Personalized Compass: A Compact Visualization for Direction and Location

Daniel Miau **Steven Feiner**
Columbia University
New York, NY, USA
{dmiau, feiner}@cs.columbia.edu

ABSTRACT

Maps on mobile/wearable devices often make it difficult to determine the location of a point of interest (POI). For example, a POI may exist outside the map or on a background with no meaningful cues. To address this issue, we present Personalized Compass, a self-contained compact graphical location indicator. Personalized Compass uses personal *a priori* POIs to establish a reference frame, within which a POI in question can then be localized. Graphically, a personalized compass combines a multi-needle compass with an abstract overview map. We analyze the characteristics of Personalized Compass and the existing Wedge technique, and report on a user study comparing them. Personalized Compass performs better for four inference tasks, while Wedge is better for a locating task. Based on our analysis and study results, we suggest the two techniques are complementary and offer design recommendations.

Author Keywords

Visualization; small screens; spatial cognition, maps.

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI)

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 could become difficult (e.g., route planning to an off-screen destination [18]). A small display also increases the probability of Desert Fog [24]—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. In this paper, we refer to them together as the "Where is *x*?" problem, occurring whenever some POI *x* is off display, or on display but with Desert Fog present.

The "Where is *x*?" problem is frequently encountered on a map, a common information space that is the focus of this

CHI'16, May 07–12, 2016, San Jose, CA, USA
© 2016 ACM. ISBN 978-1-4503-3362-7/16/05$15.00.
DOI: http://dx.doi.org/10.1145/2858036.2858068

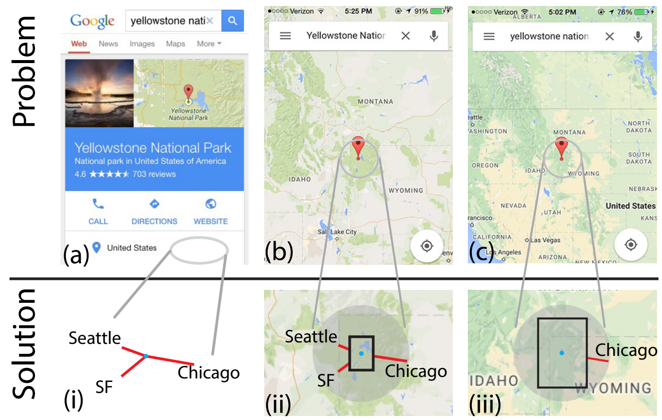


Figure 1: Where is Yellowstone National Park? Answering this question from a mobile search result often involves (a–b) unnecessary application switching and (b–c) repetitive zooming. (i–iii) A P-Compass uses important POIs to provide a first-order approximation to the answer, and can function independently or with a map. (a–c) Screenshots from a smartphone. (i–iii) Proposed P-Compass visualizations to be superimposed on the screenshots at the designated locations.

paper. 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 1(a) shows a 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 and then continue fighting Desert Fog by repeatedly zooming out (Figure 1b–c). When a satisfying overview is eventually reached (Figure 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, Internet articles, 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 1(i) shows P-Compass as a standalone location indicator. It 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 1(ii–iii). The black rectangle in the center of the compass indicates the boundary of the visible map. As a user zooms in 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 integrated P-Compass and Wedge [18], a well-known technique to visualize off-display POIs[1], into a customized iOS/OS-X map app and used the 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.

This paper thus makes the following contributions: 1) P-Compass, a compact graphical location indicator designed to address the "Where is x?" problem; 2) An analysis of P-Compass and Wedge; 3) A formal user study comparing these two approaches with tasks derived from our field experience, showing the advantages of P-Compass for inference tasks; and 4) Recommendations to designers, based on our analysis and study results.

[1]Wedge is also a frequent baseline technique in previous studies (e.g., [5, 7, 21]), which is why we chose it for comparison.

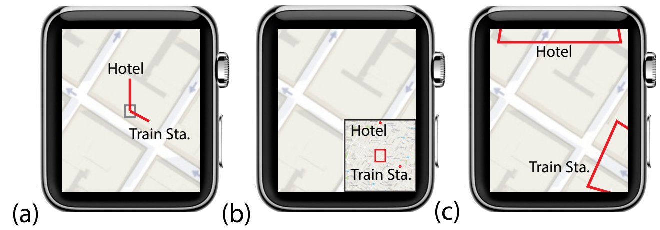


Figure 2: Where am I? Localization on a smartwatch form factor using (a) P-Compass, (b) Overview+Detail, and (c) Wedge. The two off-screen POIs are 6 cm ("Hotel") and 4 cm ("Train Sta.") from the centroid of the display. (In true scale.)

RELATED WORK

A large body of work has been dedicated to approaches for presenting overview and detail on a limited display area. Cockburn et al. [10] categorize these into four basic schemes:

Overview+Detail. This scheme, also referred to as *spatial separation* [10], presents an information space at different scales side-by-side (Figure 2b). Variants include stacking representations of an information space at different scales on top of each other [26], providing the user a magnified view [37, 25] with which to interactively explore an information space, and allowing the user to interactively build hierarchies of different regions of interest at different scales [23].

Zooming. Zooming [3, 20] separates information at different scales temporally, so it is also referred to as *temporal separation*. While zooming is well suited to interactively exploring an information space, it requires an unnecessary round trip between detail and overview for a user to gain context while focusing on detail. In Figure 1(c), for example, a reverse zoom is required if the users wants to confirm that they saw a lake in the park. Zooming can also be challenging when a user is mobile or their hands are busy.

Focus+Context. This scheme aims to blend the seams between overview and detail. Distortion [31, 4] and folding [12, 21] are typically applied to emphasize or deemphasize certain portions of an information space. Two representative frameworks [8, 14] have been proposed to include a wide range of approaches. However, it is often difficult to perform metric measurements from the modified information space.

Cue-Based. The cue-based scheme uses proxies to selectively visualize POIs that may not be visible in the displayed information space [1, 18, 21, 28, 19, 6]. A representative technique in this category is Wedge [18] (Figure 2c). 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).

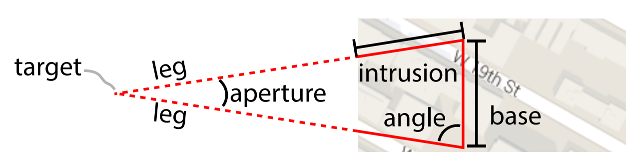


Figure 3: Anatomy of a Wedge (solid lines). The target can be located using visual shape completion (dashed lines).

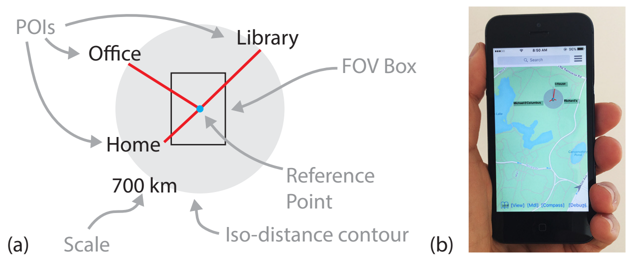


Figure 4: (a) Anatomy of a P-Compass. (b) P-Compass prototype.

P-Compass is a hybrid technique that integrates spatial separation and cue-based techniques, and can be incorporated into a zooming interface. Figure 2 compares P-Compass, Wedge and overview map. While P-Compass and overview map both add an overview to the underlying detail view (Figure 2a,b), the use of POI cues allows P-Compass to reduce the amount of occlusion. Furthermore, while P-Compass and Wedge both use proxies to indicate off-screen POIs (Figure 2a,c), the overview provided by P-Compass allows it to communicate the relationship between multiple off-screen POIs.

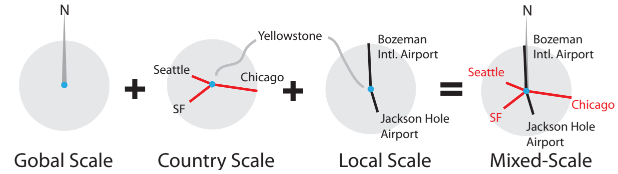
Several researchers have proposed variants of compasses that point to POIs instead of magnetic north [13, 11, 35, 9]. However, P-Compass is more than a specialized compass; P-Compass visualizes the direction and *location* of a POI in relation to personal a priori POIs (e.g., landmarks). Personal a priori POIs have been exploited in generating personalized routes [30], tourist maps [16], and textual descriptions of a POI [2]. In contrast, we propose a compact graphical location indicator that can be used with or without a map.

Much research aims to visualize a potentially large number of off-screen POIs [1, 18, 5, 7]. Instead, our focus is on using as few off-screen POIs as possible to answer the question "Where is *x*?" Furthermore, we are interested in a solution that can scale to distant off-screen POIs.

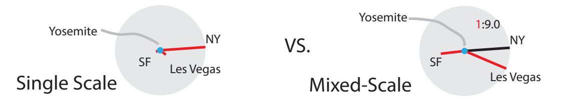
PERSONALIZED COMPASS

The anatomy of a P-Compass can be seen in Figure 4(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 signify the reference point, or stop at the edge of the FOV box, as shown in Figure 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 numerical scale denotes the distance to the iso-distance contour.

A 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 [36] (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 [15, 34], 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.



(a) Location of Yellowstone National Park and its nearest airports.



(b) Location of Yosemite National Park.

Figure 5: 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.

Proof-of-Concept Prototype

We implemented a custom iOS/OS-X map application incorporating a P-Compass (Figure 4b). 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. Other methods to collect a-priori POIs could include drawing them from personal GPS [38] and cellular network [22] 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 [32, 33]. These methods could be combined and continuously refined using machine learning.

We apply a simple greedy algorithm (detailed in the supplementary material [29]) 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 [18] in the same application. The user can toggle between P-Compass and Wedge, or show both simultaneously.

For over a year, the first 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.

Stackability

P-Compass allows multiple scales of information to be stacked on top of each other (Figure 5). Thus, a compass can be converted into a P-Compass *in place* simply by stacking relevant information on top. Users can access information at different scales simultaneously. Figure 5(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 5(b) displays all POIs at the same scale. Since Yosemite National Park is much closer to San Francisco and Les Vegas than to New York City, two of 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.

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-to-1 reference point.

Let VD be a virtual display formed by Figures 6–7. VD contains four Wedges and four P-Compasses, together visualizing four off-screen POIs (see Figure 7 for definitions of terms). It is trivial to use PC_{r_a} (Figure 6) 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 6) and W_{r2} (Figure 7), since a user cannot draw inferences from a POI without first estimating its absolute location.

Wedge is optimized for placing a POI near the boundary of a display. W_{r1} provides direct visual guidance near the edge of VD so a user can place $r1$ at its absolute location. Knowing the absolute locations of off-screen POIs can help perform tasks involving m (off-screen) POIs-to-n reference points. For example, determining which of the n on-screen hospitals is farthest away from the m off-screen traffic jams [18].

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

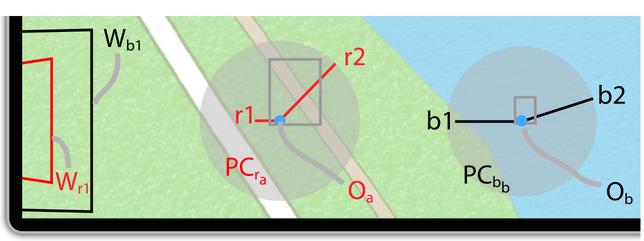


Figure 6: This figure and Figure 7 together form two corners of a virtual display nearly the size of this page. See Figure 7 caption for details.

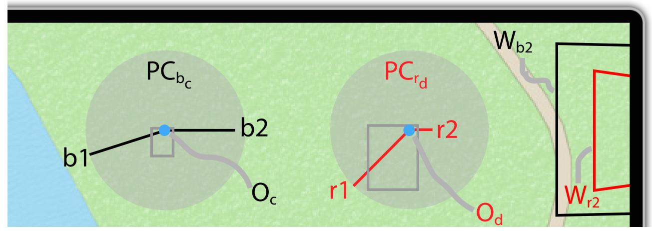


Figure 7: P-Compass vs. Wedge. Figure 6 and Figure 7 are the lower-left and upper-right corners of a virtual display (drawn in true scale). Wedges (prefixed with "W") and P-Compasses (prefixed with "PC") indicate locations of four off-display POIs, $r1$, $r2$, $b1$ and $b2$ (r* and b* are 4 cm and 50 cm off-display, respectively). P-Compasses are better for drawing inferences from POIs, while Wedges are better for locating POIs.

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}, shows 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 reference points on a display[2]. 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.

Issues of Scalability

Scalability refers to how the performance of a visualization technique is affected by the distance to an off-screen POI [17]. 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 [17]. 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, these two approaches both aim to *extend* the effective area of a display. Therefore, we refer to them as *display extension* methods.

The extended display area achieved this way is finite. Simply put, the probability that an off-screen POI falls into the small extended display area could be small. Most important, while

[2] A good example is the conventional magnetic compass: it is correct for all reference points on a local map regardless of its placement.

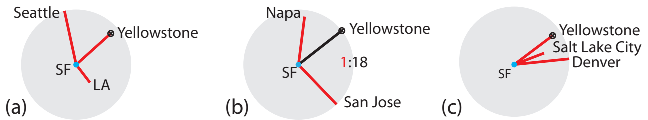


Figure 8: (a) Using supporting POIs to *extrapolate* an unknown POI (Yellowstone). (b–c) Two alternatives: (b) Emphasizes nearby POIs (mixed-scale representation); (c) Emphasizes POIs close to Yellowstone.

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.

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 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 8(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 8(b–c) shows two alternatives to Figure 8(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.

USER STUDY

Apparatus

We performed a study 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 × H 75 mm) and also emulated a circular smartwatch (36 mm diameter) for smartwatch tasks.

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 first 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.

Figure 9: (a) LOCATE and DISTANCE. (LOCATE was performed on a desktop monitor with no map background.) (b) ORIENTATION.

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. [18], 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 iPhone), along with a visualization to indicate the location of a subway station (Figure 9a). 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 [18, 5, 7] 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 visualization on an iPhone 5 (Figure 9a), and enters the answer on a nearby desktop computer. The goal is to investigate whether a visualization can support the use of a scale. In our field experience, we used other objects as the basis for measurement (e.g., a city block or a baseball field). For simplicity, we use a half-screen width for this task.

The two within-subject factors in this task are visualization type (P-Compass and Wedge) and off-screen POI distance (six off-screen distances: 7.5–32.5 cm). All off-screen POIs are on the short axis. Since the task may rapidly increase in difficulty with the distance to a POI, we asked participants to provide their best estimate.

ORIENTATION: Which direction is the subway station?

ORIENTATION asks a user to estimate the direction of an off-screen subway station with respect to the display center. The user sees the visualization on an iPhone 5 and uses their finger to rotate a blue line on the screen (Figure 9b) to indicate the answer. In our field experience, inquiring the direction of a POI was one of the most frequent questions in pedestrian and vehicle navigation. While map applications are typically able

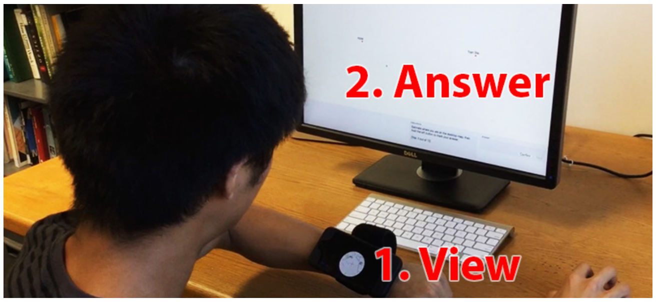


Figure 10: Setup for LOCALIZE and LOCATE+.

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 10).

In LOCALIZE, x is visible and located at the center of the smartwatch (P_1 and P_2 are both off-smartwatch; see Figure 11a). 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 11b). 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 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.

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× 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 impossible 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 **I**ntrusion, or **B**ase length (IB for short; see Figure 3 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. [18]. 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

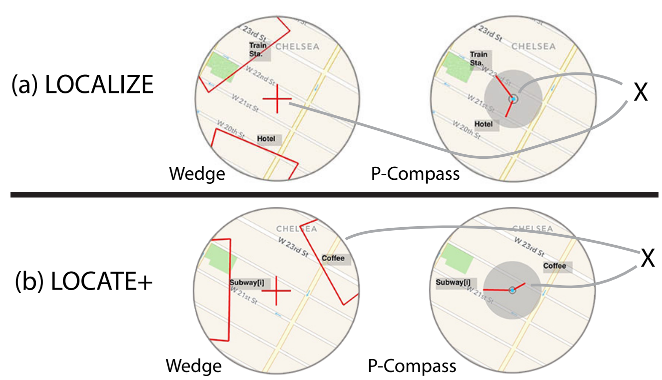


Figure 11: Smartwatch screenshots: (a) LOCALIZE; (b) LOCATE+.

respect to an on-screen reference point, users will still need to mentally combine the on-screen distance to 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. [18], so each Wedge uses its maximal IB.

Hypotheses

We formulated five hypotheses:

H.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[3] could put Wedge at a disadvantage, in comparison with P-Compass, which offers a linear distance cue[4].

H.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.

H.3, H.4. *For* LOCALIZE (H.3) *and* LOCATE+ (H.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.

H.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 [18] tries to avoid.

[3] Wedge angles scale nonlinearly as the distance to a POI changes.
[4] The ratio (size of FOV box)/(compass needle length) scales linearly as the distance to a POI changes.

	P-Compass	**Wedge**
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 1: Summary of error results (s.d.).

	P-Compass	**Wedge**
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 2: Summary of completion times (s.d.).

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. 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 background 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 (visualization) × (6 LOCATE + 16 ORIENTATION + 6 DISTANCE + 12 LOCALIZE + 12 LOCATE+) = 104 timed trials.

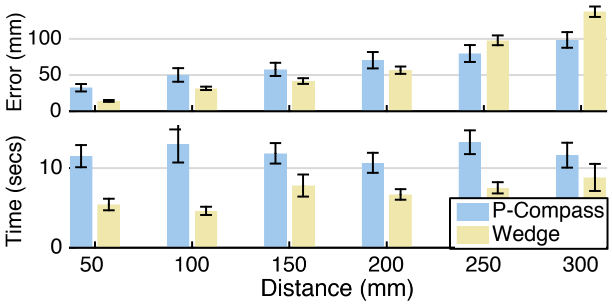


Figure 12: LOCATE Results. Errors (top), completion time (bottom).

RESULTS

Errors and task completion times were collected during the study (Tables 1–2). 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 conducted with $\alpha = 0.05$. Since each task was carried out independently, we discuss them separately.

LOCATE

Error for LOCATE (Figure 12 top) was defined as Euclidean distance between the estimated answer and ground truth. A 2×6 (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 **H.1**. A similar 2×6 (Visualization $\times$ Distance Level) ANOVA was performed on completion time (Figure 12 bottom), showing a significant main effect on visualization ($F_{1,25} = 37.22, p < 0.001$), but no interaction between visualization and distance levels ($F_{5,125} = 1.29, p = 0.27$).

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

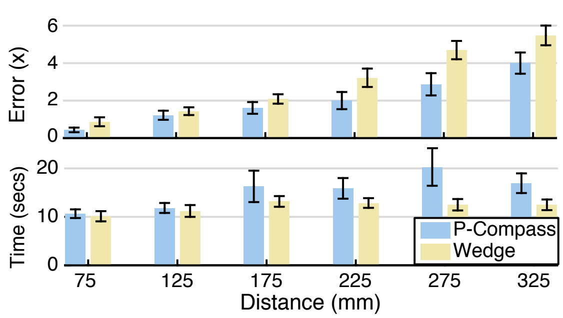


Figure 13: DISTANCE Results. Errors (top), completion time (bottom).

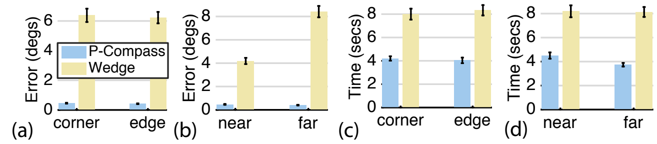


Figure 14: ORIENTATION Results.

Figure 13 (top) shows the errors of DISTANCE. The error is defined as the absolute difference between the estimated distance (in integer multiples of half-screen width) and the ground truth. A 2×6 (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 **H.2**. No interaction between visualization and distance level was found ($F_{5,110} = 1.89, p = 0.10$). A similar 2×6 (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 ($F_{5,110} = 1.34, p = 0.25$).

ORIENTATION

Errors for ORIENTATION (Figure 14a–b) were 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 14c–d), showing a significant 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 ($F_{1,725} = 1.21, p = 0.27$).

LOCALIZE

Figure 15 (top) shows errors 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 **H.3**. A similar one-way (Visualization) ANOVA was performed on completion

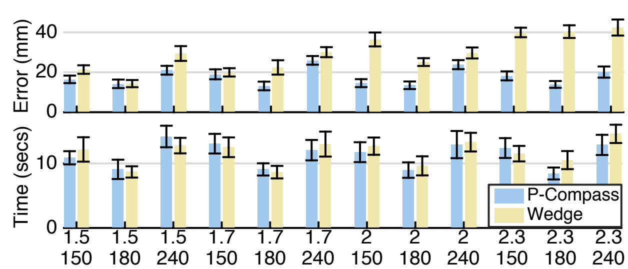


Figure 15: LOCALIZE Results. Errors (top), completion time (bottom). The x-axis indicates the parameters of the exploratory conditions (see text for details).

time, and found no significant main effect on visualization ($F_{1,25} = 0.097, p = 0.75$).

LOCATE+

Figure 16 (top) shows errors 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 **H.4**. A similar one-way (Visualization) ANOVA was performed on completion time, and found no significant main effect on visualization ($F_{1,25} = 1.87, p = 0.18$).

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 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 Distance ($p = 0.06$). These results partially support **H.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 9). 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

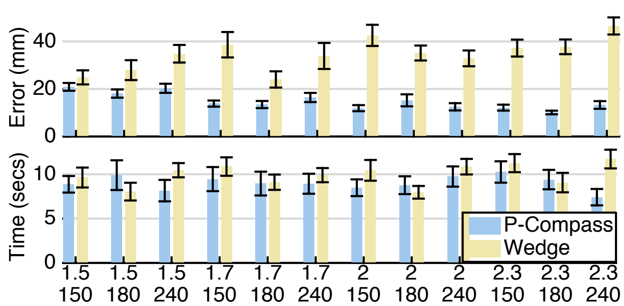


Figure 16: Locate+ Results. Errors (top), completion time (bottom). The x-axis indicates the parameters of the exploratory conditions (see text for details).

	P-Compass	Wedge	No Preference
Locate	10	12	4
Distance	16	7	3
Orientation	25	0	1
Localize	23	3	0
Locate+	19	5	2

Table 3: Participants' preferences on each task.

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.

DISCUSSION

The study results offer strong evidence to support **H.1–H.4**. **H.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, 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 each visualization technique make them suitable for different tasks. While users 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 disadvantage for Orientation, Localize, and Locate+. As reported in previous research [21] and 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 14a) 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, Figure 13 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 or similar to that of Wedge.

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 [17, 5] 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.

Transition between P-Compass and Wedge

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.

Limitations

P-Compass requires personal a priori knowledge of a space. The more detailed knowledge a user has, the better estimation a P-Compass can provide. 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.

CONCLUSIONS AND FUTURE WORK

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 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.

There are several directions for future work. 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. We would also like to distribute our P-Compass app and test it in the wild. P-Compass need not be restricted to the described setups and tasks. How well would P-Compass do for large displays, dynamic scenarios, or on-screen POIs? How could P-Compass be extended to 3D environments, for example, to support VR or AR applications?

We are also interested in evaluating the multi-scale feature. For independent information in mixed scales (Figure 5a), 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 5b), we could ask participants to localize a POI (LOCALIZE and LOCATE+), using information from two or more scales. Factors could be number of scales, ratio of scales, and distributions of POIs. The control condition could present all information at a single scale, and completion time, errors and preferences could be collected.

We believe P-Compass is an important step toward realizing a visualization technique we term *spacepiece*, which could answer "Where is x?" with a single glance, much how a *timepiece* answers "What time is it?" In addition to POIs, we are interested in incorporating entities such as paths, districts, and boundaries, which are important to wayfinding [27].

ACKNOWLEDGEMENTS

This research was funded in part by NSF Grants DGE-1144854, IIS-0905569, and IIS-1514429. We thank Mengu Sukan and Barbara Tversky for discussions.

REFERENCES

1. Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: a technique for visualizing off-screen objects. In *Proc. CHI*. 481–488.
2. Patrick M Baudisch and John C Krumm. 2013. Location-to-landmark. (Dec. 31 2013). US Patent 8,620,570.
3. Benjamin B Bederson, James D Hollan, Ken Perlin, Jonathan Meyer, David Bacon, and George Furnas. 1996. Pad++: A zoomable graphical sketchpad for exploring alternate interface physics. *Journal of Visual Languages & Computing* 7, 1 (1996), 3–32.
4. John Brosz, Miguel A Nacenta, Richard Pusch, Sheelagh Carpendale, and Christophe Hurter. 2013. Transmogrification: causal manipulation of visualizations. In *Proc. UIST*. 97–106.
5. Stefano Burigat and Luca Chittaro. 2011. Visualizing references to off-screen content on mobile devices: A comparison of Arrows, Wedge, and Overview+ Detail. *Interacting with Computers* 23, 2 (2011), 156–166.
6. Stefano Burigat, Luca Chittaro, and Silvia Gabrielli. 2006. Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In *Proc. MobileHCI*. 239–246.
7. Stefano Burigat, Luca Chittaro, and Andrea Vianello. 2012. Dynamic visualization of large numbers of off-screen objects on mobile devices: an experimental comparison of wedge and overview+ detail. In *Proc. MobileHCI*. 93–102.
8. M Sheelagh T Carpendale and Catherine Montagnese. 2001. A framework for unifying presentation space. In *Proc. UIST*. 61–70.
9. Luca Chittaro and Stefano Burigat. 2004. 3D location-pointing as a navigation aid in Virtual Environments. In *Proc. Working Conference on Advanced Visual Interfaces*. ACM, 267–274.
10. Andy Cockburn, Amy Karlson, and Benjamin B Bederson. 2008. A review of overview+ detail, zooming, and focus+ context interfaces. *ACM Computing Surveys (CSUR)* 41, 1 (2008), 2.
11. Mark Egenhofer. 1999. Spatial information appliances: A next generation of geographic information systems. In *1st Brazilian workshop on geoinformatics, Campinas, Brazil*.
12. Niklas Elmqvist, Nathalie Henry, Yann Riche, and Jean-Daniel Fekete. 2008. Melange: space folding for multi-focus interaction. In *Proc. CHI*. 1333–1342.
13. Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster. 1997. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. *Personal Technologies* 1, 4 (1997), 208–217.
14. George W Furnas. 2006. A fisheye follow-up: further reflections on focus+ context. In *Proc. CHI*. 999–1008.
15. Reginald G Golledge. 1999. Human wayfinding and cognitive maps. *Wayfinding Behavior: Cognitive Mapping and Other Spatial Processes* (1999), 5–45.
16. Floraine Grabler, Maneesh Agrawala, Robert W. Sumner, and Mark Pauly. 2008. Automatic Generation of Tourist Maps. *ACM Trans. Graph.* 27, 3, Article 100 (Aug. 2008), 11 pages.
17. Sean Gustafson. 2008. *Visualizing off-screen locations on small mobile displays*. Master's thesis. Department of Computer Science, University of Manitoba.
18. Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: clutter-free visualization of off-screen locations. In *Proc. CHI*. 787–796.
19. Sean Gustafson and Pourang P Irani. 2007. Comparing visualizations for tracking off-screen moving targets. In *CHI Extended Abstracts*. 2399–2404.
20. Takeo Igarashi and Ken Hinckley. 2000. Speed-dependent automatic zooming for browsing large documents. In *Proc. UIST*. 139–148.
21. Alexandra Ion, Y-L Betty Chang, Michael Haller, Mark Hancock, and Stacey D Scott. 2013. Canyon: providing location awareness of multiple moving objects in a detail view on large displays. In *Proc. CHI*. 3149–3158.
22. Sibren Isaacman, Richard Becker, Ramón Cáceres, Stephen Kobourov, Margaret Martonosi, James Rowland, and Alexander Varshavsky. 2011. Identifying important places in peoples lives from cellular network data. In *Pervasive Computing*. Springer, 133–151.
23. Waqas Javed, Sohaib Ghani, and Niklas Elmqvist. 2012. Polyzoom: multiscale and multifocus exploration in 2D visual spaces. In *Proc. CHI*. 287–296.
24. Susanne Jul and George W Furnas. 1998. Critical zones in desert fog: aids to multiscale navigation. In *Proc. UIST*. 97–106.
25. Dominik P Käser, Maneesh Agrawala, and Mark Pauly. 2011. FingerGlass: efficient multiscale interaction on multitouch screens. In *Proc. CHI*. 1601–1610.
26. Henry Lieberman. 1994. Powers of ten thousand: navigating in large information spaces. In *Proc. UIST*. 15–16.
27. Kevin Lynch. 1960. *The image of the city*. MIT Press.
28. J Mackinlay, Lance Good, P Zellweger, Mark Stefik, and Patrick Baudisch. 2003. City Lights: Contextual views in minimal space. In *Proc. CHI*. 838–839.
29. Daniel Miau and Steven Feiner. 2016. Personalized Compass Supplementary Material. (2016). `http://dx.doi.org/10.1145/2858036.2858068`.
30. Kayur Patel, Mike Y Chen, Ian Smith, and James A Landay. 2006. Personalizing routes. In *Proc. UIST*. 187–190.

31. Manojit Sarkar and Marc H Brown. 1992. Graphical fisheye views of graphs. In *Proc. CHI*. 83–91.
32. Noah Snavely, Rahul Garg, Steven M Seitz, and Richard Szeliski. 2008. Finding paths through the world's photos. In *ACM Trans. Graph.*, Vol. 27.
33. Noah Snavely, Steven M Seitz, and Richard Szeliski. 2006. Photo tourism: exploring photo collections in 3D. In *ACM Trans. Graph.*, Vol. 25. 835–846.
34. Molly E Sorrows and Stephen C Hirtle. 1999. The nature of landmarks for real and electronic spaces. In *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*. Springer, 37–50.
35. Riku Suomela and Juha Lehikoinen. 2000. Context compass. In *The Fourth International Symposium on Wearable Computers*. IEEE, 147–154.
36. Edward C Tolman. 1948. Cognitive maps in rats and men. *Psychological Review* 55, 4 (1948), 189.
37. Colin Ware and Marlon Lewis. 1995. The DragMag image magnifier. In *Conference Companion on Human Factors in Computing Systems*. ACM, 407–408.
38. Changqing Zhou, Nupur Bhatnagar, Shashi Shekhar, and Loren Terveen. 2007. Mining personally important places from GPS tracks. In *IEEE 23rd International Conference on Data Engineering Workshop*. 517–526.