

Guiding Visually Impaired People with NXT Robot through an Android Mobile Application

Dalal M. Alghasra¹, Hesham Y. Saeed²

¹School of Engineering and Computing Sciences, New York Institute of Technology, Kingdom of Bahrain

²Scientific Publishing Center, University of Bahrain, Kingdom of Bahrain

E-mail addresses: dalal.ghasra@gmail.com, hesham.s3eed@gmail.com

Received 14 Mar. 2013, Revised 30 May 2013, Accepted 20 Aug. 2013, Published 1 Sep. 2013

Abstract: This paper introduces a mobile application to guide visually impaired people in transportation using NXT robot to make their life easier in roads and sidewalks. Nowadays, there is a wide variety of applications on the mobiles GPS for guiding and tracking. We have used mobile GPS in our Android mobile application with the help of voice recognition feature to collect the needed coordinates to navigate the user by letting him to communicate with the mobile application and speaks his destination. NXT robot start moving after getting the needed coordinates from the mobile application without crashing because of the obstacle avoidance feature implemented in the robot program. The robot communicates with the mobile application using Bluetooth.

Keywords: NXT; Robotics; Android; Mobile; GPS coordinates; Maps; Visually Impaired.

I. INTRODUCTION

Visually impaired people always have issues in transportation. Before the technology arrives, blinds were using guiding dogs in order to transport within different places. Nowadays, there are many devices help the blinds that act as a guiding dog. Most of the other researches similar to ours provide indoor assistance for visually impaired people [1-3] while our research provides outdoor assistance. Obstacles avoidance for visually impaired people has been discussed in wide area [4-6] which is very important element and the robot we used have ultrasonic sensor to provide it. In this paper we provide LEGO Mindstorms NXT robot in order to replace the guiding dog. NXT is the programmable board which acts like the mind of the robot. The NXT robot is programmed in an intelligent way to guide and help the blinds while they are moving. The NXT robot is guided through Android mobile application which is user-friendly for blind users because it uses voice recognition and instructions. Fig.1 illustrates the block diagram of the system.

II. ANDROID

Android [7-9] is an open-source software stack for mobile devices that includes an operating system,

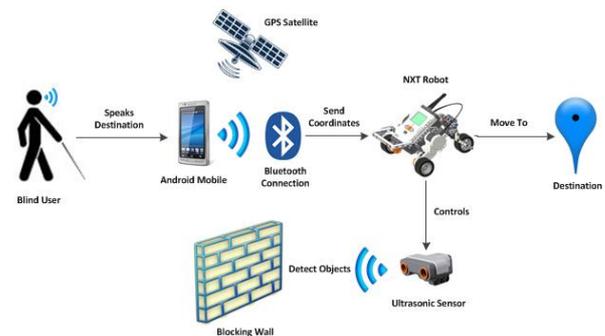


Fig. 1. The system's Block Diagram.

middleware and key applications. Android OS is Linux-based Operating System made for smart phones or tablets. It is developed by the Open Handset Alliance led by Google. Many mobile phones manufacturers use the Android operating system for their products such as Samsung, HTC, Sony, LG and Motorola.

In order to develop an Android application we need:

- Java Development Kit (JDK).
- Android Software Development Kit (SDK).
- Eclipse, which is an integrated development environment (IDE).
- Android Development Tools (ADT) plug-in.

- Android Virtual Machine (Emulator) or a real Android handset.

A. Main parts of Android application

The application contains three main parts:

- Source code which consists of a main class called MainActivity with other subclasses that are used for different sub functions inside the main class.
- UI layout called "main.xml" which consists of Buttons, Labels and MapView which is the layout object of the map.
- AndroidManifest.xml which is used to define the structure of the application and to declare permissions for some of the functionalities used within the application.

B. The reason of using Android

We have preferred Android OS over other operating systems because NXT is easier to interface with Android. Also, NXT and Android both programmed with Java which is a huge library and has flexibility in functionality. In addition, Android tools, documentations and best practices are easily available.

III. ANDROID MOBILE APPLICATION

The mobile application is an embedded Google Map which uses GPS to get the current location of the user and let the user choose his destination. The application will draw a route between two points and generate GPS coordinates of the waypoints of this route in order to provide them for the NXT robot. Since this application is designed for blind users, it uses voice recognition to get commands from the user and give voice instructions for the blind user. SM Chart for the Android application is shown in Fig.2.

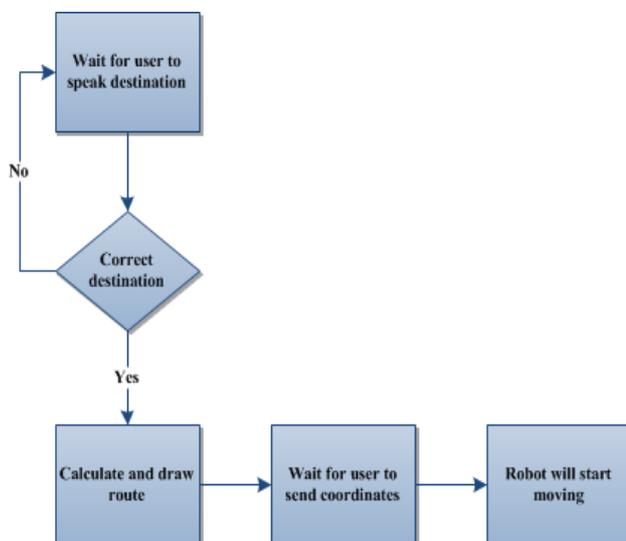


Fig. 2. SM Chart for the Android application.

A. GPS with Google Maps

The application uses the GPS [10] integrated within the Android mobile in order to get the current location and draw a marker for it over the Google Map using Satellite View as shown in the Fig.3. To use Internet and GPS, we need to declare permissions in the Manifest file of the application. The Android SDK consists of a general class called LocationManager and a listener called LocationListener which we will be using it to get the current location of the user. Google provides Android SDK dedicated for Google Maps API and the main class to create the Google Maps is called MapActivity. To use Google Maps API, we need to get an API key from Google in order to get the Satellite images.

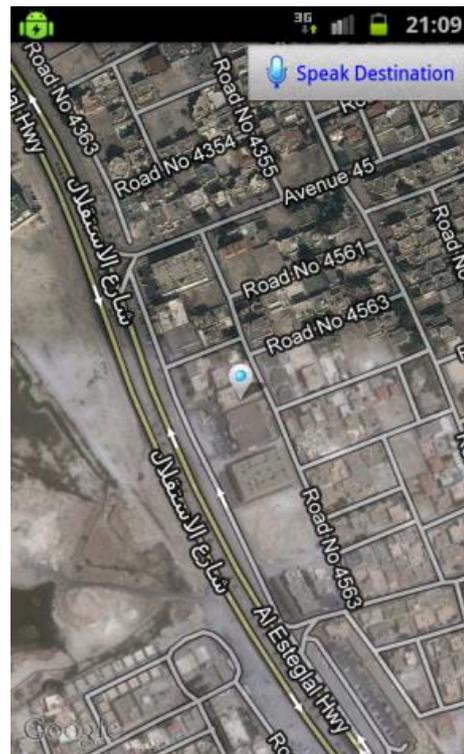


Fig. 3. Current Location Marker.

B. Calculate the Route

The route is calculated between the current location and the destination spoken by the blind user. This route is generated using Google Directions JSON API for walking instructions. The result from the API will consist of many coordinates through the route but we will take only seven waypoints of the coordinates and they will be drawn on the map. After getting the result and drawing the route, the map will span the zoom over those waypoints. These seven waypoints will be saved in an array to send them to the NXT robot. Fig.4 illustrates how the process of calculating and drawing the route happens in the application.

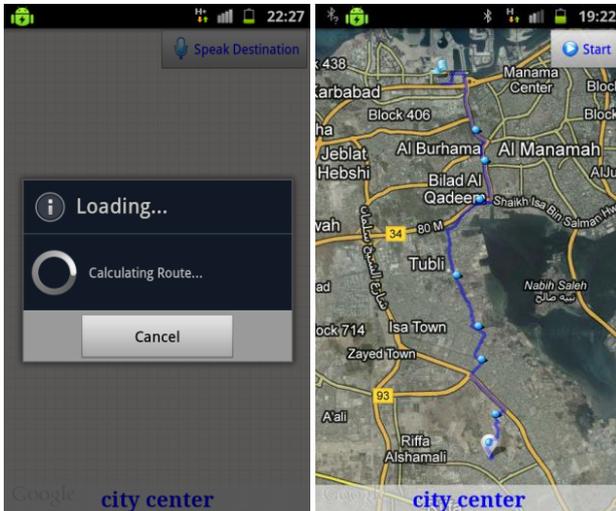


Fig. 4. Calculating and drawing the route.

C. Voice recognition and instructions

The mobile application uses the built-in voice recognition API to get commands from the blind user without having to type any text on his device. Once the application started, the user will hear voice instruction to tap on the top right edge of the screen in order to speak the destination. After that, the application will convert the spoken words to text which will be used as address for ReverseGeocoding method. ReverseGeocoding is the way of converting an address to coordinates on the map. The user will get another voice instruction to tap on the top right edge of the screen to send the coordinates for the NXT robot in order to guide it. In case the spoken destination was not understandable using ReverseGeocoding, the application will notify the user using voice instructions by saying it is a wrong address. Fig.5 shows the voice recognition part of the application in both situations voice recognized and no matches found.

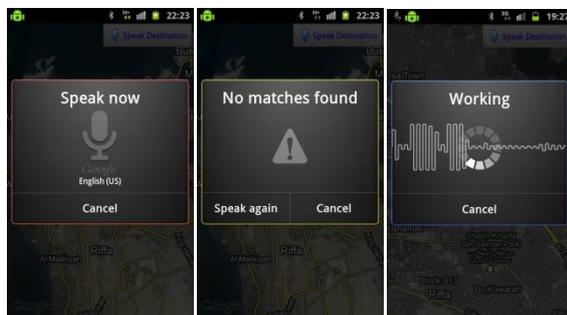


Fig. 5. Voice recognition in Android application.

IV. NXT ROBOT

Mindstorm Lego contains many parts with different shapes and sizes. The main part is called NXT controller. This is the mind of the device and the one we are programming and controlling. The robot has three servo motors used for movement. We can control the speed, distance and rotation of these motors which are attached

to three output ports A, B, and C. The NXT Robot is shown in Fig.6.



Fig. 6. NXT Robot.

The robot has sensors each with different functions which are sound sensor, ultrasonic sensor, touch sensor, and light sensor and these sensors attached to four input ports. We are considering only ultrasonic sensor in this paper. The ultrasonic sensor [11] is the sensor that gives the robot vision and makes it enables to see and detect objects. The robot can use it to avoid collisions with any obstacle in front of it by sensing and measuring distance.

LeJOS Firmware

LeJOS [12] is a firmware used for the Lego Mindstorms NXT. It replaces the original LabView firmware in the NXT brick. It includes a Java virtual machine, so it allows Lego Mindstorms robots to be programmed in Java programming language. It consists of a library of Java classes that implement the LeJOS NXJ API Fig.7 shows LeJOS firmware in the NXT screen.



Fig. 7. LeJOS firmware in NXT brick screen.

We are using LeJOS Eclipse plug-in since we are using Eclipse for programming the NXT. LeJOS plug-in used to create NXT projects for writing Java programs that run on the NXT itself and PC or mobile projects for writing Java program that remote control the NXT via USB or Bluetooth. In our case the NXT project receives the coordinates and move according to them .We needed NXT project for Android mobile to send coordinates from Android application to NXT robot.

V. NXT MOVEMENT

When the NXT receives the coordinates from the Android application[13], it saves them and starts moving after calculating the distance and angle between each pair of them. The below SM chart in Fig.8 gives us an overview of how the movement is handled by NXT robot.

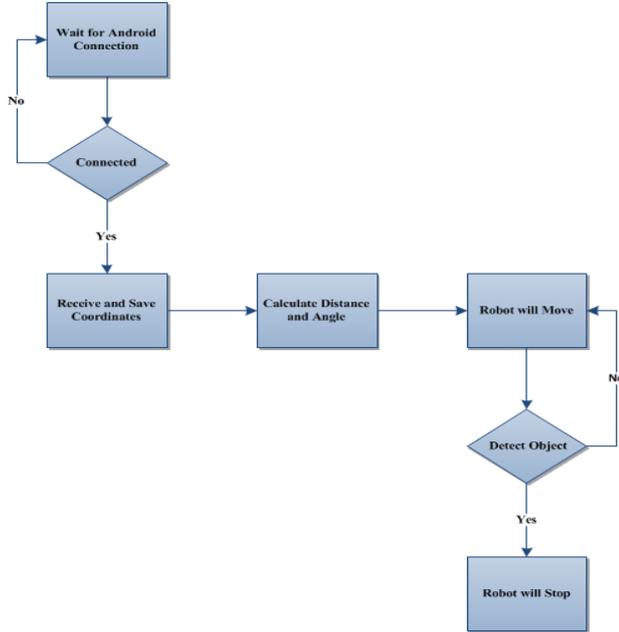


Fig. 8. SM Chart for the NXT movement.

A. Calculating distance between two points

To calculate the distance between two GeoPoints Haversine formula is used.

Haversine[14]:

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2(\Delta\lambda/2) \quad (1)$$

Formula:

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (2)$$

The Haversine formula is an important equation in navigation to calculate the shortest great-circle distance over the earth's surface for a given longitudes and latitudes.

$$\text{distance} = R \cdot c \quad (3)$$

Where **R** is the earth's radius and it is equal to 6,371 km. $\Delta\phi$ is the difference of longitudes between user location and the destination ($\phi_2 - \phi_1$) in radians.

$\Delta\lambda$ is the difference of latitudes between user location and the destination ($\lambda_2 - \lambda_1$) in radians.

Using this formula distance in equation (3) is the distance between the two points (λ_1, ϕ_1) and (λ_2, ϕ_2) in km.

B. Calculating Angle between two points

The NXT itself doesn't have GPS so having the distance is not enough and the NXT need to know the heading angle to move from one point to another. To calculate angle between two GeoPoints the following formula is used:

$$\theta = \text{atan2}(\sin(\Delta\lambda) \cdot \cos(\phi_2), \cos(\phi_1) \cdot \sin(\phi_2) - \sin(\phi_1) \cdot \cos(\phi_2) \cdot \cos(\Delta\lambda)) \quad (4)$$

$$y = \sin(\Delta\phi) * \cos(\lambda_2) \quad (5)$$

$$x = \cos(\lambda_1) * \sin(\lambda_2) - \sin(\lambda_1) * \cos(\lambda_2) * \cos(\Delta\phi) \quad (6)$$

$$\text{Heading} = \text{atan2}(y, x) \quad (7)$$

Using equation (7), we calculated the positive angle between the two points (λ_1, ϕ_1) and (λ_2, ϕ_2). To decide whether the NXT need to turn left or right we check the value of the angle if it is less than 180 the motor will rotate to the right with the positive heading value otherwise the angle will be ($360 - \text{given angle}$) and we will take the negative value for left heading. Fig.9 illustrates NXT screen with current and destination coordinates and the calculated Distance and Angle between them.



Fig. 9. NXT screen with the result data on each point.

C. Motors movement

The NXT uses motors B and C for moving and rotating. DifferentialPilot class is used to control these two motors. We also set the speed of the motors with this class. The DifferentialPilot control two functions of the motors, rotate in place and travel in a straight line. Travel function use the calculated distance between two GeoPoints to move the NXT motors with the given speed. The rotate function uses the calculated angle to control the direction of rotation of the NXT motors then continue moving in straight line using the travel function.

D. Object detection

The NXT has object detection capability. It can detect any obstacle in its path in less than 50 cm. The NXT uses Object Detector listener with a Feature Detector that detects objects using ultrasonic sensor and then take an appropriate action which is in this case stop moving. Feature Detector allow automatic scanning and retrieve of data every 500 ms. It keeps retrieving range of the detected object's distances as shown in Fig.10.



Fig. 10. Different values of different ranges of detected objects.

VI. COMMUNICATION BETWEEN ANDROID AND NXT

A. Bluetooth

The NXT robot opens Bluetooth connection [15] acting as receiver and wait for data to be sent by the mobile application. The mobile application establishes a connection acting as sender and sends one array of GeoPoints where GeoPoints consist of Latitude and Longitude of each point.

B. Method

To send and receive via Bluetooth we have used a known Java classes, `DataOutputStream` and `DataInputStream`. These two classes can open a channel between the sender and receiver in order to send data between them. Once the GeoPoints array received on the NXT robot we will calculate the distance and angle between each two elements of the array in order to use them for guiding the robot to move towards the right direction. Fig.11 illustrates the NXT screen while it is waiting to receive the coordinates and Fig.12 shows the Android application while sending the coordinates by Bluetooth.



Fig. 11. NXT is waiting for Bluetooth data.



Fig. 12. Sending Coordinates to NXT via Bluetooth after clicking on Start button.

VII. CONCLUSION

This paper has shown the successful communication between the mobile application and the NXT robot in order to help the visually impaired people in their movement. The mobile application is optimized to provide two way communication with the blind user

using voice recognition. NXT robot moves in smooth way to reach the user destination sensing any objects with the ultrasonic sensor and avoid collisions. Finally, we created a replacement for the guiding dog by a guided robot.

REFERENCES

- [1] V. Kulyukin, C. Gharpure, J. Nicholson, and G. Osborne, "Robot-assisted wayfinding for the visually impaired in structured indoor environments," *Autonomous Robots*, vol. 21, pp. 29-41, 2006.
- [2] V. Kulyukin, C. Gharpure, J. Nicholson, and S. Pavithran, "RFID in robot-assisted indoor navigation for the visually impaired," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004, pp. 1979-1984.
- [3] V. Kulyukin, C. Gharpure, and J. Nicholson, "Robocart: Toward robot-assisted navigation of grocery stores by the visually impaired," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 2845-2850.
- [4] W. M. El-Medany, A. M. Abuaesh, and N. M. Abuaesh, "An efficient car like-mobile robot design," in *EUROCON 2009, EUROCON '09. IEEE*, 2009, pp. 996-1001.
- [5] S. Shoal, I. Ulrich, and J. Borenstein, "Computerized obstacle avoidance systems for the blind and visually impaired," 2000.
- [6] I. Ulrich and J. Borenstein, "The GuideCane-applying mobile robot technologies to assist the visually impaired," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 31, pp. 131-136, 2001.
- [7] A. Developers, "What is android?," <http://developer.android.com/guide/basics/what-is-android.html>, vol. 2, 2011.
- [8] M. L. Murphy, *The Busy Coder's Guide to Advanced Android Development*: CommonsWare, LLC, 2009.
- [9] R. Meier, *Professional Android 4 application development*: Wrox, 2012.
- [10] W. El-Medany, A. Al-Omary, R. Al-Hakim, S. Al-Irhayim, and M. Nusaf, "A Cost Effective Real-Time Tracking System Prototype Using Integrated GPS/GPRS Module," in *Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on*, 2010, pp. 521-525.
- [11] P. Veelaert and W. Bogaerts, "Ultrasonic potential field sensor for obstacle avoidance," *Robotics and Automation, IEEE Transactions on*, vol. 15, pp. 774-779, 1999.
- [12] J. A. B. Moral, "Develop lejos programs step by step," *Version 0.4, July*, 2008.

- [13] S. Göbel, R. Jubeh, S.-L. Raesch, and A. Zündorf, "Using the Android Platform to control Robots," *Kassel University Germany*. [Online]. Available www.innoc.at/fileadmin/user_upload/_temp_/RiE/Proceedings/65.pdf.
- [14] C. Robusto, "The cosine-haversine formula," *The American Mathematical Monthly*, vol. 64, pp. 38-40, 1957.
- [15] S. Bluetooth, "Specification of the Bluetooth System, version 1.1," <http://www.bluetooth.com>, 2001.



Dalal M. Alghasra received her B.Sc in Computer Engineering from University of Bahrain, 2010 and hold MSc degree with honors in Electrical and Computer Engineering from New York Institute of Technology, Bahrain, 2013. She is currently working in YK Almoayyed Company as an IT Engineer. Her current areas of interest are Robotics and Mobile Apps.



Hesham Y. Saeed received B.Sc. in Computer Engineering at University of Bahrain at 2012. He is currently working in Scientific Publishing Center at University of Bahrain as Systems Developer & Journals Manager. He also works in Mobile Research Lab at University of Bahrain as Mobile Developer, He has developed multiple sophisticated Android mobile applications for Bahrain eGovernment Authority such

as NGO Directory, eService Centers Locator and Health Services.