A GPS Tracking Application with a Tilt- and Motion-Sensing Interface

Michael Mock[1] and Michael Rohs[2]

[1]Fraunhofer Institute for Intelligent Analysis- and Information Systems
Schloss Birlinghoven
53754 St. Augustin, Germany
michael.mock@iais.fraunhofer.de

[2]Deutsche Telekom Laboratories, TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany
michael.rohs@telekom.de

Abstract: Combining GPS tracks with semantic annotations is the basis for large data analysis tasks that give insight into the movement behavior of populations. In this paper, we present a first prototype implementation of a GPS tracking application that aims at subsuming GPS tracking and manual annotation on a standard mobile phone. The main purpose of this prototype is to investigate its usability, which is achieved by a tilt- and motion-sensing interface. We provide a GPS diary function that visualizes GPS trajectories on a map, allows annotating the trajectory, and navigating through the trajectory by moving and tilting the mobile phone. We present the design of our application and report on the very first user experiences.

1 Introduction

The analysis of the movement behavior of people can serve as the basis for the understanding of the habits of a city's or a country's population. In the AGMA project (financed by the Arbeitsgemeinschaft Media-Analyse e.V., ag.ma), Fraunhofer IAIS applies data-mining techniques to calculate the reach and gross contacts of poster networks based on GPS trajectories collected by over 30.000 test persons [AG07]. Currently, the test persons collect GPS trajectories with dedicated GPS devices, download them from the devices to a PC at home, and send them to a central collector, where they are manually entered into a database. A copy of this database is installed in the Fraunhofer IAIS computing cluster, where the highly computing intensive data-mining process is carried out [We08].

Beside information about the physical movements obtained by GPS devices, semantic information such as means of transportation or purpose of the trip is of high interest. Several research projects have investigated the automated derivation of semantic information from GPS trajectories, for example [Li07]. Our approach relies on the manual annotation of GPS trajectories in order to get training data for subsequent data-mining algorithms. So far, we have developed an (off-line) annotation tool that allows for visual browsing and annotating past GPS trajectories on a PC [Gu08].

In this paper, we present a first prototype implementation of a next-generation GPS tracking application that aims at subsuming GPS tracking and manual annotation on a standard mobile phone. The main purpose of this prototype is to investigate its usability and how and whether the test persons would accept it. Therefore, the user interface design was of main interest. We believe that a sensing-based interface can substantially contribute to achieving these design goals. We provide a GPS diary function that visualizes GPS trajectories on a map, allows annotating the trajectory, and allows navigating through the trajectory by moving and tilting the mobile phone. The idea of using tilting of a device as a means of input has been around for quite some time [Rek96,HP+00,CC+07]. Here, we specifically focus on continuous movement along GPS traces for revisiting past locations, defining split points between different sections, and annotating these sections.

2 System Design

This section describes the main aspects of the system and interface design. The system is implemented in Java 2 Microedition and has been tested on Nokia N95 mobile phones running Symbian S60. Although the N95 has a built-in GPS receiver, we used an external Nokia LD-3W Bluetooth GPS receiver for its better performance in terms of sensitivity to GPS signals and time to compute a position fix. The N95 contains a camera and a 3-axis accelerometer, which provide the raw data for the tilt- and motion-based interface. The access to these sensors is encapsulated in a separate process ("Sensor-Server") implemented in C++. This generic server makes the sensor data available via a local socket and can also be the used in other applications.

The core of the GPS tracking application consists of "GPS-Diary" and "GPS-Monitor." The GPS-Diary is the basic part of the application. It allows users to record, visualize, and annotate GPS trajectories. The Nokia sports tracker application [NO08] has similar goals, but (currently) does not visualize GPS trajectories on a map. Also, the Nokia sports tracker always records GPS tracks under a single label, whereas our application allows assigning different labels to different segments of a track. In particular, users can split the GPS trajectory into user-defined segments and can annotate the segments with user-defined labels. In addition, users can browse through the history of GPS tracks. The GPS-Monitor implements the client/server part of the GPS tracking application. It is intended to replace the central database repository for GPS tracks being analyzed in our computing cluster.

Figure 1 depicts the overall architecture and the individual components of the GPS tracking application. The GPS-Diary is a single process (Java Midlet) consisting of three different threads of control, whereby the main thread "User Interaction" starts and stops the other two threads according to the user commands. In particular, the thread "GPS

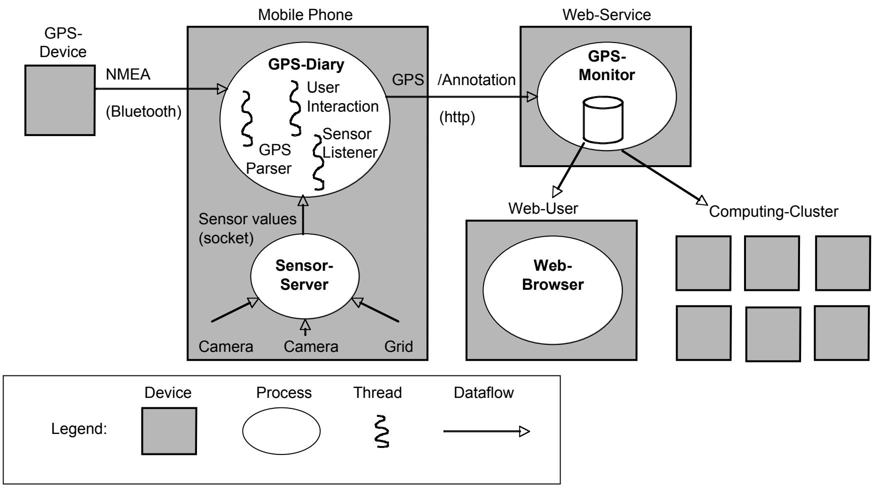


Figure 1: Overall architecture

Parser" runs while GPS tracks are recorded and the thread "Sensor Listener" runs while the user chooses to navigate a recorded track. The update rate depends on the selected sensor(s). For the accelerometer it is about 40 Hz, for the optical flow algorithm it is about 15 Hz. Regarding the interface to the user, the basic function of these components is to adjust the displayed map in such a way that the focus of interest is in the center of the mobile phone's display.

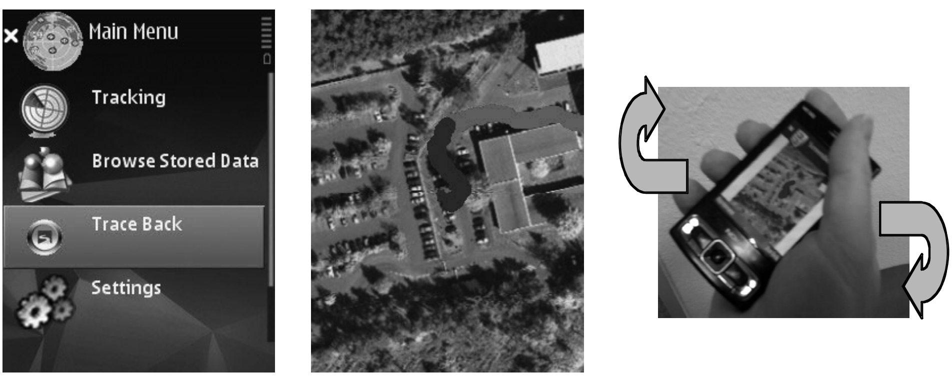
(a) Main menu (b) Tracking view (c) Mode view

Figure 2: GPS-Diary interface

Figure 2 shows the main interaction components of GPS-Diary that a user encounters when recording GPS tracks. In the main menu of the application, shown in Figure 2(a), the user selects "Tracking". Consequently, a tracking view as shown in Figure 2(b) will display the GPS track of the user on a picture of his/her current location. In our current implementation, the picture is taken from a static background image map of the area of our institute (originally taken from a free map of the city of Bonn under stadtplan.bonn.de). In general, image maps can be taken from cartographic material stored on the phone's mass media storage or from online image map sources. The color of the displayed GPS track encodes the mode with which the segment of the track is annotated. The user can change the mode as shown in Figure 2 (c), resulting in a different color to be used for the current GPS track segment.

The sensing part of the interface allows navigating through the GPS traces by moving or tilting the mobile phone as shown in Figure 3.

When the user selects "Trace Back" in the main menu of the application (Figure 3(a)), the last recorded GPS track, including the color-coding of the segments which represent the different annotation modes, will be displayed as shown in Figure 3(b). The user can virtually "walk back" on the trace by tilting the mobile as pointed out in Figure 3(b). Alternatively, the user can select the camera as the input device and use optical flow for trace navigation. Both alternatives will move the focus of interest, which is a point in time on the GPS track, (denoted as a red dot) forth and back on the GPS track. The display will adjust automatically such that the focus of interest remains in the center of the display. This navigation method is the basis for further extensions that will make it possible to insert new segment split points and changing or extending the annotation of a segment (as done in our PC-based implementation of the annotation tool [Gu08]).



(a) Activation (b) Trace view (c) Tangible interaction

Figure 3: The tilt- and motion-controlled interface

The technical basis for the trace navigation is the sensor server depicted in Figure 1. It provides a socket-based interface for accessing the raw sensor values of the current phone orientation based on the accelerometer and the moving direction of the phone based on the computed optical flow.

For each of the three axes of the Nokia N95 accelerometer, an acceleration of 1G (axis oriented towards gravity) produces a sensor value in the range from -56 to 56. There is considerable noise in the sensor readings (see Figure 4, left). Thus, the values are smoothed with a Savitzky-Golay filter. From the smoothed raw values the inclination is computed in degrees. For navigating through a GPS trace we only use inclination around the vertical display axis: Tilting left moves back in time on the GPS trace, tilting right moves forward. The scrolling rate is controlled by the angle of inclination. Users normally hold device tilted slightly upwards when viewing the display.

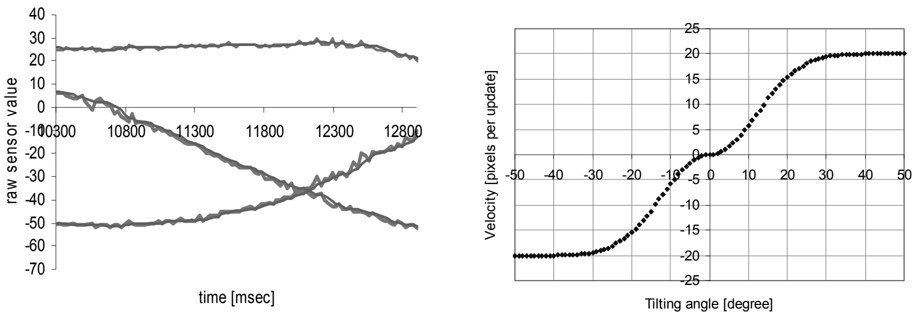


Figure 4. Smoothed sensor values (left). Mapping tilting angle to scrolling rate (right).

The horizontal display axis (x-axis of the accelerometer) is not affected by this normal reading inclination. Thus, we map the amount of tilting of the x-axis to movement along the GPS trace. The velocity of this rate-controlled interface is not mapped linearly, but is based on a double S-curve (see Figure 4, right). This allows for precise control and slow movement when the device is tilted slightly (around -5° to +5°) and fast scrolling further up the S-curve. Suitable parameters for the curve have been determined in extensive pilot testing. As an alternative to rate-controlled tilting input the user can also choose optical flow input [Ro04]. The algorithm computes the relative movement between successive camera frames and updates the position on the GPS trace accordingly. In order to ensure the predictability of the interface, the GPS trace is not just played back as recorded, but rather each update covers a distance that corresponds to the current scrolling speed. A simple playback of the trace positions would mean that the scrolling depends on the movement velocity when the trace was recorded. The scrolling might even stop at a point of zero velocity, which is clearly undesirable.

3 Conclusions and Further Work

The users have reported a slight preference for using the inertial sensor for map navigation rather than the camera-based optical-flow sensor. Although the sensors exhibit no performance difference regarding the responsiveness of the application, the inertial sensor works properly under a wider range of conditions. The camera-sensor depends on texture of the visible image. It turned out that horizontal movements of the mobile phone with the camera pointing towards the ground (for instance while walking outdoors) mostly did not generate sufficient or at least somewhat unpredictable optical flow. The camera-sensor worked fine when holding and moving the mobile phone vertically (probably due to the change of the horizon that induced a strong optical flow).

The GPS tracking application developed so far shows sufficient functionality for recording, annotating, and storing short trajectories (locally and on a central server). This allowed us to carry out the experiments regarding the user interface, but does not yet fulfill the requirements for GPS data collection in a large scale. Efficiency and security considerations regarding the storing and transmission of GPS trajectories will have to be taken into account. Regarding the interface, standard features known from commercial handheld GPS devices (e.g. the Garmin HC series), such as automatic orientation of the map, should be added.

Acknowledgments

We would like to thank the students of the B-IT (Bonn-Aachen International Center for Information Technology) summer lab 2008, carried out at Fraunhofer IAIS, for their support in the implementation of the GPS tracking application and their help in the interface evaluation: Sohail Akhtar, Senthilkumar Chandramohan, Amirhossein Jahanbekam, Jibran Javed, Masud Karim, Carolina Moreno, Atif Musaddaq, Ashkan Sharifi, Anisa Stefi, Alpesh Tarapara, Ramesh Turshani, Monica Verma, and Vina Wibowo.

References

[AG07] Arbeitsgemeinschaft Media-Analyse e.V.: Infos für Presse, Radio, TV und Online, Press Release 17.12.2007. www.agma-mmc.de (last access May 3, 2008).

[CC+07] Sung-Jung Cho, Changkyu Choi, Younghoon Sung, Kwanghyeon Lee, Yeun-Bae Kim, and Roderick Murray-Smith. Dynamics of Tilt-Based Browsing on Mobile Devices. In CHI ’07: CHI ’07 extended abstracts on Human factors in computing systems, pages 1947–1952, New York, NY, USA, 2007. ACM.

[Gu08] Baris Guc, Michael May, Yucel Saygin, and Christine Körner: Semantic Annotation of GPS Trajectories, 11th AGILE International Conference on Geographic Information Science, Girona, Spain, 2008.

[HP+00] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. Sensing Techniques for Mobile Interaction. In Proceedings of User Interface Software and Technology (UIST), pages 91–100. ACM Press, 2000.

[Li07] L. Liao, D.J. Patterson, D. Fox, and H. Kautz Learning and Inferring Transportation Routines, Artificial Intelligence, Volume 171 , Issue 5-6 (April 2007), Pages 311-331.

[NO08] research.nokia.com/research/projects/SportsTracker/ (last access May 3, 2008).

[Rek96] Jun Rekimoto. Tilting Operations for Small Screen Interfaces. In Proceedings of the 9th annual ACM symposium on User Interface Software and Technology, pages 167–168. ACM Press, 1996.

[Ro04] Michael Rohs: Real-World Interaction with Camera-Phones. Proceedings of the 2nd International Symposium on Ubiquitous Computing Systems (UCS), pp. 39-48, Tokyo, Japan, November 2004.

[We08] Dennis Wegener, Christine Körner, Michael May, and Michael Mock: Parallelization of R-programs with GridR in a GPS-trajectory mining application, submitted to the Ubiquitous Knowledge Discovery Workshop (UKD08), September 19, 2008 in Antwerp, Belgium in conjuction with ECML/PKDD 2008 Conferences.