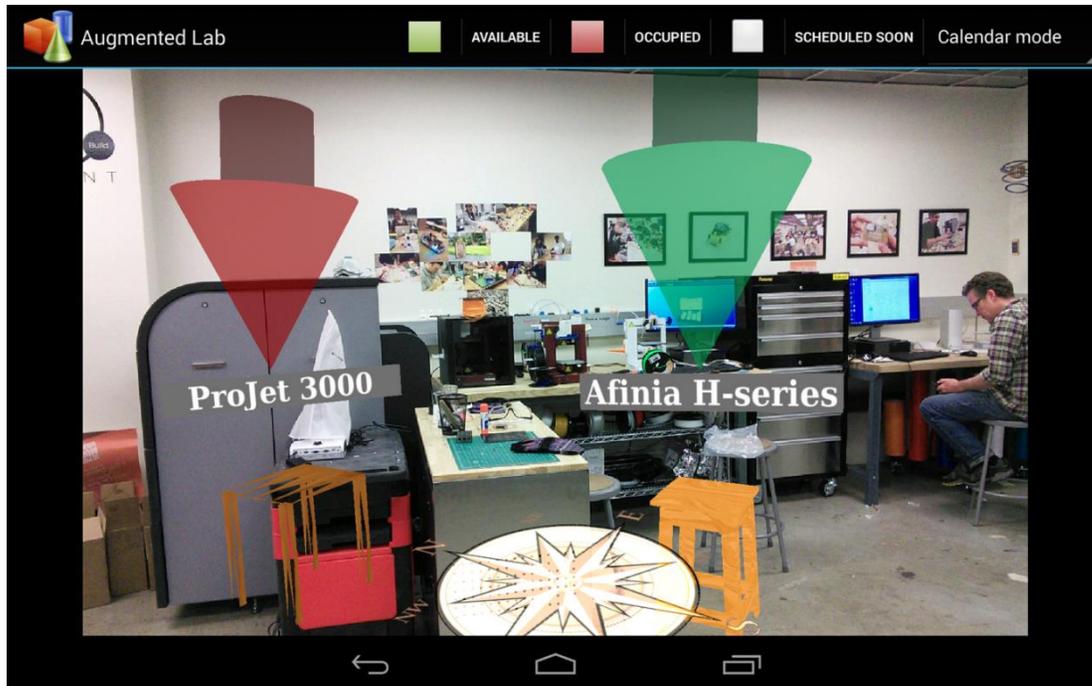


Figure 15. Calendar mode - reservation lookup returns augmented colored arrows

We also conducted two rounds of in-depth user study. The first round targeted our mockup user interface and involved 14 Invention Lab users. We asked about their preference on all four modes that we have in the mockup user interface. As presented in Figure 16, not surprisingly lab users prefer the apprentice modes the most. Most of them (8 out of 14) pointed out that this mode provides an easier way to look up procedure flow and instruction details. Calendar mode, which allows them to check device reservations, is the next favorable choice. On the other hand, visitor mode and navigation mode might not be quite relevant to existing users. They serve the needs of new visitors better. Regarding the use of augmented reality, people like it mostly. Two of them brought up that a plain map could be cleaner compared with the augmented view. We agree on that and in the next version of our application, we plan to add another “plain” view in addition to the augmented layer.

User Preferences on Different Modes on the Mockup User Interface

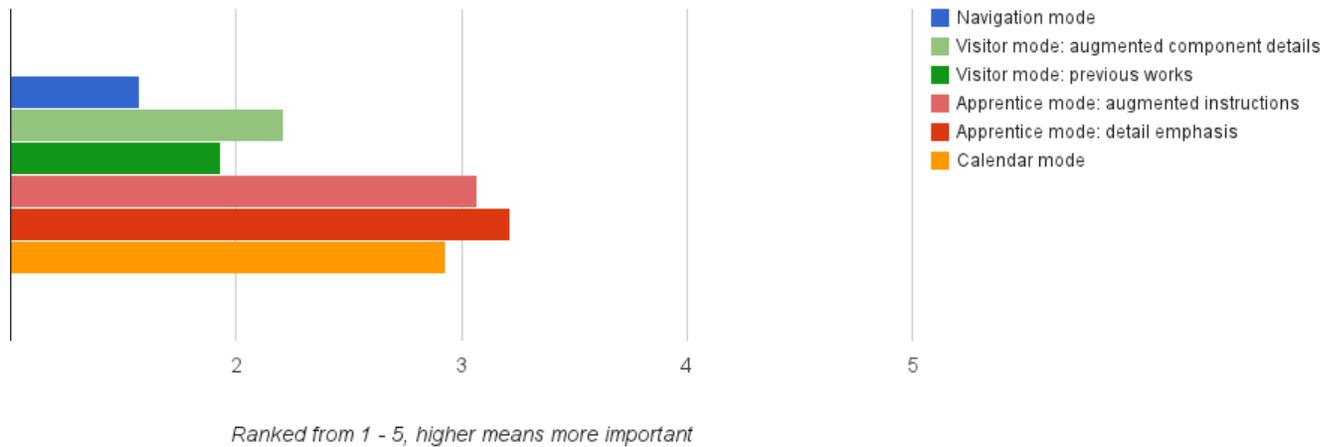


Figure 16. Users’ preferences on different modes of our mockup user interface

The second round of user study targeted the real prototype application on Android and involved 10 lab users. As illustrated in Figure 17, the results roughly match the first round of user study on the mockup user interface. Lab users prefer apprentice mode and calendar mode more than the other two modes, while navigation mode gets more favor over visitor mode.

User Preferences on Different Modes on the Prototype Application

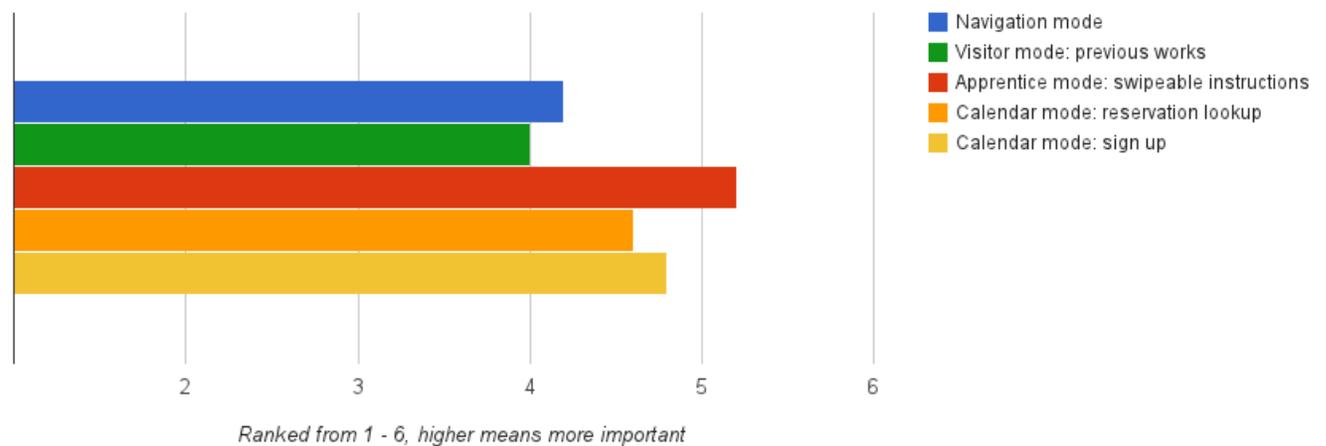


Figure 17. Users’ preferences on different modes of our prototype application

Overall, 80 percent of users prefer our application, while 50 percent of users are looking for better interactivity and 37.5 percent want extra functionality, such as more specific instructions and detailed checklists in apprentice mode.

Conclusion

Although augmented reality may still be in its infancy, through this project, we explored the enormous potential for AR applications in the use case of indoor navigation. We feel very excited to be involved in such a promising technology and look forward to deliver a high-quality prototype AR application for the CITRIS Invention Lab.

At the end, we would like to thank our faculty advisor, Prof. Björn Hartmann, Prof. Donald Wroblewski, our industry advisor Dr. Jiajian Chen and our PhD student advisor Ben Zhang for all your kindly help through the year.

References

- [1] Liu, Hui, Houshang Darabi, Pat Banerjee, and Jing Liu. "Survey of Wireless Indoor Positioning Techniques and Systems." *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)* 37.6 (2007): 1067-1080.
- [2] Farid, Zahid, Rosdiadee Nordin, and Mahamod Ismail. "Recent Advances in Wireless Indoor Localization Techniques and System." *Journal of Computer Networks and Communications* 2013 (2013): 1-12.
- [3] International Data Corporation.
Tablet Shipments Forecast to Top Total PC Shipments in the Fourth Quarter of 2013 and Annually by 2015. N.p., n.d. Web. <<http://www.idc.com/getdoc.jsp?containerId=prUS24314413>>.
- [4] "Indoor Maps Availability." N.p., n.d. Web. 14 Apr. 2014.
<<https://support.google.com/gmm/answer/1685827?hl=en>>.
- [5] Google Indoor Map. Digital image. N.p., n.d. Web.
<http://tctechcrunch2011.files.wordpress.com/2011/11/moa_before-after.jpeg>
- [6] Yelp Monocle. Digital image. N.p., n.d. Web.
<<http://esportsbusiness.com/wp-content/uploads/2013/02/yelp.png>>.
- [7] NavVis. Digital image. N.p., n.d. Web.
<<http://cdn.phys.org/newman/gfx/news/2012/1-improvingpos.jpg>>.
- [8] "Improving Positioning Indoors with Imaging Data." N.p., n.d. Web. 14 Apr. 2014.
<<http://phys.org/news/2012-09-positioning-indoors-imaging.html#jCp>>.
- [9] "Studierstube Augmented Reality Project." *Studierstube Augmented Reality Project*. N.p., n.d. Web. 14 Apr. 2014. <<http://studierstube.icg.tugraz.at/projects/mobile/arlibrary/arlibrary.html>>.
- [10] Brothaler, Kevin. "Defining Vertices and Shaders" *OpenGL ES 2 for Android: A Quick-Start Guide*, Dallas, Texas: The Pragmatic Bookshelf, 2013.
- [11] *Balsamiq Mockups*. Computer software. N.p., n.d. Web. <<http://balsamiq.com/products/mockups/>>.