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 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 expecations.

2. ACHIEVEMENTS

2.1. Obstacle mapping platform

The team delivered a platform that gatheres data from sensor and pushes them to the cloud throug a component called the Obstacle API which handles communication between data collection sensors and the processing platform. The obstacle API also provides an interface to query the platform for obstacles in a specified area.

The data sent through the Obstacle API is sent through a processing pipeline that performs filtering, aggregation and other steps to produce estimated locations of obstacles in the area. The pipeline is implemented in Apache Kafaka and Apache Spark and uses sensor fusion techniques to increase the accuracy of the obstacle location estimation.

An android application was created to show query the Obstacle API to provide the user with details of obstacles in surrounding area. It showes the obstacles on the map that is centered on the location of the user.

2.2. Others deliverables

The MSc CS students in this team worked on collecting data from sensors and sending the readings through the Obstacle API. The also researched interaction and feedback mechanisms that the client wanted explored.

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.