

Design and Experimental Analysis of Continuous Location Tracking Techniques for Wizard of Oz Testing

Yang Li¹

Evan Welbourne¹

James A. Landay^{1,2}

¹ DUB Group

² Intel Research Seattle

Computer Science and Engineering

1100 NE 45th Street

University of Washington

Suite 600

Seattle, WA 98195-2350

Seattle, WA 98105

{yangli, evan, landay}@cs.washington.edu

james.a.landay@intel.com

ABSTRACT

Wizard of Oz (WOz) testing has shown promise as an effective way to test location-enhanced applications. However, it is challenging to conduct a location-based WOz test because of the dynamic nature of target settings in the field. In particular, continuous location tracking, a major task in such a test, requires a wizard to frequently update a user's location to simulate a location system. This imposes a heavy task load on a wizard. To ease wizards' tasks for location tracking, we designed two techniques, Directional Crossing and Steering, and conducted a field experiment to investigate the performance of the two techniques. A quantitative analysis shows that Directional Crossing and Steering significantly lowered a wizard's task load for location tracking without sacrificing accuracy.

Author Keywords

Wizard of Oz, location-enhanced application, field study.

ACM Classification Keywords

I.3.6 [Methodology and Techniques]: Interaction techniques

INTRODUCTION

Location-enhanced applications, one of the most widely adopted types of ubicomp applications today, provide useful services based on the location of people, places and things [2, 7]. However, despite steady improvements and growing ubiquity, location sensing technologies are still complex to use. For example, GPS does not work indoors or in areas with high, dense buildings, while WiFi-based positioning needs lots of training for high precision and depends on RF infrastructure that may not be present in a locale. Consequently, it is expensive and time-consuming to test location-enhanced applications with existing location sensing infrastructures. This is especially true in the early stages of design when the goal is to quickly get feedback.

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2006, April 22–27, 2006, Montréal, Québec, Canada.

Copyright 2006 ACM 1-59593-178-3/06/0004...\$5.00.

Wizard of Oz (WOz) testing, which has been widely employed in user interface design, is a technique in which a wizard (often a designer) fakes an incomplete (or nonexistent) system component to conduct early user evaluations [5]. WOz testing has shown great potential as an inexpensive but effective way to test location-enhanced applications in the early stages of design [3, 6, 9].

Based on our experience [9] as well as that of others [6], we have seen that in a location-based WOz study a wizard needs to: 1) track the user's location to simulate location sensing; 2) trigger application behaviors to test usability; and 3) observe users and environments. It is extremely challenging to perform such a study in the mobile, dynamic target settings found in the field. In particular, location tracking requires much attention of the wizard.

Thus, we set out to design effective techniques to relieve wizards from heavily-loaded location tracking tasks, so that they can save their effort for performing other tasks. In this work, we describe two new techniques we designed that allow a wizard to keep track of a user's location: Directional Crossing and Steering. We conducted a field experiment to investigate the performance of the two techniques. A quantitative analysis showed that our new techniques resulted in significantly lower task load for wizard participants than two benchmark techniques.

CONTINUOUS LOCATION TRACKING TECHNIQUES

In this work, we concentrate on pedestrian-based mobile applications, an important class of location-enhanced applications that often require fine location input. In a WOz



Figure 1. The wizard's interface employed in our experiment. The red circle represents the user's location.

study of these applications, a wizard follows a user and updates the user's location with a mobile device (e.g., a PDA) to simulate a location system for the user's device. Location tracking often employs a map-based interface [9] in which a wizard moves an icon representing the user's location on a map image (see Figure 1). We call location tracking with a map-based representation "continuous location tracking." This is in contrast to tracking with a discrete, textual representation of locations (e.g., selecting among buttons labeled with place names). In general, a discrete representation is insufficient to reflect continuous spatial relations such as distance information, which is useful for many scenarios such as finding the closest bus stop. In addition, use of a discrete representation increases the reaction time required of a wizard as the number of places of interest increases, as described by Hick's Law [10]. In contrast, a map-based representation provides a direct mapping of the physical world.

There are two major challenges in designing an efficient technique for continuous location tracking. First, creation of a map that allows easy mapping to a physical space. Second, the design of an interaction technique that allows a wizard to easily update the user's location on the map while walking in the field. Substantial work on map design has been conducted by cartographers. We focus here on the second challenge in the context of using a pen-based computer (such as a Tablet PC), which offers better mobility, as a target device for a wizard.

Gesture-Based Location Tracking Techniques

We designed two new techniques for continuous location tracking that were inspired by previous work on crossing-based interactions [1].

Directional Crossing

Directional crossing allows a wizard to update the user's location on a map by crossing an icon using a pen. Each crossing will move the icon along the direction of the crossing gesture (the orientation of the stroke) for a fixed distance (three meters in our implementation). The icon is designed as a circle (see Figure 1) to avoid biasing the crossing direction. A crossing can begin or end inside or outside of the icon, but it must cross the border of the icon. The length of a crossing gesture is ignored in our current design. This was mainly because we found it difficult to draw a stroke of a specific length while walking.

This technique does not require precise positioning and allows an operation with multiple operands (i.e., the target to be updated and a movement direction) to be specified in a single stroke. As a complement to the incremental updates of directional crossing, Pick & Drop [11] is also allowed for moving an icon to any target position on a map. A stroke that completely falls within an icon is recognized as a Pick.

Steering

Another approach that we explored is the use of inexpensive sensors to reduce the wizard's task load. For example, accelerometers are increasingly appearing in devices, including Tablet PCs and mobile phones.

Accelerometer-based sensing technologies can provide reliable estimates of a person's walking speed [8].

We designed a Steering technique that employs a real-time, accelerometer-based step-estimation algorithm to estimate how far a user moves and to automatically update the user's location on the map along the current trajectory. At the same time, a wizard can *steer* the icon by specifying movement direction with directional crossing. Note that in Steering, a directional crossing does not move a crossed icon but only changes its movement direction. For example, a wizard crosses the icon to change its movement direction when a user makes a turn. Pick & Drop is again used for directly adjusting the icon's position.

Steering is similar to "dead-reckoning" [8] in that both techniques infer current position based on a known previous position and a stream of readings from onboard inertial sensors (e.g., accelerometers), and do so independently of environmental infrastructure. However, "dead-reckoning" solely relies on inference; this has proven extremely difficult in practice with current technologies. In contrast, Steering includes a human user (wizard) to provide correct directional information as well as to compensate for large cumulative error in position estimation.

THE FIELD EXPERIMENT

We ran a field experiment to investigate the performance of the two new techniques (Directional-Crossing and Steering) as well as two benchmark techniques (Pick&Drop and Drag&Drop) that are commonly used for moving graphical objects. Drag&Drop has been used for location tracking in previous WOz studies [6, 9].

Experimental Design

The experiment was conducted on one floor of a building (see Figure 2). We chose an indoor environment for the experiment so as to allow for greater control over experimental conditions. For example, the degree of reflection from a Tablet PC screen may vary based on the time and weather condition in outdoor settings.

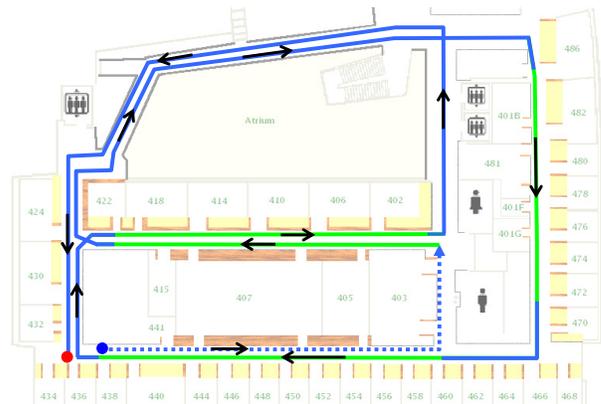


Figure 2. The experimental path starts from the blue dot and ends at the red dot (about 340 meters long). Blue lines stand for an experimenter being tracked walking at a normal speed while green lines stand for fast speed. The path denoted by the dashed line was used for warming up and was not used for analysis.

The experiment included a demographic questionnaire, an introductory tour to acquaint participants with the floor and the map, four experimental sessions with one session for each technique, and a post-study questionnaire. Each experiment took about 70 minutes in total. A 4x4 Latin Square was used to counterbalance the order of the techniques. Each session was divided into two parts: in the first part, participants learned to use a technique through extensive practice; the second part was the experimental phase in which a participant (acting as a wizard) followed an experimenter (acting as a user) around the floor keeping track of the experimenter's location using one of the four techniques. An experimental phase lasted about six minutes and participants were allowed breaks between sessions.

Forty checkpoints, unknown to participants, were distributed on the experimental path (see Figure 2). While walking the path, the experimenter recorded his arrival time at each checkpoint. The experimenter's log was later used to analyze the accuracy of each location tracking. We collected data on all experimental events including pen down/up, pen draw, location updates on the map, and times when the experimenter reached checkpoints.

Experimental Software and Hardware

The *wizard's* device, used by a participant for performing tracking tasks, was a Motion M1400 Tablet PC. This device has a Pentium M 1.1GHz processor, 512MB of RAM and a 12.1" display with a digitizer. The experiment was conducted at a resolution of 1024x768 pixels and the wizard UI (see Figure 1) was maximized on the screen.

The *user's* device (used by an experimenter) was a Toshiba Protégé Tablet PC. The interface on this device showed the same floor map as the wizard's tablet but with the previous and next four checkpoints overlaid so as to aid the experimenter in remembering where he should record time. The experimenter could record time by pressing a physical button on the Tablet PC, alleviating the need to slow down and hiding the operation from the participants.

For the Steering technique, a 3-axis accelerometer on Intel's sensor board [4], wired to the experimenter's tablet PC, was used to estimate the experimenter's movement. The accelerometer data was processed locally on the tablet and the resulting movement distance information was sent to the wizard's device. The two devices communicated via a peer-to-peer wireless connection.

Participants

Twelve adult volunteers were recruited from a university campus, four female and eight male, 19 to 28 years old. They received \$25 for their participation. One of the twelve participants was left-handed. 50% of participants had experience using a Tablet PC, and 58% of participants were familiar with the building floor for the experiment. Four of the twelve participants had heard of WOz testing but only one of them had experience in conducting a WOz study.

EXPERIMENTAL ANALYSIS

Repeated measure variance analyses (using location tracking technique as a within-subject factor) and post hoc

pairwise comparisons were performed on the tracking deviation, the operation frequency and the percentage of total interaction time. The tracking deviation measures how accurate a participant kept track of the experimenter's location using a particular technique. The operation frequency and the percentage of interaction time reflect how much attention or effort is required of a wizard participant while using a technique.

Tracking Deviation

Tracking deviation D of a session is defined as the square root of the mean square distance between a checkpoint location and the corresponding tracked location:

$$D^2 = \frac{1}{n} \sum_{i=1}^n \|e_i - s(t_e(i))\|^2 \quad (1)$$

where n is the number of checkpoints (i.e., 40 in our experiment), e_i is the 2D location of the i th checkpoint and $t_e(i)$ is the time when the experimenter reached the i th checkpoint. $s(t)$ is the experimenter's location at time t tracked by a participant, which is calculated as a weighted sum over the corresponding location sequence generated by a participant (see Equation 2).

$$s(t) = \sum_{i=1}^m w(t_s(i) - t) s_i \quad (2)$$

where m is the length of the sequence, s_i is the i th location in the sequence and $t_s(i)$ is the time when the i th location update is made by a participant. The weight function w is defined as a Gaussian over the time distance Δt (see Equation 3), where W is a normalization factor.

$$w(\Delta t) = \frac{1}{W} \exp\left(-\frac{\Delta t^2}{2\sigma^2}\right) \quad (3)$$

In this way we give higher weight to location updates that occur closer to when the experimenter reaches the checkpoint.

We tested tracking deviation with different σ values, ranging from 1 to 3 seconds. The tracking deviation of DirectionalCrossing was consistently the smallest with a mean of 1.6 meters; Steering always had the greatest deviation with a mean of 2.2 meters; Drag&Drop and Pick&Drop had similar tracking deviations with a mean ranging from 1.8 to 1.9. However, there was no significant difference found on the tracking deviation among the four techniques. While σ is 2 seconds, $F(3, 33) = 1.8, p > .05$.

Operation Frequency and Interaction Time Percentage

There was a significant difference found in the operation frequency of the four techniques, $F(3, 33) = 9.2, p < .001$. The operation frequency for Steering was the smallest with a mean of 0.16 times/second, which was significantly lower than the other three techniques ($p < .05$) (Drag&Drop: .54, DirectionalCrossing: .38, Pick&Drop: .39). There was no significant difference found in operation frequency among the other three techniques.

We also found there was a significant difference in the percentage of interaction time among the four techniques, $F(3, 33) = 52.03, p < .001$ (see Figure 3). The percentage of

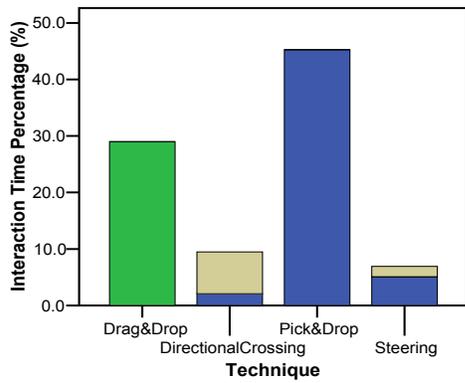


Figure 3. The mean of the percentage of interaction time during a session. Yellow represents crossing gestures and blue represents pick & drop operations.

interaction time for using Steering was the smallest with a mean of 6.9%, which was found to have no significant difference ($p > .05$) from DirectionalCrossing (Mean=9.5%). These two techniques had interaction times significantly lower than the other two techniques ($p < .001$).

Subjective Preference

Participants were asked to rate the four techniques on six dimensions: ease of learning, ease of use, accuracy, attention required and fatigue of the eyes and hands using a 5-point scale (5 for best, and 1 for worst) (see Table 1). We will discuss some of the ratings in the next section.

DISCUSSION AND FUTURE WORK

The results indicate that Steering requires a much lower operation frequency than the other techniques without significantly increasing tracking deviation. The high rating on Steering's required attention and hand and eye fatigue also support this (see Table 1). However, Steering was not significantly superior to DirectionalCrossing in terms of interaction time. This is because Pick&Drop was used significantly more frequently during Steering (Mean=47%) than DirectionalCrossing (Mean=7%), and Pick&Drop was significantly slower ($p < .001$) (Mean=1.42 second/operation) than crossing gestures. Note that there was no significant difference found on operation speed of crossing gestures between DirectionalCrossing (Mean=.20s) and Steering (Mean=.23s) ($p > .05$).

This implies that Steering required many corrections from participants. Although Steering's tracking deviation was not significantly higher than the others, the subjective preference showed the lowest rating on its accuracy. Some participants felt the location updates were not fully under their control while using Steering. Some participants did not feel fully acquainted with Steering. The lowest rating on Steering's ease of learning reflects this. In contrast, DirectionalCrossing seems to have the best global performance among the four techniques. The results suggest finding a better technique than Pick&Drop to adjust Steering's automatic location updates.

All of these experimental techniques require a wizard to explicitly specify a target to update, e.g., by crossing or tapping. This was because in a real WOz experiment a

Dimension	Drag&Drop	Directional Crossing	Pick&Drop	Steering
Learning	4.9	4.3	4.3	3.0
Use	3.4	4.1	3.7	3.4
Accuracy	3.3	3.3	3.7	2.7
Attention	2.4	3.5	2.6	4.1
Eye fatigue	3.8	4.0	3.8	4.1
Hand fatigue	3.4	4.3	3.8	4.5

Table 1. The participants' preferences for each technique.

wizard may need perform other interactions on the map (such as panning or updating content).

All of our participants were "experts" with Drag&Drop, but even so, Drag&Drop did not perform as well as the two new techniques. However, a future longitudinal study in a more realistic setting (such as using more complex testing paths) would complement these results. We are also investigating issues such as how the inaccuracy of actual sensors affects the results of a WOz study by modeling sensor inaccuracies in a WOz system.

CONCLUSION

We designed two continuous location tracking techniques for WOz testing of location-enhanced applications. We conducted a field experiment to quantitatively compare the performance of the four techniques. The results showed that compared to two benchmark techniques, our two new techniques achieved a similar tracking accuracy but significantly lowered a wizard's task load. In addition, the two new techniques support movement direction, which is not available in the two traditional techniques. Steering required the least interaction time and frequency, while DirectionalCrossing had the best global performance.

REFERENCES

- Accot, J. and Zhai, S. More than dotting the i's - Foundations for crossing-based interfaces. In Proc. CHI 2002, 73-80.
- AT&T. AT&T Wireless mMode - Find Friends. <http://www.attwireless.com/mmode/>.
- Bohnenberger, T., et al. Decision-Theoretic Planning Meets User Requirements: Enhancements and Studies of an Intelligent Shopping Guide. In Proc. Pervasive 2005, 279-296.
- Brunette, W., et al. Some Sensor Network Elements for Ubiquitous Computing. In Proc. IPSN 2005, 388-392.
- Dahlbäck, N., Jönsson, A., and Ahrenberg, L. Wizard of Oz Studies - Why and How. In Proc. IUI 1993, 193-200.
- Dow, S., et al., Wizard of Oz Support throughout an Iterative Design Process. *IEEE Pervasive Computing*, 4,4(2005): 18-26.
- ekahau. Increased visibility and monitoring of critical assets and people. <http://www.ekahau.com/>.
- Lee, S. and Mase, K., Activity and Location Recognition Using Wearable Sensors. *IEEE Pervasive Computing*, 1, 3 (2002): 24-32.
- Li, Y., Hong, J.I., and Landay, J.A. Topiary: A Tool for Prototyping Location-Enhanced Applications. In Proc. UIST 2004, 217-226.
- Raskin, J., *The Humane Interface*: Addison Wesley.
- Rekimoto, J. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In Proc. UIST 1997, 31-39.