GPS Waypoint Application

Kris Koiner, Haytham ElMiligi and Fayez Gebali
Department of Electrical and Computer Engineering
University of Victoria
Victoria, BC, Canada
Email: {kkoiner, haytham, fayez}@ece.uvic.ca

***Abstract*—In this paper, we present a software application that aims at translating data collected by a cell phone into voice signals to help the blind and visually impaired users navigate their way from a starting point to a destination point. This means that cell phone issues voice messages to guide the user along a predetermined path. The application utilizes existing hardware in cell phone devices, such as GPS, compass, etc. This project is designed to research and develop a framework for assistive technologies using Nokia mobile platform. The outcome is a software product, an application on Nokia cell phones, and the project has direct social benefits because it aids people who are visually impaired. An application was developed to show a proof-of-concept for personal trip navigation using a cell phone (Nokia-N8). Similar applications exist for driving, but this application attempts to offer a similar goal to all people's trips.**

I. Glossary

For consistency, The following nomenclature are used through this paper.

- Azimuth: the angle (in degrees) that the user (or phone) is facing, measured from magnetic north.
- Bearing: the angle (in degrees) between two points, typically between the current position and the next waypoint.
- Heading: the angle (in degrees) that the user (or phone) is moving towards, measured from magnetic north.
- Trip: a collection of waypoints that the user follows in order from one destination to another.
- Waypoint: a single set of latitude, longitude, and azimuth that acts as a physical marker.

II. Introduction

While there are many assistive technology products on the markets today benefiting blind and visually impaired people, more products need to be developed to help them practice their life easily. In this paper, we develop an innovative product that provides visually impaired people with a greater understanding of their environment, thereby improving the quality of their lives. The most interesting point is that we don't have to develop a new hardware device to do this job. Instead, we utilize the resources that already exist in most cell-phones to perform the required tasks. We are introduction a new software application developed on Nokia N-8 to help the blind and visually impaired users. This application was developed using python language on Symbian, which is an open source operating system currently used in Nokia Mobile Devices and is designed to make minimal demands on batteries and to have low memory. Since Nokia has announced that Windows Phone 7 will become the company's primary smartphone platform, the developed python code is well documented and is designed so that it can be easily converted, with a minimum effort, to other programming languages.

This paper is organized as follows. Section III highlights similar research work. Section IV lists the application features. Section V discusses the application design and modes of operation. Section VI highlights the application coding. Section VII shows the performance evaluation of the developed application and lists the limitations. Finally, we draw our conclusion in Section VIII.

III. Background

Several software development work has been done on cell phone platforms to address health care applications [1]. A literature survey shows several applications such as aiding those with Alzheimer's disease [2], developing a socially assistive mobile home robot companion for the elderly with mild cognitive impairment [3], and supporting employment for people with severe mental illness [4]. There are also few accessibility applications from various vendors that are worth mentioning such as TalkBack [5], Spiel [6], Voice Readouts [7], and VoiceOver [8]. TalkBack is one of the official accessibility services provided by Google. It uses synthesized speech to describe the results of actions, such as moving to a new control with the directional pad or clicking on a control, and events such as a notification or incoming call [5]. Spiel is an accessibility service that provides an alternative to TalkBack. It also uses synthesized speech to describe actions and events, but it has its own rules for speaking that may differ from TalkBack's [6]. Voice Readouts is the Motorola accessibility service that is analogous to TalkBack and Spiel. It also uses synthesized speech to describe actions and events according to a set of rules [7]. VoiceOver is another application implemented on iphone. VoiceOver only allows users to control the phone using simple gestures that let the user physically interact with items on screen. With VoiceOver, users just touch the screen to hear a description of the item under their finger, then gesture with a double-tap, drag, or flick to control the phone [8]. Our proposed project is totally different since we aim at translating data collected by a cell phone into voice signals to help the blind and visually impaired users navigate their way from one place to another.

IV. Application Features

The application has the following user functionalities.

- Obtain GPS information at a regular interval.
- Obtain magnetic compass readings at a regular interval.
- Read and parses a text database for multiple trips and waypoints.
- User-selectable trip and waypoint.
- Compute the distance and bearing from the user's current position to the next waypoint.
- Auto-advance waypoints as the user reaches them.
- Using text-to-speech, speak instructions relating the distance and bearing of the next waypoint.
- Speak instructions of future actions and notes when approaching a waypoint.
- Allow the user to take a picture, storing date, time, position, and azimuth.

In addition to the above features, the application also has the following administrative functionalities.

- Create new trips.
- Create and delete waypoints.
- Update waypoint location and orientation, and attach specific notes.
- Store modified trip and waypoint information back to the database.

V. Application Description

The application is split into three main modes of operation: trip, raw data, and edit mode.

A. Trip Mode

Trip operations are by default shown to the user and are immediately available upon starting the application. As shown in Figure 1, the trip screen shows the current trip, the current waypoint, the distance and bearing to the waypoint, the audible instruction and any note associated with the waypoint.

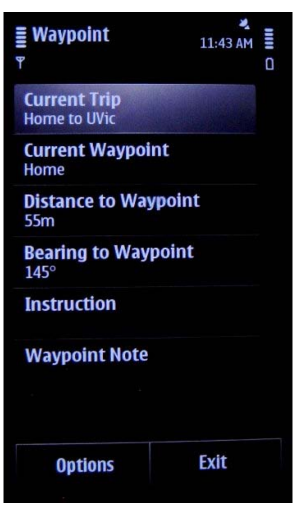

Fig. 1. Trip screen.

The current trip can be selected by pressing the ***current trip*** field and selecting a new trip from the list. Trips are named according to their start and end waypoints. When a new trip is selected, the current waypoint is set to the first waypoint of the trip.

The current waypoint automatically advances to the next waypoint in the trip when the user is within 2m of the waypoint, however a different waypoint can be chosen at any time by pressing the ***current waypoint*** field and selecting from the list.

The distance and bearing to the current waypoint are calculated from the user's current position in meters and degrees respectively.

The active instruction is displayed in part in the ***instruction*** field. Pressing it will open a dialog containing the entire instruction. In addition, the instruction is spoken every 5 seconds. The instruction is created from the distance and bearing to the current waypoint. If the user is nearing the waypoint, within 30m, the instruction also contains information about the upcoming waypoint, such as custom notes or actions to be taken when reached. Figure 2 shows a sample instruction.

Fig. 2. Typical instruction.

The waypoint note displays any custom notes associated with the current waypoint. Pressing it will open a dialog containing the full note.

The user can take a picture at any time by pressing ***menu*** and ***take picture***. This takes a picture using the rear-facing camera within approximately the next second[1] and saves the image to the phone. The filename is determined by the current date and time as well as the user's latitude, longitude, and azimuth. This picture is not shown and the user is informed that the image is saved. Figure 3 shows a sample message displayed.

B. Raw Data Mode

Raw GPS and sensor data can be viewed by pressing ***menu*** and ***show raw data***. To return to the trip view, press ***menu*** and

[1]note that time is subject to the camera interface response time.

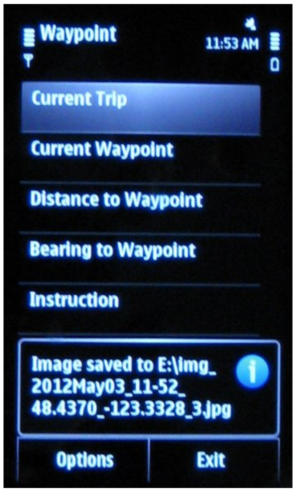

Fig. 3. Taking a picture.

show trip. The raw data view shows the raw collected data from the GPS and sensors, shown in Figure 4. When initially starting up, many of the fields will be (***...***) while a reading is being acquired. After which, the data will be shown and refreshed every second.

Fig. 4. Raw data view.

The ***GPS*** field shows the number of satellites available and which GPS module is being used. Latitude, longitude, speed, and heading are displayed along with their accuracy. The ***compass*** field shows the phone's azimuth in degrees, measured from magnetic north.

The ***magnetic*** field shows the x, y, and z readings of the magnetometer as well as the calibration level between 0 and 3. If the value is less than 3, the sensor can be calibrated by slowly rotating the phone in different directions through its entire range of motion. This allows the sensor to detect the orientation of the earth's magnetic field, becoming more calibrated. Depending on the environment, this process can take up to 30 seconds of rotating.

C. Edit Mode

Edit mode allows the user to create new trips or edit existing trips by adding, editing, or removing waypoints. To create a new trip, press ***menu*** and ***create new trip***. A new trip will be created with an initial waypoint. To edit an existing trip, ensure the trip is the currently trip and press ***menu*** and ***edit trip***. Figure 5 shows the edit waypoint menu presented.

Fig. 5. Edit waypoint menu.

The waypoint menu displays a list of all waypoints within the trip. Each waypoint can be selected to show a list of options, as shown in Figure 6: rename, set to current location, set note, insert new before, insert new after, and delete.

Fig. 6. Waypoint editing options.

Rename renames the waypoint. Names should be concise since the trip name is automatically generated from the names of the first and last waypoints of the trip. ***Set to current location*** updates the latitude, longitude, and azimuth of the waypoint to the current location. ***Set note*** sets any notes associated with the waypoint.

Insert New Before and ***Insert New After*** create a new waypoint, setting it to the current location and ordering it either before or after the selected waypoint. The newly created waypoint can then be selected and renamed similarly to other waypoints. ***Delete*** removes the selected waypoint after confirmation.

After editing is complete, press ***Back*** to return to the trip view. A dialog will appear, as shown in Figure 7, prompting to save the changes. If the user selects ***yes***, the database will be updated, otherwise the changes will be lost.

Fig. 7. Leaving edit mode.

VI. Python Coding

The application is written in Python, using the PyS60 – Python for Symbian 6.0 – libraries. The majority of functionality is provided by these libraries, such as GPS, magnetometer and compass, text-to-speech, and camera interfacing.

A. Database

The database is a custom design, stored in a simple flat-file text database. Each record is one line and fields are separated by tabs. This is done so that the database is easily parsable and modifiable by third-party applications or by the user themselves. Each line describes a single waypoint and has the format described in the following Database record structure code.

```
- Type, integer, For future use, must be 10.
- TripID, integer, A unique trip ID to
  associate waypoints
- Latitude, float, Standard latitude
  expressed in decimal degrees
- Longitude,float, Standard longitude
  expressed in decimal degrees
- Azimuth, float, 0 to 360 degrees,
  measured from magnetic north
- Name, string, Waypoint name
- Notes, string, Any waypoint notes
```

The ***Type*** field is designed for future use as it may be expanded to allow the database to hold additional information. Currently, each waypoint has a type of 10 and all other values are ignored.

The ***Trip ID*** field associates the waypoint to a particular trip. Waypoints within the same trip are parsed and organized in the same order as they appear in the database. Given the dictionary implementation, the contents of the trip ID field are not restricted to integers. However, for simplicity and consistency, trip ID values are integers starting at 0 and incrementing for each new trip.

Latitude, longitude, and ***azimuth*** are the three waypoint parameters, measured in decimal degrees. Latitude and longitude are in the range -180 to 180, and azimuth is in the 0 to 360 range, although the code is designed not to fail if a different range is used for the azimuth.

The waypoint ***name*** and ***notes*** are the only textual information and should not contain any tab or newline characters as these would interfere with the database parsing.

Comments and other lines may be added to the database without breaking the application, but will be removed if the application saves the database, such as when leaving the edit mode.

B. Updating Information

The GPS module runs an update function whenever it acquires new data, typically every second. This function reads all the GPS data and stores it locally for processing. If certain values are not available, reasonable defaults are assumed. Sensor data is read at the same time and stored similarly.

After reading new data, all other processing functions are run, updating position and waypoint information as well as generating new instructions.

Information is printed to the screen every half-second in a separate execution thread.

VII. Performance Evaluation

Throughout development and testing, the GPS and compass modules were found not to be as accurate as desired. This introduced some limitations to the attainable resolution in determining the user's position.

A. GPS

When implementing the GPS data collection, it was assumed that coordinate data would be accurate enough to reliably position and direct a user, equating to roughly +/- 3m as an absolute maximum. While GPS technology can be accurate to the centimeter range [9], the US Department of Defense regulates civilian GPS technology to be low-speed and low-accuracy devices.

In our application, we used only the internal GPS in the N8 device. Assisted and bluetooth GPS were not used. Our experiments show that the built-in GPS receiver in the Nokia phone is accurate to a minimum of +/- 10m, although this accuracy can only be attained under certain conditions. Upon start up and indoor testing, accuracies can be as low as +/-

300m, and varies widely and quickly. Outdoors, accuracies are typically under 100m.

The GPS operates best when in motion. Even at a slow walk of 0.8m/s, the GPS can quickly converge and maintain a 10m accuracy. When inside a vehicle, accuracies can drift due to the vehicle's structural interference. The build-in GPS unit maintains accuracies for all of it's fields. This can be used to selectively filter GPS data.

In addition, the GPS unit also maintains derivative data: heading and speed. Like position data, it is most accurate when the device is in motion, but is found to be much more accurate in determining heading than the magnetic compass.

B. Compass

The magnetic compass is a derived sensor from the phone's magnetometer. The magnetometer is a 3-axis sensor that senses the magnetic field in each of the axis. The compass then uses this data along with device's orientation to compute the users azimuth, relative to magnetic north.

Although the compass updates much faster then the GPS unit, surrounding metals can alter the readings. Such conditions exist indoors, in vehicles, and especially near sound equipment.

When testing the compass, values constantly varied, making it very difficult to rely on this data. In addition, the compass would often get stuck and fail to read anything or read the same value over and over again. Investigating the underlying magnetometer, a calibration flag was found that indicated, on a scale from 0 to 3, the estimated accuracy of the magnetometer.

This information is used to judge the accuracy of the compass as well as alert the user of required calibration. Moving the phone in various patterns, such as a figure 8, allows the magnetometer to calibrate and obtain more accurate results. This resulted in much better readings, and although still easily influenced, the compass would no longer get stuck.

VIII. Conclusion

In conclusion, the application demonstrated that it could be used to assist users in navigating from one destination to another. However, the accuracy provided by the GPS and compass interfaces could be improved by using the assisted and bluetooth GPS as well as the internal GPS.

To further improve the application, image processing could be also used to identify paths and obstacles to the user. This would allow the GPS to track the user's path and offer general direction, but maintain a more accurate view of the user's environment. In addition, this would allow the user to interact with their environment in real-time, a critical task when street lights, cars, and other moving objects are involved.

GPS and other sensor data could be improved by implementing filters to further reduce the noise generated. If further assumptions could be used, such as the user always moves within a certain speed range, than erroneous data could be more easily identified and removed.

External GPS modules could also be used to increase the accuracy of the GPS. These could be on the user's person and connect wirelessly to the phone. Using third party GPS receivers could employ additional processing and GPS features, such as Wide Area Augmentation System (WAAS).

Acknowledgment

The authors would like to thank Nokia Product Ltd. for funding this project. This project is funded under grant 2H11-7391.

References

[1] H. Hile, A. Liu, G. Borriello, R. Grzeszczuk, R. Vedantham, and J. Kos? andecka, "Visual navigation for mobile devices," *IEEE Multimedia*, vol. 17, no. 2, pp. 16 –25, april-june 2010.

[2] M. Donnelly, C. Nugent, S. McClean, B. Scotney, S. Mason, P. Passmore, and D. Craig, "A mobile multimedia technology to aid those with alzheimer's disease," *IEEE Multimedia*, vol. 17, no. 2, pp. 42 –51, april-june 2010.

[3] H.-M. Gross, C. Schroeter, S. Mueller, M. Volkhardt, E. Einhorn, A. Bley, C. Martin, T. Langner, and M. Merten, "Progress in developing a socially assistive mobile home robot companion for the elderly with mild cognitive impairment," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, sept. 2011, pp. 2430 –2437.

[4] Y.-J. Chang, H.-H. Liu, and T.-Y. Wang, "Mobile social assistive technology: A case study in supported employment for people with severe mental illness," in *Third International Conference on Convergence and Hybrid Information Technology (ICCIT '08)*, vol. 1, nov. 2008, pp. 442 –447.

[5] (2012, May) Talkback. [Online]. Available: http://www.androlib.com/android.application.com-google-android-marvin-talkback-zwxB.aspx

[6] (2012, May) Spiel. [Online]. Available: https://market.android.com/details?id=info.spielproject.spiel

[7] (2012, May) Voice readouts. [Online]. Available: http://responsibility.motorola.com/index.php/consumers/accessibility

[8] (2012, May) Voiceover. [Online]. Available: http://www.apple.com/accessibility/voiceover/

[9] Kowoma. (2012, May) Achievable accuracy. [Online]. Available: http://www.kowoma.de/en/gps/accuracy.htm