

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

||||| Updated upstream

1. PROJECT CONTEXT AND OBJECTIVES

The Lighthouse team collaborated with Microsoft and the Guide Dog Associations to produce applications which can improve the experience of 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 move around especially in an unfamiliar environment without assistance.

Microsoft is both the sponsor and the client in this project and particular field of study was to explore the use cases for wearable sensors in aiding the visually impaired in navigating an indoor environment. Also, the client wanted the team to use low-cost, off-of-shelf hardware in this project and to use a technique called sensor fusion to attain acceptable accuracy.

Finally, Lighthouse team created a platform for dynamic indoor obstacles mapping using wearable sensors. Deliverables included an Obstacle API which allowed for querying for obstacles in an area, an obstacle processing platform and an Android application that collects data and gives users feedback. After testing, the quality of the project meet the client's expectations.

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

~~~~~ Stashed changes 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

In the unit testing, the code coverage reaches 100% according to EcEmma. And the result of load testing proved the platform had higher quality than requirements and the package loss rate is lower than 2%. When perform the integration testing, results with sensor fusion was better than that without sensor fusion in both noisy and no noise situation.

### 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 upgraded 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 adding 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.