
A Mobile Guide for Serendipitous Exploration of Cities

Eva Hornecker

Dept. of CIS
University of Strathclyde
26 Richmond Street
Glasgow G11XH, UK
Eva.hornecker@cis.strath.ac.uk

Stuart Swindells

Dept. of CIS
University of Strathclyde
26 Richmond Street
Glasgow G11XH, UK
sswindel@cis.strath.ac.uk

Mark Dunlop

Dept. of CIS
University of Strathclyde
26 Richmond Street
Glasgow G11XH, UK
Mark.Dunlop@cis.strath.ac.uk

Copyright is held by the author/owner(s).
MobileHCI 2011, Aug 30–Sept 2, 2011, Stockholm, Sweden.
ACM 978-1-4503-0541-9/11/08-09.

Abstract

In this paper we describe the design concept, prototype development, and initial findings for a mobile guide supporting serendipitous exploration of a city. The system will allow a tourist to freely explore a new city, while providing them with peace of mind to not accidentally walk past attractions they desire to see. We have developed a proximity model and vibration patterns for alerts, and devised ways of mitigating trade-offs between battery use and location accuracy.

Keywords

Mobile guide, serendipity, battery life, tactile feedback

ACM Classification Keywords

H5.2. Information interfaces and presentation: User Interfaces.

General Terms

Design, Human Factors

Introduction

Most mobile guides are designed with an information-push model in mind. Systems built after the model of guidebooks guide a visitor along a predefined route. Contemporary context-aware systems follow a different model of providing suggestions according to a user's

interest profile. With the ever-increasing amount of geo-tagging, every spot within a city nowadays will have multiple annotations. Well-designed recommender systems may lure tourists 'off the beaten track' [8]. While such systems provide useful information when searching for something specific or may inspire going to new places, the constant flow of information may create information overload and distract visitors from exploring and experiencing the atmosphere of a city [7].

An alternative mode of sightseeing entails wandering, wondering, and discovering a city like an explorer, led by intuition and curiosity. A visitor may want to feel they are finding hidden gems and quirky places that other tourists fail to see. Walking aimlessly provides a completely different experience and gives us more ability to take in the atmosphere and the little things that make up a city's character. Getting 'lost', we might discover the real city next to the tourist area. On the other hand, the next day we might realize that we walked right past one of the places which we really wanted to see, and then have to visit the same area again. Many people feel compelled to check the map frequently to avoid this. Our project explores ways of combining both modes of sightseeing - allowing for serendipitous discovery while providing peace of mind of not accidentally missing those attractions one wants to see. The core idea is that the system only notifies the user when coming into the vicinity of previously selected attractions, and then provides further guidance.

The system idea originates from an earlier concept sketch by some of our students, based on observations of tourist navigation in Vienna. This is briefly described in [3] as an example for concept prototyping. At the time, students envisioned a dedicated device that

would be sold along with brochures and used to scan barcodes on these. It was envisioned that the sides of the device would vibrate (left, right, both) when close to a site so that users could follow tactile cues. Arriving at a site, the tourist could then ask for further information. We have now developed a working prototype implementing this concept through a student project on a mobile phone, updating it to reflect the advances in mobile technologies since its original inception.

Our project relates to work in MobileHCI on guide and recommender systems, haptic feedback [2, 6, 7], research on opportunistic information provision [4] and takes inspiration from the notion of serendipity, allowing for fortunate chance encounters. Turning common models of location and context-aware mobile guides upside down (which usually notify users of surrounding places according to their interest profile) system activity is limited. Instead of relying on the provision of serendipitous information by the system [1, 8] based e.g. upon recommendations by other users, serendipitous discovery is left up to the user. Our system is related to Tacticycle [5], a system that supports bicyclists in exploring an environment spontaneously without losing orientation by providing awareness of direction, but, in contrast, provides a middle ground between trip preparation and free exploration. Furthermore, our proximity model seeks to minimize power consumption, a practical issue often ignored in guide system design.

The Serendipitous City Guide

We start by explaining the overall system design and the use process. We focus on the proximity model for alerts and location tracking, which mitigates between conflicting requirements of accuracy and battery use.

The user first selects locations they want to see. Based on the original concept, QR-codes may be scanned from guidebooks or brochures, which are converted to GPS coordinates and may also contain textual descriptions. This approach simplifies the process of location selection and does not require interaction with maps. Alternatively, google maps can be used to select locations. Both means of selecting locations may be used, resulting in a merged list. Before starting the walk, users can review their list of locations and edit it. Furthermore, they can set preferences on how to be notified, via vibration or sound alerts or both.

X: (outside of green circles) wifi or GSM/UMTS data used to estimate location (or low-frequency GPS if no other information available)

Green half circles: GPS is polled at approximately 10 second intervals.

Brown circles: GPS is continuously updated, providing real-time information (battery intensive).

Black circles: Notification threshold. Alerts get quicker closer to the site, and slower when moving away. On moving in and leaving again, an irregular alert pattern is produced.

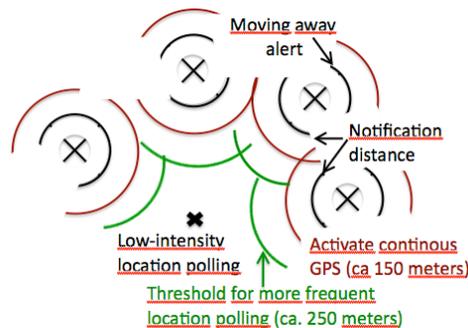


figure 1. Threshold model for location polling and notifications. The user is at the spot indicated with an X. None of the 4 sites is yet close enough for the phone to react.

Design of Tactile Alerts and the Proximity Model

In use mode, the application provides notifications on moving into proximity of chosen locations by audio and/or vibration alerts and a text notification. The device alerts users of coming close to a chosen site, and of moving away from it again. Tapping the notification text on the phone offers a choice between receiving directions, no longer receiving notifications for that loca-

tion (remove from list) or ignoring the notification. In the last case, the user is notified when coming back into proximity. This alert model aims to provide directions while allowing the user to take a detour.

One of the main design challenges in the project lay in proximity modeling, which includes the mapping between alerts and proximity as well as mitigating between location accuracy and energy consumption (see figure 1). This is because battery power was found to be a major issue, with GPS quickly draining power. There are three proximity thresholds (circles) around selected sites. Entering the outer-most circle makes the application check location more often. Within the middle circle, location is updated continuously. The inner-most boundary triggers alerts. We first describe the alert proximity model. As the literature indicates that haptic feedback is superior in terms of requiring less active attention [2; 6], we focused on vibration alerts. Non-visual feedback is furthermore preferable as users of such systems typically prefer a device which can be put in a pocket [1] and does not mark the user out from their surroundings. The audio alarm currently employs the standard alarm ring tone, but could follow the same schema as for vibration.

The alert becomes more intense if the user moves closer to the location and changes in rhythm when the user moves away again, indicating different degrees of urgency. At the proximity threshold, a regular vibration pattern based on 500 ms intervals is employed. This is then further modulated through the distance to the site, with more frequent vibration on getting closer. This simulates a 'hot-cold' game schema, which informs the user whether they are moving in the right direction (cp. [7]). If the user moves away and leaves the notifi-

cation threshold, a different kind of alert is given, irregular in pattern, and thereby different from the other alerts. Because road layouts may require detours (having to walk around the block), and users may not want to immediately attend to it, this alert should ideally be provided only once the user leaves the proximity threshold distance again after having entered this zone.

Accuracy of location detection and battery time are a core tradeoff for our application, with GPS quickly draining batteries. We thus use a combination of GPS (battery intensive, slower to activate but accurate) and WiFi as well as mobile network data location fixes (quick but inaccurate), using different location tracking modes depending on the proximity model in figure 1. Location is updated in larger intervals while all of the selected sites are further away (at least 3 minutes walking distance, equaling about 250 meters). In this stage, GSM/UMTS information is deemed sufficient, in particular as the application is designed for inner-city areas, where cells tend to be smaller. Once this threshold is reached, GPS location polling is done every 10 seconds. Only once a second threshold of about 150 meters distance to sites on the wish-list is reached, is the GPS connection upheld, resulting in continuous real-time updates, but also draining batteries. The exact distance of these thresholds has been derived based on average walking speeds, distances covered and delays in updating locations (see implementation details).

We furthermore found that the design of these thresholds should take account of the grid layout of the city area the service focuses on, and of what feels 'far away' or close [4]. If there is no direct way, and one needs to walk around a block, the system should not prematurely provide the 'moving away warning'. A dis-

tance of 1-2 street corners seems adequate in most cases. This average distance will differ between cities. Ideally, the system would know about walking routes, including e.g. alleys and underpaths. At the moment, such information is not available on google maps.

Implementation Details

The prototype was developed for a standard HTC Desire smartphone in Java on Android, using many of the standard Android-supplied user interface elements. The application was implemented as a service, allowing it to run in the background while the phone is locked (e.g. to be put in the user's pocket), or while another application is running. An XML file contains all textual elements for the application, making the application code independent of the concrete location/country it is used in. Locations and location information are stored in an SQLite database. The observer pattern is used to listen for location updates. This allows using a combination of GPS and WiFi and mobile network data location fixes.

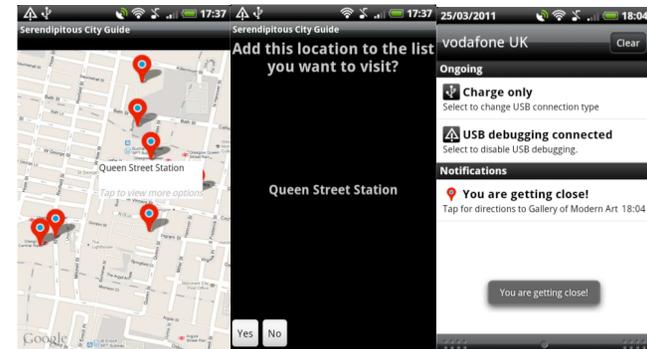


figure 2. Left to Right: Result of selecting a map marker. Confirmation screen to add location to the list. Notification alert.

On startup, the user is presented with a map view of the central city area of Glasgow (see figure 2 left). From this, a set of locations can be chosen, indicated via standard map markers. Pressing the standard Android menu icon reveals a menu for selecting preferences and scanning barcodes (figure 3). On tapping a map marker on the map, a popup window shows further information (see figure 2 middle). A further tap brings up a window asking if this site should be added to the visit list. Under preferences, users can choose how to be notified (vibration and/or sound). Once a number of locations has been selected for the visit list, a 'Go' button is added to the menu bar (figure 3), which starts the application. The user can then lock the phone to put it in the pocket, and walk about.

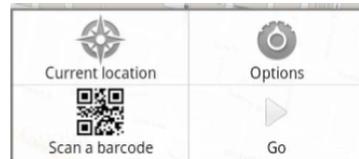


figure 3. Applications menu bar once first location has been added to list (before that there is no 'Go' button)

The Android Notification Manager service is used for proximity notifications and handles notification objects that cause the phone to make sounds or to vibrate, as well as showing textual notifications in the standard status bar. Implementation of navigation routing was complicated through the Google Maps API terms of service, which forbid use for real time navigation [9]. Instead, the Google Maps Navigation application was used, which gives walking distances and standard navigation instructions, but cannot be customized.

Preliminary Test Findings

For a preliminary test of the core functionality, the system was installed and started on an Android phone, and then handed to several test users who were given basic information about the system. For these tests, the area around our University campus was used and distances for proximity notifications were modified to limit time investment for test users. These were asked to explore the area while being observed.

User feedback quickly showed that audio notifications can be eliminated, as these were often inaudible in an urban environment, and rather were deemed distracting. Furthermore, audio notifications from the application were silenced if the phone had been set to silent or vibrate profile, the phone profile thus interfering with the application. The limited utility of audio notification is to be expected based on the literature.

The tests provided useful insights for the tuning of the proximity model. Even at normal walking speed it was possible to move a sufficiently large distance between location updates to transition from being outside the proximity threshold to inside and back again. If the location is updated every 30 seconds, then a user may cover 40 meters in this time with an average walking speed of 5 km/h. This was exacerbated by the time needed for the phone to achieve a GPS lock (an average time of about 15 seconds). This resulted in the proximity model and the distances/timings described earlier, with more frequent GPS polls (every 10 seconds) once any chosen sites come within a 3 minute walking distance (250 meters), and continuous real-time GPS tracking from about 150 meters distance.

Conclusion and Future Work

We have presented a prototype implementation for a mobile city guide supporting serendipitous discovery of an urban area, which limits information delivery to pre-selected locations. We have explored the design of tactile vibration alerts, and developed a proximity model that aims to balance conflicting requirements for accuracy of location detection and battery lifetime, as well as providing rich location feedback to users. This proximity model and the vibrotactile alerts need further testing and refinement before further research can focus on determining how users react to this kind of service and how it affects their experience of a city.

The system could in the future be extended in a number of ways to improve both usability and functionality. The user interface at the moment is very basic, using standard Android components. The 'action bar' UI principle could e.g. make the option for scanning barcodes more obvious, which is currently hidden under the menu button. Furthermore, the set of available locations at present is static after installation. This has the advantage of guaranteeing that locations are accurate and that the visitor is guided e.g. to the entrance of a larger building, and not just to its nearest part. On the other hand, besides of downloading extensions to the list of available attractions, it would be desirable to exploit the multitude of existing tags on Google maps beyond the range of those defined by the application provider. This would not only add further attractions, but also e.g. the visitor's favorite coffee bar, and potentially link to background information on wikipedia. Moreover, inspired by ideas from [5, 7], it would be interesting to add e.g. boundary regions, alerting the visitor whether they move outside of a core region they want to explore, and to provide awareness of general direction.

Acknowledgements

We thank our TU Vienna students Ali Eghdamian, Christian Wagner, Herfried Wagner, Matthias Mitsch and Stefan Schauer, who initially developed the basic idea for the serendipitous city guide.

REFERENCES

- [1] Bellotti, V. et al. Activity-based serendipitous recommendations with the Magitti mobile leisure guide. *Proc. of CHI 2008*, ACM Press (2008), 1157–1166.
- [2] Hoggan, E., Raisamo, R., Brewster, S. Mapping information to audio and tactile icons. *Proc. Multimodal interfaces ICMI-MLMI '2009*, ACM Press, 327–334.
- [3] Hornecker, E., Psik, T. Using ARToolKit Markers to Build Tangible Prototypes and Simulate Other Technologies. *Proc. of Interact 2005*. Springer, 30–42
- [4] Ludford, P.J., Frankowski, D., Reily, K., Wilms, K., Terveen, L. Because I carry my cell phone anyway: functional location-based reminder applications. *Proc. of CHI 2006*, ACM Press (2006), 889–898.
- [5] Poppinga, B., Pielot, M., Boll, S. Tacticycle: a tactile display for supporting tourists on a bicycle trip. *Proc. of MobileHCI '2009*. ACM Press, 41:1-41:4
- [6] Robinson, S., Eslambolchilar, P. and Jones, M. Evaluating haptics for information discovery while walking. *Proc. of British HCI'09*, British Computer Society (2009), 93–102.
- [7] Robinson, S., Jones, M., Eslambolchilar, P., Murray-Smith, R. and Lindborg, M. "I did it my way": moving away from the tyranny of turn-by-turn pedestrian navigation. *Proc. of MobileHCI '2010*, ACM Press, 341–344.
- [8] Tintarev, N., Flores, A., Amatriain, X. Off the beaten track: a mobile field study exploring the long tail of tourist recommendations. *Proc. of MobileHCI '2010*. ACM Press, 209-218.
- [9] <http://code.google.com/intl/uk/apis/maps/terms.html>