

# **Personalized Navigation Instruments for Map User Interfaces**

Daniel Miao

Submitted in partial fulfillment of the  
requirements for the degree of  
Doctor of Philosophy  
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2018



## ABSTRACT

### Personalized Navigation Instruments for Map User Interfaces

Daniel Miao

A map is a big multi-scale information space. The size of a computer display, however, is limited. Users of digital maps often need to repeatedly resize and reposition the map to seek information. These repeated and excess interactions mar the user experience, and create bottlenecks for efficient information processing.

We introduce *personalized navigation instruments*, a class of navigation instruments that leverage personal important spatial entities (e.g., landmarks and routes) to tackle navigation challenges in map user interfaces. Specifically, we contribute the following three instruments, each of which embodies a novel research idea: 1) Personalized Compass (P-Compass) is a multi-needle compass that extends the concept of a conventional compass to help users establish a reference frame. P-Compass localizes an unknown reference point by visualizing its relationship with respect to landmarks. P-Compass leverages what a user knows to help them figure out what they do not know. 2) *SpaceTokens* are interactive map widgets that represent locations, and help users see and link locations rapidly. With SpaceTokens, users can use locations *directly* as controls to manipulate a map, or building blocks to link with other locations. SpaceTokens make locations first-class citizens of map interaction. 3) *SpaceBar* associates a simple linear scrollbar with a complex nonlinear route, thus facilitates efficient route comprehension and interaction. SpaceBar is akin to a scrollbar for a route.

We prototyped these three instruments in a custom smartphone application, used the

application regularly in daily life, and validated our design in two formal studies. While maps are the focus in this dissertation, our ideas need not be limited to maps. For example, we have prototyped P-Compass with Google Street View and a 3D virtual earth tour application. We conclude this dissertation with several directions for future work, such as AR/VR and personalized spatial information user interfaces involving sound, gestures, and speech.



---

## *Table of Contents*

<b>List of Figures</b>	<b>vi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>Acknowledgements</b>	<b>xiv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Example: Seeking Information in a Multi-Scale World . . . . .	2
1.2 Research Questions and Dissertation Goals . . . . .	3
1.3 Thesis and Contributions . . . . .	4
1.4 Dissertation Roadmap . . . . .	8
<b>2 Background</b>	<b>10</b>
2.1 Space-Scale Diagrams . . . . .	10
2.2 Navigation Instruments: What and Why . . . . .	13
2.3 Rethinking Personalization . . . . .	15
<b>3 Related Work</b>	<b>16</b>
3.1 Visualization Techniques for Overview and Detail . . . . .	16
3.1.1 Zooming . . . . .	17

3.1.2	Overview+Detail . . . . .	19
3.1.3	Focus+Context . . . . .	21
3.1.4	Cue-Based . . . . .	23
3.1.5	P-Compass versus the Preceding Four Visualization Strategies . . .	25
3.2	Navigation and Related UI Tools . . . . .	26
3.2.1	Shortcuts for Revisitation . . . . .	26
3.2.2	Constraints for Information Selection . . . . .	28
3.2.3	Tangible UI Tools . . . . .	28
3.2.4	Scrolling, Direction Pointing Tools, and Other Navigation Aids . . .	30
<b>4</b>	<b>Personalized Compass</b>	<b>32</b>
4.1	Introduction . . . . .	32
4.2	Personalized Compass . . . . .	36
4.2.1	Proof-of-Concept Prototype . . . . .	37
4.2.2	Stackability . . . . .	38
4.2.3	Optimized for Personal Use . . . . .	39
4.2.4	Issues of Scalability . . . . .	41
4.2.5	Using POIs to Overcome Scalability . . . . .	42
4.3	User Study . . . . .	43
4.3.1	Apparatus . . . . .	43
4.3.2	Design and Tasks . . . . .	43
4.3.3	Fair Comparison . . . . .	48
4.3.4	Hypotheses . . . . .	50

4.3.5	Procedure . . . . .	51
4.4	Results . . . . .	52
4.4.1	LOCATE . . . . .	53
4.4.2	DISTANCE . . . . .	54
4.4.3	ORIENTATION . . . . .	55
4.4.4	LOCALIZE . . . . .	56
4.4.5	LOCATE+ . . . . .	57
4.4.6	Subjective Preferences and User Comments . . . . .	57
4.5	Discussion . . . . .	59
4.5.1	Design Recommendations . . . . .	61
4.5.2	Limitations . . . . .	62
4.6	Applications: Street View and 3D Map . . . . .	64
4.6.1	Street View . . . . .	64
4.6.2	3D Map . . . . .	66
4.7	Conclusions . . . . .	68
<b>5</b>	<b>SpaceTokens and SpaceBar</b>	<b>69</b>
5.1	Introduction . . . . .	71
5.2	Motivation: Seeing and Linking Locations . . . . .	73
5.3	SpaceTokens . . . . .	75
5.3.1	Using Locations as Constraints Addresses the Seeing Problem . . . .	76
5.3.2	Using Locations as Building Blocks Addresses the Linking Problem	79
5.4	SpaceBar: A Navigation Tool Built out of Locations . . . . .	82

5.5	Prototype Implementation . . . . .	85
5.6	Evaluation . . . . .	85
5.6.1	Study Design . . . . .	85
5.6.2	Hypotheses . . . . .	88
5.6.3	Pilot Study . . . . .	89
5.6.4	Formal Study . . . . .	89
5.7	Results . . . . .	90
5.7.1	Preferences . . . . .	90
5.7.2	On Addressing the Seeing Problem . . . . .	91
5.7.3	On Addressing the Linking Problem . . . . .	91
5.7.4	On SpaceBar and the Open-Ended Task . . . . .	92
5.7.5	NASA TLX Results . . . . .	94
5.8	Discussion . . . . .	94
5.8.1	Design Recommendations . . . . .	95
5.9	Limitations . . . . .	97
5.10	Conclusions . . . . .	98
<b>6</b>	<b>Conclusions and Future Work</b>	<b>100</b>
6.1	Summary of Contributions . . . . .	100
6.2	Future Work . . . . .	103
6.2.1	From P-Compass to SpacePiece . . . . .	103
6.2.2	P-Compass for VR/AR . . . . .	105
6.2.3	Audio Personalized Compass . . . . .	107

6.2.4	Spatial Entities as First-Class Citizens of Map Interaction . . . . .	110
6.2.5	Speech + Sketch Map UI . . . . .	111
6.3	Towards Personalized UIs for Spatial Information . . . . .	115
<b>Bibliography</b>		<b>118</b>
<b>Appendix A P-Compass Algorithms</b>		<b>128</b>
A.1	POI Selection Algorithm . . . . .	128
A.2	POI Clustering Algorithm . . . . .	129
<b>Appendix B User Study Questionnaires</b>		<b>131</b>
B.1	P-Compass Questionnaire . . . . .	132
B.2	SpaceTokens and SpaceBar Questionnaire . . . . .	138

---

## *List of Figures*

1.1	Where am I? (A recreation of a real life example in Google Maps.) (a) After pressing the YouAreHere button, the author was shown this detail map. (b–c) The author needed to zoom out several times until recognizable features came into view. However, details were lost. Maps users often have the contradictory needs to access both overview and detail. This often results in cumbersome interactions in map user interfaces. . . . .	1
1.2	P-Compass uses personal important locations to help users establish a reference frame. (a) Anatomy of a P-Compass. (b) P-Compass smartphone prototype. (c) P-Compass in a virtual 3D tour application. . . . .	5
1.3	(a) SpaceTokens are interactive widgets that represent clones of locations. SpaceTokens make locations first-class citizens of map interaction. For example, users can (b) tap SpaceTokens to specify which locations to be seen, or (c) drag SpaceTokens to specify where locations to be shown. (d) Holding on an on-screen location while tapping SpaceTokens makes the corresponding locations visible on the display. (e) Users can also link an on-screen location to a SpaceToken to establish a relationship (e.g., create a route). . . . .	6

1.4	SpaceBar (the instrument on the right edge of each image) is a scrollbar for a route. Users can adjust the “elevator” indicator in the SpaceBar to control the visible portion of a route. Conversely, interacting with a route also updates SpaceBar. . . . .	7
2.1	(a) A space-scale diagram of a simplified map. (b) A 2D slice of the space-scale diagram. . . . .	11
2.2	Examples of common navigation instruments on desktop (left) and mobile (right) map user interfaces. . . . .	13
3.1	Four strategies to present overview and detail, illustrated using space-scale diagrams. . . . .	17
3.2	Two examples of the overview+detail technique. (a) Classic Google Maps User Interface [Cockburn et al. 2008]. (b) FingerGlass [Käser et al. 2011]. . . . .	20
3.3	Examples of focus+context visualization techniques. (a) Fish-eye map [Keahey and Robertson 1996]. (b) Focus+Context Display [Baudisch et al. 2001]. (c) Hybrid User Interfaces [Feiner and Shamash 1991] (d) LineDrive [Agrawala and Stolte 2001] . . . . .	21
3.4	Examples of cue-based techniques. (a) City Lights [Mackinlay et al. 2003]. (b) Halo [Baudisch and Rosenholtz 2003]. (c) Wedge [Gustafson et al. 2008]. . . . .	23
3.5	Anatomy of a Wedge (solid lines). The target can be located using visual shape completion (dashed lines). . . . .	24

3.6	Where am I? Localization on a smartwatch form factor using (a) P-Compass, (b) Overview+Detail, and (c) Wedge. The two off-screen POIs are 6 cm (“Hotel”) and 4 cm (“Train Sta.”) from the centroid of the display. (In true scale.) . . .	25
3.7	In MetaDesk [Ullmer and Ishii 1997], physical props are used to manipulate a map. . . . .	29
4.1	Where is Yellowstone National Park? Answering this question from a mobile search result often involves (a–b) unnecessary application switching and (b–c) repetitive zooming, as shown in these smartphone screenshots. A P-Compass uses important POIs to provide a first-order approximation to the answer, and can function (i) independently or (ii–iii) with a map. . . . .	33
4.2	(a) Anatomy of a P-Compass. (b) P-Compass prototype. . . . .	36
4.3	Stackability. Information at multiple scales is combined into a single P-Compass. (a) Information at each scale is read independently. (b) Information at all scales is related. . . . .	38
4.4	This figure and Figure 4.5 together form two corners of a virtual display nearly the size of this page. See Figure 4.5 caption for details. . . . .	39
4.5	P-Compass vs. Wedge. This figure and Figure 4.4 are the lower-left and upper-right corners of a virtual display. Wedges (prefixed with “W”) and P-Compasses (prefixed with “PC”) indicate locations of four off-display POIs, $r_1$ , $r_2$ , $b_1$ and $b_2$ . P-Compasses are better for drawing inferences from POIs, while Wedges are better for locating POIs. . . . .	39



4.6	(a) Using supporting POIs to <i>extrapolate</i> an unknown POI (Yellowstone). (b–c) Two alternatives: (b) Emphasizes nearby POIs (mixed-scale representation); (c) Emphasizes POIs close to Yellowstone. . . . .	42
4.7	(a) LOCATE and DISTANCE. (LOCATE was performed on a desktop monitor with no map background.) (b) ORIENTATION. . . . .	45
4.8	Setup for LOCALIZE and LOCATE+. A wrist-worn iPhone 5 is used to emulate a smartwatch. . . . .	46
4.9	Smartwatch screenshots: (a) LOCALIZE; (b) LOCATE+. . . . .	47
4.10	LOCATE Results. (top) Error size. (bottom) Completion time. . . . .	54
4.11	DISTANCE Results. (top) Error size. (bottom) Completion time. . . . .	54
4.12	ORIENTATION Results. (a–b) Error size. (c–d) Completion time. . . . .	55
4.13	LOCALIZE Results. (top) Error size. (bottom) Completion time. The <i>x</i> -axis indicates the parameters of the exploratory conditions (see text for details). . . .	56
4.14	LOCATE+ Results. (top) Error size. (bottom) Completion time. The <i>x</i> -axis indicates the parameters of the exploratory conditions (see text for details). . . .	57
4.15	The idea of personalization. . . . .	62
4.16	(a–c) Interaction with Google Street View. (d) A P-Compass superimposed on a prototype developed with the Google Street View SDK. . . . .	65
4.17	(a–d) Interaction with the Apple 3D earth tour application. (e) A P-Compass prototype on the 3D earth tour application. . . . .	67

5.1	SpaceTokens make locations first-class citizens of map interaction. (a) Four SpaceTokens (gray rectangles with rounded corners) each represent a location. SpaceTokens empower users to use locations directly to address “seeing” and “linking” problems discussed in this chapter. For example, users can (b) tap SpaceTokens to make corresponding locations visible; (c) drag SpaceTokens onto map, so corresponding locations snap to positions of dragged SpaceTokens; (d) hold an on-screen location (using left thumb) and tap several SpaceTokens to see their relative positions; (e) link an on-screen location to a SpaceToken using LinkTool (bubble above left thumb). LinkTools in (c–e) also serve as visible labels for locations being touched. . . . .	70
5.2	Additional tools. (a) CollectionTool is a structure to organize locations. (b) SpaceBar is a scroll-bar-like tool for users to interact with a route. CollectionTool transforms to SpaceBar when creating a route connecting all locations in CollectionTool. . . . .	73
5.3	Creating SpaceToken to represent location. (a) After highlighting location, user taps “+” button to create SpaceToken for location. (b) New SpaceToken is added. (c) User can reorder SpaceTokens by long-pressing one and moving it to desired location. (d) SpaceToken can be removed by swiping it off display edge. . . . .	76

5.4	Using SpaceTokens to specify visibility and position constraints. (a) Where is British Museum? (b) Tap British Museum SpaceToken to see museum (specifying visibility constraint). SpaceToken turns red, screen zooms and pans as appropriate, and blue leader line indicates corresponding location. (c–d) Alternatively, drag British Museum SpaceToken to specify where to display museum (specifying position constraint). LinkTool appears above SpaceToken when touched. Multiple SpaceTokens can be used to mix and match visibility/position constraints of different locations. . . . .	77
5.5	Connect the dots using LinkTool. (a) How long does it take to walk from my hotel to Royal Albert Hall? (b) Tap highlighted location to enable LinkTool. (c) Drag LinkTool to Royal Albert Hall, and press with force to confirm connection between both locations. (d) Route and information view appear. (e) How about walking to British Museum? Drag LinkTool again to British Museum SpaceToken. (f) Route and information view appear after LinkTool touches SpaceToken. (Tapping SpaceToken again would cause display to zoom out so British Museum is visible.) . . . . .	80
5.6	CollectionTool. (a) Using one finger to drag out of CollectionTool enables picker (blue line), which allows users to pick any on-screen location or SpaceToken. (b) Composite SpaceToken on top of CollectionTool represents CollectionTool content (three locations here). Tapping composite SpaceToken shows all three locations on display. . . . .	82

5.7	(a) SpaceBar is a scrollbar for a route. (b) User can change size and position of elevator indicator to make corresponding portion of route visible. (c) Conversely, elevator indicator is also updated as user interacts with route. (d) User can jump to any portion of route quickly by tapping SpaceBar, or (e) traverse route by scrubbing SpaceBar. . . . .	84
5.8	TLX Results. Lower numbers are preferred for all measures. . . . .	94
6.1	Concepts underlying three personalized navigation instruments. (a) P-Compass visualizes locations of known landmarks with respect to a reference point. (b) SpaceTokens facilitate users to see and link locations rapidly. (c) SpaceBar is a scrollbar for a route. . . . .	101
6.2	Apple’s VR walkthrough interface. . . . .	106
6.3	Drawing interface to mark custom spatial entities. (a) Holding the drawing button at the bottom left corner of the display, the user can then (b) sketch a line to specify a path or (c–e) sketch a (nearly) closed outline and scribble inside it to specify an area. (f) Tapping the “+” button at the lower right corner of the display creates a SpaceToken for the path or area. . . . .	110
6.4	Speech + sketch map user interface on a smartphone. . . . .	112
6.5	Speech + sketch map user interface in 3D (e.g., VR/AR). . . . .	114
6.6	The high-level philosophy of our ideas. . . . .	115
6.7	(Top) Conventional user interfaces for spatial information. (Bottom) Personalized user interfaces for spatial information. . . . .	116

---

## *List of Tables*

4.1	Summary of error results (s.d.). . . . .	53
4.2	Summary of completion times (s.d.). . . . .	53
4.3	Participants' preferences on each task. . . . .	58
5.1	Participants' preferences for each task. . . . .	90

---

## *Acknowledgements*

This dissertation would not have been possible without the support from many people. I am eternally grateful for those who have provided guidance, critical feedback, help, and encouragement along this journey. I consider myself extremely fortunate to have them as mentors, collaborators, friends, and family.

Thank you, advisor and committee members, for the guidance on my research: Steven Feiner, Barbara Tversky, John Kender, Lydia Chilton, and Eugene Wu.

Thank you, members of the CGUI lab, for all the discussions and prototype development support: Mengu Sukan, Carmine Elvezio, Ohan Oda, Hong Guo, Kevin Lin, James Xu, and Blaine Bell.

Thank you, members of the Vision Lab and the VAP Lab, for helping develop my skills in conducting research: Shree Nayar, Oliver Cossairt, Changyin Zhou, Mohit Gupta, Guru Krishnan, Neeraj Kumar, Brian Smith, Shigehiko Matsunaga, Ryunosuke Yokoya, Makoto Odamaki, Kensei Jo, Daisuke Iso, Eric Kee, Daniel Sims, Qi Yin, Thomas Berg, Jiongxin Liu, and Anne Fleming.

Thank you, neighbors on the 6th floor CEPsR, for the friendship and encouragement: Hao Deng, Yinxiao Li, Chun-Yu Tsai, Fang-Hsiang Su, Adrian Tang, Dingzeyu Li, Olivia Winn, and Robert Colgan.

Special thanks to my New York-based relatives, my uncle Shaw-Fu Hwang, my aunt Sue-Mei Ho, my cousins Jeffrey Hwang, Richard Hwang, Michael Hwang and Vincent Lin. They have been treating me like their own son/brother. Because of them, New York has always been home to me. Special thanks to my parents Jiun-Jih Miao, Chu-Shian Ho, my sister Fi-Fi Miao in Taiwan, and my wife Jung Won Lee, for their constant support and encouragement. Without them, I would be nothing.

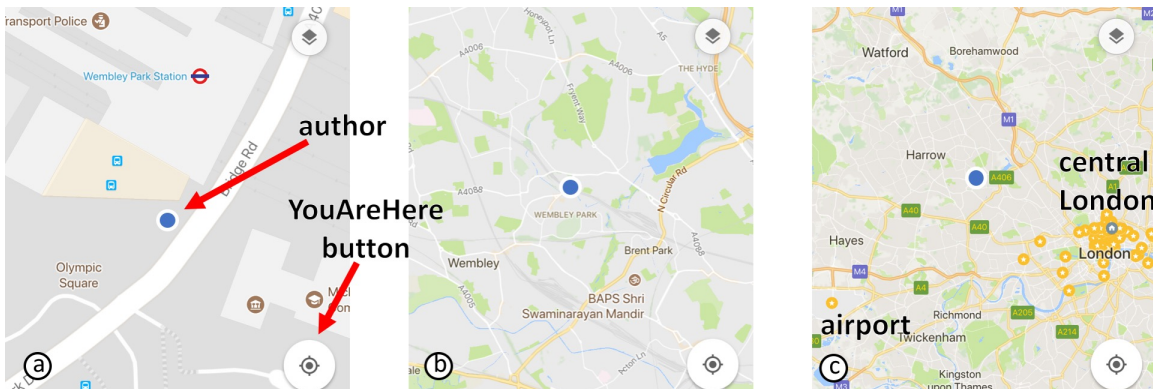
To my family



## Chapter 1

### *Introduction*

Maps are one of the most important inventions in human history. Today, digital maps are found in all corners of our lives, and serve one of the largest user bases (almost everyone). The multi-scale nature of maps, however, makes it challenging to navigate on computer displays; users often need to repeatedly resize and reposition a map in order to access and integrate information scattered over different regions and across multiple scales. These excess and repeated interactions mar the user experience, and at the same time, create bottlenecks for efficient information processing and task completion.



**Figure 1.1: Where am I? (A recreation of a real life example in Google Maps.) (a) After pressing the YouAreHere button, the author was shown this detail map. (b–c) The author needed to zoom out several times until recognizable features came into view. However, details were lost. Maps users often have the contradictory needs to access both overview and detail. This often results in cumbersome interactions in map user interfaces.**

## 1.1 Example: Seeking Information in a Multi-Scale

### World

To provide a more concrete example, let us revisit a recent instance that happened around the time of this writing. One day in late September 2017, while visiting London, the author stepped out of the Wembley Park tube station and was unsure where he was in London (or, where *Wembley Park tube station* is in London). The author pressed the YouAreHere button on Google Maps and the author's location was shown as a blue dot at the center of a map (Figure 1.1a). This map showed detail, yet was devoid of meaningful cues to provide the author the context, given that the author had limited knowledge about London. So the author zoomed out several times (Figure 1.1b–c) until recognizable features began to emerge. At the overview level, however, details were lost. To view the bus stop in the area, the intersections of the streets, the author needed to zoom in the map again.

As the author made his way to a destination in the area (an IKEA store), he continued making repeated round trips between global and local scales in the map user interface. A search of “IKEA” showed the store on the map but failed to indicate where IKEA is with respect to the Wembley Park tube station. Zooming out the map unveiled both IKEA and the tube station, but hid the road network between them. To see his progress in following a route, he zoomed out the map to see the destination and where he was. At the overview level, however, he was not able to decide whether to take a shortcut. Seeking information in a map involves lots of conflicting goals.

Readers can probably relate their own experiences with this example, as similar situations occur in many different settings, whether navigation in-situ or not: planning a trip on

a couch and looking for a restaurant on a smartphone, to name a few.

## 1.2 Research Questions and Dissertation Goals

The preceding example provides some flavor of the challenges of navigation in maps, which are big continuous multi-scale information spaces. During the course of this dissertation research, we focused our effort on the following three research questions:

1. *How can we help users establish a reference frame in a map?* One fundamental need when working with maps is to maintain a frame of reference (how the visible portion of the map relates to the global space). However, the limited size of a computer display often leaves important information off-display. When this happens, the displayed information space could be devoid of meaning cues, and make it difficult for users to *localize* the displayed information space. In the previous IKEA example, the map shown in Figure 1.1(a) contains no meaningful cues to help the author figure out where he was in London. What information could help users establish a frame of reference? Is it possible to design an instrument to provide such information?

2. *How can we help users see and link locations in a map rapidly?* Map users often need to interact repetitively with multiple important locations. For example, a traveler may frequently check her hotel or a train station on a map, use them to localize an unknown location, or investigate routes involving them. We refer to these activities as *seeing* and *linking* locations. Ironically, these location-centric tasks cannot be performed using locations directly; users must instead pan and zoom the map or use a menu to access locations. Can we design tools to empower users see and link locations efficiently?

3. *How can we help users comprehend and interact with a route in a map efficiently?*

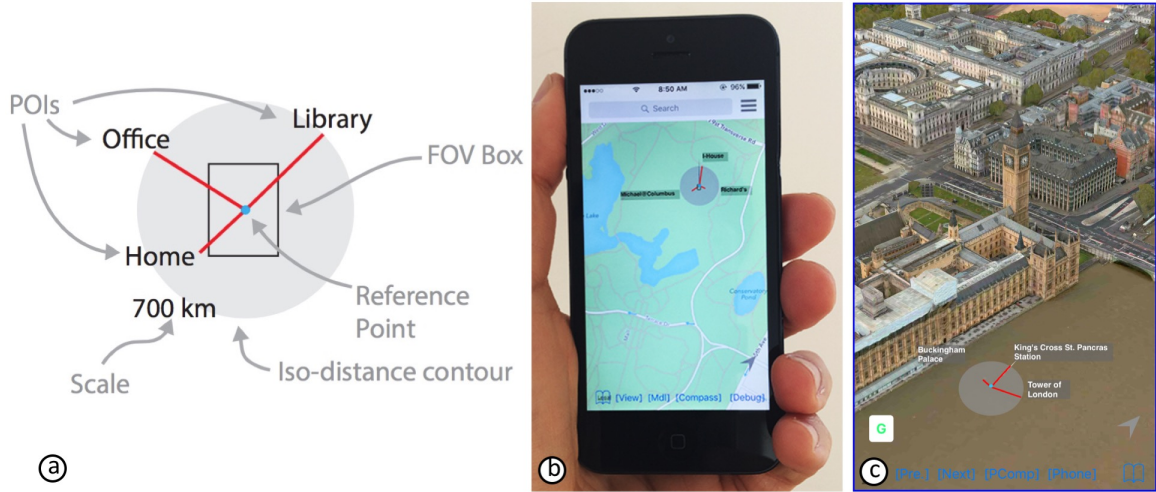
Routes are a class of geometric entity that users frequently interact with in a map. For example, a user may trace a highway (e.g., US-66)—a sequence of connected points—to learn about its geographic and historic background, or study a route from *A* to *B*—a sequence of temporary related points. During the process, a user may traverse the line, zoom in to investigate details, zoom out to gain the context, and occasionally jump to different parts of the line to enforce memory. Performing all these actions repeatedly could be cumbersome. Can a navigation instrument facilitate efficient interaction with a route in a map?

## 1.3 Thesis and Contributions

Our thesis is twofold: First, by strategically leveraging personal important spatial entities (e.g., landmarks and routes), a class of navigation instruments can be designed to tackle navigation challenges in map user interfaces. We refer to this class of navigation instruments as *personalized navigation instruments*. Second, the novel use of personal spatial entities can make these instruments simple and compact, so that they can be easily integrated into existing map user interfaces, which is especially advantageous for mobile and wearable devices.

With the research questions framed above, we make the following contributions, in the form of navigation instruments. These navigation instruments embody a set of visualization and interaction techniques.

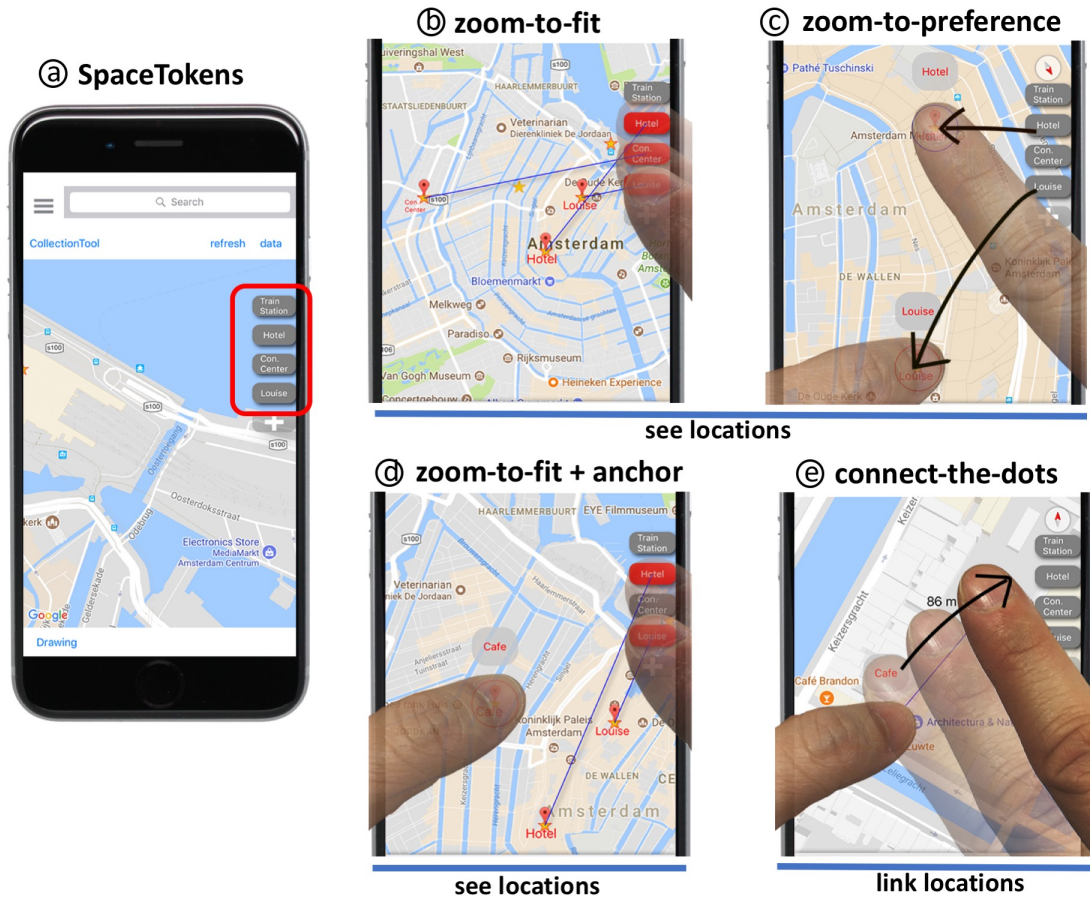
1. *To help users establish a reference frame, we propose Personalized Compass (P-Compass).*



**Figure 1.2: P-Compass uses personal important locations to help users establish a reference frame. (a) Anatomy of a P-Compass. (b) P-Compass smartphone prototype. (c) P-Compass in a virtual 3D tour application.**

P-Compass (Figure 1.2a) is a multi-needle compass, with each needle indicating the direction of and distance from a reference point to a personal *a priori* point of interest (POI), i.e., a POI known to a user. These needles collectively communicate the relationships of known POIs with respect to the reference point. In other words, P-Compass uses “landmark” POIs to help users establish a reference frame. This idea is inspired by the observation that visitors often use landmark POIs (e.g., hotel, train station, and restaurant) to localize themselves in a new environment.

We have designed and prototyped P-Compass on a iOS map application (Figure 1.2b), a 3D map application (Figure 1.2c), and a web-based StreetView application (Section 4.6). We also conducted a user study to compare P-Compass and Wedge—a well known visualization technique for communicating the locations of off-screen objects (discussed in Section 3). Our results suggest that the two techniques are complementary. However, as the distances of POIs increases, or the size of a display decreases, P-Compass becomes more effective than Wedge. Based on our prototyping experience and user study, we codify a set

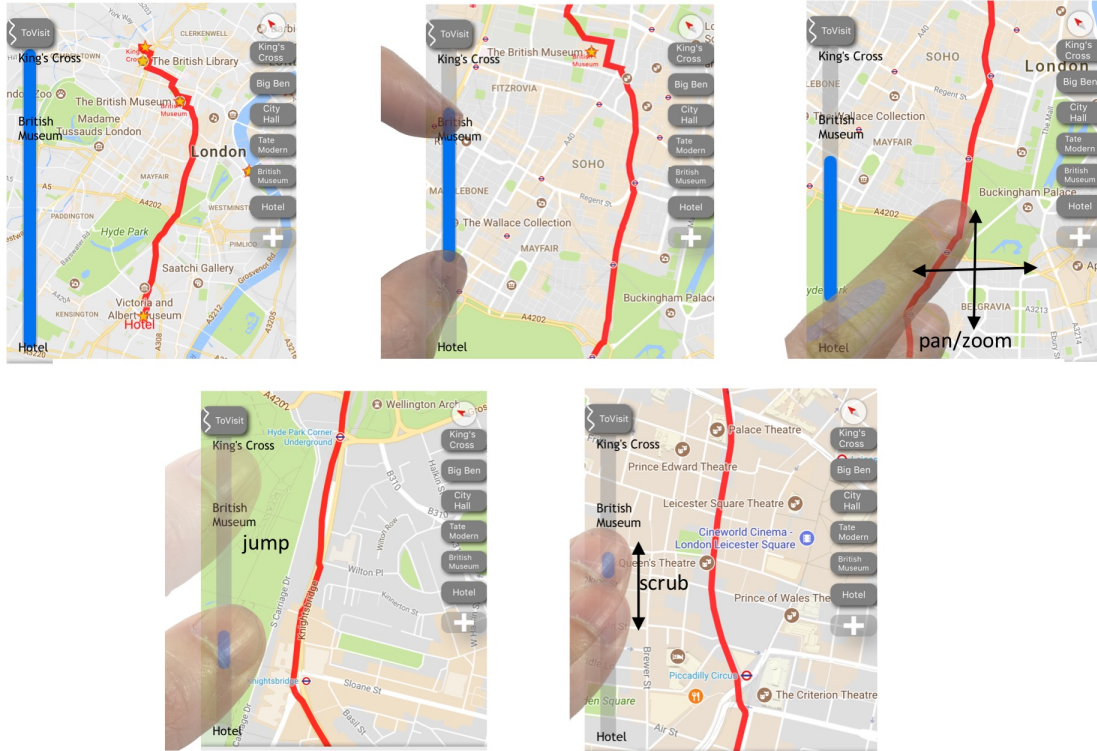


**Figure 1.3:** (a) SpaceTokens are interactive widgets that represent clones of locations. SpaceTokens make locations first-class citizens of map interaction. For example, users can (b) tap SpaceTokens to specify which locations to be seen, or (c) drag SpaceTokens to specify where locations to be shown. (d) Holding on an on-screen location while tapping SpaceTokens makes the corresponding locations visible on the display. (e) Users can also link an on-screen location to a SpaceToken to establish a relationship (e.g., create a route).

of design recommendations.

2. To help users see and link important POIs, we introduce SpaceTokens. SpaceTokens (Figure 1.3) are interactive widgets that represent clones of locations. Figure 1.3 (a) shows four SpaceTokens (gray rectangles with rounded corners) docked along one edge of a map. SpaceTokens support a set of interaction techniques that empower users to perform rapid location-centric tasks, seeing locations and linking locations, *directly* using locations. For example, users can tap SpaceTokens to make corresponding locations visible (Figure 1.3b);





**Figure 1.4:** SpaceBar (the instrument on the right edge of each image) is a scrollbar for a route. Users can adjust the “elevator” indicator in the SpaceBar to control the visible portion or a route. Conversely, interacting with a route also updates SpaceBar.

drag SpaceTokens onto a map, so corresponding locations snap to the positions of dragged SpaceTokens (Figure 1.3c); hold an on-screen location (using left thumb) and tap several SpaceTokens to see their relative positions (Figure 1.3d); and link an on-screen location to a SpaceToken using LinkTool (bubble above left thumb) (Figure 1.3e). LinkTools in Figure 1.3(c–e) also serve as visible labels for locations being touched. In all these examples, locations are the main characters of interactions. SpaceTokens make locations first-class citizens of map interaction.

3. *To help users interact with a route, we introduce a compact interactive widget called SpaceBar.* A SpaceBar (Figure 1.4) is akin to a scrollbar for a route. A SpaceBar represents a straightened and scaled route—there is a one-to-one mapping between each point on a

SpaceBar and each point on a route. Similar to a scrollbar, a SpaceBar has an “elevator” indicator that represents the portion of a route visible to a user. SpaceBar facilitates a set of convenient interactions for a user to explore and review a route. For example, a user can change the size or position of the elevator indicator to make corresponding portion of the route visible. Conversely, the elevator indicator is updated as a user interacts with the route. A user can jump to any portion of the route quickly by tapping the SpaceBar. A user can also traverse the route by scrubbing the SpaceBar.

We have implemented both SpaceTokens and SpaceBar in a custom smartphone map application, and used the application regularly throughout the iterative design process. From our field experience, we designed and carried out a formal user study to compare SpaceTokens as well as SpaceBar with Google Maps. Participants in our study overwhelmingly preferred our approaches over Google Maps for the majority of tasks. Based on our prototyping experience and user study, we provide a set of design recommendations.

A common thread connecting our proposed navigation instruments is personalization. P-Compass uses personal *a priori* POIs to help users establish a reference frame. SpaceTokens make personal POIs first-class citizens of map interaction to address the seeing and linking problems. SpaceBar personalizes a scrollbar to map the linear instrument to a route of arbitrary shape to help users interact with and comprehend a route.

## **1.4 Dissertation Roadmap**

Here we provide an overview of the structure of this dissertation. Readers could take this section as a roadmap to guide them navigate this dissertation.



Chapter 2 provides the background for this dissertation. We examine the challenges of navigating in multi-scale information spaces (e.g., maps) using space-scale diagrams. Space-scale diagrams are tools used by researchers to formally analyze the challenges underlying multi-scale navigation and evaluate different types of solutions. We also discuss our reasons of focusing on navigation instruments, and our aim to rethink personalized navigation instruments.

In Chapter 3, we review related work, specifically on techniques to visualize overview and details. We also review relevant user interface techniques and tools to help users navigate in a multi-scale information space. We discuss how our contributions are placed in the context of the related work.

In Chapter 4, we present Personalized Compass in detail. We motivate a problem that we call the “where is x?” problem, and discuss how P-Compass can address the problem. We present our prototype, and describe a formal user study to compare P-Compass with Wedge. We conclude this section with recommendations for designers.

In Chapter 5, we present SpaceTokens and SpaceBar together. We describe two problems termed the seeing and linking problems. We then describe how SpaceTokens and their interaction techniques can be used to address the seeing and linking problems. Building on the ideas of SpaceTokens, we introduce SpaceBar. We present our prototype for SpaceTokens as well as SpaceBar, and describe a formal user study that compares SpaceTokens and SpaceBar with Google Maps. We conclude with a set of recommendations.

Chapter 6 concludes this dissertation. In addition to presenting our conclusions, we discuss several directions for future work. The surveys used in the user studies presented in this dissertation are included in the appendix.

## Chapter 2

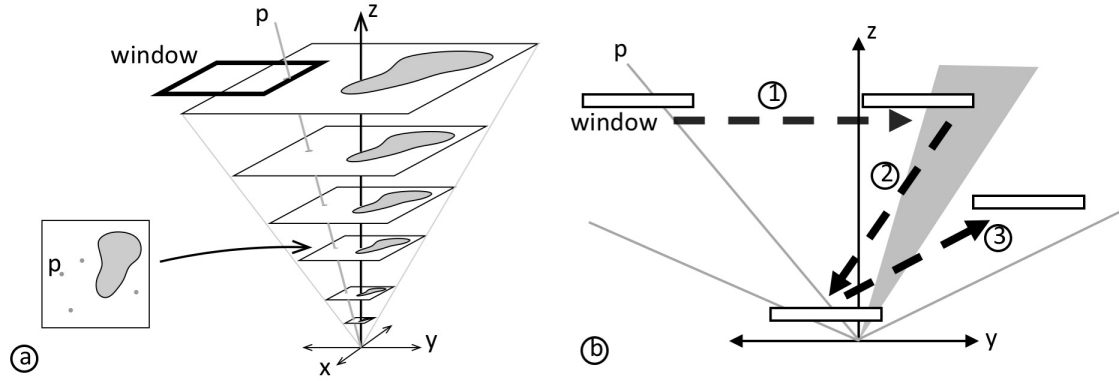
---

### *Background*

In Chapter 1, we motivated this dissertation with a real-world example. This chapter provides the background for this dissertation. We will review space-scale diagrams, a tool to formally analyze challenges of navigation in a multi-scale information world (e.g., maps, blueprints, and circuit diagrams). We will then use space-scale diagrams to examine several common challenges of navigation in map user interfaces. We will also discuss why we focus on navigation instruments, and our aim to rethink personalized navigation instruments. By the conclusion of this chapter, readers will have seen that many of the navigation challenges in user interfaces for maps apply in general to user interfaces for multi-scale spatial information. Therefore, the results of this dissertation could potentially be applied to user interfaces for other multi-scale spatial information.

### **2.1 Space-Scale Diagrams**

Space-scale diagrams [Furnas and Bederson 1995] were developed to formally analyze the challenges of navigation in large 2D information worlds (e.g., maps). A space-scale diagram consists of a 2D information world at all possible scales, and visualizes scale



**Figure 2.1:** (a) A space-scale diagram of a simplified map. (b) A 2D slice of the space-scale diagram.

as an explicit dimension.<sup>1</sup> Figure 2.1 (a) shows a space-scale diagram for a simplified map (as a simple image). The vertical  $z$ -axis indicates scale, with the upwards direction corresponding to increasingly larger scale. The horizontal  $x$ - and  $y$ -axes indicate the two dimensions of the map. It should be noted that the scale dimension is continuous in a space-scale diagram. For the purpose of illustration, pictures of several selective scales are shown. In a space-scale diagram, a point in a 2D information space becomes a ray (e.g., Figure 2.1a shows only the ray of a feature point  $p$ ). A circular region on the map becomes a cone in space-scale diagrams. An arbitrary region on a map thus becomes a generalized cone. When interacting with a map, a user essentially shifts a window in a space-scale diagram.

Without loss of generality, we can look at a 2D version of the space-scale diagram. Figure 2.1 (b) shows a 2D slice (along the  $y - z$  plane) of the space-scale diagram in Figure 2.1 (a). The gray lines are rays corresponding to feature points (e.g.,  $p$ ) in an information space. And the gray wedge corresponds to a slice of a generalized cone. In

<sup>1</sup>It is interesting to note that research in neuroscience has found evidence for a multi-scale spatial representation in the rat hippocampus [Kjelstrup et al. 2008] (a major component of the brain that is known to be associated with spatial memory [O’Keefe and Nadel 1978]). As an analogy, one may consider the map in a rat’s brain to have a “hardware” multi-scale representation (formed by space cells) of a spatial environment.

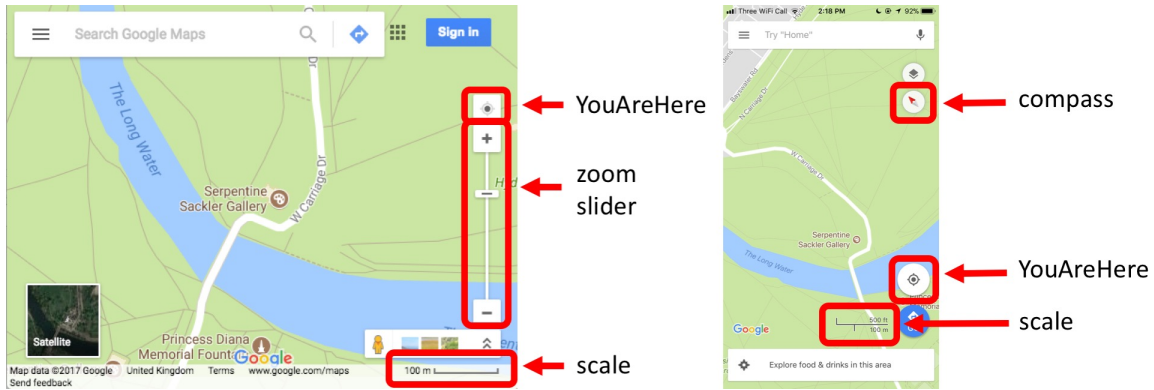
a 2D space-scale diagram, panning the map is equivalent to shifting the window at the same scale (Figure 2.1b1). If a user wishes to zoom out, the user moves the window down the space-scale diagram (Figure 2.1b2). Conversely, if a user wishes to zoom in, the user moves the window up the space-scale diagram (Figure 2.1b3).

As a user zooms out (moving the window towards the bottom of the space-scale diagram), certain features in the information world becomes indiscernible (e.g., the cross sections of the cones become increasingly smaller towards the bottom of the diagram). Even assuming a display has infinite resolution, human eyes are not capable of resolving features smaller than a certain size<sup>2</sup>. As a user zooms in (moving towards the top of the space-scale diagram), while detail information becomes recognizable, certain features could move outside of the window (become off-screen). Similarly, important features could fall off-screen as users pan the map (Figure 2.1b1). This has been referred to as the “off-screen problem” [Baudisch and Rosenholtz 2003]. One problem closely related to the off-screen problem is that at some point a window could end up containing no meaningful cues for a user (Figure 2.1b3, also as illustrated in Figure 1.1). This has been referred to as the “desert fog condition” [Jul and Furnas 1998], in which visible information is devoid of the cues needed to assist users in making decision.

Last but not the least, an information world could contain important information scattered across different regions of a space-scale diagram. Researchers have noticed that users tend to visit certain regions of an information world repeatedly. For example, before or during a trip, a traveler may repeatedly visit on a map a hotel, a train station, and the general

---

<sup>2</sup>The angular resolution of the human eye is approximately  $0.02^\circ$ , which corresponds to  $0.34mm$  at  $1m$  distance [Yanoff and Duker 2008].



**Figure 2.2: Examples of common navigation instruments on desktop (left) and mobile (right) map user interfaces.**

area of a city . This has been called the “revisitation problem” [Zhao et al. 2013].

A map is a large continuous multi-scale information world. Due to the characteristics of maps, an information-seeking task is often an exercise of moving the window within the space-scale diagram. As the display size becomes smaller, users will need to pan and zoom more frequently and more extensively. These frequent and extensive interactions require time and effort, and are sources of inefficiency.

## 2.2 Navigation Instruments: What and Why

We use the term *navigation instruments* in this dissertation to refer to tools that assist users in navigating in maps efficiently (e.g., with less time or effort). These tools can be regarded as the physical embodiments of solutions (in the forms of visualization and interaction techniques) to targeted navigation challenges. Prominent examples of existing navigation instruments include the compass rose (or “compass” for short), YouAreHere button, zoom slider, and scale bar (Figure 2.2). Navigation instruments are compact and often tucked away at the periphery of map user interfaces [Google 2017a; Microsoft 2017;

Apple 2017a]. These instruments assist users in interacting with maps in subtle yet powerful ways. We have often grown accustomed to existing instruments and thus may underestimate the value they provide. To understand the importance of navigation instruments, let's consider an alternative world in which navigation instruments such as the compass do not exist.

Without a compass, the only way to determine the orientation of a map would be to shift the window down to the bottom of a space-scale diagram and then shift the window back to the initial scale. Doing so requires unnecessary time and effort. However, it is essentially what we do when we navigate on the street to walk from an unknown location to a destination; we repeatedly zoom into and out of the map to make sure we are heading in the right direction.

A compass allows a user to focus on the currently visible information, while gaining knowledge about the relationship between that information and the global information space. In general, navigation instruments reduce the interactions required to manipulate the windows in a space-scale diagram. Most importantly, navigation instruments can be easily integrated into an existing map user interface, with no modification of how the underlying information space is presented. As we will see in Section 3, other user interface techniques exist to assist users in gaining global or contextual knowledge of an information world. However, those techniques can interrupt a user's workflow, or are costly in terms of the time for interaction or the space for visualization.

## 2.3 Rethinking Personalization

This dissertation explores a class of navigation instruments we call *personalized navigation instruments*. The notion of personalized navigation instruments may mean customizing a classical instrument to make it behave in a specific way (e.g., making a compass point to a specific POI, as opposed to north) or annotating an instrument to indicate useful information (e.g., adding scale annotations to a zooming slider). We refer to this type of personalized navigation instruments as personalized *classical* navigation instruments. Personalized classical navigation instruments address the same categories of problems as their original counterparts, albeit in more specific ways. For example, a compass with its needle pointing to a specific POI addresses the same direction point problem. Similarly, an annotated zoom slider addresses the same zoom control/visualization problem.

The goal of this dissertation is not to advance the solutions of existing problems in more specific ways. Rather, this dissertation identifies several common navigation problems that are not addressed by any of the existing navigation instruments, and attempts to tackle these problems by designing new instruments. Motivated by the observation that certain spatial entities are repeatedly accessed, or are of special importance to a person’s mental model of a spatial environment, we ask how we can design navigation instruments based on these personal important spatial entities? To this end, this dissertation attempts to rethink the idea of personalized navigation instruments: For a given challenge in map user interfaces, what spatial entities may help users address that challenge? What entities should be selected? And in what form should the entities be put together to maximize an instrument’s usability and functionality while minimizing its footprint?

## Chapter 3

---

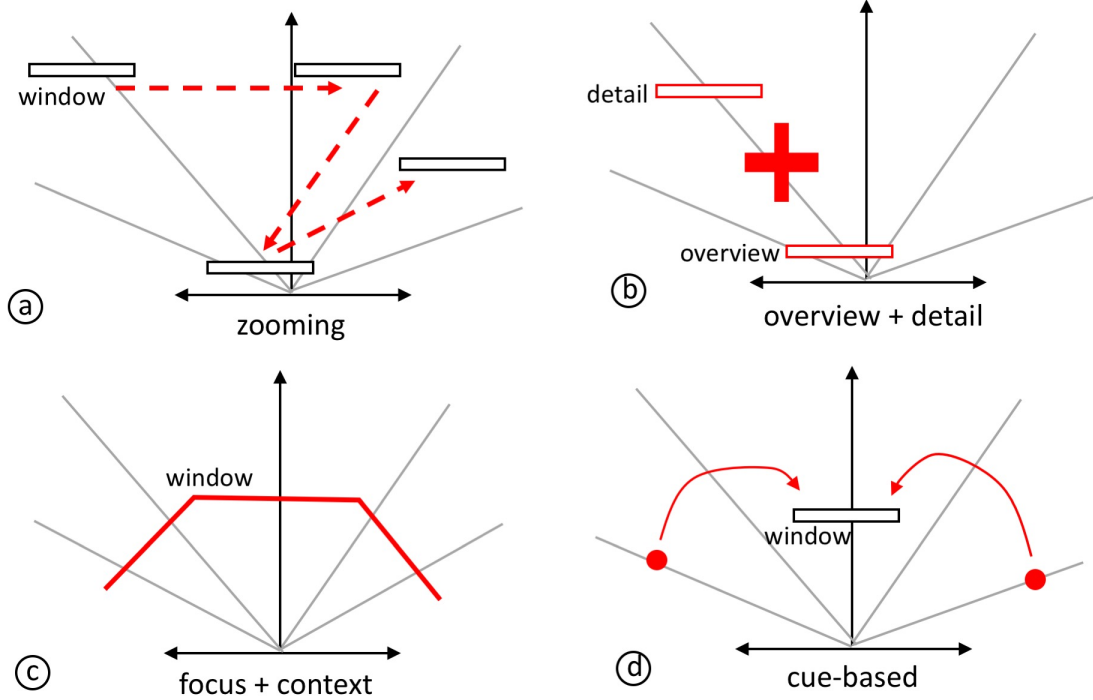
### *Related Work*

This dissertation is inspired by and builds on existing research and applications. In this chapter, we will review related work. First, we will discuss visualization techniques for presenting information in overview and in detail. In general, these techniques can be categorized into four schemes. As we will see, P-Compass is a hybrid technique that combines several of these schemes. We will then review navigation instruments and related UI tools for interacting with spatial information. This second part of the chapter is especially relevant to SpaceTokens and SpaceBar.

### **3.1 Visualization Techniques for Overview and Detail**

A large body of work has been dedicated to strategies for presenting details at multiple scales on a limited display area. Cockburn et al. [2008] proposed a taxonomy to categorize these into four basic schemes: Zooming, Overview+Detail, Focus+Context, and Cue-based techniques. Figure 3.1 uses space-scale diagrams to illustrate the four strategies. *Zooming* (Figure 3.1a) allows users to examine information at multiple scales by literally moving the window. *Overview+Detail* (Figure 3.1b) superimposes a view from a different scale on the existing view. *Focus+Context* (Figure 3.1c) seamlessly combines a distorted contextual view with an undistorted focused view. (One equivalent way to think about Focus+Context





**Figure 3.1: Four strategies to present overview and detail, illustrated using space-scale diagrams.**

is that the window is deformed in a way that edge of the window samples information at multiple scales at once.) *Cue-based* techniques (Figure 3.1d) selectively visualize off-screen objects inside the existing view.

Below we will structure our review using these four schemes, highlighting selected work that is related to our proposed ideas. (For a more through review, one can refer to several excellent publications [Cockburn et al. 2008; Ware 2012; Munzner 2014].) We will then describe how P-Compass relates to and builds on this prior research.

### 3.1.1 Zooming

In zooming user interfaces [Perlin and Fox 1993; Bederson et al. 1996; Bederson 2011], users zoom and pan (and sometimes rotate) an information space to enable information

of interest to fit into a window. These interactions are equivalent to moving a window in a space-scale diagram (Figure 3.1a). In zooming user interfaces, information at different scales is available at different points of time; therefore, zooming is also referred to as *temporal separation*.

Zooming is one of the most widely implemented technique for spatial information navigation. Several variations and extensions have been proposed under this category. Semantic zoom [Perlin and Fox 1993; Donelson 1978; Apperley et al. 1982] refers to the idea that the appearance of an information object changes as a function of the zoom level. For example, a university campus may appear differently at different zoom levels: a campus boundary enclosing several building outlines, a simple polygon filled with simple color, or an iconic representation with textual description. Speed-dependent automatic zooming [Igarashi and Hinckley 2000] is a technique that automatically adjusts the scale based on the speed at which users pan/scroll an information space. For example, as a user pans a map at a faster speed, the map zooms out to reveal more global information. As the users slows down, the map zooms back in.

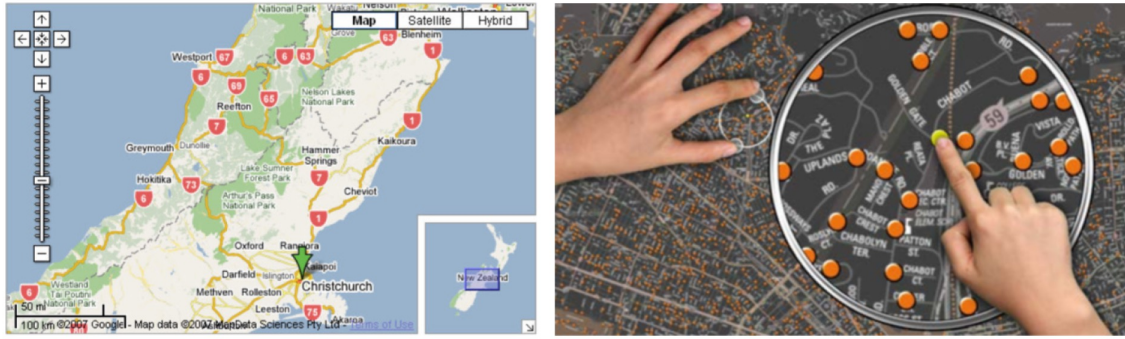
How can users manipulate an information world (or, move the window in a space-scale diagram) in zooming user interfaces? Methods differ depending on the types of devices. Desktop zooming user interfaces allow users to interact with spatial information through explicit on-screen navigation instruments (e.g., zoom sliders and buttons for zooming in and out) and with keyboard shortcuts and mouse scrolling control. Multi-touch surfaces (e.g., tablets and smartphones) allow users to use gestures such as pinch-zoom, two-finger rotate, and one-finger pan [Wellner 1991; Han 2005; Rekimoto 2002] to interact with the map. One-handed methods have also been developed for letting users hold a smart-

phone while performing panning and zooming with the same hand [Miyaki and Rekimoto 2009]. Researchers have also combined gaze and touch [Pfeuffer et al. 2014; Pfeuffer and Gellersen 2016] to reduce the gestures and hand movements on a large (tablet) surface. On smartwatches, the small interaction surface has proved to be challenging for multitouch gestures. A number of sensing techniques have been proposed to use in-air gestures [Laput et al. 2014; Zhou et al. 2016] for panning and zooming on smartwatches. There have also been mechanical means developed to support zooming for smartwatches (e.g., Apple’s digital crown [Macworld 2018] or a mechanical case that acts as a joystick [Xiao et al. 2014]). Sensing the skin under or around the smartwatch [Ogata and Imai 2015; Zhang et al. 2016] is another way to facilitate interaction on smartwatches.

### 3.1.2 Overview+Detail

This scheme, also referred to as *spatial separation* [Cockburn et al. 2008], presents an information space at different scales side-by-side. Figure 3.2(a) shows an example of the classical Google Maps user interface, in which an overview map is embedded at the lower-right corner of a detail map. In space-scale diagrams, overview+detail means combining the view from a different scale to the existing view (Figure 3.1b).

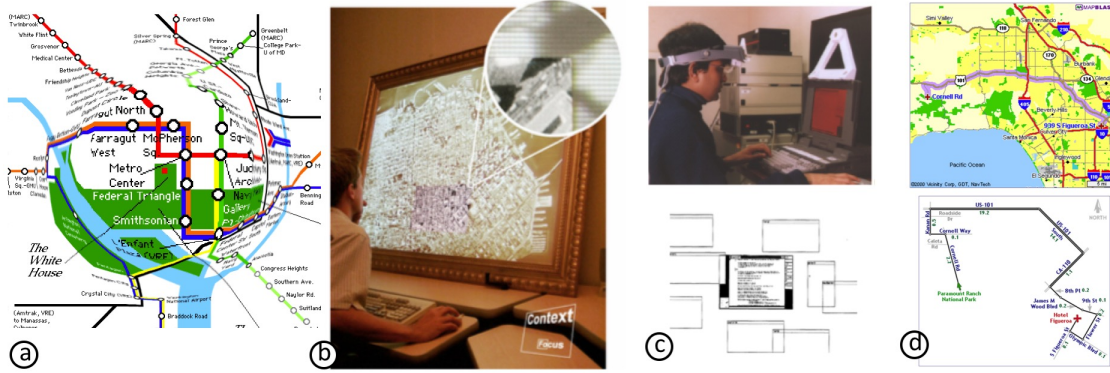
There are several variations of this scheme. Rather than showing information at multiple scales side-by-side, Powers of Ten Thousand [Lieberman 1994] stacks information at different scales on top of each other. Top views are semi-transparent so users can see the information in all layers at once. By tilting the display, users can also separate the information at different scales by motion parallax. DragMag [Ware and Lewis 1995] pro-



**Figure 3.2: Two examples of the overview+detail technique. (a) Classic Google Maps User Interface [Cockburn et al. 2008]. (b) FingerGlass [Käser et al. 2011].**

vides a lens (functioning as a magnified view), that users can interactively move to magnify different portion of the information space. FingerGlass [Käser et al. 2011] is a bimanual technique that allows users to use one hand to specify the area to be magnified (a magnified view is then shown on the display), and use another hand to select targets in the magnified area (Figure 3.2b). Polyzoom [Javed et al. 2012] allows users to interactively build hierarchies of different regions of interest at different scales, and arrange views in a tree.

The size and placement of the overview window needs to be carefully designed, since an overview window occupies valuable display real estate. A carelessly placed overview window could occlude important information on the detailed view. Additionally, large differences between overview and detail could make it difficult to integrate information at different scales. As a rule of thumb, Shneiderman et al. [2016] recommend that the zoom ratio between overview and detail should be within the range from 5 to 30. Beyond the zoom ratio, views of intermediate zoom levels should be provided.



**Figure 3.3: Examples of focus+context visualization techniques. (a) Fish-eye map [Keahey and Robertson 1996]. (b) Focus+Context Display [Baudisch et al. 2001]. (c) Hybrid User Interfaces [Feiner and Shamash 1991] (d) LineDrive [Agrawala and Stolte 2001]**

### 3.1.3 Focus+Context

This scheme aims to blend the seams between overview and detail, usually through the means of modifying the presentation of an information space. Two representative frameworks [Carpendale and Montagnese 2001; Furnas 2006] have been proposed to include a wide range of approaches. Distortion [Sarkar and Brown 1992; Sarkar et al. 1993; Brosz et al. 2013; Apperley et al. 1982] and folding [Elmqvist et al. 2008; Ion et al. 2013] are typically applied to emphasize or deemphasize certain portions of an information space. For example, fisheye views [Furnas 2006; Sarkar et al. 1993] compress the areas around certain focused regions such that focused and contextual views can be fit into a visible window. Alternatively, one may also magnify certain focused regions while keeping the contextual areas intact. Figure 3.3(a) shows an example of a fisheye metro map for Washington D.C. The focus is at the center of the image, which is magnified compared with the rest of the map. The area surrounding the focused region is distorted to smoothly transition from the magnified scale to the normal scale. One common characteristic of the focus+context techniques is that the displayed information space is modified. As a result, it is often difficult to

perform metric measurements from the displayed information. Additionally, Shneiderman et al. [Shneiderman et al. 2016] noted that the zoom ratio in published fisheye examples never exceeds 5.

A similar idea that falls into this category is the mixed use of displays with different resolution capability to display focused and contextual information. A focus+context display [Baudisch et al. 2001] (Figure 3.3b) is a display system that renders the focus region at high resolution and the contextual area at lower resolution. The system takes advantage of a high-resolution desktop display, and a projector that is capable of displaying content on a larger surface, albeit at lower resolution. Feiner and Shamash [1991] uses a head-worn (low-resolution) display to visualize contextual information seamlessly surrounding a desktop (high-resolution) display (Figure 3.3(c)). MultiFi [Grubert et al. 2015] uses a head-worn (low-resolution) display to provide a contextual view of a small map displayed on a smartwatch (high-resolution) display.

Several researchers have proposed algorithms to automatically or semi-automatically create navigation aids by strategically emphasizing and deemphasizing certain parts of an environment. LineDrive [Agrawala and Stolte 2001] is a system that renders routes in the form of hand-sketched route maps. LineDrive emphasizes the local details at decision points, especially at the areas of source and destination, and simplifies the global shape of a route. The rendered route is similar to people’s conception of a route (Figure 3.3d). Kopf et al. [2010] propose a system to generate destination maps, which emphasize the artery roads of a destination while eliminating small paths [Kopf et al. 2010]. Grabler et al. [2008] propose a system to generate tourist maps—perspective maps with 3D models of landmarks and simplified road networks designed for visitors to familiarize themselves



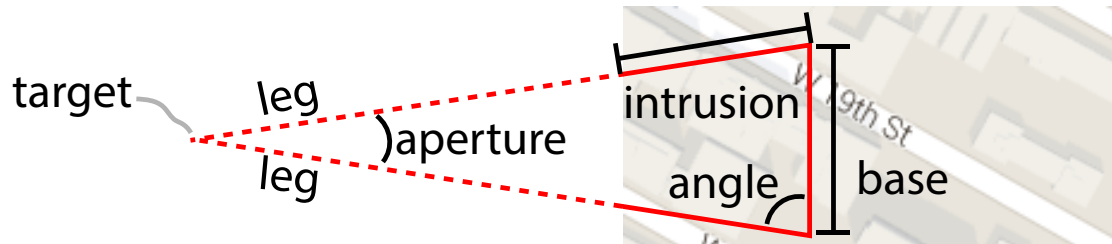
**Figure 3.4: Examples of cue-based techniques. (a) City Lights [Mackinlay et al. 2003]. (b) Halo [Baudisch and Rosenholtz 2003]. (c) Wedge [Gustafson et al. 2008].**

with new regions.

### 3.1.4 Cue-Based

The cue-based scheme uses proxies to selectively visualize POIs that may not be visible in the displayed information space (Figure 3.1d). Several visualization techniques have been proposed to communicate direction or position of off-screen information objects. 3Desque [Miller et al. 1999] and City Lights [Mackinlay et al. 2003] use lines at the edges of a display to indicate contents that fall outside of an operating system desktop (3Desque) or web browser (City Lights). Figure 3.4(a) shows a screenshot of a web browser with City Lights. On maps, arrows and distance labels have been used to indicate the direction of and





**Figure 3.5: Anatomy of a Wedge (solid lines). The target can be located using visual shape completion (dashed lines).**

distance to off-screen POIs [Burigat et al. 2006]. EdgeRadar [Gustafson and Irani 2007] reserves some space at the four edges of the display, and displays dots in the reserved space to visualize objects outside of the display edges. Canyon [Ion et al. 2013] attaches small map insets containing off-screen objects at the boundary of a map, and uses folds to signify distances of off-screen objects.

Halo [Baudisch and Rosenholtz 2003] applies the idea of shape-completion to allow users estimate the location off-screen objects. Figure 3.4(b) shows an example of Halo. The red arcs on the periphery of the display signify off-screen POIs (Figure 3.4b-left). Each off-screen POI “radiates” a circle (termed “halo”) where the centroid of the circle corresponding to the location of the POI (Figure 3.4b-right). The size of the circle is extended in a way such that portion of the circle are visible in the display. Users can estimate the centroids of the partially visible circle by mentally completing the shape of the circles. One downside of Halo is that multiple adjacent halos could overlap (Figure 3.4c-left). Wedge [Gustafson et al. 2008] (Figure 3.4c-right) is an improvement to halo. A Wedge is a partially clipped isosceles triangle, whose invisible apex is located at an off-screen POI. Users estimate the location of that POI by visually completing the shape (Figure 3.5).





**Figure 3.6: Where am I? Localization on a smartwatch form factor using (a) P-Compass, (b) Overview+Detail, and (c) Wedge. The two off-screen POIs are 6 cm (“Hotel”) and 4 cm (“Train Sta.”) from the centroid of the display. (In true scale.)**

### 3.1.5 P-Compass versus the Preceding Four Visualization Strategies

How is P-Compass related to the preceding four strategies for visualizing overview and detail? P-Compass is a hybrid technique that integrates Overview+Detail and cue-based techniques, and can be incorporated into a zooming interface. Figure 3.6 compares P-Compass, Overview+Detail (using an overview map), and Wedge. While P-Compass (Figure 3.6a) and overview map (Figure 3.6b) both add an overview to the underlying detail view, the use of POI cues allows P-Compass to reduce the amount of occlusion. Furthermore, while P-Compass and Wedge (Figure 3.6c) both use proxies to indicate off-screen POIs, the overview provided by P-Compass allows it to communicate the relationship between multiple off-screen POIs.

Much of the cue-based research aims to visualize a potentially large number of off-screen POIs [Baudisch and Rosenholtz 2003; Gustafson et al. 2008; Burigat and Chittaro 2011; Burigat et al. 2012]. Wedge [Gustafson et al. 2008] investigated up to 20 off-screen locations (on a smartphone display emulated on a 19 inch desktop monitor). Burigat et

al. [2012] investigated visualizing up to 40 off-screen locations (on a HTC Desire smartphone). The research goal of P-Compass differs from those of the existing cue-based techniques. The focus of P-Compass is on using as *few* off-screen POIs as possible to help users *establish a reference frame*. (In our iPhone 6s prototype, we showed that P-Compass could use as few as one off-screen POI and the FOV box, or two off-screen POIs to do so.) Furthermore, to our knowledge existing published work [Baudisch and Rosenholtz 2003; Gustafson et al. 2008; Burigat and Chittaro 2011; Burigat et al. 2012] focused on visualizing POIs located at no more than 10cm off the display edge. With P-Compass we are interested in a solution that can scale to distant off-screen POIs (Section 4.2.4).

## 3.2 Navigation and Related UI Tools

Here we review navigation and related UI tools to help users interact with spatial information. We begin with shortcuts, common UI widgets that help users access frequently viewed information. We then discuss using constraints to retrieve information for display and tangible tools to manipulate spatial information. Finally, we review scrolling, direction-pointing tools, and several approaches to generating visual navigation aids (e.g., route maps and tourist maps).

### 3.2.1 Shortcuts for Revisitation

Researchers have long noticed that computer users tend to revisit previously viewed information [Cockburn and McKenzie 2001; Adar et al. 2008; Alexander et al. 2009; Heer et al. 2008]. In document browsers (including web browsers), bookmarks are commonplace

navigation widgets that support revisitation.

Many map applications provide bookmark (shortcut) functionality (e.g., “Your places” in Google Maps), similar to that of document browsers, allowing users to jump to a saved location. Visual bookmarks for *areas*, in the form of “mini maps,” can also be found in research prototypes such as TrailMap [Zhao et al. 2013]. Some map applications and prototypes provide features that allow users to see multiple bookmarked POIs or areas at once. However, users need to assemble a list of locations/areas, apply the “show all” (Bing Maps [Microsoft 2017]) or “union search results” (TrailMap) feature to see all locations at once, and in some cases go through layers of menus in the process (Bing Maps). Users must then repeat similar steps when they wish to view a different set of locations. In contrast, in Chapter 4 we will discuss an interaction technique supported by SpaceTokens called “zoom-to-fit,” which allows users to rapidly make one or more important locations visible without cumbersome setup.

Fundamentally, SpaceTokens differ from existing revisitation mechanisms in two ways. First, previous work follows this “shortcut” paradigm to address revisitation. In contrast, our work takes a different approach, which we refer to as the *constraint-specification* paradigm. We show how SpaceTokens and on-screen locations can be used as *constraints* to specify which locations are to be seen (visibility constraints), as well as where these locations are to be seen (position constraints). Users are free to mix and match different constraints simultaneously. The system automatically adjusts the window on the information space based on the specified constraints.

Second, we explore the idea of making locations first-class citizens of map interaction, in contrast to the conventional approach of treating locations as supporting characters

(e.g., presenting them as menu items that can only be selected). When treated as first-class citizens, locations have broader utility than just revisitation. We demonstrate how locations can also be used as building blocks to address what we call the “linking problem”—essentially establishing relationships between locations. We also show how locations can be used to design a new navigation instrument (SpaceBar) to facilitate efficient interaction with a route.

### **3.2.2 Constraints for Information Selection**

There has been much research on selecting spatial information from a database. Researchers have demonstrated sketch-based interfaces to retrieve images [Jacobs et al. 1995; Cao et al. 2010; Chen et al. 2009], or to use the retrieved images for assisting users in creating artifacts such as drawings [Lee et al. 2011] or photo montages [Eitz et al. 2011]. There are also interactive techniques to select data from a data set [Holz and Feiner 2009; Chuah et al. 1995]. We observe that map revisitation is akin to information selection, for which a user can use locations as constraints to specify a window in a multi-scale information space [Furnas and Bederson 1995]. Maps, however, have not received much attention in this research area. We explore the idea of applying the constraint technique to map revisitation.

### **3.2.3 Tangible UI Tools**

Tangible User Interfaces (TUIs) [Ishii and Ullmer 1997] use physical props as tangible controls to manipulate digital information. TUIs have been used to support revisitation



**Figure 3.7:** In MetaDesk [Ullmer and Ishii 1997], physical props are used to manipulate a map.

for maps. For example, MetaDesk [Ullmer and Ishii 1997] (Figure 3.7) recognizes the identity of a physical model of a building that is placed on a tabletop rear-projected map and adjusts the map so that the representation of the building’s footprint coincides with the model (i.e., the building model is used as a position constraint). In other words, MetaDesk allows users to manipulate a map by using physical models to specify *position* constraints (i.e., *where* locations should be seen). As will be discussed in Chapter 5, there are other constraints called *visibility* constraints (i.e., *what* locations should be seen). SpaceTokens support users to specify visibility and position constraints individually or in combination, or the use of on-screen and off-screen locations as constraints.

Additionally, there are issues with using such physical props for revisitation. First, a passive physical prop provides a physical constraint on the information space; a user will need to remove this physical constraint if they wish to shift the window of the map to a completely new location. Second, while a user can use a physical prop to manipulate the information space, directly manipulating the information space itself (e.g., through pan, zoom, and rotation) does not update the position, scale, or orientation of the physical prop.

(Actuated props [Pangaro et al. 2002] could address updating position and orientation, though modifying scale would be more difficult.) In contrast, SpaceTokens are interactive software-based widgets designed for revisitation, for which we also propose a set of interactions.

### **3.2.4 Scrolling, Direction Pointing Tools, and Other Navigation Aids**

A scrollbar is a familiar navigation tool for document readers and image viewers. It serves as a 1D overview to a document or an image. The position and size of the “elevator” indicator communicate the relationship between the visible portion of a document to the entire document. There has been much work on adding annotations to a scrollbar [Hill et al. 1992; Olsen 1992; Laakso et al. 2000]. Speed-dependent automatic zooming [Igarashi and Hinckley 2000], described earlier, couples the zoom level of a document to the speed of scrolling. Content-aware kinetic scrolling [Kim et al. 2014] displays selective contextual information while users scroll to help users locate interesting targets more efficiently. OrthoZoom scroller [Appert and Fekete 2006] allows users to adjust scrolling precision by pulling away (in the orthogonal direction to the scrolling direction) from the scrollbar while scrolling. Content-aware scrolling [Ishak and Feiner 2006] allows users to traverse a multi-column document based on the flow of its content, rather than its pages.

A compass is a familiar navigation tool that indicates the direction of north. Several researchers have presented variants of compasses that point to POIs instead of magnetic north [Feiner et al. 1997; Egenhofer 1999; Suomela and Lehtikainen 2000; Chittaro and Burigat 2004]. There are also devices that provide tactile stimulation, such as a tactile

belt or tactile actuator on a smartphone [Tsukada and Yasumura 2004; Heuten et al. 2008; Rümelin et al. 2011], to indicate the direction of a POI. In addition, some research uses sound to indicate direction [Loomis et al. 1998; Holland et al. 2002]. However, P-Compass is more than a specialized compass; P-Compass is a navigation instrument that aims to help users establish a reference frame, which goes beyond direction indication.

There has been work on generating special-purpose navigation aids, e.g, personalized maps, routes. Patel et al. [2006] propose a system that generates personalized routes, travel directions that include personalized a priori POIs as intermediate way points. Since users are already familiar with how to reach certain known POIs, personalized routes aim to make it easier for users to understand travel directions. Grabler et al. [2008] develop algorithms to automatically generate tourist maps that highlight certain landmarks in a new environment for users to understand the environment easily. Kim et al. [2016] propose a system to generate visual analogies to describe the abstract idea of size or distance by using familiar information. For example, the area of park *ABC* (unfamiliar) is about 3.5 the size of the University *XYZ* campus (familiar). Landmark-to-location [Baudisch and Krumm 2013] is a system that generate textual descriptions to describe an unknown location by leveraging landmarks. These special-purpose navigation aids explore the idea of personalization, but for different research problems.

## Chapter 4

---

### *Personalized Compass*

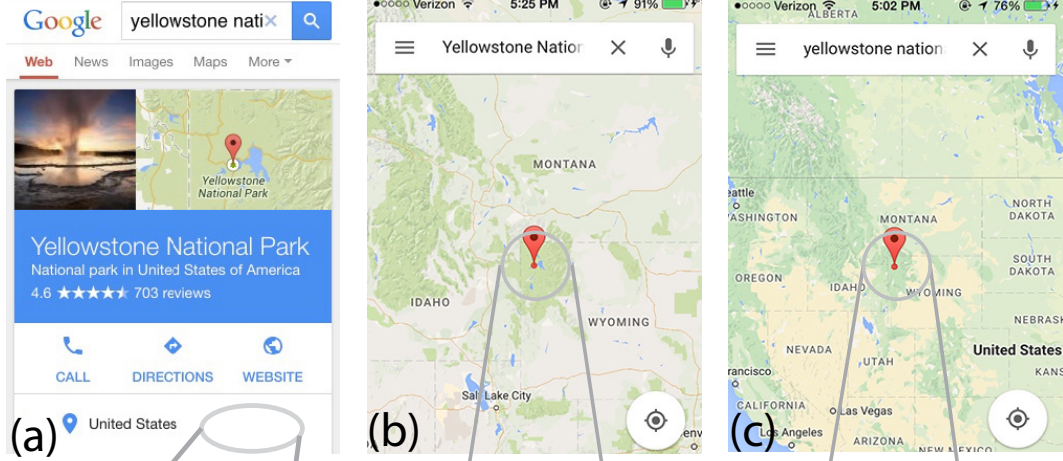
Maps on mobile/wearable devices often make it difficult to determine the location of a POI. A POI  $x$  may exist outside the map or on a background with no meaningful cues. In these two situations, a map user may find herself asking the question, “Where is  $x$ ?” For lack of a better term, we will refer to this as the “Where is  $x$ ?” problem. In this chapter, we introduce Personalized Compass to address this problem. The key idea behind Personalized Compass is the use of personal *a priori* POIs to establish a reference frame within which a POI can be localized. We analyze the characteristics of Personalized Compass and the existing Wedge technique, and report on a user study comparing them.

#### **4.1 Introduction**

As the display area of an information space becomes smaller, information is often left out. When important information required by a spatial cognition task is not displayed, an otherwise reasonable task can become difficult (e.g., route planning to an off-screen destination [Gustafson et al. 2008]). A small display also increases the probability of Desert Fog [Jul and Furnas 1998]—a condition in which the display information is devoid of meaningful cues to assist users in making decisions. These two issues are often discussed separately in the literature. We refer to these issues together as the “Where is  $x$ ?” problem,



Problem



Solution



**Figure 4.1: Where is Yellowstone National Park? Answering this question from a mobile search result often involves (a–b) unnecessary application switching and (b–c) repetitive zooming, as shown in these smartphone screenshots. A P-Compass uses important POIs to provide a first-order approximation to the answer, and can function (i) independently or (ii–iii) with a map.**

occurring whenever some POI  $x$  is off display, or on display but with Desert Fog present.

The “Where is  $x$ ?” problem is frequently encountered when using a map. Driven by the growth of mobile and wearable devices, as well as the demand of accessing geospatial data on these devices, the ever decreasing size of maps makes the “Where is  $x$ ?” problem an increasingly frustrating issue.

Take location search as an example. Figure 4.1(a) shows a Google search result for the keywords “Yellowstone National Park” on a smartphone. The embedded stamp-size map is rich in information. Ironically, due to Desert Fog, it fails to communicate *where* Yellowstone National Park is. To find out the answer, users may tap the stamp-size map to switch to a map app (Google Maps) and then continue fighting Desert Fog by repeatedly zooming

out (Figure 4.1b–c). When a satisfying overview is eventually reached (Figure 4.1c), details such as Yellowstone Lake are no longer visible, and the original application context (the web search engine) has been dismissed. Similar “Where is  $x$ ?” problems are often encountered in social media, websites, and local business reviews—whenever a stamp-size map is presented. For obvious reasons, the issue occurs even when the area of a map reduces to zero, as when users try to determine the location indicated by a geo-tag, a zip code, an address, or GPS coordinates. The “Where is  $x$ ?” issue is ubiquitous, with or without a map.

Now suppose we ask a knowledgeable person the same question, “Where is Yellowstone National Park?” They may first gauge the geographic knowledge of the questioner, either via a mental assessment or verbal questions. Based on the initial evaluation, they may then describe the location of Yellowstone National Park in relation to other geographical entities (e.g., Yellowstone National Park is located  $k$  miles northwest of  $abc$ ; Yellowstone National Park is located between  $abc$  and  $def$ ).

We make two observations based on this scenario. First, the person constructs an answer from assumed *a priori* knowledge of the questioner. Second, in terms of information complexity (measured in bits), the information provided in the answer is very compact, to the degree that it can be effectively transmitted via language. Put a different way, an approximated answer to the “Where is  $x$ ?” question, provided in a frame of reference known by the questioner, requires surprisingly little information.

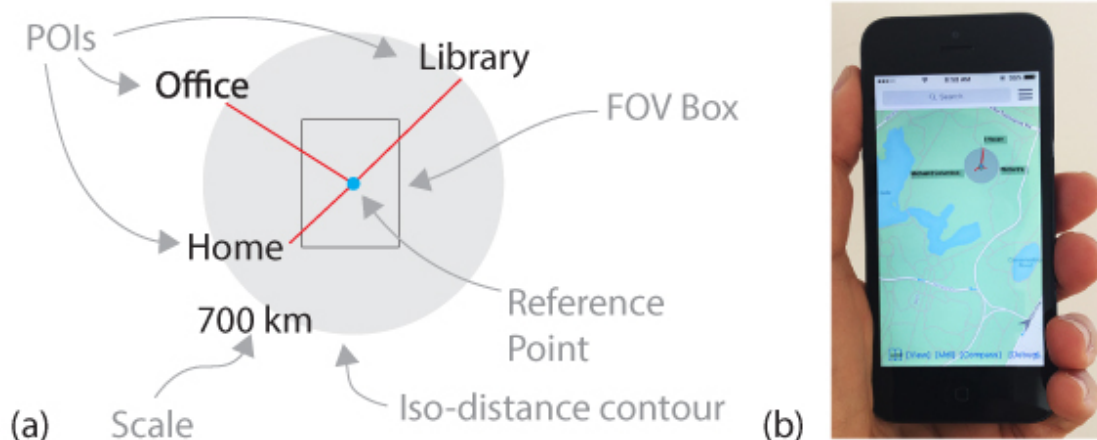
Inspired in part by these observations, we present Personalized Compass (P-Compass), a compact graphical representation that communicates the location of  $x$  with the support of personal *a priori* POIs. Figure 4.1(i) shows P-Compass as a standalone location indicator.

This example uses three major U.S. cities to establish a reference frame, which in turn indicates the location of Yellowstone National Park. Note that a P-Compass can occupy a relatively small footprint.

P-Compass can also be integrated into the zooming interface, as seen in Figure 4.1(ii–iii). The black rectangle in the center of P-Compass indicates the boundary of the visible map. As a user zooms into or out of the map, the size of the black rectangle updates accordingly, providing immediate feedback on the relative scale of the visible map to the distances to the three major cities.

We created a customized iOS/OS-X map app that integrates P-Compass and Wedge [Gustafson et al. 2008] (a well-known technique to visualize off-display POIs, described in Section 3.1.4). Wedge has been frequently used as a baseline technique in previous user studies (e.g., [Burigat and Chittaro 2011; Burigat et al. 2012; Ion et al. 2013]), which is why we chose it for comparison. We then used this app regularly in daily life for over a year. Based on our field experience, we analyzed the characteristics of Wedge and P-Compass, and designed and ran a formal user study to compare the two. To ensure a fair comparison, our study used a common set of fictional POIs, instead of personalizing the POIs for each participant.

Our analysis and study reveal that P-Compass and Wedge complement each other. Wedge’s strength, which is P-Compass’s weakness, is to place an off-screen POI at its absolute location. However, the need to locate an off-screen POI makes it difficult to perform certain inference tasks, which are better carried out with P-Compass. In addition, as the distance to an off-screen POI increases, and/or the size of a display decreases, we show that the benefits of P-Compass eventually outweigh those of Wedge.



**Figure 4.2: (a) Anatomy of a P-Compass. (b) P-Compass prototype.**

## 4.2 Personalized Compass

The anatomy of a P-Compass can be seen in Figure 4.2(a). A P-Compass consists of a reference point, and one or more needles to communicate the direction and distance to POIs. When a P-Compass is shown on a map, a field of view (FOV) box—a 2D scale, can be added as an additional cue to indicate the location of an off-display POI with respect to the boundary of the display. The needles can extend into the FOV box, to indicate the reference point, as shown in Figure 4.2, or stop at the edge of the FOV box, as shown in Figure 4.1. As the user zooms into the map, the FOV box decreases in size, eventually becoming a dot. An optional iso-distance contour assists the user in comparing distances to POIs (and separates P-Compass from the background), while an optional numeric scale label denotes the distance from the reference point to the iso-distance contour.

The central idea of P-Compass is the use of personal *a priori* POIs to establish a frame of reference. Personal POIs have been known to play an important role in cognitive maps [Tolman 1948] (mental representations of environments). As a POI—be it a landmark, an intersection, a store, or a city—becomes known to a person, the POI becomes

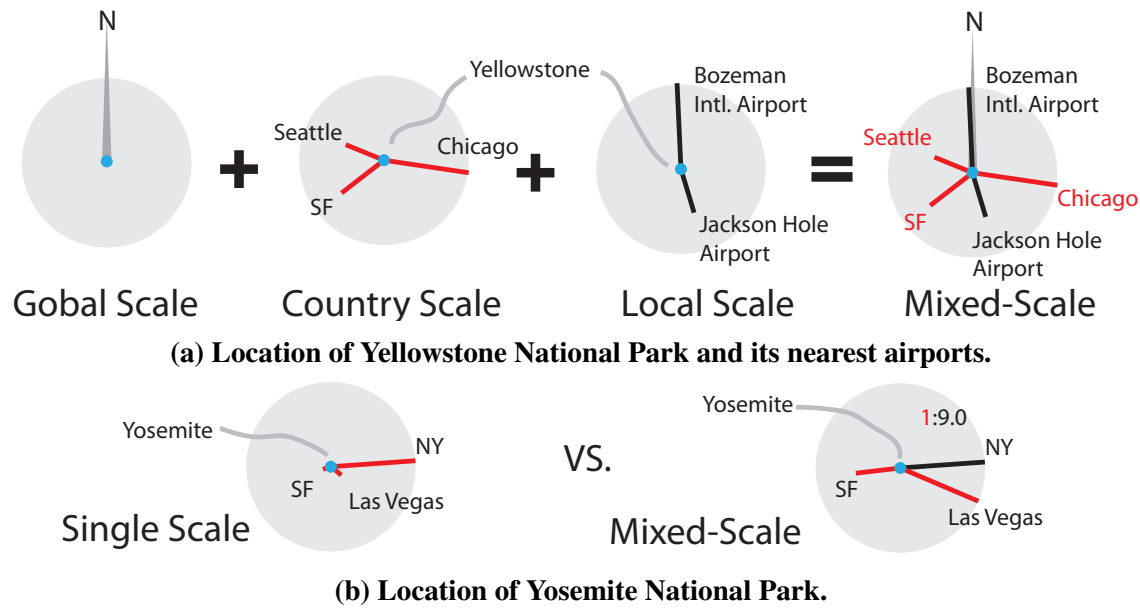
a reference with which they reason and continue to develop their cognitive maps. The use of personal POIs can effectively provide a frame of reference for the user [Golledge 1999; Sorrows and Hirtle 1999], while significantly reducing the amount of information needed to be shown on a graphical representation. As a result, a P-Compass occupies a compact footprint, and can work alone or in conjunction with a map.

#### 4.2.1 Proof-of-Concept Prototype

We implemented a custom iOS/OS-X map application incorporating a P-Compass (Figure 4.2b). The application, using Apple MapKit, supports zoom, pan, rotate, and location search. The P-Compass, implemented in OpenGL, updates dynamically as the user interacts with the map, and can be resized and translated with simple gestures.

We manually entered a master list of personal POIs for our field prototype. We applied a simple greedy algorithm (detailed in Appendix A.1) to select  $n$  POIs that are roughly equally distributed in orientation to automatically form a P-Compass. Alternatively, the POIs can be selected manually by the user. We also implemented Wedge in the same application. The user can toggle between P-Compass and Wedge, or show both simultaneously.

For over a year, the author used this application in daily life on an iPhone, as well as an iPad, laptop, and desktop, to explore the characteristics and design space of P-Compass. Below, we describe the design and analysis of P-Compass, with references to real-world scenarios as well as comparisons with Wedge.



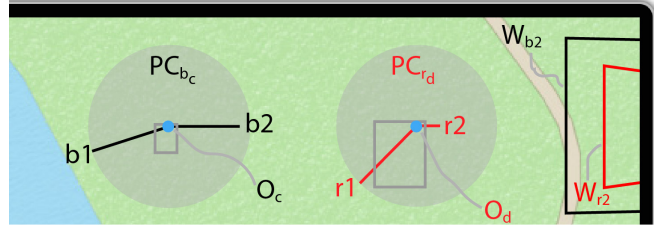
**Figure 4.3: Stackability.** Information at multiple scales is combined into a single P-Compass. (a) Information at each scale is read independently. (b) Information at all scales is related.

#### 4.2.2 Stackability

P-Compass allows multiple scales of information to be stacked on top of each other, as shown in Figure 4.3. Thus, a compass can be converted into a P-Compass *in place* simply by stacking relevant information on top of it. P-Compass makes it possible for a user to access information at different scales simultaneously. Figure 4.3(a) shows three scales combined in a single P-Compass. POIs at each scale serve a different purpose—global-scale information indicates the orientation of the diagram; country-scale information indicates the location of Yellowstone National Park; local-scale information visualizes which local airport is closer.

Representing *related* information in multiple scales can enhance readability. The P-Compass at the left of Figure 4.3(b) displays all POIs at the same scale. Since Yosemite National Park is much closer to San Francisco and Las Vegas than to New York City, two of

**Figure 4.4:** This figure and Figure 4.5 together form two corners of a virtual display nearly the size of this page. See Figure 4.5 caption for details.

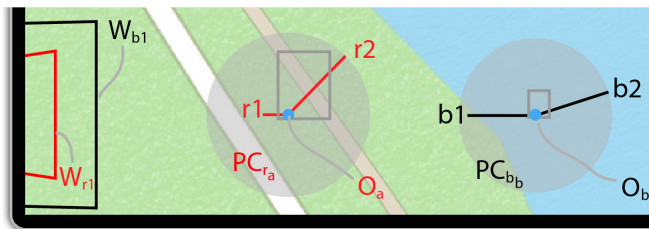


the compass needles are nearly invisible. The P-Compass at the right presents a multi-scale view of the same information. The scale indicator is essential in this case, and indicates the ratio of the two scales is 1:9. Presenting information at mixed-scale can increase the complexity of a visualization. Design techniques (e.g., color coding) can be applied to distinguish information at different scales.

### 4.2.3 Optimized for Personal Use

In the context of a mobile map application, there is often one particular POI, a reference point, that a user cares about (e.g., the location of the user in a “you-are-here” map). P-Compass, a centralized representation, is optimized for tasks involving  $n$  (off-screen) POIs for a single reference point.

Let  $VD$  be a virtual display formed by Figures 4.4–4.5.  $VD$  contains four Wedges and four P-Compasses, together visualizing four off-screen POIs (see Figure 4.5 for definitions



**Figure 4.5: P-Compass vs. Wedge.** This figure and Figure 4.4 are the lower-left and upper-right corners of a virtual display. Wedges (prefixed with “W”) and P-Compasses (prefixed with “PC”) indicate locations of four off-display POIs,  $r1$ ,  $r2$ ,  $b1$  and  $b2$ . P-Compasses are better for drawing inferences from POIs, while Wedges are better for locating POIs.

of terms). It is trivial to use  $PC_{r_a}$  (Figure 4.5) to draw inferences from  $r1$  and  $r2$  with respect to reference point  $O_a$  (e.g., between  $r1$  and  $r2$ , determining which one is closer and by how much).  $PC_{r_a}$  also makes it easy to estimate the direction of  $r1$  or  $r2$ —a common pedestrian and vehicle navigation task. In contrast, it could be difficult to perform these tasks using  $W_{r1}$  (Figure 4.5) and  $W_{r2}$  (Figure 4.4), since a user cannot draw inferences from a POI without first estimating its absolute location.

Wedge is optimized for representing a POI that is near the boundary of a display.  $W_{r1}$  provides direct visual guidance near the edge of  $VD$  so that a user can determine the absolute location of  $r1$ . Knowing the absolute locations of off-screen POIs can help a user perform tasks involving  $m$  (off-screen) POIs for  $n$  reference points; for example, determining which of the  $n$  on-screen hospitals is farthest away from the  $m$  off-screen traffic jams [Gustafson et al. 2008].

In comparison, it could be difficult to use a P-Compass to place a POI at its absolute location. Using  $PC_{r_a}$ , a user will need to first estimate the relative location of  $r1$  with respect to the FOV box, and then transfer the estimate to the actual scale. In addition, two P-Compasses, each optimized for a different reference point, could differ significantly, as can be seen from comparing  $PC_{r_a}$  and  $PC_{r_d}$ . It could be challenging to use a P-Compass to perform tasks with respect to a non-optimized point.

This reference-point dependency diminishes as the distance of an off-screen POI increases. Comparing the two pairs of P-Compasses  $PC_{r_a}$  and  $PC_{r_d}$ ,  $PC_{b_b}$  and  $PC_{b_c}$ , reveals that distant off-screen POIs make the latter pair more similar to the former pair. In other words, for distant off-screen POIs, a single P-Compass is approximately valid for all refer-



ence points on a display<sup>1</sup>. On the other hand, distant off-screen POIs significantly reduce the usability of Wedge. It becomes impossible to use  $W_{b1}$  and  $W_{b2}$  to estimate the locations of  $b1$  and  $b2$ .

In short, P-Compass is better for drawing inferences from off-screen POI(s) with respect to a single reference point, while Wedge is better for locating a nearby off-screen POI. As the distance to an off-screen POI increases, the benefits of P-Compass eventually outweigh those of Wedge.

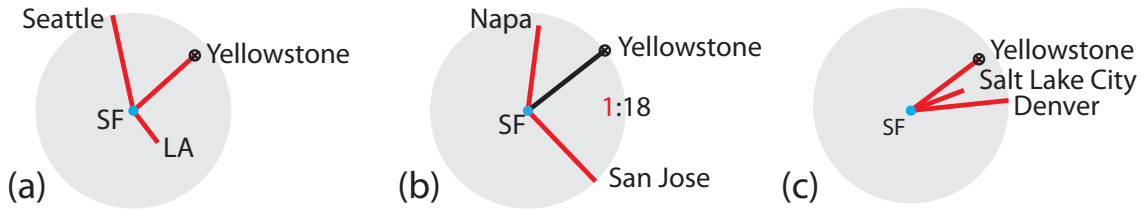
#### 4.2.4 Issues of Scalability

Scalability refers to how the performance of a visualization technique is affected by the distance to an off-screen POI [Gustafson 2008]. While scalability is rarely discussed, it is important for two reasons: Not all off-screen POIs are near the display, as often assumed in the literature; further, with sufficient zoom any nearby off-screen POI becomes distant.

Wedge and P-Compass both have issues with scalability. Wedge is limited mostly to nearby off-screen POIs. As the distance to an off-screen POI increases, the angles between the base and the two legs increase nonlinearly and the rate of change quickly becomes indistinguishable. Beyond a certain distance, the two legs of a Wedge essentially appear parallel [Gustafson 2008]. The FOV box of a P-Compass, which becomes visually indistinguishable from a point for sufficiently distant POIs, does not scale either. In a sense, both approaches aim to *extend* the effective area of a display. Therefore, we refer to them as *display-extension* methods.

---

<sup>1</sup>A good example is the conventional magnetic compass: it is correct for all reference points on a local map regardless of its placement.



**Figure 4.6:** (a) Using supporting POIs to *extrapolate* an unknown POI (Yellowstone). (b–c) Two alternatives: (b) Emphasizes nearby POIs (mixed-scale representation); (c) Emphasizes POIs close to Yellowstone.

The extended display area achieved this way is finite. Simply put, the probability that an arbitrary off-screen POI falls into the small extended display area could be small. Most important, while it may not seem obvious, display extension methods require a relatively large display. The smaller the display area, the smaller the extended display area that can be achieved. (The Wedge max base length and P-Compass FOV box size are both bounded by the size of the display.) Clearly, a different approach is needed to visualize off-screen POIs.

## 4.2.5 Using POIs to Overcome Scalability

P-Compass achieves scalability by using *a priori* POIs as references to communicate the location of an unknown POI  $x$ . We refer to this method as the POI-reference method. In the “Where is Yellowstone National Park?” example (Figure 4.1), multiple POIs are enlisted to *interpolate* an on-screen  $x$  (Yellowstone National Park). A similar strategy could be applied to *extrapolate* an off-screen  $x$ . Figure 4.6(a) shows a P-Compass in which the reference point is at San Francisco, and the off-screen  $x$  (Yellowstone National Park) is at the periphery. Unlike display-extension methods, this strategy is invariant to the distance of POIs, the scale of the information space, and the size of a display.

The use of POIs creates an interesting design space for P-Compass. Different P-

Compasses can answer the same “Where is  $x$ ?” question. A system can tailor the message of a P-Compass through its choice of POIs. Figure 4.6(b–c) shows two alternatives to Figure 4.6(a). Note that different distributions (distance, orientation) and numbers of POIs can affect the estimation accuracy of  $x$ . While our initial prototype uses a simple algorithm to select POIs, we leave exploration of this space to future work.

## **4.3 User Study**

### **4.3.1 Apparatus**

We performed a user study (under IRB-AAAO1121) that uses the prototype application described earlier; however, all test cases shown to participants are static, with all map interactions disabled. Two types of displays were used in the study: (1) A Dell 24-inch monitor (U2412M) displayed instructions and (for one task) an emulated smartphone to match a previously proposed setup; (2) An Apple iPhone 5 supported additional smartphone tasks (application area: W 50 mm  $\times$  H 75 mm) and also emulated a circular smartwatch (36 mm diameter) for smartwatch tasks.

### **4.3.2 Design and Tasks**

We are interested in how P-Compass and Wedge could assist users in resolving the “Where is  $x$ ?” problem in daily life. Based on the author’s year of field experience, we developed a “day in the city” scenario comprising five navigation tasks. A preliminary pilot study of nine participants, who were researchers not associated with our lab, was first conducted

to evaluate the tasks and gather feedback. Below we describe the five tasks and their motivations.

### **LOCATE: Where is the subway station?**

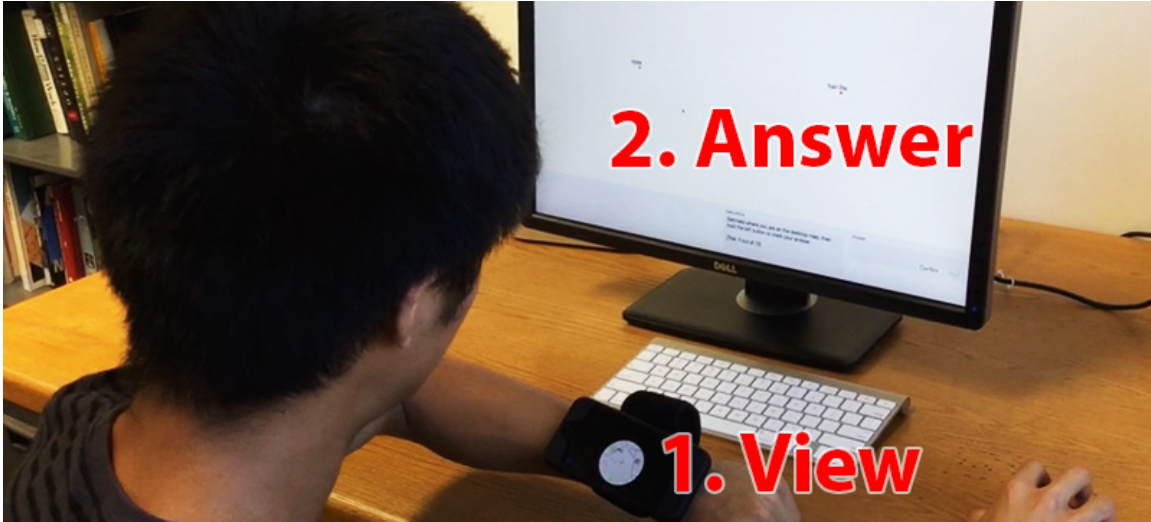
LOCATE asks a user to estimate the location of an off-screen subway station. This task is similar to one proposed by Gustafson et al. [2008], and, like it, is performed with an emulated smartphone on a desktop monitor. On a blank canvas, a user sees a wireframe rectangle representing the boundary of the smartphone display (W×H: 67 mm×105 mm, 1.3× the size of the custom map app on an iPhone 5), along with a visualization to indicate the location of a subway station (Figure 4.7a). The user must move a mouse cursor to the estimated location of the subway station.

The two within-subject factors in this task are visualization type (P-Compass and Wedge) and off-screen POI distance (six distances: 5–30 cm). We constrained all off-screen POIs to be on the short axis (the axis passing through the centroid of the display and parallel to the shorter display edge). The goal is to investigate *scalability*—how the error may change as a function of off-screen distance. From our field experience, we noticed that an off-screen POI was often not located near the display edge, yet studies of Wedge [Gustafson et al. 2008; Burigat and Chittaro 2011; Burigat et al. 2012] focus mostly on nearby off-screen POIs.

### **DISTANCE: How far is the subway station?**

DISTANCE asks a user to estimate the distance to an off-screen subway station in terms of an integral multiple of the half-screen width—a simplified scale. The user sees the





**Figure 4.8: Setup for LOCALIZE and LOCATE+. A wrist-worn iPhone 5 is used to emulate a smartwatch.**

typically able to show that “you are here,” they do not show the *direction* to a POI from that location.

The three within-subject factors in this task are visualization type (P-Compass and Wedge), the distance to a POI (near and far) and the location of a POI (edge and corner). Near POIs are roughly 5 cm from the display edge, and far POIs are roughly 20 cm from the display edge.

### **LOCALIZE: Where am I? LOCATE+: Where is the Cafe?**

In these two tasks, the user needs to estimate an unknown location  $x$  from the locations of two known POIs,  $P_1$  and  $P_2$ . The user wears a smartwatch (emulated on a wrist-worn iPhone 5), which displays the relationship between  $x$ ,  $P_1$  and  $P_2$ . The user marks the estimated location of  $x$  on a desktop monitor, on which only  $P_1$  and  $P_2$  are displayed on a blank background (Figure 4.8).

In LOCALIZE,  $x$  is visible and located at the center of the emulated smartwatch ( $P_1$

and  $P_2$  are both off-smartwatch; see Figure 4.9a). The user's task is to *interpolate*  $x$  on the desktop monitor. In LOCATE+,  $P_1$  is visible and located at the smartwatch center ( $x$  and  $P_2$  are both off-smartwatch; see Figure 4.9b). The user must *extrapolate*  $x$  on the desktop monitor.

LOCALIZE is motivated by the scenario in which a user must determine where she is with respect to two off-screen landmarks. The mobile location search scenario (Figure 4.1) provides additional motivation. LOCATE+ is motivated by the same scenario as LOCATE. Rather than asking the user to estimate the *physical* location of  $x$ , LOCATE+ asks the user to estimate  $x$  with respect to a reference frame consisting of POIs.

To study how the distribution of  $P_1$  and  $P_2$  may affect estimating  $x$ , the two tasks include 12 different  $P_1$  and  $P_2$  locations as exploratory conditions (detailed in the Results section). However, visualization type (P-Compass and Wedge) is the only within-subject factor used in data analysis.

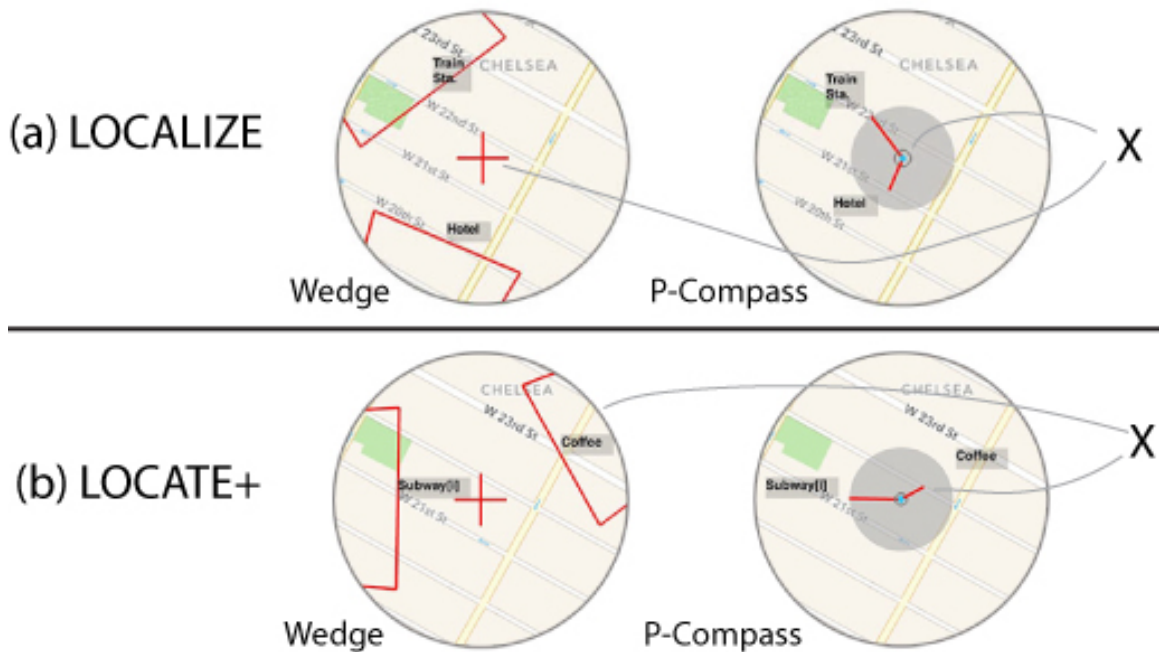


Figure 4.9: Smartwatch screenshots: (a) LOCALIZE; (b) LOCATE+.

### 4.3.3 Fair Comparison

How can one control the sizes of Wedge and P-Compass to allow them to be compared fairly? In addition, for a rectangular display, the location of an off-screen POI,  $x$ , could affect the fairness of a comparison—the best case scenario for P-Compass is when  $x$  lies along the long axis (the axis passing through the centroid of the display and parallel to the longer display edge), since the long edge of the FOV box would then be used as the standard for estimation. In contrast, the best case scenario for Wedge is when the object lies along the short axis, since a Wedge would then have the maximal length to use for its base. We are interested in the general use of P-Compass and Wedge, not their optimal cases. So we fix compass radius at  $\frac{1}{8}$  screen width, Wedge intrusion at  $\frac{1}{7}$  screen width, and maximal Wedge base at  $0.9 \times \min(W, H)$ . For LOCATE and DISTANCE, we constrain all POIs on the short axis.

In our experiment, Wedge covers a greater area than P-Compass. (The biggest Wedge occupies roughly  $2.7\times$  the area of the P-Compass.) We considered reducing maximal base length so that both cover the same area. However, the two Wedge legs could then become almost parallel, making it extremely difficult to estimate the location of an off-screen POI.

For LOCALIZE and LOCATE+, we use a circular watch face (36 mm diameter), rather than a rectangular display. These two tasks require users to estimate the distance ratio and directional difference of two off-screen POIs. Our pilot study indicated it could be challenging to use Wedge on a rectangular display to estimate the direction of off-screen POIs. With a circular display, we constrain the bases of all Wedges perpendicular to lines radiating from the display centroid. This Wedge implementation is optimized for direction



estimation, but may result in overlap for POIs with small angular differences. We chose POIs with big enough angular differences to eliminate Wedge overlap. All Wedges are guaranteed to have a fixed intrusion of 4.5 mm ( $\frac{1}{4}$  of the radius). The maximal base is fixed at 22 mm, which is 90% of the chord when the chord is  $\frac{1}{4}$  radius away from the centroid. The maximal needle length of the P-Compass is fixed at 6.25 mm. A single Wedge could occupy up to 13.2% of the display area (two Wedges are shown in each trial), while a P-Compass occupies 12% of the display area.

While it is possible to scale Wedge **Intrusion**, or **Base** length (IB for short; see Figure 3.5 for definitions) to directly reflect the true off-screen distance ratio between POIs, this method requires some of the Wedges to reduce their IBs. Doing so could negatively impact the accuracy of Wedge, and goes against the suggestions made by Gustafson et al. [2008]. Additionally, this (off-screen distance) encoding scheme does not help users infer the relationship between POIs, especially with respect to an on-screen reference point. For example, to know the distance ratio of a set of off-screen POIs with respect to an on-screen reference point, users will still need to mentally combine the on-screen distance and the off-screen distance for each off-screen POI.

One may argue that it is also possible to scale IB to directly reflect the *true* distance ratio between each POIs with respect to a point. This (true distance) encoding scheme could negatively impact the accuracy of Wedge, as mentioned earlier. Furthermore, this encoding scheme could cause confusion for a non-circular display. For example, on a smartphone display with unequal width and height, two off-screen POIs of the same distance from an on-screen reference point, one on the short axis and other on the long axis, could have different off-screen distance. Yet the two Wedges, representing these two off-screen POIs,

will have the same IBs according to the (true distance) encoding scheme. Therefore, we implement Wedge by following the algorithm suggested by Gustafson et al. [2008], so each Wedge uses its maximal IB.

#### 4.3.4 Hypotheses

We formulated five hypotheses:

**H4.1.** *For LOCATE, Wedge will yield higher accuracy for close POIs, while P-Compass will yield higher accuracy for distant POIs.* Shape completion (Wedge) is effective for close POIs, in which case only a small portion of the triangle is hidden. For distant POIs, the mostly hidden triangle, the nearly parallel Wedge legs, and the nonlinear angle-to-distance mapping<sup>2</sup> could put Wedge at a disadvantage, in comparison with P-Compass, which offers a linear distance cue<sup>3</sup>.

**H4.2.** *For DISTANCE, P-Compass will yield more accurate estimates.* Users can use P-Compass as an abstract overview map to perform distance estimation. With Wedge, however, users need to first estimate the off-screen POI's location, memorize the location, and then use the scale to measure the distance. This process induces heavier cognitive load and is more error prone.

**H4.3, H4.4.** *For LOCALIZE (H4.3) and LOCATE+ (H4.4), P-Compass will yield more accurate results.* P-Compass is a centralized reference-point representation that allows users to see the relationship among multiple POIs at once, bypassing the need to first estimate physical locations of off-screen POIs, as in the case of Wedge.

---

<sup>2</sup>Wedge angles scale nonlinearly as the distance to a POI changes.

<sup>3</sup>The ratio (size of FOV box)/(compass needle length) scales linearly as the distance to a POI changes.

**H4.5.** *For all tasks except LOCATE, users will prefer P-Compass.* The difference between LOCATE and the other tasks is that LOCATE requires a user to estimate the physical location of an off-screen POI, which is Wedge’s strength. For the other tasks, knowing the physical location of an off-screen POI is not a prerequisite, and may even complicate the tasks. The map-like property of a P-Compass makes the other tasks easier.

Since direction indication is exactly what a compass is designed for, we did not formulate an hypothesis for ORIENTATION, but did include it as an investigational task and report its results for completeness. In addition, it is possible to modify the Wedge layout algorithm to optimize direction estimation with respect to a reference point. (In fact, we did so for LOCALIZE and LOCATE+, as mentioned earlier.) Potential issues for this alternative layout algorithm are overlaps and increased Wedge footprint (for rectangular displays), which the original Wedge layout algorithm [Gustafson et al. 2008] tries to avoid.

### 4.3.5 Procedure

26 participants (13 female), ages 20–39 ( $\bar{x} = 25.5$ ,  $s = 4.9$ ), were recruited from the general population pool of our institution for a single-session (one hour) experiment in return for a small cash compensation (\$15). All participants were familiar with mobile map applications (e.g., Google Maps). All except one owned a smartphone and used it multiple times a day. Only one had a smartwatch and used it daily.

Each participant was seated at a desk for all tasks, and instructed to imagine performing the tasks as if in an outdoor urban environment—taking a quick glance at a visualization and then providing rough estimations. Participants were also told all POIs and the back-

ground image were fictional. For smartphone/smartwatch tasks, participants were asked to hold/wear the device; they were not allowed to place the device on the desk or use both hands to assist in measurements, as this would not be possible in many outdoor situations.

The study was blocked by visualization type; half the participants started with tasks using Wedge, and the other half started with tasks using P-Compass. At the start of each block, the study coordinator introduced the idea of the corresponding visualization. The participant then followed on-screen instructions for a practice block for all tasks using the corresponding visualization (3 LOCATE + 4 ORIENTATION + 3 DISTANCE + 3 LOCALIZE + 3 LOCATE+ = 16 practice trials per visualization). At the end of each practice trial, the participant could view the ground truth, but not during the actual trials. The participant next followed on-screen instructions to complete one block for each of the five tasks.

Task blocks were fully counterbalanced, with the constraint that LOCATE, ORIENTATION and DISTANCE were grouped together, and LOCALIZE and LOCATE+ were grouped together (to minimize the number of device switches). The conditions of the non-visualization factor were randomized. Each participant was given a total of  $2 \text{ (visualization)} \times (6 \text{ LOCATE} + 16 \text{ ORIENTATION} + 6 \text{ DISTANCE} + 12 \text{ LOCALIZE} + 12 \text{ LOCATE+}) = 104$  timed trials.

## 4.4 Results

Error size and task completion times were collected during the study (Tables 4.1–4.2). Since error definition differs by task, it is described along with the results of each task. Completion time is computed from trial start to participant answer. All analyses were

	<b>P-Compass</b>	<b>Wedge</b>
LOCATE	64.89 mm (53.70)	63.30 (48.96)
DISTANCE	2.02 x (2.28)	2.96 (2.49)
ORIENTATION	0.44 deg (0.42)	6.30 (6.01)
LOCALIZE	17.82 mm (11.65)	29.29 (16.87)
LOCATE+	14.84 mm (8.82)	34.67 (20.74)

**Table 4.1: Summary of error results (s.d.).**

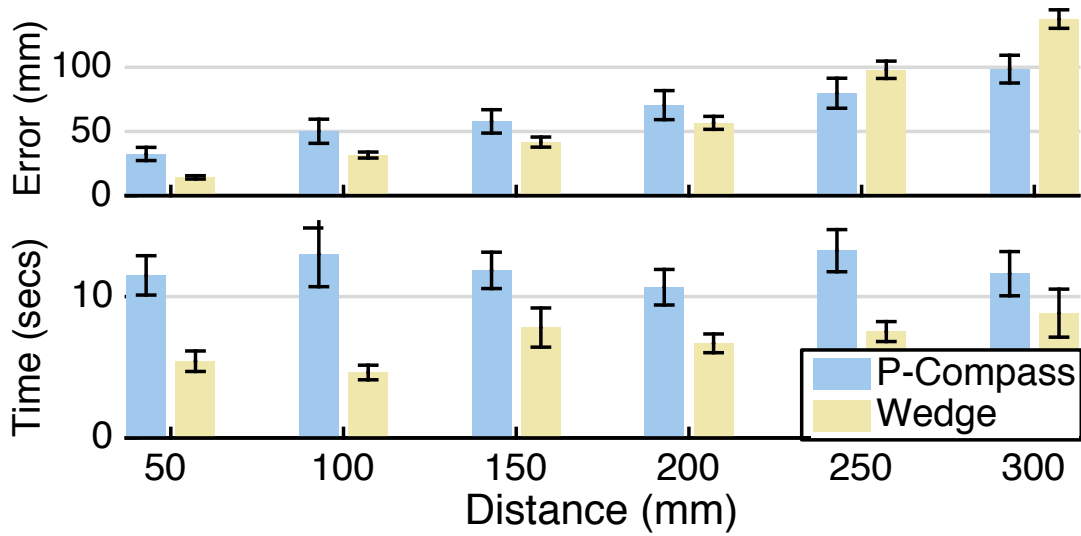
	<b>P-Compass</b>	<b>Wedge</b>
LOCATE	11.98 sec (8.00)	6.82 (5.43)
DISTANCE	15.30 sec (11.89)	12.06 (5.34)
ORIENTATION	4.12 sec (3.10)	8.16 (6.58)
LOCALIZE	11.31 sec (7.56)	11.67 (7.35)
LOCATE+	9.03 sec (5.97)	9.94 (5.01)

**Table 4.2: Summary of completion times (s.d.).**

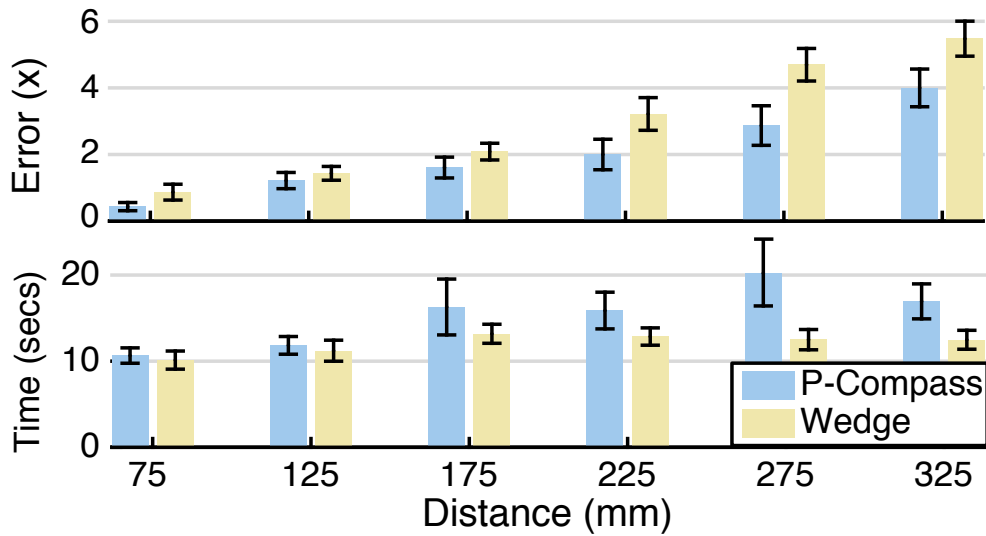
conducted with  $\alpha = 0.05$ . Since each task was carried out independently, we discuss them separately.

#### 4.4.1 LOCATE

Error for LOCATE (Figure 4.10 top) was defined as Euclidean distance between the estimated answer and ground truth. A  $2 \times 6$  (Visualization  $\times$  Distance Level) ANOVA on error indicates no significant main effect on visualization ( $F_{1,25} = 0.06, p = 0.81$ ). However, there is significant interaction between visualization and distance levels ( $F_{5,125} = 7.67, p < 0.001$ ). With post-hoc  $t$ -tests using Bonferroni correction, we found significant differences at 5 cm and 30 cm (both  $p < 0.003$ ), supporting **H4.1**. A similar  $2 \times 6$  (Visualization  $\times$  Distance Level) ANOVA was performed on completion time (Figure 4.10 bottom), showing a significant main effect on visualization ( $F_{1,25} = 37.22, p < 0.001$ ), but no interaction between visualization and distance levels ( $F_{5,125} = 1.29, p = 0.27$ ).



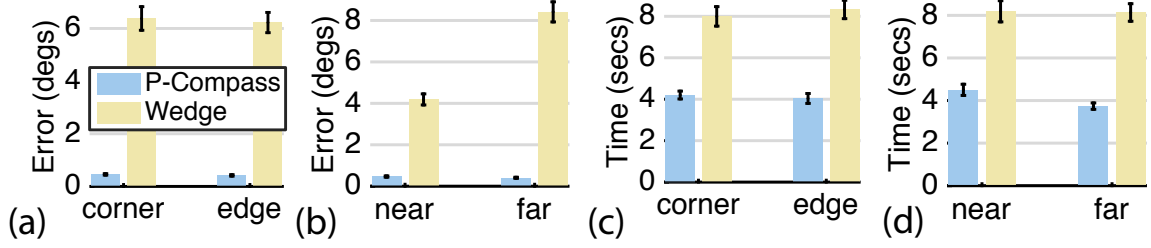
**Figure 4.10: LOCATE Results. (top) Error size. (bottom) Completion time.**



**Figure 4.11: DISTANCE Results. (top) Error size. (bottom) Completion time.**

#### 4.4.2 DISTANCE

At the farthest two distance levels, we found three participants consistently made large estimation errors using P-Compass (greater than three times the quartile), so we excluded their data from analysis. Interviewing one of these three participants, we learned that they had used the gap between the reference point and the FOV box as the basis for estimation, instead of the FOV box as expected. The FOV box almost overlapped the reference point



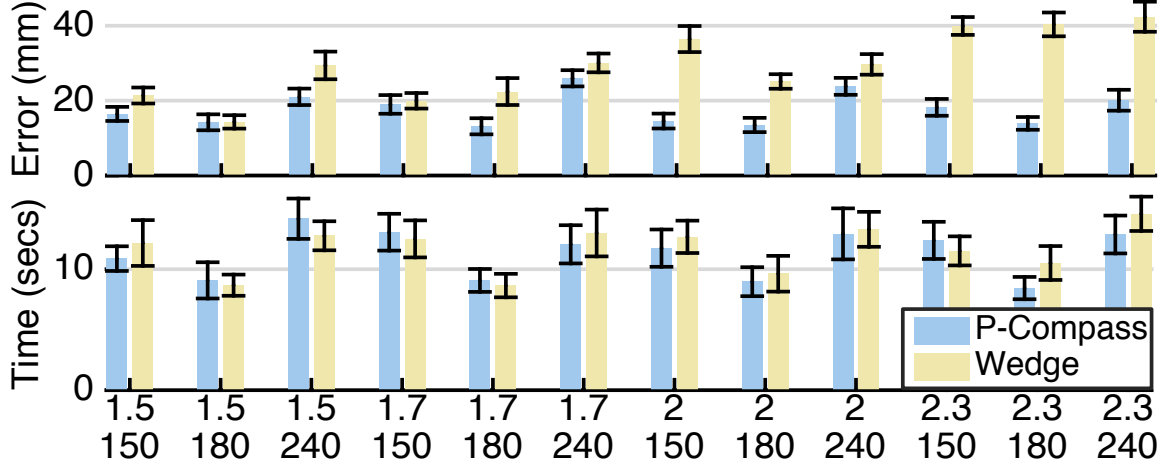
**Figure 4.12: ORIENTATION Results. (a–b) Error size. (c–d) Completion time.**

at the farthest two levels. There was no gap as the base of estimation. The participant acknowledged that their answers were random guesses.

Figure 4.11 (top) shows the error size for DISTANCE. Error is defined as the absolute difference between the estimated distance (in integer multiples of half-screen width) and the ground truth. A  $2 \times 6$  (Visualization  $\times$  Distance Level) ANOVA on error shows a significant main effect on visualization ( $F_{1,22} = 7.25, p = 0.013$ ). This result supports **H4.2**. No interaction between visualization and distance level was found ( $F_{5,110} = 1.89, p = 0.10$ ). A similar  $2 \times 6$  (Visualization  $\times$  Distance Level) ANOVA was performed on completion time. There is a significant main effect on visualization ( $F_{1,25} = 11.44, p = 0.003$ ), but no interaction between visualization and distance levels ( $F_{5,110} = 1.34, p = 0.25$ ).

### 4.4.3 ORIENTATION

Error size for ORIENTATION (Figure 4.12a–b) was defined as angular distance between the estimated answer and the ground truth. A  $2 \times 2 \times 2$  (Visualization  $\times$  Distance  $\times$  Position) ANOVA on error indicates a significant main effect on visualization ( $F_{1,25} = 89.06, p < 0.001$ ), no significant interaction between visualization and position ( $F_{1,725} = 0.16, p = 0.68$ ), but significant interaction between visualization and distance ( $F_{1,725} = 78.85, p < 0.001$ ). An ANOVA was performed on completion time (Figure 4.12c–d), showing a sig-



**Figure 4.13: LOCALIZE Results.** (top) Error size. (bottom) Completion time. The  $x$ -axis indicates the parameters of the exploratory conditions (see text for details).

nificant main effect on visualization ( $F_{1,25} = 69.18, p < 0.001$ ), no significant interaction between visualization and position ( $F_{1,725} = 0.07, p = 0.79$ ), and no significant interaction between visualization and distance ( $F_{1,725} = 1.21, p = 0.27$ ).

#### 4.4.4 LOCALIZE

Figure 4.13 (top) shows error size for LOCALIZE. The  $x$ -axis of the figure indicates the parameters of the exploratory conditions. The top row indicates the distance ratio of two off-screen POIs (base length 7.79 cm), and the bottom row indicates the angular difference in degrees between the two off-screen POIs, all with respect to the display centroid. Error is defined as Euclidean distance between the estimated answer and the ground truth. A one-way (Visualization) ANOVA on errors indicates a significant main effect on visualization ( $F_{1,25} = 69.97, p < 0.001$ ), supporting **H4.3**. A similar one-way (Visualization) ANOVA was performed on completion time, and found no significant main effect on visualization ( $F_{1,25} = 0.097, p = 0.75$ ).



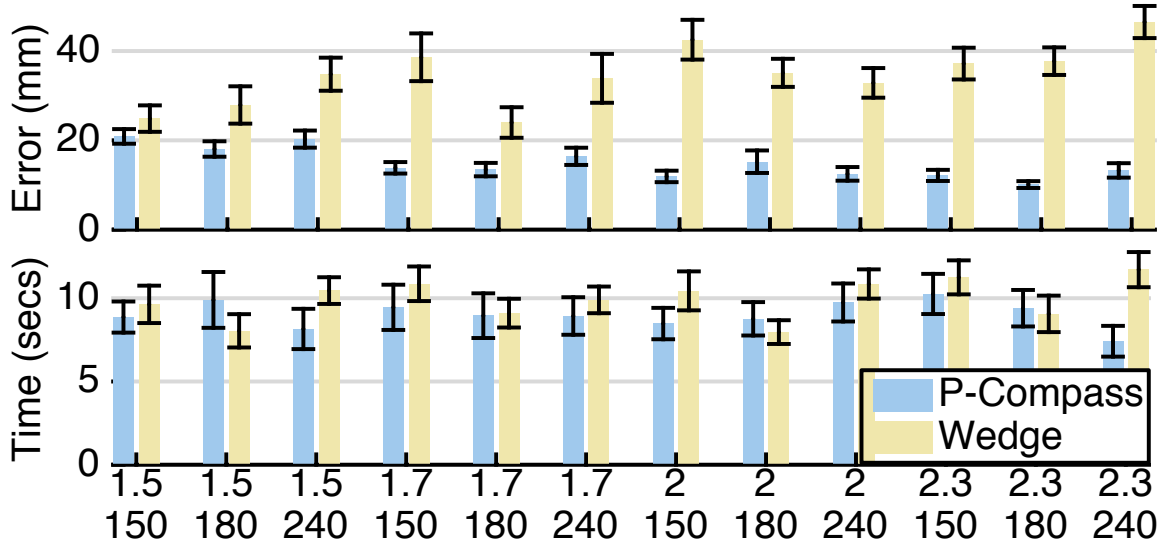


Figure 4.14: LOCATE+ Results. (top) Error size. (bottom) Completion time. The  $x$ -axis indicates the parameters of the exploratory conditions (see text for details).

#### 4.4.5 LOCATE+

Figure 4.14 (top) shows error size for LOCATE+. The  $x$ -axis of the figure indicates the parameters of the exploratory conditions (the same as in the LOCALIZE task). Error is also defined as in LOCALIZE. We performed a one-way (Visualization) ANOVA on errors and found a significant main effect on visualization ( $F_{1,25} = 49.53, p < 0.001$ ), supporting **H4.4**. A similar one-way (Visualization) ANOVA was performed on completion time, and found no significant main effect on visualization ( $F_{1,25} = 1.87, p = 0.18$ ).

#### 4.4.6 Subjective Preferences and User Comments

At the end of the session, the participant was asked to choose their preferred technique for each task and provide comments. Table 4.3 summarizes participants' preferences. Chi-Square tests reveal P-Compass was preferred for ORIENTATION, LOCALIZE, and LOCATE+ (all  $p < 0.005$ ). No significant preferences are found for LOCATE ( $p = 0.67$ ) and

	<b>P-Compass</b>	<b>Wedge</b>	<b>No Preference</b>
LOCATE	10	12	4
DISTANCE	16	7	3
ORIENTATION	25	0	1
LOCALIZE	23	3	0
LOCATE+	19	5	2

**Table 4.3: Participants’ preferences on each task.**

DISTANCE ( $p = 0.06$ ). These results partially support **H4.5**.

It is interesting to note the preference difference between LOCATE and DISTANCE, because the two share a similar setup but ask different questions (Figure 4.7). The preference reversal suggests that characteristics of a technique could make certain types of tasks more or less difficult, as also pointed out by participant 23 (P23) in a comment on Wedge, “I felt like I had trouble estimating distances well, but the tasks involving locations felt easier.”

Regardless of the technique, all participants expressed frustration with LOCATE and DISTANCE, especially for distant off-screen POIs. P27 commented on Wedge, “It’s hard to tell small differences in the angle at the base of the triangle, especially when it is very close to perpendicular.” P14 said about P-Compass, “I found it very hard to determine the distance when the mini screen [FOV box] is very small.”

For ORIENTATION, the majority of the participants preferred P-Compass and commented on the difficulty of using Wedge. P18 remarked, “it needed physical measurement help from my fingers to get the location of the tip of triangle.”

Most participants preferred P-Compass for LOCALIZE and LOCATE+. However, we were puzzled by a small number who preferred Wedge, even though quantitative data suggests P-Compass allowed them to achieve higher accuracy in comparable time. We later learned from the comments that those participants were more comfortable when all off-

screen POIs were presented at the same scale as the displayed content, and they felt Wedge performed better. Indeed, in this study we limited all POIs to a distance range where Wedge can be functional. It is possible within this setup that individual preferences overshadow the strengths of P-Compass.

## 4.5 Discussion

The study results offer strong evidence to support **H4.1–H4.4**. **H4.5** was partially supported by significant preferences for three out of the four predicted tasks.

In terms of scalability, the errors of LOCATE and DISTANCE support our analysis that the display extension method does not scale. This is also reflected in participants’ frustration, as revealed in the comments. Putting the results in real-life context, one can see that it is only meaningful to perform LOCATE and DISTANCE when a POI is at the display boundary. In LOCATE and DISTANCE, the farthest off-screen POI is 30 cm from the screen boundary, which corresponds to half the height of New York’s Central Park at the scale of 1 cm per city block—a reasonable distance for a tourist or a city resident, yet the average error at this distance amounted to 10 cm (10 city blocks). This underscores the importance of having the POI-reference method.

In terms of task type, the results support our analysis that the characteristics of the different visualization technique make them suitable for different tasks. While participants could use Wedge to achieve better accuracy for LOCATE (especially when off-screen distance was less than 20 cm), the same performance advantage was not observed for DISTANCE. Similarly, Wedge’s advantage for locating an off-screen POI becomes a disadvan-

tage for ORIENTATION, LOCALIZE, and LOCATE+. As reported in previous research [Ion et al. 2013] and our participants’ comments, participants in the Wedge condition often used fingers to denote an imaginary POI, or shift the smartphone/smartwatch away to have a better overview. The need to use fingers could make Wedge difficult for hands-busy, mobile, or dynamic scenarios.

For ORIENTATION, the results show it is more challenging for a user to estimate the direction of a distant POI than a nearby one using Wedge, but surprisingly the position condition (edge/corner, Figure 4.12a) does not yield a significant result. The reason could be due to our implementation of Wedge. We provide ample intrusion at edges and corners; the minimal intrusion at corners is the same as the maximal intrusion at edges. For LOCALIZE and LOCATE+, the exploratory conditions offer interesting patterns, suggesting that future work could investigate how the distribution of POIs may affect estimation accuracy.

Regarding completion time, on average we found using P-Compass took roughly twice as much time as Wedge for LOCATE. This could be due to the more complicated mental process required by P-Compass, as described earlier. It also took longer on average to complete DISTANCE using P-Compass. It is possible that Wedge users gave up when a POI was beyond some distance (which could explain the relatively flat completion time shown in Figure 4.11 bottom), while P-Compass users held on longer. Note that we asked users to provide a rough estimate if they were unable to provide a good answer. For ORIENTATION, LOCALIZE, and LOCATE+, P-Compass completion time was either shorter than or similar to that of Wedge.

### 4.5.1 Design Recommendations

Based on the analysis and study results, we make the following design recommendations:

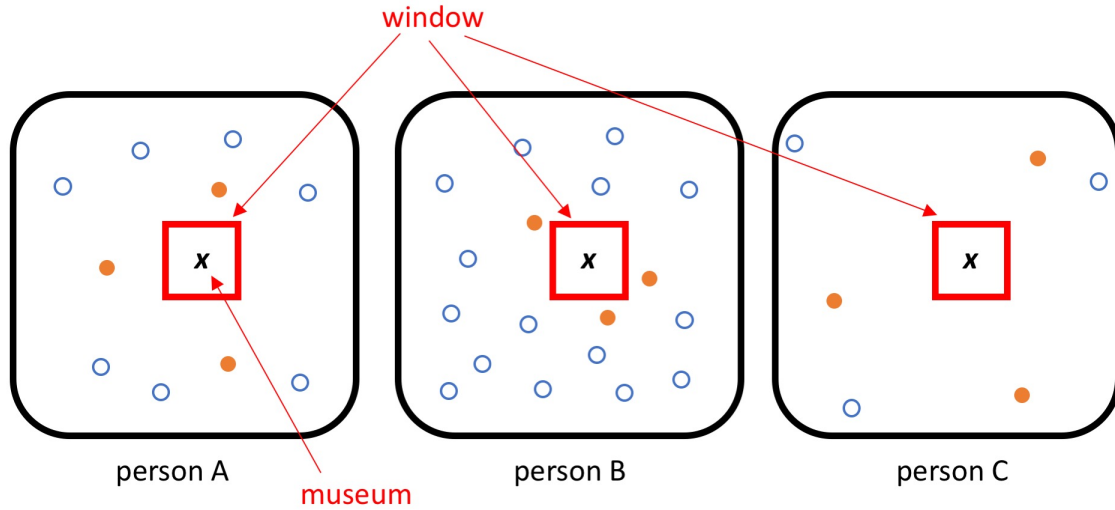
**Replace a compass with a P-Compass.** The results show P-Compass helps users infer the directions and locations of off-screen POIs, and resolves Desert Fog. The added benefits require little modification to a conventional compass.

**Use P-Compass for distant off-screen POIs.** P-Compass yields more accurate results for distant POIs, and was preferred by most study participants. Distant POIs let a single P-Compass accommodate all reference points on a display.

**Use Wedge for nearby off-screen POIs.** Wedge essentially extends the effective area of a display, and has the benefit of presenting POIs at the same scale as the displayed content. While our formal study focused on more distant POIs, in our field experience and pilot study we observed that Wedge appeared to offer similar or better performance, especially for POIs close to the display edge. Previous studies [Gustafson 2008; Burigat and Chittaro 2011] have also reported the effectiveness of Wedge for nearby POIs.

**Take display size into account.** Wedge's effective zone gets smaller as display area decreases. Designers should take display size into account to determine the distance at which to switch between Wedge and P-Compass.

**Give users control.** In practice, a gray area, rather than a clear line, may exist between effective zones for Wedge and P-Compass. In addition, depending on the task, a specific user may prefer one technique over another. Application designers should provide the option to switch between visualizations.



**Figure 4.15: The idea of personalization.**

## 4.5.2 Limitations

P-Compass requires personal *a priori* knowledge of a space. The more detailed the knowledge a user has, the better the estimate a P-Compass can provide. Figure 4.15 illustrates the concept. Without loss of generality, the three round-cornered squares represent three identical information spaces (e.g., maps). The dots (both solid and hollow) within each square indicate the POIs known by three respective users: *A*, *B*, and *C*. In this case, users *A*, *B*, and *C* all have different levels of familiarity with the information space. Suppose each of the three users observes this information space through a window (the solid red rectangle in each round-cornered square), and discovers a museum (marked as *x* in the figure). Where is the museum in this information space?

P-Compass uses *a priori* locations as “supports” to explain a unknown location to a user. In this case, the solid orange dots in each rounded rectangle represents some potential *a priori* locations that can be used to form a P-Compass to help each user interpolate the unknown location. Thus, different users may receive different supports. User *B* has

more detailed knowledge about the information space, in comparison with user  $C$ 's sparse knowledge about the information space. Therefore,  $B$  can receive more detailed support. Of course, as a user gains more knowledge about an information space, that user's *a priori* location database can be updated, and better support can be provided.

The source of personalization in our prototype came from the use of a location database manually input by a user. There are several other way to generate a personalized location database. For example, at the very first initialization of a system, the system could be designed to infer a user's geospatial knowledge by asking the users a series of questions. This technique is used by several commercial systems such as Netflix or Apple Music to provide personalized services to users. Other methods to collect *a priori* POIs could include drawing them from personal GPS [Zhou et al. 2007] and cellular network [Isaacman et al. 2011] location history, or from social network traces (e.g., Facebook check-ins and Google Maps Saved Places), and inferring them from public sources (e.g., Flickr) by data mining [Snaveley et al. 2008; Snaveley et al. 2006]. These methods could be combined and continuously refined using machine learning.

Overlapping POIs can reduce the readability of a P-Compass. However, the primary aim of a P-Compass is to use *minimal* information to answer the “Where is  $x$ ?” question. For a small number of overlapping POIs, varying the color or thickness of compass needles could be a potential solution.

## 4.6 Applications: Street View and 3D Map

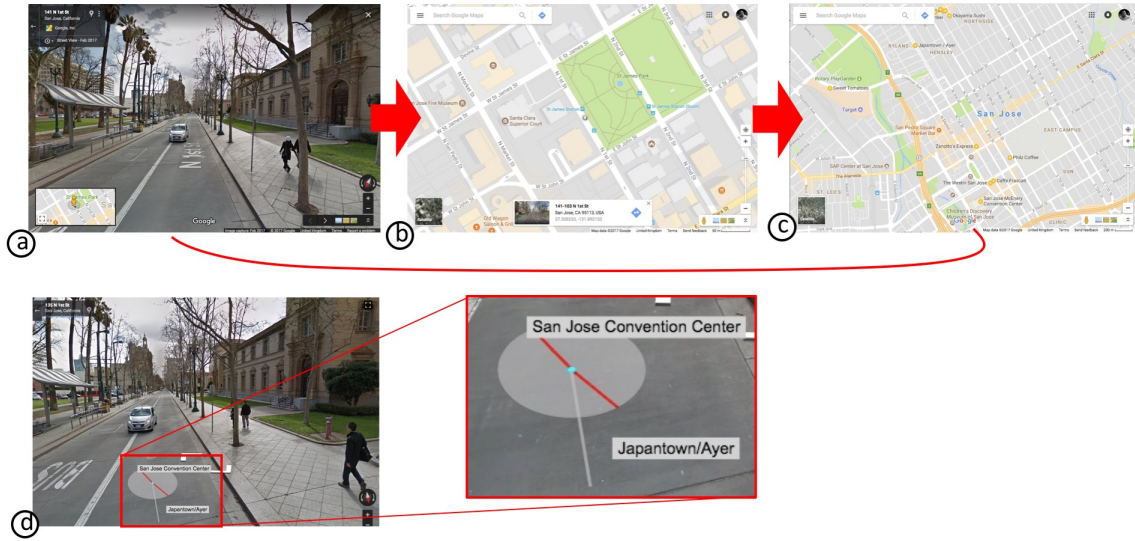
In addition to map user interfaces, we have prototyped P-Compass in user interfaces for two other types of geospatial information: (Google) Street View and (Apple) 3D map. In this section, we will briefly motivate the issues for each of these user interfaces. As we will see, the “Where is x?” problem also exists in these user interfaces and we will demonstrate how it can be tackled by integrating P-Compass.

### 4.6.1 Street View

Figure 4.16(a) shows a screenshot of Google Street View. The Street View interface indicates the address of the scene is “141 N 1st St, San Jose, California.” The minimap in the bottom-left corner of the interface indicates this location is at a corner of “St. James Park.” The compass on the right of the interface indicates the viewer is roughly facing south. For a viewer who has limited or no knowledge of the area, the above information does not help establish a frame of reference; that is, it does not place this location with respect to locations that the viewer may know (or alternatively, place some locations the viewer may know with respect to this location). To figure out “where this place is,” the viewer needs to click the minimap to switch to a larger map (Figure 4.16b), and zoom out the larger map further until recognizable features come into view (Figure 4.16c). The viewer then needs to switch back to Street View when she is done.

This example is inspired by the author’s real experience of preparing for a conference trip to San Jose in 2016. At the time, the author wished to use Street View to see what the area looked like, but was frustrated about not being able to connect the Street View scene





**Figure 4.16: (a–c) Interaction with Google Street View. (d) A P-Compass superimposed on a prototype developed with the Google Street View SDK.**

to the author’s knowledge of the environment, however limited. Figure 4.16(d) shows our Street View prototype with a P-Compass added to a similar scene. The locations were selected from a small location database manually entered by the author. In this figure, the P-Compass has three needles: the gray needle points north, and the other two needles point to Japanese Town and the San Jose Convention Center.

We developed the prototype shown in Figure 4.16(d) for research purposes <sup>4</sup>. This prototype was created by superimposing a JavaScript drawing layer on a scene rendered with the Google Street View SDK. As a user changes viewpoint (by pitching, rotating the view, or walking virtually), parameters (longitude, latitude, heading, pitch) associated with the visible scene are updated, and the P-Compass is updated accordingly. The P-Compass

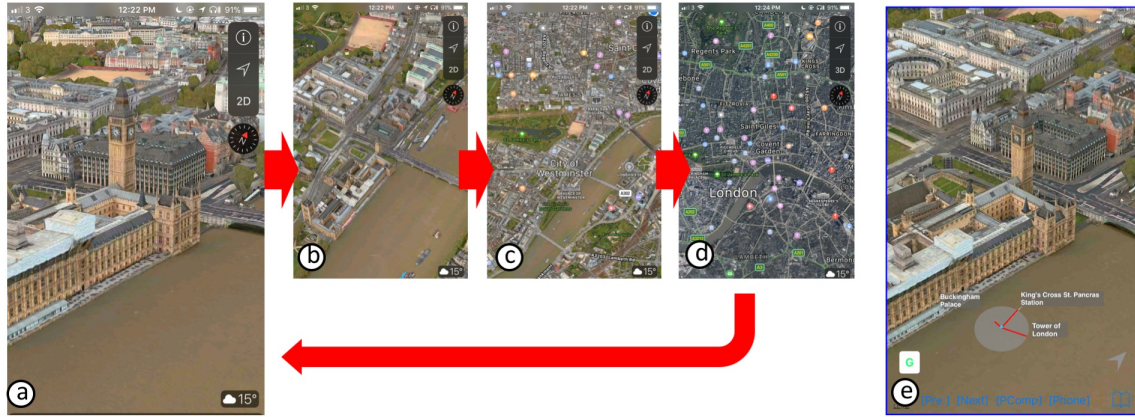
<sup>4</sup> This prototype misses several features in the original Street View application, since it was not practical to duplicate them to a research prototype. For example, one would find that the address indication message box and mini maps are missing in Figure 4.16 (d). Label placement for the compass needles could also be improved by using a labeling algorithm (e.g., [Bell et al. 2001]), but this is beyond the scope of our prototype. Labels and their placement were not implemented in other related research [Gustafson et al. 2008; Baudisch and Rosenholtz 2003; Burigat et al. 2012; Burigat and Chittaro 2011]. We made an effort implementing those features to make the prototype usable in real setting with real data.

is formed by the location-selection algorithm described in Appendix A.1, and the location database is a list of locations manually input by the author. In our implementation, the P-Compass hovers over the ground, and the face of the P-Compass is parallel to the ground.

### 4.6.2 3D Map

We also prototyped P-Compass with the Apple 3D map application [Apple 2018]. Figures 4.17 (a) shows a screenshot of the original 3D map application (running on an iPhone 6s) displaying a scene in London. A viewer who is not familiar with London might recognize the building (UK Parliament) by its appearance, but may not know where it is located. The “Where is  $x$ ?” problem exists in this situation, too. Figure 4.17 (b–d) illustrates what the viewer may do to determine where the Parliament is with respect to other locations they may know. Should the viewer then wish to return to exploring the area around the Parliament, they will need to adjust the view back. Figure 4.17 (e) shows our prototype with a P-Compass superimposed on the 3D map application. In the figure, the three needles of the P-Compass illustrate the relationship between the Parliament and three other tourist sites: Buckingham Palace, King’s Cross, and the Tower of London. A north needle (the gray needle with its length extending beyond the compass disk) overlaps “King’s Cross.”

Our prototype is based on the Apple MapKit SDK, and runs on iOS (for iPhone and iPad) and OS X (desktop) devices. An OpenGL layer for drawing the P-Compass is superimposed on the 3D scene produced by the MapKit SDK. The reference point (blue dot) of the P-Compass corresponds to the coordinates (latitude and longitude) of the underlying scene point. Similar to the Street View prototype, the face of the P-Compass is parallel to




**Figure 4.17: (a–d) Interaction with the Apple 3D earth tour application. (e) A P-Compass prototype on the 3D earth tour application.**

the ground, and the P-Compass is dynamically updated as the user explores the scene. The user can also scale and reposition the P-Compass. The application is bootstrapped with a database of manually input locations, and the P-Compass is formed and updated by the POI selection algorithm described in Appendix A.1. Alternatively, a user can manually select and fix POIs.

We demonstrated both the Street View and 3D map prototypes, along with the P-Compass 2D map prototype described earlier at several venues, including an international HCI conference, *CHI 2016* [Miau and Feiner 2016], and several local events such as the *2016 NYC Media Lab Summit* and *TWIG 2016 (Tristate Workshop on Imaging and Graphics)*. Many demo attendees echoed their experiences of encountering the “Where is x?” problem when using map-related applications, and appreciated the core idea underlying P-Compass of “using what a user knows to figure out what the user doesn’t know.” The *CHI 2016* demo attendees in particular shared their experiences of how they used places they visited recently (e.g., a coffee shop, a sandwich shop, or the convention center) in referencing new locations, since many of them were temporary visitors to the conference

venue in San Jose, CA. Many demo participants also mentioned that they would like to see P-Compass integrated into their map applications, and offered feedback or feature suggestions such as allowing users to manually input landmarks and manually control the appearance of landmarks, making use of other spatial entities, and adding iconic representations. One straightforward extension of the Google Street View and 3D map prototypes would be smartphone AR/VR. However, there are some issues regarding applying P-Compass to a 3D environment that we will discuss later when addressing future work in Section 6.2.2.

## 4.7 Conclusions

We introduced P-Compass, a compact graphical location indicator to address the “Where is  $x$ ?” problem, which generalizes the problems of off-screen POIs and Desert Fog. P-Compass uses personal *a priori* POIs to establish a reference frame in which the POI  $x$  can then be localized. A P-Compass can be integrated into a map or accompany visual content. As an extreme example, the standalone P-Compass in Figure 4.1(i) can even be directly embedded into text, much like an inline equation: [Seattle; SF;  Yellowstone; Chicago].

We examined the characteristics of P-Compass and Wedge, and conducted a formal user study to compare the two. The results showed P-Compass performs better for the four inference tasks, while Wedge is better for the locating task. Based on the analysis and study results, we suggested the two techniques are complementary and offered design recommendations. As a rule of thumb, as the display size becomes smaller, and/or the distance of a POI increases, the benefits of P-Compass eventually outweigh those of Wedge.

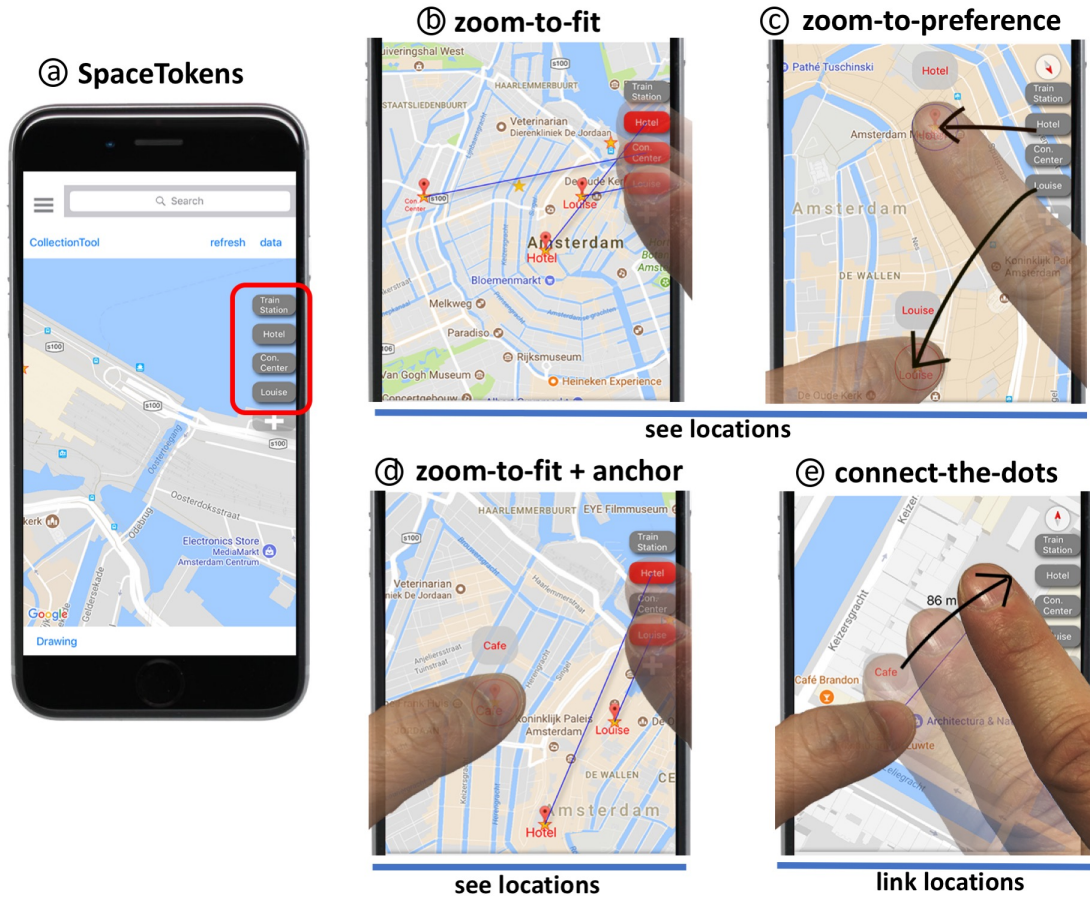
## Chapter 5

---

### *SpaceTokens and SpaceBar*

In the previous chapter, we discussed the “Where is  $x$ ?” problem, and introduced Personalized Compass, a navigation instrument that leverages personal *a priori* POIs to help users address the problem. In this chapter, we will present two additional problems frequently faced by map users: (1) “How to see and link important locations rapidly?” and (2) “How to interact with or comprehend a route (a sequence of locations associated temporally) efficiently?” Much like the way we addressed the “Where is  $x$ ?” problem through the creation of Personalized Compass, we tackle these two problems by designing navigation instruments that strategically leverage personal important locations: SpaceTokens and SpaceBar, respectively.

SpaceTokens make locations first-class citizens of map interaction, and empower users to see and link locations directly using locations. SpaceBar associates a simple linear scrollbar to a complex nonlinear route. That is, a SpaceBar essentially represents a simplified route with which a user can interact to comprehend and control the (original) route efficiently. We will conclude by presenting a user study that compares our SpaceTokens/SpaceBar prototype with Google Maps.



**Figure 5.1: SpaceTokens make locations first-class citizens of map interaction. (a) Four Space-Tokens (gray rectangles with rounded corners) each represent a location. SpaceTokens empower users to use locations directly to address “seeing” and “linking” problems discussed in this chapter. For example, users can (b) tap SpaceTokens to make corresponding locations visible; (c) drag SpaceTokens onto map, so corresponding locations snap to positions of dragged SpaceTokens; (d) hold an on-screen location (using left thumb) and tap several SpaceTokens to see their relative positions; (e) link an on-screen location to a SpaceToken using LinkTool (bubble above left thumb). LinkTools in (c–e) also serve as visible labels for locations being touched.**



## 5.1 Introduction

Think about taking a trip to an unfamiliar locale. Before or during your trip, you may research the area around your hotel, a transportation hub, and perhaps some museums and other sites; view several of them simultaneously to support decision making (e.g., is the museum on the way from  $A$  to  $B$ ?); or use them as your “landmarks” to localize an unknown location in your mental image of this unfamiliar area [Lynch 1960]. Furthermore, you may create routes involving these locations to gather transportation or direction information. In addition to these static locations, you may also interact with dynamic ones, such as your changing location and those of your friends. Ironically, these location-centric tasks cannot be performed directly with locations; instead, a user must pan and zoom to find these locations on a map, or access them from menus or search dialogs. In this chapter we ask, “Why can’t a user perform location-centric tasks *directly* using locations?”

To answer this question, we propose *SpaceTokens*, interactive widgets that are clones of locations, and that a user can create and dock at the edge of a display (Figure 5.1a), much like magnets on a whiteboard. SpaceTokens make location a first-class citizen of map interaction.

Let’s see how SpaceTokens could help Louise, a graduate student, during a conference trip to Amsterdam. In our example, each interaction technique/tool is set in **bold** in its first occurrence, and each SpaceToken is denoted with square brackets as *[location]*, where *location* is the location corresponding to the SpaceToken. Louise has docked four SpaceTokens ([train station], [hotel], [conference center], and [Louise]) at the edge of her smartphone, as shown in Figure 5.1(a). Louise would like to see where she is with respect

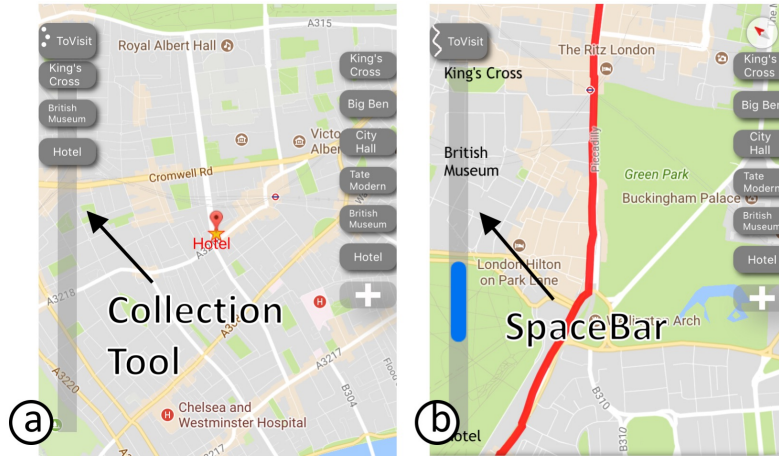
to the hotel and the conference. She taps [Louise], [hotel], and [conference center], and the map is successively adjusted to display them all (Figure 5.1b). We refer to this interaction as **zoom-to-fit**.

Louise decides to head to her hotel first, so she drags [Louise] to the bottom of the display, and then [hotel] to the top, making it easy to look for potentially interesting stops on the way to her hotel (Figure 5.1c). In response, the map window is dynamically adjusted such that the locations of Louise and her hotel on the map are snapped to the positions of [Louise] and [hotel] on the display. We refer to this interaction as **zoom-to-preference**.

Disappointed that there’s nothing of interest on the way, Louise recalls a cafe recommended by a friend. Louise types the name of the cafe in the search dialog and sees it appear in the middle of the display. Unsure where the cafe is, Louise uses one finger to hold the cafe in place (**anchor**) while tapping [Louise] and [hotel] with another finger to see all three on the display (Figure 5.1d). Louise then zooms into the cafe to take a closer look at the surrounding area. “Does it make sense to make a detour?” Louise asks herself. Louise taps the cafe to select it, and presses it again to enable the **LinkTool** (the bubble above the left thumb in Figure 5.1e). She drags the LinkTool to connect the cafe to [hotel] and to [Louise] to investigate the walking times from the cafe to the other locations (**connect-the-dots**, where Figure 5.1e shows only the action to connect to [hotel]). The walks seem short, so Louise decides to grab some coffee before heading back to her hotel.

The common thread running throughout these interactions is that location is treated as a first-class citizen; a combination of SpaceTokens and on-screen locations can be used to directly perform location-centric tasks. In addition to the LinkTool, we introduce **CollectionTool**, which organizes SpaceTokens (Figure 5.2a), and **SpaceBar**, which allows users





**Figure 5.2: Additional tools. (a) CollectionTool is a structure to organize locations. (b) SpaceBar is a scroll-bar-like tool for users to interact with a route. CollectionTool transforms to SpaceBar when creating a route connecting all locations in CollectionTool.**

to interact with a route efficiently (Figure 5.2b).

To explore the utility of SpaceTokens and their interaction techniques/tools, we developed a custom iOS map application that uses them. We then designed and ran a qualitative user study in which we asked participants to use our application to perform five common map tasks and compare their experience using Google Maps [Google 2017c] for the same five tasks, all running on identical smartphones. Participants found SpaceTokens intuitive and useful for frequent map tasks and overwhelmingly preferred the SpaceTokens prototype for the majority of tasks.

## 5.2 Motivation: Seeing and Linking Locations

Two problems motivated the development of SpaceTokens: *seeing* and *linking*, individually and in combination.

**Seeing.** *Given a user’s set of important locations, how can the user see one or more of them rapidly?*

There are many ways in which a user may wish to see her important locations. As described in the introduction to this chapter, Louise wishes to see the area around one or more locations, or the spatial relationships among them. With the increasing popularity of location sharing (e.g., Google Maps’ realtime location sharing feature [Google 2017d] or Apple’s Find Friends feature [Apple 2017b]), a user may also wish to see a combination of static locations and dynamic ones (e.g., people or vehicles) to coordinate planning. For example, Louise may want to check her colleagues’ locations, her location, and her hotel’s location, to decide who should pick up a package from the hotel.

Often, seeing (locations) is not the goal, but rather a means to achieve a goal. Users may wish to see the distribution of locations to perform clustering (e.g., to determine which three are closest to each other) or outlier detection (e.g., to determine which one is *not* close to the river). Users may then use the information to support decision making or to coordinate planning.

**Linking.** *Given a user’s set of important locations, how can the user link a location to one or more of them?*

Linking establishes a relationship between locations and is the precursor to additional tasks. For example, one may link point *A* to point *B* to seek travel directions, time, distance, or fare.

Commercial map applications typically provide “wizard” dialogues for linking locations, requiring a user to invoke the wizard and follow its flow, specifying the information needed to create a single route at a time. But suppose the user sees an interesting cafe on the display and would like to quickly compare the distances or estimated travel times from it to each of several important locations. The user must invoke a wizard and specify the

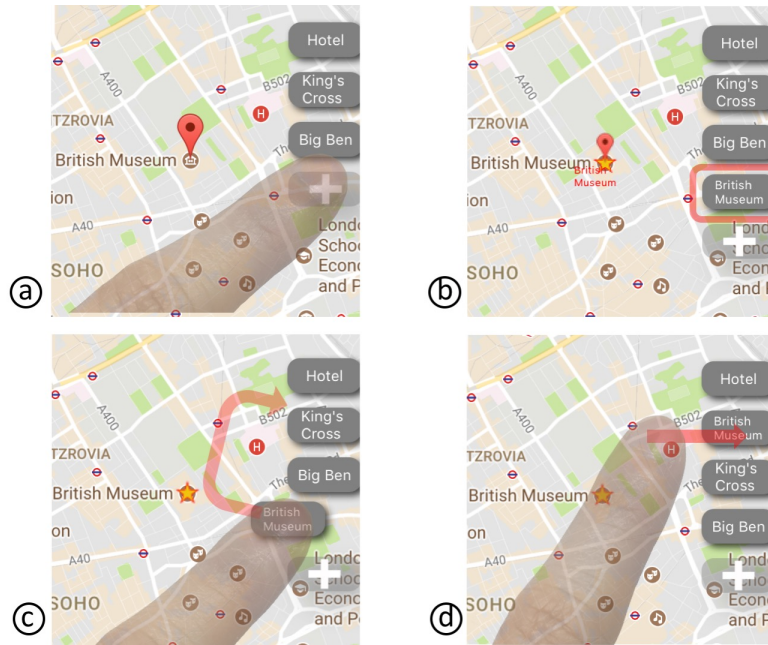
cafe and one of those locations to compute the distance or time between them, and then repeat this for each additional important location, recording the values, since they are not persistent.

Moreover, a user may wish to integrate seeing and linking, switching between them in rapid sequence. For example, Louise may wish to quickly create one or more routes to investigate transportation options among a set of locations and compare alternatives. As she starts to create a route, she may want to see how the locations on it relate to several other locations without having to exit the route-creation dialogue to specify the other locations. Block Party [Zhou et al. 2017] supports switching in and out of creating a route, without destroying it, through an interface with one mode for selecting locations and another mode for creating a persistent route. In contrast, we address not only this problem, but seeing and linking in general, through the use of SpaceTokens.

## 5.3 SpaceTokens

A SpaceToken is a clone of a location. A user creates a SpaceToken by first selecting a location on the map, and then tapping the “+” button (Figure 5.3a). A newly created SpaceToken is docked at the edge of the map (Figure 5.3b). Visually, SpaceTokens are like magnets on a whiteboard. Users can also arrange SpaceTokens in any order they prefer (Figure 5.3c). To remove a SpaceToken, a user can swipe the SpaceToken out of the display boundary (Figure 5.3d).

While we describe SpaceTokens in the context of a smartphone map app, different docking options could be used for other display form factors. For example, on a larger



**Figure 5.3: Creating SpaceToken to represent location. (a) After highlighting location, user taps “+” button to create SpaceToken for location. (b) New SpaceToken is added. (c) User can reorder SpaceTokens by long-pressing one and moving it to desired location. (d) SpaceToken can be removed by swiping it off display edge.**

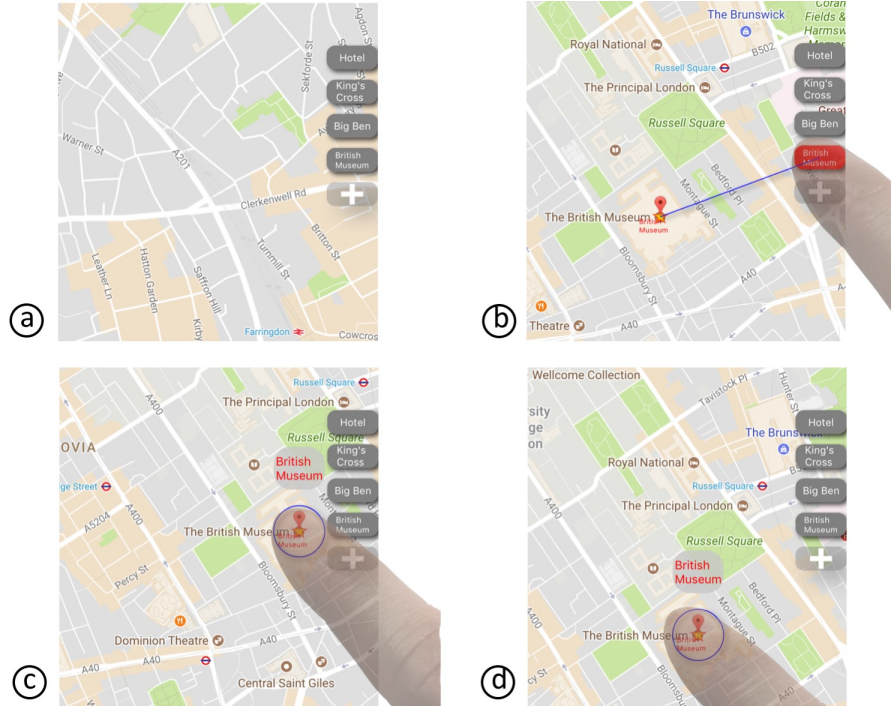
display, SpaceTokens can be organized in a palette, as we have done in an earlier iPad implementation (not shown here).

Now that a user can create SpaceTokens from map locations, what can be done with them?

### 5.3.1 Using Locations as Constraints Addresses the Seeing Problem

We saw how Louise uses a series of interaction techniques to see her important locations. At the heart of these techniques are two types of location constraints, the use of SpaceTokens to specify these constraints, and most important, the idea of allowing users to mix and match constraints to accomplish a variety of interaction techniques.

SpaceTokens support two types of location constraints: visibility constraints and posi-



**Figure 5.4: Using SpaceTokens to specify visibility and position constraints. (a) Where is British Museum? (b) Tap British Museum SpaceToken to see museum (specifying visibility constraint). SpaceToken turns red, screen zooms and pans as appropriate, and blue leader line indicates corresponding location. (c–d) Alternatively, drag British Museum SpaceToken to specify where to display museum (specifying position constraint). LinkTool appears above SpaceToken when touched. Multiple SpaceTokens can be used to mix and match visibility/position constraints of different locations.**

tion constraints. A *visibility constraint* ensures that a location (i.e., its map coordinates) must be visible somewhere on the screen. A *position constraint* ensures that a location must be at a specific set of screen coordinates. A user can use SpaceTokens to specify constraints for their corresponding locations. As the system receives a set of constraints specified by the user, a simple constraint engine adjusts the visible portion of the map to satisfy the constraints.

At its docking position, a SpaceToken serves as a toggle button that a user can tap to specify a visibility constraint for its location (Figure 5.4a–b). Multiple SpaceTokens can be toggled simultaneously to enable multiple visibility constraints. All SpaceTokens can

be reset (untoggled) by tapping an empty area of the map. A SpaceToken can be dragged to the display to specify a position constraint (Figure 5.4c–d). As a SpaceToken is dragged, it becomes a “location magnet.” That is, dragging a SpaceToken snaps its corresponding location to the screen coordinates of the SpaceToken.

Users can interact with multiple SpaceTokens to mix and match the constraints of multiple locations, and to achieve a variety of interaction techniques. If the user wants to view any subset of the SpaceTokens on the display, they need only toggle them on, and the visible portion of the map will be adjusted to fit them. We refer to this interaction technique as *zoom-to-fit* (Figure 5.1b).

While zoom-to-fit ensures certain locations are visible on the display, a user may also wish to specify *where* those locations should appear. In that case, the user can drag SpaceTokens to specify position constraints for the corresponding locations. Dragging one SpaceToken, the user essentially places a location on the display. Dragging two SpaceTokens causes the visible portion of the map to be adjusted to fit the two specified location position constraints (*zoom-to-preference*, Figure 5.1c). In theory (though quite awkwardly in practice), a user can specify three or more position constraints by dragging out that number of SpaceTokens. In this case, all constraints will in general not be able to be satisfied because of the fixed geospatial relationships of the locations. In this case, we fall back to zoom-to-fit and use leader lines to visualize how constraints are relaxed [Ullmer and Ishii 1997].

A user can also mix position constraints with visibility constraints. For example, a user can first drag out a SpaceToken and then toggle two other SpaceTokens, or vice versa. Whenever the constraints need to be relaxed, the system falls back to zoom-to-fit.

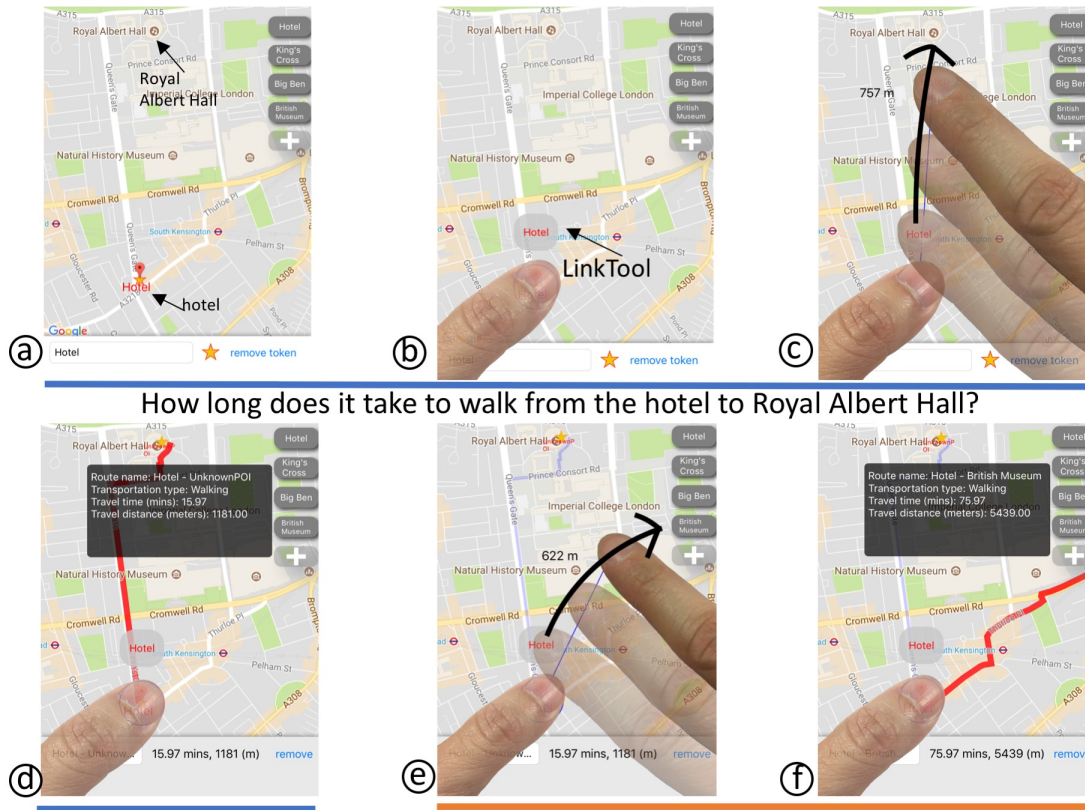
Suppose a user is interested in seeing an on-screen location  $x$  in relation to other locations represented by SpaceTokens. The user can use a finger to hold (which we will refer to as *anchor*)  $x$ , and use any subset of the other SpaceTokens to specify visibility or position constraints. By doing so, a position constraint will be created for  $x$ , and the visible portion of the map will be adjusted to fit both  $x$  and the locations corresponding to the selected SpaceTokens. This is essentially zoom-to-fit with anchor (Figure 5.1d), or zoom-to-preference with anchor. Thus, anchor offers an easy way to hold a location, much as a finger can be used to hold a location on a paper map, without having to create a SpaceToken for it.

### 5.3.2 Using Locations as Building Blocks Addresses the Linking

#### Problem

Locations can be used directly to address the linking problem through the LinkTool, which allows a user to establish a link between two locations. To access the LinkTool, a user first selects an on-screen location (Figure 5.5a, in which the hotel is selected), and then continues to touch the selected location (Figure 5.5b). The LinkTool, a bubble labeled with the location name, appears above the selected location (Figure 5.5b). Alternatively, the LinkTool is also available when the user drags a SpaceToken from its docking location (shown earlier in Figure 5.4c–d). While holding the location with one finger, the user can use another finger to drag out the LinkTool (Figure 5.5c). As it is dragged out, it acts like a tape measure that can be used to measure distance (Figure 5.5c, where the measured distance appears near the tip of the right index finger).





**Figure 5.5: Connect the dots using LinkTool.** (a) How long does it take to walk from my hotel to Royal Albert Hall? (b) Tap highlighted location to enable LinkTool. (c) Drag LinkTool to Royal Albert Hall, and press with force to confirm connection between both locations. (d) Route and information view appear. (e) How about walking to British Museum? Drag LinkTool again to British Museum SpaceToken. (f) Route and information view appear after LinkTool touches SpaceToken. (Tapping SpaceToken again would cause display to zoom out so British Museum is visible.)

A user can use LinkTool to *connect* to an on-screen location. As the user drags the LinkTool to a desired location, the user can increase their pressure (e.g., measured using 3D touch sensing on an iPhone) to confirm the establishment of a link<sup>1</sup>. In our implementation, a route between the two locations is established, and relevant information (e.g., walking distance and estimated travel time) is displayed in a transparent information view, as well as at the bottom of the display (Figure 5.5d). Once a route exists, the user can create a

<sup>1</sup>Additional aids for target selection, such as SHIFT [Vogel and Baudisch 2007], could also be implemented.

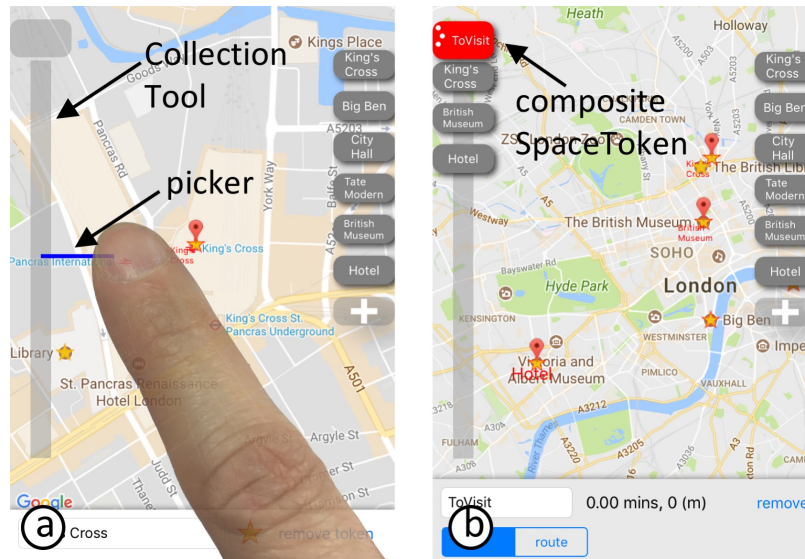


SpaceToken to represent the route (in the same fashion as creating a SpaceToken for a location, as shown in Figure 5.3a–b).

The LinkTool can also be used to connect to a SpaceToken (or the search box). Dragging the LinkTool to a SpaceToken establishes a relationship between the location corresponding to the LinkTool and the location corresponding to the SpaceToken (or a location input by the user, in the case of connecting to the search box). Similar to connecting to an on-screen location, a route between the two locations is established, and the relevant information of the route is shown. We refer to the interaction of linking locations directly as *connect-the-dots*. Figure 5.5(e–f) illustrates how the user creates an additional route by dragging the LinkTool to a SpaceToken. (As shown, previous routes remain displayed, though without highlighting; selecting any of them makes available a removal option at the bottom of the screen.) Thus, SpaceTokens support the ability to view and compare routes to different destinations simultaneously.

Rather than linking a pair of locations, a user may want to arrange locations in a sequence, and then connect them in that order. Alternatively, a user may want to arrange locations in a sequence without connecting them (for example, to create a ToVisit list). In both cases, users can add existing SpaceTokens or on-screen locations into the CollectionTool (Figure 5.6a), and order the locations based on their preferences. Users can interact with the SpaceTokens in the CollectionTool the same way they interact with any of the docked SpaceTokens.

Locations in the CollectionTool are represented as a composite SpaceToken, shown with an icon at the top of the CollectionTool (Figure 5.6b). Users can save the composite SpaceToken and load it back into the CollectionTool. Tapping the composite SpaceToken



**Figure 5.6: CollectionTool.** (a) Using one finger to drag out of CollectionTool enables picker (blue line), which allows users to pick any on-screen location or SpaceToken. (b) Composite SpaceToken on top of CollectionTool represents CollectionTool content (three locations here). Tapping composite SpaceToken shows all three locations on display.

adjusts the visible portion of the map to include all locations associated with the composite SpaceToken (Figure 5.6b).

When the composite SpaceToken is selected, a route function is made available that can be used to create a route connecting all the locations based on the order specified in the CollectionTool. Once a route is created, the CollectionTool is transformed into a SpaceBar, which we describe below.

## 5.4 SpaceBar: A Navigation Tool Built out of Locations

Building further on the idea of making location a first-class citizen of map interaction, we designed SpaceBar, a navigation tool built from the locations of a route—a continuous collection of locations.

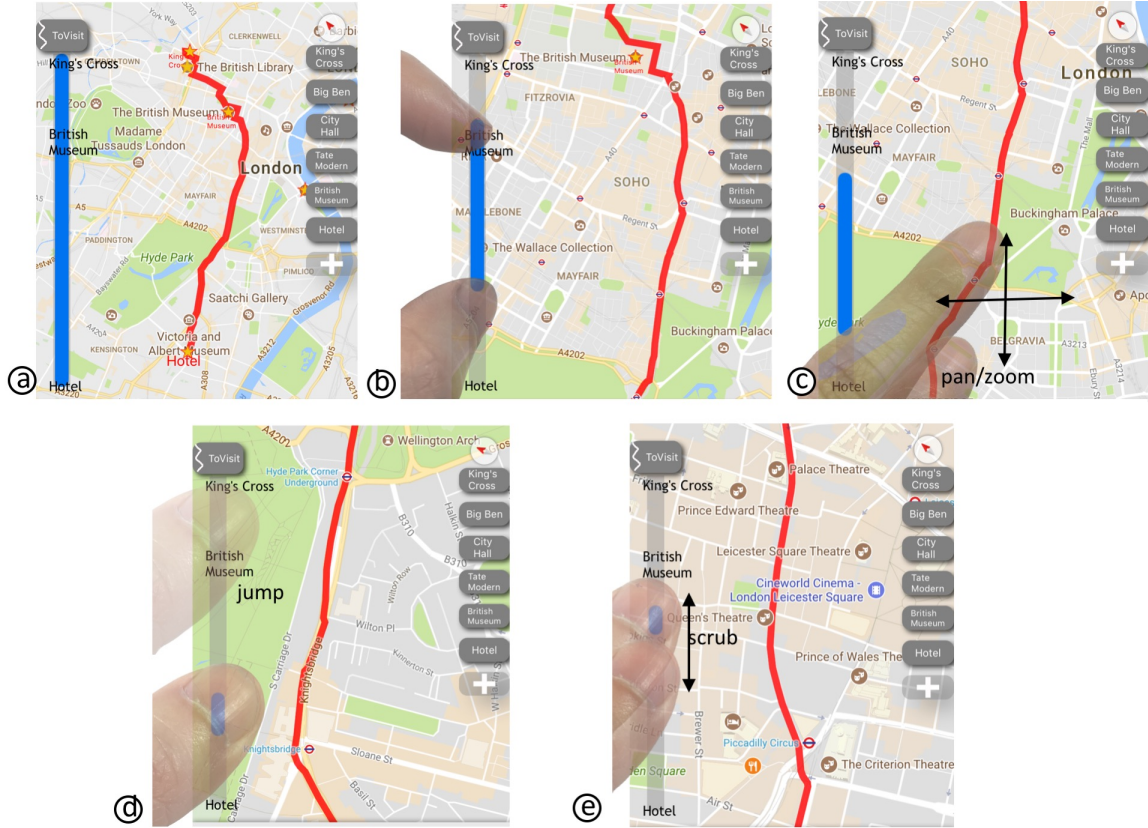
The SpaceBar, visually similar to a scrollbar, is an interactive widget to help users

interact with a route (Figure 5.7a). The entire SpaceBar represents a scaled, straightened route, with a one-to-one mapping between points on the SpaceBar and locations on the route. The SpaceBar has an “elevator” that indicates the visible portion of the current route. A user can control the two ends of the elevator to specify the portion of the route that is visible on the display (Figure 5.7b). This is similar to the zoom-to-fit interaction. The elevator changes its position and size as the user pans and zooms the map (Figure 5.7c).

A user can tap a point on the SpaceBar to jump to the corresponding portion of the route (Figure 5.7d). A user can also *scrub* the SpaceBar to traverse the route (Figure 5.7e). This interaction can be considered as an extension of content-aware scrolling [Ishak and Feiner 2006], which allows users to traverse a multi-column document based on the flow of its content, rather than its pages. Here, SpaceBar applies content-aware scrolling to a map, a continuous, multiscale information space. Annotations are overlaid on the SpaceBar to mark the locations that were added to the CollectionTool. Three locations (“King’s Cross”, “British Museum”, and “Hotel”, from top to bottom) are marked on the SpaceBar in Figure 5.7 as the starting point, a waypoint, and the destination of a route.

In our implementation, as the user scrubs the SpaceBar, the orientation of the map is adjusted such that the location corresponding to the top of the elevator is above the location corresponding to the bottom of the elevator. (The map orientation can optionally be fixed to be north-up. However, we found that this can be confusing, because in some situations the direction in which the elevator scrolls is not in sync with how the map scrolls in response.)

The shape and span of a route often make it challenging to interact with, and grasp information at different scales. To address this, many researchers have proposed techniques to distort a route [Wenig et al. 2015; Brosz et al. 2013; Agrawala and Stolte 2001; Sarkar



**Figure 5.7:** (a) SpaceBar is a scrollbar for a route. (b) User can change size and position of elevator indicator to make corresponding portion of route visible. (c) Conversely, elevator indicator is also updated as user interacts with route. (d) User can jump to any portion of route quickly by tapping SpaceBar, or (e) traverse route by scrubbing SpaceBar.

et al. 1993; Carpendale and Montagnese 2001], or to show a route at multiple scales simultaneously [Javed et al. 2012; Wang et al. 2014; Robbins et al. 2004]. We take a different approach by proposing a navigation instrument, the SpaceBar, to help users interact with an undistorted route in a single intact information space.

Another benefit of the SpaceBar is its relationship to the scrollbar in a document browser. Users' familiarity with scrollbars, and the large body of research on scrollbars (e.g., scrollbar annotations [Hill et al. 1992; Olsen 1992; Laakso et al. 2000] and scrollbar navigation [Igarashi and Hinckley 2000; Kim et al. 2014; Appert and Fekete 2006]) can all be directly transferred to SpaceBar.

## 5.5 Prototype Implementation

We implemented SpaceTokens and their interaction techniques and tools in a standalone custom iOS map application using the Google Maps SDK [Google 2017b]. Our application supports all basic map interactions such as pan, zoom, rotate, location search, and direction request, on both iPhone and iPad. While both platforms support the same features, we used an iPad to experiment with an alternative way of organizing SpaceTokens on a larger display—in a palette, rather than docked on the edge of the display.

The author used and refined the prototype (mostly the iPhone version, for reasons of portability) for around six months, including three international trips. The iPhone version was also used for the user study described below.

## 5.6 Evaluation

Our goal was to investigate whether SpaceTokens can help users address the seeing and linking problems, especially when they need to interact with more than one location frequently. We were interested in comparing SpaceTokens (referred to below as the *S* condition), embodying the idea of location-centric interactions, with Google Maps (referred to below as the *G* condition), representing the existing state-of-the-art map interactions supported by a commercial software application.

### 5.6.1 Study Design

Based on our personal field experience, we designed a trip scenario as the study backstory. Participants were asked to imagine being on a conference trip. A set of SpaceTokens (e.g.,

[hotel], [conference center], and [train station]) were preloaded to bootstrap the study for the *S* condition. A set of bookmarks for the same locations were also precreated for the *G* condition. Participants were instructed to create any number of additional locations of their choice. We created the five tasks described below to establish the study structure, so we could walk through five different interaction techniques and tools. After being introduced to each task, participants were asked to perform the task with SpaceTokens as well as Google Maps on identical iPhones. Throughout, participants were free to ask questions and express opinions to the study coordinator.

***SeeABC: Show A, B, C on the map.***

Given a set of locations (e.g., *A*, *B*, and *C*), participants were asked to solve several small problems, based on seeing some subset of the locations. These problems were similar to the ones described in Section 5.2: find an outlier (e.g., which location is *not* close to the river?), decide if *A* is on the way from *B* to *C*, and cluster locations (e.g., which two are close to each other?).

For the *S* condition, participants were asked to use zoom-to-fit/preference to perform the tasks. For the *G* condition, participants were asked to use Google Maps in ways they normally do (e.g., zooming to see locations, looking for locations from the search bar, or accessing locations from the “Your places” menu).

***SeeX+ABC: Show x with respect to A, B, C.***

This task is similar to the previous task; however, participants were asked to figure out where an unknown onscreen location *x* is with respect to some other important locations

( $A$ ,  $B$ ,  $C$ ). This occurs frequently in location search; for example, after a user searches for a cafe, she may want to know where it is with respect to her hotel. For the  $S$  condition, participants were asked to use zoom-to-fit/preference with anchor to perform the task. For the  $G$  condition, participants were asked to use Google Maps in the ways they normally do.

***Connect-the-dots: Establish links between locations.***

Participants were asked to perform quick investigations by creating several routes; for example, figuring out which candidate destination (among a set of SpaceTokens or bookmarks) is the closest from a cafe when all routes share a single source, or determining walking time between several pairs of locations when routes have multiple sources and destinations.

For the  $S$  condition, participants were asked to use the LinkTool. For the  $G$  condition, participants were asked to use the Google Maps directions dialog, which allows users to input locations and create a route for the input locations.

***CreateCollection: Create a collection of locations.***

Participants were asked to create a ToVisit list, using the CollectionTool for the  $S$  condition, and the “save to list” feature in the  $G$  condition. Participants were also shown how to create a route with the locations in a collection.

***ReviewRoute: Review and interact with a route.***

Participants were asked to study a route by interacting with it: jumping to different parts of the route, traversing it, or examining it at different scales. In the  $S$  condition, participants

used the SpaceBar. In the *G* condition, participants were asked to interact with the route in the ways they normally do.

## 5.6.2 Hypotheses

We formulated the following five hypotheses.

**H5.1, H5.2.** *For the “SeeABC” (H5.1) and “SeeX+ABC” (H5.2) tasks, users would prefer S over G.* We hypothesized that participants would appreciate being able to use locations (SpaceTokens) directly to specify which locations they would like to see, over manipulating the map in search of the locations or accessing the locations from menus/dialogs, as in the *G* condition.

**H5.3.** *For the “Connect-the-dots” task, users would prefer S over G.* LinkTool allows users to establish a relationship between two locations by directly connecting them, without the need to enter (and later exit) a directions dialog, as in the *G* condition.

**H5.4.** *For the “CreateCollection” task, users would prefer S over G.* While Google Maps also supports list creation, the layers of menus and context switches prevent users from seeing, adding, and removing locations in a tight loop. We thought users would appreciate CollectionTool, which facilitates making locations first-class interaction citizens.

**H5.5.** *For the “ReviewRoute” task, users would prefer S over G.* SpaceBar allows users to jump to any point or range on the route and scroll through the route. At the same time, SpaceBar also communicates an overview of the route while the user interacts with it. In contrast, in the *G* condition, users will need to manipulate the information space to examine different portions of the route; if the user wants to understand the context while



examining a portion of the route, they will need to zoom out.

### 5.6.3 Pilot Study

Six participants, who were our colleagues but had no knowledge of the project, were recruited for an initial pilot study, gave informed consent, and were paid a small cash amount. The pilot study was used to verify the tasks, the choice of hypotheses, and the study structure, and to identify bugs. Two of the pilot participants also used the prototype to perform free exploration and offered additional comments. For example, one commented that Google Maps' direction feature requires him to switch into and out of the direction mode, which often makes him suppress the urge to investigate interesting locations or perform quick measurements between locations.

Based on suggestions made by pilot-study participants, we decided to add to the formal study a sixth, "open-ended" task that would be performed only in the *S* condition:

#### ***PlanYourTrip: An open-ended task.***

After completing the five tasks, participants were asked to use their own scenario to explore SpaceTokens in an "open-ended" task: to prepare an upcoming trip or investigate the area around any city of their choice.

### 5.6.4 Formal Study

We recruited 15 additional participants (eight female, aged 19 – 27,  $\bar{x} = 23.1$ ,  $s = 2.9$ ) with no knowledge of the project from the general population of our institution, using mailing lists and posted flyers. In this single-session, within-subject lab study, informed consent

	SpaceTokens	GoogleMaps	No Preference
SeeABC	11	1	3
SeeX+ABC	15	0	0
Connect-the-dots	11	3	1
CreateCollection	13	0	2
ReviewRoute	8	6	1

**Table 5.1: Participants’ preferences for each task.**

was obtained and the participant was seated on an office chair and given two identical smartphones (Apple iPhone 6s), one running the SpaceTokens prototype and the other running Google Maps. The study lasted about one hour and was subdivided into three parts. During the first 15 minutes, the study coordinator introduced SpaceTokens using the prototype, as well as Google Maps. (All participants were familiar with Google Maps and mentioned using it in daily life.) The next 30 minutes formed the core of the study (the five tasks, with the *S* and *G* conditions counter-balanced, and the open-ended trip planning task), during which we encouraged participants to talk about their approach. For the last 15 minutes of the study, participants were asked to fill out a survey including preferences for each task, an unweighted NASA TLX survey, and free-form comments. Each participant received a small cash amount after the study.

## 5.7 Results

### 5.7.1 Preferences

Table 5.1 summarizes preferences expressed in the formal study. Chi-square tests reveal that participants prefer *S* over *G* for *SeeABC* ( $\chi^2_{(1,N=12)} = 8.33, p = 0.004$ ), *SeeX+ABC* ( $\chi^2_{(1,N=15)} = 15, p = 0.0001$ ), and *CreateCollection* ( $\chi^2_{(1,N=13)} = 13, p = 0.0003$ ), at Bonferroni-corrected  $\alpha = 0.01$  ( $0.05/5$ ), supporting **H5.1**, **H5.2**, and **H5.4**. Preferences for *Connect-*

*the-dots* ( $\chi^2_{(1,N=14)} = 4.57, p = 0.03$ ) and *ReviewRoute* ( $\chi^2_{(1,N=14)} = 0.29, p = 0.59$ ) do not reach significance after Bonferroni correction for **H5.3** and **H5.5**. Observations of participants and their feedback reveal insights underlying their preferences.

### 5.7.2 On Addressing the Seeing Problem

For the *SeeABC* task, we observed that two strategies were commonly applied to see two or more locations simultaneously in the *G* condition: panning and zooming the map to visually search for locations, and creating a route with multiple locations. In the *S* condition, we observed that participants were able to quickly grasp the idea of using SpaceTokens as constraints, and used zoom-to-fit and zoom-to-preference to address the seeing problem. Participant 5 (P5) noted, “The Space Tokens UI was very easy to figure out. Finding places relative to each other was especially good/useful.”

Many participants mentioned that they frequently ran into problems similar to *SeeX+ABC* (e.g., P7 wrote “I find myself zooming out after searching for a place EVERY SINGLE TIME on Google Maps, which is very frustrating”) and commented that anchor is useful for them. While some participants mentioned it was natural to use a finger to anchor a location, others noted this is complicated because it involves two fingers on a small screen; anchor may be more useful for a large display device.

### 5.7.3 On Addressing the Linking Problem

While participants quickly grasped the idea of using SpaceTokens to address the seeing problem, they required more effort to get familiar with the LinkTool. We observed two

challenges, both of which required learning: (a) To activate the LinkTool, users need to tap a location and then hold the highlighted location to reveal the LinkTool; (b) A gesture is needed to link from the selected location to another location.

After becoming familiar with the LinkTool, however, many participants appreciated being able to connect locations rapidly. “Connect-the-dots is one feature of SpaceTokens that I like very much. I used to plan trips to places and I have to search for each of the routes between every two locations in my list to visit. This is time consuming.” (P13).

All participants were familiar with using Google Maps for directions, and required no additional instructions. However, participants commented on the required mode switch, the number of menu interactions involved to create a route, and the need to repeat the same process to investigate multiple routes involving one or more common locations.

For the *CreateCollection* task, we were surprised by how strongly participants preferred using the CollectionTool, given that (a) the tool itself is merely a structure to organize/connect locations, and (b) participants were also explicitly introduced to the similar list-creation feature in Google Maps. Several participants noted that the CollectionTool could be very useful as a ToDo (or ToVisit) list. We attribute this to the ability to achieve the tight seeing and linking loop that sets the CollectionTool apart from the Google Maps list-creation feature.

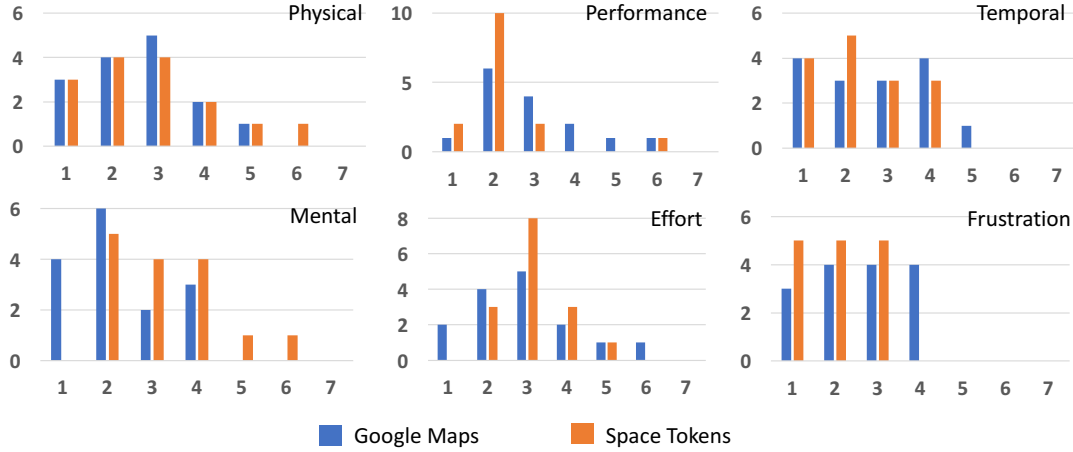
#### **5.7.4 On SpaceBar and the Open-Ended Task**

All participants quickly understood the idea of SpaceBar, due to their familiarity with scrollbars. All were able to use SpaceBar to interact with a route: jumping to different

points of the route and estimating the visible portion of the route from reading the “elevator” indicator. While participants liked the idea of the SpaceBar in general, we found several issues that we believe to have caused the preference split.

First, participants were divided about how the route was oriented while scrolling. There will be some situations in which users prefer the map to automatically align with the travel direction (track-up, as in our implementation), and others in which users prefer fixed alignment (north-up) [Aretz 1991]. We believe the best solution is to allow users to decide on the approach used. Second, participants also mentioned that certain useful features were missing from the SpaceTokens prototype (e.g., alternative routes and transportation options). Last, participants commented that they would like to have both the SpaceBar as well as the features in Google Maps, rather than having to choose between them.

For the open-ended task, participants used the SpaceTokens prototype to explore 14 cities in six countries. In general, for open-ended exploration, users may shift goals to address casual encounters and spur-of-the-moment events. To this end, participants mentioned that they liked the support offered by SpaceTokens for transitioning easily between the seeing and linking tasks. Several participants commented that it could be expensive to switch into (and out of) the Google Maps direction mode just to perform quick measurements, which is common in the planning stage: “After a while, it becomes tedious and repetitive.” (P2).



**Figure 5.8: TLX Results. Lower numbers are preferred for all measures.**

### 5.7.5 NASA TLX Results

Figure 5.8 shows the results of the unweighted NASA TLX survey. A Friedman test finds a significant difference (with  $\alpha = 0.05$ ) between Google Maps and SpaceTokens in performance ( $\chi^2_{(1)} = 4.5, p = 0.03$ ), but no significant differences in all other measurements: mental ( $\chi^2_{(1)} = 3.57, p = 0.06$ ), temporal ( $\chi^2_{(1)} = 2, p = 0.16$ ), physical demand ( $\chi^2_{(1)} = 0.14, p = 0.71$ ), effort ( $\chi^2_{(1)} = 0.11, p = 0.74$ ), and frustration ( $\chi^2_{(1)} = 2.27, p = 0.13$ ).

## 5.8 Discussion

*Learning curve.* Compared with Google Maps, a commercial map applications familiar to all participants, the SpaceTokens prototype requires some learning. The general consensus of our participants is that it requires less effort to learn how to use SpaceTokens to address the seeing problems, than to address the linking problem. However, all participants were able to use SpaceTokens to perform the tasks after practicing for several minutes. Despite the learning curve and the comparison with a mature commercial product, the vast

majority of participants still preferred the SpaceTokens prototype in their questionnaire responses. TLX results indicated participants felt their performance with SpaceTokens was significantly better than with Google Maps. No significant differences were found in other measurements. Note that the TLX results were collected based on the experience of the entire study, not for each individual task. We would be interested in collecting TLX results per task in the future.

*Advantages and disadvantages of SpaceTokens.* SpaceTokens make it possible for users to perform location-centric tasks directly using locations. Consequently, users can perform both seeing and linking tasks in a tighter loop, without the need to switch in and out of wizard dialogs, potentially interrupting the workflow. However, some learning is involved, as discussed earlier, for users to take advantage of SpaceTokens. Another disadvantage of SpaceTokens is the requirement for gestural interaction, including bimanual (e.g., anchor and LinkTool). Some participants suggested that they would appreciate it if the gestures could all be one-handed. However, participants also acknowledged that Google Maps requires extensive text input and menu interaction. We feel there are fundamental trade-offs between classical menu-based map interaction and location-centric interaction. Ultimately, both types of interaction could coexist to cater to individual users' preferences.

### **5.8.1 Design Recommendations**

**Use SpaceTokens.** We recommend map user interface designers to take advantage of SpaceTokens and replace shortcuts with SpaceTokens. Conventional shortcuts are designed for documents, which differ from maps in many aspects. First, a map is a large continuous

multi-scale information space. A user may want to see one or several locations simultaneously, or the areas between those locations (the seeing problem, discussed in Section 5.2). While shortcuts help users to quickly jump to a position in a document, or a webpage in a web browser, shortcuts do not allow users to see multiple important locations at the same time, or specify where the locations should be shown. SpaceTokens provide more flexibility than shortcuts to tackle the seeing problem.

Second, map users often need to link one location to another to investigate travel direction, time, and distance (the linking problem, discussed in Section 5.2). This need to link two locations for the purpose of investigating the properties of the path between them is uncommon in document reading, and thus is not supported by shortcuts. With SpaceTokens, users can use locations directly as building blocks to address the linking problems.

**Integrate SpaceBar into the existing map user interfaces.** Although our user study compared SpaceBar with the direction review interface in Google Maps, we believe these two are not mutually exclusive. Google’s direction review interface provides detailed textual information (e.g., the name of the route, distance, and time) about the visible portion of a route, but only supports users in reviewing the route in detail and in temporal order. SpaceBar, much like a scrollbar, provides a simple and linear 1D overview to a potentially complex and nonlinear path. At the same time, SpaceBar also allows users to revisit any portion of a route efficiently. SpaceBar and Google’s existing route reviewing interface are actually complementary. Many participants have also commented that they would like to see the two co-exist. As demonstrated, SpaceBar occupies a small footprint and can be integrated into existing map user interfaces with minimal modifications.

**Consider alternative map interaction models.** Our last recommendation is a philo-



sophical one. The aim is to encourage designers to think about different interaction models that can complement the zooming user interface and menu/dialogue-based interactions. In this chapter, we discussed the idea of making locations first-class citizens of map interaction. Can this idea be extended to spatial entities in general, such as paths and areas? Additionally, rather than using interactive widgets such as SpaceTokens to support location-centric (or spatial-entity-centric) interactions, can we leverage speech and gesture to support interaction with maps? These are interaction questions that we believe that designers should think about. In Section 6.2, we will discuss two more specific directions we are pursuing.

## 5.9 Limitations

**SpaceTokens need to be created before they can be used.** SpaceTokens are useful only if they are available when a user needs them. One pilot study participant commented that he would probably use SpaceTokens if he knew the locations with which he would repeatedly interact (e.g., during a trip or when executing planned activities); in other situations, he would most likely use the existing search and autocomplete feature to revisit a location. Bookmarks have a similar limitation in that they are also useful only if they are available when a user needs them. That limitation has been addressed by implicit bookmark creation based on personalized usage patterns [Zhao et al. 2013]—an approach we believe could be adapted to create SpaceTokens.

**Links are created manually.** The LinkTool requires a user to *explicitly* establish a link between two locations. Other approaches could be supported to preemptively establish a

link whenever two locations are selected. For example, when a user selects two SpaceTokens, a suggested route could appear on the map. The user could then tap the suggested route to formally establish the link.

**SpaceBar works best for simple routes.** While our SpaceBar prototype works well for simple routes, some cases require more thought. For example, how can the SpaceBar handle a spiral route? There will be cases in which it will be impossible for the SpaceBar elevator to make only a desired portion of the route visible while the rest of the route is invisible. In this case, the system will need to relax the constraints specified by the elevator and provide proper visual feedback. For example, the route and the elevator could be color-coded to indicate the correspondence. The elevator may also need to be segmented into several pieces to represent discontinuous parts of a route that are clipped against the edges of the screen.

## 5.10 Conclusions

In this work, we asked the question, “Why can’t users perform location-centric tasks *directly* using locations?” To attempt to answer it, we introduced SpaceTokens, interactive widgets that make location a first-class citizen of map interaction. SpaceTokens empower users to perform location-centric tasks rapidly. We explored the ideas of using locations as constraints to address the seeing problem, and using locations as building blocks to address the linking problem. Building on the idea of making location a first-class citizen, we also presented SpaceBar, a scrollbar-like navigation instrument that leverages decades of work on scrollbars to help users interact with a route.

We implemented SpaceTokens and related tools in a prototype iOS application. In a qualitative user study, we evaluated the prototype application on an iPhone in comparison with Google Maps. Participants overwhelmingly preferred the approaches used in our prototype for the majority of the tasks they tried. Ultimately, we envision spatial entities in general (locations, paths, areas) could be first-class citizens of map interaction.

### *Conclusions and Future Work*

#### **6.1 Summary of Contributions**

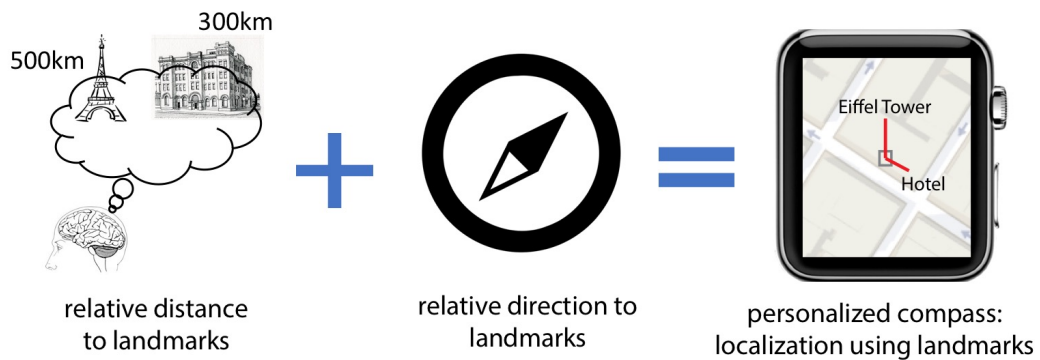
Digital maps play an integral role in our daily lives. However, the multi-scale nature of maps makes it difficult to use them on a computer display. This issue worsens on small display devices (e.g., smartphones and smartwatches). Ironically, small display devices are the ones people rely upon when on the move or lost; that is, when maps are needed the most.

We have introduced three personalized navigation instruments to tackle three navigation challenges. Each instrument is the result of a novel combination of visualization and interaction techniques.

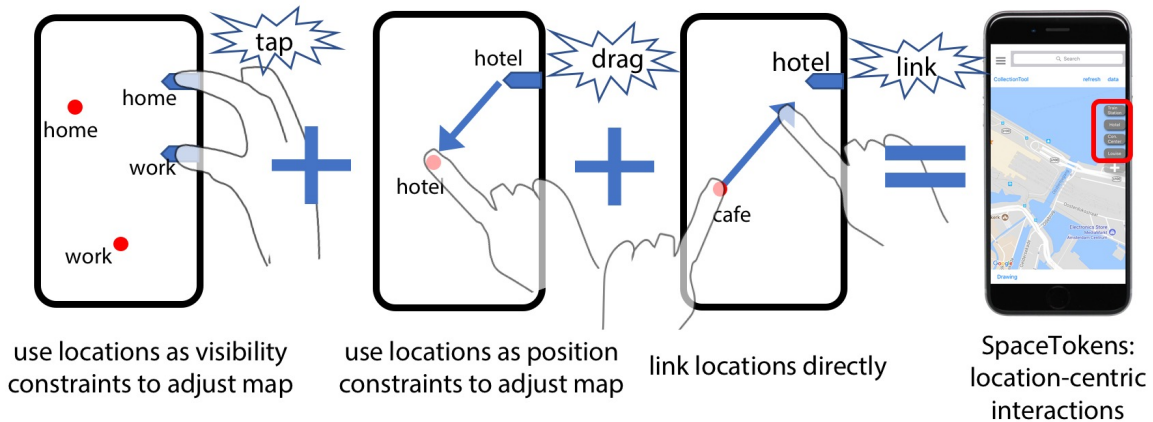
To help users establish a frame of reference, we introduced P-Compass, a multi-needle compass that extends the concept of a conventional compass. The design of P-Compass is inspired by the observation that people often use landmarks as references to localize unfamiliar locations. Combining the information given by relative distances to landmarks and relative directions to landmarks, P-Compass visualizes the locations of known landmarks with respect to a reference point, as shown in Figure [6.1\(a\)](#).

To help users see and link locations, we introduced SpaceTokens. SpaceTokens are

③ how to help users establish a frame of reference?



④ how to help users see and link locations?



⑤ how to help users interact with route?



**Figure 6.1: Concepts underlying three personalized navigation instruments. (a) P-Compass visualizes locations of known landmarks with respect to a reference point. (b) SpaceTokens facilitate users to see and link locations rapidly. (c) SpaceBar is a scrollbar for a route.**

interactive widgets that appear like whiteboard magnets hovering on the edges of a map. SpaceTokens can be used to specify visibility and position constraints to adjust the visible part of a map. SpaceTokens can also be used as building blocks to help users establish relationships between locations. Combining the two concepts, SpaceTokens empower users to perform location-centric tasks (see and link locations) directly and rapidly, as shown in Figure 6.1(b).

Finally, to help users interact with a route on a map efficiently, we introduced SpaceBar. SpaceBar is essentially a scrollbar for a route. We adapt the idea of a scrollbar in a document browser to a route on a map, as illustrated in Figure 6.1(c). A SpaceBar allows a user to understand a route in context when examining its details, and to jump to and revisit any portion of the route efficiently.

A common thread running through our navigation instruments is *personalization*—the strategic use of personal important spatial entities. Personalization makes these navigation instruments simple and compact, which are advantageous for mobile devices.

We prototyped the three navigation instruments in a custom map application, used them in real-world settings, and derived scenarios for evaluating them. We validated our designs in two formal studies. In one study of five navigation tasks, we compared P-Compass with Wedge, a state-of-the-art interface technique for visualizing off-screen objects. P-Compass performs better for four inference tasks, while Wedge is better for a locating task. In another study that compared SpaceTokens and SpaceBar with Google Maps, participants overwhelmingly preferred a research prototype using SpaceTokens and SpaceBar over Google Maps on identical smartphones for three of five common map tasks. Based on our analyses and study results, we offered design recommendations, showed how P-

Compass and Wedge were complementary, and suggested how SpaceTokens and SpaceBar could be integrated with Google Maps.

## 6.2 Future Work

During the course of this dissertation, we discovered several areas for future work. In this section, we organize the discussion into five directions: 1) From P-Compass to SpacePiece; 2) P-Compass for VR/AR; 3) Audio personalized compass; 4) Spatial entities as first-class citizens of map interactions; and 5) Speech and Sketch Map UI. The first three are related to P-Compass (Chapter 4) and the remaining two are loosely related to SpaceTokens and SpaceBar (Chapter 5). Each direction is self-contained and readers can approach each independently. While we make an effort to provide examples and sketches for these ideas, each direction is meant to stimulate discussion and explore research questions, not to offer solutions.

### 6.2.1 From P-Compass to SpacePiece

The P-Compass described in Chapter 4 has more features than could feasibly be evaluated in our initial user study. There are several aspects of P-Compass that we would like to investigate further. For example, we would like to explore how the distribution of POIs on a P-Compass may affect readability and estimation accuracy, and develop a POI-selection algorithm (Section 4.2.1).

Our analysis suggests that Wedge and P-Compass are complementary. In fact, a P-Compass can be smoothly transformed into a set of Wedges, and vice versa. As a set of

Wedges are pulled in from the periphery of a display to a single reference point, and each triangle is reduced to a needle during the process, a P-Compass is formed. An inverse transformation can convert a P-Compass to a set of Wedges. Future research is needed to investigate several aspects of the transition described above: How should the transition be visualized? When should the transition happen? What are the advantages and disadvantages to users of supporting the transition? We believe that implementing this transition could shed some light in unifying P-Compass and Wedge; however, this research direction is beyond the scope of this dissertation.

We are also interested in evaluating the multi-scale feature (Section 4.2.2) through a formal user study. For independent information in mixed scales (Figure 4.3a), we could ask participants to use information at a single scale to perform any of the five tasks. For related information in mixed scales (Figure 4.3b), we could ask participants to localize a POI (the LOCALIZE and LOCATE+ tasks, discussed in Section 4.3.2), using information from two or more scales. Factors could be the number of scales, ratio of scales, and distribution of POIs. The control condition for this study could present all information at a single scale. Performance measurements such as completion time, errors and preferences could be collected.

As discussed in Section 4.5.2, our prototype was bootstrapped with a list of manually determined locations as the location database. As we also noted, there is research in data mining and machine learning on building a personalized location database. While that research is beyond the scope of this dissertation, we are interested in working with others investigating these areas to refine our prototype so that the POIs are automatically personalized for the user.



Finally, we believe that P-Compass is an important step toward realizing a visualization technique we call a *SpacePiece*. A timepiece is a familiar instrument, often worn on a wrist, to answer the question “What time is it?” with a single glance. In contrast, we ask whether it would be possible to design a SpacePiece that could answer the question “Where is  $x$ ?” with a single glance? This leads us to ask what visual representation should be used by a SpacePiece? We suspect a SpacePiece will require the design of a representation that enlists not just POIs, but also spatial entities such as paths, districts, and boundaries, which are important to wayfinding [Lynch 1960].

### 6.2.2 P-Compass for VR/AR

P-Compass need not be restricted to map user interfaces. In Section 4.6, we described our attempt to integrate a P-Compass into Google StreetView and Apple’s 3D map application. The StreetView application suggests how a P-Compass might be used for AR, and the 3D map application suggests how a P-Compass might be integrated into VR. In both cases, users employ either a keyboard and mouse or multi-touch gestures to interact with the environment. Inspired by the popularity of smartphone VR and AR, we next sketch a P-Compass application in which users can use physical body movement to explore a virtual world.

The current version (iOS11) of Apple Maps on iOS devices introduces a VR walk-through mode [Apple 2018]. In this VR walkthrough mode, a user’s movement in a physical environment is mapped to the camera movement in a virtual environment. While holding an iPhone (or an iPad), a user sees the world from the perspective of a giant. As the



**Figure 6.2: Apple’s VR walkthrough interface.**

user walks, turns, or kneels down to change their height, the virtual world is updated as if the user is walking in the (virtual) world. Figure 6.2 shows three screenshots of the Apple VR walkthrough mode. The user first sees an initial view of London (left image). As the user turns her head to the right, the UK Parliament comes into view (center image). As the user continues turning her head to the right and pitches her head downward to see the river, the Parliament and the initial park exit the field of view (right image).

The smartphone VR walkthrough mode offers a novel experience of exploring a virtual world. However, the user may feel this experience to be similar to seeing the world through a keyhole, albeit a movable one. As in the case of StreetView and 3D map exploration (Section 4.6), the “Where is  $x$ ?” problem described in Chapter 4 exists here too. It should be noted that even if the user has a field of view equivalent to that of full human binocular vision, they can still see only a fraction of the world at any given time; there will always be something behind the user.

Similar to the StreetView and 3D map applications described in Section 4.6, a P-Compass

can be integrated into this smartphone VR application. However, for all these experiences, a number of questions need to be answered. For example, how can a P-Compass effectively communicate the relationships of off-screen POIs with respect to a user's field of view in a 3D environment? In a 2D map, a 2D box can be used to indicate the boundary of a display. But how about in 3D? How can a P-Compass gracefully and effectively communicate the moment when an off-screen POI comes into view, or a visible POI becomes invisible? Further, how can a P-Compass visualize its needles in 3D for a user to easily read their relative lengths and directions? When compass needles are distributed on a sphere rather than on a circular plane, foreshortening will become an issue.

In the context of a geospatial application such as StreetView, a 3D map, or smartphone VR/AR, we suspect P-Compass can be integrated into these scenarios with relatively small modifications. For example, one may use a wedge, rather than a box, to indicate field of view. A planar P-Compass will also likely suffice when orientation is conceived of as 2D compass direction on an idealized locally planar surface. In contrast, when exploring a more general 3D environment (e.g., a 3D CAD model or a scan of an anatomical organ), we believe that more effort will need to be applied to the design of a 3D P-Compass.

### **6.2.3 Audio Personalized Compass**

P-Compass need not be visual. Is it possible to design an *audio* P-Compass? One advantage of having an audio rather than a visual P-Compass is that a user's visual attention can be reserved for tasks that demand it. Here again, we would like to note that by audio P-Compass we do not mean an audio compass that simply uses spatial sound to communicate

the *direction* (and potentially distance) of a POI, for the purpose of helping a user to localize where a POI is respect to a user <sup>1</sup>. Instead, we are interested in a *reverse* problem: how can we help a user localize herself with respect to certain *a priori* POIs by using spatial sound (potentially multiple sources at the same time). In other words, we are interested in designing a simple and efficient *auditory* representation that could accomplish what a P-Compass achieves visually.

The terms “simple” and “efficient” are important. One naïve solution would be to use computer-generated speech to announce the address or describe the details of a user’s location; for example, “You are at the intersection of Bedford Row and Princeton St., WC1R 4LR, London.” This approach is neither simple nor efficient, as it requires effort to parse the information spanning the duration of the announcement of the sentence. Additionally, for a user without detailed knowledge about the area (e.g., a visitor), the information may not be useful.

Another naïve solution would be to use computer-generated speech to literally describe the geometric relationship between a reference point and landmarks. For example, “King’s Cross is in front of you 37 degree to the right and is about 1.5 km away. Big Ben is behind you 71 degrees to the left, and is about 5 km away.” This approach is neither simple nor efficient. Speech transmission and processing takes time, and is often not suitable for frequent real-time updates. Is it possible to design an audio representation that can communicate all the information within a split second?

Police cars and ambulances on the street may offer some clues. Suppose a user standing

---

<sup>1</sup>As discussed in Section 3.2.4, this is a problem with many known solutions [Loomis et al. 1998; Wenzel et al. 1993; Begault and Trejo 2000].

outside hearing the sound of an ambulance and a police car. For simplicity, suppose the two cars are stationary (e.g., stuck in a traffic jam), located in different directions, on the same ground plane (no elevation difference), and at different distances. Can the user tell the direction to the police car, and the direction to the ambulance? Furthermore, can the user tell which one is closer? And can the user have a rough sense of where the police car and the ambulance are *with respect to* some other POI?

Research has shown that users with normal hearing are capable of distinguish the direction to a sound source and can roughly estimate the distance to a sound source [Loomis et al. 1998; Wenzel et al. 1993; Begault and Trejo 2000]<sup>2</sup>. Now, suppose the user *knows* where the police car and the ambulance are, but the user does not know where she is. Can she use her knowledge of the locations of the police car, the ambulance, and their positions relative to her, to interpolate her location? This is the motivation for an *audio* personalized compass.

One can imagine a world in which buildings can talk to users. Suppose a user is lost in the middle of Manhattan. Suppose the Empire State Building and World Trade Center can send distinguishable audio signals to the users, can the user leverage the audio signals to localize herself?



**Figure 6.3: Drawing interface to mark custom spatial entities.** (a) Holding the drawing button at the bottom left corner of the display, the user can then (b) sketch a line to specify a path or (c–e) sketch a (nearly) closed outline and scribble inside it to specify an area. (f) Tapping the “+” button at the lower right corner of the display creates a SpaceToken for the path or area.

## 6.2.4 Spatial Entities as First-Class Citizens of Map Interaction

In Chapter 5, we discussed using SpaceTokens to represent one or more locations, or a route between locations created with the LinkTool. SpaceTokens should be able to represent more general spatial entities. We are currently experimenting with SpaceTokens that represent arbitrary paths and areas. We have prototyped a simple drawing interface, shown in Figure 6.3, with which a user sketches a path or area that can be turned into a SpaceToken. We extended our prototype to support using these more general spatial entities as

<sup>2</sup>Humans can tell the *direction* better than the *distance* of a spatial sound source. It is considered a solved problem to generate spatial sound for a user to perceive at adequate directional accuracy [Loomis et al. 1998; Wenzel et al. 1993]. However, it is still challenging to generate spatial sound for a user to perceive distance accurately [Loomis et al. 1998].

position or visibility constraints to address the seeing problem. At the conclusion of the SpaceTokens and SpaceBar formal study (Section 5.6.4), we introduced the drawing interface and these extended capabilities to participants, many of whom said it was fun to create their own spatial entities and that it would be helpful to use SpaceTokens representing any spatial entities they interact with frequently.

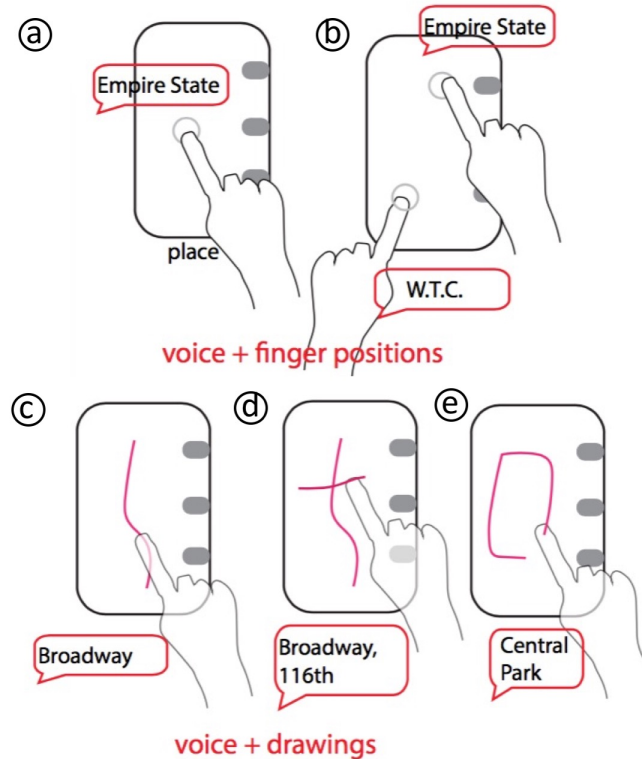
Spatial entities could also be used as building blocks to address the linking problem. However, what would it mean to link locations, paths, and areas to each other in various combinations? (One possible behavior is to create a path that connects the closest points of two spatial entities.) Our prototype currently supports linking locations only, leaving the exploration of other kinds of linking to future work.

In addition to the seeing and linking problems, SpaceTokens can be used to address other problems, such as the searching problem. For example, a user could select a number of SpaceTokens, and perform a location search under the constraints specified by the selected SpaceTokens.

## 6.2.5 Speech + Sketch Map UI

“Can a user perform spatial-entity–centric interactions *without* SpaceTokens?” This question emerged as we investigated SpaceTokens and the new paradigm of map interaction supported by them. People have a rich variety of ways to communicate their thoughts—speech, gestures, drawing, to name a few. Can we leverage our rich power of expression to support spatial-entity–centric map interactions? Here we will briefly describe an idea that we term “Speech+Sketch Map User Interface.”





**Figure 6.4: Speech + sketch map user interface on a smartphone.**

When we describe a location (e.g., a restaurant) to a friend, we often use gestures to draw imaginary spatial entities (e.g., roads, landmarks) in the air, and use speech to label the drawn entities. Through imaginary drawing and labeling, we set up a mental map for ourself and the friend to facilitate the discussion. Now, can a computer system take the same information as inputs and display a physical map that satisfies the specified constraints?

Let us use several examples to walk through this idea. Suppose a user wants to examine the Empire State Building and its vicinity on a map. The user could point with their finger to denote where the Empire State Building should be on a smartphone display (i.e., specifying a position constraint), and then use speech to label the constraint. The system would next adjust the visible map such that the Empire State Building appears at the specified position

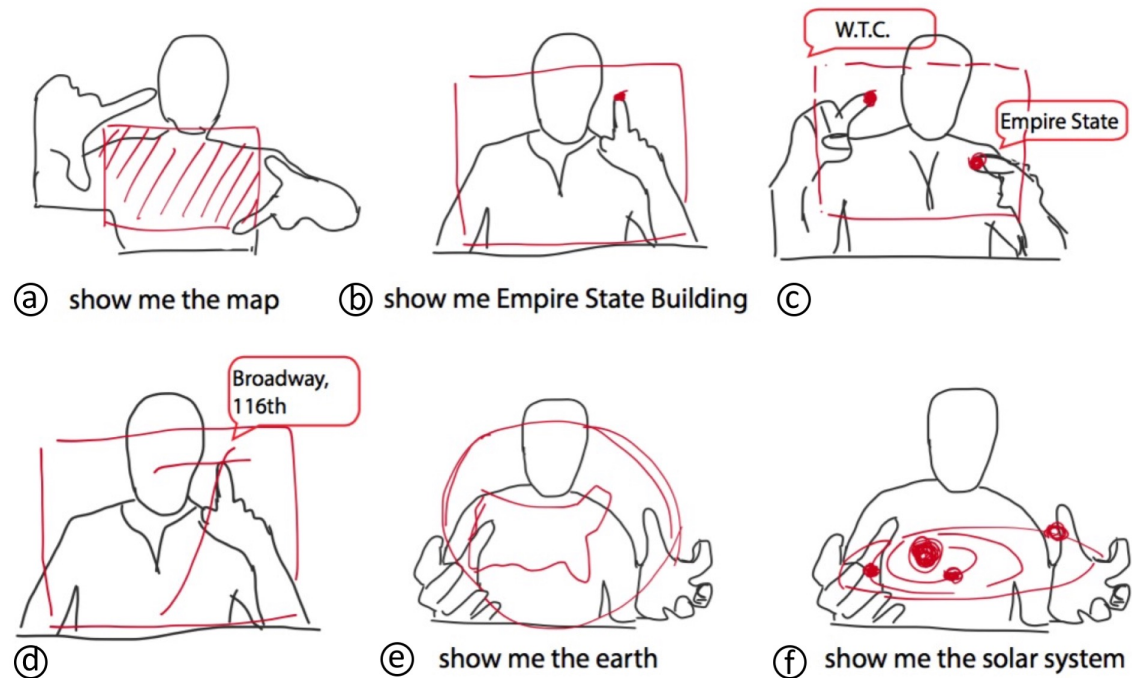


(Figure 6.4a).

Users can also specify multiple constraints at once. For example, a user can use her right index finger to specify the position of the Empire State Building, accompanying the voice label “Empire State Building.” While the right index finger touches the display, the user uses the left index finger (and voice) to indicate the position of World Trade Center (Figure 6.4b). Upon receiving the constraints, the system adjusts the visible portion of the map to meet the user’s specifications.

Alternatively, one could sketch to specify constraints. Suppose the user wants to see a portion of Broadway that has two small bends. The user can sketch the shape of Broadway, accompanied by voice labeling (Figure 6.4c). The system then adjusts the visible map to meet the specified constraints. Similarly, a user can sketch two lines, labeled as Broadway and 116th Street, to designate the intersection of the two roads (Figure 6.4d). A map is then adjusted to meet the specified constraints. Finally, the user can sketch out shapes as constraints. Figure 6.4(e) shows a user sketching the boundary of Central Park while verbally labeling the sketched contour as Central Park. The system adjusts the map such that Central Park is shown at the scale and position satisfying the constraints specified by the user.

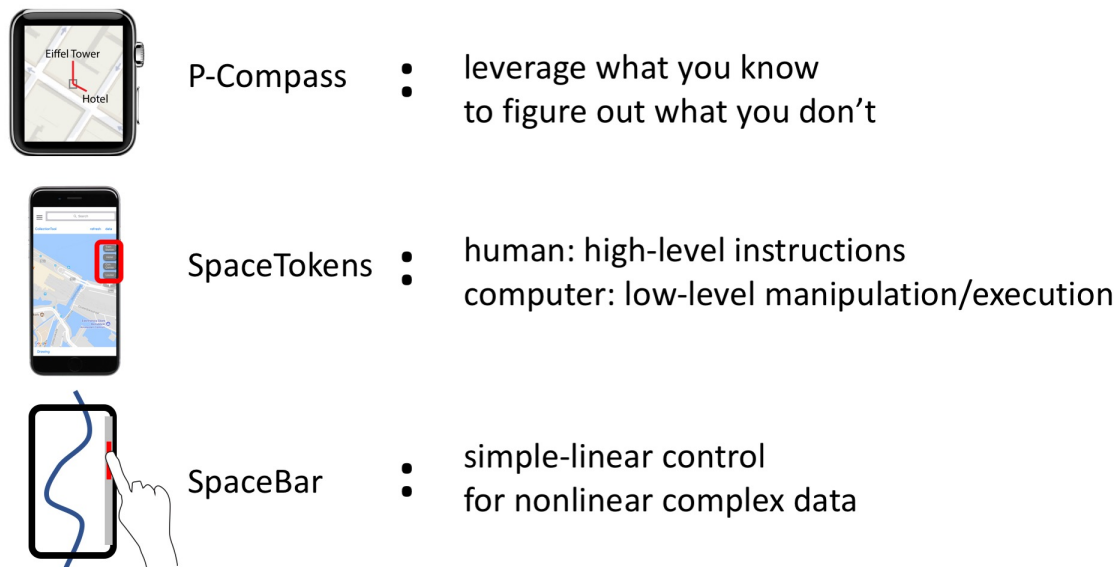
While we have just used hypothetical examples on a smartphone to illustrate the concept of using sketch and speech to specify constraints, one can imagine applying this concept to table-top or wall-size user interfaces. Furthermore, we believe this concept could also be extended to 3D user interfaces. Figure 6.5 illustrates a scenario in a VR or AR user interface. Here, a user first uses her hands to mark a region in which a map is to be displayed (Figure 6.5a). The user then uses speech and sketch to specify the constraints



**Figure 6.5: Speech + sketch map user interface in 3D (e.g., VR/AR).**

(Figure 6.5b-d). The user may also use gestures to specify volume constraints. For example, the user may use two hands to specify where a globe or solar system (accompanied by voice labeling) should be displayed (Figure 6.5e-f). A user may apply the same concept to specify the constraints of other spatial information, such as a CAD model of a machine, and let the system retrieve and display the 3D furniture model.

Prior work such as the imaginary user interface [Gustafson et al. 2010], or furniture model retrieval with gestures [Holz and Wilson 2011], touches upon parts of the concepts sketched here. However, none of this earlier work leveraged the combination of speech and sketch, or had a system to output visual information in response to a user’s inputs. We believe Speech + Sketch Map User Interfaces could change the way how users interact with maps and other spatial information.

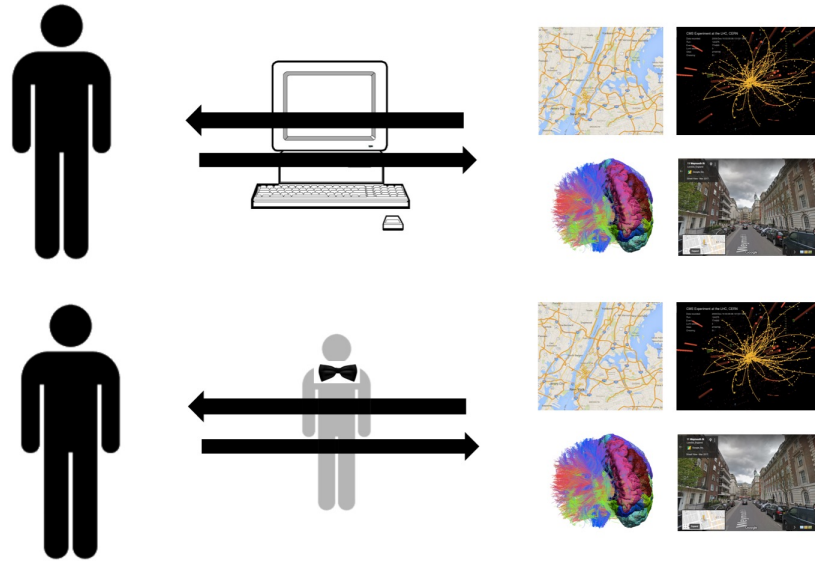


**Figure 6.6: The high-level philosophy of our ideas.**

## 6.3 Towards Personalized UIs for Spatial Information

Until now, we have structured our discussion in a problem-solution pattern. We have focused our effort on challenges of navigation in map user interfaces. In response, we have introduced three personalized navigation instruments. While our contributions are in the form of navigation tools, it is worthwhile to take a step back and look at the ideas embodied by these tools. Figure 6.6 lists the ideas represented by each navigation instrument:

- In P-Compass, we aimed to leverage what a user know to help them figure out what they do not know (to establish a reference frame for an unknown location in our case).
- In SpaceTokens, we aimed to design a tool to allow a user to specify high-level instructions, so that a system could carry out low-level tasks such as manipulating an information space or connecting spatial entities.
- In SpaceBar, we aimed to devise a simple and linear control that would enable a user



**Figure 6.7: (Top) Conventional user interfaces for spatial information. (Bottom) Personalized user interfaces for spatial information.**

to comprehend and interact with complex, nonlinear data.

It should be noted that each instrument is our attempt to realize one instance of an idea. There are many other instances of these ideas, five of which are discussed above in Section 6.2.

While maps and their related geospatial information are the foci in this dissertation, we believe our research questions are universal across many types of spatial information, and that the ideas to address these questions are also general. Thus, we would like to extend our investigation to fields such as architecture, computer-aided design, and scientific or medical visualization.

Finally, we should take a step back and examine how we interact with maps or spatial information from a meta-perspective. Each of us has a certain level of knowledge about the world (i.e., the cognitive maps in our head are not empty), and has a rich variety of ways (e.g., speech and gesture) to express our thoughts and intents. Unfortunately, conventional

map user interfaces take none of this into account. The predominant way of seeking information through a map user interface is through repeated low-level mechanical interactions (e.g., key pressing, mouse motion, or multi-touch gesture). A conventional spatial user interface is akin to a window, for which a user adjusts the portion of the information space to see (Figure 6.7). It is almost as if the interface is the bottleneck for information seeking.

Can a spatial information user interface behave like a friendly assistant, with whom a user can work to complete information-seeking or analysis tasks? For example, when a user provides free-form, high-level instructions to a (computer) assistant, the assistant can infer the user's intention and carry out the low-level tasks of manipulating and presenting the information space. When a user has questions, the assistant may convey important and unknown information in the context of what a user has already known. Many more examples are possible. We envision this type of user interface as a personalized spatial-information user interface (Figure 6.7), in which a computer acts as a personal assistant rather than a window. Each of our navigation instruments individually represents a solution to a research question. Taken together, they form the foundation for a vision towards personalized user interfaces for spatial information.

---

## Bibliography

- Adar, E., Teevan, J., and Dumais, S. T. (2008). “Large scale analysis of web revisitation patterns.” In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, pp. 1197–1206.
- Agrawala, M. and Stolte, C. (2001). “Rendering effective route maps: improving usability through generalization.” In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, pp. 241–249.
- Alexander, J., Cockburn, A., Fitchett, S., Gutwin, C., and Greenberg, S. (2009). “Revisiting read wear: analysis, design, and evaluation of a footprints scrollbar.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1665–1674.
- Apperley, M. D., Tzavaras, I., and Spence, R. (1982). “A bifocal display technique for data presentation.” In: *Proceedings of Eurographics*. Vol. 82, pp. 27–43.
- Appert, C. and Fekete, J.-D. (2006). “OrthoZoom scroller: 1D multi-scale navigation.” In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, pp. 21–30.
- Apple (2017b). *Apple Maps location sharing*. URL: <https://support.apple.com/en-us/HT201493> (visited on 04/23/2017).
- (2017a). *Apple Maps*. URL: <https://www.apple.com/ios/maps/> (visited on 12/06/2017).
- (2018). *3D and Flyover*. URL: <https://help.apple.com/iphone/11/#/iph81a3f978> (visited on 01/08/2018).
- Aretz, A. J. (1991). “The design of electronic map displays.” In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 33.1, pp. 85–101.
- Baudisch, P. M. and Krumm, J. C. (2013). *Location-to-landmark*. Tech. rep.

- Baudisch, P. and Rosenholtz, R. (2003). “Halo: a technique for visualizing off-screen objects.” In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pp. 481–488.
- Baudisch, P., Good, N., and Stewart, P. (2001). “Focus plus context screens: combining display technology with visualization techniques.” In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, pp. 31–40.
- Bederson, B. B. (2011). “The promise of zoomable user interfaces.” In: *Behaviour & Information Technology* 30.6, pp. 853–866.
- Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., and Furnas, G. (1996). “Pad++: A zoomable graphical sketchpad for exploring alternate interface physics.” In: *Journal of Visual Languages & Computing* 7.1, pp. 3–32.
- Begault, D. R. and Trejo, L. J. (2000). “3-D sound for virtual reality and multimedia.” In:
- Bell, B., Feiner, S., and Höllerer, T. (2001). “View management for virtual and augmented reality.” In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, pp. 101–110.
- Brosz, J., Nacenta, M. A., Pusch, R., Carpendale, S., and Hurter, C. (2013). “Transmogrification: causal manipulation of visualizations.” In: *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, pp. 97–106.
- Burigat, S. and Chittaro, L. (2011). “Visualizing references to off-screen content on mobile devices: A comparison of Arrows, Wedge, and Overview+ Detail.” In: *Interacting with Computers* 23.2, pp. 156–166.
- Burigat, S., Chittaro, L., and Gabrielli, S. (2006). “Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches.” In: *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*. ACM, pp. 239–246.
- Burigat, S., Chittaro, L., and Vianello, A. (2012). “Dynamic visualization of large numbers of off-screen objects on mobile devices: an experimental comparison of wedge and overview+ detail.” In: *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM, pp. 93–102.
- Cao, Y., Wang, H., Wang, C., Li, Z., Zhang, L., and Zhang, L. (2010). “Mindfinder: interactive sketch-based image search on millions of images.” In: *Proceedings of the 18th ACM international conference on Multimedia*. ACM, pp. 1605–1608.

- Carpendale, M. S. T. and Montagnese, C. (2001). "A framework for unifying presentation space." In: *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, pp. 61–70.
- Chen, T., Cheng, M.-M., Tan, P., Shamir, A., and Hu, S.-M. (2009). "Sketch2photo: Internet image montage." In: *ACM Transactions on Graphics (TOG)* 28.5, p. 124.
- Chittaro, L. and Burigat, S. (2004). "3D location-pointing as a navigation aid in Virtual Environments." In: *Proceedings of the working conference on Advanced visual interfaces*. ACM, pp. 267–274.
- Chuah, M. C., Roth, S. F., Mattis, J., and Kolojejchick, J. (1995). "SDM: Selective dynamic manipulation of visualizations." In: *Proceedings of the 8th annual ACM symposium on User interface and software technology*. ACM, pp. 61–70.
- Cockburn, A. and McKenzie, B. (2001). "What do web users do? An empirical analysis of web use." In: *International Journal of human-computer studies* 54.6, pp. 903–922.
- Cockburn, A., Karlson, A., and Bederson, B. B. (2008). "A review of overview+ detail, zooming, and focus+ context interfaces." In: *ACM Computing Surveys (CSUR)* 41.1, p. 2.
- Donelson, W. C. (1978). "Spatial management of information." In: *ACM SIGGRAPH Computer Graphics*. Vol. 12. 3. ACM, pp. 203–209.
- Egenhofer, M. (1999). "Spatial information appliances: A next generation of geographic information systems." In: *1st Brazilian workshop on geoinformatics, Campinas, Brazil*.
- Eitz, M., Richter, R., Hildebrand, K., Boubekeur, T., and Alexa, M. (2011). "Photosketcher: interactive sketch-based image synthesis." In: *IEEE Computer Graphics and Applications* 31.6, pp. 56–66.
- Elmqvist, N., Henry, N., Riche, Y., and Fekete, J.-D. (2008). "Melange: space folding for multi-focus interaction." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1333–1342.
- Feiner, S. and Shamash, A. (1991). "Hybrid user interfaces: Breeding virtually bigger interfaces for physically smaller computers." In: *Proceedings of the 4th annual ACM symposium on User interface software and technology*. ACM, pp. 9–17.
- Feiner, S., MacIntyre, B., Höllerer, T., and Webster, A. (1997). "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment." In: *Personal Technologies* 1.4, pp. 208–217.



- Furnas, G. W. (2006). “A fisheye follow-up: further reflections on focus+ context.” In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, pp. 999–1008.
- Furnas, G. W. and Bederson, B. B. (1995). “Space-scale diagrams: Understanding multiscale interfaces.” In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., pp. 234–241.
- Golledge, R. G. (1999). “Human wayfinding and cognitive maps.” In: *Wayfinding behavior: Cognitive mapping and other spatial processes*, pp. 5–45.
- Google (2017b). *Google Maps iOS SDK*. URL: <https://developers.google.com/maps> (visited on 04/23/2017).
- (2017c). *Google Maps mobile*. URL: <https://www.google.com/maps/about/> (visited on 04/23/2017).
- (2017d). *Google Maps Real-Time Location Sharing*. URL: <https://blog.google/products/maps/share-your-trips-and-real-time-location-google-maps/> (visited on 04/23/2017).
- (2017a). *Google Maps*. URL: <https://maps.google.com> (visited on 12/06/2017).
- Grabler, F., Agrawala, M., Sumner, R. W., and Pauly, M. (2008). *Automatic generation of tourist maps*. Vol. 27. 3. ACM.
- Grubert, J., Heinisch, M., Quigley, A., and Schmalstieg, D. (2015). “Multifi: Multi fidelity interaction with displays on and around the body.” In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, pp. 3933–3942.
- Gustafson, S. G. and Irani, P. P. (2007). “Comparing visualizations for tracking off-screen moving targets.” In: *CHI’07 Extended Abstracts on Human Factors in Computing Systems*. ACM, pp. 2399–2404.
- Gustafson, S. (2008). “Visualizing off-screen locations on small mobile displays.” MA thesis. University of Manitoba.
- Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. (2008). “Wedge: clutter-free visualization of off-screen locations.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 787–796.
- Gustafson, S., Bierwirth, D., and Baudisch, P. (2010). “Imaginary interfaces: spatial interaction with empty hands and without visual feedback.” In: *Proceedings of the 23rd annual ACM symposium on User interface software and technology*. ACM, pp. 3–12.

- Han, J. Y. (2005). “Low-cost multi-touch sensing through frustrated total internal reflection.” In: *Proceedings of the 18th annual ACM symposium on User interface software and technology*. ACM, pp. 115–118.
- Heer, J., Mackinlay, J. D., Stolte, C., and Agrawala, M. (2008). “Graphical histories for visualization: Supporting analysis, communication, and evaluation.” In: *Visualization and Computer Graphics, IEEE Transactions on* 14.6, pp. 1189–1196.
- Heuten, W., Henze, N., Boll, S., and Pielot, M. (2008). “Tactile wayfinder: a non-visual support system for wayfinding.” In: *Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges*. ACM, pp. 172–181.
- Hill, W. C., Hollan, J. D., Wroblewski, D., and McCandless, T. (1992). “Edit wear and read wear.” In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pp. 3–9.
- Holland, S., Morse, D. R., and Gedenryd, H. (2002). “AudioGPS: Spatial audio navigation with a minimal attention interface.” In: *Personal and Ubiquitous computing* 6.4, pp. 253–259.
- Holz, C. and Feiner, S. (2009). “Relaxed selection techniques for querying time-series graphs.” In: *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, pp. 213–222.
- Holz, C. and Wilson, A. (2011). “Data miming: inferring spatial object descriptions from human gesture.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 811–820.
- Igarashi, T. and Hinckley, K. (2000). “Speed-dependent automatic zooming for browsing large documents.” In: *Proceedings of the 13th annual ACM symposium on User interface software and technology*. ACM, pp. 139–148.
- Ion, A., Chang, Y.-L. B., Haller, M., Hancock, M., and Scott, S. D. (2013). “Canyon: providing location awareness of multiple moving objects in a detail view on large displays.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 3149–3158.
- Isaacman, S., Becker, R., Cáceres, R., Kobourov, S., Martonosi, M., Rowland, J., and Varshavsky, A. (2011). “Identifying important places in peoples lives from cellular network data.” In: *Pervasive computing*. Springer, pp. 133–151.
- Ishak, E. W. and Feiner, S. K. (2006). “Content-aware scrolling.” In: *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, pp. 155–158.

- Ishii, H. and Ullmer, B. (1997). “Tangible bits: towards seamless interfaces between people, bits and atoms.” In: *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. ACM, pp. 234–241.
- Jacobs, C. E., Finkelstein, A., and Salesin, D. H. (1995). “Fast multiresolution image querying.” In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, pp. 277–286.
- Javed, W., Ghani, S., and Elmqvist, N. (2012). “Polyzoom: multiscale and multifocus exploration in 2d visual spaces.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 287–296.
- Jul, S. and Furnas, G. W. (1998). “Critical zones in desert fog: aids to multiscale navigation.” In: *Proceedings of the 11th annual ACM symposium on User interface software and technology*. ACM, pp. 97–106.
- Käser, D. P., Agrawala, M., and Pauly, M. (2011). “FingerGlass: efficient multiscale interaction on multitouch screens.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 1601–1610.
- Keahey, T. A. and Robertson, E. L. (1996). “Techniques for non-linear magnification transformations.” In: *Information Visualization’96, Proceedings IEEE Symposium on*. IEEE, pp. 38–45.
- Kim, J., Zhang, A. X., Kim, J., Miller, R. C., and Gajos, K. Z. (2014). “Content-aware kinetic scrolling for supporting web page navigation.” In: *Proc. of UIST*.
- Kim, Y.-S., Hullman, J., and Agrawala, M. (2016). “Generating Personalized Spatial Analogies for Distances and Areas.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 38–48.
- Kjelstrup, K. B., Solstad, T., Brun, V. H., Hafting, T., Leutgeb, S., Witter, M. P., Moser, E. I., and Moser, M.-B. (2008). “Finite scale of spatial representation in the hippocampus.” In: *Science* 321.5885, pp. 140–143.
- Kopf, J., Agrawala, M., Barger, D., Salesin, D., and Cohen, M. (2010). “Automatic generation of destination maps.” In: *ACM Transactions on Graphics (TOG)* 29.6, p. 158.
- Laakso, S. A., Laakso, K. P., and Saura, A. J. (2000). “Improved scroll bars.” In: *CHI’00 Extended Abstracts on Human Factors in Computing Systems*. ACM, pp. 97–98.
- Laput, G., Xiao, R., Chen, X., Hudson, S. E., and Harrison, C. (2014). “Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors.” In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, pp. 389–394.

- Lee, Y. J., Zitnick, C. L., and Cohen, M. F. (2011). “ShadowDraw: real-time user guidance for freehand drawing.” In: *ACM Transactions on Graphics (TOG)*. Vol. 30. 4. ACM, p. 27.
- Lieberman, H. (1994). “Powers of ten thousand: navigating in large information spaces.” In: *Proceedings of the 7th annual ACM symposium on User interface software and technology*. ACM, pp. 15–16.
- Loomis, J. M., Golledge, R. G., and Klatzky, R. L. (1998). “Navigation system for the blind: Auditory display modes and guidance.” In: *Presence: Teleoperators and Virtual Environments 7.2*, pp. 193–203.
- Lynch, K. (1960). *The image of the city*. MIT press.
- Mackinlay, J., Good, L., Zellweger, P., Stefik, M., and Baudisch, P. (2003). “City Lights: Contextual views in minimal space.” In: *CHI’03 extended abstracts on Human factors in computing systems*. Citeseer, pp. 838–839.
- Macworld (2018). *How to use Maps on Apple Watch*. URL: <https://www.macworld.co.uk/how-to/apple/how-use-maps-on-apple-watch-3609161/> (visited on 01/02/2018).
- Miau, D. and Feiner, S. (2016). “Personalized Compass: A Demonstration of a Compact Visualization for Direction and Location.” In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, pp. 3731–3734.
- Microsoft (2017). *Bing Maps*. URL: <https://www.bing.com/maps/> (visited on 04/23/2017).
- Miller, G., Grisedale, S., and Anderson, K. T. (1999). “3Desque: interface elements for a 3D graphical user interface.” In: *The Journal of Visualization and Computer Animation* 10.2, pp. 109–119.
- Miyaki, T. and Rekimoto, J. (2009). “GraspZoom: zooming and scrolling control model for single-handed mobile interaction.” In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, p. 11.
- Munzner, T. (2014). *Visualization analysis and design*. CRC press.
- Ogata, M. and Imai, M. (2015). “SkinWatch: skin gesture interaction for smart watch.” In: *Proceedings of the 6th Augmented Human International Conference*. ACM, pp. 21–24.
- O’Keefe, J. and Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford: Clarendon Press.

- Olsen Jr., D. R. (1992). "The Interaction Technique Notebook: Bookmarks: An Enhanced Scroll Bar." In: *ACM Trans. Graph.* 11.3, pp. 291–295.
- Pangaro, G., Maynes-Aminzade, D., and Ishii, H. (2002). "The actuated workbench: computer-controlled actuation in tabletop tangible interfaces." In: *Proceedings of the 15th annual ACM symposium on User interface software and technology*. ACM, pp. 181–190.
- Patel, K., Chen, M. Y., Smith, I., and Landay, J. A. (2006). "Personalizing routes." In: *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, pp. 187–190.
- Perlin, K. and Fox, D. (1993). "Pad: an alternative approach to the computer interface." In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. ACM, pp. 57–64.
- Pfeuffer, K. and Gellersen, H. (2016). "Gaze and Touch Interaction on Tablets." In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, pp. 301–311.
- Pfeuffer, K., Alexander, J., Chong, M. K., and Gellersen, H. (2014). "Gaze-touch: combining gaze with multi-touch for interaction on the same surface." In: *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, pp. 509–518.
- Rekimoto, J. (2002). "SmartSkin: an infrastructure for freehand manipulation on interactive surfaces." In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pp. 113–120.
- Robbins, D. C., Cutrell, E., Sarin, R., and Horvitz, E. (2004). "ZoneZoom: map navigation for smartphones with recursive view segmentation." In: *Proceedings of the working conference on Advanced visual interfaces*. ACM, pp. 231–234.
- Rümelin, S., Rukzio, E., and Hardy, R. (2011). "NaviRadar: a novel tactile information display for pedestrian navigation." In: *Proceedings of the 24th annual ACM symposium on User interface software and technology*. ACM, pp. 293–302.
- Sarkar, M. and Brown, M. H. (1992). "Graphical fisheye views of graphs." In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pp. 83–91.
- Sarkar, M., Snibbe, S. S., Tversky, O. J., and Reiss, S. P. (1993). "Stretching the rubber sheet: a metaphor for viewing large layouts on small screens." In: *Proceedings of the 6th annual ACM symposium on User interface software and technology*. ACM, pp. 81–91.

- Shneiderman, B., Plaisant, C., Cohen, M. S., Jacobs, S., Elmqvist, N., and Diakopoulos, N. (2016). *Designing the user interface: strategies for effective human-computer interaction*. Pearson.
- Snaveley, N., Seitz, S. M., and Szeliski, R. (2006). "Photo tourism: exploring photo collections in 3D." In: *ACM transactions on graphics (TOG)*. Vol. 25. 3. ACM, pp. 835–846.
- Snaveley, N., Garg, R., Seitz, S. M., and Szeliski, R. (2008). "Finding paths through the world's photos." In: *ACM Transactions on Graphics (TOG)*. Vol. 27. 3. ACM, p. 15.
- Sorrows, M. E. and Hirtle, S. C. (1999). "The nature of landmarks for real and electronic spaces." In: *Spatial information theory. Cognitive and computational foundations of geographic information science*. Springer, pp. 37–50.
- Suomela, R. and Lehtikainen, J. (2000). "Context compass." In: *Wearable Computers, The Fourth International Symposium on*. IEEE, pp. 147–154.
- Tolman, E. C. (1948). "Cognitive maps in rats and men." In: *Psychological review* 55.4, p. 189.
- Tsukada, K. and Yasumura, M. (2004). "Activebelt: Belt-type wearable tactile display for directional navigation." In: *UbiComp 2004: Ubiquitous Computing*. Springer, pp. 384–399.
- Ullmer, B. and Ishii, H. (1997). "The metaDESK: models and prototypes for tangible user interfaces." In: *Proceedings of the 10th annual ACM symposium on User interface software and technology*. ACM, pp. 223–232.
- Vogel, D. and Baudisch, P. (2007). "Shift: a technique for operating pen-based interfaces using touch." In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, pp. 657–666.
- Wang, F., Li, Y., Sakamoto, D., and Igarashi, T. (2014). "Hierarchical route maps for efficient navigation." In: *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, pp. 169–178.
- Ware, C. (2012). *Information visualization: perception for design*. Elsevier.
- Ware, C. and Lewis, M. (1995). "The DragMag image magnifier." In: *Conference companion on Human factors in computing systems*. ACM, pp. 407–408.
- Wellner, P. (1991). "The DigitalDesk calculator: tangible manipulation on a desk top display." In: *Proceedings of the 4th annual ACM symposium on User interface software and technology*. ACM, pp. 27–33.

- Wenig, D., Schöning, J., Hecht, B., and Malaka, R. (2015). “StripeMaps: Improving Map-based Pedestrian Navigation for Smartwatches.” In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, pp. 52–62.
- Wenzel, E. M., Arruda, M., Kistler, D. J., and Wightman, F. L. (1993). “Localization using nonindividualized head-related transfer functions.” In: *The Journal of the Acoustical Society of America* 94.1, pp. 111–123.
- Xiao, R., Laput, G., and Harrison, C. (2014). “Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 193–196.
- Yanoff, M. and Duker, J. (2008). *Ophthalmology*. 3rd. Mosby.
- Zhang, Y., Zhou, J., Laput, G., and Harrison, C. (2016). “Skintrack: Using the body as an electrical waveguide for continuous finger tracking on the skin.” In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 1491–1503.
- Zhao, J., Wigdor, D., and Balakrishnan, R. (2013). “TrailMap: facilitating information seeking in a multi-scale digital map via implicit bookmarking.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, pp. 3009–3018.
- Zhou, C., Bhatnagar, N., Shekhar, S., and Terveen, L. (2007). “Mining personally important places from GPS tracks.” In: *Data Engineering Workshop, 2007 IEEE 23rd International Conference on*. IEEE, pp. 517–526.
- Zhou, H., Edrah, A., MacKay, B., and Reilly, D. (2017). “Block Party: Synchronized Planning and Navigation Views for Neighbourhood Expeditions.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, pp. 1702–1713.
- Zhou, J., Zhang, Y., Laput, G., and Harrison, C. (2016). “AuraSense: Enabling Expressive Around-Smartwatch Interactions with Electric Field Sensing.” In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, pp. 81–86.

## Appendix A

---

### *P-Compass Algorithms*

#### **A.1 POI Selection Algorithm**

Assume there is a sufficiently large number of a priori *points of interest* (POIs). In this case, it is necessary to apply a POI selection algorithm to determine a subset of POIs to form a P-Compass. Each POI is associated with a 2D position relative to a designated P-Compass *reference point*. The goal is to choose at most  $n$  off-screen POIs that form a roughly balanced P-Compass layout (i.e., POIs spread out in all directions around the P-Compass reference point). To accomplish this, we perform the following steps:

1. Pre-filter: Sort all off-screen POIs by their distances to the P-Compass reference point, and keep the  $k$  ( $k \geq n$ ) closest POIs. A larger  $k$  results in a larger pool of POIs for the later steps, and could increase the likelihood of achieving a balanced P-Compass layout. However, a larger pool of POIs could increase the amount of computation.
2. Sort POIs based on orientation: Each POI is associated with an orientation  $\theta$  ( $0^\circ \leq \theta < 360^\circ$ ) with respect to the P-Compass reference point. Arrange POIs in a loop such that the order of POIs conforms to the orientation order.
3. Compute inter-orientation differences between POIs: For each POI, compute one



inter-orientation difference (i.e., the orientation difference between a POI and its previous POI).

4. Backward select POIs: Select the  $n$  POIs with the largest inter-orientation differences.

We also include a post-filter stage to reduce clutter, when the number of post-selection POIs  $\geq 3$  (clutter is not an issue when there are only one or two POIs):

1. Compute inter-orientation differences between POIs.
2. Keep the POIs that have inter-orientation  $\geq threshold$  ( $40^\circ$  in our prototype).

## A.2 POI Clustering Algorithm

We now describe an algorithm to decide whether the selected POIs should be presented in a single-scale, or in mixed-scale (Section 4.2.2). Recall the goal of a P-Compass is to use as few off-screen POIs as possible to answer the question "Where is x?". In our field prototype, we select at most four POIs, and use the clustering algorithm to divide the selected POIs into at most two clusters.

Given a P-Compass reference point and a set of POIs, we first define a measurement we call *Distance-Ratio*, or *DR*:

$$DR = \frac{POI_{farthest}.distance}{POI_{closest}.distance},$$

where  $POI_{farthest}.distance$  is the distance of the farthest POI (measured from the P-Compass reference point), and  $POI_{closest}.distance$  is the distance of the closest POI. When

the set of POIs is divided into two clusters, two  $DR$ s,  $DR_1$  and  $DR_2$ , can be computed, one for each cluster.  $DR_{overall}$  for the entire set of POIs is defined as:

$$DR_{overall} = \frac{\max(DR_1, DR_2)}{\min(DR_1, DR_2)}.$$

When the entire set of POIs is in a single cluster,  $DR_{overall} = DR$ .

The POI clustering algorithm aims to cluster a given set of POIs, such that  $DR_{overall}$  can be minimized:

1. Compute  $DR_{overall}$  of a single cluster. Skip the clustering algorithm if  $DR_{overall} < threshold$ ; otherwise, perform the following steps:
  - a) Exhaustively compute the  $DR_{overall}$ s of all possible ways to cluster POIs (at most three additional computations in our case, on top of computing  $DR_{overall}$  for the single cluster).
  - b) Pick the configuration corresponding to the lowest  $DR_{overall}$ .
2. Present all POIs in a single scale if a single cluster results in the lowest  $DR_{overall}$ ; otherwise, present POIs in mixed-scale.

## Appendix B

---

### *User Study Questionnaires*

## **B.1 P-Compass Questionnaire**

# Smartphone/Smartwatch Navigation Aid Study

\* Required

1. **Participant ID (Ask the study coordinator) \***

.....

2. **Age \***

.....

3. **Gender \***

*Mark only one oval.*

☐ Female

☐ Male

4. **I use a computer... \***

*Mark only one oval.*

☐ Never

☐ Monthly

☐ Weekly

☐ Daily

☐ Multiple times per day

5. **I use a smartphone... \***

*Mark only one oval.*

☐ Never

☐ Monthly

☐ Weekly

☐ Daily

☐ Multiple times per day



6. **I use a smartwatch...** \*

Mark only one oval.

- ☐ Never
- ☐ Monthly
- ☐ Weekly
- ☐ Daily
- ☐ Multiple times per day

7. **I have experience using a compass and a scale on a map** \*

Mark only one oval.

- ☐ No
- ☐ Yes

8. **I use smartphone map applications (e.g., Google Maps)...** \*

Mark only one oval.

- ☐ Never
- ☐ Monthly
- ☐ Weekly
- ☐ Daily
- ☐ Multiple times per day

## Part1: Technique W

---

For the following questions, please choose a number from 1 through 7 to describe your experience with technique W.

For each question, we would appreciate any additional comments you have in the "Comments" section.

9. **Mental Demand** \*

How mentally demanding was the task?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. **Physical Demand** \*

How physically demanding was the task?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**11. Temporal Demand \***

How hurried or rushed was the pace of the task?  
*Mark only one oval per row.*

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**12. Performance \***

How successful were you in accomplishing what you were asked to do?  
*Mark only one oval per row.*

1 (Perfect)	2	3	4	5	6	7 (Failure)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**13. Effort \***

How hard did you have to work to accomplish your level of performance?  
*Mark only one oval per row.*

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**14. Frustration \***

How insecure, discouraged, irritated, stressed and annoyed were you?  
*Mark only one oval per row.*

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**15. Please provide any additional comments about or reactions to technique W. \***

.....

.....

.....

.....

.....

## Part 2: Technique C

---

For the following questions, please choose a number from 1 through 7 to describe your experience with technique C.

For each question, we would appreciate any additional comments you have in the "Comments" section.

16. **Mental Demand \***

How mentally demanding was the task?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

17. **Physical Demand \***

How physically demanding was the task?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

18. **Temporal Demand \***

How hurried or rushed was the pace of the task?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

19. **Performance \***

How successful were you in accomplishing what you were asked to do?

Mark only one oval per row.

1 (Perfect)	2	3	4	5	6	7 (Failure)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

20. **Effort \***

How hard did you have to work to accomplish your level of performance?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

21. **Frustration \***

How insecure, discouraged, irritated, stressed and annoyed were you?

Mark only one oval per row.

1 (Very Low)	2	3	4	5	6	7 (Very High)
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



22. Please provide any additional comments about or reactions to technique C. \*

.....

.....

.....

.....

.....

## Part 3: Overall Preferences

---

23. Specify your preferred technique for each task \*

*Mark only one oval per row.*

	W	C	No Preference
Locate	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Distance	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Orientation	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Triangulate	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Locate+	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

24. Please provide any additional comments below.

.....

.....

.....

.....

.....

## **B.2 SpaceTokens and SpaceBar Questionnaire**

# Smartphone Navigation Aid Study

\* Required

1. Participant ID (Ask the study coordinator) \*

---

2. Age \*

---

3. Gender \*

*Mark only one oval.*

☐ Female

☐ Male

4. I use a computer... \*

*Mark only one oval.*

☐ Never

☐ Monthly

☐ Weekly

☐ Daily

☐ Multiple times per day

5. I use a smartphone... \*

*Mark only one oval.*

☐ Never

☐ Monthly

☐ Weekly

☐ Daily

☐ Multiple times per day

6. I use smartphone map applications (e.g., Google Maps)... \*

*Mark only one oval.*

☐ Never

☐ Monthly

☐ Weekly

☐ Daily

☐ Multiple times per day



For the following questions, please choose a number from 1 through 7 to describe your experience with Google Maps.

For each question, we would appreciate any additional comments you have in the "Comments" section.

## Part 1: Google Maps

---

### 7. Mental Demand \*

How mentally demanding was the task?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

### 8. Physical Demand \*

How physically demanding was the task?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

### 9. Temporal Demand \*

How hurried or rushed was the pace of the task?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

### 10. Performance \*

How successful were you in accomplishing what you were asked to do?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Perfect	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Failure

### 11. Effort \*

How hard did you have to work to accomplish your level of performance?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High



**12. Frustration \***

How insecure, discouraged, irritated, stressed and annoyed were you?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

**13. Please provide any additional comments about or reactions to Google Maps. \***

---

---

---

---

---

## Part 2: SpaceToken

---

**14. Mental Demand \***

How mentally demanding was the task?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

**15. Physical Demand \***

How physically demanding was the task?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

**16. Temporal Demand \***

How hurried or rushed was the pace of the task?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

**17. Performance \***

How successful were you in accomplishing what you were asked to do?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Perfect	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Failure



18. **Effort** \*

How hard did you have to work to accomplish your level of performance?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

19. **Frustration** \*

How insecure, discouraged, irritated, stressed and annoyed were you?  
*Mark only one oval.*

	1	2	3	4	5	6	7	
Very Low	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very High

20. **Please provide any additional comments about or reactions to Space Tokens.** \*

---

---

---

---

---

## Overall Preferences

21. **Specify your preferred technique for each task** \*

*Mark only one oval per row.*

	Google Maps	Space Tokens	No Preference
Revisiting A, B, C...	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Where is X with respect to A, B, C?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Connecting the dots: investigating multiple routes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Creating and interacting with a collection	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reviewing a route	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

22. **Please provide any additional comments below.** \*

---

---

---

---

---

