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

Abstract goes here

Additional Key Words and Phrases: Sensor Fusion, Software Engineering, Visually Impaired, Blind, Microsoft

ACM Reference Format:
Johann Thor Kristthorsson, Ifeanyi Ndu, Veselin Pavlov and Shuang Zhang, 2016. *ACM Trans. Softw. Eng. Methodol.* V, N, Article A (January YYYY), 11 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

There are two million people living with visual impairment in the UK and despite that fact indoor environments are often not designed with this large group of people in mind. [NHS 2016] This means that the visually impaired have to depend on mobility tools to make their way around their environment. Mobility canes and guide dogs are the most important ones but are limited in their use. The canes have limited use to a person needing to navigate from place to place and the guide dogs are not currently available to assist all visually impaired persons in the UK. The Guide Dogs Association is working towards the goal of providing as many people with guide dogs as possible but are not able to meet demand. The project falls under the Cities Unlocked umbrella, an initiative that was started to respond to the lack of available guide dogs in the UK [Cit 2016]. Cities Unlocked has recently been moving more towards improving and enriching the experiences of the visually impaired, rather than focusing solely on their mobility.

Microsoft has been exploring different technologies that could help the visually impaired in their day to day lives and wanted to elicit the help of UCL in this exploration. A team of 8 MSc students, 4 from software systems engineering and 4 from computer science was put together and one member was appointed as team leader. To clarify the specific topic of the project several brainstorming sessions were held to explore the aims and goals. The specific field of study that the clients wanted the team to explore was the use of wearable sensors that could be used by the visually impaired. Additionally the clients wanted to use cheap, off the shelf, hardware and use a technique called Sensor Fusion to maintain accuracy. After exploring several possible use cases for these technologies the client and the UCL team decided to find a way to improve the experience of the members of the visually impaired population when entering specific indoor environments. Specifically in environments that are unfamiliar to the user and

This work was made in collaboration with Microsoft and the Guide Dogs association.

DOI: http://dx.doi.org/10.1145/0000000.0000000

that are not equipped with any specific navigation infrastructure. There has already been thorough research done on indoor navigation but the identification of obstacles seemed to be a good fit for the hardware and technology requirements of the client.

2. LITERATURE REVIEW

Obstacle detection is a large topic and has been popular in the last few years with the advent of autonomous vehicles and unmanned aerial vehicles. The Defence Advanced Research Project Agency (DARPA) in the United States has held several competitions to push the research community towards this specific topic. The DARPA Grand Challenge in 2005 and the Urban Challenge in 2007 offered millions of dollars in prizes and sparked great interest within the research community [DAR 2005, 2007].

2.1. Obstacle detection

In the specific case of obstacle detection as an assistive technology there have been several papers exploring the topic. Cardin et al. [2007] developed a wearable system that uses an array of ultrasonic transducers to do sonar sensing of obstacles around a subject. The subject is notified of the obstacles through vibrotactile instruments sewn into clothing around the subjects torso. Experiments were conducted by blindfolding test subjects and having them navigate an environment filled with obstacles. The time it took the subjects to navigate was recorded and the use of the system resulted in a 50% reduction in navigation time after a short training time [Cardin et al. 2007]. The system developed by Shin and Lim [2007] is similar to Cardin et al. [2007] but adds audible feedback through a headset [Shin and Lim 2007]. These methods are entirely reactive, in that they sense and identify obstacles in a space but do not record its position for possible future use. In addition to this the evaluation of the detection accuracy is lacking. The studies do not describe their experimental setups accurately and the data collection methods are never mentioned.

2.2. Sensor Fusion

Sensor fusion, or cooperative fusion, is a well known term in the context of obstacle detection and navigation. Labayrade et al. [2005] describe algorithms and architectures that facilitate cooperative fusion of two different kinds of sensors, stereovision and LIDAR to detect obstacles in an autonomous driving scenario [Labayrade et al. 2005]. Cooperative fusion is used to detect obstacles in the road in front of a vehicle and determine its distance with good accuracy. The use of sensor fusion decreased the rate of false negatives in the detection from 14.7% and 5.2% for LIDAR and stereovision respectively, down to 5.2%. In addition the rate of false negatives went down from 4.5% and 3.2% respectively to 1.2%. Cho et al. [2014] reached similar results showing an increase in the rate of true positives in obstacle detection from 83.2% to 89.9% by using fusion instead of using actual sensor values [Cho et al. 2014]. Contrary to the research found for the use cases for the visually impaired these papers provide a thorough explanation of the evaluation and data gathering performed and will prove useful in designing the evaluation strategy for this project. These methods have been developed specifically for use in the scenario of autonomous driving but are directly applicable to the scenario of detecting obstacles around a visually impaired pedestrian. Sensor fusion was introduced into the Windows 8 Operating system to provide stability to applications relying on clean and stabilised data. Gear [2012] described a situation where engineers at Microsoft attempted to map a 3D virtual environment to a real environment in an app running on a tablet. First of all they tried emulate up and down movements in the virtual environment using an accelerometer. Almost immediately they encountered an issue when tilting the tablet and holding the tablet in a stationary position. Noise coming from the accelerometer alone caused jittery movements in

the applications. To achieve a much more stable movement, the accelerometer readings was passed through a low-pass filter. The low-pass filter fulfilled its duty but simultaneously introduced a bottleneck into the app. Movement was significantly smoother but the motion in the app felt unresponsive and sluggish. The engineers also used a 6-axis compass which is a 3D accelerometer and 3D magnetometer to emulate left and right movements however it did not yield practicable result due to the instability of the 6-axis sensor. Finally the engineers hit a breakthrough with sensor fusion using the 3D accelerometer, 3D magnetometer and 3D gyroscope. Data collected from all 9-axis resulted in a smooth and fluid experience in the virtual app. Additionally the sensors complemented each other in terms of areas where they fell short [Gear 2012]. In light of research described in the literature reviewed the team will be able to reach the aforementioned aims of this project by developing a data collection platform that gathers data from many different cheap sensors. The sensor data will be used to identify obstacles and the accuracy of the results will be increased by using sensor fusion.

3. PROPOSED SYSTEM

3.1. Preliminary work

The initial plan for this project was to make use of hardware developed by students in the Electrical Engineering department. The hardware consists of an armband that can detect ultrasonic frequencies. Those frequencies are emitted by at least 3 beacons in strategically predefined locations in the room. The armband, in conjunction with a computer and an Android device, then triangulates the position of the armband in space. The use cases that were to be explored were centred around detecting gestures and possible indoor location possibilities. However, soon after the project was presented to the team the hardware was found to have several limitations. The team found that the location accuracy was considerably less than what was initially described. The initial description of the hardware stated that the location accuracy was sub-centimetre, which is the theoretical minimum, but the functional accuracy was between 10 and 50 centimetres. This eliminates the gesture use case since more accuracy is needed to get accurate gesture recognition. Furthermore the hardware can only functions if 3 beacons are within a 130°cone in front of the receiver. This severely limits the practicality of the device for indoor location since a wearable piece of hardware is usually blocked by the wearers body in addition to any obstacle in the environment such as static or movable objects. Therefore the high probability of getting unacceptable performance out of the hardware platform was found to cause to much risk to the project and the team mitigated that risk by pivoting the focus of the project to different topic.

3.2. Stakeholders and Project Context

In order to identify and prioritise the goals and objectives of the SSE team first enumerated the stakeholders of the project and their desired outcomes. This would prove valuable to maintain a clear vision throughout the development process and to limit the risk of non-acceptance later on in the project.

3.2.1. Microsoft. Microsoft is both the sponsor and the client in this project, in addition to that Microsoft provides the team with access to domain experts and other resources. The main outcome that Microsoft wants from the project is research into available assistive technologies based on obstacle detection and sensor fusion. The team is therefore responsible for conducting that research, enumerating the technologies available and developing a proof of concept prototype to illustrate the possibilities of a system using them. Mr. Jarnail Chudge is representing Microsoft and will be the teams client for the duration of the project.

3.2.2. Academic. The academic administrators and supervisors of this project have a specific vision for how the team should apply their knowledge to the development process and client communication. The outcomes the academic stakeholders are interested in are having the team maintain rigorous documentation of the project, upholding state of the art software engineering processes and maintaining good communication with the client throughout the process.

3.2.3. Team members. The team members have a different stake in the project since they want, not only, a good mark but also a rewarding learning experience. They want to see the project fulfilling the requirements of the other stakeholders while also getting a chance to tackle interesting and challenging problems that will widen their knowledge base in the field of Software Engineering. In order to ensure a passing mark for every member of the team the SSE team aims to minimise the overall risk of the project and put in place contingencies in the event of possible non-acceptance of the project.

3.3. High level goals

The teams initial contact with the client was a brainstorming session where he expressed the vision for the project. He expressed Microsoft's desire to have an end product which would feel natural to the user with very little work on the user's part in terms of learning getting themselves familiar with the product. Then they raised the idea of using a sensor or combination of wearable sensors to achieve the ultimate goal. Ideas were then bounced around and a subject area was found that was open for research and of value to our client. Microsoft Research have done some investigation into using Sensor Fusion and wanted to see how that technology could be used in aiding the visually impaired [Gear 2012].

After the brainstorming session the team met several times to brainstorm further and research possible applications around the sensor fusion concept and decided on a few possible project proposals. These were presented to the client and he expressed interest in continuing with one of them. For the final project proposal our client agreed on a research area that will help the visually impaired navigate an indoor environment. Using sensor fusion the team was to build a system that could collect data from various disparate sensors and identify the location of the user and obstacles in its environment.

The goal is to use cheap, off-the-shelf sensors and mitigate any error in the sensor readings by correlating different readings. This will help us reduce error and increase accuracy by using various sensor fusion methods. In order to identify the main goals of our client the team came up with scenarios and use cases, within the scope that was mentioned in section 3.1, and presented them to the client. This was followed up with a discussion that lead to the identification of the following high level goals.

The main objective was to improve the experience of the visually impaired in a specific scenario. The scenario in question involved a visually impaired individual entering a space like a coffee shop and having to navigate around obstacles like tables and chairs. This made obstacle detection a clear objective and secondary objectives relating to that were then further explored. The use of the obstacle detection in conjunction with detection of points of interest was discussed as a possible objective but was deemed less important by the client. The use of Open Street Maps was discussed in this context as a database to work from with regards to gathering information on points of interest [Foundation 2016]. Another secondary objective that came up on the topic of obstacle detection was the use of sensor fusion. Members of the Microsoft Research team had shown interest in the team evaluating sensor fusion techniques in

the context of obstacle detection. This objective was deemed important since it fit well with the rest of the project.

3.4. Requirements

Table **??** shows a list of requirements our system will fulfil (please see section 9 for a full description of our requirement patterns).

List of Requirements	
Identifying points of interest in the environment.	Detecting obstacles
Feeding back mapping information	Manually identifying obstacles in a space
Manually removing an obstacle from the map	Providing navigation for a user to a point of interest
Voice interactions - listing	Voice interactions - acceptance
Screen interactions - listing	Screen interactions - acceptance
Classification of obstacles/Points Of Interest	3D Directional Sound
Head Gestures	IR and Ultrasound

4. SYSTEM ARCHITECTURE

Given the goals the team and the client set for this project there were several considerations to be made to make the architecture flexible and perform enough to satisfy the requirements. Having the requirement of being able to accept data from many kinds of sensors made it clear that the architecture had to be made up of a variety of strategies to homogenise the data so that it could be processed in a more abstract way. Subsequent processing could then be performed based on a fixed data model that would make it easier to rapidly develop and evaluate different processing strategies.

4.1. Data collection

The data collection aspect of the project was a major architectural concern since the data processing part of the pipeline is supposed to handle data coming from may different kinds of sensors. This part would have to be able to filter and make sense of a high volume of sensor readings and prepare them for further processing. Therefore the SSE team set up a simple, well documented, API for the data collection applications to communicate with. This decoupled the processing platform from the data collection by exposing an interface that clearly stated the data expected by the processing platform from the data collection applications. The initial implementation plan for this part of the architecture was to have the CS students implement each of the data collection applications and the SSE team would implement the processing platform. A concern about this structure was raised by our academic stakeholders where they wanted the SSE team to further decouple their work from the CS students. This was done by having one of the data collection application implemented by the SSE team, thereby making the SSE team responsible for a full slice through the system. Any additional data collection by the CS students would therefore be additive and not essential to the success of the project.

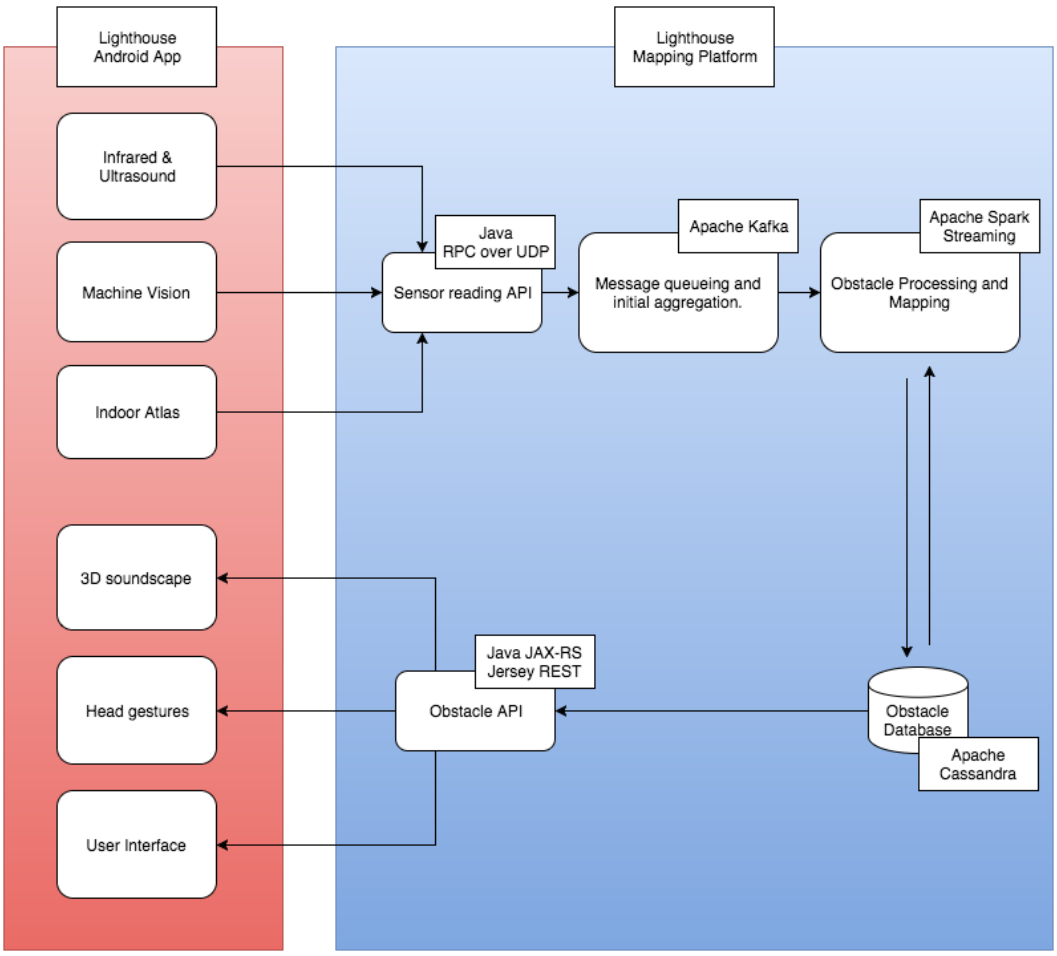

Fig. 1. Temporary architecture figure

4.2. Aggregation

The API mentioned in Section 4.1 was the only contact point for the data collection applications. The component needed to be designed and implemented to handle large quantities of incoming readings, from many different sources, and efficiently buffer and pass them to the processing component. The aggregation component would group together readings and post them to the processing component in a producer/consumer pattern. To manage this the S[illegible]eam evaluated several message broker systems for the job, Apache Kafka, Azure EventHub and RabbitMQ. The criteria used was three-fold, one was the applicability of the technology to the producer/consumer pattern, their cost and their ease of integration with the technology used in the processing component. Azure EventHub was considered as a robust and scalable possibility but was discarded because of the cost associated with running it for the duration of the project, see Appendix B.1. RabbitMQ and Apache Kafka both fit well with the architecture and patterns the team had in mind but ultimately Apache Kafka was chosen because of it's ease of integration with Apache Spark.

4.3. Processing

With Apacha Kafka working as a producer for the producer/consumer pattern and Apache Spark working as the consumer the team managed to handle the level of throughput needed for the purposes of the project. Spark consumes streams of sensor readings coming from Kafka. Kafka has already indexed, filtered and sorted the sensor readings into topics. Spark then took over and pushes the sensors reading streams through several mappings, calculations and reductions to form a processing pipeline that results in the identification of an obstacle.

4.4. Storage and API

The obstacles identified by the Spark stream processing are finally persisted in a database for further processing and use by the feedback applications. The database used for persistence needed to fulfil several requirements to fit the project, scalability with increased load, flexibility of schema and cost. The scalability can be solved by most distributed databases but the flexibility of schema and cost are more difficult to work around. The first step in the choice of database technology was to decide on a SQL or NoSQL database, where NoSQL provided the flexibility needed. After NoSQL was selected the next question was whether to go for a document storage or column storage database. Both provide a flexible schema data persistence but the data modelling needed for each is quite a bit different. After these deliberations there were 3 possibilities explored, Apache Cassandra, Azure DocumentDB and MongoDB. Here again the cost of running an Azure product is prohibitive to an academic prototype project, in addition to that the documentation and troubleshooting resources of the open source technologies is far better than the ones available for Azure. Apache Cassandra and MongoDB both seemed to fit well with the needs of the architecture, both are easily integrated with Spark and both have more than adequate documentation. The deciding factor in choosing between the two was the fact that Cassandra is an Apache distributed system that inherently works with Zookeeper in the same way that Spark and Kafka do. This meant that the time and effort in setup and maintenance would be lessened since all these technologies run in the same managed distributed environment.

4.5. Feedback

The obstacles detected are exposed through an API that queries Cassandra and serves the data to the feedback application. The feedback application uses the locations of the obstacles to notify the user if they are in danger of colliding with them.

5. IMPLEMENTATION

As described in section 4 the chosen architecture uses a variety of different technologies to achieve its goal. Each component therefore had a predefined best practice in implementation. The team aimed to follow best practices when working with each technology and align their work so that each member would be able to work on each component interchangeably.

The first concern the SSE team took on was the choice of programming language. Each of the Apache technologies supported Java, Scala and Python for development so those were the options evaluated. Java and Scala are the most common implementation languages for these technologies and provide the most comprehensive documentation and support. The team felt more comfortable with Java than Scala and the choice of using Android for the front end applications drove the decision towards using Java for the server side as well. This would help the team work together in an environment

everyone was comfortable with and limit the risk posed to the project that comes with the team learning a new environment during development.

5.1. Kafka

A communication handler is set up to listen on a UDP socket for sensor readings. The readings are sent to the socket from the data collection applications in JSON format which the handler receives and deserialises before pushing them into the Kafka message queue.

5.2. Spark

After aggregation, data will be processed/transformed before it is stored inside Cassandra. We agreed on using a Kalman filter or using a moving average in Spark to produce a confident measurement we can store in Cassandra. Vesko you can elaborate just a bit

5.3. Cassandra

The data modelling for Cassandra was fairly simple, the number of data points needed to persist an obstacle were few and clear. After setting up the Cassandra server the data model was set up through CQL and drivers with Java set up for testing and querying.

5.4. Obstacle Query API

To query the database an API was set up that was written in Java using the Jersey framework. The API is hosted on the same machine as Cassandra using the Tomcat web server. Upon processing by spark, a single obstacle location is stored in Cassandra with a confidence of 0 - 100. Confidence states how certain we are the obstacle would move position with time. For example a pillar would have a higher confidence in comparison to a chair or trash can because it will hardly change its position.

5.5. Android app and Jar file

A socket on our VPS will be listening indefinitely on a particular port for incoming data from our various sensing hardware. Each sensor or combination of sensors will measure data, do some preprocessing before sending data across to our VPS over a UDP connection. For instance one of the CS students would send three distance estimates based on the measurements collected from 2 ultrasound and 1 infrared sensor.

5.6. Kafka Producer Listener

At the listening end we would essentially do a partial deserialisation of the data packet into a JSON object to obtain the topics from each packet. Afterwards the JSON object is published into the kafka cluster.

5.7. Spark Processing

An option for the consumption part of the architecture could be carried out using a standard kafka consumer however we went ahead with Spark to provide these services. Spark receives the data from a kafka direct stream, aggregates and processes the data. Kafka and spark interact through a spark kafka streaming plug-in.

5.8. Interaction

Before diving into suitability arguments for the chosen architecture, it would be of immense importance to discuss how components in our architecture will interact with each other.

6. PROJECT MANAGEMENT

The team consists of 8 Masters students, 4 are doing a MSc level conversion course in Computer Science and 4 are doing a MSc degree in Software Systems Engineering. As discussed earlier the 4 CS students are each responsible for a data collection or feedback application that would be the topic of their dissertation project. The role of the SSE team with regards to the CS students was to coordinate their efforts towards creating value for the client and to provide them with guidance with the implementation and architecture of their projects.

6.0.1. Processes. The SSE team managed their work by relying on the Scrum methodology. Work was organised into 2 week sprints and standup meetings were scheduled for communication and status updates. Mondays were set up as meeting days to synchronise with stakeholders, academic supervisors and our client. Standup meetings were held the following Tuesday which was used to re-organise and adapt the work schedule over the week to the most recent project vision of our client. Fridays were used to write reports and inform our stakeholders of the progress made. The team would hand in a weekly report and send demonstration material to the client to show progress and receive feedback.

6.1. Testing strategy

The testing strategy was a large consideration for the SSE team, since the many components of the architecture had different interfaces so each needed a strategy tailor made for it. In order to manage and enforce the testing strategy the team set up a continuous integration server, Jenkins, to run tests as code was pushed to Git and run unit, load and integration tests.

6.1.1. Kafka. The main quality requirement for the UDP socket server for the Kafka producer was the throughput. Therefore load tests were the main focus in the testing strategy for this component. Since the component uses UDP the metrics gathered were to measure number of packets processed, number of packets dropped and the processing speed of each packet from the socket and into the message queue. The team considered using unit testing and integration tests to evaluate this component but the work involved in mocking out dependencies was not the most efficient use of time. The Kafka producer should contain as little logic as possible and therefore unit tests would be of little value. Integration tests on this component would be run on the component when testing the integration with the Spark processing component.

6.1.2. Spark. This component has a similar testing strategy to the Kafka component. The processing throughput is a large concern since the entire processing pipeline needs to be checked for bottlenecks so load test will need to be performed. Integration tests are also required to test if the processing is producing the correct output. Here the dependency of the processing component on the Spark environment creates real problems to unit testing. Mocking this out is a major effort so the integration tests were found to be sufficient to assure the team that the processing is producing the correct results.

6.1.3. Obstacle API. The obstacle REST API has several logic components that need unit testing in addition to other testing methods. The logic will be tested through unit tests and the HTTP compliance and error checks will be performed through integration tests. In addition to this some load tests will be performed to make sure the API stands up to a minimum level of load expected in our quality requirements.

6.1.4. Indoor Atlas.

(a) Figure 1 (b) Figure 2 (c) Figure 3

Fig. 2. A figure with two subfigures

(a) Figure 4 (b) Figure 5 (c) Figure 6

Fig. 3. A figure with two subfigures

7. EVALUATION

7.1. Performance and Scalability

Discussion of the amount of messages that can be processed in Kafka -¿ Spark -¿ Cassandra. Talking about aggregation and such. Packet drop, burst of messages vs sustained.

7.2. Detection accuracy

Describe the testing scenario, the simulator and show diagrams that compare both.

7.3. Acceptance

8. LIFE CYCLE AND FUTURE WORK

8.1. Current state

Describe the current state of the project
Capabilities, how many of the Goals and Requirements have been fulfilled.
Quality requirements and the tests we used to evaluate them

8.2. Maintenance and Scaling

Describe how the project can be maintained and scaled in the event of deployment.
Talk about the migration plan to Azure.

9. CONCLUSIONS

APPENDIX

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

REFERENCES

2005. DARPA 2005 DARPA Grand Challenge. http://archive.darpa.mil/grandchallenge05/. (2005). Accessed: 2016-08-05.

2007. DARPA 2007 DARPA Urban Challenge. http://archive.darpa.mil/grandchallenge/. (2007). Accessed: 2016-08-05.

2016. Blindness and vision loss, NHS website. http://www.nhs.uk/conditions/Visual-impairment/Pages/Introduction.aspx. (2016). Accessed: 2016-08-08.

2016. Cities Unlocked website. http://www.citiesunlocked.org.uk/about. (2016). Accessed: 2016-08-08.

Sylvain Cardin, Daniel Thalmann, and Frédéric Vexo. 2007. A wearable system for mobility improvement of visually impaired people. *The Visual Computer* 23, 2 (2007), 109–118. DOI:http://dx.doi.org/10.1007/s00371-006-0032-4

H. Cho, Y. W. Seo, B. V. K. V. Kumar, and R. R. Rajkumar. 2014. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 1836–1843. DOI:http://dx.doi.org/10.1109/ICRA.2014.6907100

Open Street Maps Foundation. 2016. OSM Open Street Maps. https://www.openstreetmap.org/about. (2016). Accessed: 2016-08-18.

Gavin Gear. 2012. Supporting sensors in Windows 8. https://blogs.msdn.microsoft.com/b8/2012/01/24/supporting-sensors-in-windows-8/. (2012). Accessed: 2016-08-17.

Raphaël Labayrade, Cyril Royere, Dominique Gruyer, and Didier Aubert. 2005. Cooperative Fusion for Multi-Obstacles Detection With Use of Stereovision and Laser Scanner. *Autonomous Robots* 19, 2 (2005), 117–140. DOI:http://dx.doi.org/10.1007/s10514-005-0611-7

Byeong-Seok Shin and Cheol-Su Lim. 2007. *Obstacle Detection and Avoidance System for Visually Impaired People*. Springer Berlin Heidelberg, Berlin, Heidelberg, 78–85. DOI:http://dx.doi.org/10.1007/978-3-540-76702-2_9

Online Appendix to: 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

A. THIS IS AN EXAMPLE OF APPENDIX SECTION HEAD

B. APPENDIX SECTION HEAD

B.1. Architecture cost analysis

DOI: http://dx.doi.org/10.1145/0000000.0000000