

SpaceTokens: Interactive Map Widgets for Location-centric Interactions

Daniel Miao Steven Feiner
Dept. of Computer Science, Columbia University
New York, NY 10027, USA
{dmiao, feiner}@cs.columbia.edu

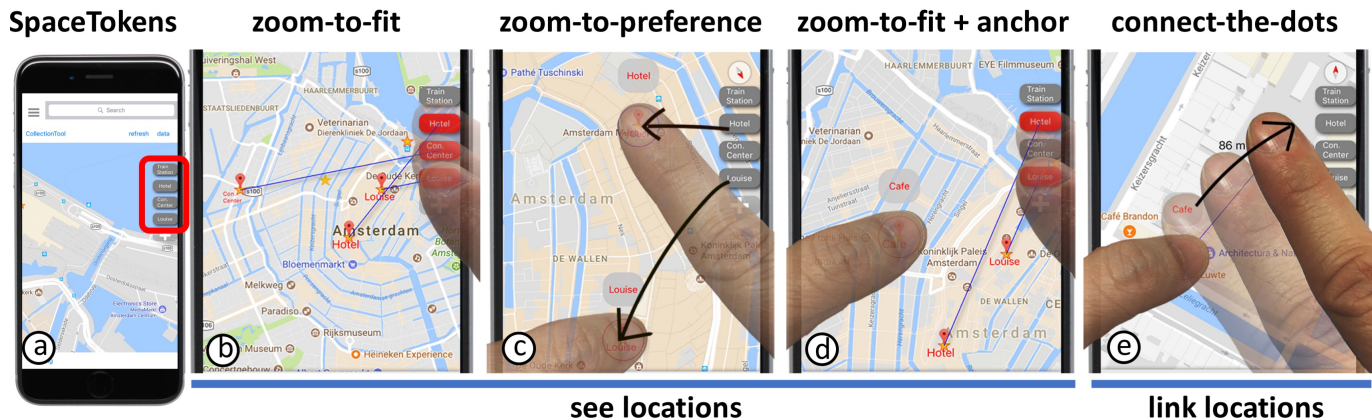


Figure 1. SpaceTokens make locations first-class citizens of map interaction. (a) Four SpaceTokens (gray rectangles with rounded corners), each representing a location. SpaceTokens empower users to use locations directly to address “seeing” and “linking” problems discussed in paper. For example, users 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.

ABSTRACT

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. 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. We propose SpaceTokens, interactive widgets that act as clones of locations, and which users can create and place on map edges like virtual whiteboard magnets. SpaceTokens make location a first-class citizen of map interaction. They empower users to rapidly perform location-centric tasks directly using locations: users can select combinations of on-screen locations and SpaceTokens to control the map window, or connect them to create routes. Participants in a study overwhelmingly preferred a SpaceTokens prototype over Google Maps on identical smartphones for the majority of tasks.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

© 2018 ACM. ISBN 978-1-4503-5620-6/18/04...\$15.00

DOI: <https://doi.org/10.1145/3173574.3173822>

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation: User Interfaces: Graphical User Interfaces (GUI)

Author Keywords

maps; location-centric interaction; bookmarks; navigation; revisitation

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 [34]. 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 paper, 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 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 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 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 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 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 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 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 2a), and **SpaceBar**, which allows users to interact with a route efficiently (Figure 2b).

To explore the utility of SpaceTokens and their interaction techniques/tools, we developed a custom iOS map application that uses them, and refined it during regular use by the first author. 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 [18] 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.

Contributions

We make four main contributions:

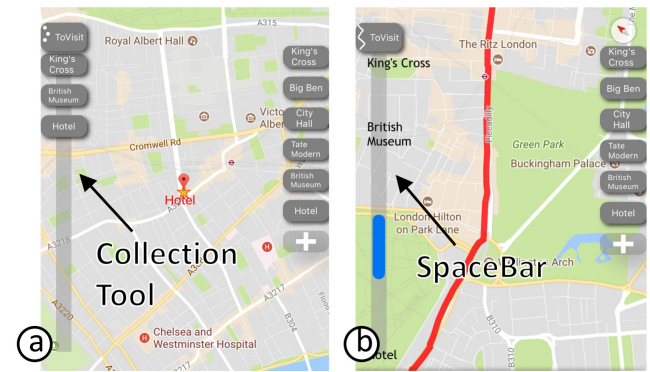


Figure 2. Additional tools. (a) CollectionTool is a structure to organize locations. (b) SpaceBar is a scrollbar-like tool for users to interact with a route. CollectionTool transforms to SpaceBar when creating a route connecting all locations in CollectionTool.

- SpaceTokens, interactive widgets that are clones of locations and facilitate users performing location-centric tasks.
- A set of related interaction techniques that help users revisit important saved locations.
- A set of tools that help users establish links between locations and interact with routes.
- A user study that shows participants overwhelmingly preferred a prototype smartphone application incorporating SpaceTokens over Google Maps for three out of five common map tasks.

RELATED WORK

Our work builds on four key research areas: visualization techniques for off-screen contents and focus+context, shortcuts for revisitation, constraints for information selection, and tangible information interactions.

Visualization techniques for off-screen contents and focus+context. A body of research has investigated visualizing off-screen contents [7, 20, 24, 26, 35]. For example, Halo [7] and Wedge [20] use partially occluded circles and triangles, respectively, to represent off-screen objects. By visually completing these partial shapes, users estimate the locations of the off-screen objects. Focus+context approaches (e.g., [40, 15]) present focused areas in detail while preserving their surrounding context, often by distorting the information space. Our work differs from these two research areas by providing users with an undistorted view of certain important locations, so they can examine the areas around and between those locations.

Shortcuts for revisitation. Researchers have long noticed that computer users tend to revisit previously viewed information [13, 1, 3, 21]. In document browsers (including web browsers), bookmarks are commonplace navigational widgets that address 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 [47]. Some map applications and prototypes provide features similar to zoom-to-fit in SpaceTokens. However, users need to

assemble a list of locations/areas, apply the “show all” (Bing Maps [36]) or “union search results” (TrailMap) feature to see all locations at once, and in some cases traverse layers of menus in the process (Bing Maps). Users must then repeat similar steps to view a different set of locations.

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 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 merely being selected for 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 navigational instrument (SpaceBar) to facilitate efficient interaction with a route.

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 [29, 9, 11], or to use the retrieved images for assisting users in creating artifacts such as drawings [33] or photo montages [14]. There are also interactive techniques to select data from a data set [23, 12]. 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 [16]. Maps, however, have not received much attention in this research area. We explore the idea of applying the constraint technique to map revisitation.

Tangible information interactions. Tangible User Interfaces (TUIs) [28] use physical props as tangible controls to manipulate digital information. While our SpaceTokens prototype is not a TUI, it is important to discuss relevant TUI research, which has addressed revisitation for maps. For example, MetaDesk [42] 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). While MetaDesk implements position constraints, it does not support visibility constraints, the combination of visibility and position constraints, 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 win-

dow 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 [38] 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.

TUI research has also explored embodying information (e.g., videos and whiteboard drawings) as physical objects (MediaBlocks [43]) and allowing users to transport and combine that information using the objects. While we explore using a SpaceToken directly as a location building block for a route, or an item in a todo list (the connect-the-dots interaction, and the CollectionTool described in the introduction, respectively), a major distinction between SpaceTokens and TUIs is their foci. MetaDesk and MediaBlocks aim to make information tangible. In contrast, SpaceTokens focus on making location a first-class citizen of map interaction and empowering users to perform location-centric tasks effectively.

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, 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 [19] or Apple’s Find Friends feature [5]), 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

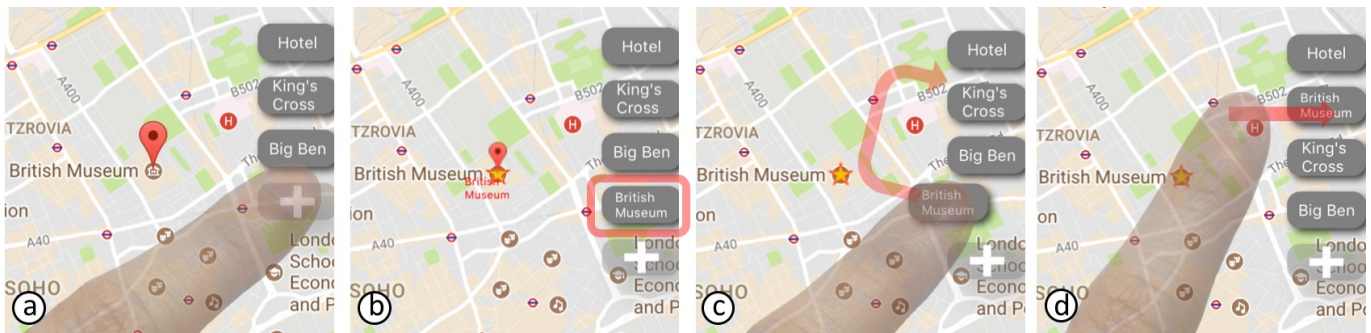


Figure 3. Creating SpaceToken to represent location. (a) After highlighting location, user taps “+” button to create SpaceToken for location. (b) New SpaceToken is added. (c) User can reorder SpaceTokens by long-pressing one and moving it to desired location. (d) SpaceToken can be removed by swiping it off display edge.

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 [48] showed how to support 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, SpaceTokens address not only this problem, but seeing and linking in general.

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 3a). A newly created SpaceToken is docked at the edge of the map (Figure 3b). Visually, SpaceTokens are like magnets on a whiteboard. Users can also arrange SpaceTokens in any order they prefer (Figure 3c). To remove a SpaceToken, a user can swipe the SpaceToken out of the display boundary (Figure 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 display, SpaceTokens can be organized in a palette, as we have done in an iPad implementation (not shown here).

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

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 position 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 those 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 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 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 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 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 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 [42].

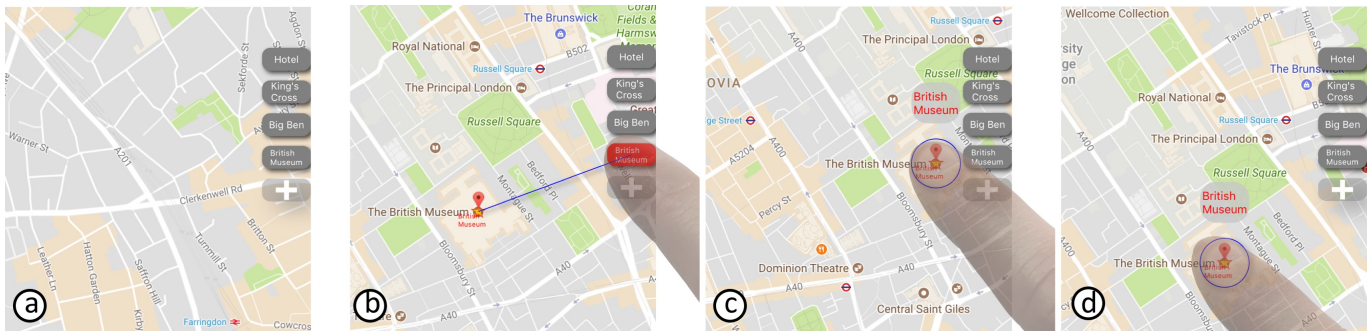


Figure 4. Using SpaceTokens to specify visibility and position constraints. (a) Where is British Museum? (b) Tap British Museum SpaceToken to see museum (specifying visibility constraint). SpaceToken turns red, screen zooms and pans as appropriate, and blue leader line indicates corresponding location. (c–d) Alternatively, drag British Museum SpaceToken to specify where to display museum (specifying position constraint). LinkTool appears above SpaceToken when dragged. Multiple SpaceTokens can be used to mix and match visibility/position constraints of different locations.

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 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 is created for x , and the visible portion of the map is adjusted to fit both x and the locations corresponding to the selected SpaceTokens. This is essentially zoom-to-fit with anchor (Figure 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.

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 5a, in which the hotel is selected), and then continues to touch the selected location (Figure 5b). The LinkTool, a bubble labeled with the location name, appears above the selected location (Figure 5b). Alternatively, the LinkTool is also available when the user drags a SpaceToken from its docking location (shown earlier in Figure 4c–d). While holding the location with one finger, the user can use another finger to drag out the LinkTool (Figure 5c). As it is dragged out, it acts like a tape measure that can be used to measure distance (Figure 5c, where the measured distance appears near the tip of the right index finger).

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 link¹. 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 5d). Once a route is created, the user can create a SpaceToken to represent the route (in the same fashion as creating a SpaceToken for a location, as shown in Figure 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(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 wish 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 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 6b). Users can save the composite SpaceToken and load it back into the CollectionTool. Tapping the composite SpaceToken adjusts the visible portion of the map to include all locations associated with the composite SpaceToken (Figure 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.

¹Target-selection aids such as SHIFT [44] could be implemented, and 3D touch sensing could be replaced with a long press.

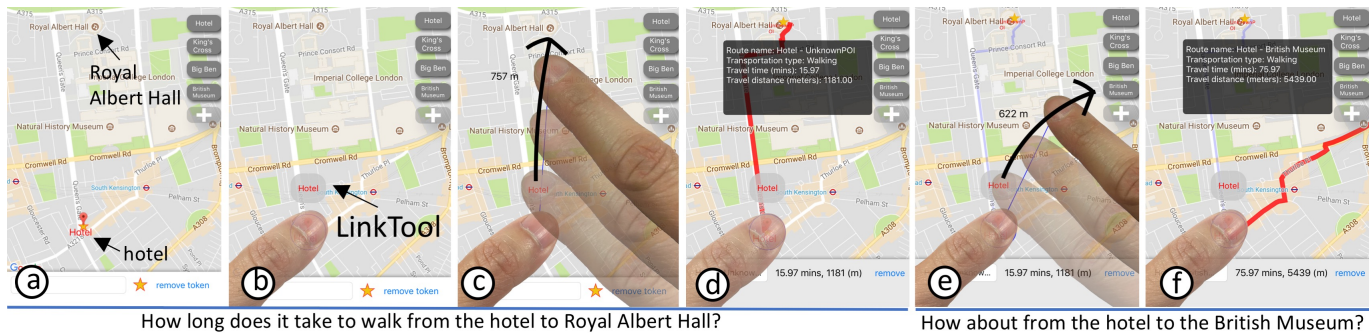


Figure 5. Connect the dots using LinkTool. (a) How long does it take to walk from my hotel to Royal Albert Hall? (b) Tap highlighted location to enable LinkTool. (c) Drag LinkTool to Royal Albert Hall, and press with force to confirm connection between both locations. (d) Route and information view appear. (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.)

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 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 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 7c).

A user can tap a point on the SpaceBar to jump to the corresponding portion of the route (Figure 7d). A user can also *scrub* the SpaceBar to traverse the route (Figure 7e). This interaction can be considered as an extension of content-aware scrolling [27], 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 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 [46, 8, 2, 41, 10], or to show a route at multiple scales simultaneously [30, 45, 39]. We take a different ap-

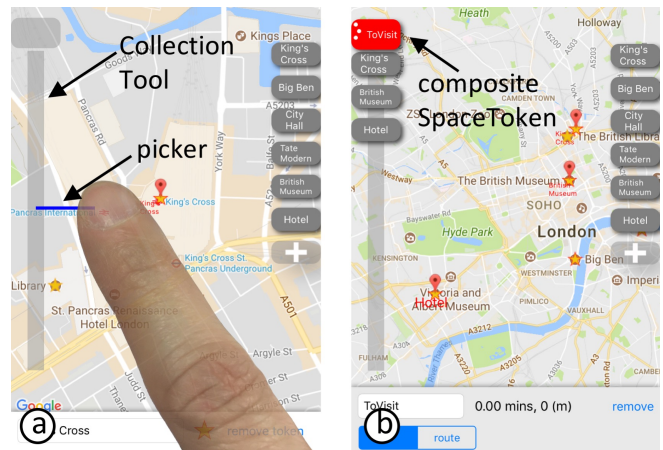


Figure 6. CollectionTool. (a) Using one finger to drag out of CollectionTool enables picker (blue line), which allows users to pick any on-screen location or SpaceToken. (b) Composite SpaceToken at top of CollectionTool represents CollectionTool content (three locations here). Tapping composite SpaceToken shows all three locations on display.

proach 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 [22, 37, 32] and scrollbar navigation [25, 31, 4]) can all be directly transferred to SpaceBar.

PROTOTYPE IMPLEMENTATION

We implemented SpaceTokens and their interaction techniques and tools in a standalone custom iOS map application using the Google Maps SDK [17]. 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 first author used and refined the prototype (mostly the more portable iPhone version) for around six months, including three international trips. The ideas of “Anchor” and “Link-

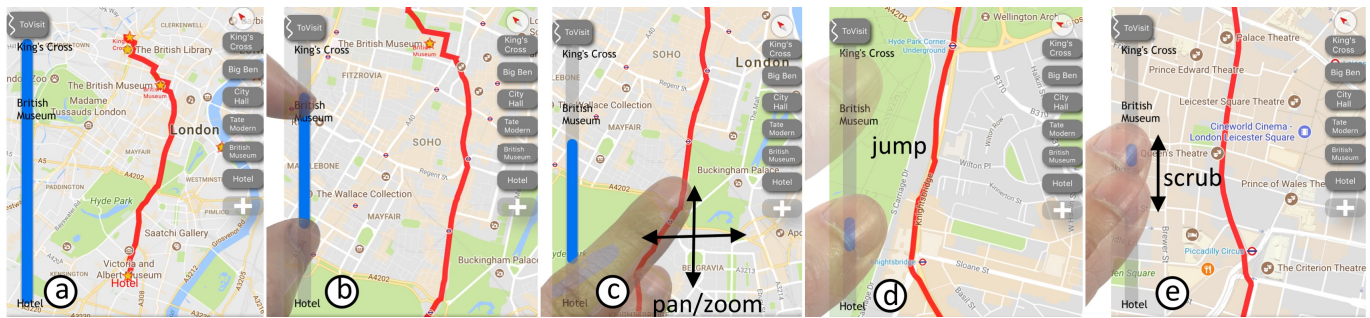


Figure 7. (a) SpaceBar is a scrollbar for a route. (b) User can change size and position of elevator indicator to make corresponding portion of route visible. (c) Conversely, elevator indicator is also updated as user interacts with route. (d) User can jump to any portion of route quickly by tapping SpaceBar, or (e) traverse route by scrubbing SpaceBar.

Tool” were conceived during this period, to support the frequent need to hold an onscreen location, or to assist the author in rapidly linking locations to investigate travel path, distance, time. We also made the list of SpaceTokens scrollable to support a large number of SpaceTokens.

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.

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 the motivation section on seeing: 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 Link-Tool. 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.

Hypotheses

We formulated the following five hypotheses.

H1, H2. *For the “SeeABC” (H1) and “SeeX+ABC” (H2) 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.

H3. 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.

H4. 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. 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.

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 received \$15 for their participation. 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.

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 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 divided 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

| | 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 1. Participants’ preferences for each task.

daily life.) The next 30 minutes formed the core of the study (the five tasks, with *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, 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 \$15.

RESULTS

Preferences

Table 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 **H1**, **H2**, and **H4**. 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 **H3** and **H5**. Observations of participants and their feedback reveal insights underlying their preferences.

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: “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.

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.

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) [6]. 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).

NASA TLX Results

Figure 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

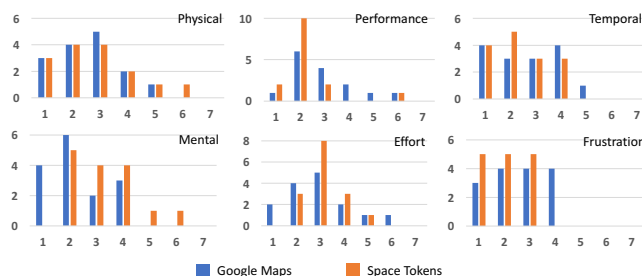


Figure 8. TLX Results. Lower numbers are preferred for all measures.

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 ($\chi^2_{(1)} = 2.27, p = 0.13$).

Discussion

Learning curve. Compared with Google Maps, a commercial map application familiar to all participants, the SpaceToken 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 problem, 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 TLX results were collected for 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.

LIMITATIONS AND FUTURE WORK

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., on a trip, executing planned activities); in other situations, he would most likely use the existing search and autocomplete feature to revisit a

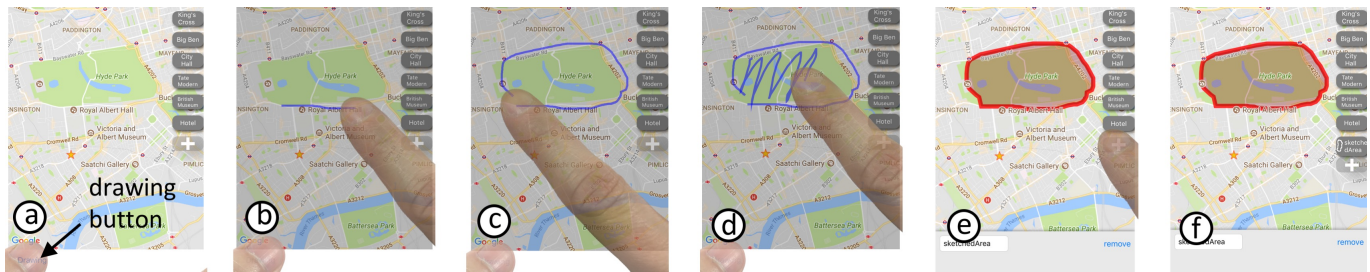


Figure 9. Drawing interface to mark custom spatial entities. (a) Holding drawing button at bottom left corner of display, (b) activates drawing mode, dimming screen, allowing user to sketch line to specify path or (c–d) sketch (nearly) closed outline and scribble inside it to specify area. (e) Tapping “+” button creates SpaceToken for the marked area. (f) An area SpaceToken is added above the “+” button.

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 [47]—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.

SpaceTokens should be able to represent more general spatial entities. SpaceTokens currently represent one or more locations or a route between locations created with the LinkTool. We are currently experimenting with SpaceTokens that represent arbitrary paths and areas, and have prototyped a simple drawing interface with which a user sketches a path or area that can be turned into a SpaceToken (Figure 9). We extended our prototype to support using these more general spatial entities as position or visibility constraints to address the seeing problem. At the conclusion of the formal study, 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 cur-

rently 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.

Future studies. It would be interesting to perform TLX per task per technique, rather than per technique, as we did. Quantitative measurements of how users interact with our prototype versus Google Maps could be collected and analyzed (e.g., type and number of interactions, time, and accuracy). Finally, we would like to distribute our prototype for a larger scale in-the-wild study.

CONCLUSIONS

We have asked the question, “Why can’t map users perform location-centric tasks *directly* using locations?” 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 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 navigational 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.

ACKNOWLEDGEMENTS

This research was funded in part by NSF Grants DGE-1144854 and IIS-1514429. We thank Brian A. Smith for discussions.

REFERENCES

1. Eytan Adar, Jaime Teevan, and Susan T Dumais. 2008. Large scale analysis of web revisitation patterns. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 1197–1206.
2. Maneesh Agrawala and Chris Stolte. 2001. Rendering effective route maps: improving usability through

- generalization. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 241–249.
3. Jason Alexander, Andy Cockburn, Stephen Fitchett, Carl Gutwin, and Saul Greenberg. 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, 1665–1674.
 4. Caroline Appert and Jean-Daniel Fekete. 2006. OrthoZoom scroller: 1D multi-scale navigation. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 21–30.
 5. Apple. 2017. Apple Maps location sharing. (2017). <https://support.apple.com/en-us/HT201493>
 6. Anthony J Aretz. 1991. The design of electronic map displays. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 33, 1 (1991), 85–101.
 7. Patrick Baudisch and Ruth Rosenholtz. 2003. Halo: a technique for visualizing off-screen objects. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 481–488.
 8. John Brosz, Miguel A Nacenta, Richard Pusch, Sheelagh Carpendale, and Christophe Hurter. 2013. Transmogrification: causal manipulation of visualizations. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 97–106.
 9. Yang Cao, Hai Wang, Changhu Wang, Zhiwei Li, Liqing Zhang, and Lei Zhang. 2010. Mindfinder: interactive sketch-based image search on millions of images. In *Proceedings of the 18th ACM international conference on Multimedia*. ACM, 1605–1608.
 10. M Sheelagh T Carpendale and Catherine Montagnese. 2001. A framework for unifying presentation space. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*. ACM, 61–70.
 11. Tao Chen, Ming-Ming Cheng, Ping Tan, Ariel Shamir, and Shi-Min Hu. 2009. Sketch2photo: Internet image montage. *ACM Transactions on Graphics (TOG)* 28, 5 (2009), 124.
 12. Mei C Chuah, Steven F Roth, Joe Mattis, and John Kolojejchick. 1995. SDM: Selective dynamic manipulation of visualizations. In *Proceedings of the 8th annual ACM symposium on User interface and software technology*. ACM, 61–70.
 13. Andy Cockburn and Bruce McKenzie. 2001. What do web users do? An empirical analysis of web use. *International Journal of human-computer studies* 54, 6 (2001), 903–922.
 14. Mathias Eitz, Ronald Richter, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. 2011. Photosketcher: interactive sketch-based image synthesis. *IEEE Computer Graphics and Applications* 31, 6 (2011), 56–66.
 15. Niklas Elmqvist, Nathalie Henry, Yann Riche, and Jean-Daniel Fekete. 2008. Melange: space folding for multi-focus interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1333–1342.
 16. George W Furnas and Benjamin B Bederson. 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., 234–241.
 17. Google. 2017a. Google Maps iOS SDK. (2017). <https://developers.google.com/maps>
 18. Google. 2017b. Google Maps mobile. (2017). <https://www.google.com/maps/about/>
 19. Google. 2017c. Google Maps Real-Time Location Sharing. (2017). <https://blog.google/products/maps/share-your-trips-and-real-time-location-google-maps/>
 20. Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 787–796.
 21. Jeffrey Heer, Jock D Mackinlay, Chris Stolte, and Maneesh Agrawala. 2008. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *Visualization and Computer Graphics, IEEE Transactions on* 14, 6 (2008), 1189–1196.
 22. William C Hill, James D Hollan, Dave Wroblewski, and Tim McCandless. 1992. Edit wear and read wear. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 3–9.
 23. Christian Holz and Steven Feiner. 2009. Relaxed selection techniques for querying time-series graphs. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 213–222.
 24. Zahid Hossain, Khalad Hasan, Hai-Ning Liang, and Pourang Irani. 2012. EdgeSplit: facilitating the selection of off-screen objects. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM, 79–82.
 25. Takeo Igarashi and Ken Hinckley. 2000. Speed-dependent automatic zooming for browsing large documents. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. ACM, 139–148.
 26. 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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3149–3158.
 27. Edward W Ishak and Steven K Feiner. 2006. Content-aware scrolling. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, 155–158.

28. Hiroshi Ishii and Brygg Ullmer. 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, 234–241.
29. Charles E Jacobs, Adam Finkelstein, and David H Salesin. 1995. Fast multiresolution image querying. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 277–286.
30. Waqas Javed, Sohaib Ghani, and Niklas Elmqvist. 2012. Polyzoom: multiscale and multifocus exploration in 2d visual spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 287–296.
31. Juho Kim, Amy X Zhang, Jihee Kim, Robert C Miller, and Krzysztof Z Gajos. 2014. Content-aware kinetic scrolling for supporting web page navigation. In *Proc. of UIST*.
32. Sari A Laakso, Karri Pekka Laakso, and Asko J Saura. 2000. Improved scroll bars. In *CHI'00 Extended Abstracts on Human Factors in Computing Systems*. ACM, 97–98.
33. Yong Jae Lee, C Lawrence Zitnick, and Michael F Cohen. 2011. ShadowDraw: real-time user guidance for freehand drawing. In *ACM Transactions on Graphics (TOG)*, Vol. 30. ACM, 27.
34. Kevin Lynch. 1960. *The image of the city*. MIT press.
35. Daniel Miao and Steven Feiner. 2016. Personalized Compass: A Compact Visualization for Direction and Location. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 5114–5125.
36. Microsoft. 2017. Bing Maps. (2017). <https://www.bing.com/maps/>
37. Dan R. Olsen, Jr. 1992. The Interaction Technique Notebook: Bookmarks: An Enhanced Scroll Bar. *ACM Trans. Graph.* 11, 3 (July 1992), 291–295.
38. Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. 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, 181–190.
39. Daniel C Robbins, Edward Cutrell, Raman Sarin, and Eric Horvitz. 2004. ZoneZoom: map navigation for smartphones with recursive view segmentation. In *Proceedings of the working conference on Advanced visual interfaces*. ACM, 231–234.
40. Manojit Sarkar and Marc H Brown. 1992. Graphical fisheye views of graphs. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 83–91.
41. Manojit Sarkar, Scott S Snibbe, Oren J Tversky, and Steven P Reiss. 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, 81–91.
42. Brygg Ullmer and Hiroshi Ishii. 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, 223–232.
43. Brygg Ullmer, Hiroshi Ishii, and Dylan Glas. 1998. mediaBlocks: physical containers, transports, and controls for online media. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM, 379–386.
44. Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 657–666.
45. Fangzhou Wang, Yang Li, Daisuke Sakamoto, and Takeo Igarashi. 2014. Hierarchical route maps for efficient navigation. In *Proceedings of the 19th international conference on Intelligent User Interfaces*. ACM, 169–178.
46. Dirk Wenig, Johannes Schöning, Brent Hecht, and Rainer Malaka. 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, 52–62.
47. Jian Zhao, Daniel Wigdor, and Ravin Balakrishnan. 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, 3009–3018.
48. Huiyuan Zhou, Aisha Edrah, Bonnie MacKay, and Derek Reilly. 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, 1702–1713.