

Virtual Environments for Testing Location-Based Applications

Raihan Masud, Stephen Fickas

Department of Computer & Information Science
University of Oregon, Eugene, 97403, USA
{raihan, fickas}@cs.uoregon.edu

ABSTRACT

Location-based applications often require field testing before launching the product. Setting up "the field" to produce various testing scenarios is often a logistical nightmare; real environments (e.g., an urban core) do not bend easily to controlled experiments. As an alternative, we wanted to explore the possibility of using virtual environments (VE) to test location-based applications. In this paper, we describe the construction of a virtual environment and the process of testing a location-based application for exiting a bus at a specific stop. Our approach is to start with an existing VE, Google StreetView, and then extend it to our needs. We discuss both the VE, and our efforts to connect it to an existing phone app.

Author Keywords

Virtual environment, field testing, location-based application

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Location-based applications [1,2,3] are becoming very popular to help travelers to be aware of the surrounding environment, and be more informed. Some of these applications also help travelers to navigate successfully. All these location-based applications are very helpful to get the sense of "Where am I?" and "What is around?" However, testing these applications is not always easy and time efficient. Applications that help with physical navigation [4,5] take a lot of time and effort to test in the field, i.e., in the actual situations they will be used. This led us to

explore the possibility of testing location-based applications in virtual environments.

In this paper, we describe the construction of a virtual environment, and a process for testing a location-based application in this virtual environment. This virtual environment was built on top of Google StreetView. We also built a location-based application, "MyStop" that had already been tested in the field but now we wanted to test in the virtual environment. MyStop reminds a traveler to get off the bus at the next stop. We demonstrate a bus riding scenario to describe the testing process for this mobile application in our virtual environment.

RELATED WORK

The real world environment can be represented with a computer programmed display to allow users to interact and explore in the synthetic world. The value of virtual reality applications for risky situations such as flight training [6] or teaching surgeons complex procedures [7] is well recognized. Turning to our interest in travel assistance, Joanne Lloyd et. al. [8] performed a study of route learning task and compared the performance of the subjects in VE with the real world subjects' performance. The result indicated equivalence between the real and virtual environments, with comparable error rates and no differences in strategy preferences.

Arthur and Hancock [9] performed a study for navigation training in virtual environments. This study unveils that spatial knowledge acquisition from navigation in VE can be similar to actual navigation when viewing condition is unconstrained.

Ruth Conroy in her PhD dissertation [10] on Spatial Navigation in immersive virtual environments tried to investigate whether we move through virtual worlds in a manner that is analogous to our behavior in the real world. In one experiment she compared the virtual navigation data to the movement observations of people made in the real visit of the indoor setup of the London Tate Gallery. This experiment found strong statistical correlation between these two data sets and concluded that in the pattern of pedestrian movement there is relationship between real and virtual space.

All these studies [8,9,10] show that participants actually exert similar kind of behaviors when put in to the virtual environments. This finding is a strong catalyst for setting up a virtual environment to test location-based applications and expect test results in virtual environments to be similar to the field test.

MyStop

MyStop is part of a mobile phone application to help travelers make successful transitions along an itinerary. The application is loaded and tested on a Motorola Droid phone. Travelers wear an ear bud connected to the phone, and the reminders are given by voice.

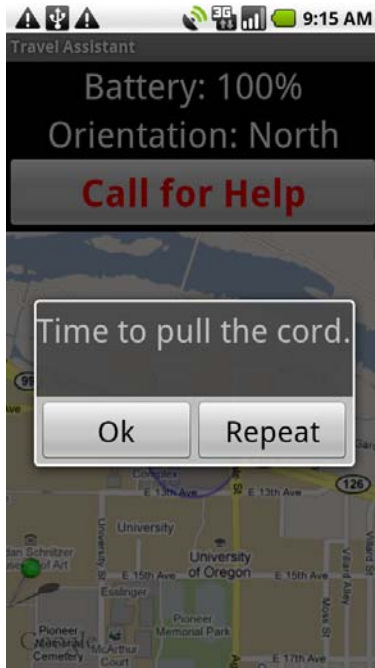


Figure 1. MyStop application

As shown in the application screen in Figure 1, the application will mirror on the screen what is played through audio. In this case, the application knows the traveler's current location by onboard GPS, knows the correct stop to get off (by loading the user profile with desired stops), and can compute the optimal time to signal the driver (by pulling the cord) and give the reminder to the user.

VIRTUAL ENVIRONMENT

We tested MyStop with a variety of travelers in the Eugene OR area. We had good results. However, the field testing was logistically challenging to say the least. We decided to see if similar tests could be carried out in a VE. To test MyStop in a VE, there are two pieces to consider. First, we need a VE that mimics a real bus ride. This VE can be used irrespective of whether there are phone-based applications involved. For instance, it might be used for training people on how to use the bus. (There is a critical service in most cities that trains those with a cognitive impairment to use

public transportation.) We have built a first prototype of the VE. As can be seen in Figure 2, we use Google StreetView as our "outside" VE. We have extended StreetView, using its API, to detect GPS coordinates and compare them to the known GPS coordinates of all bus stops in Eugene OR. API calls to Google StreetView provide the GPS coordinates of user's current location, and the GPS coordinates of Eugene bus stops are pre-loaded to the VE. The VE can be projected on a larger screen or displayed on a desktop screen, and in either case, users can interact with the VE by using arrow keys. When a user gets close to a bus stop, we ask if they want to board the bus.



Figure 2. Prompting for confirmation in StreetView

If the traveler decides to board the bus, StreetView is replaced with a bus riding video as shown in Figure 3. This video is not part of StreetView, but was instead shot by our project team. It is played in a video-controller that we extend with potential actions by the user.

The "Pull the string to get down" button simulates the string on the bus that is used to signal the driver to request to stop at next stop. The traveler can pull the string anytime when they are on the bus. It will cause the video to pause at the next bus stop for 10 seconds and then resume playing, i.e., continue the journey along the route.

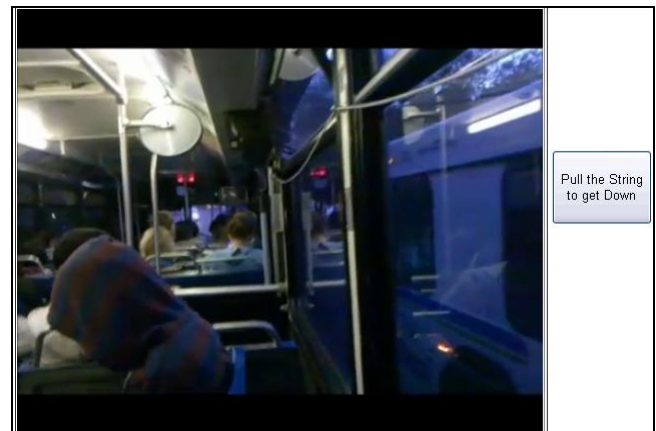


Figure 3. Bus riding video with "String Pulling" button on

When the bus stops (i.e., the video pauses), a new button will appear: “Get down from the bus” as show in Figure 4.



Figure 4. Bus riding video with the “Get Down” button on

Bus riding video is time-sampled to match with the bus stop coordinates, and if the traveler clicks the “Get down from the bus” button, he will be placed back in StreetView at the bus stop where he exited as shown in Figure 5.



Figure 5. Exit to Google StreetView

This virtual environment can be found in this location:
<http://ix.cs.uoregon.edu:7340/ve.jsp>

TESTING MYSTOP

The VE as described has no location-aware testing component. It could be used for mobile applications that do not rely on context, e.g., give periodic reminders to pay attention to surroundings. However, our interest is in testing location-aware applications like MyStop. What we would like to do is for a user to start the VE and at the same time, start the MyStop app on a real phone. (With the Droid, we can also consider using the Android emulator to show both VE and phone app simultaneously on the screen.) As we move around the VE, we want the phone app to act as if we are moving in the real environment, e.g., remind us to pull the cord when we get near our bus stop.

There are two things missing to make this happen.

1. There is no broadcast of events happening within the VE that might be of interest to a context-aware app. The critical one in our case is user location within StreetView and within the bus. One could see different information being passed out that might interest other apps, e.g., ambient light, ambient noise level. Our phone app will be useless if it does not believe it is moving on the streets or on the bus.

2. Suppose our VE is extended to provide context information to the outside. The phone app has no way of receiving it. The phone app is set up to run in the real world, with real GPS satellites. Where are the satellites in the VE?

The easier problem is extending the VE to provide context information. We have already done that for StreetView and our bus ride video. The VE will stream GPS information to whoever is listening as the user walks the streets or rides the bus.

The more difficult problem is connecting this information to apps like MyStop. It is more of a software-architecture challenge than anything else. For now, we have used a short-cut and made use of the telnet feature on every Android phone, and in particular, the fix command that allows sending arbitrary GPS coordinates to the phone. These will be used internally to alert any GPS listeners (apps) of locations. In other words, Android has built-in a means to simulate movement below the app level, and hence, no changes are needed to the app itself. One should note that (a) other sensors cannot be simulated in Android in this way, only location, and (b) Google has made no guarantees that this feature will be permanent in future versions of Android.

To make us less dependent on the fancy of hardware and software vendors of mobile devices, the other option we are exploring is the construction of context-aware apps that are "VE ready". For example, we are considering designing a Proxy location provider that will feed locations to our phone app regardless of the test environments, i.e. field or virtual. The idea is that if we opt for testing the app in the field, then the onboard GPS will be in use to provide locations to the app, and in case of VE testing, the Proxy location provider will hook up to VE for location coordinates (generated by StreetView API calls). During the app construction process, it is recognized that testing may occur in a VE. Hence, VE hooks are built-in (and not tacked on at the very end). This work is in its early stages, and we hope to have more to report at the workshop.

In conclusion, we have two current efforts: (1) validating that our MyStop testing results in the VE mirror those we received in the field, and (2) framing a software engineering model that will allow future apps to be built for VE testing.

ACKNOWLEDGMENTS

The authors wish to thank Jason Prideaux for providing support for building the MyStop application.

REFERENCES

1. Yelp for Mobile: <http://www.yelp.com/yelpmobile>
2. Loopt: <http://www.loopt.com/>
3. Google Latitude: <http://www.google.com/latitude/>
4. Fickas, S., Sohlberg, M., Hung, P., Route-following assistance for travelers with cognitive impairments: A comparison of four prompt modes, *International Journal of Human-Computer Studies*, Volume 66, Issue 12, December 2008, p. 876-888.
5. Fickas, S., Sohlberg, M., A Community Safety-Net for the Brain-Injured Traveler, CHI 2005 Workshop, Engaging The City: Public Interfaces As Civic Intermediary, Portland, 2005.
6. Hays, R. T., Jacobs, J. W., Prince, C., & Salas, E. (1992). Requirements for future research in flight simulation training: Guidance based on a meta-analytic review. *The International Journal of Aviation Psychology*, 2, p. 143-158.
7. McCloy RF, Stone R. Virtual reality in surgery. *BMJ* 2001, Volume 323, p. 912-5.
8. Joanne Lloyd, Nathan V. Persaud, Theresa E. Powell. (2009), Equivalence of Real-World and Virtual-Reality Route Learning: A Pilot Study, *Cyber Psychology & Behavior*, 12(4), p. 423-427.
9. Arthur, E. J. & Hancock, P. A. (2001). Navigation Training in Virtual Environments. *International Journal of Cognitive Ergonomics*, 5(4), p. 387-400.
10. Ruth Conroy, (2001), Spatial Navigation in immersive virtual environments, PhD Thesis, University of London, London, England.