Dynamic obstacle mapping for the visually impaired using sensor fusion.

Johann Thor Kristthorsson, University College London
Ifeanyi Ndu, University College London
Veselin Pavlov, University College London
Shuang Zhang, University College London

1. PROJECT CONTEXT AND OBJECTIVES

The Lighthouse team collaborated with Microsoft and the Guide Dog Associations to produce applications which can improve the living experience for the visually impaired users. Since few environments and applications were designed for the huge amount of blind and partly sighted individuals, it is difficult for them to moving around especially in an unfamiliar environment and without others' assistance.

Microsoft is both the sponsor and the client in this project. The particular field of study that the client wanted the team to explore was using wearable sensors which could be utilised by the visually impaired. Also, the client wanted low-cost, off-of-shelf hardware in this project. After analysing the requirements from the client, the Lighthouse team decided to find a way to improve the experience of visually impaired when entering an unfamiliar indoor environment.

Finally, Lighthouse team created a platform for dynamic indoor obstacles mapping, which included an Obstacle API, a processing platform and an Android application to give user feedback. After testing, the quality of the project meet client's standards.

2. ACHIEVEMENTS

2.1. Obstacle API

Obstacle API in this project provided communication between sensors and processing platform. It was written in Java and implemented by a server which listened on a UDP socket.

2.2. Processing platform

The processing platform was based on Spark and it computed the data collected from sensors. It can not only compute the data but also reduce the error by filters.

2.3. Android application

An android application was created to show user the details of surroundings by querying the database. It showed the nearby obstacles on the map of the environment.

2.4. Others

The MSc CS students in this team worked on collecting data from environment and detecting obstacles by sensors. They had their Android applications which was supposed to send data to the Lighthouse platform.

3. EVALUATION

3.1. Testing strategy

Unit testing, load testing and integration testing were performed on each component in the project. Besides that, other testings such as testing the response latency and the positioning accuracy were also used in this project to test the quality of the project.

3.2. Results

The results of testing prove the quality of the project.

4. IMPACT

Microsoft

- The Lighthouse project help to enlarge the market of a large amount of visually impaired people.
- Save funds by using cheap sensors and components since the sensors used in project are cheap and easy to get.

Visually impaired people

- The Lighthouse project help them move around easily and avoid the barriers without assistance from others.
- It also help them regain the confidence of exploring the world and become more independent.

5. CHALLENGES

- The initial plan of using device from Electrical Engineering students failed due to the poor quality of the beacons. We chose better sensors after research.
- The testing was not performed with real data because the CS students didn't deliver expected data till the end of project. We tested the project with a simulator.
- The limited quality of server slowed down the progress. And we ungraded it timely.
- The Obstacle API cannot work on Android, since Android does not support data transfer via main thread. So we modified the API by anding more methods.
- Programming in Java for Spark took a long time, so we changed to write the code in Scala.
- Spark cannot work with sensor data in different batches. Therefore, so we completely changed the processing approach.

6. LESSONS LEARNT

- Research is necessary in the beginning of the project.
- Risk management plan is needed to avoid failure of the project.
- Hardware requirements should be decided as early as possible.
- Think of the feature of both side first when create API.
- New programming language maybe make coding easier, such as Scala for Spark.
- Implementation of features should be right away done for the technology which is going to be used.