Search Your Mobile Sketch: Improving the Ratio of Interaction to Information on Mobile Devices

Aaron Wolin  
Texas A&M University,  
College Station  
Department of Computer Science  
awolin@cs.tamu.edu

Brian Eoff  
Texas A&M University,  
College Station  
Department of Computer Science  
bde@cs.tamu.edu

Tracy Hammond  
Texas A&M University,  
College Station  
Department of Computer Science  
hammond@cs.tamu.edu

ABSTRACT
A mobile device’s small interaction space and undersized keyboard can sometimes make textual input difficult and impractical. Many mobile devices are predisposed for sketching as they come with a stylus or touch-screen capabilities, and sketched icons are a natural way to label objects on such a device. In this paper we present a sketch recognition overlay in Google Maps that allows users to search for location markers based on simple graphics and hand-drawn symbols.

Author Keywords
Handheld Devices and Mobile Computing, Pen-based UIs, Pen and Tactile Input

ACM Classification Keywords
H.5.2 User Interfaces: Input devices and strategies

INTRODUCTION
Web searching is a competitive market that allows users to accurately find information based on metadata. Most of this metadata includes keywords associated with websites, images, and other media. Whereas desktop web browsers have easy access to full-size keyboards to perform these metadata searches, mobile devices only have a small interaction space. The emergence of affordable touch-screen phones and PDAs with internet access has opened the possibilities for sketch-based searching.

The purpose of sketch recognition is to identify free-hand drawings so that computation can be performed using what they embody. Sketches represent more than a surrogate to match to a particular object; they also represent the dimensions and positioning of the object. Sketching is a superior medium of representing ideas/concepts in many domains. Sketching enables spacing and dimensions to be understood better than textual descriptions of the same idea. An example is an architect who conveys his design through drawings. Many engineering domains (civil, electrical) use sketching as a means to quickly understand a problem before moving on to a higher-fidelity representation.

Sketch recognition also moves away from the QWERTY keyboard paradigm that has plagued small mobile devices. Many mobile devices come with a stylus for the purpose of navigation, selection, and handwriting input, but the full expressive power of a stylus has not been incorporated into mobile applications. Sketches can convey information difficult to express through a keyboard. Studies have shown that simple visual symbols can be easily recalled [6]. The addition of sketch information as a search functionality enhances a user’s ability to find data by symbols when the linguistic information is elusive. A picture is often language-independent, whereas a textual description requires the user to speak the language of the designer. For instance, on a ski slope, the visual images supplied for the trail markers for easy, medium, difficult, and very difficult transcend beyond languages and countries. All countries use the standard circle, square, diamond, and double diamond notation. However, a textual search would vary from language to language.

In this paper we present a prototype system that allows for the querying of iconic images and sketches for geographic locations. Our system allows users to label real-world locations with iconic sketches through the Google Maps API. Users can then sketch an icon to search for their labeled
sketches or preloaded icons. Google Maps (and other web-based maps), typically require many textual keywords to find a location: Business, Street, City, and Zip code. Since mobile devices are beginning to use GPS information, a search using an iconic sketch and the device's current location can provide a significant benefit to users.

We will describe our prototype and the allowed interactions. A variety of scenarios were created to test our application and to demonstrate the contribution of a sketch labeling and searching system. The final section describes some user reactions and our future work to further develop our prototype.

PREVIOUS WORK
Jacobs et al. created an image querying application that extracts information out of images using wavelet decomposition [4]. Their system uses color both in image extraction and sketching; a mobile user will not have access to a full color palate due to the small interaction space.

A search engine for 3D models was developed by Funkhouser et al. [2]. Their system extracts 2D contours for the 3D models at different rotations and tilts and compares this information to a 2D sketch. We feel that 3D models are unnecessary for our domain, since icons provide a better interface mapping to equivalent paper map legends.

Recently, the MARQS system has demonstrated that graphical symbols can be used as queries for media object retrieval [7]. The system allows for multi-stroke symbols to be drawn without constraints. Unfortunately, the algorithm's search time can be large in some cases, and the online algorithm requires constant retraining. Both issues are unacceptable for a mobile, web-based application.

Image template matching is also an important sketch recognition technique and is more impervious to symbol drawing style than feature-based algorithms, such as [8]. The Hausdorff distance between two templates has been widely used to match images. Buttenlocher et al. use the Hausdorff distance model to compare complex 2D images, including the ability to match partial images [3]. Dubuissone and Jain propose the use of a modified Hausdorff distance to match templated images while avoiding outlier issues presented in a regular Hausdorff distance function [1]. More recently, Kara and Stahovich use three separate image comparison techniques, including Dubuissone's Hausdorff distance metric, to match drawn strokes with defined templates [5]. Our recognition algorithm relies on using the modified Hausdorff distance to search for sketches and images. We found the algorithm to be robust enough to handle sketch-sketch searches as well as sketch-image searches.

IMPLEMENTATION
Our interface is built using JavaScript, the Google Maps API, and the Hausdorff distance algorithm.

Google Maps API
The Google Maps API is a JavaScript map implementation that displays world satellite photographs, road information, route information, and allows for the placement and manipulation of location markers. Our search interface utilizes custom location markers that contain stroke or image information; these markers enable users to search for real-world locations through meaningful graphics.

We override the default Google Map marker, which is a pin-point latitude and longitude location, in two ways:

1. SketchMarkers are user-drawn sketches associated with a Google Map location. They allow a user to create personalized markers on the fly. Users can sketch multi-stroke symbols for these markers.

2. ImageMarkers are jpg and gif files associated with a Google Map location. ImageMarkers are ideal for use by business chains, map legend icons, and other repeating locations. Ideally, the creator of the ImageMarker should make is clear in the icon how to draw the symbol.

Additionally, we altered the Google Maps API to provide a space where users can draw and search for sketches as described below.

Hausdorff Distance
The Hausdorff distance algorithm measures the differences between two sets of points, or images [3]. The Hausdorff distance, \( h(A, B) \), defines the directed distance between the sets of points \( A \) and \( B \) as the maximum of the minimum neighbor distances between a point \( a \in A \) and all points in \( B \). In other words, \( h(A, B) \) finds the maximum distance bound for a point \( a \in A \) to be away from a point \( b \in B \). The Hausdorff distance equation calculates the maximum of both the directed Hausdorff distances between the point sets \( A \) and \( B \).

We use a modified Hausdorff distance function that accounts for outliers. Instead of finding the maximum distance bound between points, the \( \lVert a - b \lVert \) distances are sorted in ascending order, and the \( K^{th} \) distance is chosen as the directed distance value. In our implementation, we take the 90\(^{th}\) percentile distance that ignores the top 10\% distances.

The use of Hausdorff distance matching allows people to draw and search for multi-stroke sketches. All sketches can be simplified to a set of points, and two drawn sketches can be compared to each other by feeding their sets of points to the Hausdorff distance equation. Although time values and other data is ignored when examining only the points, the benefits of this approach include:

- Freedom from drawing direction (e.g., clockwise circles versus counterclockwise circles)
- Handles multi-stroke sketches
- Impervious to stroke overtracing where a single object is drawn multiple times in one location
- Scale invariant

To effectively use the Hausdorff distance equation, both the sets of points need to be scaled to a similar bounding box. A
(a) A user searches for any previously labeled Double Black Diamond trails by sketching the marker’s custom symbol.

(b) The system finds the nearest Double Black Diamond trail and moves to the corresponding SketchMarker.

(c) The system could also find the nearest Double Black Diamond ImageMarker.

Figure 2. Users can search for custom SketchMarkers or ImageMarkers, such as personal markers for the ski and snowboarding trails at Copper Mountain in Uninc Summit County, Colorado, USA.

drawn sketch or image’s bounding box is stretched or shrunk to a standardized width and height, which distorts some of the original graphic’s features. Due to this scaling, certain symbols, such as rectangles and squares, cannot be distinguished from one another. Yet, the benefit of scaling the symbol allows search sketches to be drawn differently invariant to size, centering, and proportion skewing. In our application, each drawn sketch is scaled to a normalized, $256 \times 256$ pixel bounding box. The scaled sketch is then compared to each of the SketchMarker and ImageMarker graphics and scored using the Hausdorff distance function, and the Marker with the best score is presented to the user.

It is important to note that our Hausdorff distance equation is not rotation invariant. A method to find optimal translations and rotations is discussed in [1, 3], but these techniques are currently not integrated into our prototype.

INTERFACE
Our system lets a user view drawn sketches or ImageMarkers, label custom SketchMarkers with user-drawn symbols, and search for markers through sketches.

Viewing
The sketch interface overlay includes the addition of SketchMarkers and ImageMarkers, which are Google Map latitude and longitude location markers that are respectively associated with either sketch or image graphics. When selected, a SketchMarker opens a translucent square, called a SketchPad, that displays the stroke associated with the marker (Figure 2(b)). The SketchPad is displayed to the right of the current marker location. A ImageMarker displays the associated image and its name through a pop-up balloon emitting from the marker (Figure 2(c)). Sketch displaying is done through the JavaScript Vector Graphics library, which can be found at: http://www.walterzorn.com/jsgraphics/jsgraphics_e.htm.

Adding a SketchMarker
SketchMarkers can be created by double-clicking on a location on the map. When the SketchMarker is single-clicked for the first time, the SketchPad opens and also allows the user to draw a multi-stroke sketch to associate with the SketchMarker. When the SketchPad is in drawing mode, the map becomes rigid and cannot be navigated by dragging. After the SketchPad is closed, the drawn sketch is saved in the user’s local session and can be searched for. The map can then be navigated again by dragging.

By empowering users with the ability to label Google Maps markers with sketches, people with mobile devices can easily mark important locations on a map graphically without
having to switch to a keyboard interface. For example, if a user is on a ski slope, labeling a geomarker with a mobile keyboard would be a hassle when wearing thick snow gloves. Instead, a user could more easily label a marker though a simple sketch that requires much less precision (Figure 3). Users can similarly search for previously labeled SketchMarkers, such as in Figure 2.

![Figure 3. Four standard ski and snowboarding trail rating symbols and their corresponding image template representations. Each template can be searched for through sketching.](image)

**Adding an ImageMarker**

Image matching is an important functionality in our sketch-searching prototype since image icons greatly increase the power and utility of a graphical search system. Continuing with our ski and snowboarding example, ski trail difficulty symbols are simple, universal symbols (Figure 3). As such, having the symbols predefined, marked, and available for mobile users will assist skiers since they will not need to define their own symbols for each slope. Figure 2 depicts the ability for a user to search for trails marked with images.

Currently, our system requires all ImageMarkers to be predefined in the system. JavaScript’s image handling techniques are poor, and the language cannot extract pixel information from images. To compensate for this issue, the pixel information is extracted beforehand, and both the image location and pixel information are stored within a defined Template class. Pixel information in an \((x, y)\) point list can be extracted from an image by searching for black pixels, and we created a C# application that automatically generates the JavaScript template code for a given image. These image templates are then loaded through a JavaScript file for use in ImageMarkers within prototype applications.

**Searching for Sketches or Images**

Once a SketchMarker or ImageMarker exists in the system, it can then be searched for by a hand-drawn sketch. In order to search for a labeled location, users click the Search button at the top left corner of the map (Figure 2(a)). A SearchPad then opens and allows the user to draw a multi-stroke sketch. After the sketch is drawn, the user closes the SearchPad by reclicking on the Search button. The best template match is found, and the map moves to the closest SketchMarker or ImageMarker’s location (Figure 2).

**RESULTS**

We have built a prototype system of MobileSketch. We have tested our application on both a standard web-page and an IPhone emulator. We preformed a preliminary user study where users searched for shapes using a ski slope trail, a golf course hole identification, and a business location application. Additionally, they created their own SketchMarkers for them to search for later.

We have had three different users experiment with our prototype system. Each user was enthusiastic about the application and the possibility of the application’s broader impact. The users stated that they enjoyed using the application and would use such a system if available in a wide variety of situations. Several other domains were suggested for this device including marking rock-climbing, hiking trails, and various map legends. Additionally, they suggested some improvements mentioned in our future work section.

**Skiing Application**

Ski slopes have a basic symbol notation that designates the difficulty of the slope. These slope symbols include circles, squares, and diamonds, which can all be easily represented as either images or simple sketches (Figure 3). Ski resorts can create ImageMarkers for each slope (Figure 2), which allows skiers with touch-screen, web-enabled phones to access the map and then easily search for a slope without having to take off their thick gloves to type.

**Golf Application**

On a golf course, markers with flag symbols can indicate the hole locations. Golfers unfamiliar with a course can then utilize their mobile devices for assistance. Figure 4 shows a screenshot of our system searching for flag markers on a golf course.

![Figure 4. Flag markers for a golf course.](image)

**Business Application**
Many businesses have icons that are easily recognizable and simple to sketch. For instance, McDonald’s arch is memorable and more elementary to search for than typing the string “McDonald’s”. Other, more ubiquitous icons include an icon for gas stations, parks, churches, or airports.

FUTURE WORK
Despite the enthusiasm surrounding our system, it is not yet available for public deployment. We are currently looking into the the backend requirements to make our system robust for simultaneous searches performed by multiple users through the same shape set. Additionally, and in particular, the following issues must be addressed and implemented for our public version:

- The system does not currently save its state such that users can create images to be shared across machines. The backend must be updated to allow for server-side storage, user account creation, security systems, and information sharing between users.

- MarkerImage icons currently have to be added programmatically. The system needs to be updated to allow users to upload marker icons.

- The system currently does not take into account GPS distance away. Thus, a user may be presented with a template match across the world. We are developing an updated distance metric that combines Hausdorff distance with geodesic distance.

- Currently the system does not allow for multiple markers for be returned for a single sketch, and will necessarily return the best template match, even if it is not very close. We are working on a rejection threshold that effectively takes into account our updated distance threshold that combines geodesic and Hausdorff distance. Once an effective threshold is found, all matching nearby sketches will be returned, and possibly no sketches will be returned if there is not an appropriate matching sketch. This threshold will have to be variable to take into account the case in which there are no close sketches by a geodesic measure, but there are close sketches by a Hausdorff measure. Thus, having an appropriate measure for choosing the appropriate variable cutoff threshold will require further studies.

- Some additional user interface issues still need to be addressed. For instance, the system needs to provide an effective way to provide non-trivial names to markers, as well as provide a way to display information about multiple results on a small mobile screen.

CONCLUSION
This paper describes a prototype system that allows users to hand-sketch labels onto Google Maps and search for these locations using sketches. The use of a modified Hausdorff algorithm allows for an image and sketch searching technique independent of drawing style. Several users have tried our system with domains including golf and ski trails, and we have found the use of graphics and sketching as a search technique to be a promising application. Although there are technical issues that still need to be addressed with the system, this paper demonstrates a novel application to move mobile UIs away from QWERTY keyboards.

ACKNOWLEDGMENTS
The authors of this paper would like to thank the members of the Sketch Recognition Lab for their insight on the ideas of this paper. This research is supported by the NSF IIS Creative IT Grant #0757557 Pilot: Let Your Notes Come Alive: The SkRUI Classroom Sketchbook.

REFERENCES