Features to Support Map-Based Mobile Collaboration

Technical Report HMT-11-02

Eli Hooten

Abstract

The intent of this project is to develop methods that will eventually be used to support mobile collaboration. This project uses the Google Maps API along with additional JavaScript code to produce a map-based interface that supports drawing with either a mouse or a user's finger. Map Features, a means of organizing information on a map by its spatial locality, have also been developed. These two concepts, drawing and Map Features, will ultimately be used alongside other features to support mobile collaboration.

I. Introduction

Collaboration is defined as a specific interaction between two or more people for accomplishing a particular task [1], and typically requires activity coordination and cooperation among collaborators [2]. Mobile collaboration consists of a team of individuals completing related tasks within an environment. In mobile collaboration is facilitated through the use of mobile devices that allow team members to communicate in order to assign tasks, share information, and support group decision-making.

Previous research has determined that mobile collaboration is necessary and relevant to a multitude of domains [3, 4]. Mobile technologies (e.g., smartphones, tablet PCs, netbooks) are continuously offering more functionality, computational power, and seeing wider use [5]. Recent trends in smartphone sales indicate that by the end of 2012, global unit shipments of smartphones and tablets will surpass that of desktop and notebook PCs [6]. Lyytinen and Yoo argue that there is an increasing dependency on organizational computing, leading to a need for mobile collaboration scenarios that require the ability to perform computing anywhere and at anytime [7]. Therefore, it can be argued that the need for mobile collaboration for the everyday user will increase as mobile devices empower more users to perform work and collaborate “on the go”.

This project presents two interaction features that have been developed utilizing web technologies to ultimately support mobile collaboration. Two features have been developed. The first is incorporating drawing functionality into Google Maps such that users can leverage natural gestures (i.e., drawing via continuous motion using a finger or mouse) to add information to the digital map. The second is the development of Map Features. Map Features are geospatially designated areas on the map that serve to hold information related to that area. Map Features can be created and modified by users to provide environmental knowledge to the map.

II. Background

Mobile collaboration focuses on the processes and tools that allow users to work together using mobile devices [8]. Mobile collaboration is an extremely broad area, and in many ways, can be thought of as a specialization of computer supported cooperative work (CSCW). Research in the CSCW field indicates two primary forms of cooperative work that can occur: closely-coupled and loosely-coupled [9]. Loosely-coupled CSCW is work that typically supports an extremely high
level of autonomy amongst collaborators. For example, a distributed group of users editing a shared document at their leisure is one type of loosely-coupled CSCW. Closely-coupled CSCW is typically concerned with a group of individuals working together closely to achieve specific goals. The emphasis on teams and the roles of individuals is much more important in closely-coupled CSCW, as it is assumed that the team must work closely together to achieve goals in part due to the specific skills team members possess and the unique roles team members perform [9]. A wilderness search and rescue team performing the time sensitive task of finding a missing person is one example of closely-coupled CSCW.

Closely-coupled mobile collaboration (i.e., collaboration which imposes team dynamics and roles, among other things) is less provided for, however. Previous research has focused on closely-coupled mobile collaboration in a domain specific sense (e.g., [10–12]), and while this previous research is relevant to mobile collaboration, it does not attempt to generalize a mobile collaboration framework to support work domains external to the research. Middlewares for mobile collaboration support also exist [13–17], but these focus primarily on providing reliable communication network support. Some provide front end widgets that can assist with mobile collaboration (e.g., [13]), but these are typically simplistic and do not leverage team dynamics (e.g., roles, leadership hierarchies, team membership, etc.).

There is an increasing need for mobile collaboration support as mobile devices increase in computational power and functionality, allowing more individuals to perform tasks unfettered from traditional computing devices. Mobile collaboration is a necessary and relevant part of many work domains [3, 4], and as mobile device sales continue to increase [6], it is expected that more domains will benefit from mobile collaboration software applications. In addition, there is an increased need for organizational computing, requiring individuals to perform computational tasks in varying locations and times of day [7].

Despite the increasing popularity of mobile devices, it is doubtful that a standard platform for all mobile devices will emerge. Many mobile platforms currently exist today, with iOS and Android forming a large portion of the mobile device user base [6]. However, web standards offer a platform independent approach to mobile software design, since much of the web’s functionality can be utilized on any relatively capable mobile device that possesses an up-to-date web browser.

Utilizing web technologies helps to ensure cross-platform and cross-device functionality. It is important to support device heterogeneity within a mobile collaboration scenario to accommodate users across varied domains and hardware requirements [8]. Web technologies will ensure that the only requirement for a mobile device is to run a web browser, which is included with many current mobile devices. Relying on the latest web-technology standards (e.g., JavaScript, HTML5, CSS3, etc.) will also assist in future proofing, accessibility, and generalization. For example, providing web-services alongside a robust application programmer interface (API) framework, web-based geographic information systems (GIS) achieved a high level of accessibility and popularity in both academia and in general use [18–21]. It is believed that the same benefits experienced by the GIS community may be realized for mobile collaboration.

In the mobile collaboration domain, maps are logical components to the user interface; individuals are inherently distributed throughout a geographic region and information can possess geospatial components that a map can easily visualize. However, utilizing a digital map in a generalized mobile collaboration context proposes difficulties. For one, mobile device screen sizes are inherently diminutive, leading to difficulties displaying information effectively [22]. Information clutter is also a concern on map-based interfaces and various techniques have been developed to reduce visual clutter [23–25]. Coupling a digital map with a mobile device poses unique information visualization
Fig. 1: The Google Maps API default drawing overlay. Drawing controls are shown as a callout and the three types of shapes that can be drawn are shown.

challenges in terms of the shared workspace. Information must be presented in a way that it can be easily interacted with and shared, and such a presentation must be mindful of the limited screen space.

Drawing is one interaction method that has employed with success in a map-based interface [26]. Previous work has determined that, for touch interaction, drawing using a finger is a faster method of specifying lines and areas than a user specifying each individual point. The gesture associated with drawing (i.e., dragging the finger across the screen to form a continuous line) was also reported to feel more natural than drawing by specifying individual points [26].

Shared feedback provides information concerning the individual collaborator’s actions to other team members [27], and provides a viable solution to promoting awareness. A digital map-based shared workspace presents opportunities to provide shared feedback to other users. For example, the map can be updated automatically with the locations of other collaborators by utilizing GPS sensors found in many current mobile devices. Other items can be provided to promote shared feedback on the map as well, such as information and tasks created by collaborators. This work is an attempt to promote shared feedback for mobile collaboration by using a digital map to represent a geospatial ordering of relevant, collaborator supplied, information.

III. Extending Google Maps

The Google Maps API provides a robust means of extending the service’s base offerings. This project focused on extending Google Maps by providing drawing support and introducing a new concept, Map Features. Drawing support is already supported by Google Maps in a relatively rudimentary fashion. Users can specify points that are connected to form lines, polygons, or circles via the Google Maps Drawing Overlay (see Figure 1). This project extended Google Map’s rudimentary drawing functionality by adding the ability to draw continuous lines. Continuous lines facilitate the simple creation of curved lines and complex shapes. Point-to-point drawing tools can also support the drawing of complex shapes, but each point must be specified individually, so drawing the shape can be time consuming.
In order to support line and area drawing in Google Maps, an application was developed which enabled the drawing of continuous lines. Several design challenges were present when implementing drawing functionality, with the largest being providing drawing support that did not interfere with the rest of the Google Map application’s function. For example, we did not want to implement a drawing approach that limited the user’s ability to interact with the Map’s UI elements (e.g., buttons, sliders, etc.).

In order to support drawing, an HTML5 canvas element was inserted into the Google Map. The canvas element is the size of the view and follows the view as the user pans the map, such that the canvas is always the full size of the visible map area the user can interact with. Since the canvas is embedded into the Google Map, it resides underneath any interface button overlays, such as Google’s zoom controls, therefore the user can still interact with button overlays while the canvas is active. While not implemented, adding multi-touch gesture support (e.g., two finger panning and pinching to zoom) would allow the user to pan and zoom the map while drawing.

Interaction with the canvas is facilitated through selecting a drawing button (see Figure 2) overlaid on the map. Once selected, the user can draw a shape using either the mouse or, if on a mobile device, his/her finger. Drawing is accomplished by using the canvas to sample a user’s touch/mouse points. As these points are sampled, lines are drawn to connect each point. The result is a continuous line that, due to the sampling rate of the input device, appears to be a freeform line (see Figure 3). Once drawn, the line is embedded into the map at the correct geospatial coordinates, so that panning or zooming the map causes the drawn line to transform appropriately.

The application runs on web browsers as well as mobile devices. In both cases, the application is accessed through the web browser. The map utilizes CSS styling to automatically resize itself to be the full size of the browser window. On a mobile device, this is full screen, on a desktop or laptop, this is the size of the running browser window.

Mobile devices have limited screen real-estate, and multiple items placed near to one another on a map can result in a large amount of on-screen clutter complicating interaction with individual items. Previous methods exist to remove on-screen clutter that work by grouping similarly typed items within close proximity [23]. However, group overlap can create cluttering when many unique groups overlap. Rather than a type-based grouping, a geospatial, feature-based grouping may reduce clutter and allow for a higher level organization of information items.

Map Features were developed to provide geospatial grouping functionality to Google Maps. Map Features are any geospatial map region (e.g., a building, a section of road, an abstract patch of wilderness, etc.) that can be used as containers to store information relevant to that geospatial...
location. Selecting a feature will expand it to reveal the associated information (see Figure 4).

Map Features are also created by drawing. A user can select the Map Features button (see Figure 2) from the drawing buttons UI and then proceed to draw a feature on the map using the same approach that is used to draw lines. In the current implementation, features appear as blue filled shaped that match the shape drawn by the user. Once the shape is created, the user can click on the feature to initiate a Google Maps InfoWindow pop-up which contains information relevant to the Map Feature (see Figure 5).

The Google Maps InfoWindow pop-up is a very robust option for displaying Map Feature information. Since the InfoWindow is simply a web page, it can be initialized using HTML. IFrames can even be embedded in the InfoWindow, essentially loading an entire web site and multimedia
Fig. 5: Four Map Features displayed in the current application. One of the Map Features is “active” displaying its InfoWindow.

Fig. 6: Map Features (shown in blue) revealed through drawing (left), additional features are revealed upon drawing additional areas (right).

content, such as video, into the InfoWindow.

Map Feature selection has also been implemented into this application. Map Feature selection is the ability to draw an area on the map and, if any features are contained within that area, they will become visible. The purpose of Map Feature selection is to allow Map Features to remain hidden, so they do not clutter the screen when they are not in use.

Currently, Map Feature selection is handled using a bounding box selection algorithm. Upon creation, both Map Features and the user’s drawn input have a bounding box calculated. When the user draws to select Map Features, a Map Feature is displayed on the map if its bounding box is located within the bounds of the drawn area’s bounding box. Figure 6 demonstrates the bounding box approach to selection. In the left image, a bounding box is completely enclosing two
Fig. 7: A Map Feature displaying a notification icon (shown in red) to alert the user that one new item has been associated with the Map Feature.

Map Features (which are shown in blue) and partially enclosing a third Map Feature in the lower right. Since the third Map Feature is not fully enclosed by the drawn area (shown in red), it is not displayed. The right image in Figure 6 shows that, when enclosed by a new drawn area, the third Map Feature does appear to the user.

Map Feature selection is just one approach to manage the display of Map Feature. Another implementation could be placing Map Features in layers whose visibility could be toggled through an additional menu. However, the drawing approach does prevent the need for additional interface components, such as menus to manage a list of Map Feature layers.

IV. DISCUSSION

The developed methods, drawing and Map Features, have been implemented with the intent of eventually integrating them into a system to support mobile collaboration. Drawing was developed to provide a more natural interaction method for data input into a mobile, map-based interface, this facilitating the communication of information between collaborators. Map Features were designed to directly support collaboration. Map Features may prevent on-screen clutter, and they may also improve a collaborator’s overall awareness in terms of recognizing when new information from other collaborators has been added to the system, this supporting shared feedback. For example, Map Features can be combined with a notification system to provide salient, geospatially enforced updates to a collaborator’s user interface. An example of how a notification system when combined with Map Features may work is shown in Figure 7.

V. CONCLUSIONS AND FUTURE WORK

This project represents the first efforts to design features that can benefit a mobile collaboration system. Drawing functionality has been implemented into Google Maps to facilitate natural user input. Map Features have also been created to provide a geospatial filtering of information in an attempt to reduce map clutter. Future work will attempt to evaluate the developed features and determine whether or not they are beneficial to mobile collaboration. Development will also
continue to take place such that Map Features can be provided with more functionality, such as a notification system.

REFERENCES


