# Intuitive Map Navigation on Mobile Devices

Stefan Winkler, Karthik Rangaswamy, and ZhiYing Zhou

Interactive Multimedia Lab
Department of Electrical and Computer Engineering
National University of Singapore (NUS)
Singapore 117576
{winkler,elerk,elezzy}@nus.edu.sg

**Abstract.** In this paper, we propose intuitive motion-based interfaces for map navigation on mobile devices with built-in cameras. The interfaces are based on the visual detection of the devices self-motion. This gives people the experience of navigating maps with a virtual looking glass. We conducted a user study to evaluate the accuracy, sensitivity and responsiveness of our proposed system. Results show that users appreciate our motion-based user interface and find it more intuitive than traditional key-based controls, even though there is a learning curve.

**Key words:** Virtual map navigation, user interface, motion detection, pose estimation.

## 1 Introduction

Mobile phones and PDA's are becoming increasingly powerful and provide more and more functionality to the user. However, the user interface (UI) is still severely limited due to the small form factor of these devices. Using the small, un-ergonomic keys for control in applications other than calling is tedious and not always intuitive.

In order to overcome these problems, we have developed novel, motion-based user interfaces for mobile devices, which are based on the detection of the 3D self-motion of the device using the integrated camera. Possible movements involve translation and rotation with a total of 6 degrees of freedom. These movements can then be used to control applications on the device.

3D motion tracking using the phone's built-in camera forms the backbone of this work. The process involves motion detection, computing the pose of the phone in 3D space from the motion registered, and tracking [1]. The process has to run in real-time under the constraints of the limited memory and computing resources available on a hand-held device [2]. Work on this subject is very recent, and the current studies focus mainly on gaming [3]. We have presented a car racing game using motion-based interaction elsewhere [4]. Here we extend this concept to virtual map navigation, which also benefits greatly from this new way of interaction. Arhippainen et al. [5] investigated a similar problem involving gesture-based control for scrolling and zooming maps on mobile devices with motion feedback from sensors and accelerometers. However, these sensors are not always available on standard mobile devices. Since a camera is available with almost all phones in the market today, we propose to use visual motion detection and pose estimation to achieve our goal.

For the work described in this paper, we use fiducial markers as references to simplify the detection of the phone's motion. These markers can be carried in personal wallets, printed on maps, or fixed on traveler log books, which are often used for route planning. The pose of the phone in 3D space can be estimated with respect to the marker by analyzing the frames returned by the camera for any detected markers and determining the transformation from marker to camera coordinates. The vectors in the transformation matrix are mapped to various types of interactions that drive applications on the mobile device.

To test our interface, we developed an application for navigating virtual maps that involves standard controls for scrolling across the map as well as zooming in and out to see the map at different resolutions. We define specific device movements that translate into these basic controls. We designed three such user interfaces and conducted a user study to compare the performance of our motion-based interfaces with traditional key-based controls.

The paper is organized as follows: Section 2 introduces the system specifications and the system design. Section 3 explains the three user interfaces we designed and how the user can control the application with them. Section 4 discusses the user-study conducted and the results of the survey. Concluding remarks are given in Section 5.

## 2 System Overview

Our application runs on a Hewlett-Packard rw6828 PocketPC equipped with a 416 MHz Intel PXA272 processor and 64MB RAM, running Windows Mobile 5.0 OS. The phone has a built-in 2-megapixel camera, which is used to track the markers at a video resolution of 176x144 pixels.

The marker tracking is implemented with ARToolkitPlus [6], a modified version of ARToolkit [7] that is compiled for the mobile platform. ARToolkitPlus accepts the frames from a DirectShow FrameGrabber filter and returns the 3D position and pose vectors of the phone if a marker was detected. These vectors are then mapped to different controls on the user interface to drive the application. The complete flow of control in the system is summarized in Figure 1.

Maps are downloaded from Google Servers on the Internet whenever the user scrolls to unexplored regions or zooms to a different resolution of the map. The Map-Navigation application was developed in C++, making use of the Direct3D Mobile libraries for displaying maps as textures on a vertex grid. The vectors obtained from ARToolkitPlus are used to translate a virtual camera over the map textures by a pre-defined offset for every rendered frame, thus giving the notion of continuous scrolling. Zooming is implemented by a similar operation to view different resolutions of the map.

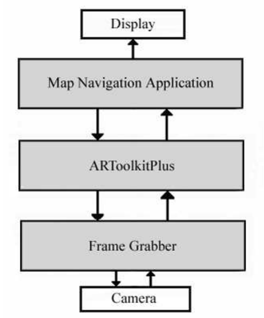

**Fig. 1.** Control flow in the system.

Figure 2 shows the Map-Navigation application running on our PocketPC with a marker in the background that helps with pose estimation. A small video feedback window on the top-left corner of the application displays the camera view. This window is provided so that the user can keep track of the marker and maintain it within the field of view of the camera. It also helps the user to scroll the maps in arbitrary directions, which will be explained in Section 3.

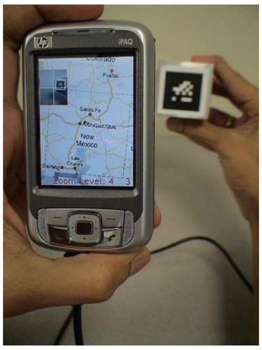

**Fig. 2.** ID-encoded ARToolkitPlus marker used for computing the pose of the mobile device in 3D space.

Furthermore, the mobile device can be rotated by any arbitrary angle to view a specific section of the map in landscape or portrait mode or any orientation in between, as shown in Figure 3; the orientation of the virtual map does not change, similar to how real maps would behave under a looking glass.

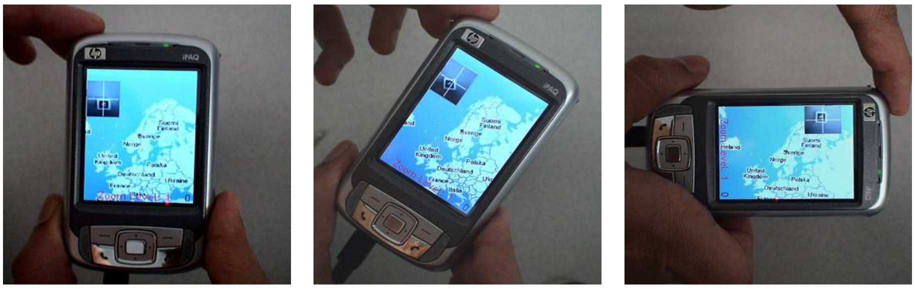

**Fig. 3.** The phone can be rotated to view different regions of the map, whose orientation remains fixed, as shown here for three different phone poses.

## 3 Interaction Methods

In terms of interaction, the well-established key-based interfaces are robust enough to navigate maps, but allow only a limited set of discrete movements and generally restrict the user's freedom. In our implementation, the user instead moves his phone over a pre-defined set of markers that are constantly being tracked by the phone's camera. Moving the phone over a given marker performs a particular operation, like scrolling the map in a certain direction, or zooming to a different level of detail on the map. To zoom in or out, the user has to move the phone towards or away from the marker, for example. In order to aid the participant in controlling zooming, we provide a 'zoom-meter' whose arrow-head indicates the zoom level as a function of the distance between the phone and the marker. This helps people to know the current resolution of the map and to get a feel of how far they have to move in order to get to the next level of resolution. The definition of zoom levels is explained in more detail below.

Although many different UI designs are possible with such a setup, we restrict ourselves to three designs for the time being. Our goal was not to find the best user interface design, but to get a general idea of how people view such user interfaces when compared to using keys on the phone.

### 3.1 Key-based Setup

For comparison purposes, we implemented a simple key-based interface for scrolling and zooming the virtual map. The maps can be scrolled with the four directional keys on the phone along the four cardinal directions. Two other keys are allocated for zooming in and out of map resolutions incrementally.

### 3.2 Five Marker Setup

This is the most basic setup involving four markers that guide continuous scrolling of the map along the four cardinal directions and one marker for zooming, as shown in Figure 4. The marker located in the center aids in zooming, where the user moves the phone towards or away from it. This requires that only one marker be visible within the camera field of view at any instant for unambiguous selection of the corresponding operation. The distance between the phone and the marker along the depth direction is used as a scale to fix the zoom levels for the application. The phone cannot be moved too close to the marker as the complete area of the marker has to be visible within the camera's field of view. On the other hand, the phone cannot be moved too far away either as the marker pattern might not be detected accurately. These two conditions impose physical limits within which we confine all the zoom levels for the application.

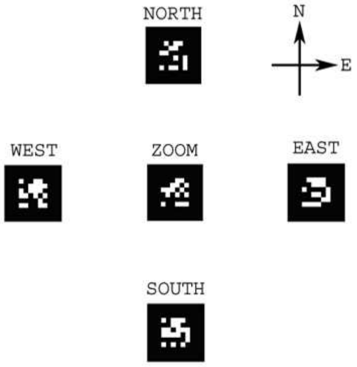

**Fig. 4.** Five-marker setup for map navigation.

We split the detectable range (maximum distance minus minimum distance) into as many segments as we have zoom levels in the application. In the current implementation, four zoom levels are implemented, from country level to city level. In this case, a direct jump from any given resolution of the map to any other is possible just by moving the phone more than a segment's length in order to skip the resolution(s) in-between.

### 3.3 Single Marker Setup

Five markers may be considered an unnecessary burden and also restrict the scrolling to one of the four cardinal directions. Therefore, we designed a user interface wherein both the scrolling and zooming operations can be achieved with a single marker.

For scrolling, the user has to move the phone over the marker such that the map scrolls in a direction defined by a vector drawn on the marker plane from the optical axis to the center of the marker center. As a result, scrolling is possible in any direction. Scroll speed is based on the relative distance of the marker from the optical axis. At the optical axis the scroll speed is zero, but it increases as the marker moves away from the axis in proportion to the distance between the marker-center and the axis on the marker plane.

The zooming operation here is very similar to zooming in the five-marker setup, i.e. moving the phone towards or away from the marker. Either the user can fix the marker at a place and move the phone over it or hold the phone at a constant distance from his view and move the marker in the background.

A limitation to the zooming operation in this setup is that as the phone is moved closer to the marker, the usable scroll movement range is reduced as the marker area in the camera view increases. As a solution, the scroll speed is set to constant at these zoom levels, and the users only have to move the marker center relative to the optical axis in their preferred scroll direction. Additionally, a thresholded region around the axis is defined where the scroll speed drops to zero. The user can move the marker center to this region to stop scrolling.

### 3.4 Three Marker Setup

We provided another alternative to the above strategies wherein zooming need not be controlled by moving the phone towards or away from the marker. In this setup, a single marker aids in scrolling while two other markers are used for zooming in and out of the map respectively. This setup is shown in Figure 5. Whenever the user moves his phone over a zoom marker, the zoom level increments or decrements in fixed time intervals defined by a timer count.

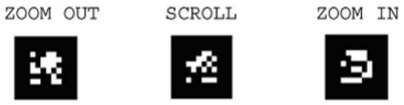

**Fig. 5.** Three-marker setup for map navigation.

Alternatively, the markers can also be fixed to a cube which can be moved by the user, with the phone held at a constant distance from the cube, as shown in Figure 2. A sample cube with markers attached to its faces is shown in figure Figure 6. The system is designed to tolerate more than one marker in the field of view, especially when the cube is rotated. The display freezes whenever more than one marker is seen by the camera, and the system waits until only one marker appears in the field of view.

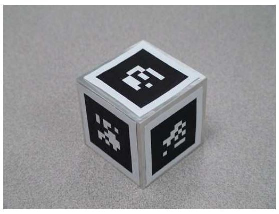

**Fig. 6.** A cube with markers attached to three of its faces.

## 4 User Study

We designed a user study in which people could evaluate each of the interfaces introduced above individually. The goal of the study was to evaluate the intuitiveness of the interface, the comfort in usage and the responsiveness of the system. These aspects were tested for both scrolling and zooming operations.

The initial group of people who tested our interfaces felt that the five-marker setup was quite similar to the key-based interface except for the automatic scrolling and zooming. Five markers were a burden to carry along with the phone, especially when the scrolling operation seemed redundant compared to scrolling with keys on the phone. This was the reason that led to the design of the single and three marker setups and the expulsion of the five marker setup from the user study.

12 male participants and 4 female participants between 16 and 28 years of age took part in the study. Almost all of them had experience with a map navigation application on the PC platform, while only one of them had used it on a mobile PDA. Owing to this lack of experience, we let the users experiment with our key-based interface for map navigation on the mobile phone for some time in order to give them a feel of the application on a mobile platform. Once they were comfortable, they were asked to navigate the map with the single-marker and the three-marker setup. The three-marker setup was fixed on a cube for easy handling. The order in which the interfaces were tested were key-based, single-marker and finally three-marker setup.

The participants were then asked to rate scrolling and zooming of the map with each interface independently a scale from 0 to 10 for intuitiveness, comfort and responsiveness. The average ratings of all participants is shown in Table 1.

Though the average ratings in Table 1 did not allow us to draw any conclusions, the individual ratings were wide-spread. Hence, we felt that rather than choosing an interface based on the above ratings, it would be wise to identify what led to the users decisions in making their ratings.

The ratings for the intuitiveness of the interfaces are quite similar for key-based and single-marker setup, while the three-marker setup suffered in zooming. This was expected, because the three-marker setup was designed with the accuracy and ease of the zooming operation in mind, which is reflected in the

**Table 1.** Average Ratings (scale of 0–10).

| | Key-based Interface | | Single-Marker Setup | | Three-Marker Setup | |
|---|---|---|---|---|---|---|
| | Scroll | Zoom | Scroll | Zoom | Scroll | Zoom |
| Intuitiveness | 7.3 | 7.3 | 7.3 | 7.1 | 7.3 | 6.4 |
| Comfort | 6.8 | 6.7 | 7.3 | 6.0 | 6.4 | 6.3 |
| Responsiveness | 6.2 | 6.3 | 6.6 | 5.9 | 6.5 | 6.3 |

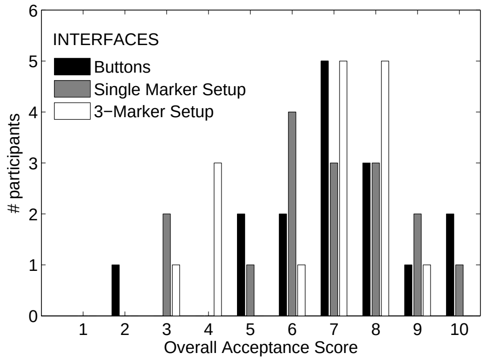

**Fig. 7.** Bar plot showing the distribution of overall acceptance scores given by the participants to each interface.

ratings for system's responsiveness and comfort in zooming with the single- and three-marker setups.

Clearly, the single-marker interface was much more comfortable than both the key-based and the three-marker setup for scrolling. Although the scrolling methodology was the same for the single- and the three-marker interfaces, we felt that the lower rating for the three-marker setup in terms of scrolling might have been influenced by the number of markers involved in the operation and the order of testing the interfaces. In their feedback, the participants who favored the single marker setup felt that it was more intuitive than the three marker setup and carrying one marker along with their phones was easier than carrying three markers on a cube.

On the other hand, people who favored the three-marker setup argued that the single-marker interface was too sensitive to use in the first attempts, especially during zooming, which took some getting used to. The ratings for the responsiveness of the zooming operation confirms this. Participants felt that human hands are not always steady, and they found it difficult initially when they had to scroll maps by moving the phone and at the same time hold it at a steady distance from the marker in order to maintain the zoom level. In comparison, they found the three-marker interface to be rather straightforward, as scrolling and zooming were handled by distinct markers, and it took less time to get used to it.

The preference ratings for the additional features we provided in the application, namely the video feedback window, map rotation with tilting the phone and the 'zoom-meter', are shown in Figure 8 as bar plots. People found especially the rotating window feature, which is not normally found in other mobile map navigation applications, to be very useful. As for the video feedback window and 'zoom-meter', the participants felt that they were indispensable to controlling the application and thus more of a requirement than a feature.

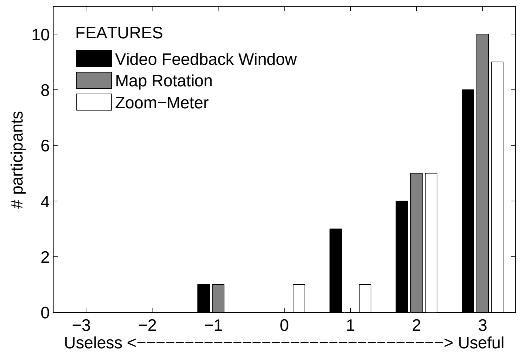

**Fig. 8.** Bar plot showing feature preference ratings for the video feedback window, map window rotation with tilting the phone, and the 'zoom-meter'.

The ratings of the participants were also heavily affected by their experience with such interfaces, especially in similar applications where user interaction is constrained by online downloading of data and network speed. People have used keys on the phone all along and when switching to markers, the more time they spent with marker-based interfaces, the more they got adapted to using them. Yet the scores for the key-based interface are not so high either, and people seemed to be ambivalent, which is also seen in Figure 7. Hence, we believe that with adaptation and experience, people would find virtual map navigation and other similar applications easier to handle with markers than with keys on the phone.

## 5 Conclusions

We proposed intuitive motion-based interfaces for an online virtual map navigation application. The application runs on mobile devices with a built-in camera, and the phone becomes a virtual looking glass over the map. The interface is based on pose estimation from fiducial markers extracted from the frames captured by the phones camera using ARToolkitPlus. We developed different user interfaces to control the application, and we conducted a user study to evaluate their accuracy, sensitivity and responsiveness. The results show that the motion-based interfaces are well appreciated for their intuitiveness and perform equally well when compared with a key-based interface even for a first trial. Adaptation and experience would certainly give our proposed interfaces a lift in performance and general acceptance in comparison to working with keys on the phone.

In spite of all the possibilities to make markers 'mobile', they still represent a considerable burden and inconvenience. Our future work focuses on enhancing the system with marker-less methods of detecting the phones pose and movement in 3D space with techniques like feature tracking and optical flow. We are also investigating applications for combining our map navigation techniques with a GPS device.

## References

1. Foxlin, E.: Motion tracking requirements and technologies. In Stanney, K.M., ed.: Handbook of Virtual Environment Technology. Lawrence Erlbaum Associates, Hillsdale, N.J., USA (2002) 163–210
2. Moehring, M., Lessig, C., Bimber, O.: Optical tracking and video see-through AR on consumer cell phones. In: Proc. Workshop on Virtual and Augmented Reality of the GI-Fachgruppe AR/VR. (2004) 193–204
3. Drab, S.A., Artner, N.M.: Motion detection as interaction technique for games & applications on mobile devices. In: Pervasive Mobile Interaction Devices (PERMID) Workshop, Munich, Germany (May 2005)
4. Winkler, S., Rangaswamy, K., Zhou, Z.: Intuitive user interface for mobile devices based on visual motion detection. In: Proc. SPIE Multimedia on Mobile Devices. Volume 6507., San Jose, CA, USA (January 2007)
5. Arhippainen, L., Rantakokko, T., Tähti, M.: Navigation with an adaptive mobile map-application: User experiences of gesture- and context-sensitiveness. In Murakami, H., Nakashima, H., Tokuda, H., Yasumura, M., eds.: UCS. Volume 3598 of Lecture Notes in Computer Science., Springer (2004) 62–73
6. Wagner, D., Schmalstieg, D.: ARToolkit on the PocketPC platform. In: Proceedings of the 2nd IEEE Intl. Augmented Reality Toolkit Workshop (ART032), Tokyo, Japan (October 2003) 14–15
7. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR), San Francisco, CA (1999) 85–94