A flexible model for haptic-assisted pedestrian navigation mobile applications

Ricky Jacob, Peter Mooney, Padraig Corcoran, Adam Winstanley

Department of Computer Science, National University of Ireland Maynooth,
Maynooth,Co. Kildare. Ireland.
email {ricky.jacob}@nuim.ie

1. Introduction

Pedestrian navigation applications, specifically for mobile devices, have recieved much research and development attention over the past decade or so with many different types of solutions developed (Kenteris et al.; 2011). The most common interface is a map-based interface with written and/or verbal turn-by-turn directions (possibly including landmark information). Haptic technology, or haptics, is a tactile feedback technology that takes advantage of our sense of touch by applying forces, vibrations, and/or motions to the user (Nakao et al.; 2010). The potential of haptic technology has only recently started to receive the attention of the research community (Jacob et al.; 2010). In mobile devices haptic-feedback is delivered in the form of vibrations which can be programmatically controlled using the phone software API. In this paper we describe a simple, flexible, model for the integration of haptic feedback into pedestrian navigation applications on mobile devices. A constraint is that the mobile device must have an onboard GPS and compass. The vibration motor on the mobile device must also be capable of being controlled from software running on the device itself. Our model allows a “heads up” approach to pedestrian navigation with the mobile device where the user is not required to keep looking down to check the screen of the mobile device. For testing purposes text-based navigation assistance is provided in conjunction with the haptic-feedback on the device screen in our prototype implementation. Three distinct modes of vibration of the device are used to provide haptic feedback to the user.

2. Description of Model

Our model is presented in Algorithm 1 and described in Section 3.. The user starts our application on their mobile device. The first step involves choosing both the start location (default is their current location taken from the device GPS) and destination. A simple slippy map interface is provided for this purpose. When the user has selected the route start and end points the Cloudmade routing service (Cloudmade; 2011) is automatically invoked with the parameters describing the requested route. A GPX file (illustrated in Figure 1) is returned to our application on the device. This file is immediately parsed and stored in the spatial database (PostGIS). The application then indicates that the use must scan (see the $Scan()$ function in Algorithm 1 below) for the direction they should proceed in. $Scan()$ requires the user to hold the device in front of them and slowly move it to find the correct forward direction. The circular buffer size around each route point is also set (see Figure 2). If the user is inside the buffer of a route point the application causes the device to vibrate (pattern

1) to indicate the user must scan for the correct direction. When the correct direction is found the device vibrates again (pattern 2) to indicate the correct direction. If the use strays off in the wrong direction the application vibrates (pattern 3) and they must scan again or physically return to the route themselves. This process is repeated until the user has reached their destination.

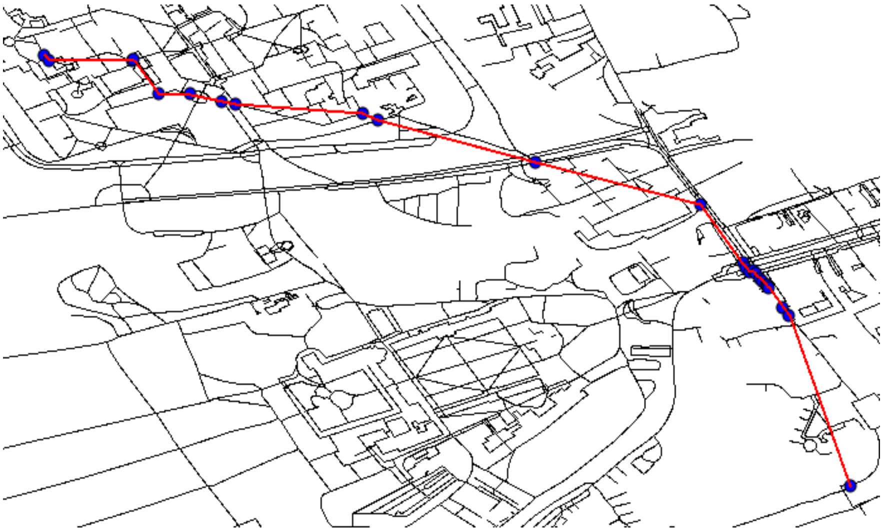

Figure 1: A GPX file output from the Cloudmade Routing Service overlayed on an OpenStreetMap *roads* shape file. Circular icons represent route points in the computed shortest route.

3. Implementation of our Haptic-Feedback Model

A HTC Magic, running the Android Mobile Operating system, was used for development and user testing. All software development on the software device was carried out in Java. The Android Software Development Kit provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language. The Cloudmade Routing service (Cloudmade; 2011) was used as the web-service for generating the shortest pedestrian paths. Cloudmade use the global OpenStreetMap database for computation of shortest paths. With the Cloudmade routing service we have more “human orientated” walking routes. If the OSM data has paths across open areas properly tagged then these are considered in the computation of the shortest walking route. The Cloudmade routing service returns computed routes in JSON or GPX formats. Figure 1 shows the resulting GPX output file generated by the Cloudmade Routing Service for the shortest pedestrian route from the Computer Science Department at NUIM to the Maynooth Business park. The circular icons represent route or turning points in the computed shortest pedestrian route. A schematic diagram of our implementation is shown in Figure 2. A PHP script runs on the database webserver and this script acts as a broker service between our local spatial database and the mobile device. The script recieves the user location data every t seconds. As described in Algorithm 1 the computed route is stored in this database. The PHP script matches the user location to this route and returns a response to indicate if the user is: going in the correct direction (no vibration), within the buffer of a route point (vibration 1), going in the wrong direction or is off route (vibration 2), or has reached their destination (no vibration). Initial user trials with this application have focused on observing and eliciting feedback on how test participants: interact with the haptic application, use the application at the route points, respond to feedback, and rate the usability or usefulness of the application. We will present quantitative results of a series of user trials at the GeoComputation conference and in future papers.

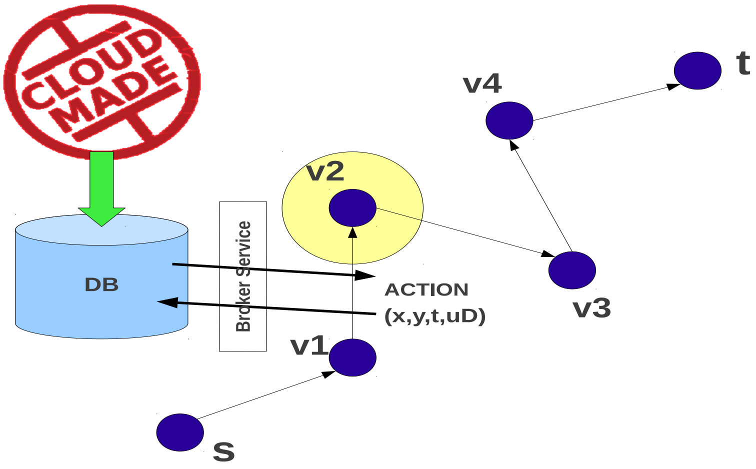

Figure 2: An example route. The user position and bearing is taken from the mobile device every x seconds. Upon delivery of this information to the broker software at the DB an action is initiated on the mobile device

4. Conclusions and Future Work

This paper has given a brief overview of a flexible model for integrating haptic-feedback into pedestrian navigation applications on mobile devices. To properly quantify the advantages and disadvantages of the model we will need to carry out additional user tests which will involve a wider range of participants. Initial feedback from our trials of the application with a small group of test participants was positive. The users found the application novel and quickly learned the vibration patterns and how to respond to this feedback. However, users, unfamiliar with the route, who used the haptic-enabled device, paused for longer times at route points compared to users familiar with the test routes. In an extensive study Ishikawa et al. (2008) compared the wayfinding behavior and acquired knowledge by participants who received information about routes from a GPS-based navigation system, from maps, and from direct experience of the routes. With respect to wayfinding behavior, participants who used the GPS-based navigation system traveled longer distances, made more stops during the walk than participants who viewed maps, and walked slower overall. We shall be carrying out similiar user testing to investigate if walking performance of pedestrians, using the haptic application on their mobile device, improves as less “head down phone viewing” is required. Finally, using the vibrate function on the mobile device can drain power quickly from the battery. Most mobile device users would find the possibility of a dead battery a serious drawback to haptic-assisted navigation. Battery lifetime is a major usability concern to mobile phone users (Rahmati and Zhong; 2009). Rahmati and Zhong (2009), in a study of mobile phone users, “found that most recharges are driven by pressures of time and location instead of low battery”.

Acknowledgements

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. Peter Mooney is funded by the Irish Environmental Protection Agency STRIVE programme (grant 2008-FS-DM-14-S4).

References

Cloudmade (2011). Cloudmade routing web services, Online API available at `http://developers.cloudmade.com/projects/show/routing-http-api` - Checked January 2011.

Ishikawa, T., Fujiwara, H., Imai, O. and Okabe, A. (2008). Wayfinding with a gps-based mobile navigation system: A comparison with maps and direct experience, *Journal of Environmental Psychology* **28**(1): 74 – 82.

Jacob, R., Mooney, P., Corcoran, P. and Winstanley, A. C. (2010). Haptic-gis: Exploring the possibilities, *SIGSPATIAL Special* **2**: 13–19.

Kenteris, M., Gavalas, D. and Economou, D. (2011). Electronic mobile guides: a survey, *Personal and Ubiquitous Computing* **15**: 97–111.

Nakao, M., Kitamura, R., Sato, T. and Minato, K. (2010). A model for sharing haptic interaction, *Haptics, IEEE Transactions on* **3**(4): 292 –296.

Rahmati, A. and Zhong, L. (2009). Human-battery interaction on mobile phones, *Pervasive and Mobile Computing* **5**(5): 465 – 477.

A Description of Algorithm for a Haptic-assisted Pedestrian Navigation Application

```
Data: The input start location $s$ and the destination location $t$ (both longitude,latitude)
      of the route required
Result: In-route Haptic-assistance for pedestrian using their mobile device on route
        between $s$ and $t$

Call Cloudmade Routing Service;
Download XML-encoded result from Cloudmade;
Parse and store route in Database;

begin
    $d \longleftarrow setRoutePointBufferSize(10m)$;

    while $(U.location \neq buffer(t, d))$ do
        $U \longleftarrow getCurrentUserLocation()$ This includes user direction;
        $D \longleftarrow getLocationOfNextRoutePoint()$;
        if $(U.location = buffer(v, d))$ then
            repeat
                Until user points their mobile device in the correct direction;
                Vibrate when correct direction is found;
            until $(Scan() = true)$;

            User can now proceed in the correct direction;
        end
        if $(U.location \neq buffer(v, d))$ then
            if $(U.direction = D)$ then
                Everything is OK;
                Put the green light logo on the phone display;
                No feedback necessary;
            else
                Not going in the correct direction;
                Vibrate;
                repeat Until user corrects their direction until $(Scan() = true)$;
                Update direction variables;
                OK to proceed;
            end
        end
    end
end
```

Algorithm 1: The algorithm describing the integration of haptic feedback and a pedestrian route