

2017

Flattening methods for adaptive location-based software to user abilities

Thitivatr Patanasakpinyo
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/etd>



Part of the [Computer Sciences Commons](#)

Recommended Citation

Patanasakpinyo, Thitivatr, "Flattening methods for adaptive location-based software to user abilities" (2017). *Graduate Theses and Dissertations*. 16191.

<https://lib.dr.iastate.edu/etd/16191>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Flattening methods for adaptive location-based software to user abilities

by

Thitivatr Patanasakpinyo

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Major: Computer Science

Program of Study Committee:
Les Miller, Co-major Professor
Wallapak Tavanapong, Co-major Professor
Shashi Gadia
Johnny Wong
Stephen Gilbert

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2017

Copyright © Thitivatr Patanasakpinyo, 2017. All rights reserved.

DEDICATION

I would like to dedicate this dissertation to my family: my son, wife, mother, father, brother, grandmother, and in-laws, without their support I would not have earned this accomplishment. I would also like to thank my major professors, my committees, and all my friends for their guidance, suggestion, and any priceless assistance during my time at Iowa State University. All of them made the moment of my life in United States of America precious and glorious.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
ACKNOWLEDGEMENTS	xiii
ABSTRACT	xv
CHAPTER 1. INTRODUCTION	1
1.1 Problem	2
1.2 Deliverables	4
1.3 Contributions	5
CHAPTER 2. LITERATURE REVIEW	6
2.1 The effect of individual differences on software utilization and performance	6
2.2 Interesting studies that relate to spatial ability	10
2.3 Impact of spatial visualization (VZ) in computer-based applications	18
2.4 The location-based survey software	20
2.5 The adaptive user interfaces (AUI) and problems that lead to it	22
2.6 Task-based issues	31

CHAPTER 3. OBJECT-INDEXING	32
3.1 Introduction	32
3.2 Design of Object-Indexing	32
3.2.1 Line	35
3.2.2 Point	38
3.2.3 Area	42
3.3 Conceptual Object-Indexing Usage	45
3.3.1 Line	47
3.3.2 Point	52
3.3.3 Area	55
3.4 Using Object-Indexing in Study Software	59
CHAPTER 4. STUDY I: A STUDY TO INVESTIGATE PARTICIPANTS' PER- FORMANCE WITH DIFFERENT USER INTERFACES	74
4.1 Study Procedure	75
4.1.1 Test of Spatial Visualization	75
4.1.2 Field Portion of the Study	77
4.2 Study Software	79
4.2.1 Software Interface Components	80
4.2.2 Software Features	80
4.3 Raw Results from Study	83
4.4 Preliminary Analyzed Results	87
4.4.1 Hypotheses for Study I	88
4.4.1.1 Significant Hypotheses	91
4.4.1.2 Suggestive Hypotheses	95
4.5 Post-study Discussion	99

CHAPTER 5. STUDY II: A STUDY TO INVESTIGATE PARTICIPANTS' PERFORMANCE WITH A UNIVERSAL USER INTERFACE USING A LARGE MAP SPACE	100
5.1 Objectives	100
5.2 Methodology	101
5.2.1 Study Designs	101
5.2.2 Study Procedures	101
5.2.2.1 Test of Spatial Visualization	101
5.2.2.2 Field Portion of the Study	101
5.2.3 Study Software	103
5.2.3.1 Software Components	103
5.2.3.2 Software Features	105
5.3 Results	105
5.4 Discussion	109
5.4.1 Research Question 1: Was there a significant difference in verification score?	110
5.4.2 Research Question 2: Was there a significant difference in time?	110
5.4.3 Research Question 3: Was there a significant difference in distance?	112
5.4.4 Research Question 4: Was there a significant difference in a number of error taps?	114
5.4.5 Research Question 5: Was there a significant difference in a number of pan taps?	114
5.4.6 Research Question 6: Was there a significant difference in a number of zoom taps?	116
5.5 Conclusions	118

CHAPTER 6. STUDY III: A STUDY TO INVESTIGATE PARTICIPANTS' PERFORMANCE WITH AN ADAPTIVE USER INTERFACE	119
6.1 Objectives	119
6.2 Methodology	120
6.2.1 Study Designs	120
6.2.2 Study Procedure	122
6.2.2.1 Test of Spatial Visualization	122
6.2.2.2 Field Portion of the Study	123
6.2.3 Study Software	125
6.2.3.1 Software Components	126
6.2.3.2 Software Features	126
6.2.3.2.1 Features on Traditional UI	129
6.2.3.2.2 Features on the Adaptive UI for Low VZ	129
6.2.3.2.3 Features on the Adaptive UI for High VZ	132
6.2.3.2.4 Prediction Module	132
6.3 Results	136
6.4 Discussion	138
6.4.1 Improvement with Adaptive UI	138
6.4.1.1 Normalization of Times	138
6.4.1.1.1 Normalization by Overall Average	139
6.4.1.1.2 Normalization by Individual Average	141

6.4.1.2	Tests of Hypotheses	141
6.4.1.2.1	Research Question 1: For participants who were assigned treatment 0, was there a significant difference between time in phase 1 and phase 2?	142
6.4.1.2.2	Research Question 2: For participants who were assigned treatment 1, was there a significant difference between time in phase 1 and phase 2?	144
6.4.1.2.3	Research Question 3: Focusing on phase 1, was there a significant difference between time used by participants of treatment 0 and treatment 1? . . .	145
6.4.1.2.4	Research Question 4: Focusing on phase 2, was there a significant difference between time used by participants of treatment 0 and treatment 1? . . .	146
6.4.2	Analysis of Prediction Methods	148
6.4.3	Additional Analysis: Market Basket Analysis on Participant's Behavior	151
6.4.3.1	Rule 1	152
6.4.3.2	Rule 2	154
6.4.3.3	Rule 3	154
6.4.4	Additional Analysis: Mispredicted Participants	155
6.5	Conclusions	156
CHAPTER 7. CONCLUSIONS AND CONTRIBUTIONS		159
BIBLIOGRAPHY		162
APPENDIX A. IRB APPROVAL DOCUMENTS		175
APPENDIX B. COMPLETE STUDY RESULTS		178

LIST OF TABLES

Table 4.1	Address Verification Possibilities.	78
Table 4.2	Example Raw Results from Study.	87
Table 4.3	Hypotheses for Low VZ Participants.	89
Table 4.4	Hypotheses for High VZ Participants.	90
Table 5.1	Example Results from Study II.	109
Table 6.1	Possible Scenarios in Study III.	125
Table 6.2	Partial Results from Study III.	137
Table 6.3	Example Results of Actual Time and Normalized Times.	142
Table 6.4	Complete Prediction Results of Different Methods.	150
Table B.1	Complete Results of Study I.	179
Table B.2	Complete Results of Study II.	180
Table B.3	Complete Results of Study III.	181
Table B.4	Association Rules: Rules 1 - 36	183
Table B.5	Association Rules: Rules 37 - 62	184
Table B.6	Association Rules: Rules 63 - 86	185
Table B.7	Association Rules: Rules 87 - 120	186
Table B.8	Association Rules: Rules 121 - 146	187
Table B.9	Association Rules: Rules 147 - 149	188

LIST OF FIGURES

Figure 3.1	Example of a Vector Map Generated from a Shapefile.	33
Figure 3.2	The Conceptual Class of Street Object.	34
Figure 3.3	The Concept of Creating a Street Object.	35
Figure 3.4	A Raster Map Consists of Two Streets without Street Names. . . .	37
Figure 3.5	Street Object of A Street.	37
Figure 3.6	Street Object of B Street.	38
Figure 3.7	The Conceptual Class of Intersection Object.	39
Figure 3.8	The Concept of Creating an Intersection Object.	39
Figure 3.9	A Raster Map Consists of One 5-way Intersection without Name. . .	41
Figure 3.10	Intersection Object of I Intersection.	41
Figure 3.11	The Conceptual Class of Wetland Object.	42
Figure 3.12	The Concept of Creating a Wetland Object.	43
Figure 3.13	A Raster Map Consists of One Wetland without Name.	44
Figure 3.14	Wetland Object of W Lake.	45
Figure 3.15	Expected Scenario.	46
Figure 3.16	Flow Chart of Object-Indexing Implementation for Line Object Model.	48
Figure 3.17	How Object-Indexing Operates for Line Object Model.	49
Figure 3.18	Street Object Detecting in Our Software (1).	50
Figure 3.19	Street Object Detecting in Our Software (2).	50
Figure 3.20	Flow Chart of Object-Indexing Implementation for Point Object Model.	53

Figure 3.21	How Object-Indexing Operates for Point Object Model.	53
Figure 3.22	Flow Chart of Object-Indexing Implementation for Area Object Model.	56
Figure 3.23	How Object-Indexing Operates for Area Object Model.	57
Figure 3.24	Example of Streets Text File.	60
Figure 3.25	A User Taps on an Intersection in Tiger Map.	62
Figure 3.26	A User Taps on an Intersection in Google Map.	63
Figure 3.27	A User Taps on Gray Avenue on Tiger Line Map.	64
Figure 3.28	A User Taps on S 4 th Street on Tiger Line Map.	65
Figure 3.29	A User Taps on Agg Avenue on Tiger Line Map.	66
Figure 3.30	A User Taps on S 4 th Street on Google Map.	67
Figure 3.31	A User Taps on Agg Avenue on Google Map.	68
Figure 3.32	A User Taps on 3-way Intersection on Tiger Line Map.	70
Figure 3.33	A User Taps on 4-way Intersection on Tiger Line Map.	71
Figure 3.34	A User Taps on 3-way Intersection on Google Map.	72
Figure 3.35	A User Taps on 4-way Intersection on Google Map.	73
Figure 4.1	Map of Selected Neighborhood in Ames, Iowa.	76
Figure 4.2	Paper Folding Test [20].	77
Figure 4.3	Software Screen.	81
Figure 4.4	Relationship between Primary Map and Mini Map.	84
Figure 4.5	Mini Map in the Software Interface.	85
Figure 4.6	Current Location Displayed on Map.	85
Figure 4.7	Average Address Verification Score between 2 Datasets.	92
Figure 4.8	Average Address Verification Score between 2 Datasets.	93
Figure 4.9	Average Address Verification Score between 2 Datasets.	94
Figure 4.10	Average Address Verification Score between 2 Datasets.	95
Figure 4.11	Average Distance between 2 Datasets.	96
Figure 4.12	Average Distance between 2 Datasets.	97

Figure 4.13	Average Distance between 2 Datasets.	98
Figure 5.1	UI of Study I (left) and Study II (right).	102
Figure 5.2	Software UI of Study II.	104
Figure 5.3	A Satellite Map in Study II.	106
Figure 5.4	Three UI Features of Study II.	107
Figure 5.5	A Box Plot Illustrated Address Verification Score of High and Low VZ Participants.	111
Figure 5.6	A Box Plot Illustrated Times of High and Low VZ Participants. . .	112
Figure 5.7	A Box Plot Illustrated Distances of High and Low VZ Participants.	113
Figure 5.8	A Box Plot Illustrated Numbers of Error Taps of High and Low VZ Participants.	115
Figure 5.9	A Box Plot Illustrated Numbers of Pan Taps of High and Low VZ Participants.	116
Figure 5.10	A Box Plot Illustrated Numbers of Zoom Taps of High and Low VZ Participants.	117
Figure 6.1	Ten addresses to be verified in three phases.	121
Figure 6.2	Participant's Work Flow in Study III.	124
Figure 6.3	Traditional UI of Study III.	127
Figure 6.4	Address List in UI of Study III.	128
Figure 6.5	Quit Button in UI of Phase 3.	129
Figure 6.6	Major Landmarks and Minor Landmarks were located on the Mini Map.	130
Figure 6.7	Minor Landmarks were located on the Primary Map.	131
Figure 6.8	Implementation of Object-Indexing (Type I).	131
Figure 6.9	Implementation of Object-Indexing (Type II).	133
Figure 6.10	Complete Work Flow of Study III.	135

Figure 6.11 Walking Paths in Phase 1 (left) and 2 (right) of a Participant. . . . 139

Figure 6.12 Addresses to be verified in Phase 1. 140

Figure 6.13 Addresses to be verified in Phase 2. 140

Figure 6.14 A Box Plot Illustrated a Time in Phase 1 (Normalized by Overall
Average) and Phase 2 for Treatment 0. 143

Figure 6.15 A Box plot Illustrated a Time in Phase 1 (Normalized by Overall
Average) and Phase 2 for Treatment 1. 145

Figure 6.16 A Box Plot Illustrated a Time in Phase 1 (Normalized by Overall
Average) for Treatment 0 and 1. 146

Figure 6.17 A Box Plot Illustrated a Time in Phase 2 for Treatment 0 and 1. . . 147

Figure 6.18 The Set of Parameters of the Associator for Market Basket Analysis. 153

Figure 6.19 Time Used in Phase 3 by Different Groups of Participants. 156

Figure 6.20 Distance Used in Phase 3 by Different Groups of Participants. . . . 157

ACKNOWLEDGEMENTS

I would like to express my profound thanks to those who helped, advised, supported, and encouraged me during the time I was conducting my research and writing this dissertation. First and foremost, Dr. Les Miller for his great patience, support, and supervision throughout this research and the writing of this dissertation. I am very grateful to be mentored by him. I would also like to thank Dr. Wallapak Tavanapong, my co-major professor, for her kind guidance and sustenance for both research and teaching. I would also like to thank my committee members for their worthy suggestion and contribution on my work: Dr. Shashi Gadia, Dr. Johnny Wong, and Dr. Stephen Gilbert.

Without a support from my family: my mother, father, brother, and grandmother, I could not make it to the final state of my graduate study. Their suggestion and encouragement helped me overcome all issues in the past years. I would also like to thank my workplace, the Faculty of Information and Communication Technology, Mahidol University, which is counted as my second home, for the financial assistance along six years that I had pursued my graduated degrees in the United States of America. Especially, I have a thank to Dr. Jarernsri Mitranont, the dean of the Faculty of Information and Communication Technology, who appointed me as the candidate of this assistantship

I would like to express my thank to Dr. Kenneth Platt and Dr. Ratree Platt, the former advisers of Thai Student Association of Iowa State University, who are always be the beloved parents of every Thai student who came to Iowa State University. I would also like to thank every member of Thai Student Association of Iowa State University who made my living in Ames happy and cozy like I was in my hometown.

I would like to thank every group of my fellows that is always together with me, which consists of the group of my colleagues from my workplace, the group of my research team, and the group of my classmates and laboratory-mates

A special thank is provided to the Office of Educational Affairs, Royal Thai Embassy, for being my legal guardian during my graduated study in the United State of America.

My last thank is for both my greatest life partner and my greatest creation: Natsima Thongsuriyapong and Alanvat Andy PatanasapkPinyo, my wife and my baby boy, who always take care of me, who always love me with their whole hearts, who always back me up when I am down, and who unconditionally fulfill my life with their love and care.

ABSTRACT

Individual spatial visualization has been proved to be an important factor that impacts software usage performance, especially on location-based software, which has become a major trend in this ubiquitous computing era. Developing a solution that could enhance a location-based software to handle different users with various levels of spatial visualization abilities could yield improvement in both software usability and user performance. In this dissertation, we started by setting up an hypothesis with a goal to discover the factors that are helpful in minimizing/flattening those differences. Users with various spatial visualization abilities could generate a similar performance on the same software. To test this, we have run experiments to verify this claim. An application used in this study is address verification using location-based software on a portable computing device, which is a professional task used by US census bureau office. We have developed an adaptive location-based software that is able to adapt its user interface in real time according to the user's spatial visualization level.

CHAPTER 1. INTRODUCTION

A location-based software is an application software that has a map or similar spatial locating tools as a component. Location-based software has become an important concept of software development because of its property to consolidate the comfortable use of application software with the advantage of a map representing the current environment.

Along with location-based software, most developers pay attention to software development on portable devices such as iOS devices, Android devices, Windows Phone devices, etc. Because of the portable property and high performance requirement, portable devices have become a major player in the information technology market. Most companies also have a specific version of their software for these portable devices such as Facebook[®], Skype[®], Yelp[®], or Weather Channel[®] apps. The use of portable devices provides a perfect solution for location-based application software. Nowadays, we can find many location-based software applications on portable devices. Most of them have become an important part of our daily life. In addition to personal devices, many businesses and government agencies have taken advantage of location-based software in their field operations.

The usability of software on portable devices is an important issue. The fact that people are different makes it more difficult to achieve a high level of usability for all users. Some people can struggle using software because of this reason. The user's level of spatial visualization (VZ) has been shown to be an important factor for how successful a software user is [93][110][76]. This dissertation has set up and implemented three experiments to

investigate of how individual differences, in terms of individual spatial visualization, can affect people’s performance on location-based software using portable devices. Finally, we developed an adaptive user interface software application that helped users with different skill sets, particularly high and low spatial visualization, to have competitive performance when doing a location-based task. We briefly mentioned the scope of problem, deliverables and our contributions in the remainder of this chapter.

1.1 Problem

It is well-known that people have different skills [11][18]. An interesting question is whether one universal interface design for location-based software on a portable device is effective for all types of users, high and low spatial visualization ability. Most people with low spatial ability tend to have lower performance in software usage than high spatial ability people [10]. In this dissertation, we look at factors/treatments that could help reduce this difference and how the factors/treatments might be used together to bridge the difference between users for location-based software. Our focus will be on the use of the address verification task used by the Census Bureau.

Our strategy has been to develop the location-based software, which is an Android application, to be used as a tool for this address verification task. Given our strategy, we divided our experiments into three studies:

- **Study I**

The first study consisted of two sessions. For the first session, each participant took a Paper Folding Test [20] to evaluate their level of spatial visualization (low/high). Next, the second session asked them to verify ten addresses in the neighborhood using our location-based software application. For each participant, one treatment (out of three treatments) was randomly assigned. Each treatment deactivated one feature

on the user interface. Since there were three features, each participant had only two active features to use while verifying addresses. Study I was described in detail in Chapter 4. The objective of Study I was to observe how different treatments impacted participant's performance. Understanding which UI features were benefit for which type of participants (low/high spatial visualization) guided us for the development of adaptive user interface software application, which was the ultimate goal of this dissertation.

- **Study II**

The Study II also consisted of two sessions. The first session was the same as the first session of Study I. The second session was also similar, but there was a difference. In the second session, all three UI features were always active for every participant. In addition, a larger map size was used to make map operations easier to handle. Study II was described in detail in Chapter 5. The objective of study II was to observe how effective it was when a participant, regardless of that participant was a low or high spatial visualization person, used software that activated every UI feature all the time. Providing a software application with the same UI for every participant is referred to as universal UI software. The question was whether a larger map size and the universal UI would result in equivalent performance for both high and low VZ participants.

- **Study III**

The Study III also consisted of two sessions. The first session was the same as both previous studies. The difference took place in the second session where the software that participants used for address verification was adaptive UI software. The adaptive UI software can adapt its user interface by predicting participant's level of spatial visualization. After the software got prediction result, it adapted its user interface based on that prediction result. Study III was described in detail in Chapter 6. The

objective of Study III was to observe an improved performance of participants with different levels of spatial visualization when they verified addresses using the adaptive UI software application.

1.2 Deliverables

This dissertation provided five deliverables, which were:

1. Adaptive UI Application Software

We introduced new adaptive UI software application for address verification that had an ability to predict user's spatial visualization and adapt its user interface according to that prediction result to fit each user with different spatial visualization. The adaptive UI software was described in Chapter 6.

2. New Data Structure for Map Components

We introduced a new data structure, "Object-Indexing", that can be implemented to represent a map component. This methodology allowed any map component to have an object-oriented accessibility on a traditional raster map. Object-Indexing was described in Chapter 3.

3. Study Results

Results of Study I, Study II, and Study III were exhibited in Chapter 4, 5, and 6, respectively. For each study, we explained study design and study procedure along with its result.

4. Analysis of Performance Improvement & Prediction of VZ Level

We analyzed our results to ensure a significant improvement of participants' performance when we assigned the adaptive UI software (Study III). This analysis was described in Chapter 6. Furthermore, we discussed the accuracy of our prediction

unit, which was developed to provide ‘real-time’ prediction of a user’s VZ level, in Chapter 6 as well. The prediction model was built on the work of Batinov [2].

5. Analysis of Users’ Behavior

We analyzed and extracted important features from the logs of the participants to generate the set of association rules using Market Basket technique (Chapter 6).

1.3 Contributions

The contributions of the dissertation consisted of three key areas:

1. The development of our new data structure for maps that provides fast performance combined with indexable map components.
2. A suite of user interface features that provide different levels of support for users based on the predictive value of their spatial visualization level. A robust set of such features is critical for the successful development of an adaptive user interface based on the prediction of a user’s level of spatial visualization.
3. The development of the adaptive user interface software that demonstrates how prediction of spatial visualization and the suite of user interface features can be combined to improve the performance of a user independent of their level of spatial visualization.

The remainder of this dissertation is organized as follows: Chapter 2 provided a review of the literature relevant to the dissertation. The Object-indexing fundamentals were given in Chapter 3. Chapter 4, 5, and 6 look at an explanation of Study I, Study II, and Study III, respectively. Chapter 7 summarized the results of this dissertation.

CHAPTER 2. LITERATURE REVIEW

This is the review of literature that is related to the proposed work in this dissertation proposal. This chapter is organized by topic.

2.1 The effect of individual differences on software utilization and performance

Individual differences play an important role in the interaction between humans and computers. Carroll [11] gave eight categories of human intelligence: crystallized intelligence, fluid intelligence, general memory and learning, broad visual perception, broad auditory perception, broad retrieval ability, broad cognitive speed, and processing speed. Carroll [11] also classified spatial ability as a factor that affects broad visual perception. Thus, among varying types of abilities, spatial ability was recognized as the most important representative for human difference. Lohman [58] defined spatial ability as “the ability to generate, retain, retrieve, and transform well-structured visual images” [Lohman, *Human abilities: Their nature and measurement*, p.98, 1996].

Egan [18] selected spatial ability as the most important ability for predicting human performance on a computer. We see similar statements in [16][18][19][110]. Egan [18] suggested that an individual’s spatial ability, memory, reasoning ability, verbal aptitude, and possibly personality played an important role in a user’s performance when accessing information on a computer.

Benyon et al. [6] noted that, among all types of intelligence, spatial ability was often selected for study when looking at the connection between human performance and individual differences.

Büiring, Gerken, and Reiterer [10] created a study to understand the relationship between spatial ability and the user's performance on a zoomable user interface. In this study, there were two software screens. The first one was the screen which represented the whole information and it came with an overview window. The second screen came with only one detailed screen. The result of this study showed that the higher spatial ability participants could complete the task faster without the overview window (Search task). However, the lower spatial ability users could complete the task faster given the overview window. This study is a good example for showing a way that spatial ability can affect users. Ziefle and Bay [117] constructed an experiment by providing two types of navigation aids on a mobile device screen. The first aid was a category aid and the second one was tree aid. The category aid had a header which contained a link to subcategories. The tree aid was almost the same, except that it was able to show the links with a deeper level based on the properties of the trees. They discovered that the second option was more useful for the users with low spatial ability and older adults.

Kozlowski and Bryant [44] noted that there existed a difference between people with a good sense of direction and people with a poor sense of direction. The first group did better at pointing to unseen goals. The accuracy of the first group was also improved if the additional exposure in which orientation was emphasized was given whereas no change occurred for the poor group.

One factor to be considered when designing software for users with varying spatial ability is the informative instructions on the interface. Craik and Lockhart [14] noted that the

depth processing was directly variable to the depth of the encoding. Nielson [72] suggested that minimizing user memory requirements is the appropriate way to study usability because users could focus on the task on the software screen more than thinking about how to work on the task. A guided software could help by reducing the usage of a user's memory.

The software user interface that comes with informative guidance significantly relates to the use of internal memory and the cognitive process [115]. Zhang [115] did conduct an experiment on external information and revealed that external representations had the ability to impact cognitive behavior. The information that has passed from internal memory to the external display enhanced effectiveness. Zhang's conclusion was that the external information was able to be an advantage if the use of it could compensate for the cost of that external information. van Nimwegen and van Oostendorp [108] discovered that an unguided software interface yielded better results for performance aspects than a guided software interface because the unguided software interface allowed the users to be more involved with the software. The users would have more chance to think and consider with unguided software interface. Although an issue of a guidance interface could be questioned, in our perspective, we believed that it was a matter of users and also the nature of the guidance used. If a user had high spatial ability, he could work well with an unguided interface which allowed him to learn the software while using it. Whereas if a user had low spatial ability but had some background knowledge about the task, a guided interface software could be a better option. Zhang and Norman [116] proposed that the representation of a cognitive task is always both internal and external. Berger, Lu, Belzer, and Voss [7] have found that if a computer system was developed for the purpose of learning by discovering, students who had higher spatial ability would get more benefit as opposed to students who had lower spatial ability. Vicente and Williges [111] published work to show that the difference in software usage performance between the higher spatial ability users and the lower spatial ability users can be reduced by adding some extra guidance on the software interface.

Consequently, spatial ability is a vital individual difference that could be used to determine the potential of software utilization and to design effective user interfaces that yield optimal results for users with different levels of spatial ability. Rusch [89] found a correlation between spatial visualization, and map software usage ability. The purpose of her first study was to examine the direction of the relationship between spatial ability and user performance based on alternative button layouts. The purpose of her second study was to examine whether the relationship was affected by offering task guidance. Rusch [89] made a conclusion that the good button labels could help alleviate the performance difference among users with different levels of spatial ability. The difference between guided and unguided systems could be seen via the way that the information processing was formed. Guided systems always came with external information for guidance as to where the user was in the task. The goal of this kind of information processing was to cause the reduction of working memory usage because the users did not need to recall information to keep track of as long as they had the guidance along all steps when using the software. The strategy to design the appearance of those external information was challenging because it came with two negative effects. The first negative effect was that the software interface could come up with overkill of information, which required the users to use more working memory to digest the extra information. The second negative effect was that it reduced the chance of the user to explore and learn the software by themselves. For the unguided systems, there was no guiding information on the software screen. Any information that users gathered had to be stored in the memory, which became internal information. The advantage of internal information, or internalization, was that the user needed to do explorative learning process. Hence, recalling of information in the unguided systems was better than recalling of information in guided systems because it solely depended on the knowledge stored in the working memory. Nevertheless, the problem of internal information came when dealing with low spatial ability users. Rusch [89] concluded that unguided systems would benefit

the users with higher spatial ability which involved the effective memory usage for software interface as long as the amount of information was reasonably suitable. However, the guided systems would be the most appropriate option for the users with lower spatial ability. Both individual difference and designing the systems, regardless of unguided systems or guided systems, played a significant role in affecting the performance of users.

2.2 Interesting studies that relate to spatial ability

Spatial ability, which can be branched into multiple sub-categories such as spatial visualization or spatial cognition, has been shown to be a significant factor in the success of computer users. In this section, we selected some interesting studies about spatial ability and related areas to be discussed.

Landau [51] gave a definition of spatial cognition as “Capacity to discover, mentally transform, and use spatial information about the world to achieve a variety of goals, including navigating through the world, identifying and acting on objects, talking about objects and events, and using explicit symbolic representations such as maps and diagrams to communicate about space” [Landau. “Spatial cognition.” *Encyclopedia of the human brain* 4, p.395-418, 2002].

Some work had been done to answer the question whether spatial ability reflected human performance. Cutmore, Hine, Maberly, Langford, and Hawgood [15] discovered that a gender matters. Based on their result, males gathered survey knowledge from the task faster than females. Moreover, the study also confirmed that more proficiency in spatial cognition resulted in better performance. Geary, Saults, Liu, and Hoard [27] found that males got higher scores than females in arithmetical computations, arithmetical reasoning, and spatial cognition measures. The reason was that males have an advantage in computa-

tion and spatial cognition. Although this study did not relate directly to our dissertation proposal, we can imply that a spatial ability mattered because men, which had higher score in spatial ability test, can do better in several tasks such as those authors mentioned.

Golledge and Gärling [28] explicitly examined the relationship between cognitive maps and travel behavior. The authors noted that there was a relationship in transportation between travel behavior, path selection criteria, navigation and way finding, route learning, etc. Those issues were prioritized by individual differences. Halligan, Fink, Marshall, and Vallar [30] studied the structure of spatial process and mental representation using neuropsychological syndrome of unilateral spatial neglect. Unilateral spatial neglect is neuropsychological condition such that a person who has brain damage to one hemisphere loses an ability to cognize one side of the body or environment. From the study, they found that the spatial cognition subserving process consisted of multiple domains and it was extensive.

Hegarty, Montello, Richardson, Ishikawa, and Lovelace [31] tested 221 participants for their psychometric measures of spatial abilities, spatial updating, verbal abilities, and working memory. Their experiment consisted of learning from real world environment, learning from desktop virtual environment, and learning from environment provided by a videotape. This idea led our research group to design multiple platforms of maps in our study. Before this proposed study, our research group did similar experiments on Windows desktop, Windows Mobile device, virtual reality, and an actual survey on a paper map. Ishikawa and Montello [37] also conducted a study about spatial acquisition from direct experience in the environment by studying twenty four college students. Those participants were driven using two routes for ten weekly sessions and were asked to illustrate the routes' spatial properties based on what they had learned. Most of them could not generate the correct result. Ishikawa, Fujiwara, Imai, and Okabe [36] worked on a study that related spatial abilities and wayfinding using a Global Positioning System (GPS) feature on mobile nav-

igation compared to a paper map approach and direct experience. The outcome showed that the GPS participants consumed more resources, such as time and distance, than the other two approaches and also provided more errors. From this study, we questioned that a participant might rely on GPS when it was active since GPS was the best landmark on any map software. This issue led us to include GPS feature as a treatment in our Study I (Chapter 4) to verify its usefulness.

Klippel, Tappe, and Habel [41] investigated a process called spatial chunking. Spatial chunking is a process where users are able to combine the route segment in route direction. The authors divided route directions into two types: advanced route direction and accompanying route direction. The first one was used to give the directions before traveling. In contrast, the second type gave the directions during traveling. Spatial chunking was investigated for the second type. The authors noted that spatial chunking was effectively conceptual and robust and related to directional tasks.

Kuipers [47] introduced the spatial semantic hierarchy, which is a model of large scale space knowledge by integrating information from the human's spatial cognitive map. This model could be implemented to enhance both human cognitive map and robotic applications. Kuipers, Modayil, Beeson, MacMahon, and Savelli [46] extended their work to a hybrid version, which supported both small-scale space and large-scale space.

Kulhavy and Stock [49] did a study about how people learned and remembered cognitive maps due to the relationship between human cognitive systems and cartographic maps. In the cognitive systems, the limitation of working memory affected the performance of information acquisition and retrieval. It also solely influenced a map image regardless of the characteristics of a map [69][95]. This showed that a human's cognitive ability causes a difference in user performance.

Another interesting map-based study on spatial ability was done by Lanca [50]. She did an experiment to verify whether participants created three-dimensional representations of contour maps in the process of learning and retrieving or not. The study was started by asking participants to study contour maps and landsurface maps. The author partitioned participants into four groups. For the first group, participants did both a cross-section test and a recognition test on the contour maps. For the second group, participants did a cross-section test on the contour maps and a recognition test on the landsurface maps. For the third group, participants did a cross-section test on the landsurface maps and a recognition test on the contour maps. For the last group, participants did both tests on the landsurface maps. After studying their maps, participants were asked to recall the map they had studied. The result she found was that male participants constructed a three-dimensional representations and recognized maps better than female participants. Furthermore, three-dimensional spatial ability was also suggested as an effective skill in processing long term memory on contour maps.

Phillips [80] investigated a way to improve a readability of contour maps based on the fact that most people could do map reading better on layer tint maps than contour maps [81]. Surprisingly, the results showed that there existed no method or solution to help enhance contour map readability unless it gave a obviousness to contour maps. Thus, to avoid any readability-related effect that can occur by using a contour map, maps that we provided in our software was a layer tint version.

Liben and Downs [54] did an experiment where one adult went to any position of a classroom and pointed straight forward, then children were asked to point to location of that adult and the direction the adult was pointing to on a map. There were two versions of maps in this experiment, which were an aligned map and a 180-degree rotation map. The

result showed that most children performed well on an aligned map. Moreover, younger children failed to understand the point of view and boys did better than girls which corresponded to the authors' hypothesis of spatial concept of mapping. Liben, Kastens, and Stevenson [55] discussed the benefits of instructing children to use maps that were not only for navigation purposes. It was also to enhance their spatial skills. The author developed a curriculum called "Where Are We?" for teaching map navigation in school with the goal to stimulate actual experience of a real world navigation using maps.

There exists some work that is aimed to find a factor that caused spatial abilities to be different among individuals. Linn and Peterson [56] hypothesized about sex difference in spatial ability. They proposed results that sex difference affected only some types of spatial ability, mostly on mental rotation and less on spatial perception. With the evidence from this article, we can expect minimum effect from sex difference in our work because no mental rotation activities are involved. Montello, Lovelace, Golledge, and Self [66] focused more on differences and similarities in geographic and environmental spatial abilities due to sex. They invited forty three females and thirty six males to perform spatial tasks. Those tasks consisted of psychometric tests, campus route learning test, map-learning tests, extent geographic knowledge test, object location memory test, verbal spatial description tests, and self-report measures. The result of the study said that males outperformed females for acquiring spatial knowledge from direct experience, whereas females did better for acquiring knowledge from a map.

Voyer, Voyer, and Bryden [112] also examined sex difference with respect to spatial abilities. The authors invited participants to do tests related to spatial abilities. The result showed that there existed sex differences in the multiple tests along with other differences as well. Their conclusion was that the rate of sex differences were decreased in recent years and it also depended on types of spatial tests. However, Rapp, Culpepper, Kirkby, and

Morin's [84] experiment and a result was counter to the statement that said sex difference mattered. In their study, they invited earth science students to view a map. There were three types of maps which were a map that had shading, a map that had stereo visualization (a map that supported three dimensional perception), and a map that combined both features. Students were asked line-of-sight questions and allowed to use their assigned map. The results revealed that, for this study, sex did not influence the performance of students. Factors that were involved were background characteristics, such as grade or map usage experience. This conflict was caused by a sensitivity to the type of test.

Mou and McNamara [67] investigated the role of intrinsic frames of reference in human spatial memory. Frames of reference can be thought of as a tool to help us to recognize or describe spatial information. For example, if we would like to point out a location on earth, we use a pair of latitude and longitude to be the frame of reference. Thus, spatial ability, especially spatial memory, should always be defined by the concept of frame of reference.

Nadel and Hardt [68] studied the brain in respect to spatial ability. The authors stated that two types of spatial systems, which are egocentric and allocentric, had dissimilar properties. An egocentric spatial system is a process that deals with objects and an environment using one's self-center as the primary origin. But an allocentric spatial system deals with using other objects as a reference related to the environment. One example of distinguishing egocentric and allocentric was navigation using a GPS routing device in an automobile. A driver who preferred to use north-based map (traditional map) tended to have an allocentric spatial system. On the other hand, a driver who preferred to use a track-based map (see the map in the same direction as driver was currently facing) tended to have an egocentric spatial system. Marshall and Fink [60] described ongoing research work related to spatial cognition and brain functions. One interesting aspect of spatial cognition is a human's ability to perceive a distance which was related to brain function. McNamara [63] proposed

a new spatial memory theory based on a spatial reference system, which was a system that involved people learning the spatial structure of their environment. The spatial reference system is not fixed along the learning process. It could be updated to a new one if people perceived a new correct view of their environment.

Penn [79] explored work related to space syntax theory and spatial cognition. The author gave the definition of a cognitive space as a space that encourages our understanding of a visual environment. The co-presence, which was defined by the local visual field and the way that the configuration applied movement routes, influences the effect of spatial configuration in both structural communication and transactions between persons.

Remolina and Kuipers [87] proposed a theory on topological maps. The difference between causal maps and topological maps was that topological maps were a way to represent spatial information in a graph where a node represented a state of an agent's vocabularies and an edge represented a traversal path from one state to another state. However, the representation of each topological map was based on the application. The author pointed out that there were three elements that were common among topological maps. Those were the use of sensory descriptions to identify each node, the use of relations among nodes, and local metrical information that was linked to edges.

Tversky [105] analyzed the characteristics of maps and how they answered what humans needed to represent spatial information. The author noted that maps consisted of five characteristics. The first characteristic was that maps were two-dimensional. For cognitive achievement, humans created two-dimensional representations for three-dimensional environments and three-dimensional illustrations were difficult to construct and handle. Second, maps ignored information. Third, maps were regularized. Fourth, maps provided inconsistent scale and perspective. The obvious example of this characteristic is the tra-

ditional world map that we frequently see on classroom boards. Last, the map contained a message, for example; aesthetic, political, spiritual, and humorous. Thus, in terms of spatial representation, effective communication had higher priority than correctness as long as it satisfied human cognition.

So, we turn our focus to the question: why is a map an important tool in the study area of cognitive, learning memory and spatial ability? A map is one type of graphic, which was used to represent real entities such as objects, places, or environments. Tversky, Morrison, and Betrancourt [106] mentioned that the graphic has been used to illustrate both inherently spatiovisual data such as maps or floor plans and metaphorically spatiovisual, such as organization charts or mind maps. With the visual-oriented property, graphics literally increase human potential in inference, communication, learning, and memory. Moreover, computer systems that were difficult for users could be understandable by graphics. Maps, which simply are a representation of the real world environment, could be an effective measurement when we were dealing with spatial visualization.

In 2000, Uttal [107] studied the relation between map use and spatial cognition development. The author invited children to participate in his experiment. He found that learning from maps helped children to develop their spatial cognition and also helped them understand the concept of space. A significant difference between this style of learning and learning from one's direct environment also exists. Furthermore, large-scale spatial learning was also observed via map use.

Kim and Penn [40] investigated the relationship between spatial syntax of cognitive maps and the real environment by asking the residents of Hampstead Garden Suburb to participate in sketching cognitive maps of their place. The result of this study found that the relationship between spatial syntax in cognitive maps and the real world environment

was positive. What can implied from this study was that handling a map was a significant activity that depended on human spatial ability or spatial visualization in particular.

Zipf and Richter [118] also mentioned the importance of maps as an effective representation material for spatial knowledge. Geographic Information Systems (GIS), which support the implementation of maps as feature of a software system, are very popular in multiple task units.

2.3 Impact of spatial visualization (VZ) in computer-based applications

There were several abilities that were categorized as spatial ability. Spatial visualization (VZ) was one of them. In this dissertation, we aimed to develop an adaptive user interface software for users with different level of VZ.

Salthouse, Babcock, Mitchell, Palmon, and Skovronek [90] defined VZ as “mental manipulation of spatial information to determine how a given spatial configuration would appear if portions of that configuration were to be rotated, folded, repositioned or otherwise transformed” in their study report [Salthouse, Timothy A., et al. “Sources of individual differences in spatial visualization ability.” *Intelligence* 14.2 (1990): 187-230]. They did two studies to find sources of differences in VZ. For the first study, there were fifty participants. Each participant participated in five sessions. In the first session, a participant was asked to do four paper-and-pencil tests and the WAIS-R Block Design. A participant was asked to do a working memory task and paper folding task in session 2. In session 3, a participant did cube folding and block design tasks. A participant did a task of cube comparison in session 4. The spatial integration task was done in session 5. For each test, a participant could score based on number of corrected items/answers. A result of the first study did lead them to a hypothesis that differences in VZ played a major role in differences in internal

representation quality. This hypothesis drove them to do the second study to find more deeper detail by comparing multiple characteristics of people with both high and low VZ. From study 2, they found that diversification in VZ did not correspond to diversification of encoding spatial information ability nor properties of information.

Norman [74] defined an impact of VZ to computer-based technology usage performance as “The primary cognitive factor driving differences in performance using computer-based technology is spatial visualization ability” [Norman, Kent L. “Spatial visualization - A gateway to computer-based technology.” *Journal of Special Education Technology* 12.3 (1994): 195-206]. In his experiment, he wanted to find out which technique was the best to compensate low VZ people. Those four techniques were *spatial metaphors, graphical user interface, interface apparency, and interface manipulatability*. The spatial metaphor was a technique that encouraged users to think of metaphors of system UI. For instance, users could view a UI of WYSIWYG word processor software as a classic typewriter. This technique could help users to recall their past experience to help improving their performance. The graphical user interface was a technique that implementing a software interface using spatial representation object such as buttons, sliders, switches, etc. The interface apparency was a technique that designing a graphical user interface that made relationship among items on an interface visible to users, which could be done in many ways. A good example of the interface apparency technique was to use a graph or a diagram. The interface manipulatability was a technique that reducing/decreasing components that required spatial processing off an interface. This could be done by creating an interface that allowed users to directly handle any spatial item on the interface. He discovered that the latter technique was the best to compensate users with low VZ. Multiple literatures found that VZ was an important factor, which could be used as a predictor for success in real-world applications including software usage performance [59][62][96][26].

Salthouse, Babcock, Skovronek, Mitchell, and Palmon [91] questioned whether people with different age and experience would have different VZ. They did studies to confirm a hypothesis that for adults, age was inverse variation with VZ regardless of their VZ experience. Their studies consisted of three studies. For the first study, they asked participants to do four tests of VZ: foam board test, paper folding test, surface development test, and cube comparison test. All of them were paper-and-pencils tests. In the second study, participants were asked to do extra four computer-controlled tasks. Those extra tasks were the paper-folding task, the cube-folding task, the spatial-integration task, and the cube-comparison task. The third study contains the surface-development task, the paper-folding task, and the spatial-integration task. The last two tasks were computer-controlled. According to these studies, they concluded that if an age was increased, then, a level of VZ was dropped. This assumption was also true regardless of previous experience about VZ.

Age was not the only factor that differentiated VZ, sex difference also did [56][112][34][61]. Kaufman [38] did a study to find that a difference in working memory capacity between different genders involved in this inequality of VZ or not. He did tests of three-dimensional mental rotation and VZ together with tests of spatial working memory and verbal working memory. A result revealed that spatial working memory engagement was a reason that made different VZ between genders.

2.4 The location-based survey software

Location-based software is an application software that has a location in terms of a pair of geographic coordinates as preliminary data or input data to the computing process. Most location-based software helps improve a user's task or comfortability. The examples of these applications are Google Map[®], Bing Map[®], GPS-navigation software, Yelp[®], etc. In the academic field, location-based software is also an interesting research topic. The address

verification task is a procedure used by the Bureau of Census where participants are asked to verify whether the street address indicator on a map exists and if so is it in the correct position. Whitney, Batinov, Miller, Nusser, and Ashenfelter [113] conducted a study to see whether different levels of cognitive ability affected performance for the address verification task. They used three tests to identify a participant's level of cognitive ability, which were a spatial visualization test, a visual memory test, and a perspective-taking test. They provided a paper map for participants to verify street addresses in the real neighborhood.

Willis, Hölscher, Wilbertz, and Li [114] did a study to compare the spatial knowledge acquisition with different sources: paper maps and mobile maps by two groups of participants. The first group was assigned a task to learn an environment from a paper map. Another group was assigned a mobile map. The participants with mobile maps did worse on the task of distance estimation. What we can imply from this study is that the map software with different map size affected the usability.

In 2012, Chellappan [12] developed a location-based map application on the Microsoft Windows Presentation Foundation platform. The software focused on the way to apply the concept of an object-oriented design to map components. The software read the ESRI Shapefiles for the map space and then created objects representing all map components such as streets or rivers. On the software interface, it drew geometrical objects to render these map components. Each geometrical object rendered each map component object one by one. However, because the application had to read the Shapefiles every time it did perform a map operation, it did tend to be rather slow. Thus, when dealing with large size of Shapefiles, this overhead caused significant impact on the performance of the software. Anytime the user would like to pan/zoom the map, the application would re-process and re-draw the map again. It should be noted that our Object-Indexing described in Chapter 3 has been built on the objects defined by Chellappan.

Some studies have attempted to find the relationship between the location-based software and individual difference. Batinov, Whitney, Miller, Nusser, Stanfill, and Ashenfelter [3] examined the impact of spatial ability of diverse users by asking them to use their software in two environments: the real world environment and the virtual environment. They created two regression models for time and distance, respectively. For both regression models, spatial visualization was a significant metric. For the regression model for distance, an interaction between the environment (real or virtual environment) and spatial visualization was suggestive. The result revealed that there was no significant difference between low spatial ability participants and high spatial ability participants based on the type of environment. Thus, they found that spatial visualization is a significant factor from the study.

2.5 The adaptive user interfaces (AUI) and problems that lead to it

An adaptive user interface (AUI) is one kind of user interface that has the ability to adapt itself to the level of skills of the user. The purpose of an adaptive user interface is to help minimize the gap between users with different capabilities and software. Varying user ability is an important part that needs to be considered for developing application software. Benyon [4] stated that understanding the individual differences between users was important in terms of increasing the usability for users with different spatial ability. Stanney and Salvendy [97] conducted a study to assign the searching task to users with high and low spatial ability. They found that the lower spatial ability group would be assisted by visual mediators. The result showed that the use of a visualization technique to assist the low spatial ability users yielded better software usage performance.

Benyon, Innocent, and Murray [5] defined an adaptive system as “Adaptive systems are systems which can alter aspects of their structure, functionality or interface in order to

accommodate the differing needs of individuals or groups of users and the changing needs of users over time” [5].

Ramalingam [83] pointed out the dissimilarity between adaptive user interfaces and adaptable user interfaces. Adaptable user interface was a user interface that users were given the options, whether directly or indirectly, to select among set of multiple possible outputs of interface. Whereas, an adaptive user interface might or might not be fully controlled by users.

Schmidt, Beigl, and Gellersen [92] mentioned the importance of the surrounding information as the parameter used to approximate context, which they used to develop context-aware applications. They proposed two implementations, which consisted of a light-sensitive display and an orientation-aware PDA interface. Both were using a sensor technology to implement. For a light-sensitive display, they attached the light sensor to the PDA device (Palm Pilot). When the light sensor sent information about the light condition to the software, the software display adapted its backlight based on retrieved information. For an orientation-aware PDA interface, they attached the sensor to another PDA device (Newton MessagePad) to detect the device’s orientation. So the orientation of the software user interface rotated the same way as the orientation of the PDA. Extending their two implementations of sensor, they presented a new approach to move to the next level of context that was more abstract than light condition or orientation. For example, a user interface should adapt based on whether a software was used indoor or outdoor, etc. With this approach, they used a concept of multi-sensory context-awareness, which consisted of four components. Those four components were sensor, cues, context, and scripting layer. The data from the four components were combined to answer what is the current context (e.g. indoor) for further user interface adapting. For example, if a context is “indoor”, combined data from the four components might consist of artificial light, room temperature, etc.

A number of studies and experiments concerning designing the interface based on individual differences have been developed. Newell and Gregor [71] proposed a new paradigm for developing software interfaces by considering universal access, which included users with disabilities. Keates, Clarkson, and Robinson [39] presented the cutting edge methodological approach for interface design, which was called “Inclusive Interface Design”. The purposed of this approach was to design a user interface that could support users with varying capabilities.

Pattison and Stedmon [78] suggested the way to implement an inclusive design is to design a mobile phone for older users. This kind of user had different requirements compared to younger users because of these factors: vision, hearing, motor function, and cognitive aspects of aging.

Ramachandran [82] used two adaptation techniques in order to develop an adaptive user interface for health care applications, which were “adaptive presentation” and “adaptive navigation”. The objective of adaptive presentation was to personalize a content to be displayed to different types of users. The objective of adaptive navigation was to customize the way to do the task based on task types and users. The example of how adaptive presentation worked was that if there was an input “anesthesia level” to the software system in the process of creating a patient’s profile, this input mattered to doctor users. Thus, the system showed this input to doctors. Whereas the system did not show it to system administrator users because system administrator users can do nothing about this type of input. Another example was the example of adaptive navigation. If there was a section “blood test” in the system, the system should display a single page to administrator users in order for administrators to create a patient’s profile. Whereas it should display multiple tabs to the doctor who were in charge of the blood test task.

Findlater and McGrenere [22] conducted a study to compare the preference between three kinds of software user interfaces, which were static menus, adaptable (user-controlled) menus, and adaptive (system-controlled) menus. The results showed that the static method was faster than the remaining methods but the adaptable and adaptive user interfaces gained the advantage in terms of customization and flexibility.

Rogers, Fiechter, and Thompson [88] investigated the effect of adaptive user interfaces in an automobile system. They provided three adaptive user interface systems, which were Adaptive Route Advisor, Adaptive News Reader, and Adaptive Place Advisor. The objectives of these systems were, respectively, to navigate, to read news and stories, and to select the restaurant. These systems gathered information from the interaction between the driver and the user interface. They created a user model and adjusted the interface based on the driver preference. The purpose of this study was to show that replacing original tasks while driving a car with adaptive user interface could reduce the risk of accidents. With this study, a driver did not need to use a smartphone (while driving) to find a restaurant address nor take a look at a GPS. Those activities were replaced with adaptive user interface system functions.

Kules [48] suggested multiple methods about modeling user behavior and adaptive software systems. One of those methods was to design the adaptive user interface using the analysis of user behavior modeling.

Ramalingam [83] noted that a rigorous solution that could deal with an adaptive user interface and individual differences was to develop the software that was capable of adjusting and adapting its interface based on user preference as well as individual difference. The automated adaptive system became essential because of the diversity of factors.

Langley [52] described the effort to develop adaptive user interface using machine learning techniques. Horvitz, Breese, Heckerman, Hovel, and Rommelse [33] implemented a Bayesian user model on the interaction and background of the users to create an intelligent user interface. Bahanovic (1998) constructed a simulation of systems that applied two strategies to return the related documents based on the content and the learning process from the user model. Liu, Wong, and Hui [57] designed a dynamic adaptive user interface that learned user behavior and provided the adaptive assistance based on such behavior.

Viano et al. [109] introduced an Auto-Adaptive Multimedia Interface. This kind of adaptive user interface approach supported the adaptability of the user interface, which could enhance usability as well. The purpose of this study was to overcome the problem of a tradition multimedia interface that was narrow and limited, which could support only a few types of presentation (visual and auditory) and lack of adaptivity. So it could not handle an emergency situation such as a very high rate of input data stream. The Auto-Adaptive Multimedia Interface was developed using collaboration with an intelligent agent. In this project, the method of Auto-Adaptive Multimedia Interface tried to map between the process data and MMI (multimedia interface) objects. The consequence of this mapping was that the system (which was called “AMEBICA”) can dynamically select the type of presentation used in the user interface corresponding to the situation occurring in real time. In particular, in chaotic situations the user interface greatly simplified what the user saw in the interface.

Gajos, Czerwinski, Tan, and Weld [24] set up two experiments with three adaptive graphical interfaces and concluded that there were multiple factors that impacted the acceptance of adaptive user interface. They suggested that the adaptive user interface that duplicated frequently-used functions enhanced the user performance and improved satisfac-

tion. Shankar, Louis, Dascalu, Hayes, Houmanfer [94] conducted a user study to examine whether an adaptive user interface could adapt itself based on the users' environment and whether it would improve usability and user experience.

Sukaviriya and Foley [100] examined how adaptive help support could be extended in a knowledge-based user interface environment. They designed a framework called "User Interface Design Environment (UIDE)". This framework can provide a suggestion to adaptive user interface features, such as adapting menus and a layout of dialog box, by combining knowledge model and collected information from users. The result of using UIDE was that it suggested two useful solutions to implement an adaptive user interface. The first solution for an adaptive interface was recognizing menus, dialog boxes, and macro suggestions. The second solution was an adaptive help for users.

Innocent [35] mentioned the trend in software development to be more natural and support interactive systems. The idea of self-adaptive systems and its condition was also discussed.

Kühme [45] introduced an approach to design an adaptive interface software which allowed users to tell the system how to adapt itself. An approach consisted of an adaptive adaptation and implicit adaptation. This work could lead to the implementation guidance of a map-based survey software in which the software itself can interact with users so users can inform the system when they get lost.

Fischer [23] reviewed the HCI-related research about user modeling for an adaptive software to make it be able to perform the task appropriately with different-skill users. Norcio and Stanley [73] suggested that an adaptive interface has to encompass knowledge of the interaction, the system, the task domain and the user. Their efforts to delineate information

for both humans and computers was important for developing effective human-computer systems. The problem that came with this approach was that the range of adaptive behavior was narrow.

Mezhoudi [64] implemented a Rule Management Engine to enhance the interaction between a user and the user interface. The concept of the work was to allow users to instruct the adaptation strategies based on the feedback from users.

Miller and Thomas [65] identified behavioral problems that they linked to the usage of interactive computer systems by non-professional general users into two topics: System Characteristics and Interface Characteristics. Langley [52] suggested using machine learning to help create an adaptive user interface. The outcome of the study was that there were two characteristics of adaptive user interface that could distinguish it from general user interface, those were informative interface and general interface. The limitation of domain has still been the issue with this improvement.

Ramalingam [83] mentioned that most field tasks are location-based software applications because they provided an advantage to the workers in the field. However, implementing location-based software applications involve multiple issues, such as the screen size of the computing device, the difference of the individual user abilities, or the extreme conditions such as the unexpected incidents of the environment, for instance, the weather condition and the visibility condition. The software that can support an adaptive user interface might be the solution for this problem. The advantage of the adaptive user interface is that it helps promote the effective use of the software in the critical situations.

In the same study, Ramalingam [83] also proposed that one way to develop software with an adaptive user interface was to build on Taylor et al.'s [102] error model. The re-

sults from his work consisted of two models. The first error type was reversal error. Any tapping error would be counted as reversal error when the user tapped on the software screen followed by, immediately, another tap that reversing the first action. For instance, in the map application, users tap zoom-in button then immediately tap zoom out button. The second error type was missing error. Any tapping error would be counted as missing error when the users attempt to tap on some component (such as a button or a check box) but that tap does not occur inside the area of that such component. There was a threshold value representing the distance imposed by the software to distinguish whether the tap is considered missed or not. These errors would guide the software user interface to adapt itself based on the errors made. He also proposed that modeling the user behavior might be the good answer for designing adaptive user interface because the resulting software would have an interface that naturally adapted based on how well the interaction between the user and the interface was going.

There are several methods to define how an adaptive interface should adapt itself. User modeling is one method that can be used to understand an individual user. This method yields an adaptive user interface that solves the problem, which can be vary based on each software, because the interface will change according to the current user at the time of use. Tsandilas and Schraefel [104] presented a user model of interaction for adaptive hypermedia, which was the technique that combined the functionality of adaptive interface and direct-manipulation interface. Taylor et al. [102] examined the implementation of an adaptive web application for elderly users using error detection for modeling the users. They compared the results between error detection methods and observation methods (labor intensive) and found them to be comparable.

Knutov, De Bra, and Pechenizkiy [42] did a study focused on using a versioning approach that enhanced adaptive hypermedia behavior. The versioning approach could be

extended to our study using multiple versions of the user interface on the same survey task. However, we did not use versioning approach because we used different treatments to test how each software feature affected a participant's performance instead (Chapter 4). Steichen, Ashman, and Wade [98] ran a survey to compare adaptive hypermedia and personalized information retrieval as both of them were the trend for adaptive system that adjusted based on the preference of users. The authors performed a study by examining the retrieval process. The results revealed that there existed some differences between those fashions. However, they have shared the same principle, which was to make the software interactive to handle users with different skills. In our proposed work, the idea of adaptive hypermedia is more related than personalized information retrieval due to the limitation of software screen and number of operations.

Another information retrieval related work that was interesting is the study of de Campos, Fernández-Luna, Huete, and Vicente-López [17]. The authors proposed multiple methods for XML information retrieval such that it helped the system to return the most related XML document based on user preference. The approach they presented was the combination of query reformulation, results re-ranking, and model modification.

Kotzyba, Siegert, Gossen, Wendemuth, and Nürnberger [43] designed a system that was able to support both children and adult users by enabling a voice controlled search module. This was also a good work about adaptation in a software layer. The result of this work showed that user interaction was increased.

2.6 Task-based issues

Task-based surveys were the scenery that were involved in most user-related experiments. The objective of task-based survey are to assign a specific task for participants to perform. This could be done with/without the engagement of software.

Rath, Devaurs, and Lindstaedt [85] studied components that impacted task detection performance using their ontology approach (classifiers, features, task types, and methods for training classifier). Those components worked well for classifying both knowledge intensive tasks and routine tasks. The authors utilized their methods on a computer desktop. This approach led us to the platform of portable devices where we focused on a task performed by participants.

In Chapter 3, we introduce our development of a data structure for indexing survey components, “Object-Indexing”, which overcomes a drawback of a traditional raster map software.

CHAPTER 3. OBJECT-INDEXING

This chapter focuses on the details of our development of a data structure for indexing survey components, such as streets and intersections, on user maps. We use the name “Object-Indexing” in this presentation to represent our approach.

3.1 Introduction

To our knowledge, the Object-Indexing that we have designed and implemented is original work. While it is used in this dissertation proposal to index streets and intersections in address verification tasks, the Object-Indexing can be applied to any location-based application that there are identifiable map components that can be indexed.

3.2 Design of Object-Indexing

Based on the necessity of map usage in survey tasks, Chellappan and Miller [13] proposed a way to directly access and operate on every survey unit on a map such as housing units, streets, intersections, etc. Chellappan and Miller [13] applied an object-oriented approach to a vector map by generating a map using plotting tools in C# language instead of loading an image to represent a map. With this approach, every survey unit has an accessibility to a user. This implementation lead us to realize the benefit of accessibility, which is essential in survey tasks. Thus, our Object-Indexing approach needs to be able to preserve this property. The problem that Chellappan and Miller [13] faced in their work was that the software was extremely slow as any action on the map required them to redraw.

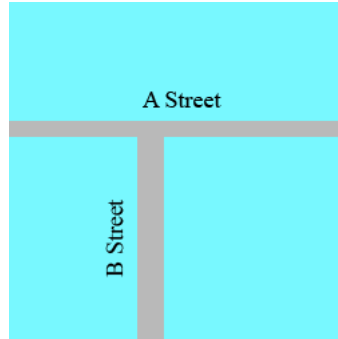


Figure 3.1 Example of a Vector Map Generated from a Shapefile.

As we have mentioned in Chapter 1, the goal of the Object-Indexing is to combine advantages of both raster maps and objects based on vector map coordinates. A benefit of a raster map is that it is simple and fast to implement. A raster map is simply an image of a map combined with a world file. The cost to handle a raster map is not much more complicated than handling one single image. Hence, the loading time is quite fast. On the other hand, the vector map used by Chellappan and Miller [13] was generated by processing shapefiles [21] and creating geometrical objects as map components such as streets, railways, or intersections. The shapefile objects are indexable. However, important map components such as streets are still an issue because they cannot be directly indexed. Figure 3.1 shows a vector map with two streets, A and B, illustrated from three polygons (three rectangles for this example). These three rectangles are accessibly indexed because they have records in the shapefile. But the important map components, which are street A and street B, cannot be accessed.

Without an ability to directly access primary map components, we need to create a new data structure to overcome this issue. There are many map components that can be counted as primary map components based on the purpose of applications. For the address verification task, streets and intersections are the primary map components. For a National

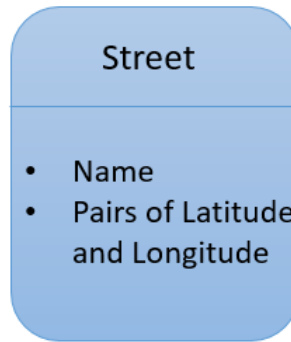


Figure 3.2 The Conceptual Class of Street Object.

Resources Inventory application, wetland areas and water ways are primary map components.

Our object model is built on the work of Chellappan and Miller [13]. The difference between our work compared with Chellappan and Miller [13] is that our Street object, for instance, mimics a real world street by having all geometrical data, pairs of longitude and latitude, stored as a property of a Street object. This is required since our components are actually indexed objects rather than map representation objects. Figure 3.2 shows the conceptual class of our Street object. Figure 3.3 shows the concept of creating a Street object from processing a shapefile [21]. The list of pairs of longitude and latitude can be seen as a line type in more traditional shapefiles.

As previously mentioned, primary map components such as streets, intersections, or wetlands cannot be indexably represented by records read from shapefiles based on polygons. We need to define a new object model for those primary map components based on data that we have, which are records from shapefiles. In this section, we would discuss three major types of object models that can be implemented as an indexed object for primary

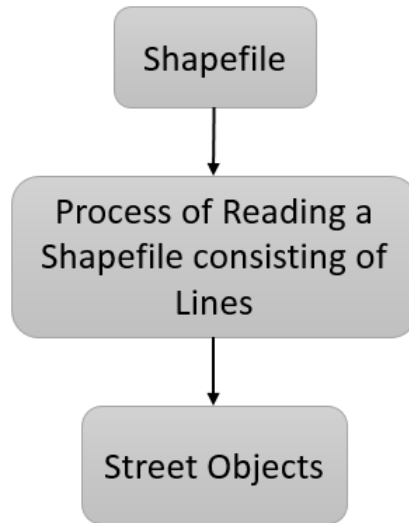


Figure 3.3 The Concept of Creating a Street Object.

map components. Those three types are Line, Point, and Area. We can view these object models as an abstract class that can be used to instantiate an indexed object for primary map components.

3.2.1 Line

A line is a conceptual type of our object model that can be implemented as an indexed object for any pathway or driveway components such as streets or waterways.

For Line type, we start by giving an example of a street, which is a primary map component corresponding to the Line object model. We define a Street object as an indexed object represented a street. A Street object is generated from a Street class, which is extended from the Line object.

According to Figure 3.3, we first read a shapefile [21] to gather all geometrical data of each street. This is the same starting process as Chellappan and Miller [13]. Next, for each street, a Street object is instantiated to store a street name and all pairs of longitude and latitude that are read from a shapefile [21]. When a shapefile [21] is completely read, all Street objects are generated and ready for the Object-Indexing to operate.

After the process of creating Street objects has been completed, we would have the Street objects and a raster map, which is an image combined with a world file. Those two items are necessary to implement the Object-Indexing. We recall that our objective is to index primary map components such as streets and intersections so those map components are accessible to users. In the perspective of portable device usage, which comes with a small screen, accessing a map component in a map means that users could be able to tap on a map component they are focusing at and be able to retrieve properties of the map component. The most obvious example is that users tap on an unknown street on a map, then, a name of that street is displayed on the screen.

As we previously stated that the conceptual structure of Object-Indexing consists of a raster map and Street objects. We would describe this conceptual structure using the following explicit example, which is similar to Figure 3.1. In this example, we have one raster map (Figure 3.4) and two Street objects: A street and B street. For the raster map (Figure 3.4), there are two streets in this map. The horizontal street is A street. The vertical street is B street. There is not any street name displayed in this map.

There are two Street objects represented by A street and B street. For each Street object, based on 3.2, it stores a street name and geometrical data (pairs of longitude and latitude that street locates on). Note that all geometrical data come from reading a shapefile [21]. Figure 3.5 illustrates the conceptual Street object of A street. A street contains

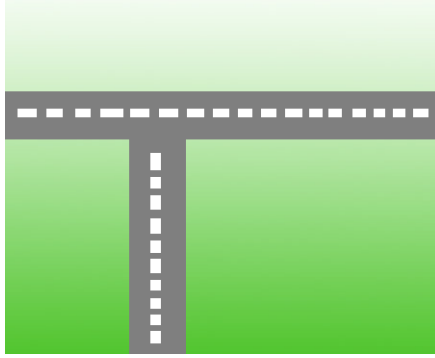


Figure 3.4 A Raster Map Consists of Two Streets without Street Names.

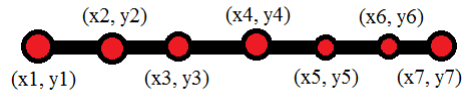


Figure 3.5 Street Object of A Street.

seven pairs of longitude and latitude. A pair of longitude and latitude is represented in the format of (x, y) where x is a longitude and y is a latitude. Thus, all seven pairs of longitude and latitude belong to A street are (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) , (x_6, y_6) , and (x_7, y_7) . Figure 3.6 illustrates the conceptual Street object of B street. B street contains five pairs of longitude and latitude, which are (x_1', y_1') , (x_2', y_2') , (x_3', y_3') , (x_4', y_4') and (x_5', y_5') . In Section 3.3, we would talk about how Object-Indexing works using the raster map (Figure 3.4) and Street objects (Figure 3.5 and 3.6).

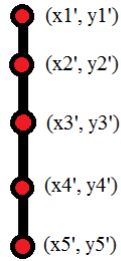


Figure 3.6 Street Object of \mathbb{B} Street.

3.2.2 Point

A Point is a conceptual type of our object model that can be implemented as an indexed object for any point component, such as intersections.

For Point type, we give an example of an intersection, which is a primary map component that corresponds to the Point object. We define an Intersection object as an indexed object represented an intersection. An Intersection object is generated from an Intersection class, which is extended from the Point object model.

Figure 3.7 shows the conceptual class of our Intersection object. Figure 3.8 shows the concept of creating an Intersection object from processing a shapefile [21].

Based on Figure 3.8, we first read a shapefile [21] to gather all geometrical data of each intersection in a record of point. This is the same starting process as Chellappan and Miller [13]. Next, for each intersection, an Intersection object is instantiated to store a name, a latitude, and a longitude that are read from a shapefile [21]. When a shapefile [21] is completely read, all Intersection objects are generated and ready for the Object-Indexing to operate.

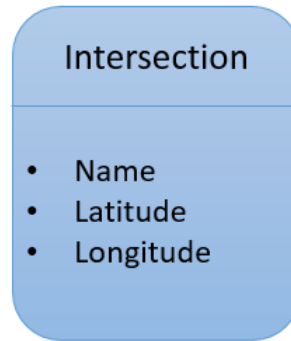


Figure 3.7 The Conceptual Class of Intersection Object.

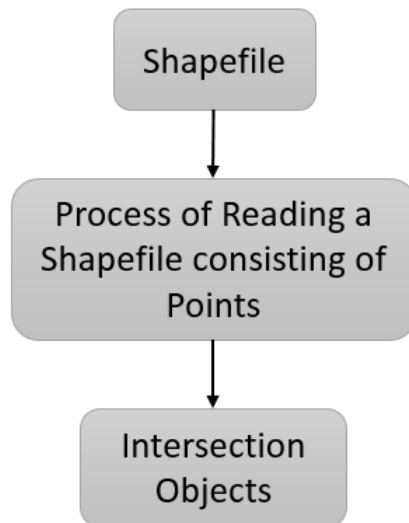


Figure 3.8 The Concept of Creating an Intersection Object.

After the process of creating Intersection objects has been completed, we would have all Intersection objects and a raster map, which is an image combined with a world file. We recall that our objective is to index primary map components such as streets and intersections so those map components are accessible to users. In the perspective of portable device usage, which comes with a small screen, accessing a map component in a map means that users could be able to tap on a map component they are focusing at and be able to retrieve properties of that such map component. The most obvious example for this case is that users tap on an unknown intersection on a map, then, the names of streets that intersect are displayed on the screen.

We would describe this conceptual structure using the following explicit example. In this example, we have one raster map (Figure 3.9) and one Intersection object: \mathbb{I} intersection. For the raster map (Figure 3.9), there is one 5-way intersection, which is \mathbb{I} intersection, in this map. There is not any intersection name displayed in this map.

There is one Intersection object represented \mathbb{I} intersection. For the Intersection object, based on Figure 3.7, it stores a name and geometrical data, which are longitude and latitude, that the intersection locates on. Note that all geometrical data come from reading a shapefile [21]. Figure 3.10 illustrates the conceptual Intersection object of \mathbb{I} intersection. \mathbb{I} intersection contains one pairs of longitude and latitude. A pair of longitude and latitude is represented in the format of (x, y) where x is a longitude and y is a latitude. Thus, a pair of longitude and latitude point belong to \mathbb{I} intersection is (x, y) . In Section 3.3, we would discuss how Object-Indexing works using the raster map (Figure 3.9) and Intersection object (Figure 3.10).

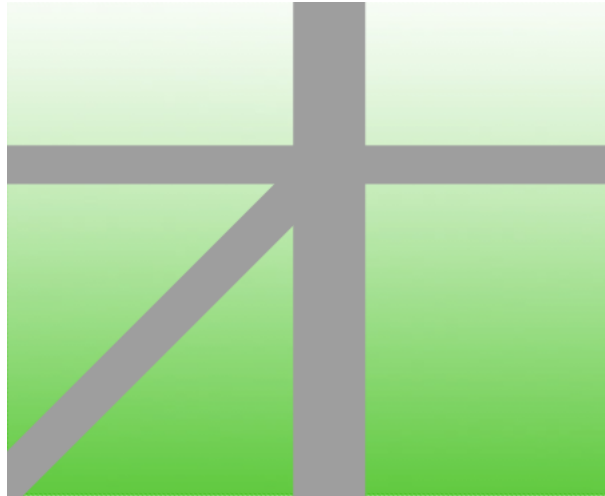


Figure 3.9 A Raster Map Consists of One 5-way Intersection without Name.



Figure 3.10 Intersection Object of I Intersection.



Figure 3.11 The Conceptual Class of Wetland Object.

3.2.3 Area

An Area is a conceptual type of our object model that can be implemented as an indexed object for any area component such as wetlands.

For the Area type, we would provide an example of a wetland, which is a primary map component that corresponds to the Area object model. We define a Wetland object as an indexed object representing a wetland. A wetland is an area that contains enough water to support wildlife such as a swamp, a pond or a lake. A Wetland object is generated from the Wetland class, which is extended from the Area object.

Figure 3.11 shows the conceptual class of our Wetland object. Since this is based on the area object, the pairs of longitude and latitude form a polygon (i.e. first point and last point are the same). Figure 3.12 shows the concept of creating a Wetland object from processing a shapefile [21].

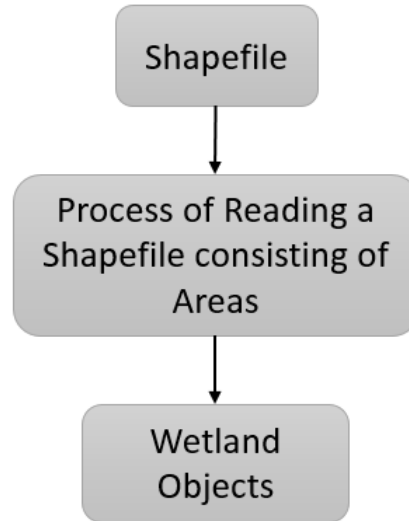


Figure 3.12 The Concept of Creating a Wetland Object.

Based on Figure 3.12, we first read a shapefile [21] to gather all geometrical data of each wetland in a record of polygon. This is the same starting process as Chellappan and Miller [13]. Next, for each wetland, a Wetland object is instantiated to store a name and all pairs of longitude and latitude that are read from a shapefile [21]. When a shapefile [21] is completely read, all Wetland objects are generated and ready for the Object-Indexing to operate.

After the process of creating Wetland objects has been completed, we would have all Wetland objects and a raster map. We recall that our objective is to index primary map components such as streets and intersections so those map components are accessible to users. In the perspective of portable device usage, which comes with a small screen, accessing a map component in a map means that users could be able to tap on a map component they are focusing on and be able to retrieve properties of that such map component. The



Figure 3.13 A Raster Map Consists of One Wetland without Name.

most obvious example for this case is that users tap on any region of an unknown wetland in a map, then, a name of that wetland is displayed on the screen.

We describe this conceptual structure using the following explicit example. In this example, we have one raster map (Figure 3.13) and one Wetland object: W lake. There is not any wetland name displayed in this map.

There is one Wetland object representing W lake. For the Wetland object, based on Figure 3.11, it stores a name and the geometrical data, which are pairs of longitude and latitude, that are boundary points of that wetland. Note that all geometrical data come from reading a shapefile [21] made up of polygon type data. Figure 3.14 illustrates the conceptual Wetland object of W lake. The W lake object contains six pairs of longitude and latitude. A pair of longitude and latitude is represented in the format of (x, y) where x is a longitude and y is a latitude. Thus, the pairs of longitude and latitude that belong to W lake are (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , (x_4, y_4) , (x_5, y_5) and (x_6, y_6) . Because a wetland

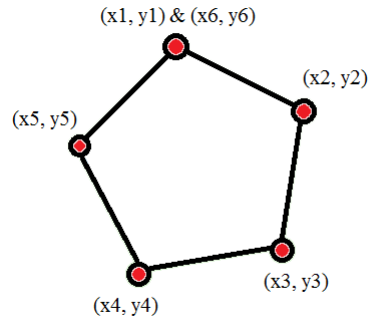


Figure 3.14 Wetland Object of W Lake.

is a closed area, the first and last pair of longitude and latitude must be the same point. Particularly, (x_1, y_1) and (x_6, y_6) are the same location. In Section 3.3, we discuss how Object-Indexing works using the raster map (Figure 3.13) and Wetland object (Figure 3.14).

3.3 Conceptual Object-Indexing Usage

We recall that we would like to provide users with an ability to access primary map components, which means users should be able to tap on a map component they are focusing at and be able to retrieve properties of that map component. In this example, the scenario we would like is that when a user taps on any part of A street in the raster map (Figure 3.4), the name of A street should be displayed and vice versa for B street. Figure 3.15 illustrates this scenario. In Figure 3.15, a user taps on a part of an unknown street (it is actually A street), then, the name of A street is displayed on the screen. So, a user can know that he/she is currently tapping on A street.

To implement this, first, we need to know the location on the map that has been tapped. After we get that location, which is represented by a point (p_x, p_y) , we calculate the latitude

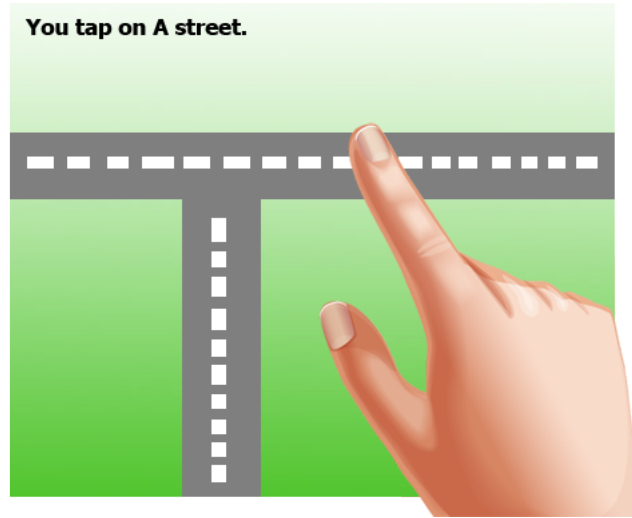


Figure 3.15 Expected Scenario.

and longitude of that location from the point (p_x, p_y) using parameters from a world file comes with the raster map. It contains the latitude and longitude of the top left corner of the map image, as well as the change in latitude as we go down from that point (Δy) and the change in longitude as we go across the image (Δx). Each raster map comes with a uniquely corresponding world file. After we get a latitude (denoted as *lat*) and a longitude (denoted as *lng*) of tapped location, here comes the time that we will use our Street objects. This section discusses algorithms necessary to make the Object-Index work for all three types of our object models: Line, Point, and Area. Each type needs to be handled with a different approach.

3.3.1 Line

For Line object, we start with the example set up in Subsection 3.2.1, which consists of a raster map and two Street objects representing A street and B street (Figure 3.4, 3.5, and 3.6).

We define a square region to cover that tapped location, where the tapped location is the center of this square region, as a region of interest (it is the red square in Figure 3.17). Then we calculate boundaries of this region of interest in term of latitudes and longitudes using the same world file method. Next, we iteratively visit each Street object. For each Street object, we iteratively check each pair of longitude and latitude whether it locates inside the region of interest or not. If it does, that Street object would be marked as a street that a user taps on. If there are more than one Street object satisfying this region of interest condition, it is potentially that a user taps on an intersection of multiple streets. If an intersection is detected, street names of all streets in the intersection will be displayed. Figure 3.16 illustrates a flowchart that describes the algorithm.

In our example, we have two Street objects, A street and B street (Figure 3.5 and 3.6). Therefore, when we apply our conceptual implementation, we have to start from Street object of A street. A street consists of seven pairs of longitude and latitude. We begin with the first pair, $(x1, y1)$, by checking whether this pair locates inside the region of interest or not. If it does, we are done with Street object of A street. A street would be counted as a street that a user taps on. So, we can move to the next Street object. If it does not, we move to next pair of longitude and latitude, $(x2, y2)$, and keep checking. When we complete all Street objects, names of all Street objects that are counted as tapped streets would be displayed on the screen. Based on our example (Figure 3.4, 3.5, 3.6, and 3.15), A street is displayed because a pair of $(x4, y4)$ locates inside the region of interest corresponding to the tapped location. Whereas B street is not displayed because there are no pair of longitude

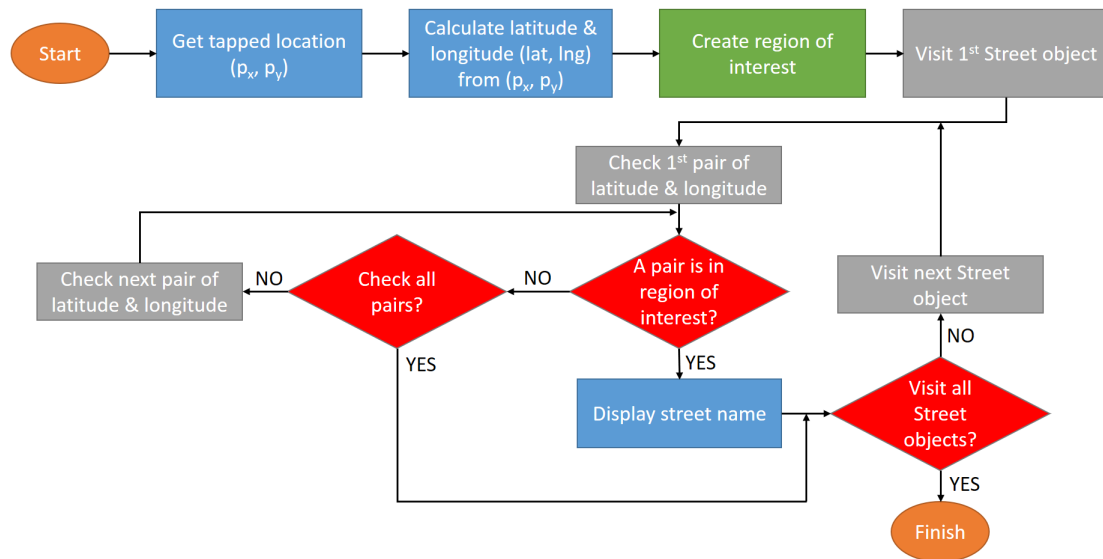


Figure 3.16 Flow Chart of Object-Indexing Implementation for Line Object Model.

and latitude of the Street object of B street locates inside the region of interest. Figure 3.17 illustrates how we can obtain this result.

The complexity of our Object-Indexing for the Line object is $O(nm)$ where n is the maximum number of pairs of longitude and latitude and m is the number of Street objects. Thus, this is a polynomial algorithm.

We briefly explain the user interface implementation of the algorithm shown in Figure 3.16. When a user taps on a street, A red rectangle is drawn to cover the tapped area and returns the name of the street or streets that it can detect (Figure 3.17). A name of a detected street is displayed at the bottom-left of the software screen. Figure 3.18 shows a screenshot of our software where the user has tapped on Chamberlain Street. Figure 3.19 show the entire screen of steps illustrated in Figure 3.18.

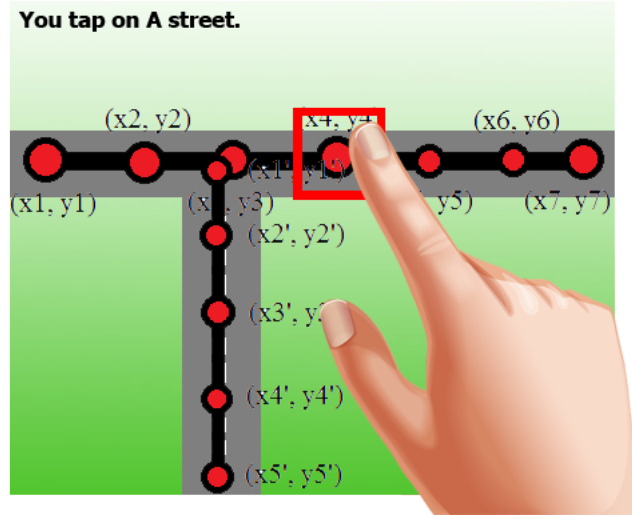


Figure 3.17 How Object-Indexing Operates for Line Object Model.

The Object-Indexing starts working by getting an X-Y coordinate of a point that user has tapped. Next, it calculates a value of X_{max} , X_{min} , Y_{max} , and Y_{min} by the following formula:

$$\begin{aligned}
 X_{max} &= X + \frac{width}{2} \\
 X_{min} &= X - \frac{width}{2} \\
 Y_{max} &= Y + \frac{height}{2} \\
 Y_{min} &= Y - \frac{height}{2}
 \end{aligned} \tag{3.1}$$

From Equation 3.1, X is an X-value from the X-Y coordinate. Y is a Y-value from the X-Y coordinate. $width$ is a width of the red rectangle and $height$ is a height of the red rectangle (Figure 3.19). After we have those four parameters, we convert both X_{max} and X_{min} to $longitude_{max}$ (LNG_{max}) and $longitude_{min}$ (LNG_{min}), respectively, and Y_{max} and

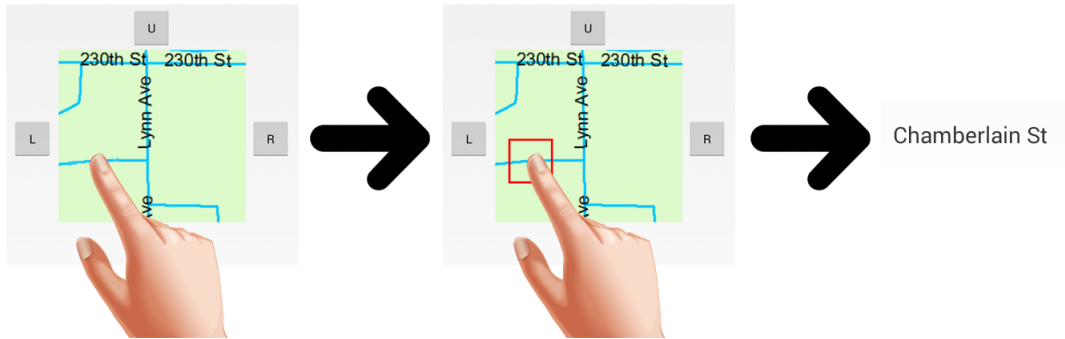


Figure 3.18 Street Object Detecting in Our Software (1).

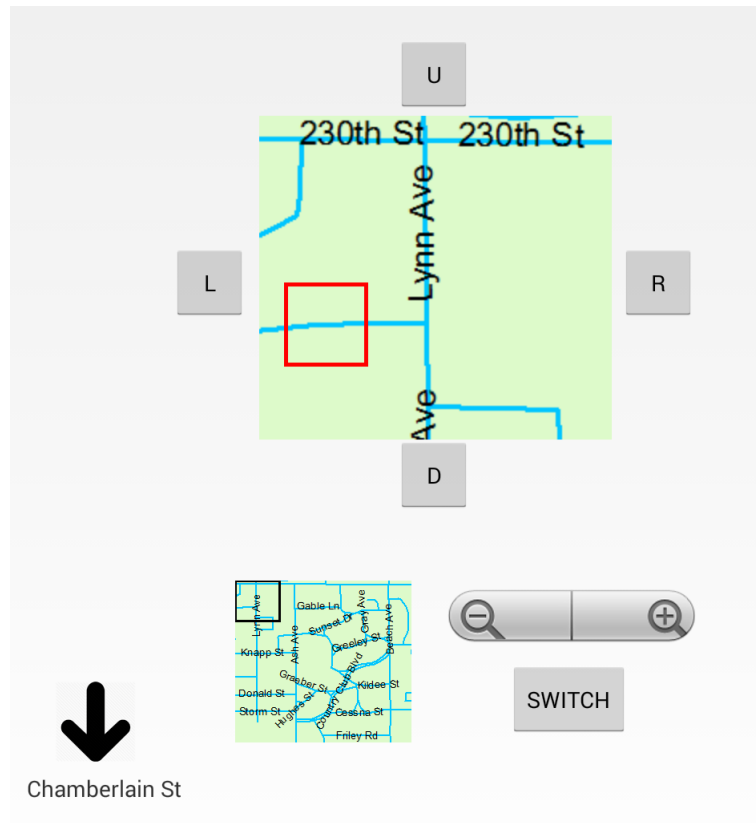


Figure 3.19 Street Object Detecting in Our Software (2).

Y_{min} to $latitude_{min}$ (LAT_{min}) and $latitude_{max}$ (LAT_{max}), respectively, using a world file. It is noted that the conversion of Y to latitude is a reverse variation due to the difference in vertical scaling between device screen and geographical system.

Next, it starts searching for streets that belong to this area bound by LNG_{max} , LNG_{min} , LAT_{min} , and LAT_{max} using Algorithm 1. According to Algorithm 1, it scans through every Street object. For each Street object, it starts checking the first coordinate of longitude and latitude. If a latitude falls between LAT_{min} and LAT_{max} and a longitude falls between LNG_{min} and LNG_{max} , then it adds that Street object to a collection of found streets and continues on next Street object. Otherwise, it goes to the next coordinate and performs the comparison again. If there is no coordinate that belongs to that area, it moves to the next Street object. When it scans through all Street objects, it returns a collection of found streets. Those found streets are displayed to users on the screen as early mentioned.

Algorithm 1 Algorithm for Detecting Streets.

```

StreetFound  $\leftarrow$  null
for all StreetObject  $S \in$  AllStreets do
  for all  $(LAT, LNG) \in S.LatLongCollection$  do
    if  $LAT_{min} \leq LAT \leq LAT_{max}$  then
      if  $LNG_{min} \leq LNG \leq LNG_{max}$  then
        StreetFound.Add( $S$ )
        break
      end if
    end if
  end for
end for
return StreetFound

```

3.3.2 Point

For Point object model, we would provide the conceptual implementation using the example set up in Subsection 3.2.2, which consists of a raster map and one Intersection object represented \mathbb{I} intersection (Figure 3.9 and 3.10).

The algorithms for the Point object are similar to Line object algorithms (Subsection 3.3.1) but they are less complicated. We similarly define a square region to cover that tapped location, where the tapped location is the center of this square region, as a region of interest (it is the red square in Figure 3.21). Then we calculate boundaries of this region of interest in term of latitudes and longitudes using the same world file method. Next, we iteratively visit each Intersection object. For each Intersection object, we check its pair of longitude and latitude whether it is located inside the region of interest or not. If it does, that Intersection object would be marked as an intersection that a user taps on. Figure 3.20 illustrates a flowchart that explains the whole process of implementation.

In our example, we have one Intersection objects, \mathbb{I} intersection (Figure 3.10). Therefore, when we apply our conceptual implementation, we start from Intersection object of \mathbb{I} intersection. \mathbb{I} intersection has one pair of longitude and latitude, (x, y) . We begin checking whether this pair (x, y) locates inside the region of interest or not. If it does, we are done with Intersection object of \mathbb{I} intersection. \mathbb{I} intersection would be counted as an intersection that a user taps on. When we complete all Intersection objects, names of all Intersection objects that are counted as tapped intersection would be displayed on the screen. Based on our example (Figure 3.9 and 3.10), \mathbb{I} intersection is displayed because a pair of (x, y) locates inside the region of interest corresponded to the tapped location. Figure 3.21 illustrates how we can obtain this result.

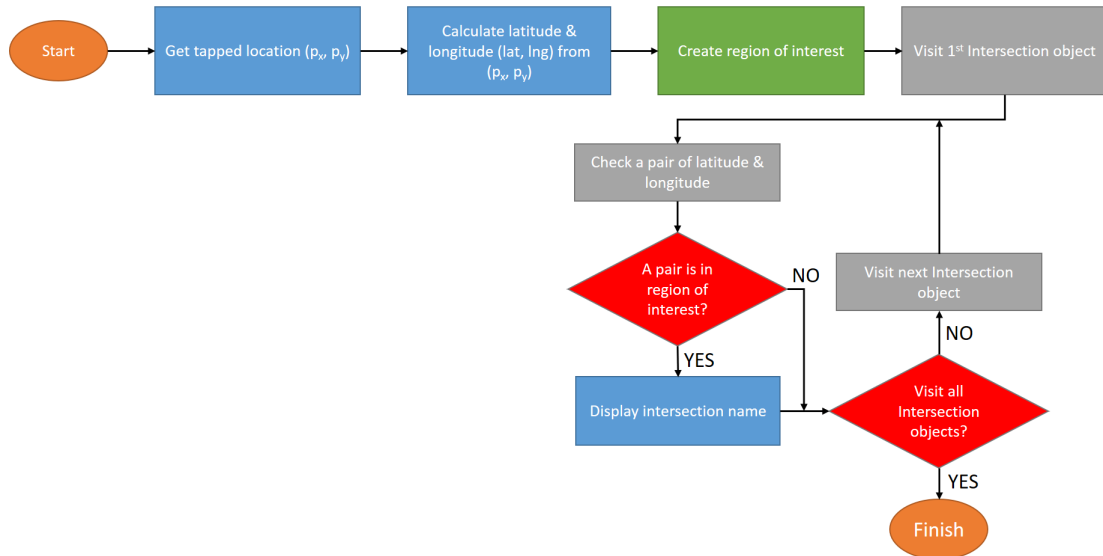


Figure 3.20 Flow Chart of Object-Indexing Implementation for Point Object Model.

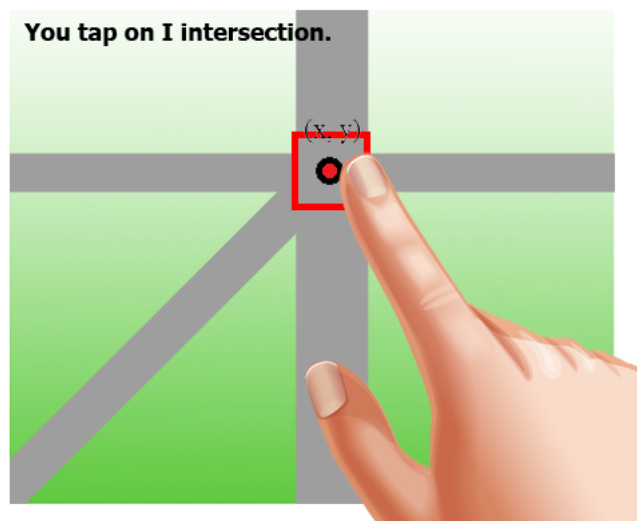


Figure 3.21 How Object-Indexing Operates for Point Object Model.

The complexity of our Object-Indexing for Point object model is $O(n)$ where n is the number of Intersection objects. Thus, this is a polynomial algorithm.

We explain the algorithm described in Figure 3.20. When a user taps on an intersection, we draw a red rectangle to cover that tapped area and return an intersection that it can detect to the user. A name of detected intersection is displayed at the bottom-left of the software screen.

The Object-Indexing starts working by getting an X-Y coordinate of a point that user has tapped. This is similar to Line algorithm 3.3.1. Next, it calculates a value of X_{max} , X_{min} , Y_{max} , and Y_{min} by the following formula:

$$\begin{aligned}
 X_{max} &= X + \frac{width}{2} \\
 X_{min} &= X - \frac{width}{2} \\
 Y_{max} &= Y + \frac{height}{2} \\
 Y_{min} &= Y - \frac{height}{2}
 \end{aligned} \tag{3.2}$$

For Equation 3.2, X is an X-value from the X-Y coordinate. Y is a Y-value from the X-Y coordinate. $width$ is a width of the red rectangle and $height$ is a height of the red rectangle (Figure 3.19). After we have those four parameters, we convert both X_{max} and X_{min} to *longitude_{max}* (LNG_{max}) and *longitude_{min}* (LNG_{min}), respectively, and Y_{max} and Y_{min} to *latitude_{min}* (LAT_{min}) and *latitude_{max}* (LAT_{max}), respectively, using a world file. It is noted that the conversion of Y to latitude is a reverse variation due to the difference in vertical scaling between device screen and geographical system.

Next, it starts searching for intersection that belong to this area bound by LNG_{max} , LNG_{min} , LAT_{min} , and LAT_{max} using Algorithm 2. According to Algorithm 2, it scans through every Intersection object. For each Intersection object, it starts checking its coordinate of longitude and latitude. If a latitude falls between LAT_{min} and LAT_{max} and a longitude falls between LNG_{min} and LNG_{max} , then it adds that Intersection object to a collection of found intersection and continues on next Intersection object. If its coordinate does not belongs to that area, it moves to the next Intersection object. When it scans through all Intersection objects, it returns a collection of found intersections. The names of the found intersections are displayed to users on the screen as mentioned earlier.

Algorithm 2 Algorithm for Detecting Intersections.

```

IntersectionFound ← null
for all IntersectionObject  $I \in AllIntersections$  do
  if  $LAT_{min} \leq I.LAT \leq LAT_{max}$  then
    if  $LNG_{min} \leq I.LNG \leq LNG_{max}$  then
      IntersectionFound.Add(I)
    end if
  end if
end for
return IntersectionFound

```

3.3.3 Area

For Area object model, we would provide the conceptual implementation using the example set up in Subsection 3.2.3, which consists of a raster map and one Wetland object represented W lake (Figure 3.13 and 3.14).

Dissimilar to Line and Point implementation (Subsection 3.3.1, 3.3.2), we do not define a region of interest for Area implementation. For Area implementation, we have an objective to identify what area (lake, pond, etc.) that the tapped location is inside. We start by getting the tapped location and calculating longitude and latitude of that tapped

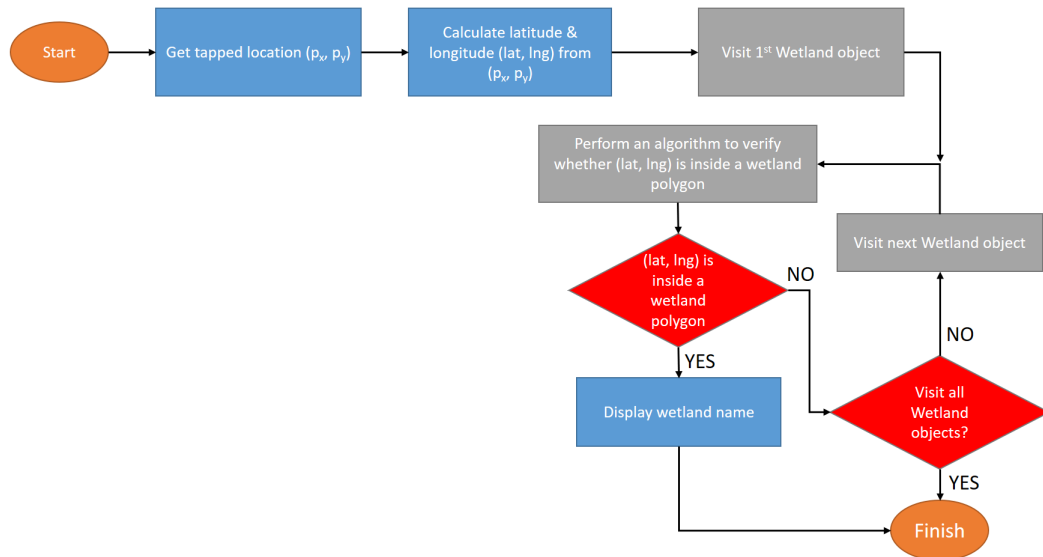


Figure 3.22 Flow Chart of Object-Indexing Implementation for Area Object Model.

location. Next, we iteratively visit each Wetland object. For each Wetland object, we verify whether the tapped location is inside that Wetland object or not. Every Area object model is represented by a polygon. Thus, this is exactly the same problem as “Point-in-polygon” problem, which is one of famous problems in the computational geometry [32]. To solve this, we can use an algorithm for determining if a point lies on the interior of a polygon invented by Paul Bourke [8]. If the verification result confirms that the tapped location is inside the wetland, that Wetland object is marked as a wetland that a user taps on. Otherwise, we move to the next Wetland object. Figure 3.22 illustrates a flowchart that explains the whole process of implementation.

In our example, we have one Wetland object, W lake (Figure 3.14). Therefore, when we apply our conceptual implementation, we have to start from a Wetland object of W lake. W lake consists of six pairs of longitude and latitude. These six pairs of longitude

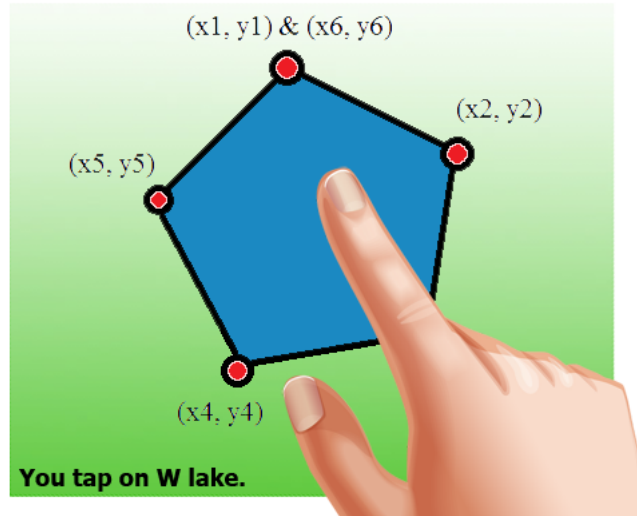


Figure 3.23 How Object-Indexing Operates for Area Object Model.

and latitude are vertices of a polygon, which is a pentagon for this example (Figure 3.14). Next, we use the Bourke's algorithm [8] to verify whether the tapped location (x, y) is inside this pentagon or not (we would not discuss this algorithm in detail). If the verification result shows that the tapped location (x, y) is inside a pentagon, the name of that Wetland object is displayed on the screen. Based on our example (Figure 3.13 and 3.14), W lake is displayed because the tapped location (x, y) locates inside the area of W lake. Figure 3.23 illustrates how we can obtain this result.

The complexity of our Object-Indexing for Area object model is $O(nm)$ where n is the maximum number of pairs of longitude and latitude and m is the number of Wetland objects. Thus, this is a polynomial algorithm.

We explain the algorithm described in Figure 3.22. When a user taps on a location that is inside a wetland, we return a wetland that the Object-Indexing can detect to the user. The name of the detected wetland is displayed at the bottom-left of the software screen.

The Object-Indexing starts working by getting an X-Y coordinate of a point that user has tapped. This is similar to Line implementation (Subsection 3.3.1).

X is an X-value from the X-Y coordinate. Y is a Y-value from the X-Y coordinate. Next, we calculate LAT and LNG from Y and X , respectively, using a world file. It is noted that the conversion of Y to latitude is a reverse variation due to the difference in vertical scaling between device screen and geographical system.

Next, it starts searching for a wetland that the tapped location (LAT, LNG) belongs to using Algorithm 3. According to Algorithm 3, it scans through every Wetland object. For each Wetland object, it starts checking whether the tapped location (LAT, LNG) is inside that wetland or not using Bourke’s algorithm [8]. If a result verifies that (LAT, LNG) is inside that wetland, then it adds that Wetland object to a collection of found wetlands. If its coordinate (LAT, LNG) does not belongs to that wetland, it moves to the next Wetland object. A found wetland is displayed to users on the screen as early mentioned.

Algorithm 3 Algorithm for Detecting Wetlands.

```

WetlandFound  $\leftarrow$  null
for all WetlandObject  $W \in$  AllWetlands do
    verificationResult  $\leftarrow$  Bourke( $W, LAT, LNG$ )
    if verificationResult = TRUE then
        WetlandFound.Add( $W$ )
        break
    end if
end for
return WetlandFound

```

3.4 Using Object-Indexing in Study Software

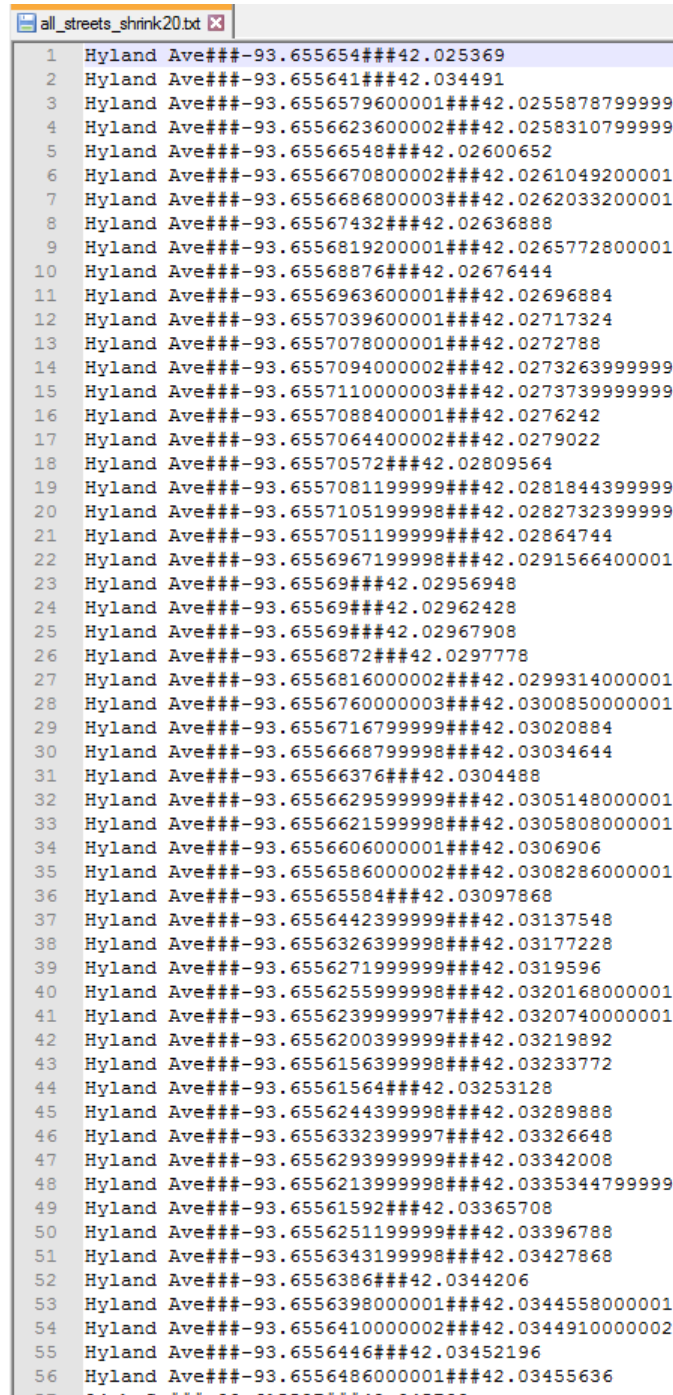
We deployed the algorithm discussed in Section 3.3 to Android application using Java language because we would like the map integrated with Object-Indexing to be the primary map for location-based software on portable device for address verification purpose.

The Android version consists of multiple modules. In this section, we discuss only the object-indexing handling module.

Object-Indexing Handling module has a responsibility to handle all Street objects since the software starts and until users quit. The Object-Indexing Handling module consists of three submodules, which are Street Records Reading, Street Objects Initiating, and Street Objects Detecting. A description and function of each submodule is described in the following paragraphs.

The Street Records Reading submodule is in charge for reading street records from a streets text file and gathering all street properties. A streets text file that contains all street records was interpreted from an actual shapefile. Each line of streets text file consists of three values. The first one represents a street name. The second value represents a longitude and the third value represents a latitude. A “###” is a separating symbol. Because one street can have more than one latitude-longitude coordinate, one street can consume more than one line in the streets text file. Figure 3.24 shows that only Hyland Avenue itself consumes 56 lines in the streets text file because there are 56 latitude-longitude coordinates of Hyland Avenue. If there exists an actual driveway but no name has been assigned to it yet, the first value would be blank but we would not omit it.

After the reading process is completely done, then the Street Objects Initiating submodule is in charge. According to Figure 3.3, it takes all streets records to instantiate



The image shows a screenshot of a text editor window titled "all_streets_shrink20.txt". The window contains a list of 56 lines of text, each representing a street entry. Each entry consists of a line number (from 1 to 56), the street name "Hyland Ave###", a long alphanumeric string, and a numerical value. The alphanumeric strings appear to be a combination of a street ID and a coordinate. The numerical values are also listed on the right side of each line.

```

1 Hyland Ave###-93.655654###42.025369
2 Hyland Ave###-93.655641###42.034491
3 Hyland Ave###-93.6556579600001###42.0255878799999
4 Hyland Ave###-93.6556623600002###42.0258310799999
5 Hyland Ave###-93.65566548###42.02600652
6 Hyland Ave###-93.6556670800002###42.0261049200001
7 Hyland Ave###-93.6556686800003###42.0262033200001
8 Hyland Ave###-93.65567432###42.02636888
9 Hyland Ave###-93.6556819200001###42.0265772800001
10 Hyland Ave###-93.65568876###42.02676444
11 Hyland Ave###-93.6556963600001###42.02696884
12 Hyland Ave###-93.6557039600001###42.02717324
13 Hyland Ave###-93.6557078000001###42.0272788
14 Hyland Ave###-93.6557094000002###42.0273263999999
15 Hyland Ave###-93.6557110000003###42.0273739999999
16 Hyland Ave###-93.6557088400001###42.0276242
17 Hyland Ave###-93.6557064400002###42.0279022
18 Hyland Ave###-93.65570572###42.02809564
19 Hyland Ave###-93.6557081199999###42.0281844399999
20 Hyland Ave###-93.6557105199998###42.0282732399999
21 Hyland Ave###-93.6557051199999###42.02864744
22 Hyland Ave###-93.6556967199998###42.0291566400001
23 Hyland Ave###-93.65569###42.02956948
24 Hyland Ave###-93.65569###42.02962428
25 Hyland Ave###-93.65569###42.02967908
26 Hyland Ave###-93.6556872###42.0297778
27 Hyland Ave###-93.6556816000002###42.0299314000001
28 Hyland Ave###-93.6556760000003###42.0300850000001
29 Hyland Ave###-93.6556716799999###42.03020884
30 Hyland Ave###-93.6556668799998###42.03034644
31 Hyland Ave###-93.65566376###42.0304488
32 Hyland Ave###-93.6556629599999###42.0305148000001
33 Hyland Ave###-93.6556621599998###42.0305808000001
34 Hyland Ave###-93.6556606000001###42.0306906
35 Hyland Ave###-93.6556586000002###42.0308286000001
36 Hyland Ave###-93.65565584###42.03097868
37 Hyland Ave###-93.6556442399999###42.03137548
38 Hyland Ave###-93.6556326399998###42.03177228
39 Hyland Ave###-93.6556271999999###42.0319596
40 Hyland Ave###-93.6556255999998###42.0320168000001
41 Hyland Ave###-93.6556239999997###42.0320740000001
42 Hyland Ave###-93.6556200399999###42.03219892
43 Hyland Ave###-93.6556156399998###42.03233772
44 Hyland Ave###-93.65561564###42.03253128
45 Hyland Ave###-93.6556244399998###42.03289888
46 Hyland Ave###-93.6556332399997###42.03326648
47 Hyland Ave###-93.6556293999999###42.03342008
48 Hyland Ave###-93.6556213999998###42.0335344799999
49 Hyland Ave###-93.65561592###42.03365708
50 Hyland Ave###-93.6556251199999###42.03396788
51 Hyland Ave###-93.6556343199998###42.03427868
52 Hyland Ave###-93.6556386###42.0344206
53 Hyland Ave###-93.6556398000001###42.0344558000001
54 Hyland Ave###-93.6556410000002###42.0344910000002
55 Hyland Ave###-93.6556446###42.03452196
56 Hyland Ave###-93.6556486000001###42.03455636

```

Figure 3.24 Example of Streets Text File.

Street objects. A Street object is instantiated using a Street class. This class consists of two fields, which are street name and a collection of latitude-longitude coordinates of that street. We used “java.util.ArrayList” to store that collection. Every street that belongs to the neighborhood would have its own corresponding Street object. Every Street object would last until users quit the application software.

The last submodule, Street Object Detecting, is significantly important for our study. This submodule does a deployment of design and implementation concept of Object-Indexing in Section 3.2 and 3.3. This submodule enhances a usability of our raster map software by allowing users to tap on any street, at any zoom level and any map type, in the primary map where they would like to know the name of that such street. Particularly, Figure 3.18 shows how this module works.

In our study software, we provide two types of raster map, which are a Tiger Line map [9] and a Google map. For a Tiger Line map, Figure 3.25 shows a series of screens when a user taps on an intersection of Knapp Street and Lynn Avenue in a Tiger Line map. Similarly, Figure 3.26 shows a series of screens when a user taps on the same intersection in a Google map. For both maps, names of streets in the intersection, Knapp Street and Lynn Avenue, are displayed at the bottom-left of the screen.

The advantage of having Object-Indexing can be obviously observed in Figure 3.27, 3.28, and 3.29 in which a user taps on a street without street name. Figure 3.27 shows a scenario that a user taps on an unknown street. With Object-Indexing, a name of Gray Avenue is displayed at the bottom-left of the screen to identify that unknown street. The similar scenarios are in Figure 3.28 and 3.29. For Figure 3.28, an unknown street is S 4th Street. For Figure 3.29, an unknown street is Agg Avenue. All three scenarios are occurred on a Tiger Line map.

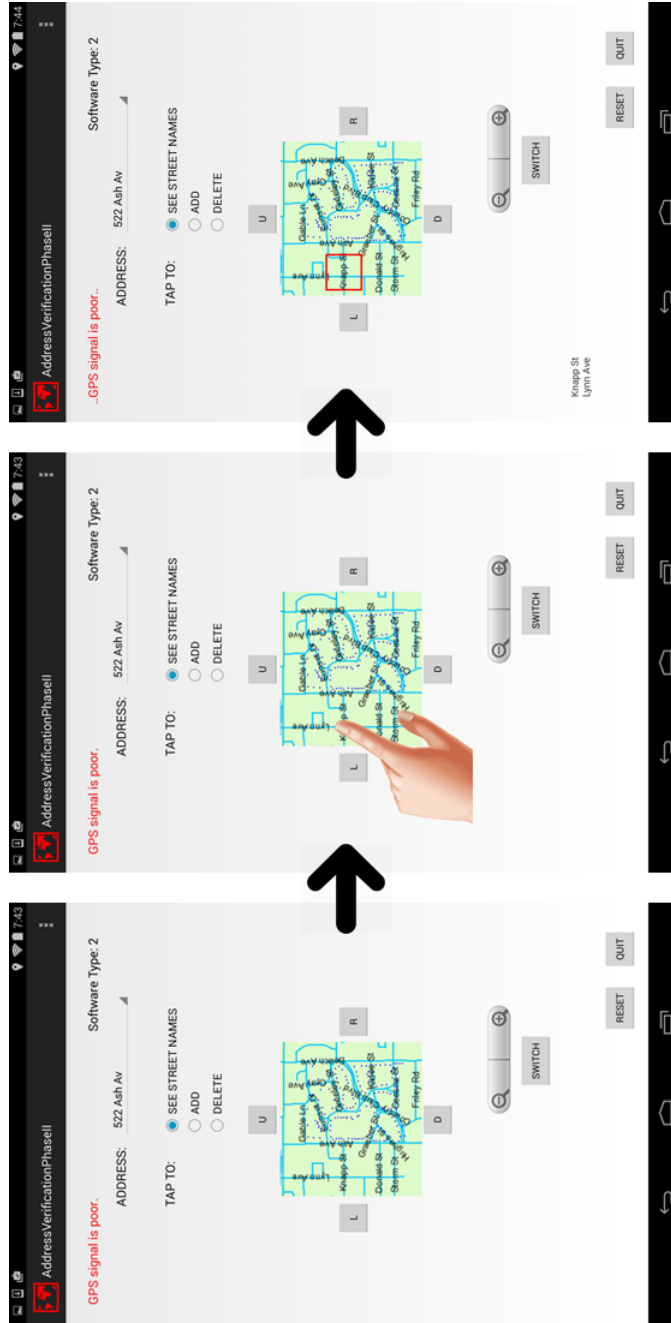


Figure 3.25 A User Taps on an Intersection in Tiger Map.

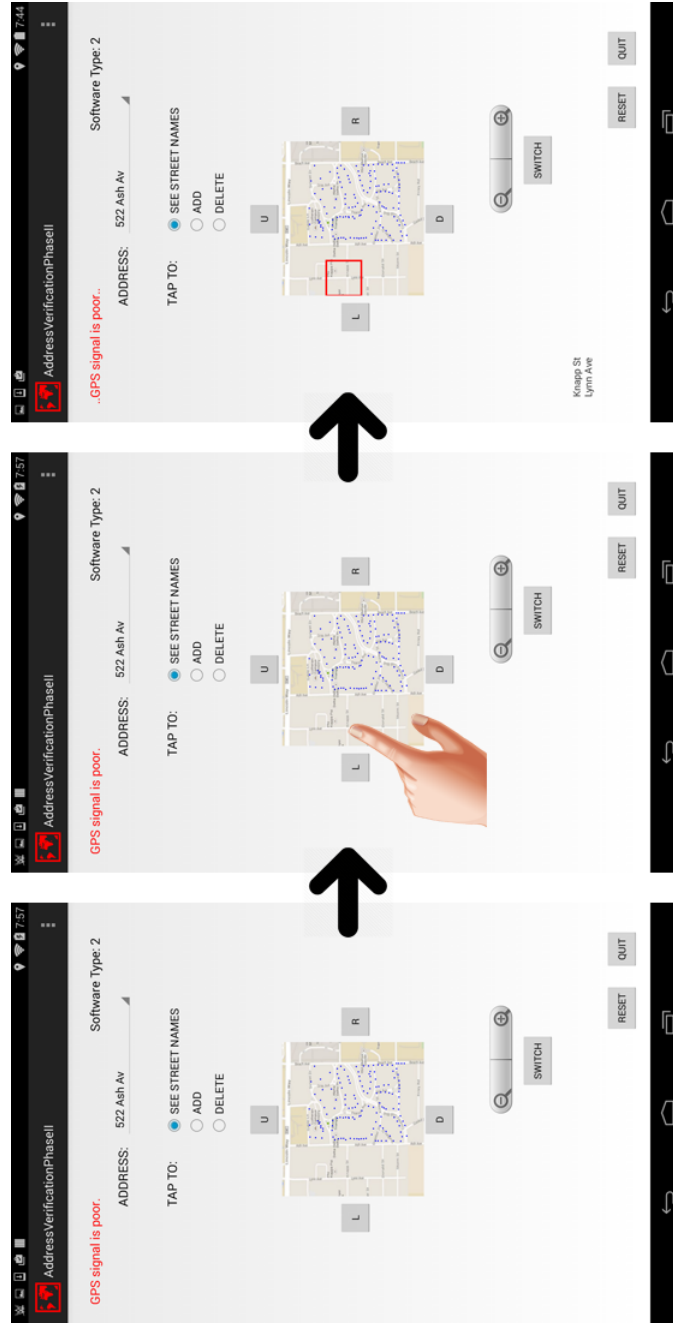


Figure 3.26 A User Taps on an Intersection in Google Map.

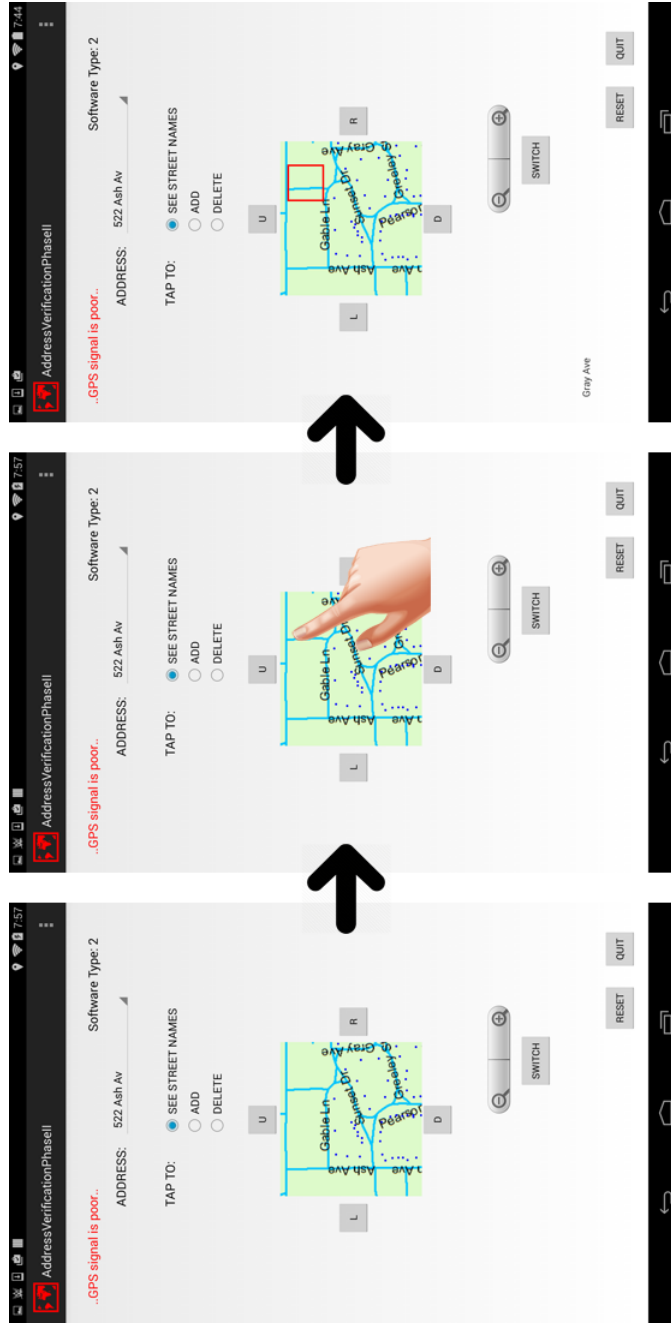


Figure 3.27 A User Taps on Gray Avenue on Tiger Line Map.

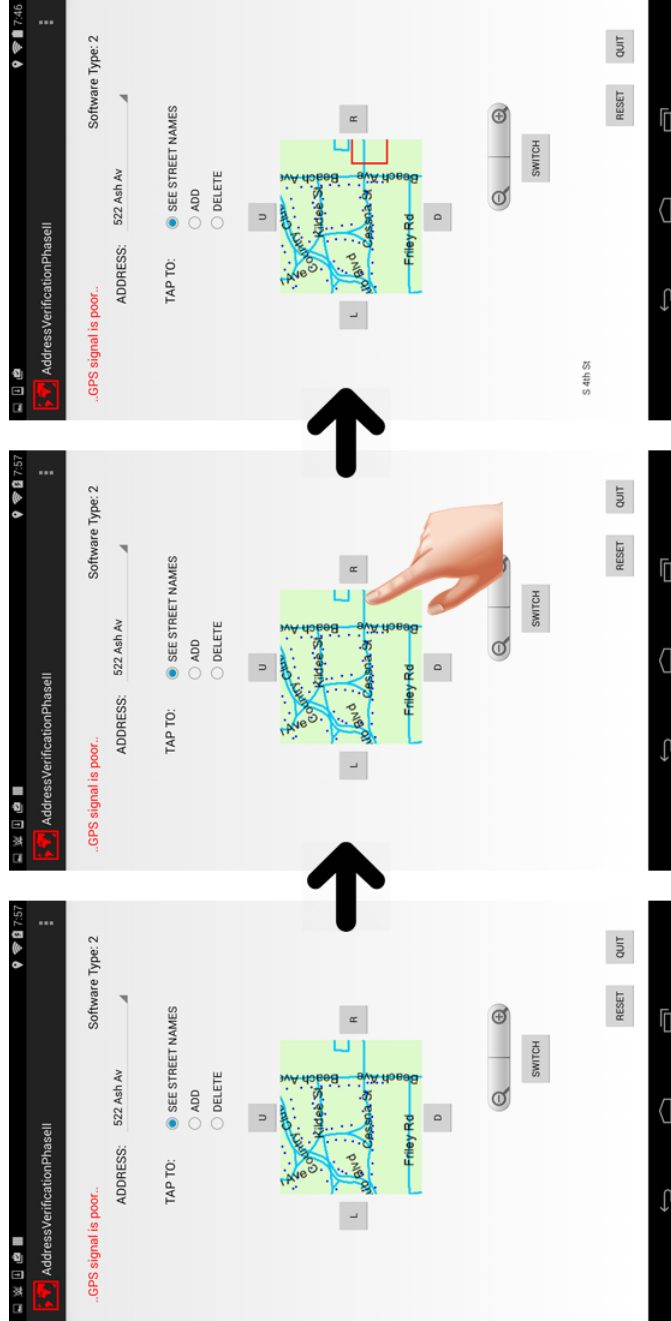


Figure 3.28 A User Taps on S 4th Street on Tiger Line Map.

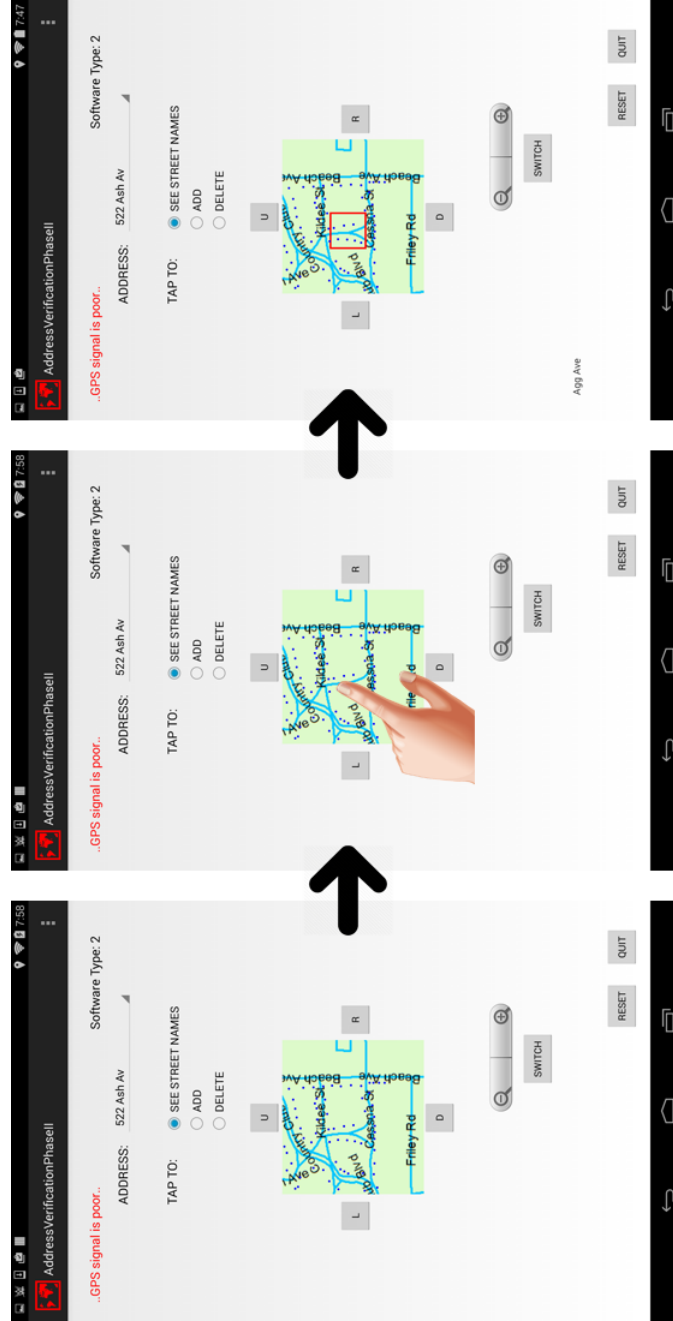


Figure 3.29 A User Taps on Agg Avenue on Tiger Line Map.

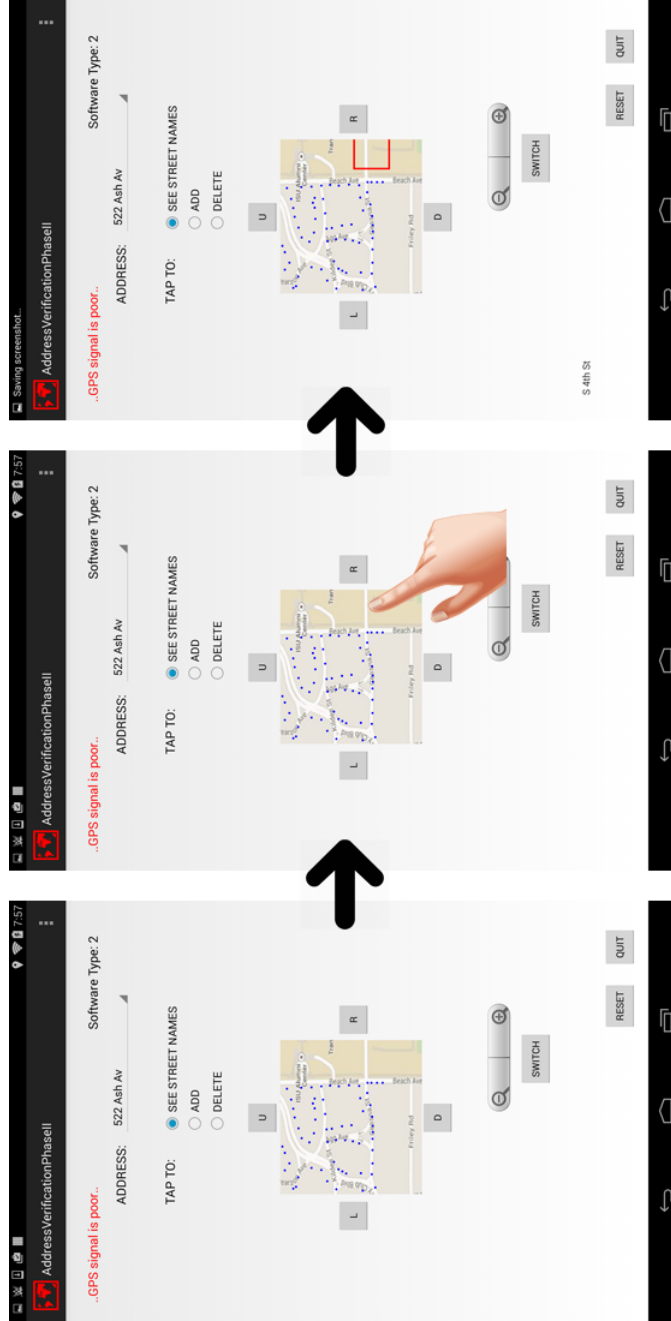


Figure 3.30 A User Taps on S 4th Street on Google Map.

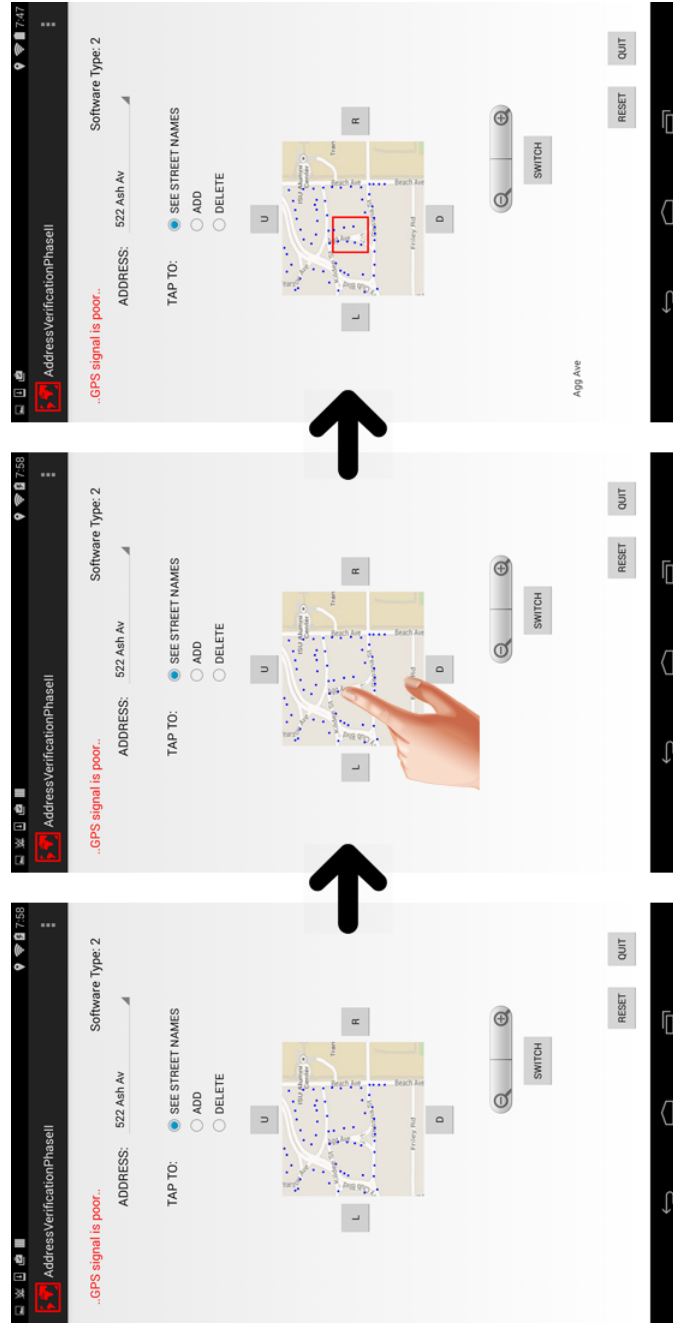


Figure 3.31 A User Taps on Agg Avenue on Google Map.

We also shows the same scenarios as Figure 3.28 and 3.29 in Figure 3.30 and 3.31, respectively. The difference between both groups is map type. Figure 3.30 and 3.31 show the scenarios occurred in a Google map.

The last scenarios are in Figure 3.32 and 3.33. In Figure 3.32, a user taps on a 3-way intersection. The names of all streets in that 3-way intersection are displayed, which are Friley Road and Gaskill Drive. In Figure 3.33, a user taps on a 4-way intersection. The names of all streets in that 4-way intersection are displayed, which are Country Club Boulevard, Pearson Avenue, and Kildee Street. These two scenarios are occurred on a Tiger Line map.

We also shows the same scenarios as Figure 3.32 and 3.33 in Figure 3.34 and 3.35, respectively. The difference between both groups is map type. Figure 3.34 and 3.35 show the scenarios occurred in a Google map.

In this chapter, we discuss about Object-Indexing, which is a new concept of combining a traditional raster map with objects generated from shapefiles to represent the primary map components to grant accessibility to these components to users of a map software. General concept, Design, algorithms, and implementation of Object-Indexing are also covered. In the next chapter, we would discuss about our map software, experiment (address verification), and analysis of experiment result.

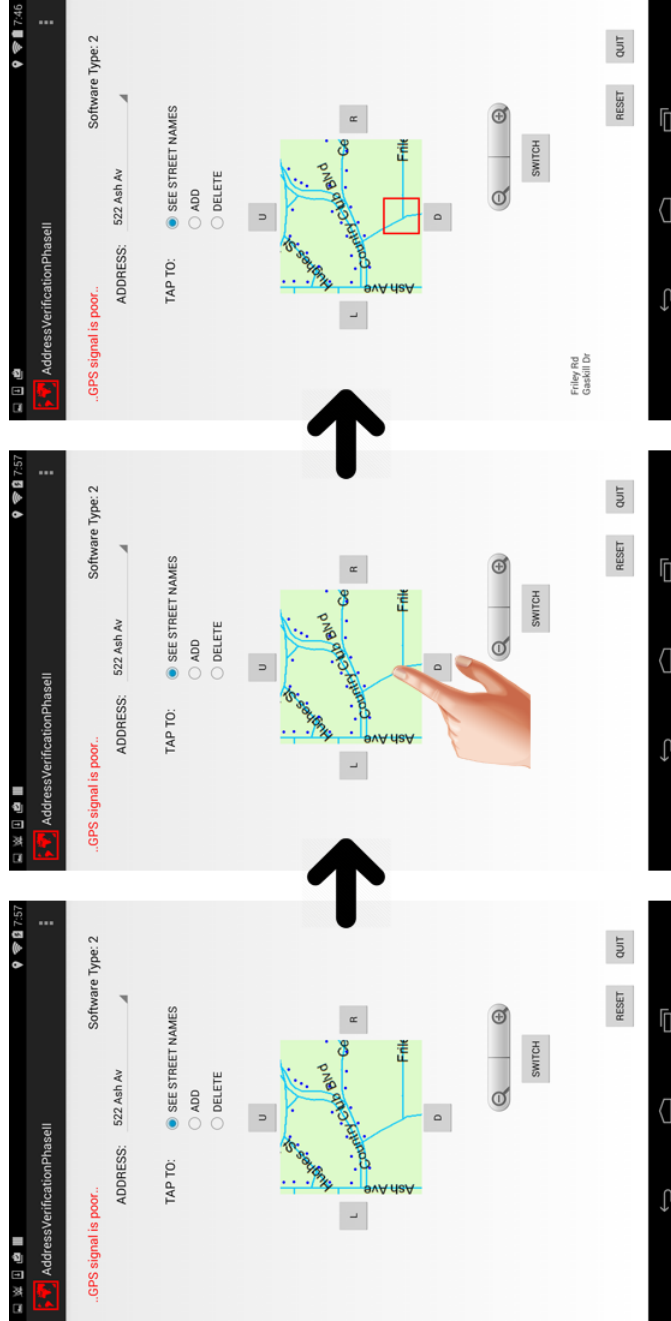


Figure 3.32 A User Taps on 3-way Intersection on Tiger Line Map.

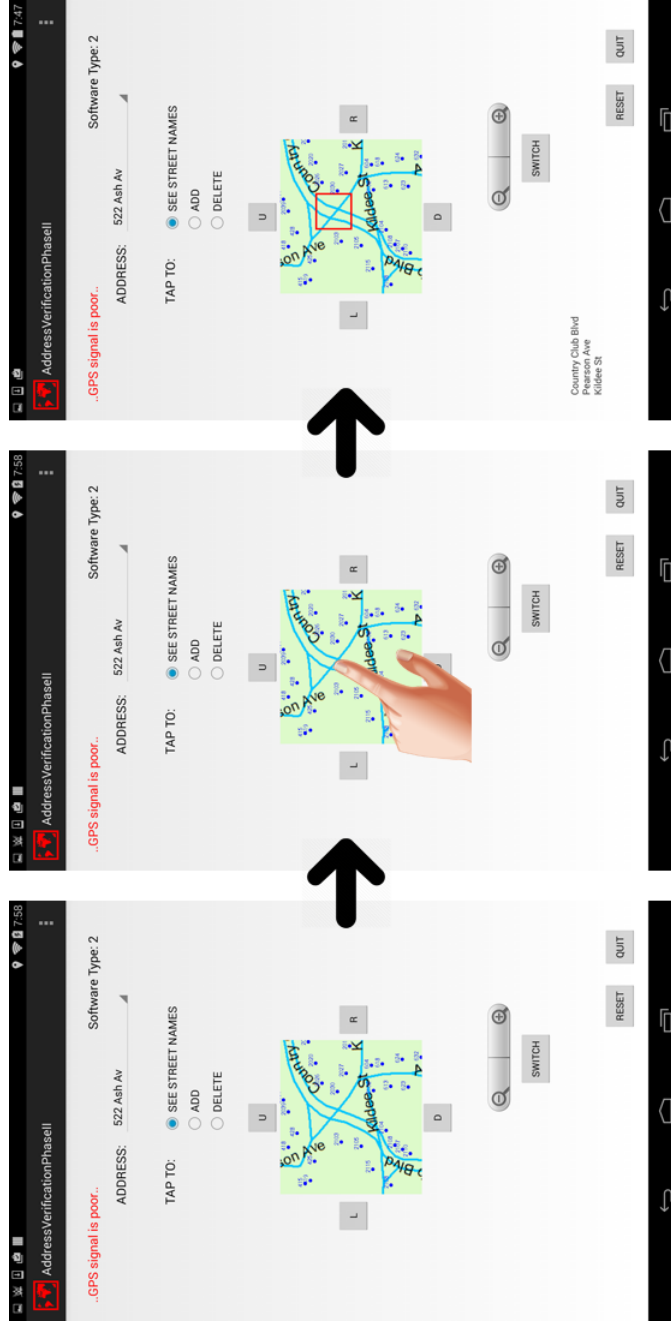


Figure 3.33 A User Taps on 4-way Intersection on Tiger Line Map.

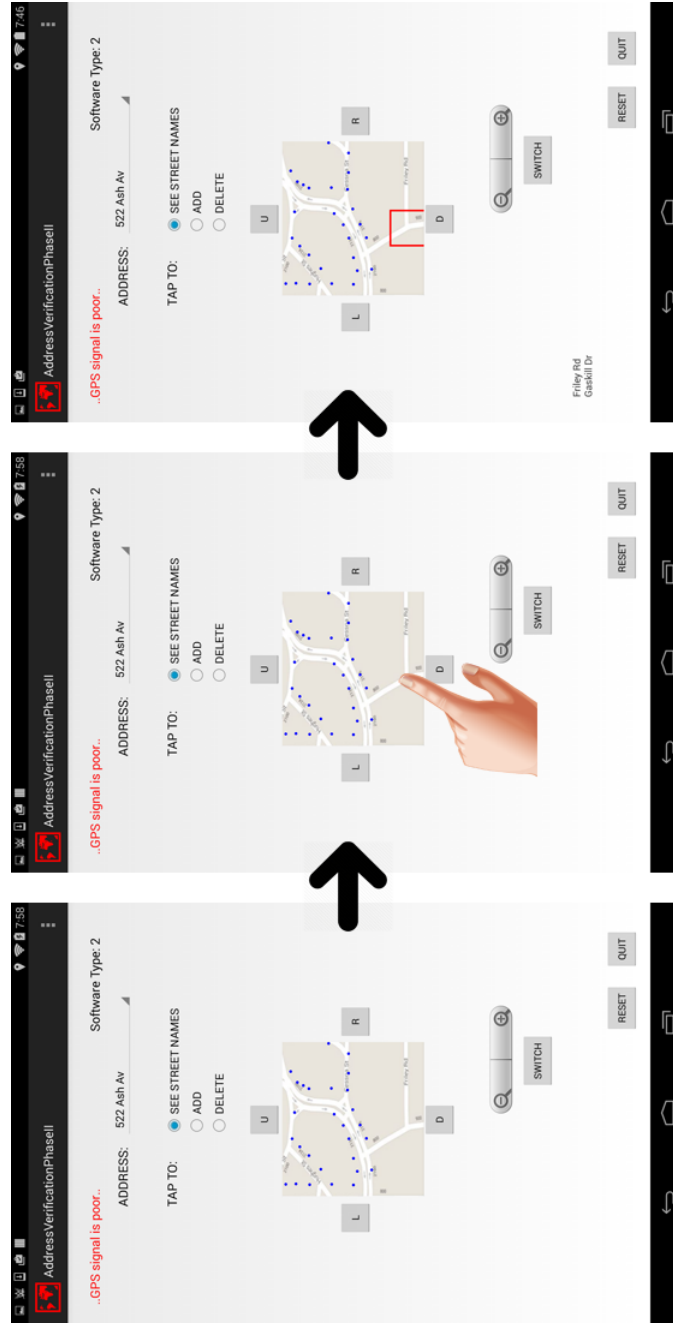


Figure 3.34 A User Taps on 3-way Intersection on Google Map.

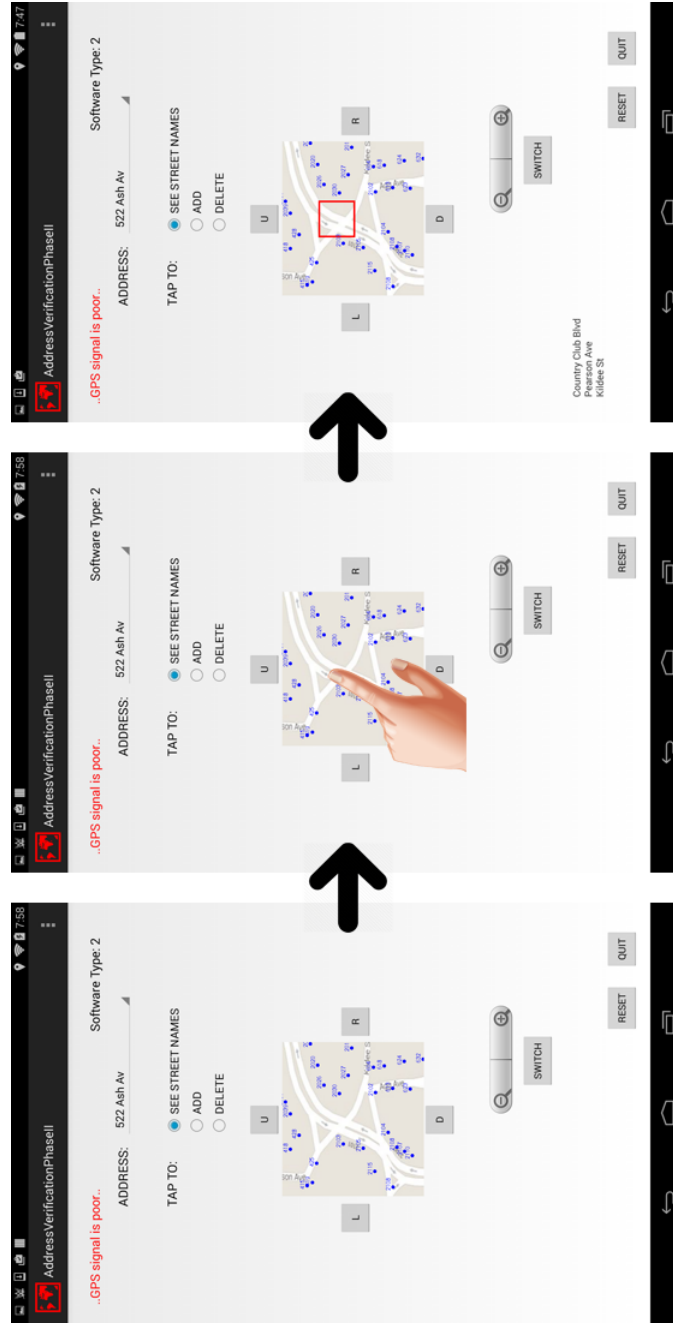


Figure 3.35 A User Taps on 4-way Intersection on Google Map.

CHAPTER 4. STUDY I: A STUDY TO INVESTIGATE PARTICIPANTS' PERFORMANCE WITH DIFFERENT USER INTERFACES

This chapter described a procedure and results from our experiment of address verification on an Android tablet. Spatial visualization test scores, address verification scores, and statistical outcomes from the study are discussed. Study I was approved by the Institutional Review Board (IRB-ID: 15-494). The IRB approval document was provided in [Appendix A](#).

As we proposed in [Chapter 1](#), the purpose of this research was to flatten the difference between low and high spatial visualization people performing the address verification task on a location-based software. The focus of the study was to test three user interface features. In particular, we were interested in understanding the relationship between user performance and the user's level of spatial visualization. The three features are described in [Subsection 4.2.2](#). Each study treatment was setup by activating two of the three user interface features and disabling one of the features. Participants were randomly assigned one of the three treatments. In [Section 4.3](#), we show raw results from this study. The metrics that we were interested in are the participant's address verification score, the total time that each participant used, the total distance that each participants walked, the total number of errors, the total number of pans, and the total number of zooms. All of these interesting metrics and how they reflected performance are discussed in [Section 4.3](#).

4.1 Study Procedure

The study consisted of two parts. The first part was a test of the participant's level of spatial visualization. The second part was the field study. We invited participants to participate in our study by announcement via flyers and in-person communication. Thirty-two participants took part. The study started by asking participants to read and sign a letter of informed consent. Next, we asked participants to take a spatial visualization test and questionnaire about their general information, such as the frequency of map usage. After that, we moved to the neighborhood (Figure 4.1) and started the address verification task using our software on an Android tablet. When a participant completed the task, they received a \$20 gift card as a compensation. The Android tablet that we selected for this study was a Google Nexus 7. We selected this model because it has built-in GPS.

4.1.1 Test of Spatial Visualization

For testing spatial visualization, we used a Paper Folding test (the official name is VZ-2 test), which could be found in Ekstrom's Manual for Kit of Factor-Referenced Cognitive Tests [20]. The Paper Folding test we used was a multiple choice test that consisted of two problem sets. One problem set had ten problems and allowed three minutes to do all ten. Every problem asked the same question, which was: it illustrated a couple of steps to fold a piece of square paper, then it specified a point to punch a hole on the folded paper. Finally, it asked the test taker to select what the paper looked like when it was unfolded. Figure 4.2 showed an example of the Paper Folding test, which had two columns. The left column showed how the paper was folded and punched. The right columns contained five answer choices. A test taker had to mentally fold and punch a paper, then selected the output from five choices. One point was given for one correct answer. A negative one point was given for one incorrect answer. Zero point was given for no answer. Thus, the possible maximum score was twenty and the possible minimum score was negative twenty.

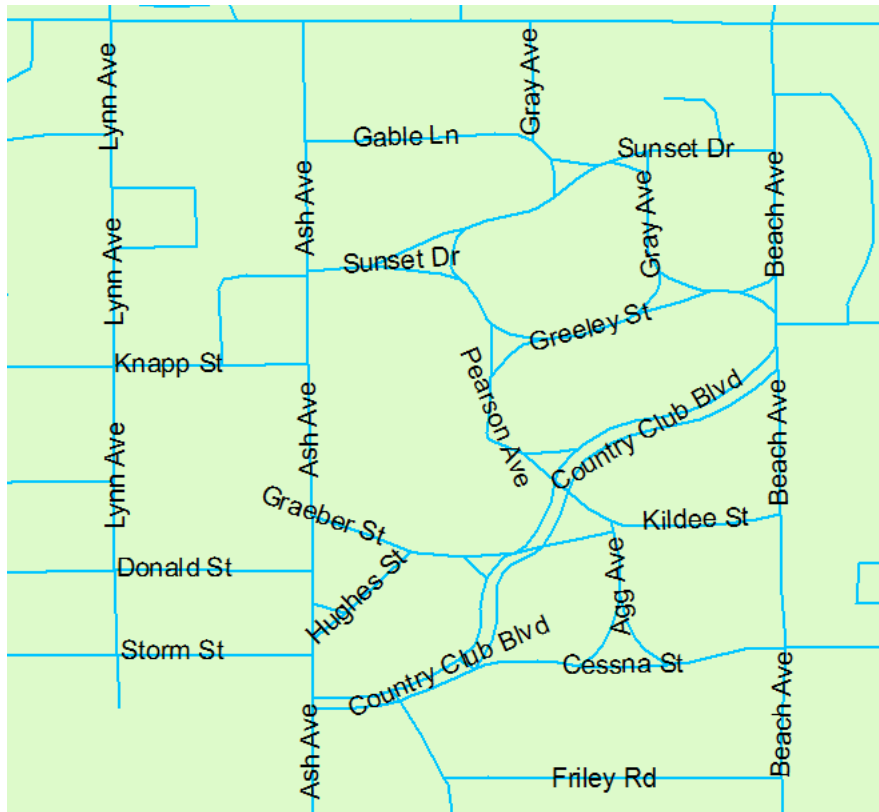


Figure 4.1 Map of Selected Neighborhood in Ames, Iowa.

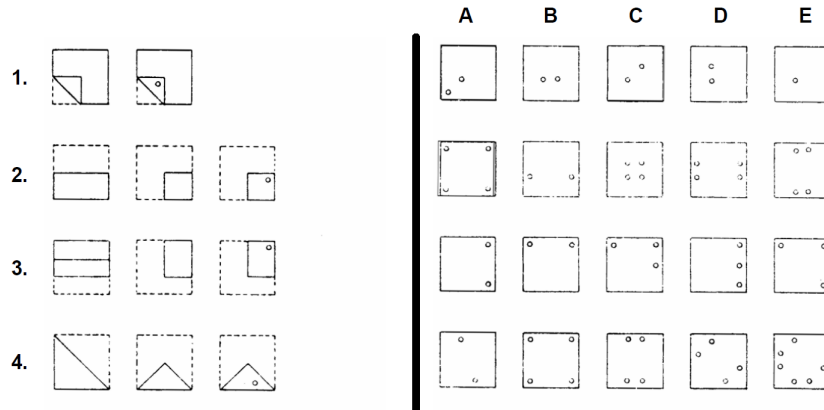


Figure 4.2 Paper Folding Test [20].

The high spatial visualization group was a group where every member had a test score greater than or equal to 13. The low spatial visualization group was a group where every member had a test score less than or equal to 8. These thresholds were from a discussion with Maria Kozhevnikov, a psychology professional, when she gave a presentation at Iowa State University. These thresholds were also used in previous work [2][89]. We graded the test after finishing the field study.

4.1.2 Field Portion of the Study

When participants finished the spatial visualization test, we moved to the selected neighborhood to start the address verification task using our software on the Android tablet. We selected this neighborhood because its layout was complicated. Before participants started the verification process, they were given instruction of how to use the software.

Before we discussed the address verification procedure, we gave a brief explanation of possible solutions for address verification. For verification of each address, there were four possible options and solutions described in Table 4.1.

Table 4.1 Address Verification Possibilities.

Options	Solutions
There existed an address in the neighborhood and there also existed an address spot in the map.	Participants needed to do nothing.
There existed an address in the neighborhood but there was no address spot in the map.	Participants needed to add that address spot on the map.
There was no address in the neighborhood but there existed an address spot in the map.	Participants needed to delete that address spot on the map.
There existed an address in the neighborhood and there also existed an address spot in the map but that address spot was located on opposite site of a street	Participants needed to delete that address spot on the map and add a new address spot on the map at the correct side of a street.

After participants understood all four potential options for each address, we gave instruction on the address verification procedure as a suggestion to participants. Participants may do or may not do according to our procedure based on their preference. The address verification procedure that we suggested to participants was given as follows:

1. Participants select one address that they would like to verify from the Address Drop-down List.
2. Participants should find their current location on the primary map.
3. Participants should search for the street on the map where the address was located.
4. Participants start walking from their current location to that street.

5. When participants arrived at that street, participants searched for that address spot on the primary map.
 - If the address spot existed on the primary map, participants should walk to that address based on the primary map.
 - If that address really existed on that street, this was the first verification option. Participants did nothing.
 - Else, if that address really existed, but was physically located somewhere else, this was the fourth verification option. Participants deleted the wrong address spot on the primary map and added a new address spot at the correct location.
 - Otherwise, that address did not exist, this was the third verification option. Participants deleted the address spot from the primary map.
 - Otherwise, if there was no address spot on the primary map, participants should walk along that street to verify whether that address actually existed or not.
 - If that address existed, this was the second verification option. Participants added a new address spot for that address on the primary map.
 - Else, participants did nothing.
6. Participants chose the next address until they completed all ten addresses.

We observed participants all the time. When participants completed all ten addresses, they returned the tablet and were given a \$20 gift card as compensation.

4.2 Study Software

We take a brief look at our location-based software used in the study. The first part is about the software interface components. The last part was about the software features

that we added to the software to see how effective they were with participants of different level of spatial visualization.

4.2.1 Software Interface Components

Our software for the address verification task was the only tool that our participants used when they were in the neighborhood. Figure 4.3 shows a screen shot that shows where all of the components were located.

The Address Drop-down List is a drop-down list that contains all ten addresses that participants needed to verify. The Operation Radio Buttons were used for participants to perform any verification operation on the primary map. The Primary Map is a raster map of the neighborhood area. Participants would use the primary map to perform address verification. The Pan Buttons were for participants to pan the primary map in four directions: up, down, left, and right. The Zoom Control was for participants to zoom in or zoom out the primary map. The Switch Button was for participants to switch a map type (Tiger Line Map and Google-like Map). If the current primary map was a Google-like map, it would switch to a Tiger Line map, and vice versa. The Reset Button was for participants to restore a software to the starting state. Finally, the Quit Button was for participants to quit the software after they completed the final address verification task. After the Quit Button was hit, the software would generate an activity log file, an address verification result, and a location log file.

4.2.2 Software Features

There were three user interface features that were in the software to see whether they helped participants with different spatial visualization or not. These three features were

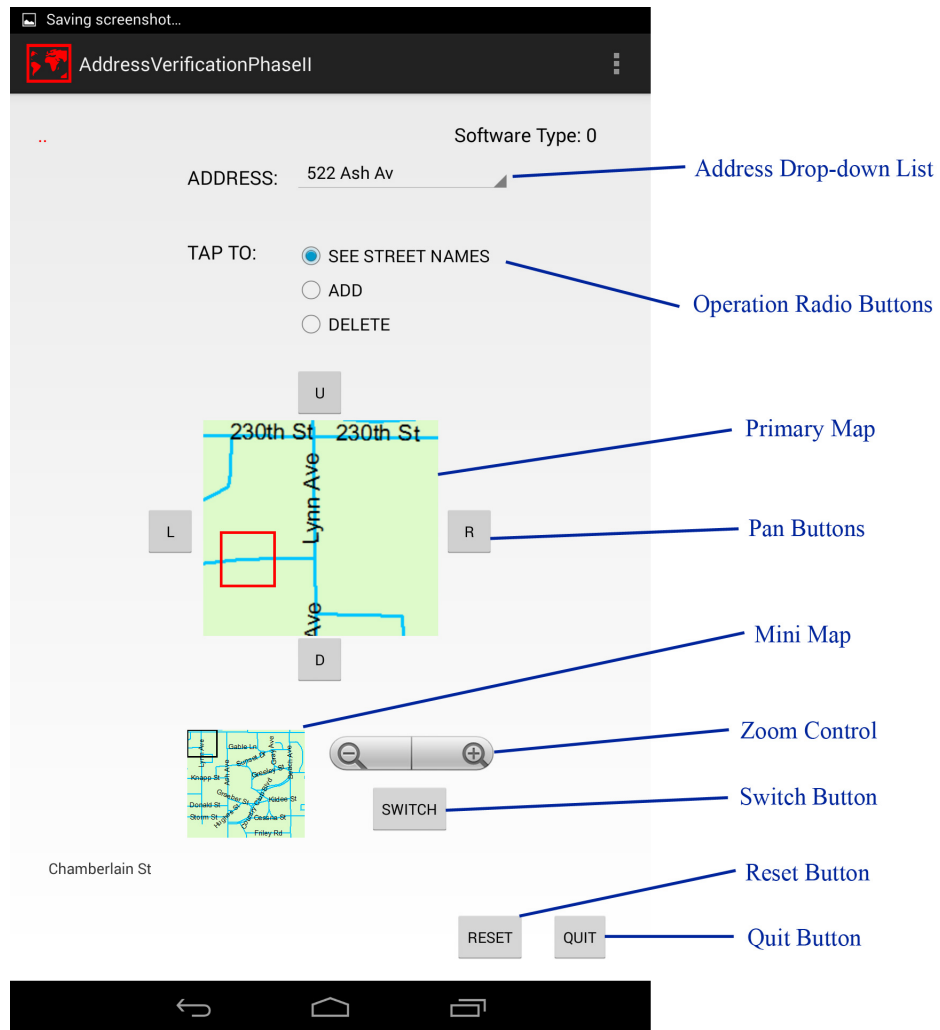


Figure 4.3 Software Screen.

Global Positioning System (GPS), Object-Indexing, and Mini Map. For each treatment that participant received, one of these features would be randomly disabled. Thus, for each participant, only two features were available for them to use during their address verification task.

1. Object-Indexing

We deployed the Object-Indexing approach that we discussed in Chapter 3 in this feature. For the implementation in the study software, we implemented the Street object because street was a primary map component in the address verification. With this feature, it allowed participants to tap on any street in the map to see a street name or any intersection to see street names of all streets intersected. We have already shown how Object-Indexing actually worked in our software in Section 3.4 in Chapter 3.

2. Mini Map

Based on the study of Buring, Gerken, and Reiterer [10], we knew that an overview window was an essential interface component for low spatial people. For our location-based software, the concept of overview window was implemented by the Mini Map. The Mini Map was also a map that was located below the primary map at the left side of the screen (Figure 4.3). The size of Mini Map was smaller than the primary map. The Mini Map showed the map of the full neighborhood. The difference between the primary map and the Mini Map was that the primary map would be changed to a new map image when participants panned or zoomed the map. Whereas the Mini Map stayed the same, which always showed the whole map of the neighborhood. Figure 4.4 showed how the Mini Map functioned as an overview window for our software. Figure 4.5 showed the Mini Map in the software interface. In Figure 4.5, the black rectangle represented the area that was zoomed (by a participant). This area was shown to a

participant in the primary map. With this feature, participants could see two maps at a single time where the primary map showed the area that they are focusing and the Mini Map showed the overview of the neighborhood.

3. Global Positioning System (GPS)

The Global Positioning System allowed participants to see their current location on the map. This feature was refreshed every two seconds when possible so participants could obtain their current location in real-time. The current location on the map was represented by a red square as shown in Figure 4.6.

4.3 Raw Results from Study

This section explains the raw results that we gathered from the thirty-two participants. Due to the protocol of IRB, we could not record the names or any identifiable data of participants. For each participant, there were six metrics that we focused on. Those six metrics were an address verification score, a total time each participant used in address verification, a total distance each participant walked during address verification, a number of error taps the participant did during address verification, a number of pan taps the participant did during address verification, and a number of zoom taps the participant did during address verification. We gathered these data and calculated the metrics from a log file that was generated by our software when a participant finished the task.

The address verification score was the score that showed how many addresses a participant correctly verified. The top score of the address verification score was 10 because there were ten addresses that the participant needed to verify in the field study. There were some addresses that required participants to move the map spot to the correct location on the map by deleting the misplaced spot and adding the new spot to the correct location. For

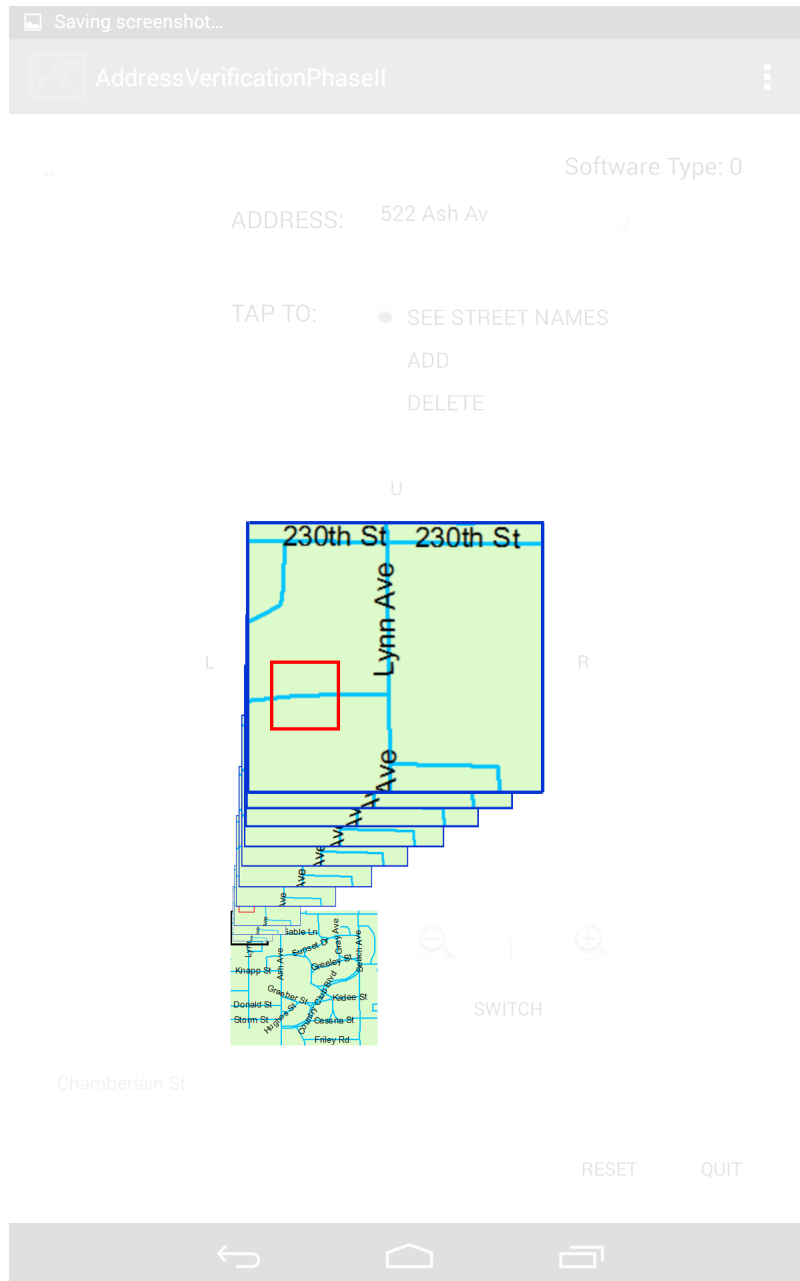


Figure 4.4 Relationship between Primary Map and Mini Map.

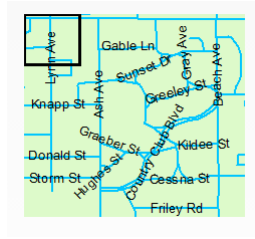


Figure 4.5 Mini Map in the Software Interface.

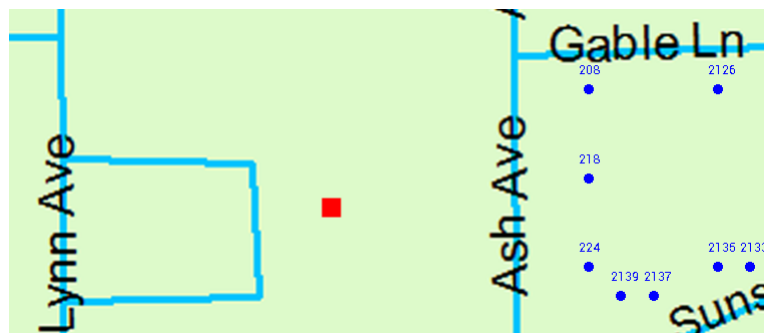


Figure 4.6 Current Location Displayed on Map.

this case, if participants partially completed the task by only deleting the wrong spot but not adding the correct spot, or another way around, they would get a score of 0.5 instead of 1. For the total time that the participant used in address verification, we started recording when the tablet was given to the participant and stopped when the participant tapped on Quit button (Section 4.2.1). The basic unit of the total time was minutes.

For the total distance that the participant walked during the address verification task, we calculated it by using the time stamp captured in the log file of that participant. When a participant started the task, our software kept track of the participant's location using a set of longitude and latitude coordinates. Thus, the total route of the participant could be represented by the sequence of longitude-latitude coordinates. The distance was calculated in kilometers along the path defined by the coordinates.

The number of error taps was a number that the participant tapped outside the sensitive regions. We defined, based on the work of Taylor [101], that the sensitive region was an area on the screen such that when a user tapped on, it initiated or triggered another activity to execute. The example of a sensitive region was a button. The number of pan taps was the number that each participant tapped on the pan buttons to pan the map. There were four pan buttons, which were pan up, pan down, pan right, and pan left (Figure 4.3). The number of zoom taps was the number that each participant tapped on the zoom button to zoom the map. There were two zoom buttons, which were zoom in and zoom out (Figure 4.3).

Table 4.2 showed a fragment of the raw results from our study that consisted of a participant ID, and six metrics that we have mentioned. For the last two columns, the number of pan taps and the number of zoom taps, they only showed the total number of pan taps that each participant tapped and the total number of zoom taps that each participant tapped,

respectively. The pattern of pan taps (the combination of up, down, right, and left) and the pattern of zoom taps (combination of in and out) were recorded in the log file.

Table 4.2 Example Raw Results from Study.

	A	B	C	D	E	F	G
1	Participant	AddressVerification(10)	TimeUsed(Minute)	TotalDistance(km)	ErrorTap	PanTap	ZoomTap
2	1	9.5	52.68	2.672098188	51	48	31
3	2	9	43.65	2.31425989	2	75	37
4	3	8.5	64.43	3.510059499	45	156	61
5	4	8	56.8	2.767203439	13	123	51
6	5	10	41.72	2.242753113	50	48	30

4.4 Preliminary Analyzed Results

After we finished the field study with every participant, we had the following data for each participant:

1. Participant's spatial visualization
2. Participant's treatment
3. Participant's metrics

We knew that each participant was either high or low spatial visualization based on his/her score of a Paper Folding Test (Subsection 4.1.1). For a participant's treatment, we knew that each participant received what treatment, which meant that we knew which two features that were enabled and another one feature that was disabled for that participant (Subsection 4.2.2). Participant's metrics were the six metrics that we mentioned in Subsection 4.3.

We set up hypotheses and did a data analysis according to the corresponding hypothesis. For each hypothesis, we addressed the hypothesis, the result of hypothesis tests, and provided a bar chart. We used the Mann-Whitney test [70]. The objective of a Mann-Whitney test was to test whether two sets of data were significantly difference or not.

Based on results of hypothesis tests, we divided our write up into two parts. The first part investigated the significant hypotheses and the second part looked at the suggestive hypotheses. Significant hypotheses were hypotheses that had a significant test result, that was, a p -value calculated from a hypothesis test was significant (p -value ≤ 0.05). Suggestive hypotheses were hypotheses that did not have a significant test result, but their p -values were very close to significant value.

4.4.1 Hypotheses for Study I

This subsection sets up interesting hypotheses that were tested for their significance. There are thirty-six hypotheses in total where eighteen hypotheses are for low VZ participants (Table 4.3) and other eighteen hypotheses are for high VZ participants (Table 4.4). Four hypotheses are significant. Three hypotheses are suggestive. Twenty-nine hypotheses are failed to reject. Every significant hypothesis and every suggestive hypothesis are discussed in detail.

Table 4.3 Hypotheses for Low VZ Participants.

#	Null Hypothesis H(0)	p-value	Significance
1	For low VZ, those who had GPS hadn't higher address verification score than those who hadn't.	0.01253	Reject H(0)
2	For low VZ, those who had Object-Indexing had higher address verification score than those who hadn't.	0.02058	Reject H(0)
3	For low VZ, those who had Mini Map hadn't higher address verification score than those who hadn't.	1	Fail to Reject H(0)
4	For low VZ, those who had GPS hadn't lower time than those who hadn't.	0.6622	Fail to Reject H(0)
5	For low VZ, those who had Object-Indexing hadn't lower time than those who hadn't.	0.5915	Fail to Reject H(0)
6	For low VZ, those who had Mini Map hadn't lower time than those who hadn't.	0.3614	Fail to Reject H(0)
7	For low VZ, those who had GPS hadn't lower distance than those who hadn't.	1	Fail to Reject H(0)
8	For low VZ, those who had Object-Indexing had lower distance than those who hadn't.	0.07585	Suggestive to Reject H(0)
9	For low VZ, those who had Mini Map hadn't lower distance than those who hadn't.	0.09427	Fail to Reject H(0)
10	For low VZ, those who had GPS hadn't lower error taps than those who hadn't.	0.205	Fail to Reject H(0)
11	For low VZ, those who had Object-Indexing hadn't lower error taps than those who hadn't.	0.231	Fail to Reject H(0)
12	For low VZ, those who had Mini Map hadn't lower error taps than those who hadn't.	1	Fail to Reject H(0)
13	For low VZ, those who had GPS hadn't lower pan taps than those who hadn't.	0.06649	Suggestive to Reject H(0)
14	For low VZ, those who had Object-Indexing had lower pan taps than those who hadn't.	0.07585	Suggestive to Reject H(0)
15	For low VZ, those who had Mini Map hadn't lower pan taps than those who hadn't.	1	Fail to Reject H(0)
16	For low VZ, those who had GPS hadn't lower zoom taps than those who hadn't.	0.3818	Fail to Reject H(0)
17	For low VZ, those who had Object-Indexing hadn't lower zoom taps than those who hadn't.	0.5912	Fail to Reject H(0)
18	For low VZ, those who had Mini Map hadn't lower zoom taps than those who hadn't.	0.8194	Fail to Reject H(0)

Table 4.4 Hypotheses for High VZ Participants.

#	Null Hypothesis H(0)	p-value	Significance
19	For high VZ, those who had GPS hadn't higher address verification score than those who hadn't.	0.4247	Fail to Reject H(0)
20	For high VZ, those who had Object-Indexing hadn't higher address verification score than those who hadn't.	0.3126	Fail to Reject H(0)
21	For high VZ, those who had Mini Map hadn't higher address verification score than those who hadn't.	0.05295	Reject H(0)
22	For high VZ, those who had GPS hadn't lower time than those who hadn't.	0.7694	Fail to Reject H(0)
23	For high VZ, those who had Object-Indexing hadn't lower time than those who hadn't.	0.2465	Fail to Reject H(0)
24	For high VZ, those who had Mini Map hadn't lower time than those who hadn't.	0.4398	Fail to Reject H(0)
25	For high VZ, those who had GPS hadn't lower distance than those who hadn't.	0.3048	Fail to Reject H(0)
26	For high VZ, those who had Object-Indexing hadn't lower distance than those who hadn't.	0.8772	Fail to Reject H(0)
27	For high VZ, those who had Mini Map hadn't lower distance than those who hadn't.	0.1891	Fail to Reject H(0)
28	For high VZ, those who had GPS hadn't lower error taps than those who hadn't.	0.3785	Fail to Reject H(0)
29	For high VZ, those who had Object-Indexing hadn't lower error taps than those who hadn't.	0.757	Fail to Reject H(0)
30	For high VZ, those who had Mini Map hadn't lower error taps than those who hadn't.	0.5881	Fail to Reject H(0)
31	For high VZ, those who had GPS had lower pan taps than those who hadn't.	0.02789	Reject H(0)
32	For high VZ, those who had Object-Indexing hadn't lower pan taps than those who hadn't.	0.3539	Fail to Reject H(0)
33	For high VZ, those who had Mini Map hadn't lower pan taps than those who hadn't.	0.1891	Fail to Reject H(0)
34	For high VZ, those who had GPS hadn't lower zoom taps than those who hadn't.	0.3042	Fail to Reject H(0)
35	For high VZ, those who had Object-Indexing hadn't lower zoom taps than those who hadn't.	0.6989	Fail to Reject H(0)
36	For high VZ, those who had Mini Map hadn't lower zoom taps than those who hadn't.	0.536	Fail to Reject H(0)

4.4.1.1 Significant Hypotheses

From Table 4.3 and Table 4.4, there were 4 significant hypotheses, which were:

- **Hypothesis#1:** For low spatial visualization participants, those who had GPS did not have higher address verification scores than those who did not have GPS.
 - An alternative hypothesis (H(a)) for this case was: *“For low spatial visualization participants, those who had GPS had higher address verification scores than those who did not have GPS.”*
 - For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the address verification score of every low spatial visualization participants whose treatment had GPS enabled. The second dataset contained the address verification score of every low spatial visualization participants whose treatment had GPS disabled.
 - The hypothesis test returned a p -value of 0.013. This value told us that the difference between two datasets was significant.
 - Figure 4.7 shows the bar chart that compared the average of address verification score of low spatial visualization participants who had GPS and those who did not have GPS. The average score of those who had GPS was 9.63 and the average score of those who did not have GPS was 8.3.
- **Hypothesis#2:** For low spatial visualization participants, those who had Object-Indexing did not have lower address verification scores than those who did not have Object-Indexing.
 - An alternative hypothesis (H(a)) for this case was: *“For low spatial visualization participants, those who had Object-Indexing had lower address verification scores than those who did not have Object-Indexing.”*

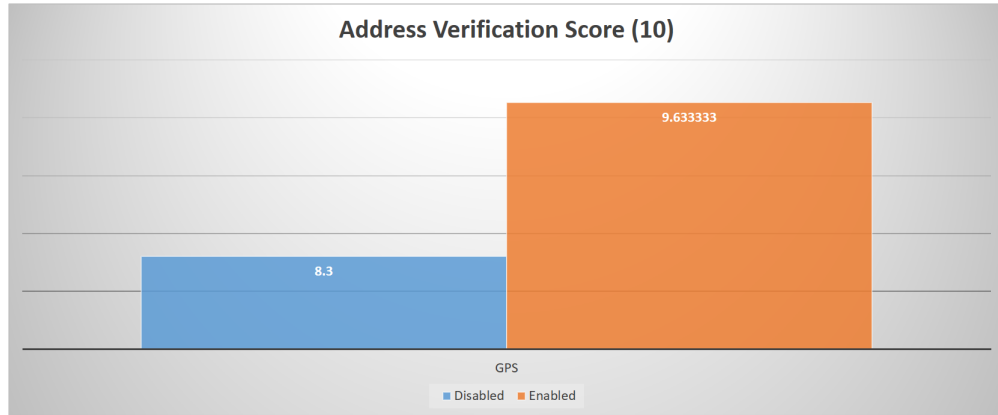


Figure 4.7 Average Address Verification Score between 2 Datasets.

- For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the address verification score of every low spatial visualization participants whose treatment had Object-Indexing enabled. The second dataset contained the address verification score of every low spatial visualization participants whose treatment had Object-Indexing disabled.
- The hypothesis test returned a p -value of 0.02. This value told us that a difference between two datasets was significant.
- Figure 4.8 shows the bar chart that compared the average of address verification score of low spatial visualization participants who had Object-Indexing and those who did not have Object-Indexing. The average score of those who had Object-Indexing was 9.03 and the average score of those who did not have Object-Indexing was 9.92.

- **Hypothesis#21:** For high spatial visualization participants, those who had Mini Map did not have higher address verification scores than those who did not have Mini Map.

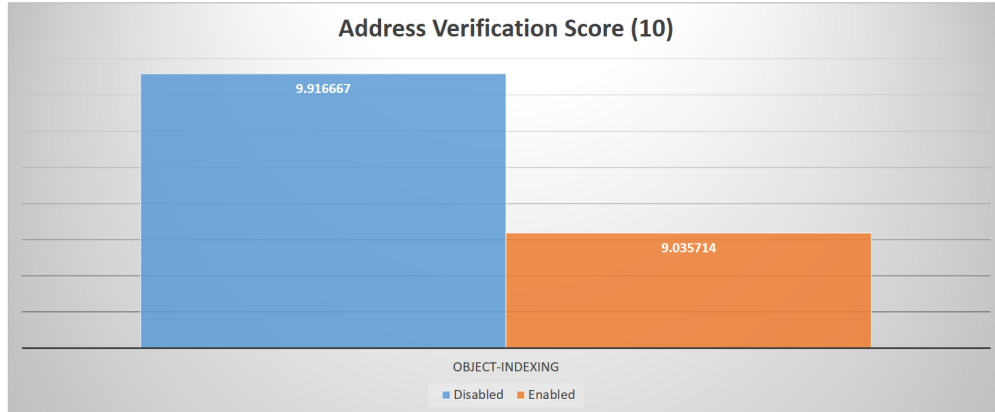


Figure 4.8 Average Address Verification Score between 2 Datasets.

- An alternative hypothesis ($H(a)$) for this case was: *“For high spatial visualization participants, those who had Mini Map had higher address verification scores than those who did not have Mini Map.”*
- For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the address verification score of every high spatial visualization participants whose treatment had Mini Map enabled. The second dataset contained the address verification score of every high spatial visualization participants whose treatment had Mini Map disabled.
- The hypothesis test returned a p -value of 0.05. This value told us that the difference between two datasets was significant.
- Figure 4.9 shows the bar chart that compared the average of address verification score of high spatial visualization participants who had Mini Map and those who did not have Mini Map. The average score of those who had Mini Map was 9.7 and the average score of those who did not have Mini Map was 8.75.

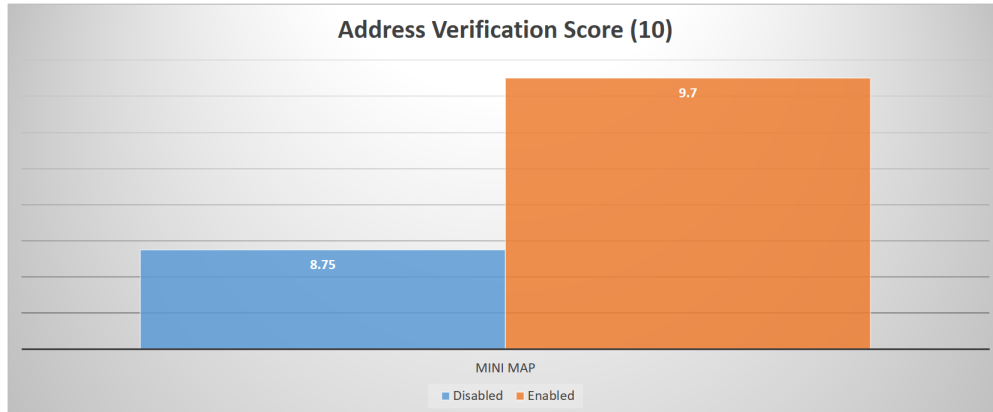


Figure 4.9 Average Address Verification Score between 2 Datasets.

- **Hypothesis#31:** For high spatial visualization participants, those who had GPS did not have higher number of pan taps than those who did not have GPS.
 - An alternative hypothesis (H(a)) for this case was: *“For high spatial visualization participants, those who had GPS had higher number of pan taps than those who did not have GPS.”*
 - For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the number of pan taps of every high spatial visualization participants whose treatment had GPS enabled. The second dataset contained the number of pan taps of every high spatial visualization participants whose treatment had GPS disabled.
 - The hypothesis test returned a p -value of 0.02. This value told us that the difference between two datasets is significant.
 - Figure 4.10 shows the bar chart that compared the average of number of pan taps of high spatial visualization participants who had GPS and those who did not have GPS. The average number of pan taps of those who had GPS was 118.88 and the average score of those who did not have GPS was 66.2.

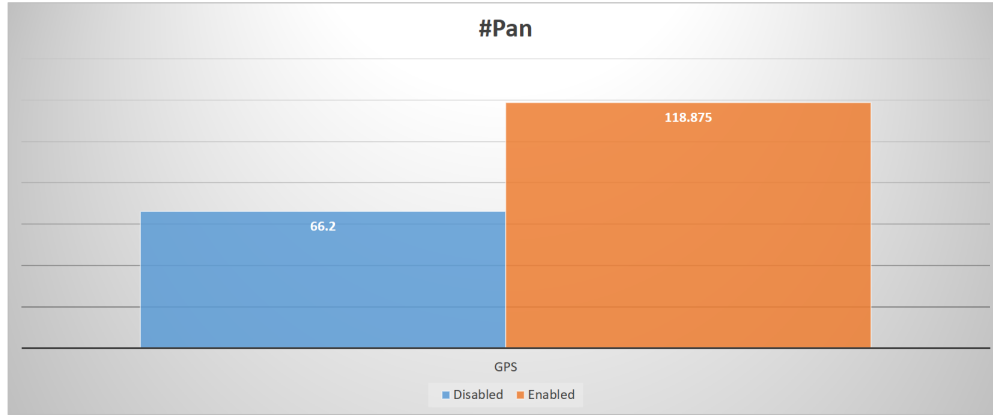


Figure 4.10 Average Address Verification Score between 2 Datasets.

4.4.1.2 Suggestive Hypotheses

From Table 4.3 and Table 4.4, there are 3 suggestive hypotheses, which are:

- **Hypothesis#8:** For low spatial visualization participants, those who had Object-Indexing did not have higher distance than those who did not have Object-Indexing.
 - An alternative hypothesis (H(a)) for this case was: *“For low spatial visualization participants, those who had Object-Indexing had higher distance than those who did not have Object-Indexing.”*
 - For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the distance of every low spatial visualization participants whose treatment had Object-Indexing enabled. The second dataset contained the distance of every low spatial visualization participants whose treatment had Object-Indexing disabled.
 - The hypothesis test returned a p -value of 0.07. This value showed that the difference between the two datasets was very close to significant value (0.05).

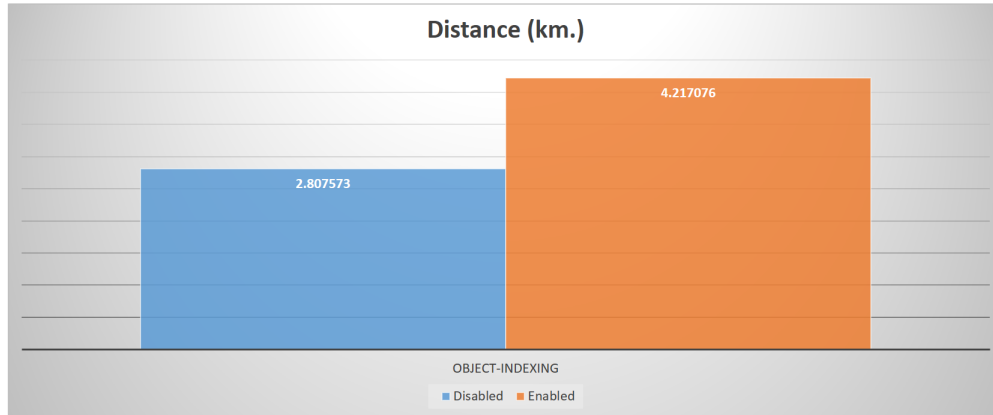


Figure 4.11 Average Distance between 2 Datasets.

- Figure 4.11 shows the bar chart that compared the average of distance of low spatial visualization participants who had Object-Indexing and those who did not have Object-Indexing. The average distance of those who had Object-Indexing was 4.22 kilometers and the average score of those who did not have Object-Indexing was 2.81 kilometers.
- **Hypothesis#13:** For low spatial visualization participants, those who had GPS did not have lower number of pan taps than those who did not have GPS.
 - An alternative hypothesis (H(a)) for this case was: *“For low spatial visualization participants, those who had GPS had lower number of pan taps than those who did not have GPS.”*
 - For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the number of pan taps of every low spatial visualization participants whose treatment had GPS enabled. The second dataset contained the number of pan taps of every low spatial visualization participants whose treatment had GPS disabled.

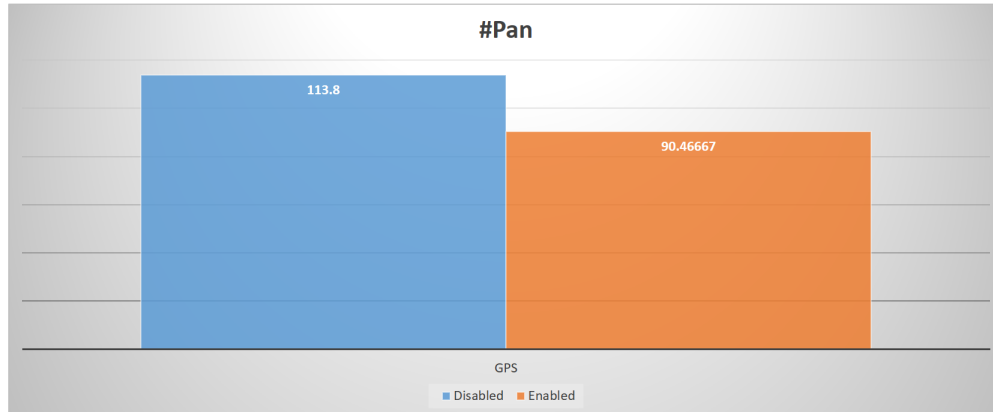


Figure 4.12 Average Distance between 2 Datasets.

- The hypothesis test returned a p -value of 0.06. This value showed that the difference between the two datasets was very close to significant value (0.05).
- Figure 4.12 shows the bar chart that compared the average of number of pan taps of low spatial visualization participants who had GPS and those who did not have GPS. The average total number of pan taps of those who had GPS was 90.47 and the average score of those who did not have GPS was 113.8.
- **Hypothesis#14:** For low spatial visualization participants, those who had Object-Indexing did not have higher number of pan taps than those who did not have Object-Indexing.
 - An alternative hypothesis ($H(a)$) for this case was: *“For low spatial visualization participants, those who had Object-Indexing had higher number of pan taps than those who did not have Object-Indexing.”*
 - For this hypothesis, we used two datasets to run a hypothesis test. The first dataset contained the number of pan taps of every low spatial visualization participants whose treatment had Object-Indexing enabled. The second dataset

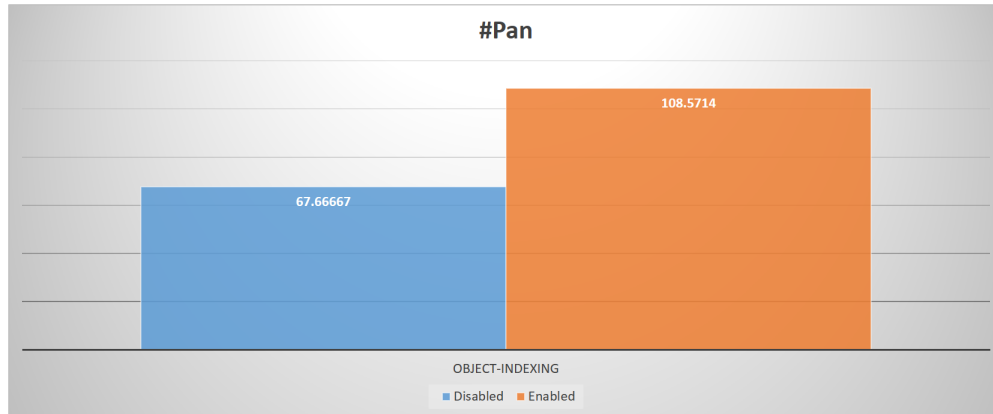


Figure 4.13 Average Distance between 2 Datasets.

contained the number of pan taps of every low spatial visualization participants whose treatment had Object-Indexing disabled.

- The hypothesis test returned a p -value of 0.07. This value showed that the difference between the two datasets was very close to significant value (0.05).
- Figure 4.13 shows the bar chart that compared the average of number of pan taps of low spatial visualization participants who had Object-Indexing and those who did not have Object-Indexing. The average total number of pan taps of those who had Object-Indexing was 108.57 and the average score of those who did not have Object-Indexing was 67.67.

4.5 Post-study Discussion

Our hypotheses and the results from hypothesis tests showed that GPS enhanced the performance of low spatial visualization participants by significantly increasing the address verification score and decreasing the number of panning taps. However, GPS reduced the performance of high spatial visualization participants by significantly increasing the number of panning taps.

The results also showed that Object-Indexing did not seem to be a feature for low spatial visualization participants because it reduced the performance by significantly decreasing the address verification score and increasing the number of panning taps.

Based on this preliminary study, Mini Map was the only feature that benefited high spatial visualization participants because it significantly increased the address verification score of high spatial visualization participants.

We gained some clues about which software features might potentially help the two groups of participants. As we previously mentioned that a purpose of this study was to see how each feature impacted participants' performance in address verification task, all the clues that we had from this study will be used in a development of an adaptive user interface software. Not only clues that we could get, we would also make a use of the prediction model [2] in order to develop an adaptive version of software interface. In addition we will continue to examine relevant literature for other features that have been shown to be helpful to high or low participants.

CHAPTER 5. STUDY II: A STUDY TO INVESTIGATE PARTICIPANTS' PERFORMANCE WITH A UNIVERSAL USER INTERFACE USING A LARGE MAP SPACE

In this chapter, we provide a detailed look at Study II. Study II was approved by the Institutional Review Board (IRB-ID: 14-526). The IRB approval document was provided in Appendix [A](#).

5.1 Objectives

As mentioned previously, the goal of this research is to find ways to flatten the space based on the users' level of spatial ability. The focus of this study was to look at map size in a universal interface format where all of the features used in Study I were present for users to take advantage of.

The positive aspect of this approach is that a larger map size gives users a better chance to efficiently use the map. The negative issue is that for really large scale surveys, like a decennial census, using tablets for all enumerators in the field drives costs too high for agency budgets.

The basic research question this study was designed to address is whether by integrating the larger map size with the Object-Indexing, the Mini Map, and GPS we would see no difference between high and low spatial visualization users.

5.2 Methodology

5.2.1 Study Designs

There were also two sessions in Study II. The first session, spatial visualization test, was the same as was used in Study I.

The field session of Study II incorporates a larger map with the interface features used in Study I. Figure 5.1 compares the size of the map between Study I (left) and Study II (right). All three UI features (GPS, Object-Indexing, and Mini Map) were active for each participant throughout the study. Another small change was the inclusion of a satellite image map of area.

Figure 5.2 - 5.4 provide snapshots of the interface design.

5.2.2 Study Procedures

5.2.2.1 Test of Spatial Visualization

We started Study II by asking a participant to take the Paper Folding test [20] to find the level of spatial visualization of the participant. The process of the test was same as Study I. It consisted of two problem sets where each set had ten questions. Each participant was allowed three minutes for each problem set. When a participant finished the test, we moved to the neighborhood to do the address verification task.

5.2.2.2 Field Portion of the Study

When arriving in the neighborhood, the investigator instructed each participant on how to use the software to verify addresses. The investigator then handed the tablet to the

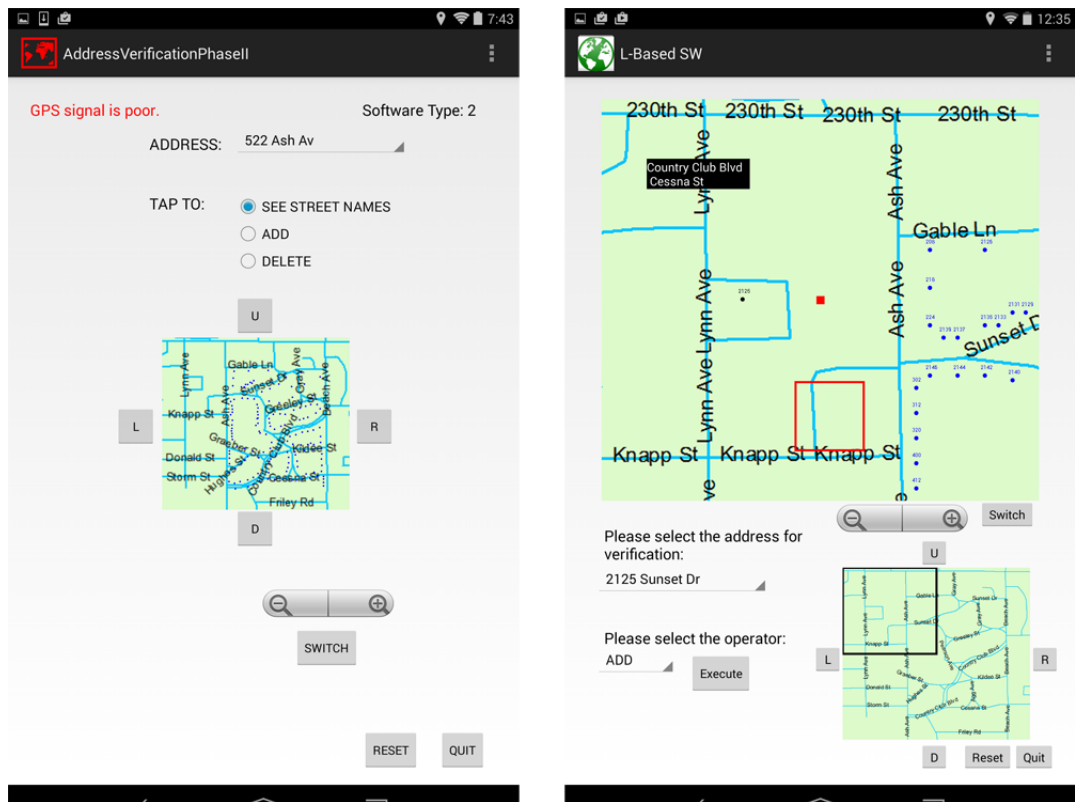


Figure 5.1 UI of Study I (left) and Study II (right).

participant. Since there was no time limit in the field study, the participant could take a time as long as he/she wanted. When the participant finished, he/she returned the tablet to the investigator. The investigator observed the participant all the time that they were in the neighborhood.

For each address to be verified, there were four possible solutions of verification as described in Table 4.1. Each participant was given instruction on address verification including a suggestion on a potential approach. The instruction and suggestion were described in Subsection 4.1.2.

5.2.3 Study Software

We take a brief look at the location-based software used in Study II. The two most compelling differences about the software are the size of the primary map relative to what was used in Study I and the inclusion of all three features discussed in the previous chapter.

5.2.3.1 Software Components

Figure 5.2 illustrates the UI of the software that participants used for address verification in Study II.

Most UI components were similar to the UI of software used in Study I. There were two maps in the UI. The first map was a Primary Map, which is the working map for participants. The second map was a Mini Map representing the complete neighborhood. There were four Pan Buttons (up, down, right, and left) for panning and a Zoom Control for zooming in/out of the Primary Map. A Switch Button was for switching between a Tiger Line map, a Google map, and a satellite image map (Figure 5.3). An Address Drop-down List contained all ten addresses that participants have to verify. An Operation Drop-down



Figure 5.2 Software UI of Study II.

List and an Operation Execute Button were for participants to select which map operation (add/delete) they would like to do with an address they were working on. A Reset Button was for resetting the UI to its beginning state. A Quit Button was for quit the software when participants finished verification.

5.2.3.2 Software Features

There were also the three user interface features that were used in Study I (Global Positioning System, Object-Indexing, and Mini Map). Since this was a universal user interface study, we included all three features for every participant. Therefore, every participant can use these three features at anytime during verification. Figure 5.4 illustrated these three features.

5.3 Results

This section explains the raw results that we gathered from thirty-one participants. We could not record the names or any identifiable data of participants due to the IRB protocol. For each participant, there were six metrics that we focused on. Those six metrics were the address verification score, the total time each participant used in address verification, the total distance each participant walked during address verification, the number of error taps the participant did during address verification, the number of pan taps the participant did during address verification, and the number of zoom taps the participant did during address verification. We gathered these data and calculated the metrics from a log file that was generated by our software when a participant finished the task.

The address verification score was the score that showed how many addresses a participant correctly verified. The top score also was 10 because there were ten addresses. For the total time that the participant used in address verification, we started recording when the

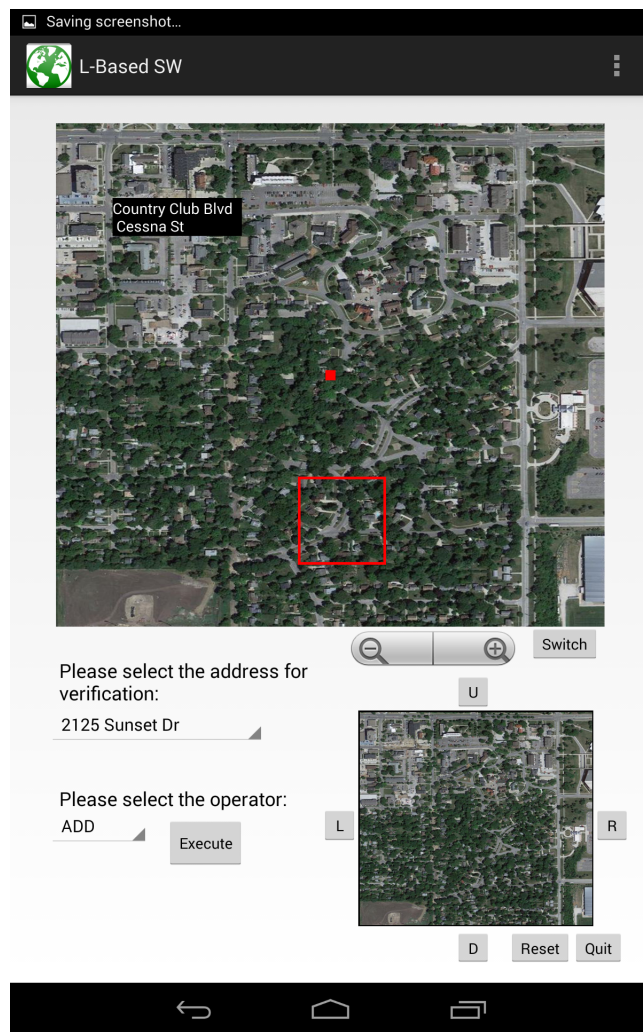


Figure 5.3 A Satellite Map in Study II.

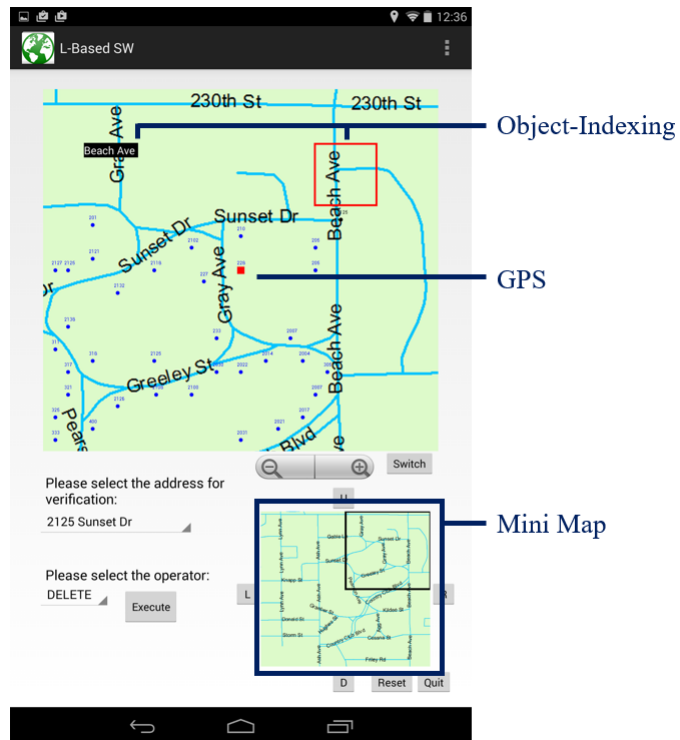


Figure 5.4 Three UI Features of Study II.

tablet was given to the participant and stopped when the participant tapped on the Quit button (Section 5.2.3.1). The basic unit of the total time was minutes.

For the total distance that the participant walked during address verification, we also calculated it by using the log file of that participant. When a participant started the task, our software kept track of the participant's location using a set of longitude and latitude coordinates. Thus, the total route of the participant could be represented by a sequence of longitude-latitude coordinates. The distance was calculated in kilometers along the path defined by the coordinates.

The number of error taps was a number that the participant tapped outside the sensitive regions [101]. The number of pan taps was the number that each participant tapped on the pan buttons to pan the map. There were four pan buttons, which were pan up, pan down, pan right, and pan left (Figure 5.2). The number of zoom taps was the number that each participant tapped on the zoom button to zoom the map. There were two zoom buttons, which were zoom in and zoom out (Figure 5.2).

Table 5.1 shows the raw results from Study II that consists of the participant ID (simply a number assigned after their identifying information had been stripped off), the scores of spatial visualization test, and six metrics that we mentioned above. For the last two columns, the number of pan taps and the number of zoom taps, they showed the total number of pan taps that each participant tapped and the total number of zoom taps that each participant tapped, respectively. The pattern of pan taps (the combination of up, down, right, and left) and the pattern of zoom taps (combination of in and out) were recorded in the log file. A partial result was published in [77].

Table 5.1 Example Results from Study II.

Participant	PaperFolding(20)	AddressVerification(10)	TimeUsedMinute	TotalDistance	ErrorTap	PanTap	ZoomTap
1	13	7.5	51.13333333	2.549739428	0	12	22
2	2	10	49.91666667	2.884057287	10	6	26
3	10	7.5	43.05	2.52628196	6	13	29
4	8	10	29.86666667	1.895308955	3	4	12
5	14	1	22.98333333	2.25754593	7	0	0
6	8	9	41.76666667	2.860666989	1	11	6
7	6	10	37.56666667	2.447426997	13	16	1
8	8	8.5	60.16666667	2.488870352	9	47	33
9	7	9.5	44.5	2.089999332	1	20	31
10	10	10	53.5	2.934723495	9	42	7
11	10	10	43.23333333	2.011961827	15	11	14
12	13	10	36.5	2.177052017	10	6	13
13	12	9	34.48333333	2.184057861	11	11	12
14	3	9.5	36.05	1.668492888	6	13	29
15	6	9	24.41666667	1.724652242	19	8	11
16	12	7.5	67.33333333	3.490689909	8	17	26
17	11	10	43.26666667	2.168070535	1	18	8
18	13	6.5	30.45	1.636737486	11	10	17
19	7	10	29.4	2.12518771	108	6	17
20	14	10	32.4	2.035000652	15	12	4
21	-4	8.5	56.8	2.755071209	3	23	37
22	8	8.5	55.51666667	2.591428279	14	28	53
23	16	8.5	32.7	2.055284613	1	12	9
24	3	7.5	52.73333333	3.220694769	4	7	22
25	5	8.5	58	2.718299514	35	17	6
26	11	10	56.71666667	2.784774017	8	4	26
27	6	9.5	49.13333333	2.640335013	19	10	42
28	7	10	71.21666667	3.200804792	15	44	30
29	18	10	27.85	1.834687365	1	4	23
30	11	10	47.41666667	2.563733803	7	9	16
31	12	7.5	31.43333333	1.818983654	58	10	33

5.4 Discussion

We set up research questions, hypotheses, and did a data analysis according to the corresponding hypothesis. For each research question, we addressed two hypotheses: a null hypothesis and an alternative hypothesis, the results of the hypothesis test, and provided a box plot. We used the Mann-Whitney hypothesis test. An objective of a Mann-Whitney test was to test whether two sets of data were significantly difference or not.

An objective of this process was to investigate whether there existed any significant difference between low and high spatial visualization groups of participants or not. There were six research questions. All hypotheses and their details of each research question are reported in this section.

5.4.1 Research Question 1: Was there a significant difference in verification score?

To answer this question, we set up following hypotheses:

H_0 : Verification scores of high and low VZ participants were **NOT** significantly different.

H_a : Verification scores of high and low VZ participants were significantly different.

We did a hypothesis test to test two sets of data. The first data set was a verification score of every high VZ participant. The second data set was a verification score of every low VZ participant.

After we ran a hypothesis test, we got a p -value of 0.4233. Since the p -value was greater than the significance level (α) of 0.05, we failed to reject H_0 . Even though we failed to reject the null hypothesis, the box plot (Figure 5.5) shows overlap of the two groups.

5.4.2 Research Question 2: Was there a significant difference in time?

To answer this question, we set up following hypotheses:

H_0 : Times of high and low VZ participants were **NOT** significantly different.

H_a : Times of high and low VZ participants were significantly different.

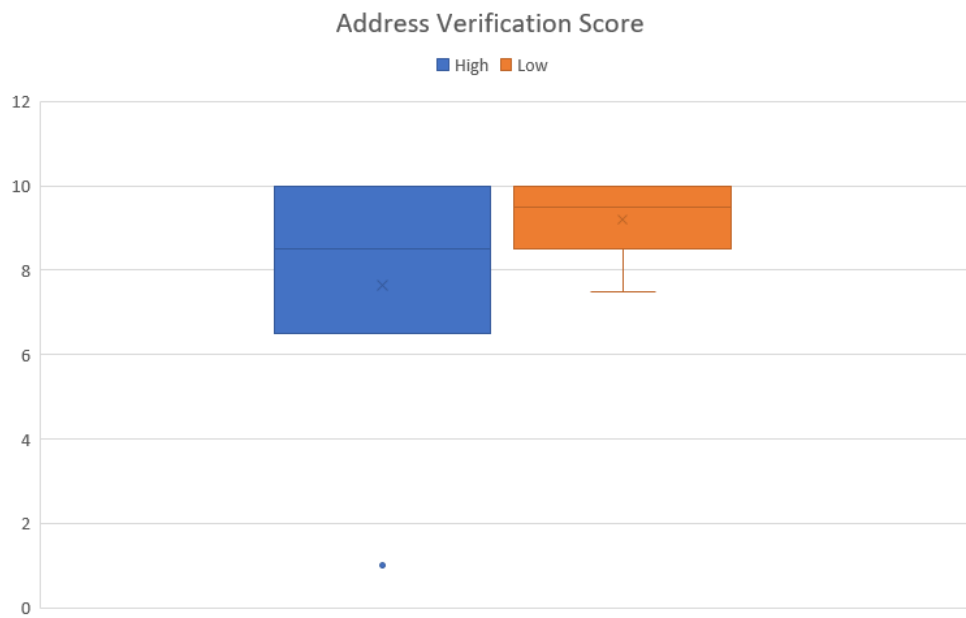


Figure 5.5 A Box Plot Illustrated Address Verification Score of High and Low VZ Participants.

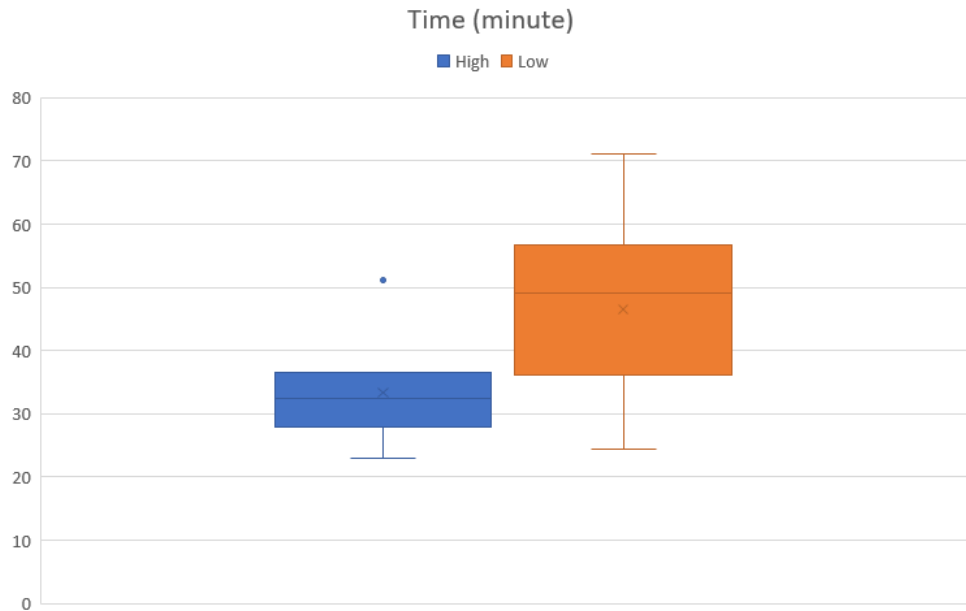


Figure 5.6 A Box Plot Illustrated Times of High and Low VZ Participants.

We did a hypothesis test to test two sets of data. The first data set was a time of every high VZ participant. The second data set was a time of every low VZ participant.

After we ran a hypothesis test, we got a p -value of 0.03868. Since the p -value was smaller than the significance level (α) of 0.05, we rejected H_0 . Therefore, times of high and low VZ participants were significantly different. A box plot in Figure 5.6 illustrated times of both high and low VZ participants.

5.4.3 Research Question 3: Was there a significant difference in distance?

To answer this question, we set up following hypotheses:

H_0 : Distances of high and low VZ participants were **NOT** significantly different.

H_a : Distances of high and low VZ participants were significantly different.

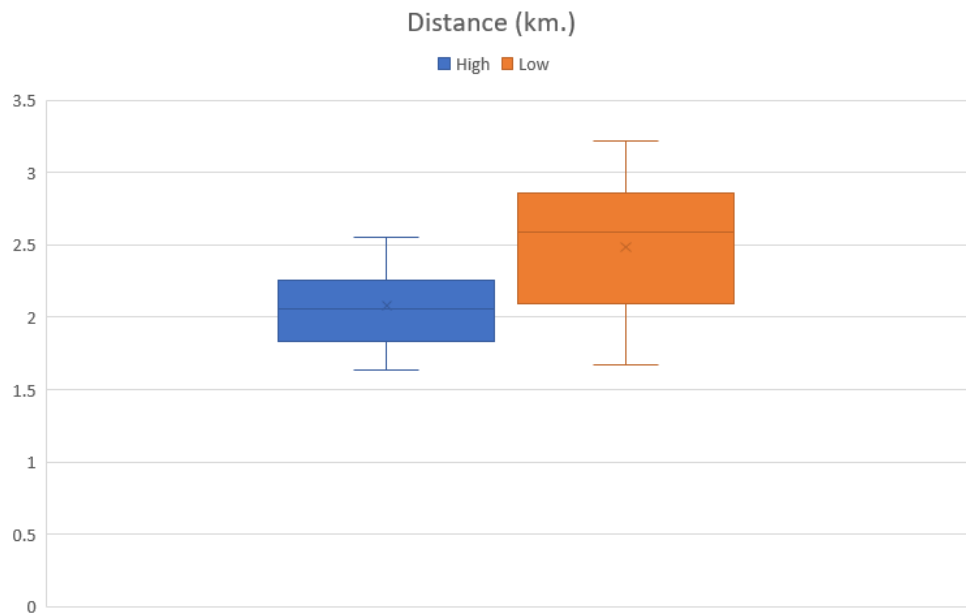


Figure 5.7 A Box Plot Illustrated Distances of High and Low VZ Participants.

We did a hypothesis test to test two sets of data. The first data set was a distance of every high VZ participant. The second data set was a distance of every low VZ participant.

After we ran a hypothesis test, we got a p -value of 0.05555. Since the p -value was greater than the significance level (α) of 0.05, we failed to reject H_0 . The p -value of 0.05555, which not significant, is suggestive of the two groups being different with respect to time. The box plot (Figure 5.7) illustrates some overlap between the two groups.

5.4.4 Research Question 4: Was there a significant difference in a number of error taps?

To answer this question, we set up following hypotheses:

H_0 : Numbers of error taps of high and low VZ participants were **NOT** significantly different.

H_a : Numbers of error taps of high and low VZ participants were significantly different.

We did a hypothesis test to test two sets of data. The first data set was a number of error taps of every high VZ participant. The second data set was a number of error taps of every low VZ participant.

After we ran a hypothesis test, we got a p -value of 0.2027. Since the p -value was greater than the significance level (α) of 0.05, we failed to reject H_0 . As with research question 1, by failing to reject the null hypothesis we can't get anything out of the Mann-Whitney test. The box plot (Figure 5.8) shows a fairly high degree of overlap.

5.4.5 Research Question 5: Was there a significant difference in a number of pan taps?

To answer this question, we set up following hypotheses:

H_0 : Numbers of pan taps of high and low VZ participants were **NOT** significantly different.

H_a : Numbers of pan taps of high and low VZ participants were significantly different.

We did a hypothesis test to test two sets of data. The first data set was a number of pan taps of every high VZ participant. The second data set was a number of pan taps of every low VZ participant.

After we ran a hypothesis test, we got a p -value of 0.104. Since the p -value was greater than the significance level (α) of 0.05, we failed to reject H_0 . Again here, failing to reject

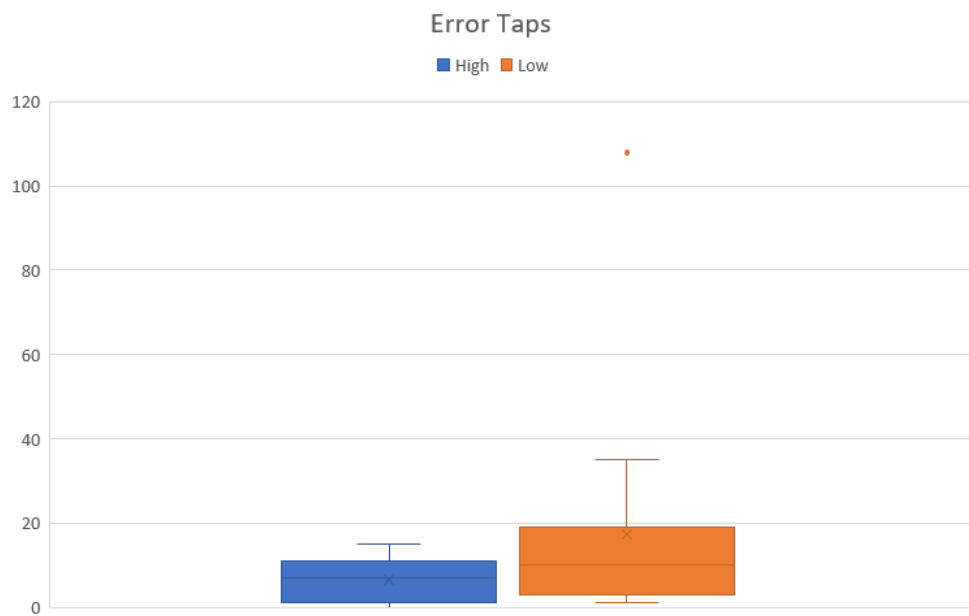


Figure 5.8 A Box Plot Illustrated Numbers of Error Taps of High and Low VZ Participants.

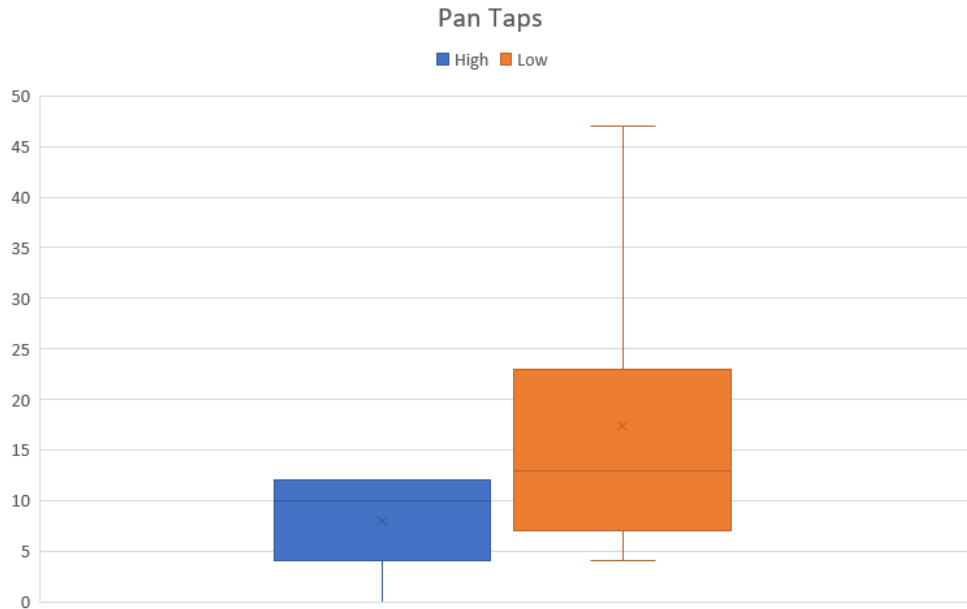


Figure 5.9 A Box Plot Illustrated Numbers of Pan Taps of High and Low VZ Participants.

the null hypothesis doesn't give us much information. We do see overlap between the two groups in the box plot (Figure 5.9).

5.4.6 Research Question 6: Was there a significant difference in a number of zoom taps?

To answer this question, we set up following hypotheses:

H_0 : Numbers of zoom taps of high and low VZ participants were **NOT** significantly different.

H_a : Numbers of zoom taps of high and low VZ participants were significantly different.

We did a hypothesis test to test two sets of data. The first data set was a number of zoom taps of every high VZ participant. The second data set was a number of zoom taps of every low VZ participant.

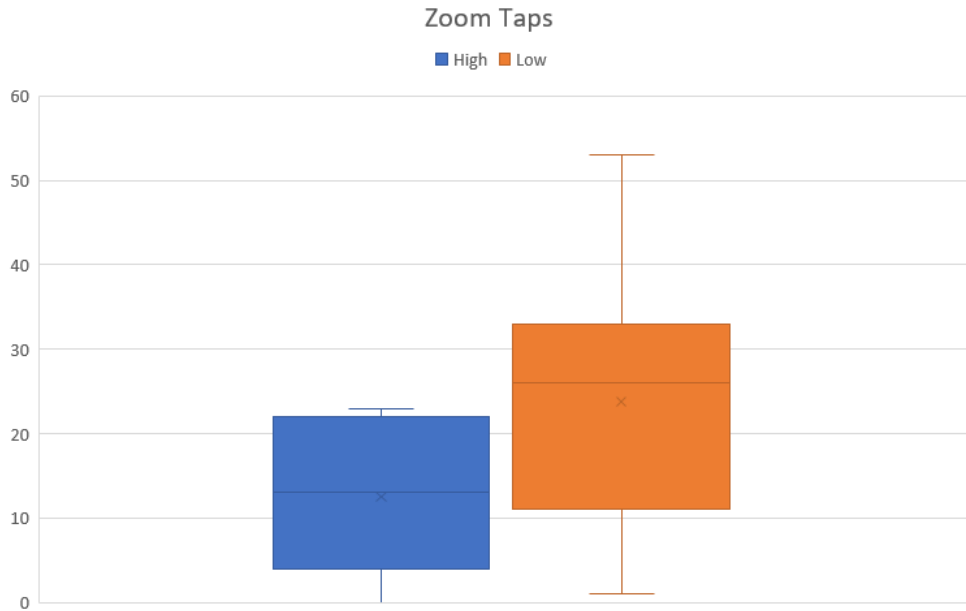


Figure 5.10 A Box Plot Illustrated Numbers of Zoom Taps of High and Low VZ Participants.

After we ran a hypothesis test, we got a p -value of 0.09042. Since the p -value was greater than the significance level (α) of 0.05, we failed to reject H_0 . The p -value of 0.09042 is just beyond being suggestive and we see a large area of overlap in the box plot (Figure 5.10).

5.5 Conclusions

As we previously mentioned, an objective of Study II was to confirm that a universal UI was not a solution for improving usability [53]. In Study II, we did a study by asking a participant to verify addresses in the neighborhood using a universal UI software. Thirty one participants participated in the study. We ran hypothesis tests, which comparing between low and high VZ participants, to investigate whether there existed any significant difference in six metrics, which were verification score, time, distance, number of errors, number of pan taps, and number of zoom taps.

The result from this study are mixed. We see some degree of flattening looking at the plots (Figure 5.5 - 5.10), but did reject the null hypothesis for research question 2 (time) with a significant p -value (0.03868) and research question 3 (distance) with a suggestive p -value (0.05555). This result corresponded with what we found in the literature [6][7] that confirmed high VZ users had better performance when using any software.

Therefore, a universal UI software doesn't appear to be a solution to improve usability of users with different VZ. This was true in spite of using a larger primary map. In Chapter 6, we look at a study of address verification using an adaptive UI software, which an adaptation was based on VZ level of a user, to see whether we could find a significant improvement of performance of users with different VZ.

CHAPTER 6. STUDY III: A STUDY TO INVESTIGATE PARTICIPANTS' PERFORMANCE WITH AN ADAPTIVE USER INTERFACE

In this chapter, we provide a detailed look at Study III. The focus of Study III was to show the advantage of an adaptive UI. Study III was approved by the Institutional Review Board (IRB-ID: 16-288). The IRB approval document is provided in Appendix [A](#).

6.1 Objectives

The primary objective of Study III was to prove that adaptive UI software to verify street addresses improved overall performance for both high and low VZ participants. This significant performance indicated that participants can do the task better with the adaptive UI software than a non-adaptive UI (which we denote the traditional UI). Statistical hypothesis tests were executed and reported to verify how they did better with the adaptive version.

Additionally, Study III had two additional objectives. The first of these objectives was that we would like to implement a prediction method in the software such that it was able to predict a user's VZ based on their actions on the UI. This was an extension of Batinov's study [2].

Since the prediction method was going to be implemented and deployed. Another objective was to investigate the results of the prediction, which included observing participants'

performance after the UI had adapted based on a prediction result regardless of the accuracy of the prediction. We will take a look on both participants who were correctly predicted and who were incorrectly predicted.

6.2 Methodology

6.2.1 Study Designs

There were two sessions in Study III. The first session, the spatial visualization test, was the same as Studies I and II.

For the second session, the field study was divided it into three phases. A participant was asked to verify four addresses, four addresses, and two addresses in phase 1, phase 2, and phase 3, respectively. All ten addresses were selected by our research group and were located in the same neighborhood as the previous studies. A difference between each phase was the UI of the software that the participants worked with. A set of addresses to be verified in each phase was different so a participant cannot remember the result of verification from the previous phase. Figure 6.1 shows all ten addresses.

- **Traditional UI** - The regular UI was the traditional UI. This UI is explained in detail in Section 6.2.3.
- **Adaptive UI for Low VZ** - An adaptive UI for low VZ included the traditional UI features and additional features that potentially enhanced the performance of low VZ participants. This UI is explained in detail in Section 6.2.3.
- **Adaptive UI for High VZ** - An adaptive UI for high VZ included the traditional UI features and additional features that potentially enhanced a performance of high VZ participants. This UI is explained in detail in Section 6.2.3.

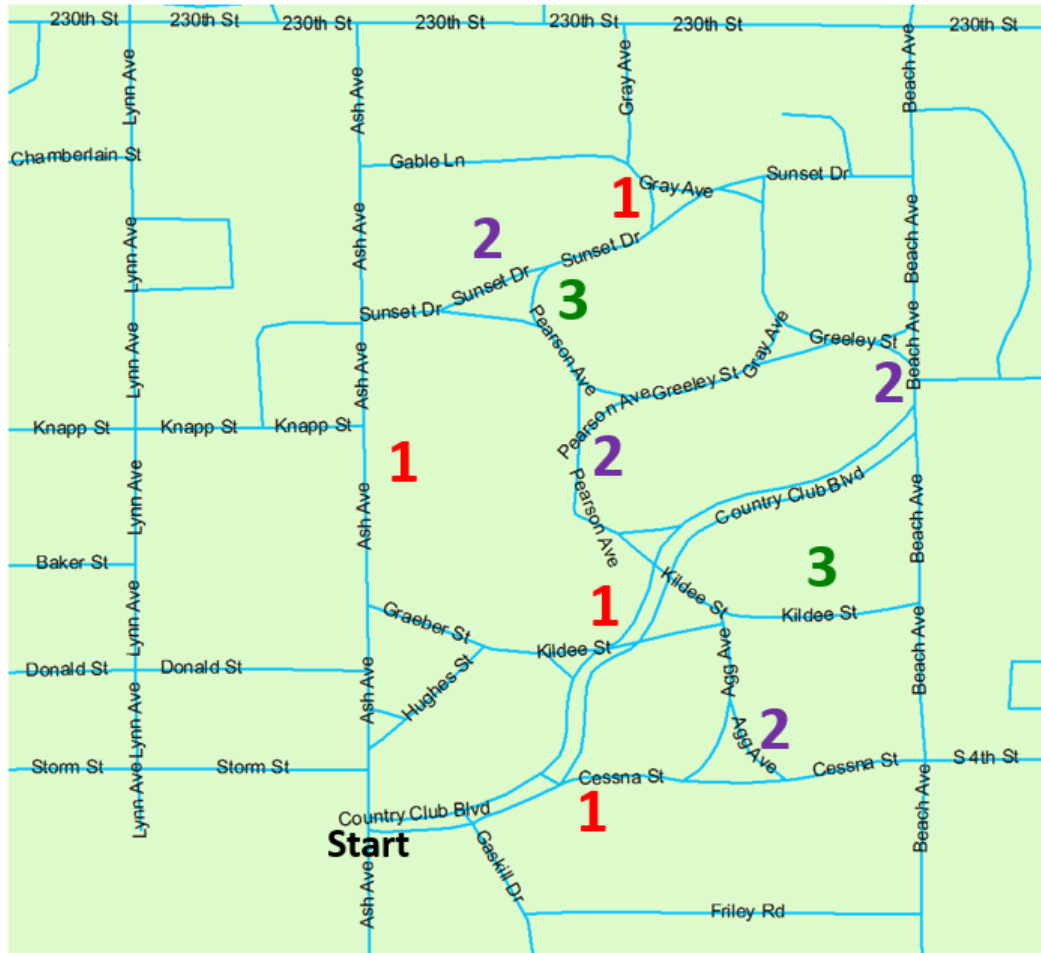


Figure 6.1 Ten addresses to be verified in three phases.

Each participant was randomly assigned a treatment for doing the verification task. There were two treatments, which were Treatment 0 and Treatment 1. For Treatment 0, the software assigned the traditional UI for a participant to work in phase 1, followed by the adaptive UI based on participant's VZ test to work in phase 2. When a participant finished first two phases, the software predicted participant's VZ level and assigned the appropriate adaptive UI for the participant to use in the final phase based on the prediction result.

Treatment 1 was set up in the opposite way as the software assigned the adaptive UI based on participant's VZ test for a participant to work in phase 1, followed by the traditional UI to use in phase 2. When a participant finished first two phases, the software predicted participant's VZ and assigned the adaptive UI for the participant to use in the final phase based on the prediction result.

The objective of having two treatments was that we would like to ensure that any significant result we would obtain from the study would not be affected by any learning effect, i.e., regardless of treatment, we expected that any participant would do better with the adaptive UI than the traditional UI.

6.2.2 Study Procedure

Study III was divided into two sessions. The first session was a spatial visualization test. The second session was a field study, which was divided into three phases.

6.2.2.1 Test of Spatial Visualization

Study III was started by asking each participant to take the Paper Folding test [20] to determine the VZ level of the participant. The process of the test was same as both Studies

I and II. There were two problem sets where each set had ten questions. Each participant had three minutes for each problem set. When a participant finished the test, we then graded the test immediately right after because the test score was going to be an input to the software. After we knew VZ level of the participant, we moved to the neighborhood to do address verification task.

6.2.2.2 Field Portion of the Study

When arriving in the neighborhood, the investigator instructed each participant on how to use the software to verify address. When the participant was ready, the investigator started the software and gave the software an input of participant's score of the Paper folding test. The software classified the participant's VZ level using the same criteria as described in Chapter 4. The participant would be classified as high VZ if the Paper Folding score was greater than or equal to 13. The participant would be classified as low VZ if the Paper Folding score was lower than or equal to 8. The investigator then handed the tablet to the participant. There was no time limit in the field study.

The software randomly assigned the treatment to the participant. Since there were two treatments and the participant could be either high or low VZ, there were four possible scenarios in Study III (Table 6.1). Figure 6.2 illustrates the work flow of the participant in Study III. From the work flow, when the participant verified all four addresses in phase 1, he/she had to submit phase 1 in order to go to phase 2 (and also from phase 2 to phase 3). Once the participant submitted, he/she could not go back to the previous phase again.

When the participant finished, he/she returned the tablet to the investigator and then received \$20 gift card as the compensation. The investigator observed the participant all the time that they were in the neighborhood. For each address to be verified, there were

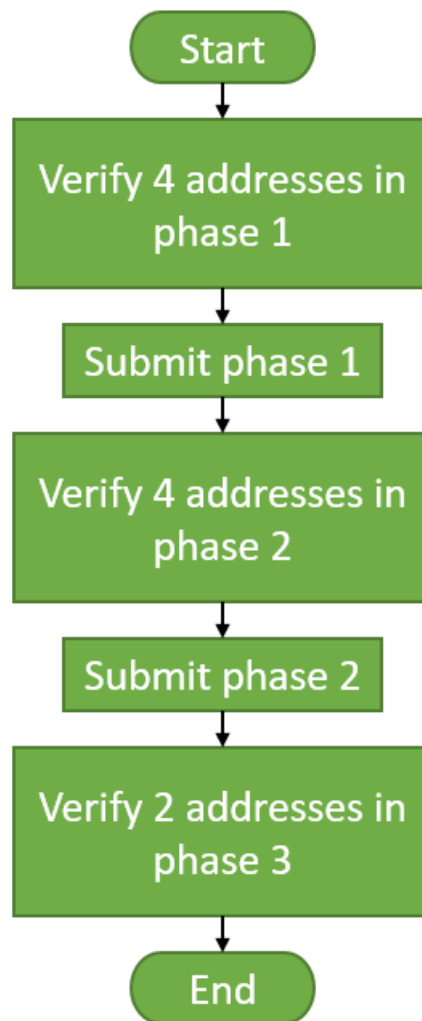


Figure 6.2 Participant's Work Flow in Study III.

Table 6.1 Possible Scenarios in Study III.

Scenario	VZ	Treatment	UI of Phase 1	UI of Phase 2	UI of Phase 3
1	Low	0	Regular UI	Adapted UI for Low VZ	Up to Prediction Result
2	Low	1	Adapted UI for Low VZ	Regular UI	Up to Prediction Result
3	High	0	Regular UI	Adapted UI for High VZ	Up to Prediction Result
4	High	1	Adapted UI for High VZ	Regular UI	Up to Prediction Result

four possible solutions of verification as described in Table 4.1. Each participant was given instruction on address verification including a suggestion on a potential approach. The instruction and suggestion were described in Subsection 4.1.2.

6.2.3 Study Software

We take a brief look at the location-based software used in Study III. The overall the UI was similar to the UI in Study I. It served all map functions that a participant needed to verify an address. However, since the software had to support more than one UI, there were some additional features that made the UI different in the adaptive states. The UI for low VZ consisted of the traditional UI and additional features for low VZ users. In a similar fashion, the UI for high VZ consisted of the traditional UI and additional features for high VZ users. These additional features will be described later in this subsection.

Furthermore, the software in Study III had a module to predict a user's VZ level based on a set of user's action on the UI. The detail of the prediction module are discussed later in this subsection as well.

6.2.3.1 Software Components

Figure 6.3 illustrates the traditional UI of the software that participants used for address verification in Study III in either phase 1 or phase 2 according to the treatment.

Most UI components were similar to the UI of software used in Studies I and II. There were two maps in the UI. The first map was a Primary Map, which is the working map for participants. The second map was a Mini Map representing the complete neighborhood. There were four Pan Buttons (up, down, right, and left) for panning and a Zoom Control for zooming in/out of the Primary Map. A Switch Button was for switching between a Tiger Line map, and a Google map. A Change Button was available for opening up an address list (Figure 6.4). An address list contained four addresses, four addresses, and two addresses in phase 1, phase 2, and phase 3, respectively. An Operation Radio Buttons List was available for participants to select which map operation (add/delete) they would like to do with an address they were working on. A Submit Phase Button was available in phases 1 and 2 for submitting the current phase and continuing on to the next phase. A Quit Button was for quitting the software when participants finished verification. Note that a Quit Button would be appear only at phase 3 (Figure 6.5).

6.2.3.2 Software Features

We take a look at additional software features in the traditional UI and both adaptive UIs (high VZ & low VZ). We also discussed about the idea of the prediction module that could predict a user's VZ.

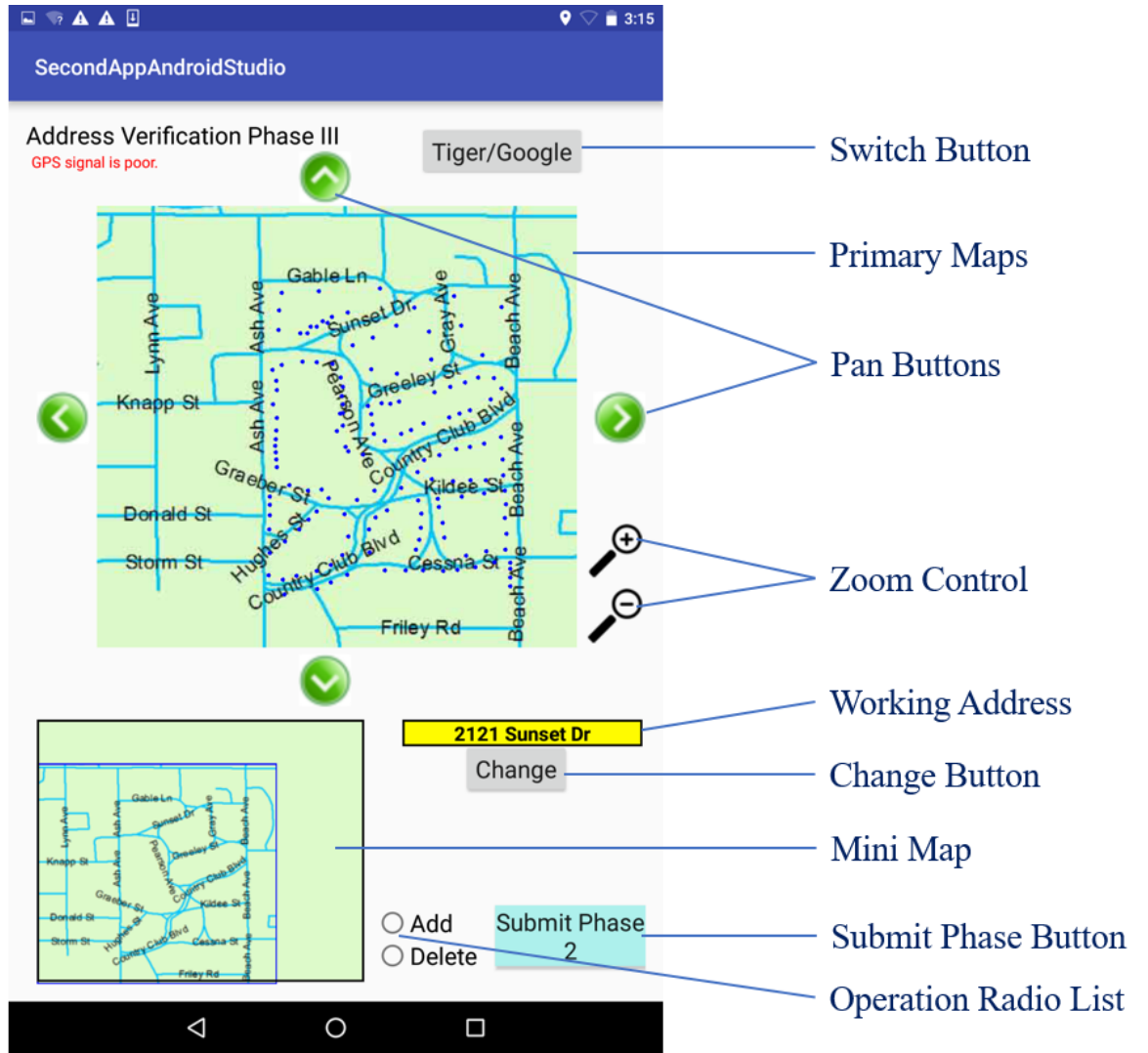


Figure 6.3 Traditional UI of Study III.

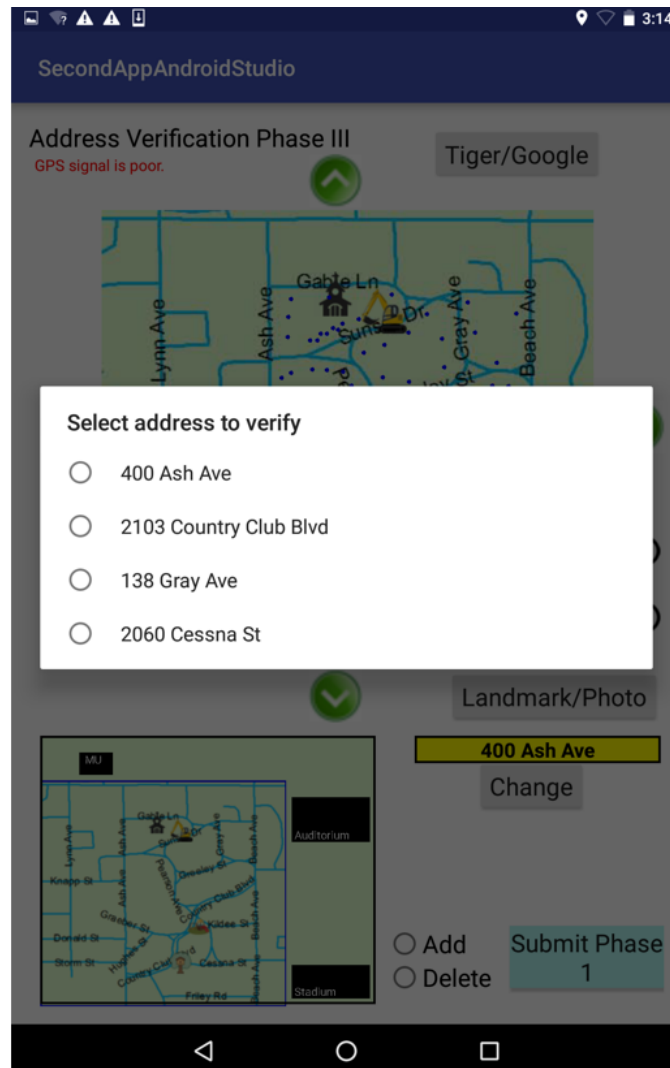


Figure 6.4 Address List in UI of Study III.

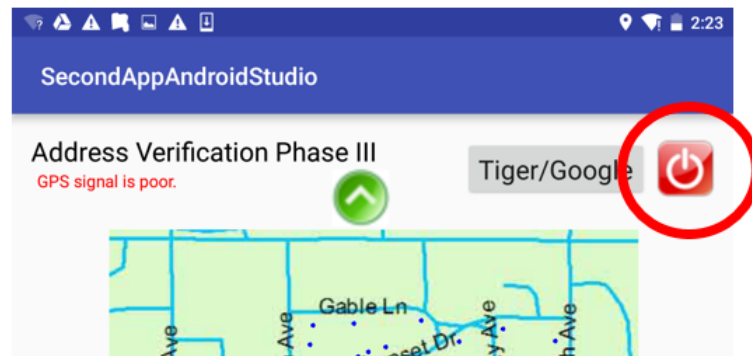


Figure 6.5 Quit Button in UI of Phase 3.

6.2.3.2.1 Features on Traditional UI

The traditional UI came with two standard features: GPS and Mini Map. Both GPS and Mini Map functioned exactly the same as they did in Study I and Study II.

6.2.3.2.2 Features on the Adaptive UI for Low VZ

A low VZ participant was assigned the adaptive UI for low VZ in either phase 1 or phase 2 based on the assigned treatment. The adaptive UI for low VZ contained two additional features, which were the implementation of landmarks and the Object-Indexing (Type I). There was no particular preference nor hierarchy between these two features.

Combining the fact that landmarks were suggested as an effective aid for map-involved tasks such as way-finding or route-learning [75][25][99][86][103] and low VZ users preferred a UI that provided aids rather than a UI with a limited number of aids, i.e., low VZ users did not prefer to learn a software by discovering by themselves [7], we decided to add land-

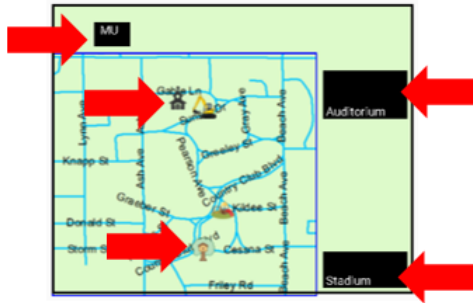


Figure 6.6 Major Landmarks and Minor Landmarks were located on the Mini Map.

marks on the adaptive UI for low VZ participants. In Study III, there were two types of a landmark that were implemented; a major landmark and a minor landmark. The major landmarks were located on the Mini Map (Figure 6.6). An example of major landmarks is the ISU Memorial Union. The minor landmarks were located on both Primary Map and Mini Map (Figure 6.6 & 6.7). An example of minor landmarks is the playground. We put the major landmarks only on the Mini Map because all available major landmarks (ISU Memorial Union, Stephen Auditorium, and Jack Trice stadium) were located outside the neighborhood. The Landmark/Photo Button (Figure 6.7) allowed a participant to switch between two landmark representations; either icon or photo.

The Object-Indexing (Type I) was a deployment of the Object-Indexing approach (Chapter 3). It was re-designed for low VZ users. The Object-Indexing (Type I) allowed participants to tap on any street in the Primary Map to see a street name or any intersection to see street names of all streets intersected. A street name then was displayed right at the tap location (Figure 6.8).

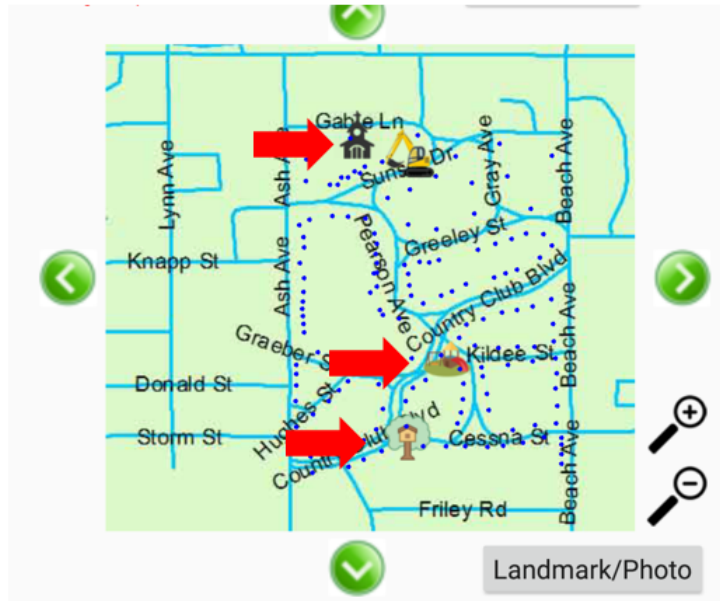


Figure 6.7 Minor Landmarks were located on the Primary Map.

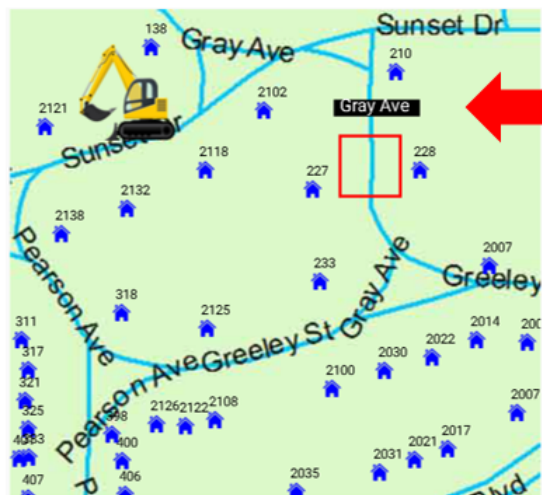


Figure 6.8 Implementation of Object-Indexing (Type I).

6.2.3.2.3 Features on the Adaptive UI for High VZ

A high VZ participant was assigned the UI for high VZ in either phase 1 or phase 2 based on the assigned treatment. The UI for high VZ contained the Object-Indexing (Type II).

The Object-Indexing (Type II) was also a deployment of Object-Indexing approach (Chapter 3). It allowed participants to see a street name if they tapped on any street on the map. The difference between Type I and Type II was a location where a street name was displayed. In Type II, a street name was displayed under the bottom-left corner of the Primary Map (Figure 6.9). We decided to display a street name outside the Primary Map because high VZ users preferred simpler UI with a minimum number of objects on a screen [10][117].

6.2.3.2.4 Prediction Module

The goal of the prediction module was to predict a participant's VZ level in real time. The prediction module was an extension of Batinov's study [2]. However, we made some modification on Batinov's work to suitably fit with Study III.

Regarding to Batinov's work, a Bagging algorithm was selected to be the classifier in the prediction module. The only difference between our prediction module and Batinov's work was a set of parameters to the classifier. In the prediction module, there were thirteen parameters, which were:

1. Time
2. Distance

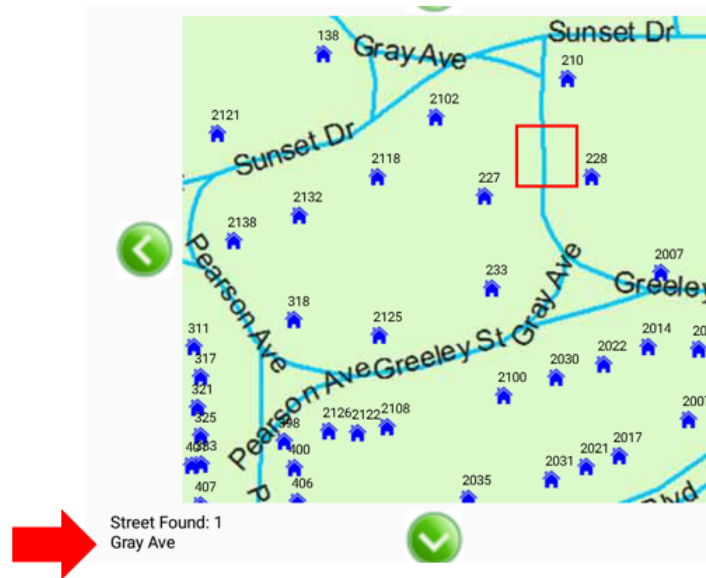


Figure 6.9 Implementation of Object-Indexing (Type II).

3. Total Taps
4. Number of Error Taps
5. Number of Pan Taps
6. Number of Zoom Taps
7. Number of Zoom In & Zoom Out Reversals
8. Number of Zoom Out & Zoom In Reversals
9. Number of Pan Left & Pan Right Reversals
10. Number of Pan Right & Pan Left Reversals
11. Number of Pan Up & Pan Down Reversals

12. Number of Pan Down & Pan Up Reversals

13. Total Reversals

We provide a brief description of each parameter. Time was the amount of time (minute) that a participant used. Distance was a distance (km.) that a participant walked. Total taps was a total number of participant's taps on the UI. Error taps was the number of error taps on the UI. The error tap means a tap on a non-sensitive region as described in Subsection 4.3. Pan taps was the number of pan taps on the UI. Zoom taps was the number of zoom taps on the UI. Reversals of zoom in & zoom out was the number that a participant zoomed out the map right after he just zoomed in the map. The idea of reversals was from the study of Batinov et al. [1]. The next five parameters were reversals of other pairs of opposite operators, which were zoom out & zoom in, pan left & pan right, pan right & pan left, pan up & pan down, and pan down & pan up. The last parameter of the prediction module was the total reversals, which was a number of every reversal a participant executed.

We implemented the prediction module using the library of Weka [29] for an Android application. We invited three anonymous participants to do a pilot test. Data from the three participants were used as the training set of the prediction module. To make the prediction module more accurate, the software re-trained the module every time we added a new participant with the data of the new participant.

The prediction module started when the software started. During phase 1 and 2, the prediction module prepared the set of parameters according to participant's actions on UI (in both phases). When a participant finished phase 2, the prediction module predicted the participant's VZ and returned the result of prediction to the software. The software then adapted the UI for phase 3 based on the prediction result. Figure 6.10, which is the extension of Figure 6.2, shows the complete work flow of every unit involved in Study III; the participant, the prediction module, and the main module of the software.

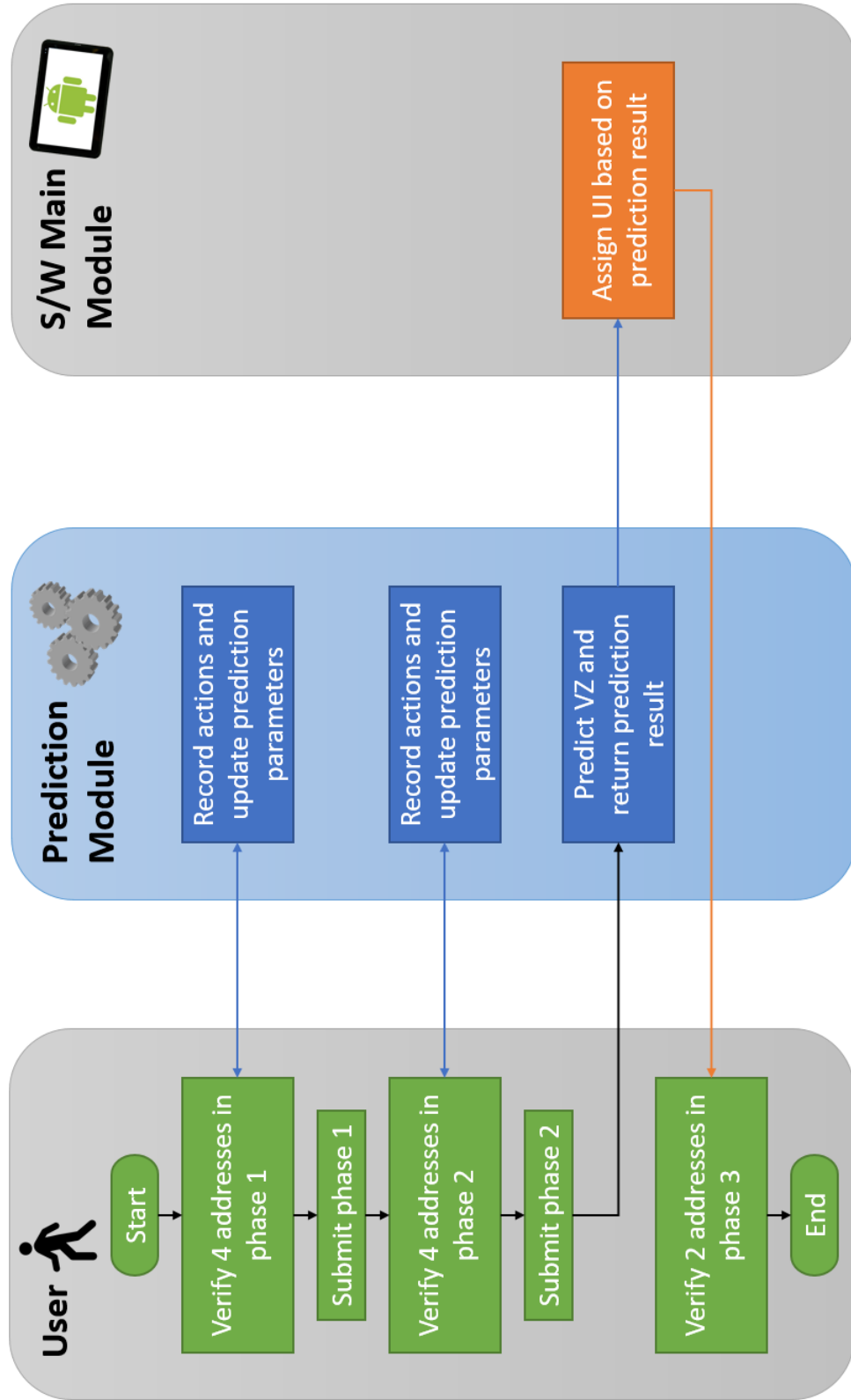


Figure 6.10 Complete Work Flow of Study III.

6.3 Results

This section explains the raw results that we gathered from thirty participants. We could not record the names or any identifiable data of participants due to the IRB protocol. For each participant, there were four metrics that we focused on for the purpose of showing the improvement of participants when using the adaptive UI. The four metrics were the time each participant used in phase 1, the time each participant used in phase 2, the distance each participant walked in phase 1, and the distance each participant walked in phase 2. Note that we see phase 3 as providing information on a separate question that we discuss in detail later in this section. We no longer used an address verification score because the number of addresses to be verified in each phase was small.

We gathered the data and calculated the metrics from a log file that was generated by our software when a participant finished the task. Table 6.2 shows the partial raw results from Study III that consists of the participant ID (simply a number assigned after their identifying information had been stripped off), the assigned treatment, the four metrics we mentioned. The remaining columns (column 5 and column 8 to 27) are metrics that were required by the prediction module.

We statistically take a look at some interesting metrics from the raw data. From thirty participants, fifteen participants were assigned treatment 0. Another half was assigned treatment 1. For the VZ level, twenty participants tested high VZ whereas ten participants tested low VZ. The average times that all participants used in phases 1 and 2 are 34.441 and 25.401 minutes, respectively. For the distance, the average distances that all participants walked during phases 1 and 2 are 1.9092 km. and 1.3731 km., respectively.

Table 6.2 Partial Results from Study III.

Participant	Treatment	Time1	Time2	TotalTime	Distance1	Distance2	TotalDistance	TotalTap	Error	Pan	Zoom	IO	OI	LR	RL	UD	DU	TotalReversal	VZ	+	-	<	>	A	V	Z
1	0	44.48	23.82	68.3	1.940672	1.157933	3.098005	60	6	32	21	0	2	0	2	1	3	8L	12	9	5	8	7	12	9	
2	0	43.21	23	66.21	2.065943	1.154338	3.220281	105	34	48	15	2	3	3	1	4	4	17L	8	7	8	10	14	16	9	
3	0	33.19	24.53	57.72	1.864976	1.426553	3.291529	104	21	53	28	1	2	2	0	2	3	10H	16	12	9	12	11	21	9	
4	0	43.02	29.87	72.89	2.186764	1.723629	3.910393	146	29	99	17	1	2	2	5	8	7	25H	9	8	18	21	27	33	8	
5	0	37.76	23.84	61.6	2.1849	1.41218	3.59708	81	4	52	24	0	1	3	2	2	4	12L	15	9	10	14	13	15	7	
8	0	27.12	19.72	46.84	1.567819	1.391131	2.95895	51	12	31	6	1	0	1	1	0	0	3H	4	2	7	8	7	9	9	
9	0	37.94	16.56	54.5	2.402449	1.149063	3.551512	106	13	43	41	2	6	0	1	2	1	12H	21	20	6	12	12	13	13	
10	1	33.52	36.04	69.56	1.880562	1.631403	3.511965	131	39	40	43	6	5	0	2	2	3	18L	22	21	3	7	8	22	12	
11	1	24.72	27.42	52.14	1.382096	1.263462	2.645498	131	21	92	13	0	4	3	2	13	14	36H	7	6	10	16	30	36	8	
12	1	35.64	25.75	61.39	2.111686	1.291287	3.402973	107	10	76	17	1	3	5	0	9	6	24H	8	9	15	13	21	27	10	
13	1	15.7	20.13	35.83	1.242403	1.147462	2.389865	20	0	16	4	0	0	2	1	1	1	5H	3	1	3	3	5	5	5	
14	0	42.84	34.81	77.65	1.62034	1.609099	3.229439	96	32	20	27	0	4	0	0	0	1	5H	14	13	3	5	4	8	7	
15	1	38.88	26.53	65.41	2.371247	1.305636	3.676883	116	17	74	23	0	0	4	1	4	5	14L	15	8	17	20	19	18	8	
16	1	49.92	35.58	85.5	2.337857	1.528143	3.866	336	63	152	62	4	13	8	8	7	7	47H	32	30	32	43	37	40	17	
17	1	35.89	33.7	69.59	1.906242	1.236735	3.142977	115	5	76	16	0	2	2	1	1	7	21H	12	4	10	15	23	28	7	
18	1	25.49	24.35	49.84	1.54554	1.327583	2.873123	72	19	29	24	0	3	1	1	1	1	7H	12	12	6	12	5	6	7	
19	0	31.5	18.42	49.92	1.774191	1.132711	2.906902	96	17	69	7	0	1	3	5	3	7	19H	4	3	15	15	19	20	10	
20	0	33.32	23.31	56.63	1.935252	1.620101	3.555353	96	16	33	41	2	3	1	0	2	2	10H	21	20	7	10	8	8	8	
21	1	26.23	27.01	53.24	1.550172	1.355012	2.905184	64	9	44	5	0	0	0	4	4	3	11H	3	2	7	8	15	14	5	
22	0	41.41	20.29	61.7	2.340612	1.523861	3.864473	147	19	120	4	0	0	6	7	13	11	37L	3	1	24	25	35	36	7	
23	1	33.89	28.22	62.11	1.507296	1.206937	2.714233	256	29	142	58	7	11	6	10	14	6	54H	29	29	24	33	36	49	13	
24	0	51.9	27.1	79	2.3778	1.571852	3.949652	281	166	54	59	5	12	4	3	5	3	32L	31	28	10	19	12	13	8	
25	1	17.4	35.31	52.71	1.532011	2.158521	3.690532	139	17	120	1	0	0	10	8	5	4	27H	1	0	35	32	25	28	10	
26	0	29.94	18.16	48.1	2.011567	1.257397	3.268964	177	42	77	27	1	5	2	5	6	4	23L	14	13	16	18	21	22	8	
27	1	21.46	17.77	39.23	2.158475	1.229457	3.387932	72	18	31	21	0	0	0	1	1	1	3L	11	10	8	8	6	9	11	
28	0	36.08	21.18	57.26	1.672831	1.320273	2.993104	78	24	22	30	3	7	0	0	1	1	12H	15	15	2	6	7	7	7	
29	0	34.07	16.91	50.98	2.341011	1.113249	3.45426	49	2	10	28	1	7	0	0	0	0	8H	15	13	0	5	2	3	8	

6.4 Discussion

In this section, we present the evidence of participants' improvement with the adaptive UI. Furthermore, we take a look on additional interesting topics related to the result of Study III, particularly, the analysis of the prediction module, the analysis of the participants' behavior, and the analysis of the mispredicted participants.

6.4.1 Improvement with Adaptive UI

As we mentioned in Section 6.3, there were four metrics that we focused on. The metrics were the time each participant used in phase 1, the time each participant used in phase 2, the distance each participant walked in phase 1, and the distance each participant walked in phase 2. To answer the question of improvement, we might simply compare the times between two phases, the phase with the traditional UI and the phase with the adaptive UI. However, we could not compare them directly because it was biased due to the unequal distance required to complete the task between phase 1 and phase 2. Note that the lists of addresses to be verified in phases 1 and 2 were fixed regardless of the assigned treatment (Figures 6.12 and 6.13). Figure 6.11 shows the distinct walking paths of one participant in phases 1 and 2. To overcome this issue, we discuss the process of time normalization. After we did the time normalization, we discuss the related hypotheses that served as the evidence of the improvement of the adaptive UI over the traditional UI.

6.4.1.1 Normalization of Times

Figure 6.12 and 6.13 shows the addresses that a participant needed to verify in phase 1 and phase 2, respectively. Although the overall walking distances to complete all addresses of both phases looked equivalent, they were really not. When we tested the software before Study III started, we visited, by walking, every address of both phases using the optimal

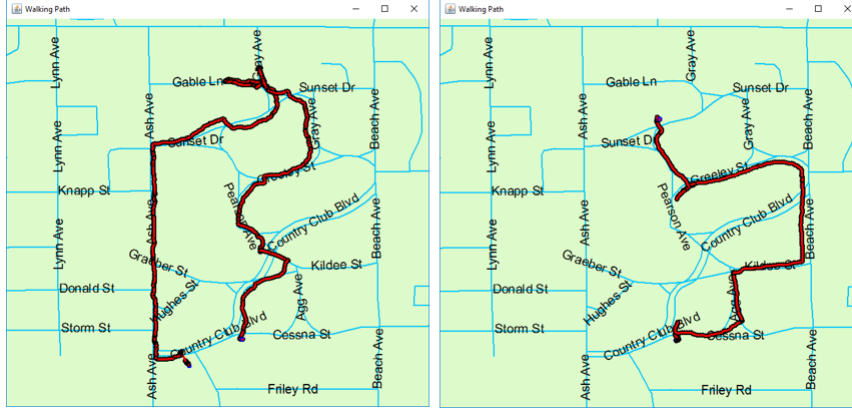


Figure 6.11 Walking Paths in Phase 1 (left) and 2 (right) of a Participant.

path. We found out that the optimal path of phase 1 was 300 meters longer than the optimal path of phase 2. Therefore, we could not compare times of both phases directly. In order to make them comparable, we used a time normalization for every participant's time in phase 1. We did two techniques of normalization; normalization by overall average and normalization by individual average.

6.4.1.1.1 Normalization by Overall Average

To normalize the time that a participant used in phase 1 by the overall average, we multiplied the time in phase 1 with the ratio of the average distance of phase 2 to the average distance of phase 1. The following formula was applied:

$$T'_1(i) = T_1(i) \left(\frac{\bar{D}_2}{\bar{D}_1} \right) \quad (6.1)$$

We take a look at the definition of the formula (Equation 6.1). $T'_1(i)$ was a normalized time that a participant i used in phase 1. $T_1(i)$ was an actual time that a participant i used

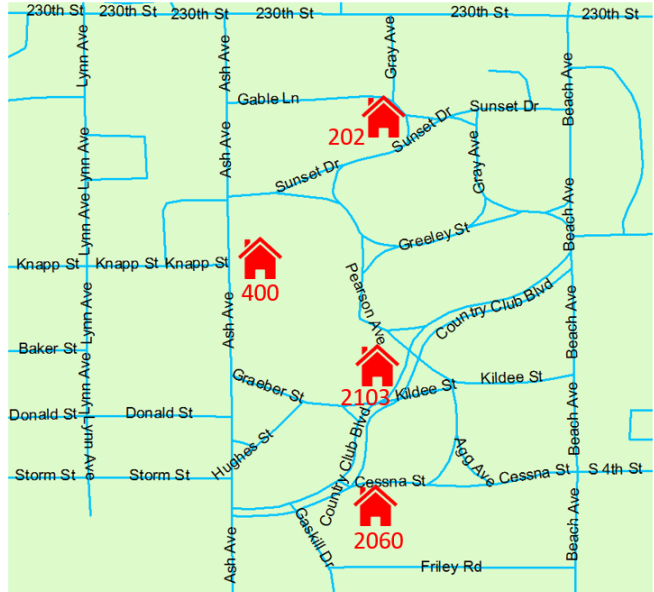


Figure 6.12 Addresses to be verified in Phase 1.

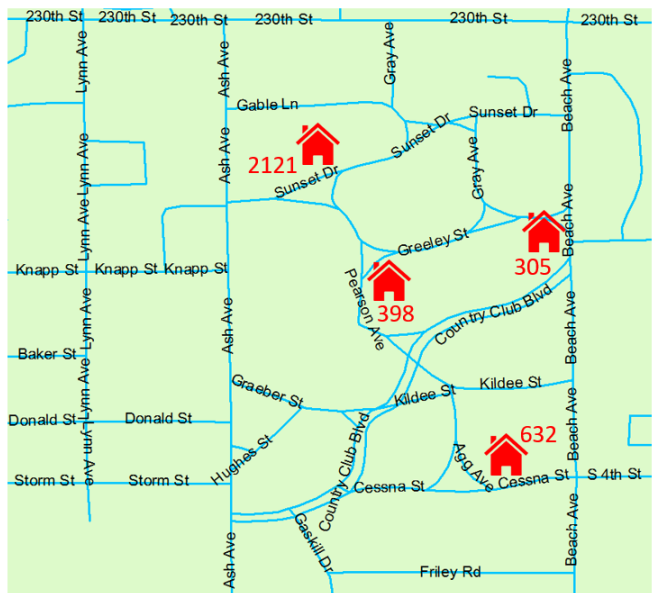


Figure 6.13 Addresses to be verified in Phase 2.

in phase 1. \bar{D}_2 was an average distance that every participant walked in phase 2. \bar{D}_1 was an average distance that every participant walked in phase 1.

6.4.1.1.2 Normalization by Individual Average

To normalize the time that a participant used in phase 1 by the individual average, we multiplied the time in phase 1 with the ratio of the distance that the participant walked in phase 2 to the distance that the participant walked in phase 1. The following formula was applied:

$$T'_1(i) = T_1(i) \left(\frac{D_2(i)}{D_1(i)} \right) \quad (6.2)$$

We take a look at the definition of the formula (Equation 6.2). $T'_1(i)$ was a normalized time that a participant i used in phase 1. $T_1(i)$ was an actual time that a participant i used in phase 1. $D_2(i)$ was the distance that a participant i walked in phase 2. $D_1(i)$ was the distance that a participant i walked in phase 1.

Table 6.3 shows the example results of the actual time that a set of participants used in phase 1: column 3 shows the time that was normalized by the overall average, and column 4 shows the time that was normalized by the individual average.

6.4.1.2 Tests of Hypotheses

We set up research questions, hypotheses, and did a data analysis according to the corresponding hypothesis. For each research question, we addressed two hypotheses: a null hypothesis and an alternative hypothesis, the results of the hypothesis test, and provided a box plot. For the test of unpaired continuous data, we used the Mann-Whitney hypothesis

Table 6.3 Example Results of Actual Time and Normalized Times.

Participant	Time Phase 1	Time Phase 1 (Overall AVG)	Time Phase 1 (Individual AVG)
1	44.48	31.98941479	26.52595175
2	43.21	31.07604796	24.14342747
3	33.19	23.86979939	25.38761575
4	43.02	30.93940253	33.90878923
5	37.76	27.15648162	24.40565554
8	27.12	19.50433744	24.06366597
9	37.94	27.28593519	18.14625418
10	33.52	24.10713093	29.07887566
11	24.72	17.77828988	22.59910787
12	35.64	25.63180628	21.79370829
13	15.7	11.29122779	14.50024944
14	42.84	30.80994896	42.54280038
15	38.88	27.96197049	21.40777729

test. For the test of paired data, we used the Paired T-Test hypothesis test. An objective of both a Mann-Whitney test and a Paired T-Test were to test whether two sets of data were significantly different or not.

6.4.1.2.1 Research Question 1: For participants who were assigned treatment 0, was there a significant difference between time in phase 1 and phase 2?

To answer this question, we set up following hypotheses:

H_0 : For treatment 0, times in both phases are **NOT** significantly different.

H_a : For treatment 0, times in both phases are significantly different.

We did two hypothesis tests to test two pairs of data sets. The first pair consisted of a normalized, by overall average, time used in phase 1 and an actual time used in phase 2 of every participant who was assigned the treatment 0. The second pair consisted of a

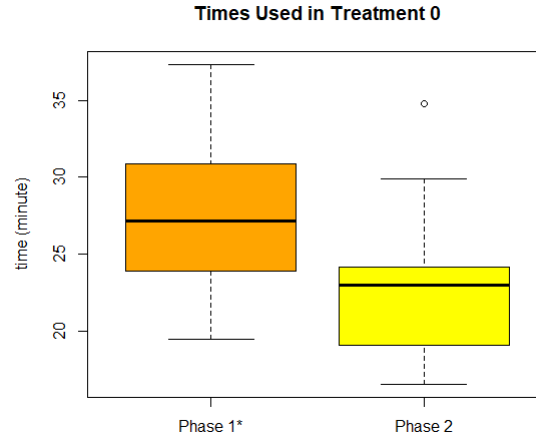


Figure 6.14 A Box Plot Illustrated a Time in Phase 1 (Normalized by Overall Average) and Phase 2 for Treatment 0.

normalized, by individual average, time used in phase 1 and an actual time used in phase 2 of every participant who was assigned the treatment 0. Since the time in phase 1 and phase 2 were paired data, we used a Paired T-Test.

After we ran a hypothesis test on the first pair, we got a p -value of 0.001806. Since the p -value was smaller than the significance level (α) of 0.05, we rejected H_0 . We then ran a hypothesis test on the second pair. We got a p -value of 0.000439. Since the p -value was smaller than the significance level (α) of 0.05, we also rejected H_0 .

Figure 6.14 shows a box plot of a time in phase 1 (normalized by the overall average) and an actual time in phase 2 of every participant who was assigned the treatment 0. We observed that a participant used shorter time in phase 2 (adaptive UI) than phase 1 (traditional UI).

6.4.1.2.2 Research Question 2: For participants who were assigned treatment 1, was there a significant difference between time in phase 1 and phase 2?

To answer this question, we set up following hypotheses:

H_0 : For treatment 1, times in both phases are **NOT** significantly different.

H_a : For treatment 1, times in both phases are significantly different.

We did two hypothesis tests to test two pairs of data sets. The first pair consisted of a normalized, by overall average, time used in phase 1 and an actual time used in phase 2 of every participant who was assigned the treatment 1. The second pair consisted of a normalized, by individual average, time used in phase 1 and an actual time used in phase 2 of every participant who was assigned the treatment 1. Since the time in phase 1 and phase 2 were paired data, we used a Paired T-Test.

After we ran a hypothesis test on the first pair, we got a p -value of 0.00508. Since the p -value was smaller than the significance level (α) of 0.05, we rejected H_0 . We then ran a hypothesis test on the second pair. We got a p -value of 0.0001746. Since the p -value was smaller than the significance level (α) of 0.05, we also rejected H_0 .

Figure 6.15 shows a box plot of a time in phase 1 (normalized by the overall average) and an actual time in phase 2 of every participant who was assigned the treatment 1. Although there is an overlap in Figure 6.15, we observed that a participant used shorter time in phase 1 (adaptive UI) than phase 2 (traditional UI).

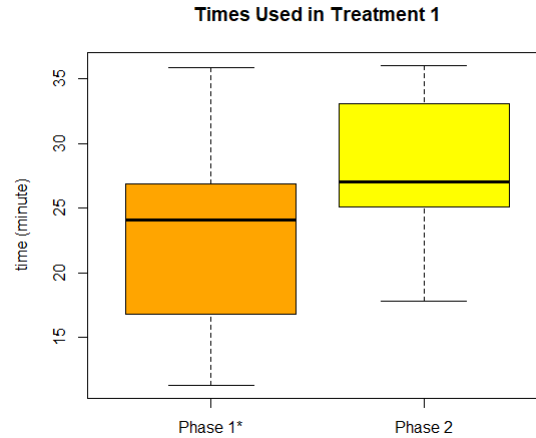


Figure 6.15 A Box plot Illustrated a Time in Phase 1 (Normalized by Overall Average) and Phase 2 for Treatment 1.

6.4.1.2.3 Research Question 3: Focusing on phase 1, was there a significant difference between time used by participants of treatment 0 and treatment 1?

To answer this question, we set up following hypotheses:

H_0 : For phase 1, times in both treatments are **NOT** significantly different.

H_a : For phase 1, times in both treatments are significantly different.

We did a hypothesis test to test a group of two data sets. The group consisted of a normalized, by overall average, time in phase 1 used by participant from treatment 0 and 1. Since the times from treatment 0 and 1 were not paired data and they are continuous variables, we used a Mann-Whitney test.

After we ran a hypothesis test, we got a suggestive p -value of 0.0675. We failed to reject H_0 . Even though we failed to reject the null hypothesis, the box plot (Figure 6.16) shows overlap of the two sets. However, we can also observe that a participant who was assigned

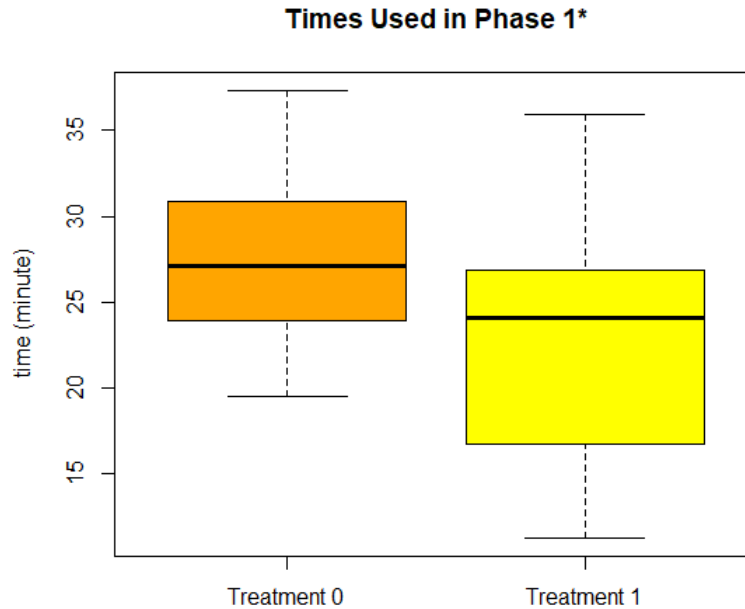


Figure 6.16 A Box Plot Illustrated a Time in Phase 1 (Normalized by Overall Average) for Treatment 0 and 1.

the treatment 0 (traditional UI) tended to use more time than treatment 1 (adaptive UI) in phase 1.

6.4.1.2.4 Research Question 4: Focusing on phase 2, was there a significant difference between time used by participants of treatment 0 and treatment 1?

To answer this question, we set up following hypotheses:

H_0 : For phase 2, times in both treatments are **NOT** significantly different.

H_a : For phase 2, times in both treatments are significantly different.

We did a hypothesis test to test two sets of data. The first set was a time in phase

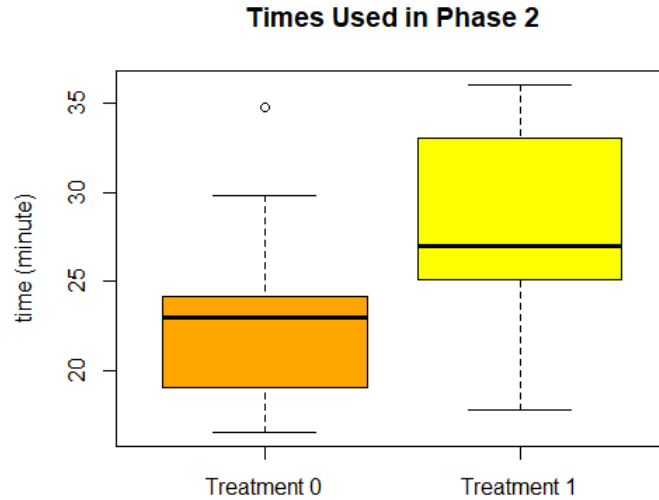


Figure 6.17 A Box Plot Illustrated a Time in Phase 2 for Treatment 0 and 1.

2 used by participant from treatment 0. The second set was a time in phase 2 used by participant from treatment 1. We used a Mann-Whitney test with the same reason as the third research question.

After we ran a hypothesis test, we got a p -value of 0.006568. Since the p -value was smaller than the significance level (α) of 0.05, we rejected H_0 . Figure 6.17 shows a box plot of a time in phase 2 for both treatment 0 and 1. We found no overlap in Figure 6.17. We also observed that a participant who was assigned treatment 0 (Adapted UI) used shorter time than treatment 1 (traditional UI) in phase 2.

From six hypothesis tests, five hypothesis tests show a significant difference between times that were spent when using the traditional UI and the adaptive UI. One hypothesis test shows that the difference was suggestively significant. The box plots (Figure 6.14 - 6.17) shows that participants, regardless of the assigned treatment or phase, used shorter time when they used the adaptive UI than the traditional UI.

6.4.2 Analysis of Prediction Methods

We previously mentioned the design and implementation of the prediction module in Section 6.2.3. In this section, we take a look at the accuracy of the prediction module.

From thirty participants, the prediction module correctly predicted twenty three participants. The accuracy rate was 77%. From seven mispredicted participants, there were three high VZ participants that were predicted as low VZ. The other four mispredicted participants were low VZ that were predicted as high VZ. Since this was a real-time prediction, a prediction was made based on the supplied training set at that time. The supplied training set sometimes was unbalanced, which meant that it contained more data records of high VZ than low VZ (or more low than high). The unbalanced training set potentially affected the accuracy of the prediction. Not only unbalanced training set, a small training set (three training instances from the pilot test) also reduced the prediction accuracy.

Therefore, we decided to re-predict every participant again when we were done with all thirty participants. We re-predicted using the same method we did in the prediction module, i.e., we used the same classifier and the same set of parameters. The difference was that we re-predicted all thirty participants at the same time. After re-prediction, there were five participants that were mispredicted. The new accuracy rate then became 83%, which was comparable to Batinov's study [2]. From five mispredicted participants, all of

them were low VZ who were predicted as high VZ. What this tells us is that having a large training set before starting the study would likely provide better results.

However, one might see our prediction method was preferable for a training purpose rather than an address verification task because we had a time and a distance as the parameter of the prediction method. Both a time and a distance were a neighborhood-dependent parameter. Hence, we re-predicted again by removing both the time and the distance from the set of parameters. Without both parameters, six participants were mispredicted. All of them were low VZ who were predicted as high VZ. The new accuracy rate dropped to 80%.

Furthermore, we also would like to apply the method from Batinov's study [2] to our data set of thirty participants. So we re-predicted again using the same set of parameters as what Batinov used. With Batinov's method, five participants were mispredicted. All of them were low VZ again. The accuracy rate of this attempt was 83%.

Table 6.4 shows the complete prediction results of thirty participants using different prediction methods. The first column contains a participant's ID. The second column is the participant's actual VZ according to the result of Paper Folding test [20]. The third column is the result of the prediction module of Study III software. The fourth column is the result of the post-study re-prediction using the same method as the prediction module did. The fifth column is the result of post-study re-prediction using the same method as the prediction module did but removing a time and a distance from the set of parameters. Finally, the last column is the result of post-study prediction using Batinov's method of prediction [2].

Table 6.4 Complete Prediction Results of Different Methods.

Participant	Tested VZ	Predicted VZ	Re-Predict	Re-Predict (no time nor distance)	Batinov's
1	Low	Low	High	High	High
2	Low	Low	Low	Low	Low
3	High	Low	High	High	High
4	High	Low	High	High	High
5	Low	Low	High	High	Low
8	High	High	High	High	High
9	High	High	High	High	High
10	Low	Low	Low	Low	High
11	High	High	High	High	High
12	High	High	High	High	High
13	High	High	High	High	High
14	High	High	High	High	High
15	Low	Low	High	High	Low
16	High	High	High	High	High
17	High	High	High	High	High
18	High	High	High	High	High
19	High	High	High	High	High
20	High	High	High	High	High
21	High	Low	High	High	High
22	Low	High	High	High	High
23	High	High	High	High	High
24	Low	High	Low	Low	Low
25	High	High	High	High	High
26	Low	High	Low	Low	High
27	Low	High	High	High	High
28	High	High	High	High	High
29	High	High	High	High	High
30	High	High	High	High	High
31	High	High	High	High	High
32	Low	Low	Low	High	Low

The main objectives of Study III, the improvement shown when using the adaptive UI and the analysis of the prediction of VZ level, have already been discussed. In the next sections, we take a look at the additional analyses of Study III results.

6.4.3 Additional Analysis: Market Basket Analysis on Participant’s Behavior

After we were done with thirty participants, we took a look at the log of each participant. We noticed that there were some features that might be interesting if we did a further analysis on them. These features might be a link to the hidden pattern that could be useful for the area of location-based systems and individual difference study. We then extracted those features from each participant’s log. There were seven features that were extracted, which were a participant’s VZ, an assigned treatment, the first address that was selected to verify in phase 1, the last address that was selected to verify in phase 1, the first address that was selected to verify in phase 2, the last address that was selected to verify in phase 2, and a street that a participant used as a backbone. A street that a participant used as a backbone was a primary route that a participant mostly used to travel from the current address to the next address when he completed the current one. There were two possible values for a backbone street, which were “Ash-or-Beach” and “street-inside-neighborhood”. Ash-or-Beach referred to the Ash avenue and the Beach avenue, which were a straight north-south street located at the west and east of the neighborhood, respectively. Street-inside-neighborhood referred to the streets that were located inside the neighborhood. These streets were mostly not a straight line. A good example of this type of street was Country Club Blvd. We have already shown a map of the neighborhood with all streets in [Figure 4.1](#).

For each participant, the participant’s VZ could be either low or high. The assigned treatment was either 0 (traditional UI came first) or 1 (adaptive UI came first). The first address that was selected to verify in phase 1 could be any address in the address list of

phase 1. The address list of phase 1 consisted of 400 Ash, 202 Gray, 2103 Country Club, and 2060 Cessna. Similarly, the last address that was selected in phase 1 could also be any address the address list of phase 1. Figure 6.12 shows every address needed to be verified in phase 1. The first and last addresses that were selected to verify in phase 2 could be any address in the address list of phase 2, which consisted of 2121 Sunset, 398 Pearson, 305 Beach, and 632 Agg. Figure 6.13 shows every address needed to be verified in phase 2.

We did a Market Basket analysis using Weka [29]. The Apriori was selected as an algorithm for the association. For the parameters, we set the minimum support to be 33% and the minimum confidence to be 80%. The minimum support of 33% was chosen because we would like to obtain rules that can classify the VZ level of a participant. Since we had twenty high VZ participants (66%) and ten low VZ participants (33%), 33% was the maximum number of occurrences of the instance of low VZ (all ten participants) that made rules persuasive. Figure 6.18 shows how we configured the parameters of the associator.

After we ran the associator, it generated 149 rules. We selected some interesting rules that had VZ involved to discuss. We also provided the complete set of rules in Appendix B.

6.4.3.1 Rule 1

The first rule that we selected was:

$$\mathbf{last_address_phase2 = 632\ Agg \Rightarrow vz = high}$$

Rule 1 was interpreted as if the last address of phase 2 that a participant verified was 632 Agg, then, then participant was high VZ. The support of Rule 1 was 36%. The confidence of Rule 1 was 100%. Rule 1 was reasonable because 632 Agg was the last address to be verified in phase 2 on the optimal route. Note that the optimal route was designed

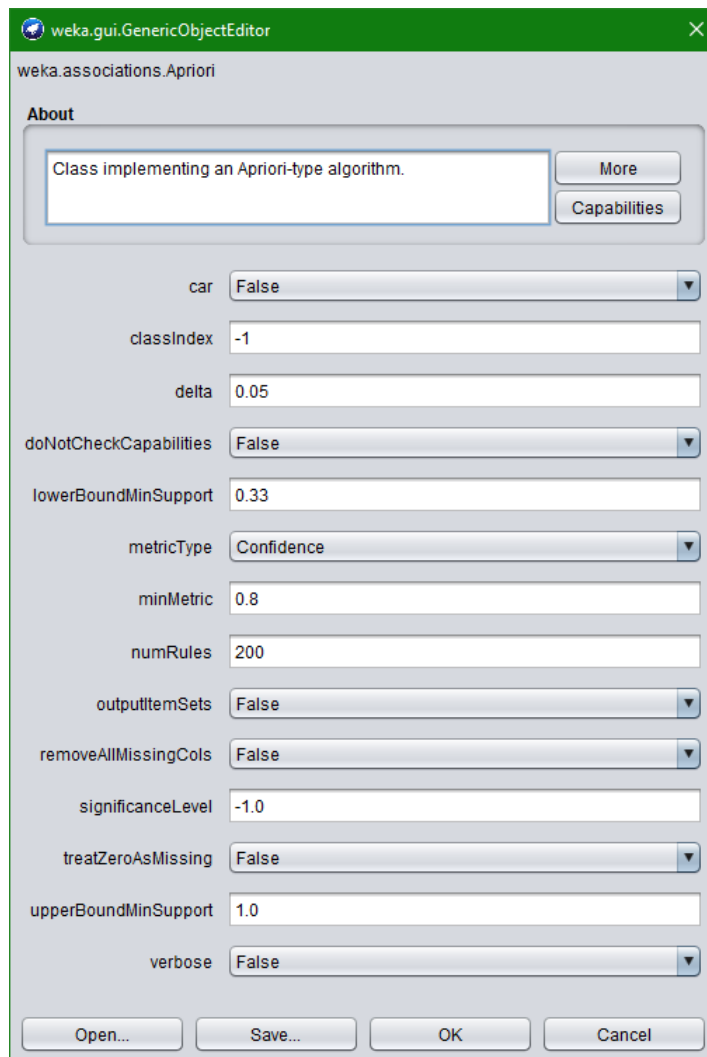


Figure 6.18 The Set of Parameters of the Associator for Market Basket Analysis.

and confirmed by our research group. The optimal route required the minimum distance to complete both phase 1 and 2. The minimum distance was 2.389865 kilometers. The sequence of addresses to be verified in the optimal route was 2060 Cessna, 2103 Country Club, 202 Gray, 400 Ash, 212 Sunset, 398 Pearson, 305 Beach, and 632 Agg.

6.4.3.2 Rule 2

The second rule that we selected was:

first_address_phase1 = 2060 Cessna \wedge backbone_street = street-inside-neighborhood

↓

vz = high

Rule 2 was interpreted as if the first address that a participant verified in phase 1 was 2060 Cessna **AND** a participant used streets inside the neighborhood as a backbone, then, the participant was high VZ. The support of Rule 2 was 33%. The confidence of Rule 2 was 100%. Using streets inside the neighborhood as a backbone was a behavior of high VZ participants that corresponded to Batinov's study [2].

6.4.3.3 Rule 3

The third rule that we selected was:

backbone_street = street-inside-neighborhood \Rightarrow vz = high

Rule 3 was interpreted as if a participant used streets inside the neighborhood as a backbone, then, the participant was high VZ. The support of Rule 3 was 43%. The confidence of Rule 3 was 92%.

With the selected three rules, we found some features that could anticipate if a participant was high VZ. However, we could not tell much about low VZ due to the small number of low VZ participants in Study III. In the next section, we discuss the last additional analysis of Study III.

6.4.4 Additional Analysis: Mispredicted Participants

As we mentioned earlier, we implemented and deployed the prediction module in the software of Study III. The third column in Table 6.4 shows the result of the prediction module. From Table 6.4, there were seven participants that were mispredicted. We take a look on the mispredicted participants in this section.

From seven participants who were mispredicted, three tested high VZ and four tested low VZ. Therefore, in phase 3, three high VZ participants were assigned the adaptive UI for low VZ and four low VZ participants were assigned the adaptive UI for high VZ. We drew box plots to observe times and distances that every participant used in phase 3. Figure 6.19 shows box plots of the times used by the four groups of participants: high VZ who were correctly predicted, low VZ who were correctly predicted, high VZ who were mispredicted, and low VZ who were mispredicted, respectively. We observed that the box plot of low VZ participants who were correctly predicted (yellow) had similar distribution as the box plot of high VZ participants who were mispredicted (blue). This distribution led us to the idea that there potentially might be more than just two groups of participants (high/low) when it came to the location-based task using a location-based software. It raised the possibility that a low VZ participant who highly focused on the task could perform better than a high VZ who did not focus. Note that Batinov et al. [3] had speculated that this might occur based on earlier studies.

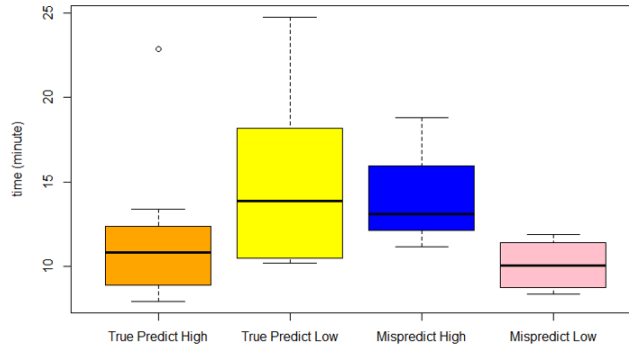


Figure 6.19 Time Used in Phase 3 by Different Groups of Participants.

The box plot of high VZ participants who were correctly predicted (orange) also had similar distribution as the box plot of low VZ participants who were mispredicted (pink). However, we did not calculate the p -value to confirm that the distributions were statistically the same because the small number of mispredicted participants in Study III was not appropriate for any hypothesis test.

Figure 6.20 shows box plots of the distances used by four groups of participants; high VZ who were correctly predicted, low VZ who were correctly predicted, high VZ who were mispredicted, and low VZ who were mispredicted, respectively. The box plots show the distributions of distances were similar to the distributions of times for the same four groups of participants.

6.5 Conclusions

We did Study III to verify our claim that the adaptive UI was the answer to improve the user's performance of doing the address verification task using our location-based software. Thirty participants participated in Study III. The address verification task consisted

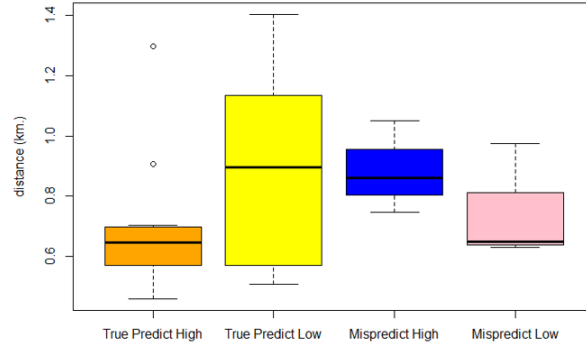


Figure 6.20 Distance Used in Phase 3 by Different Groups of Participants.

of three phases; phase 1, phase 2, and phase 3. A participant was randomly assigned either treatment 0 or 1. For treatment 0, a participant was assigned the traditional UI to work in phase 1 and the adaptive UI in phase 2. For treatment 1, a participant was assigned the adaptive UI to work in phase 1 and the traditional UI in phase 2. A participant, regardless of the assigned treatment, was assigned the adaptive UI based on the prediction result (high VZ or low VZ) in phase 3. The prediction was made by the prediction module, which was an extension of Batinov's study [2].

The results of five hypothesis tests show that participants, regardless of their VZ level, significantly did better by spending less time when they verified addresses using the adaptive UI than the traditional UI. The box plots in Figure 6.14 and 6.15 show that there was a small overlap between times of using the adaptive UI and the traditional UI. The box plot in Figure 6.17 shows no overlap between the two UIs, which confirms the improvement of the adaptive UI over the traditional UI.

The prediction module had a real-time accuracy rate of 77%. However, the accuracy rate was affected by the unbalanced number of high VZ participants and low VZ participants in

the training set, as well as the small size (3) of the training set at the beginning. We did a post-study re-prediction again and the rate became 83%, which corresponded to Batinov's study [2]. For mispredicted participants, we found that most mispredicted participants had some similar characteristics/metrics to the group that they were predicted correctly (Figure 6.19 and 6.20).

Additionally, we also extracted some interesting rules from a Market Basket analysis of participants' behavior. One example rule was that if a participant used the streets inside the neighborhood as a backbone, that participant might be high VZ. This statement was also true in Batinov's model of high VZ users [2].

A limitation that we confronted in Study III was that the number of high VZ participants and low VZ participants was not balanced enough.

In Chapter 7, we provide a discussion, a conclusion, and a contribution of the dissertation.

CHAPTER 7. CONCLUSIONS AND CONTRIBUTIONS

The ultimate objective of this dissertation was to develop the adaptive location-based software for the address verification task that could adapt the UI based on a user's VZ level in real time. The software should be able to adjust the UI for low VZ users, which consisted of the regular UI plus the additional useful features for low VZ users, when it could predict that the current user was low VZ. Similarly, when the software predicted that the current user was high VZ, it adjusted the UI for high VZ users.

In order to develop the adaptive UI software, we did three studies; Study I, Study II, and Study III. The objective of Study I (Chapter 4) was used to test which software features increased the low VZ participants' performance in the address verification task and which software features decreased the performance. Not only focusing on low VZ participants, Study I was also determined good features and bad features for high VZ participants. There were three software features that were tested; GPS, Object-Indexing, and Mini Map. The GPS allowed participants to see their current location on the map. The Mini Map showed the map of the full neighborhood. Object-Indexing was the deployment of our approach (Chapter 3) to empower a participant with an ability to directly access a map component, particularly, a street. Each participant was randomly assigned two features to use during the field study.

The result from Study I revealed some interesting relationships between VZ and the software features. The GPS increased the performance of low VZ participants whereas it

decreased the performance of high VZ participants. Object-Indexing reduced the performance of low VZ participants. The Mini Map was the only feature that increased the performance of high VZ participants.

For Study II (Chapter 5), the objective was to test how well participants performed with the larger map size on the universal UI. The only difference between Study II and Study I was the software. The software used in Study II came with the large map size. All three software features (GPS, Object-Indexing, & Mini Map) were available for every participant during the address verification task. We found from the result of Study II that a universal UI and a larger size of a map were not a solution to improve a user's performance in the address verification task. Although high VZ participants tended to have better performance than low VZ participants when using the universal UI, this was common in any software.

For Study III (Chapter 6), we developed the adaptive UI software that could adapt the UI based on the VZ level of a participant. The field study in Study III was divided into three phases. Each participant was asked to verify four addresses in phase 1, four addresses in phase 2, and two addresses in phase 3. A participant was randomly assigned either treatment 0 or treatment 1. If a participant was assigned treatment 0, the software used a traditional UI for the participant to use in phase 1, the adaptive UI based on the score of a Paper Folding test in phase 2, and the adaptive UI based on the prediction result in phase 3. If a participant was assigned treatment 1, the software used an adaptive UI based on the score of a Paper Folding test for the participant to use in phase 1, the traditional UI in phase 2, and the adaptive UI based on the prediction result in phase 3.

Our analysis of the result of Study III demonstrates that the participants, regardless of VZ level, treatment, or phase, had better performance when they verified addresses using the adaptive software rather than the traditional software. A metric that served as an

evidence was time. Given that the distances required to complete the task in phase 1 and phase 2 were equivalent due to the normalization process, participants used shorter amount of time when they used the adaptive UI software.

We also reported the accuracy of the prediction module that was implemented in the software of Study III. The accuracy of the prediction module was 77%, which was potentially enforced by the unbalanced training set and the small number of data records in the training set at the beginning of Study III. The assumption that those two factors really affected the prediction accuracy was confirmed when we got 83% as the accuracy of re-predicting all participants again at the same time after Study III was done. The main result here is that it is important to start a user with a prediction algorithm that has some pretraining.

It turned out to be that Study III yielded interesting concepts beyond just the improvement of performance with the adaptive UI and the prediction of VZ. The last two additional contributions of this dissertation were the set of association rules that was generated from the participants' behavior and the report of the performance of mispredicted participants in Study III. We applied a Market Basket analysis to generate association rules related to participant's behavior when they did address verification task. Some rules were useful as they can imply a participant's VZ level from the participant's behavior.

For the mispredicted participants, we found that participants who were mispredicted had distributions of metrics, particularly, the times and the distances, similar to the group that they were predicted as. This confirmed that when it came to the address verification task using a location-based software, there can be more than two classes (low VZ & high VZ) that a user can be classified to.

BIBLIOGRAPHY

- [1] Batinov, G., Rusch, M., Meng, T., Whitney, K., Patanasakpinyo, T., Miller, L., and Nusser, S. (2015). Understanding map operations in location-based surveys. In *Proceedings of the Eighth International Conference on Advances in Computer-Human Interactions (ACHI2015)*, pages 144–149, Lisbon, Portugal. International Academy, Research, and Industry Association.
- [2] Batinov, G. I. (2017). *Computer detection of spatial visualization in a location-based task*. Graduate Theses and Dissertations: <http://lib.dr.iastate.edu/etd/15103>.
- [3] Batinov, G. I., Whitney, K., Miller, L., Nusser, S. M., Stanfill, B., and Ashenfelter, K. T. (2013). Evaluating the impact of spatial ability in virtual and real world environments. In *Proceedings of the Sixth International Conference on Advances in Computer-Human Interactions*, pages 274–279, Nice, France. International Academy, Research, and Industry Association.
- [4] Benyon, D. (1993). Accommodating individual differences through an adaptive user interface. *Human Factors in Information Technology*, 10:149–149.
- [5] Benyon, D., Innocent, P., and Murray, D. (1987). *System adaptivity and the modelling of stereotypes*. National Physical Laboratory, Division of Information Technology and Computing Teddington.
- [6] Benyon, D. and Murray, D. (2000). Special issue on intelligent interface technology: editor’s introduction. *Interacting with Computers*, 12(4):315–322.

- [7] Berger, C., Lu, C., Belzer, S., Voss, B., and Gabel, D. (1994). Handbook of research on science teaching and learning. *Handbook on research on science teaching and learning*.
- [8] Bourke, P. (1987). Determining if a point lies on the interior of a polygon. *Internet: <http://local.wasp.uwa.edu.au/~pbourke/geometry/insidepoly>*.
- [9] Bureau, U. C. (2013). Tiger/line shapefiles.
- [10] Büring, T., Gerken, J., and Reiterer, H. (2006). Usability of overview-supported zooming on small screens with regard to individual differences in spatial ability. In *Proceedings of the working conference on Advanced visual interfaces*, pages 233–240, Venezia, Italy. ACM.
- [11] Carroll, J. B. (1993). *Human cognitive abilities: A survey of factor-analytic studies*. Cambridge University Press.
- [12] Chellappan, S. K. (2012). *An object-oriented approach to maps*. Graduate Theses and Dissertations. 12293. <http://lib.dr.iastate.edu/etd/12293>.
- [13] Chellappan, S. K. and Miller, L. (2014). An object oriented approach to dynamic survey unit maps. In *Proceedings of the 31st INTERNATIONAL CONFERENCE ON COMPUTERS AND THEIR APPLICATIONS (CATA2014)*, pages 273–278, Las Vegas, Nevada. International Society for Computers and their Applications.
- [14] Craik, F. I. and Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior*, 11(6):671–684.
- [15] Cutmore, T. R., Hine, T. J., Maberly, K. J., Langford, N. M., and Hawgood, G. (2000). Cognitive and gender factors influencing navigation in a virtual environment. *International Journal of Human-Computer Studies*, 53(2):223–249.
- [16] Dahlbäck, N., Höök, K., and Sjölander, M. (1996). Spatial cognition in the mind and in the world: The case of hypermedia navigation. In *The Eighteenth Annual Meeting of the*

- Cognitive Science Society*, pages 195–200, University of California, San Diego, California. Lawrence Erlbaum Associates.
- [17] De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., and Vicente-Lopez, E. (2014). Using personalization to improve xml retrieval. *Knowledge and Data Engineering, IEEE Transactions on*, 26(5):1280–1292.
- [18] Egan, D. E. (1988). Dealing with diversity: Individual differences in human-computer interaction. In *Proceedings of Human Factors in Computing Systems (CHI'88)*, pages 79–81, Washington, DC. Association for Computing Machinery.
- [19] Egan, D. E. and Gomez, L. M. (1985). Assaying, isolating, and accommodating individual differences in learning a complex skill. *Individual differences in cognition*, 2:173–217.
- [20] Ekstrom, R. B., French, J. W., Harman, H. H., and Dermen, D. (1976). *Manual for kit of factor-referenced cognitive tests*. Princeton, NJ: Educational testing service.
- [21] ESRI (1998). *Shapefile technical description*. <https://www.esri.com/library/whitepapers>.
- [22] Findlater, L. and McGrenere, J. (2004). A comparison of static, adaptive, and adaptable menus. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 89–96, Vienna, Austria. ACM.
- [23] Fischer, G. (2001). User modeling in human-computer interaction. *User modeling and user-adapted interaction*, 11(1-2):65–86.
- [24] Gajos, K. Z., Czerwinski, M., Tan, D. S., and Weld, D. S. (2006). Exploring the design space for adaptive graphical user interfaces. In *Proceedings of the working conference on Advanced visual interfaces*, pages 201–208, Venice, Italy. ACM.
- [25] Galea, L. A. and Kimura, D. (1993). Sex differences in route-learning. *Personality and individual differences*, 14(1):53–65.

- [26] Gavurin, E. I. (1967). Anagram solving and spatial aptitude. *The Journal of psychology*, 65(1):65–68.
- [27] Geary, D. C., Saults, S. J., Liu, F., and Hoard, M. K. (2000). Sex differences in spatial cognition, computational fluency, and arithmetical reasoning. *Journal of Experimental child psychology*, 77(4):337–353.
- [28] Golledge, R. G. and Gärling, T. (2004). Cognitive maps and urban travel. In *Handbook of transport geography and spatial systems*, pages 501–512. Emerald Group Publishing Limited.
- [29] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- [30] Halligan, P. W., Fink, G. R., Marshall, J. C., and Vallar, G. (2003). Spatial cognition: evidence from visual neglect. *Trends in cognitive sciences*, 7(3):125–133.
- [31] Hegarty, M., Montello, D. R., Richardson, A. E., Ishikawa, T., and Lovelace, K. (2006). Spatial abilities at different scales: Individual differences in aptitude-test performance and spatial-layout learning. *Intelligence*, 34(2):151–176.
- [32] Hormann, K. and Agathos, A. (2001). The point in polygon problem for arbitrary polygons. *Computational Geometry*, 20(3):131–144.
- [33] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 256–265, Madison, Wisconsin. Morgan Kaufmann Publishers Inc.
- [34] Hyde, J. S. (1981). How large are cognitive gender differences? a meta-analysis using d and d^2 . *American Psychologist*, 36(8):892.

- [35] Innocent, P. (1982). Towards self-adaptive interface systems. *International Journal of Man-Machine Studies*, 16(3):287–299.
- [36] Ishikawa, T., Fujiwara, H., Imai, O., and Okabe, A. (2008). Wayfinding with a gps-based mobile navigation system: A comparison with maps and direct experience. *Journal of Environmental Psychology*, 28(1):74–82.
- [37] Ishikawa, T. and Montello, D. R. (2006). Spatial knowledge acquisition from direct experience in the environment: Individual differences in the development of metric knowledge and the integration of separately learned places. *Cognitive psychology*, 52(2):93–129.
- [38] Kaufman, S. B. (2007). Sex differences in mental rotation and spatial visualization ability: Can they be accounted for by differences in working memory capacity? *Intelligence*, 35(3):211–223.
- [39] Keates, S., Clarkson, P. J., and Robinson, P. (2002). Developing a practical inclusive interface design approach. *Interacting with computers*, 14(4):271–299.
- [40] Kim, Y. O. and Penn, A. (2004). Linking the spatial syntax of cognitive maps to the spatial syntax of the environment. *Environment and Behavior*, 36(4):483–504.
- [41] Klippel, A., Tappe, H., and Habel, C. (2002). Pictorial representations of routes: Chunking route segments during comprehension. In *International Conference on Spatial Cognition*, pages 11–33, Rome, Italy. Springer.
- [42] Knutov, E., De Bra, P., and Pechenizkiy, M. (2009). Ah 12 years later: a comprehensive survey of adaptive hypermedia methods and techniques. *New Review of Hypermedia and Multimedia*, 15(1):5–38.
- [43] Kotzyba, M., Siegert, I., Gossen, T., Nrnberger, A., and Wendemuth, A. (2015). Exploratory voice-controlled search for young users : Challenges and potential benefits. *Kognitive Systeme*, 3:25–27.

- [44] Kozlowski, L. T. and Bryant, K. J. (1977). Sense of direction, spatial orientation, and cognitive maps. *Journal of Experimental Psychology: human perception and performance*, 3(4):590.
- [45] Kühme, T. (1993). *Adaptive Action Prompting: A Complementary Aid to Support Task-Oriented Interaction in Explorative User Interfaces*. Technical Report. Georgia Institute of Technology.
- [46] Kuipers, B., Modayil, J., Beeson, P., MacMahon, M., and Savelli, F. (2004). Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4845–4851, New Orleans, Louisiana. IEEE.
- [47] Kuipers, J. B. et al. (1999). *Quaternions and rotation sequences*, volume 66. Princeton university press Princeton.
- [48] Kules, B. (2000). User modeling for adaptive and adaptable software systems. In *ACM Conference on Universal Usability*, pages 16–17, Washington, DC. ACM.
- [49] Kulhavy, R. W. and Stock, W. A. (1996). How cognitive maps are learned and remembered. *Annals of the Association of American Geographers*, 86(1):123–145.
- [50] Lanca, M. (1998). Three-dimensional representations of contour maps. *Contemporary educational psychology*, 23(1):22–41.
- [51] Landau, B. (2002). Spatial cognition. *Encyclopedia of the human brain*, 4:395–418.
- [52] Langley, P. (1997). Machine learning for adaptive user interfaces. In *KI-97: Advances in artificial intelligence*, pages 53–62, Freiburg, Germany. Springer.
- [53] Law, C. M., Yi, J. S., Choi, Y. S., and Jacko, J. A. (2007). Unresolved problems in accessibility and universal design guidelines. *Ergonomics in Design: The Quarterly of Human Factors Applications*, 15(3):7–11.

- [54] Liben, L. S. and Downs, R. M. (1993). Understanding person-space-map relations: Cartographic and developmental perspectives. *Developmental Psychology*, 29(4):739.
- [55] Liben, L. S., Kastens, K. A., and Stevenson, L. M. (2002). Real-world knowledge through real-world maps: A developmental guide for navigating the educational terrain. *Developmental Review*, 22(2):267–322.
- [56] Linn, M. C. and Petersen, A. C. (1985). Emergence and characterization of sex differences in spatial ability: A meta-analysis. *Child Development*, 56(1):1479–1498.
- [57] Liu, J., Wong, C. K., and Hui, K. K. (2003). An adaptive user interface based on personalized learning. *IEEE Intelligent Systems*, (2):52–57.
- [58] Lohman, D. F. (1996). Spatial ability and g. *Human abilities: Their nature and measurement*, 97:116.
- [59] Lohman, D. F., Pellegrino, J. W., Alderton, D. L., and Regian, J. (1987). Dimensions and components of individual differences in spatial abilities. *Intelligence and cognition: Contemporary frames of reference*, pages 253–312.
- [60] Marshall, J. C. and Fink, G. R. (2001). Spatial cognition: where we were and where we are. *Neuroimage*, 14(1):S2–S7.
- [61] Masters, M. S. and Sanders, B. (1993). Is the gender difference in mental rotation disappearing? *Behavior genetics*, 23(4):337–341.
- [62] McGee, M. G. (1979). Human spatial abilities: Psychometric studies and environmental, genetic, hormonal, and neurological influences. *Psychological bulletin*, 86(5):889.
- [63] McNamara, T. P. (2002). How are the locations of objects in the environment represented in memory? In *Spatial cognition III*, pages 174–191. Springer.

- [64] Mezhoudi, N. (2013). User interface adaptation based on user feedback and machine learning. In *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*, pages 25–28, Santa Monica, CA. ACM.
- [65] Miller, L. A. and Thomas Jr, J. C. (1976). *Behavioral issues in the use of interactive systems*. Springer.
- [66] Montello, D. R., Lovelace, K. L., Golledge, R. G., and Self, C. M. (1999). Sex-related differences and similarities in geographic and environmental spatial abilities. *Annals of the Association of American geographers*, 89(3):515–534.
- [67] Mou, W. and McNamara, T. P. (2002). Intrinsic frames of reference in spatial memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(1):162.
- [68] Nadel, L. and Hardt, O. (2004). The spatial brain. *Neuropsychology*, 18(3):473.
- [69] Neisser, U. (1987). From direct perception to conceptual structure. In *Emory symposia in cognition, 1. Concepts and conceptual development: Ecological and intellectual factors in categorization*, pages 11–24, New York: Cambridge University Press.
- [70] Neuhäuser, M. (2011). Wilcoxon–mann–whitney test. In *International encyclopedia of statistical science*, pages 1656–1658. Springer.
- [71] Newell, A. F. and Gregor, P. (2000). user sensitive inclusive design in search of a new paradigm. In *Proceedings on the 2000 conference on Universal Usability*, pages 39–44, Washington, DC. ACM.
- [72] Nielsen, J. (1994). *Usability engineering*. Elsevier.
- [73] Norcio, A. F. and Stanley, J. (1989). Adaptive human-computer interfaces: A literature survey and perspective. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(2):399–408.

- [74] Norman, K. L. (1994). Spatial visualizationa gateway to computer-based technology. *Journal of Special Education Technology*, 12(3):195–206.
- [75] O’Laughlin, E. M. and Brubaker, B. S. (1998). Use of landmarks in cognitive mapping: Gender differences in self report versus performance. *Personality and Individual Differences*, 24(5):595–601.
- [76] Pak, R., Rogers, W. A., and Fisk, A. D. (2006). Spatial ability subfactors and their influences on a computer-based information search task. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 48(1):154–165.
- [77] PatanasakPinyo, T., Batinov, G., Whitney, K., and Miller, L. (2016). Methods that flatten the user space for individual differences in location-based surveys on portable devices. In *31st International Conference on Computers and Their Applications (CATA 2016)*, pages 65–70, Las Vegas, Nevada. International Society for Computers and their Applications (ISCA).
- [78] Pattison, M. and Stedmon, A. W. (2006). Inclusive design and human factors: Designing mobile phones for older users. *PsychNology Journal*, 4(3):267–284.
- [79] Penn, A. (2003). Space syntax and spatial cognition or why the axial line? *Environment and behavior*, 35(1):30–65.
- [80] Phillips, R. J. (1979). An experiment with contour lines. *The Cartographic Journal*, 16(2):72–76.
- [81] Phillips, R. J., Lucia, A., and Skelton, N. (1975). Some objective tests of the legibility of relief maps. *The Cartographic Journal*, 12(1):39–46.
- [82] Ramachandran, K. (2009). Adaptive user interfaces for health care applications. *IBM developerWorks*.

- [83] Ramalingam, K. N. (2011). *Modeling error-based Adaptive User Interfaces*. Graduate Theses and Dissertations. 10081. <http://lib.dr.iastate.edu/etd/10081>.
- [84] Rapp, D. N., Culpepper, S. A., Kirkby, K., and Morin, P. (2007). Fostering students' comprehension of topographic maps. *Journal of Geoscience Education*, 55(1):5.
- [85] Rath, A. S., Devaurs, D., and Lindstaedt, S. N. (2010). Studying the factors influencing automatic user task detection on the computer desktop. In *Sustaining TEL: From Innovation to Learning and Practice*, pages 292–307. Springer.
- [86] Raubal, M. and Winter, S. (2002). Enriching wayfinding instructions with local landmarks. In *International Conference on Geographic Information Science*, pages 243–259. Springer.
- [87] Remolina, E. and Kuipers, B. (2004). Towards a general theory of topological maps. *Artificial Intelligence*, 152(1):47–104.
- [88] Rogers, S., Fiechter, C.-N., and Thompson, C. (2000). Adaptive user interfaces for automotive environments. In *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, pages 662–667. IEEE.
- [89] Rusch, M. L. (2008). *Relationships between user performance and spatial ability in using map-based software on pen-based devices*. Iowa State University.
- [90] Salthouse, T. A., Babcock, R. L., Mitchell, D. R., Palmon, R., and Skovronek, E. (1990a). Sources of individual differences in spatial visualization ability. *Intelligence*, 14(2):187–230.
- [91] Salthouse, T. A., Babcock, R. L., Skovronek, E., Mitchell, D. R., and Palmon, R. (1990b). Age and experience effects in spatial visualization. *Developmental Psychology*, 26(1):128.

- [92] Schmidt, A., Beigl, M., and Gellersen, H.-W. (1999). There is more to context than location. *Computers & Graphics*, 23(6):893–901.
- [93] Sein, M. K., Olfman, L., Bostrom, R. P., and Davis, S. A. (1993). Visualization ability as a predictor of user learning success. *International Journal of Man-Machine Studies*, 39(4):599–620.
- [94] Shankar, A., Louis, S. J., Dascalu, S., Houmanfar, R., and Hayes, L. J. (2007). Xcs for adaptive user-interfaces. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 1876–1876. ACM.
- [95] Shepard, R. N. and Hurwitz, S. (1984). Upward direction, mental rotation, and discrimination of left and right turns in maps. *Cognition*, 18(1):161–193.
- [96] Smith, I. M. (1964). *Spatial ability: Its educational and social significance*. RR Knapp.
- [97] Stanney, K. M. and Salvendy, G. (1995). Information visualization; assisting low spatial individuals with information access tasks through the use of visual mediators. *Ergonomics*, 38(6):1184–1198.
- [98] Steichen, B., Ashman, H., and Wade, V. (2012). A comparative survey of personalised information retrieval and adaptive hypermedia techniques. *Information Processing & Management*, 48(4):698–724.
- [99] Streeter, L. A. and Vitello, D. (1986). A profile of drivers’ map-reading abilities. *Human factors*, 28(2):223–239.
- [100] Sukaviriya, P. N. and Foley, J. D. (1992). Built-in user modelling support, adaptive interfaces, and adaptive help in uide. Technical report, Georgia Institute of Technology.
- [101] Taylor, A. (2012). *Improving web accessibility for older adults based on error detection strategies*. Digital Repository@ Iowa State University.

- [102] Taylor Sr, A., Miller, L., Nilakanta, S., Sander, J., Mitra, S., Sharda, A., and Chama, B. (2009). Using an error detection strategy for improving web accessibility for older adults. In *Advances in Computer-Human Interactions, 2009. ACHI'09. Second International Conferences on*, pages 375–380, Cancun, Mexico. IEEE.
- [103] Tlauka, M. and Wilson, P. N. (1994). The effect of landmarks on route-learning in a computer-simulated environment. *Journal of Environmental Psychology*, 14(4):305–313.
- [104] Tsandilas, T. and Schraefel, M. C. (2004). Usable adaptive hypermedia systems. *New Review of Hypermedia and Multimedia*, 10(1):5–29.
- [105] Tversky, B. (2000). Some ways that maps and diagrams communicate. *Spatial Cognition II: Integrating Abstract Theories, Empirical Studies, Formal Methods, and Practical Applications*, pages 72–79.
- [106] Tversky, B., Morrison, J. B., and Betrancourt, M. (2002). Animation: can it facilitate? *International journal of human-computer studies*, 57(4):247–262.
- [107] Uttal, D. H. (2000). Seeing the big picture: Map use and the development of spatial cognition. *Developmental Science*, 3(3):247–264.
- [108] van Nimwegen, C. and van Oostendorp, H. (2009). The questionable impact of an assisting interface on performance in transfer situations. *International Journal of Industrial Ergonomics*, 39(3):501–508.
- [109] Viano, G., Parodi, A., Alty, J., Khalil, C., Angulo, I., Biglino, D., Crampes, M., Vaudry, C., Daurensan, V., and Lachaud, P. (2000). Adaptive user interface for process control based on multi-agent approach. In *Proceedings of the working conference on Advanced visual interfaces*, pages 201–204, Palermo, Italy. ACM.
- [110] Vicente, K. J., Hayes, B. C., and Williges, R. C. (1987). Individual differences in computer-based information retrieval. In *Ergonomics and human factors*, pages 225–229. Springer.

- [111] Vicente, K. J. and Williges, R. C. (1988). Accommodating individual differences in searching a hierarchical file system. *International Journal of Man-Machine Studies*, 29(6):647–668.
- [112] Voyer, D., Voyer, S., and Bryden, M. P. (1995). Magnitude of sex differences in spatial abilities: a meta-analysis and consideration of critical variables. *Psychological bulletin*, 117(2):250.
- [113] Whitney, K., Batinov, G. I., Miller, L., Nusser, S. M., and Ashenfelter, K. T. (2011). Exploring a map survey task’s sensitivity to cognitive ability. In *Statistics Conference Proceedings, Presentations and Posters*, Gosier, France. http://lib.dr.iastate.edu/stat_las_conf/1.
- [114] Willis, K. S., Hölscher, C., Wilbertz, G., and Li, C. (2009). A comparison of spatial knowledge acquisition with maps and mobile maps. *Computers, Environment and Urban Systems*, 33(2):100–110.
- [115] Zhang, J. (1997). The nature of external representations in problem solving. *Cognitive science*, 21(2):179–217.
- [116] Zhang, J. and Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive science*, 18(1):87–122.
- [117] Ziefle, M. and Bay, S. (2006). How to overcome disorientation in mobile phone menus: A comparison of two different types of navigation aids. *Human-Computer Interaction*, 21(4):393–433.
- [118] Zipf, A. and Richter, K.-F. (2002). Using focus maps to ease map reading. *Künstliche Intelligenz*, 4(02):35–37.

APPENDIX A. IRB APPROVAL DOCUMENTS

IRB Approval Document: Study I

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Institutional Review Board
Office for Responsible Research
Vice President for Research
1138 Pearson Hall
Ames, Iowa 50011-2207
515 294-4566
FAX 515 294-4267

Date: 10/1/2015
To: Thitivatr Patanasakpinyo
226 Atanasoff Hall
CC: Dr. Les Miller
112 Atanasoff Hall
From: Office for Responsible Research
Title: Address Verification Using Location-Based Application on Tablet
IRB ID: 15-494
Approval Date: 9/30/2015
Date for Continuing Review: 9/29/2017
Submission Type: New
Review Type: Expedited

The project referenced above has received approval from the Institutional Review Board (IRB) at Iowa State University according to the dates shown above. Please refer to the IRB ID number shown above in all correspondence regarding this study.

To ensure compliance with federal regulations (45 CFR 46 & 21 CFR 56), please be sure to:

- Use only the approved study materials in your research, including the recruitment materials and informed consent documents that have the IRB approval stamp.
- Retain signed informed consent documents for 3 years after the close of the study, when documented consent is required.
- Obtain IRB approval prior to implementing any changes to the study by submitting a Modification Form for Non-Exempt Research or Amendment for Personnel Changes form, as necessary.
- Immediately inform the IRB of (1) all serious and/or unexpected adverse experiences involving risks to subjects or others; and (2) any other unanticipated problems involving risks to subjects or others.
- Stop all research activity if IRB approval lapses, unless continuation is necessary to prevent harm to research participants. Research activity can resume once IRB approval is reestablished.
- Complete a new continuing review form at least three to four weeks prior to the date for continuing review as noted above to provide sufficient time for the IRB to review and approve continuation of the study. We will send a courtesy reminder as this date approaches.

Please be aware that IRB approval means that you have met the requirements of federal regulations and ISU policies governing human subjects research. Approval from other entities may also be needed. For example, access to data from private records (e.g. student, medical, or employment records, etc.) that are protected by FERPA, HIPAA, or other confidentiality policies requires permission from the holders of those records. Similarly, for research conducted in institutions other than ISU (e.g., schools, other colleges or universities, medical facilities, companies, etc.), investigators must obtain permission from the institution(s) as required by their policies. IRB approval in no way implies or guarantees that permission from these other entities will be granted.

Upon completion of the project, please submit a Project Closure Form to the Office for Responsible Research, 1138 Pearson Hall, to officially close the project.

Please don't hesitate to contact us if you have questions or concerns at 515-294-4566 or IRB@iastate.edu.

IRB Approval Document: Study II

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Institutional Review Board
Office for Responsible Research
Vice President for Research
1138 Pearson Hall
Ames, Iowa 50011-2207
515-294-4509
FAX 515-294-4207

Date: 11/12/2014
To: Thilivatr Patanasakpinyo
226 Atanasoff Hall
CC: Dr. Les Miller
112 Atanasoff Hall
From: Office for Responsible Research
Title: Address Verification Using Location-Based Application on Tablet
IRB ID: 14-526
Approval Date: 11/11/2014
Date for Continuing Review: 11/10/2016
Submission Type: New
Review Type: Expedited

The project referenced above has received approval from the Institutional Review Board (IRB) at Iowa State University according to the dates shown above. Please refer to the IRB ID number shown above in all correspondence regarding this study.

To ensure compliance with federal regulations (45 CFR 46 & 21 CFR 56), please be sure to:

- **Use only the approved study materials** in your research, including the recruitment materials and informed consent documents that have the IRB approval stamp.
- **Retain signed informed consent documents for 3 years after the close of the study**, when documented consent is required.
- **Obtain IRB approval prior to implementing any changes** to the study by submitting a Modification Form for Non-Exempt Research or Amendment for Personnel Changes form, as necessary.
- **Immediately inform the IRB of (1) all serious and/or unexpected adverse experiences** involving risks to subjects or others; and (2) any other unanticipated problems involving risks to subjects or others.
- **Stop all research activity if IRB approval lapses**, unless continuation is necessary to prevent harm to research participants. Research activity can resume once IRB approval is reestablished.
- **Complete a new continuing review form** at least three to four weeks prior to the date for continuing review as noted above to provide sufficient time for the IRB to review and approve continuation of the study. We will send a courtesy reminder as this date approaches.

Please be aware that IRB approval means that you have met the requirements of federal regulations and ISU policies governing human subjects research. Approval from other entities may also be needed. For example, access to data from private records (e.g. student, medical, or employment records, etc.) that are protected by FERPA, HIPAA, or other confidentiality policies requires permission from the holders of those records. Similarly, for research conducted in institutions other than ISU (e.g., schools, other colleges or universities, medical facilities, companies, etc.), investigators must obtain permission from the institution(s) as required by their policies. **IRB approval in no way implies or guarantees that permission from these other entities will be granted.**

Upon completion of the project, please submit a Project Closure Form to the Office for Responsible Research, 1138 Pearson Hall, to officially close the project.

Please don't hesitate to contact us if you have questions or concerns at 515-294-4566 or IRB@iastate.edu.

IRB Approval Document: Study III

IOWA STATE UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Institutional Review Board
Office for Responsible Research
Vice President for Research
1138 Pearson Hall
Ames, Iowa 50011-2207
515 294-4566
FAX 515 294-4267

Date: 6/30/2016

To: Thitivatr Patanasakpinyo
226 Atanasoff Hall

CC: Dr. Les Miller
112 Atanasoff Hall

From: Office for Responsible Research

Title: Address Verification Using Location-Based Application on Tablet 3

IRB ID: 16-288

Approval Date: 6/30/2016 **Date for Continuing Review:** 6/29/2018

Submission Type: New **Review Type:** Expedited

The project referenced above has received approval from the Institutional Review Board (IRB) at Iowa State University according to the dates shown above. Please refer to the IRB ID number shown above in all correspondence regarding this study.

To ensure compliance with federal regulations (45 CFR 46 & 21 CFR 56), please be sure to:

- **Use only the approved study materials in your research, including the recruitment materials and informed consent documents that have the IRB approval stamp.**
- **Retain signed informed consent documents for 3 years after the close of the study**, when documented consent is required.
- **Obtain IRB approval prior to implementing any changes** to the study by submitting a Modification Form for Non-Exempt Research or Amendment for Personnel Changes form, as necessary.
- **Immediately inform the IRB of (1) all serious and/or unexpected adverse experiences** involving risks to subjects or others; and (2) **any other unanticipated problems involving risks** to subjects or others.
- **Stop all research activity if IRB approval lapses**, unless continuation is necessary to prevent harm to research participants. Research activity can resume once IRB approval is reestablished.
- **Complete a new continuing review form** at least three to four weeks prior to the **date for continuing review** as noted above to provide sufficient time for the IRB to review and approve continuation of the study. We will send a courtesy reminder as this date approaches.

Please be aware that IRB approval means that you have met the requirements of federal regulations and ISU policies governing human subjects research. **Approval from other entities may also be needed.** For example, access to data from private records (e.g. student, medical, or employment records, etc.) that are protected by FERPA, HIPAA, or other confidentiality policies requires permission from the holders of those records. Similarly, for research conducted in institutions other than ISU (e.g., schools, other colleges or universities, medical facilities, companies, etc.), investigators must obtain permission from the institution(s) as required by their policies. **IRB approval in no way implies or guarantees that permission from these other entities will be granted.**

Upon completion of the project, please submit a Project Closure Form to the Office for Responsible Research, 202 Kingland, to officially close the project.

Please don't hesitate to contact us if you have questions or concerns at 515-294-4566 or IRB@iastate.edu.

APPENDIX B. COMPLETE STUDY RESULTS

We have provided the complete results of all three studies: Study I, Study II, and Study III. Furthermore, we also included the complete set of rules we generated from the logs of thirty participants in Study III.

Study I Results

Table [B.1](#) shows the complete results of thirty participants in Study I.

Study II Results

Table [B.2](#) shows the complete results of thirty-one participants in Study II.

Study III Results

Table [B.3](#) shows the complete results of thirty participants in Study III.

Table B.1 Complete Results of Study I.

Participant	VZ	SoftwareType	AddressVerification	Time	Distance	Error	Pan	Zoom	TotalTap
1	H	0	9.5	52.68333	2.672098188	511	48	31	590
2	H	0	9	43.65	2.31425989	2	75	37	114
3	H	2	8.5	64.43333	3.510059499	45	156	61	262
4	M	2	8	56.8	2.767203439	13	123	51	187
5	H	1	10	41.71667	2.242753113	50	48	30	128
6	H	1	9	81.78333	2.612324983	11	113	28	152
7	L	1	10	45.43333	2.985044657	4	41	21	66
8	M	1	10	52.11667	3.067628738	11	87	42	140
9	L	1	10	59.13333	2.557397494	12	51	52	115
10	M	2	9	44	2.60691043	13	76	70	159
11	L	2	9	42.5	2.60913863	1	57	18	76
12	H	0	10	40.8	2.478767424	26	108	43	177
13	M	1	10	39.13333	2.808666125	28	85	41	154
14	L	0	9	64.5	2.853850823	18	130	11	159
15	L	0	9.5	52.65	3.858040275	24	117	66	207
16	L	2	10	51.08333	3.344754175	42	112	21	175
17	L	2	10	59.88333	3.08317255	7	60	45	112
18	L	2	9.5	31.46667	2.225275618	2	43	15	60
19	H	1	10	38.85	2.611821147	16	185	96	297
20	H	2	10	35.61667	2.110350848	11	92	28	131
21	H	2	8.5	35.75	2.598791329	26	112	39	177
22	M	1	9.5	32.15	2.126374855	7	72	21	100
23	L	1	9.5	55.1	2.767136814	11	76	20	107
24	M	0	10	59.31667	3.048339203	20	41	23	84
25	M	2	9.5	51.3	2.673720966	4	119	47	170
26	L	2	8.5	116.3667	14.36764209	62	270	49	381
27	L	2	10	68.38333	4.268375026	18	97	70	185
28	H	0	10	24.98333	1.76022729	5	30	15	50
29	L	2	10	70.63333	3.626565267	19	92	36	147
30	M	1	10	43.83333	3.399165986	38	134	110	282
31	L	1	10	68.88333	3.135858703	57	138	45	240
32	L	0	9	35.08333	2.082242843	26	74	55	155

Table B.2 Complete Results of Study II.

Participant	VZ	AddressVerification	Time	Distance	Error	Pan	Zoom
1	H	7.5	51.13333333	2.549739	0	12	22
2	L	10	49.91666667	2.884057	10	6	26
3	M	7.5	43.05	2.526282	6	13	29
4	L	10	29.86666667	1.895309	3	4	12
5	H	1	22.98333333	2.257546	7	0	0
6	L	9	41.76666667	2.860667	1	11	6
7	L	10	37.56666667	2.447427	13	16	1
8	L	8.5	60.16666667	2.48887	9	47	33
9	L	9.5	44.5	2.089999	1	20	31
10	M	10	53.5	2.934723	9	42	7
11	M	10	43.23333333	2.011962	15	11	14
12	H	10	36.5	2.177052	10	6	13
13	M	9	34.48333333	2.184058	11	11	12
14	L	9.5	36.05	1.668493	6	13	29
15	L	9	24.41666667	1.724652	19	8	11
16	M	7.5	67.33333333	3.49069	8	17	26
17	M	10	43.26666667	2.168071	1	18	8
18	H	6.5	30.45	1.636737	11	10	17
19	L	10	29.4	2.125188	108	6	17
20	H	10	32.4	2.035001	15	12	4
21	L	8.5	56.8	2.755071	3	23	37
22	L	8.5	55.51666667	2.591428	14	28	53
23	H	8.5	32.7	2.055285	1	12	9
24	L	7.5	52.73333333	3.220695	4	7	22
25	L	8.5	58	2.7183	35	17	6
26	M	10	56.71666667	2.784774	8	4	26
27	L	9.5	49.13333333	2.640335	19	10	42
28	L	10	71.21666667	3.200805	15	44	30
29	H	10	27.85	1.834687	1	4	23
30	M	10	47.41666667	2.563734	7	9	16
31	M	7.5	31.43333333	1.818984	58	10	33

Table B.3 Complete Results of Study III.

Participant	Treatment	Time1	Time2	TotalTime	Distance1	Distance2	TotalDistance	TotalTrap	Error	Pan	Zoom	IO	OI	LR	RL	UD	DU	TotalReversal	VZ	+	-	<	>	A	V	Z
1	0	44.48	23.82	68.3	1.940672	1.157333	3.098005	60	6	32	21	0	2	0	2	1	3	8L	12	9	5	8	7	12	9	
2	0	43.21	23	66.21	2.065943	1.154338	3.220281	105	34	48	15	2	3	3	1	4	4	17L	8	7	8	10	14	16	9	
3	0	33.19	24.53	57.72	1.864976	1.426553	3.291529	104	21	53	28	1	2	2	0	2	3	10H	16	12	9	12	11	21	9	
4	0	43.02	29.87	72.89	2.186764	1.723629	3.910393	146	29	99	17	1	2	2	2	5	8	7	25H	9	8	18	21	27	33	8
5	0	37.76	23.84	61.6	2.1849	1.41218	3.59708	81	4	52	24	0	1	3	2	2	4	12L	15	9	10	14	13	15	7	
8	0	27.12	19.72	46.84	1.567819	1.391131	2.95895	51	12	31	6	1	0	1	1	0	0	3H	4	2	7	8	7	9	9	
9	0	37.94	16.56	54.5	2.402449	1.149063	3.551512	106	13	43	41	2	6	0	1	2	1	12H	21	20	6	12	12	13	13	
10	1	33.52	36.04	69.56	1.880562	1.631403	3.511965	131	39	40	43	6	5	0	2	2	3	18L	22	21	3	7	8	22	12	
11	1	24.72	27.42	52.14	1.382036	1.263462	2.645498	131	21	92	13	0	4	3	2	13	14	36H	7	6	10	16	30	36	8	
12	1	35.64	25.75	61.39	2.111686	1.291287	3.402973	107	10	76	17	1	3	5	0	9	6	24H	8	9	15	13	21	27	10	
13	1	15.7	20.13	35.83	1.242403	1.147462	2.389865	20	0	16	4	0	0	2	1	1	1	5H	3	1	3	3	5	5	5	
14	0	42.84	34.81	77.65	1.62034	1.609099	3.229439	96	32	20	27	0	4	0	0	0	1	5H	14	13	3	5	4	8	7	
15	1	38.88	26.53	65.41	2.371247	1.305636	3.676883	116	17	74	23	0	0	4	1	4	5	14L	15	8	17	20	19	18	8	
16	1	49.92	35.58	85.5	2.337857	1.528143	3.866	336	63	152	62	4	13	8	8	7	7	47H	32	30	32	43	37	40	17	
17	1	35.89	33.7	69.59	1.906242	1.236735	3.142977	115	5	76	16	0	2	2	2	1	7	21H	12	4	10	15	23	28	7	
18	1	25.49	24.35	49.84	1.54554	1.327583	2.873123	72	19	29	24	0	3	1	1	1	1	7H	12	12	6	12	5	6	7	
19	0	31.5	18.42	49.92	1.774191	1.132711	2.906902	96	17	69	7	0	1	3	5	3	7	19H	4	3	15	15	19	20	10	
20	0	33.32	23.31	56.63	1.935252	1.620101	3.555353	96	16	33	41	2	3	1	0	2	2	10H	21	20	7	10	8	8	8	
21	1	26.23	27.01	53.24	1.550172	1.355012	2.905184	64	9	44	5	0	0	0	4	4	3	11H	3	2	7	8	15	14	5	
22	0	41.41	20.29	61.7	2.340612	1.523861	3.864473	147	19	120	4	0	0	6	7	13	11	37L	3	1	24	25	35	36	7	
23	1	33.89	28.22	62.11	1.507296	1.206937	2.714233	256	29	142	58	7	11	6	10	14	6	54H	29	29	24	33	36	49	13	
24	0	51.9	27.1	79	2.3778	1.571852	3.949652	281	166	54	59	5	12	4	3	5	3	32L	31	28	10	19	12	13	8	
25	1	17.4	35.31	52.71	1.532011	2.158521	3.690532	139	17	120	1	0	0	10	8	5	4	27H	1	0	35	32	25	28	10	
26	0	29.94	18.16	48.1	2.011567	1.257397	3.268964	177	42	77	27	1	5	2	5	6	4	23L	14	13	16	18	21	22	8	
27	1	21.46	17.77	39.23	2.158475	1.229457	3.387932	72	18	31	21	0	0	0	1	1	1	3L	11	10	8	8	6	9	11	
28	0	36.08	21.18	57.26	1.672831	1.320273	2.993104	78	24	22	30	3	7	0	0	1	1	12H	15	15	2	6	7	7	7	
29	0	34.07	16.91	50.98	2.341011	1.113249	3.45426	49	2	10	28	1	7	0	0	0	0	8H	15	13	0	5	2	3	8	
30	1	39.1	25.25	64.35	1.736361	1.305367	3.041728	145	15	101	28	2	5	6	4	11	5	33H	16	12	25	30	20	26	4	
31	1	21.9	25	46.9	1.333521	1.181463	2.514984	87	13	56	18	1	3	1	1	8	6	20H	8	10	8	10	17	21	8	
32	1	45.71	32.46	78.17	2.395338	1.462265	3.857645	181	31	68	71	5	12	1	2	1	3	24L	37	34	12	21	20	15	9	

Complete Set of Association Rules

We generated 149 rules from the logs of thirty participants from Study III. Tables [B.4](#) - [B.9](#) show all of those rules.

Table B.4 Association Rules; Rules 1 - 36

Rule#	IF	THEN	SUPPORT	CONFIDENCE
1	first_address_phase2=632Agg	last_address_phase1=2060Cessna	15	1
2	first_address_phase2=632Agg backbone_street=Ash-or-Beach	last_address_phase1=2060Cessna	13	1
3	last_address_phase1=2060Cessna backbone_street=Ash-or-Beach	first_address_phase2=632Agg	13	1
4	last_address_phase2=632Agg	vz=high	11	1
5	last_address_phase1=400Ash	first_address_phase1=2060Cessna	11	1
6	first_address_phase1=2060Cessna	last_address_phase1=400Ash	11	1
7	first_address_phase1=2060Cessna	first_address_phase2=2121Sunset	11	1
8	last_address_phase1=400Ash	first_address_phase2=2121Sunset	11	1
9	first_address_phase1=400Ash first_address_phase2=632Agg	last_address_phase1=2060Cessna	11	1
10	last_address_phase1=400Ash first_address_phase2=2121Sunset	last_address_phase1=2060Cessna	11	1
11	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	last_address_phase1=400Ash	11	1
12	first_address_phase1=2060Cessna last_address_phase1=400Ash	first_address_phase2=2121Sunset	11	1
13	last_address_phase1=400Ash	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	11	1
14	first_address_phase1=2060Cessna	last_address_phase1=400Ash first_address_phase2=2121Sunset	11	1
15	vz=high last_address_phase1=400Ash	first_address_phase1=2060Cessna	10	1
16	vz=high first_address_phase1=2060Cessna	last_address_phase1=400Ash	10	1
17	vz=high first_address_phase1=2060Cessna	first_address_phase2=2121Sunset	10	1
18	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	vz=high	10	1
19	vz=high first_address_phase1=2060Cessna	backbone_street=street-inside-neighborhood	10	1
20	vz=high last_address_phase1=400Ash	first_address_phase2=2121Sunset	10	1
21	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	vz=high	10	1
22	vz=high last_address_phase1=400Ash	backbone_street=street-inside-neighborhood	10	1
23	first_address_phase2=2121Sunset last_address_phase2=632Agg	vz=high	10	1
24	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	vz=high	10	1
25	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna	10	1
26	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	last_address_phase1=400Ash	10	1
27	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna	10	1
28	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	last_address_phase2=2121Sunset	10	1
29	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	last_address_phase1=400Ash	10	1
30	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	first_address_phase2=2121Sunset	10	1
31	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	first_address_phase1=2060Cessna	10	1
32	vz=high first_address_phase1=2060Cessna first_address_phase2=2121Sunset	last_address_phase1=400Ash	10	1
33	vz=high first_address_phase1=2060Cessna last_address_phase1=400Ash	first_address_phase2=2121Sunset	10	1
34	vz=high last_address_phase1=400Ash	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	10	1
35	vz=high first_address_phase1=2060Cessna	last_address_phase1=400Ash first_address_phase2=2121Sunset	10	1
36	backbone_street=street-inside-neighborhood	vz=high	10	1

Table B.5 Association Rules: Rules 37 - 62

Rule#	IF	THEN	SUPPORT	CONFIDENCE
37	vz=high last_address_phase1=400Ash backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna	10	1
38	vz=high first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	last_address_phase1=400Ash	10	1
39	vz=high first_address_phase1=2060Cessna last_address_phase1=400Ash backbone_street=street-inside-neighborhood	backbone_street=street-inside-neighborhood	10	1
40	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	vz=high first_address_phase1=2060Cessna	10	1
41	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	vz=high last_address_phase1=400Ash	10	1
42	vz=high last_address_phase1=400Ash	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	10	1
43	vz=high first_address_phase1=2060Cessna	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	10	1
44	backbone_street=street-inside-neighborhood	vz=high	10	1
45	vz=high first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna	10	1
46	vz=high first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	first_address_phase2=2121Sunset	10	1
47	vz=high first_address_phase1=2060Cessna first_address_phase2=2121Sunset	backbone_street=street-inside-neighborhood	10	1
48	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	vz=high first_address_phase1=2060Cessna	10	1
49	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	vz=high first_address_phase2=2121Sunset	10	1
50	vz=high first_address_phase1=2060Cessna	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	10	1
51	last_address_phase1=400Ash first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	vz=high	10	1
52	vz=high first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	last_address_phase1=400Ash	10	1
53	vz=high last_address_phase1=400Ash backbone_street=street-inside-neighborhood	first_address_phase2=2121Sunset	10	1
54	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	backbone_street=street-inside-neighborhood	10	1
55	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	vz=high last_address_phase1=400Ash	10	1
56	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	vz=high first_address_phase2=2121Sunset	10	1
57	vz=high last_address_phase1=400Ash	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	10	1
58	backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna	10	1
59	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	last_address_phase1=400Ash	10	1
60	backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna last_address_phase1=400Ash	10	1
61	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna last_address_phase1=400Ash	10	1
62	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	10	1

Table B.6 Association Rules: Rules 63 - 86

Rule#	IF	THEN	SUPPORT	CONFIDENCE
63	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	last_address_phase1=400Ash first_address_phase2=2121Sunset	10	1
64	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high	10	1
65	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	first_address_phase1=2060Cessna	10	1
66	backbone_street=street-inside-neighborhood	last_address_phase1=400Ash	10	1
67	vz=high first_address_phase1=2060Cessna last_address_phase1=400Ash	first_address_phase2=2121Sunset	10	1
68	backbone_street=street-inside-neighborhood	backbone_street=street-inside-neighborhood	10	1
69	vz=high first_address_phase1=400Ash first_address_phase2=2121Sunset	vz=high first_address_phase1=2060Cessna	10	1
70	backbone_street=street-inside-neighborhood	vz=high last_address_phase1=400Ash	10	1
71	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high first_address_phase2=2121Sunset	10	1
72	backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna last_address_phase1=400Ash	10	1
73	vz=high last_address_phase1=400Ash backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	10	1
74	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	10	1
75	vz=high first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	last_address_phase1=400Ash first_address_phase2=2121Sunset	10	1
76	vz=high first_address_phase1=2060Cessna first_address_phase2=2121Sunset	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	10	1
77	vz=high first_address_phase1=2060Cessna last_address_phase1=400Ash	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	10	1
78	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	vz=high first_address_phase1=2060Cessna last_address_phase1=400Ash	10	1
79	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	vz=high first_address_phase1=2060Cessna first_address_phase2=2121Sunset	10	1
80	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	10	1
81	vz=high last_address_phase1=400Ash	first_address_phase1=2060Cessna first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	10	1
82	vz=high first_address_phase1=2060Cessna	last_address_phase1=400Ash first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	10	1
83	last_address_phase1=2060Cessna	first_address_phase2=632Aagg	16	0.94
84	first_address_phase2=2121Sunset	vz=high	14	0.93
85	backbone_street=street-inside-neighborhood	vz=high	13	0.92
86	first_address_phase1=400Ash	last_address_phase1=2060Cessna	13	0.92

Table B.7 Association Rules: Rules 87 - 120

Rule#	IF	THEN	SUPPORT	CONFIDENCE
87	first_address_phase1=400Ash last_address_phase1=2060Cessna	first_address_phase2=632Agg	12	0.92
88	first_address_phase1=2060Cessna	vz=high	11	0.91
89	last_address_phase1=400Ash	vz=high	11	0.91
90	first_address_phase1=2060Cessna	backbone_street=street=inside-neighborhood	11	0.91
91	last_address_phase1=400Ash	backbone_street=street=inside-neighborhood	11	0.91
92	last_address_phase2=632Agg	first_address_phase2=2121Sunset	11	0.91
93	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high	11	0.91
94	last_address_phase1=400Ash	vz=high first_address_phase1=2060Cessna	11	0.91
95	first_address_phase1=2060Cessna	vz=high last_address_phase1=400Ash	11	0.91
96	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	vz=high	11	0.91
97	first_address_phase1=2060Cessna	vz=high first_address_phase2=2121Sunset	11	0.91
98	first_address_phase1=2060Cessna	vz=high backbone_street=street=inside-neighborhood	11	0.91
99	last_address_phase1=400Ash first_address_phase2=2121Sunset	vz=high	11	0.91
100	last_address_phase1=400Ash	vz=high first_address_phase2=2121Sunset	11	0.91
101	last_address_phase1=400Ash	vz=high backbone_street=street=inside-neighborhood	11	0.91
102	vz=high last_address_phase2=632Agg	first_address_phase2=2121Sunset	11	0.91
103	last_address_phase2=632Agg	vz=high first_address_phase2=2121Sunset	11	0.91
104	first_address_phase1=2060Cessna last_address_phase1=400Ash	backbone_street=street=inside-neighborhood	11	0.91
105	last_address_phase1=400Ash	first_address_phase1=2060Cessna backbone_street=street=inside-neighborhood	11	0.91
106	first_address_phase1=2060Cessna	last_address_phase1=400Ash backbone_street=street=inside-neighborhood	11	0.91
107	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	backbone_street=street=inside-neighborhood	11	0.91
108	first_address_phase1=2060Cessna	first_address_phase2=2121Sunset backbone_street=street=inside-neighborhood	11	0.91
109	last_address_phase1=400Ash first_address_phase2=2121Sunset	backbone_street=street=inside-neighborhood	11	0.91
110	last_address_phase1=400Ash	first_address_phase2=2121Sunset backbone_street=street=inside-neighborhood	11	0.91
111	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high	11	0.91
112	last_address_phase2=2121Sunset	vz=high first_address_phase1=2060Cessna	11	0.91
113	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	vz=high last_address_phase1=400Ash	11	0.91
114	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high first_address_phase2=2121Sunset	11	0.91
115	last_address_phase1=400Ash	vz=high first_address_phase1=2060Cessna first_address_phase2=2121Sunset	11	0.91
116	first_address_phase1=2060Cessna	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	11	0.91
117	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high backbone_street=street=inside-neighborhood	11	0.91
118	last_address_phase1=400Ash	vz=high first_address_phase1=2060Cessna backbone_street=street=inside-neighborhood	11	0.91
119	first_address_phase1=2060Cessna	vz=high last_address_phase1=400Ash backbone_street=street=inside-neighborhood	11	0.91
120	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	vz=high backbone_street=street=inside-neighborhood	11	0.91

Table B.8 Association Rules: Rules 121 - 146

Rule#	IF	THEN	SUPPORT	CONFIDENCE
121	first_address_phase1=2060Cessna	vz=high first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	11	0.91
122	last_address_phase1=400Ash first_address_phase2=2121Sunset	vz=high backbone_street=street-inside-neighborhood	11	0.91
123	last_address_phase1=400Ash	vz=high first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	11	0.91
124	first_address_phase1=2060Cessna last_address_phase1=400Ash	backbone_street=street-inside-neighborhood	11	0.91
125	last_address_phase1=400Ash first_address_phase2=2121Sunset	first_address_phase1=2060Cessna backbone_street=street-inside-neighborhood	11	0.91
126	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	last_address_phase1=400Ash backbone_street=street-inside-neighborhood	11	0.91
127	first_address_phase1=2060Cessna last_address_phase1=400Ash	first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	11	0.91
128	last_address_phase1=400Ash	backbone_street=street-inside-neighborhood	11	0.91
129	first_address_phase1=2060Cessna	last_address_phase1=400Ash first_address_phase2=2121Sunset	11	0.91
130	first_address_phase1=2060Cessna last_address_phase1=400Ash	backbone_street=street-inside-neighborhood	11	0.91
131	last_address_phase1=400Ash first_address_phase2=2121Sunset	vz=high backbone_street=street-inside-neighborhood	11	0.91
132	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	vz=high first_address_phase1=400Ash backbone_street=street-inside-neighborhood	11	0.91
133	first_address_phase1=2060Cessna last_address_phase1=400Ash	vz=high last_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	11	0.91
134	last_address_phase1=400Ash	vz=high first_address_phase2=2060Cessna first_address_phase2=2121Sunset backbone_street=street-inside-neighborhood	11	0.91
135	first_address_phase1=2060Cessna	backbone_street=street-inside-neighborhood	11	0.91
136	first_address_phase2=632Aagg	vz=high last_address_phase1=400Ash first_address_phase2=2121Sunset	15	0.87
137	last_address_phase1=2060Cessna first_address_phase2=632Aagg	backbone_street=street-inside-neighborhood	15	0.87
138	first_address_phase2=632Aagg	backbone_street=street-inside-neighborhood	15	0.87
139	first_address_phase1=400Ash	backbone_street=street-inside-neighborhood	13	0.85
140	first_address_phase1=400Ash	last_address_phase1=2060Cessna first_address_phase2=632Aagg	13	0.85
141	vz=high backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna	12	0.83
142	vz=high backbone_street=street-inside-neighborhood	last_address_phase1=400Ash	12	0.83
143	vz=high backbone_street=street-inside-neighborhood	first_address_phase2=2121Sunset	12	0.83
144	vz=high backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna last_address_phase1=400Ash	12	0.83
145	vz=high backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna first_address_phase2=2121Sunset	12	0.83
146	vz=high backbone_street=street-inside-neighborhood	last_address_phase1=400Ash first_address_phase2=2121Sunset	12	0.83

Table B.9 Association Rules: Rules 147 - 149

Rule#	IF	THEN	SUPPORT	CONFIDENCE
147	vz=high backbone_street=street-inside-neighborhood	first_address_phase1=2060Cessna last_address_phase1=400Ash	12	0.83
148	last_address_phase1=2060Cessna	first_address_phase2=2121Sunset backbone_street=Ash-or-Beach	16	0.81
149	last_address_phase1=2060Cessna	first_address_phase2=632Agg backbone_street=Ash-or-Beach	16	0.81