NELSON MANDELA
UNIVERSITY

Masters Dissertation

USING SOUND LOCALIZATION TO GAIN DEPTH PERCEPTION FOR THE VISUALLY IMPAIRED THROUGH SENSORY SUBSTITUTION

James Carmichael de Klerk

Submitted in fulfilment of the requirements for the degree of Masters of Computer Science and Information Systems in the Faculty of Science at the Nelson Mandela University

April 2020

Supervisor: **Dr. Dieter Vogts**
Co-Supervisor: **Prof. Janet Wesson**

Acknowledgments

I would like to start off by thanking the Nelson Mandela Telkom Centre of Excellence and the Nelson Mandela University Department of Research Capacity Development for providing the research funding and equipment to enable this research to take place. It largely due to the funding from groups and initiatives such as these that makes masters, doctoral and other research feasible. For that, I am grateful.

There are many people in my life who have made the writing of this dissertation possible. I would like to take this opportunity to sincerely thank them. First of all, I would like to thank the many staff at the Nelson Mandela University Department of Computing Sciences who have been involved in my university career up until this point. The experience I have had would not have been the same without the countless hours that go into running such a great department. More than that, I have thoroughly enjoyed being lectured and mentored by such wonderful people.

To my postgraduate friends: We have been through a lot together and I consider that a blessing. In so many ways we have learnt so much, and yet, we realise that there is still so much to learn. Joshua, Thashen, Rhys, Clara, George, Dean, Brandon, Tim, Grant and Peter. I want to thank each and every one of you for playing an important role in this journey.

To my family, Hennie, Hayley, Jason and Samantha, and to my good friend Scott: Thank you all for always being there and supporting me throughout all the highs and lows. I would not have been able to complete this without your continued support and encouragement.

Last but certainly not least, my supervisors – Dr. Dieter Vogts and Prof. Janet Wesson. I appreciate every moment of advice, encouragement, support, assistance and patience. For all that, and so much more, I cannot thank you enough. The countless hours of work that each of you put into lecturing and supervising – among many other things – often goes unnoticed. It is only at times like this, when I reflect on all that you do, that I realize how grateful I am to you both. You are definitely part of what makes the Department of Computing Sciences such a special place. So once again, thank you so much for all you have done for me in making this dissertation possible.

Abstract

The visually impaired do not have the visual ability to localize objects in three-dimensional space, rather, they rely on their other senses to gain depth perception. Sensory substitution is the concept of substituting one sense for another, normally substituting an impaired sense with a functioning sense. Visual-to-auditory sensory substitution substitutes an impaired visual sense with a functioning auditory sense. This research aimed to investigate and develop techniques for visual-to-auditory sensory substitution – using sound localization as a sensory substitution for depth perception.

The research started by investigating the characteristics of human audition with a particular focus on how humans localize sounds. It then looked at existing visual-to-auditory sensory substitution systems and the techniques they used. From the existing systems, a system known as MeloSee was chosen as a baseline for developing and evaluating further sensory substitution prototypes.

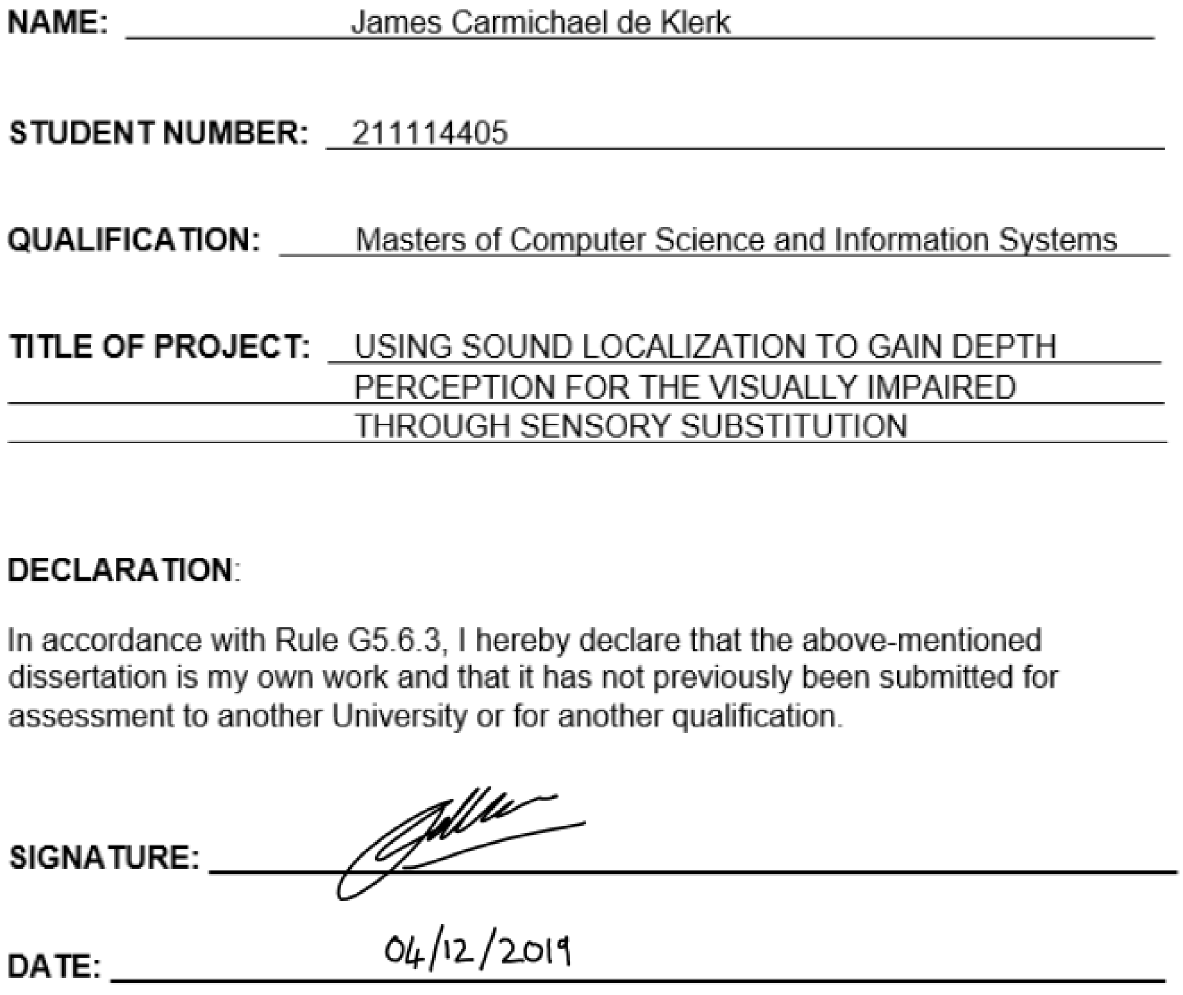
The baseline prototype (P_0) was then implemented and a preliminary study performed. Based on the knowledge gained from the preliminary study, baseline implementation and the background research, a set of improvement recommendations were generated. The next iteration – Prototype 1 (P_1) – was then developed based on the recommendations. A comparative study between P_0 and P_1 was then performed. Based on the study, another set of improvement recommendations were generated. From the recommendations, a final prototype was developed – Prototype 2 (P_2). The last comparative study was then performed between P_0 and P_2, with a third set of recommendations being generated as a result.

From the studies it was found that participants using P_0 were able to identify when they were approaching large objects such as walls. P_1 built on that, improving the ability to identify the quadrant of a nearby isolated object. P_2 built on P_0 and P_1, achieving similar results to P_1 for identifying the quadrant of nearby isolated objects, and improving on P_0 and P_1 with regard to depth discrimination – especially for navigation tasks where there were no obstacles.

Based on the three sets of recommendations and what was learnt over the course of the research, a set of visual-to-auditory sensory substitution techniques were presented. The techniques aim to be useful for implementing visual-to-auditory sensory substitution systems, which would provide the visually impaired with the visual ability to localize objects in three-dimensional space through sound.

Keywords: *sensory substitution, visually impaired, depth perception, sound localization, visual-to-auditory*

Declaration by Candidate



NAME: James Carmichael de Klerk

STUDENT NUMBER: 211114405

QUALIFICATION: Masters of Computer Science and Information Systems

TITLE OF PROJECT: USING SOUND LOCALIZATION TO GAIN DEPTH PERCEPTION FOR THE VISUALLY IMPAIRED THROUGH SENSORY SUBSTITUTION

DECLARATION:

In accordance with Rule G5.6.3, I hereby declare that the above-mentioned dissertation is my own work and that it has not previously been submitted for assessment to another University or for another qualification.

SIGNATURE:

DATE: 04/12/2019

Contents

Acknowledgments i

Abstract ii

Declaration by Candidate iii

Contents iv

List of Figures vii

List of Tables xi

List of Abbreviations xii

Chapter 1 : Introduction 1
1.1 Background 1
1.2 Problem Statement 2
1.3 Research Aim 2
1.4 Research Questions 2
1.5 Research Methodology 3
1.6 Ethical Considerations 5
1.7 Dissertation Structure 6

Chapter 2 : Background Research 9
2.1 Introduction 9
2.2 Psychoacoustics 9
2.2.1 Anatomy of the Human Ear 9
2.2.2 Audible Range 11
2.2.3 Sound Localization 12
2.2.4 Sound Interference 19
2.2.5 Discussion of related technologies 20
2.3 Conclusion 20

Chapter 3 : Sensory Substitution 21
3.1 Introduction 21
3.2 Types of Sensory Substitution Techniques 21
3.3 Sensory Substitution and the Brain 22
3.4 Visual-to-Auditory Sensory Substitution Devices 23
3.4.1 The vOICe 23
3.4.2 The PSVA 25
3.4.3 MeloSee 26
3.4.4 Notable Sensory Substitution Devices 28
3.4.5 Overview of Sensory Substitution Devices 29
3.5 Evaluation Procedures for Sensory Substitution Devices 31
3.6 Baseline Prototype 33
3.7 Conclusions 33

Chapter 4 : Sensory Substitution Framework 35
4.1 Introduction 35
4.2 Prototype Development Objectives 35
4.3 Motivation for Developing a Framework 36
4.4 Framework Development 37
4.4.1 Robotic Operating System (ROS) 37
4.4.2 Sensory Substitution Framework Overview 39
4.4.3 Sensory Substitution Framework Features 41
4.4.4 Dashboards 42
4.4.5 SSF Core 43
4.4.6 Language Used 43
4.5 Conclusions 44

Chapter 5 : Evaluation Design 45
5.1 Introduction 45
5.2 Participant Selection 46
5.3 Objects Used for Evaluations 46
5.4 Evaluation Tasks 47
5.5 Questionnaires 52
5.6 Comparative Study Evaluation Procedure 52
5.6.1 System Evaluation Procedure 53
5.7 Evaluation Metrics 54
5.7.1 Evaluation Task Metrics 55
5.7.2 Questionnaire Metrics 58
5.8 Conclusions 60

Chapter 6 : Design Cycle 1 61
6.1 Introduction 61
6.2 System Overview 61
6.3 Hardware and Software Used 62
6.3.1 Hardware Design 62
6.3.2 Software 64
6.4 Design Cycle 1 Overview 65
6.5 Baseline Prototype ($P0$) 66
6.5.1 Implementation 66
6.5.2 Preliminary Study 69
6.5.3 Recommendations 71
6.6 Prototype 1 ($P1$) 73
6.6.1 Implementation 73
6.6.2 Preliminary Study 82
6.6.3 Comparative Study 82
6.6.4 Recommendations 93
6.7 Conclusion 94

Chapter 7 : Design Cycle 2 96
7.1 Introduction 96
7.2 Design Cycle 2 Overview 96

7.3 Prototype 2 ($P2$)....97
7.3.1 Implementation....97
7.3.2 Preliminary Study....100
7.3.3 Comparative Study....100
7.3.4 Recommendations....112
7.4 Conclusion....115

Chapter 8 : Conclusion....116
8.1 Introduction....116
8.2 Research Contributions....116
8.3 Research Achievements....117
8.4 Challenges and Limitations....119
8.5 Summary....120

References....121

Appendices....127
Appendix A : Ethics Approval....127
Appendix B : Written Information....128
Appendix C : Consent Form....129
Appendix D : Pre-Evaluation Questionnaire....131
Appendix E : Post-Evaluation Questionnaire....132
Appendix F : Final Questionnaire....133
Appendix G : Evaluators Sheet....134
Appendix H : SSF Usage....139
Appendix I : Sensory Substitution Framework – README....141

List of Figures

Figure 1-1: Design science research process (DSRP) model (Peffers et al., 2006)............................3

Figure 1-2: Design Science Research Cycles (Hevner, 2007)..4

Figure 1-3: Overview of the dissertation structure and how it links to the DSR methodology 7

Figure 2-1: Schematic of the outer, middle and inner ear (Zwicker & Fastl, 1999).........................10

Figure 2-2: Estimated new equal-loudness contours (Suzuki, 2003)..11

Figure 2-3: Coordinate system for sound localization (Plack, 2005) 13

Figure 2-4: A top-down view of the head with a sound source on the right (Plack, 2005).............13

Figure 2-5: Acoustic shadow for high and low-frequency sounds (modified) (Wolfe et al., 2014).14

Figure 2-6: Frequency spectrum for the same sound at two elevations (Goldstein, 2009)............15

Figure 2-7: Visual representation of the Inverse Square Law (Nave, 2017b).................................16

Figure 2-8: Relationship between a sound source’s distance and loudness (Begault, 2000) 17

Figure 2-9: Sound source reflections (Plack, 2005)...17

Figure 2-10: Superposition of waves seen on the right (Nave, 2017a) 19

Figure 2-11: Audio Clipping...19

Figure 3-1: An illustration of the sensory substitution device and its conversion principles. (Proulx, Stoerig, Ludowig, Knoll, & Schnitzler, 2008)..24

Figure 3-2: An artificial retina with four levels of resolution, amounting to a total of 208 pixels (Capelle et al., 1998) 25

Figure 3-3: MeloSee SSD flowchart (Stoll et al., 2015) 26

Figure 3-4: Grayscale depth image (a). Activation computation (b). RF activities (c). (Stoll et al., 2015) 27

Figure 3-5: EyeMusic Overview (Abboud et al., 2014)...29

Figure 3-6: The 15 basic patterns used rotated by 45 or 90 to produce 50 images used in the PSVA evaluation (Capelle et al., 1998) 32

Figure 3-7: Paths A (left) and B (right) used in the MeloSee experiment (Stoll et al., 2015) 32

Figure 4-1: Simple ROS message description 38

Figure 4-2: ROS Master, Node and Message Interaction 38

Figure 4-3: Sensory Substitution Framework pipeline 39

Figure 4-4: Example of a Depth Image (a), Pre-processed Image (b) and a Retinal Encoded Image (c). 40

Figure 4-5: Three views of a 3D visualisation provided by the SSF. The grey sphere represents the listeners head, and the coloured spheres represent sound emitters placed in 3D space relative to the listener. 41

Figure 4-6: Sensory Substitution Framework prototyping dashboard 42

Figure 5-1: Three box sizes used in the evaluation, small (a) medium (b) and large (c) together with the quadrant task object (d) – all measurements are given in meters 46

Figure 5-2: Scale diagram of the passages used in the evaluation – all measurements are in meters 47

Figure 5-3: The quadrant task object being held up 49

Figure 5-4: Two configurations of the evaluation room. Quadrant Task configuration (a). The Box Placement Task and Object Count Task configuration (b) – all measurements are in meters 49

Figure 5-5: Quadrant Task marking system 57

Figure 5-6: Box Task marking system 57

Figure 5-7: Multiple Boxes Task marking system 58

Figure 5-8: System Preference Rating 59

Figure 6-1: Visualisation of the virtual speaker placement in a scene, for a SSD which uses sound localization 62

Figure 6-2: Hardware used for the prototype. The full sensory substitution system (a), a close-up of the depth camera mounted to a hard hat (b), a close-up of the bone conduction headphones (c), multiple angles of the 3D printed mount used to connect the depth camera to the hard hat (d). 63

Figure 6-3: Intel RealSense D435 mounted onto a hard hat next to a Microsoft Kinect 64

Figure 6-4: Sensory Substitution Framework Structure 65

Figure 6-5: Overview of Design Cycle 1 66

Figure 6-6: Depth Image with overlay of the sampling process (a) next to the output from the Baseline Prototype’s Retinal Encoder (b) 67

Figure 6-7: Waveforms generated in Audacity by combining 264 Hz and 352 Hz sound waves 69

Figure 6-8: RAW Depth Image 71

Figure 6-9: Pre-processor Event Flow 74

Figure 6-10: Original Depth Image (Suh, Kim, Park, & Suh, 2010) used to generate 9 x 9 scaled down depth images 75

Figure 6-11: Quantization Levels 76

Figure 6-12: Grid overlaid on top of the Pre-processed depth image showing the resolution for the Baseline Prototype (a) and Prototype 1 (b) Retinal Encoders 77

Figure 6-13: MeloSee Retinal Encoder (b) vs Prototype 1 Retinal Encoder (c) 78

Figure 6-14: Perspective Projection (“Script Tutorials,” 2018) 80

Figure 6-15: Segmentation of the image for generating the unit vector map 81

Figure 6-16: Graph legend 83

Figure 6-17: Average number of $Ewall$ errors made per participant ($n = 8$) in the Navigation Task (no obstacles) 87

Figure 6-18: Average number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles) – grouped by error type 87

Figure 6-19: Total number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles) 88

Figure 6-20: Total number of correct answers for the Quadrant Task across all participants ($n = 8$); separated into correct answers for left-right (L-R) and top-bottom (T-B) 89

Figure 6-21: Total number of correct answers for the Quadrant Task across all participants ($n = 8$) 90

Figure 6-22: Total number of correct answers across all participants ($n = 8$) for the Box Task – grouped by metric 91

Figure 6-23: Total number of correct answers across all participants ($n = 8$) for the Multiple Box Task – grouped by metric 92

Figure 7-1: Overview of Design Cycle 1 and Design Cycle 2 96

Figure 7-2: Average number of $Ewall$ errors made per participant ($n = 8$) in the Navigation Task (no obstacles) 104

Figure 7-3: Average number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles) – grouped by error type 105

Figure 7-4: Total number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles) 105

Figure 7-5: Total number of correct answers for the Quadrant Task across all participants ($n = 8$); separated into correct answers for left-right (L-R) and top-bottom (T-B) 106

Figure 7-6: Total number of correct answers for the Quadrant Task across all participants ($n = 8$) 107

Figure 7-7: The corrected graph of the total number of correct answers for the Quadrant Task across all participants ($n = 8$); separated into correct answers for left-right (L-R) and top-bottom (T-B) 108

Figure 7-8: The corrected graph of the total number of correct answers for the Quadrant Task across all participants ($n = 8$) 109

Figure 7-9: Total number of correct answers across all participants ($n = 8$) for the Box Task – grouped by metric 110

Figure 7-10: Total number of correct answers across all participants ($n = 8$) for the Multiple Box Task – grouped by metric 111

Figure 7-11: A participant wearing the MeloSee hardware (Stoll et al., 2015) 113

Figure 7-12: Camera mounted at eye level 113

List of Tables

Table 3-1 Frequencies of the ’just intonation’ scale (Fristot et al., 2012) 28

Table 3-2: The Benefits and Shortcomings of Existing SSDs 30

Table 5-1: Metrics Recorded for the Navigation Task (no obstacles) 55

Table 5-2: Metrics Recorded for the Navigation Task (with obstacles) 56

Table 6-1: Intel RealSense D435 Specifications (Pruitt, 2018) 63

Table 6-2: Apple MacBook Specifications 64

Table 6-3: Aftershokz Sportz 2 (AS320) Specifications (Newegg, n.d.) 64

Table 6-4: Baseline Prototype Qualitative Feedback 70

Table 6-5: Baseline Prototype ($P0$) Qualitative Feedback from the $P0$ vs $P1$ study ($\mathrm{n} = 8$) 84

Table 6-6: Prototype 1 ($P1$) Qualitative Feedback from the $P0$ vs $P1$ study ($\mathrm{n} = 8$) 85

Table 6-7: System preference rating for $P0$ vs $P1$ normalized to be between -3 and 3 ($\mathrm{n} = 8$) 86

Table 6-8: Contingency table for left-right discrimination 89

Table 6-9: Contingency table for top-bottom discrimination 89

Table 6-10: Contingency table size discrimination 91

Table 6-11: Contingency table for middle box identification 92

Table 7-1: Baseline Prototype ($P0$) Qualitative Feedback from the $P0$ vs $P2$ study ($\mathrm{n} = 8$) 101

Table 7-2: Prototype 2 ($P2$) Qualitative Feedback from the $P0$ vs $P2$ study ($\mathrm{n} = 8$) 102

Table 7-3: System preference rating for $P0$ vs $P2$ normalized to be between -3 and 3 ($n = 8$) .. 103

Table 7-4: Contingency table for left-right discrimination 106

Table 7-5: Contingency table for top-bottom discrimination 107

Table 7-6: Corrected contingency table for top-bottom discrimination 108

List of Abbreviations

DSR	Design Science Research
FoV	Field of View
HRTF	Head-Related Transfer Function
ILD	Interaural Level Difference
ITD	Interaural Time Difference
NMU	Nelson Mandela University
P_0	Baseline Prototype (Prototype 0)
P_1	Prototype 1
P_2	Prototype 2
PDO	Prototype Development Objective
PP	Pre-processor
PyAL	Python wrapper for OpenAL (a cross-platform 3D audio library)
RE	Retinal Encoder
REC-H	Research Ethics Committee: Human
ROS	Robotic Operating System
RQ	Research Question
SG	Sound Generator
SS	Sensory Substitution
SSD	Sensory Substitution Device
SSF	Sensory Substitution Framework

Chapter 1: Introduction

1.1 Background

An estimated 285 million people worldwide are visually impaired, of which approximately 39 million people are blind (“WHO | Visual impairment and blindness,” 2014). Visual impairment is classified (in ICD-10) as blindness – the inability to see – or low vision, which is a loss of vision that interferes with daily life and is not correctable (Dandona & Dandona, 2006; World Health Organization, 2016).

The visually impaired have a limited understanding of their surroundings due to the lack of vision, relying on sound and touch to gain an understanding of their environment. Due to a lack of vision, depth perception – the visual ability to perceive the three-dimensional world and gauge the distance of objects – is also impacted in the visually impaired. For this reason, navigating an environment as well as object detection and avoidance, especially in an unfamiliar environment can be a challenge. These challenges impact the independence of the visually impaired. To gain a form of depth perception and an understanding of their surroundings, several assistive technologies are commonly used (American Foundation for the Blind, 2017). The white cane and guide dogs are the most prevalent of these due to their simplicity and reliability (Dakopoulos & Bourbakis, 2010).

Sensory substitution – the concept of substituting one sense (e.g. vision) for another (e.g. hearing) – has been investigated for many years in several studies, proving valuable in giving the visually impaired a sense of independence (Elli, Benetti, & Collignon, 2014; Renier & De Volder, 2005). However, outside of laboratories, sensory substitution devices (SSDs) are still not commonly used among the visually impaired; this is thought to be due to their slow refresh rates and interference with existing senses (Elli et al., 2014). However, due to scientific progress and advances in technology, sensory substitution appears to be a promising solution for providing a form of depth perception to the visually impaired (Renier & De Volder, 2005). In recent years several neuroimaging studies have shown that the visual cortices of the blind are used in processing the auditory input of certain vision substitution devices (Stronks, Nau, Ibbotson, & Barnes, 2015), meaning the blind can gain a sense of depth perception through these devices (Renier et al., 2005).

1.2 Problem Statement

The visually impaired do not have the visual ability to perceive depth, resulting in a poor understanding of their surroundings – this makes navigating those surroundings a greater challenge.

1.3 Research Aim

The aim of this research is:

To investigate and develop visual-to-auditory sensory substitution techniques – using sound localization as a sensory substitution for depth perception.

The purpose of achieving this aim is to give the visually impaired an accurate understanding of their surroundings through sound.

1.4 Research Questions

Main research question

How can sound localization be used as a sensory substitution for depth perception, to give the visually impaired an accurate understanding of their surroundings through audition?

Sub-questions

RQ 1. What characteristics of audition (hearing) and sound localization can be used for visual-to-auditory sensory substitution?

RQ 2. What are the benefits and shortcomings of existing visual-to-auditory sensory substitution techniques?

RQ 3. How can visual-to-auditory sensory substitution prototypes be developed to allow for the testing of different visual-to-auditory sensory substitution techniques?

RQ 4. How can visual-to-auditory sensory substitution prototypes be evaluated to provide insight into the effectiveness of the different visual-to-auditory sensory substitution techniques?

RQ 5. What visual-to-auditory sensory substitution techniques can be used to develop a visual-to-auditory sensory substitution prototype for depth perception?

RQ 6. Does the prototype developed provide the visually impaired with an accurate understanding of their surroundings through audition?

1.5 Research Methodology

The research method to be used for this project is the Design Science Research (DSR) methodology. DSR is a methodology for understanding the existing base of knowledge, then using that knowledge to create rigorously tested new and innovative artefacts (Hevner, March, Park, & Ram, 2004). By doing so, the DSR methodology helps researchers contribute new knowledge to the knowledge base. DSR is an established research methodology in the field of Information Systems (Hevner, 2007).

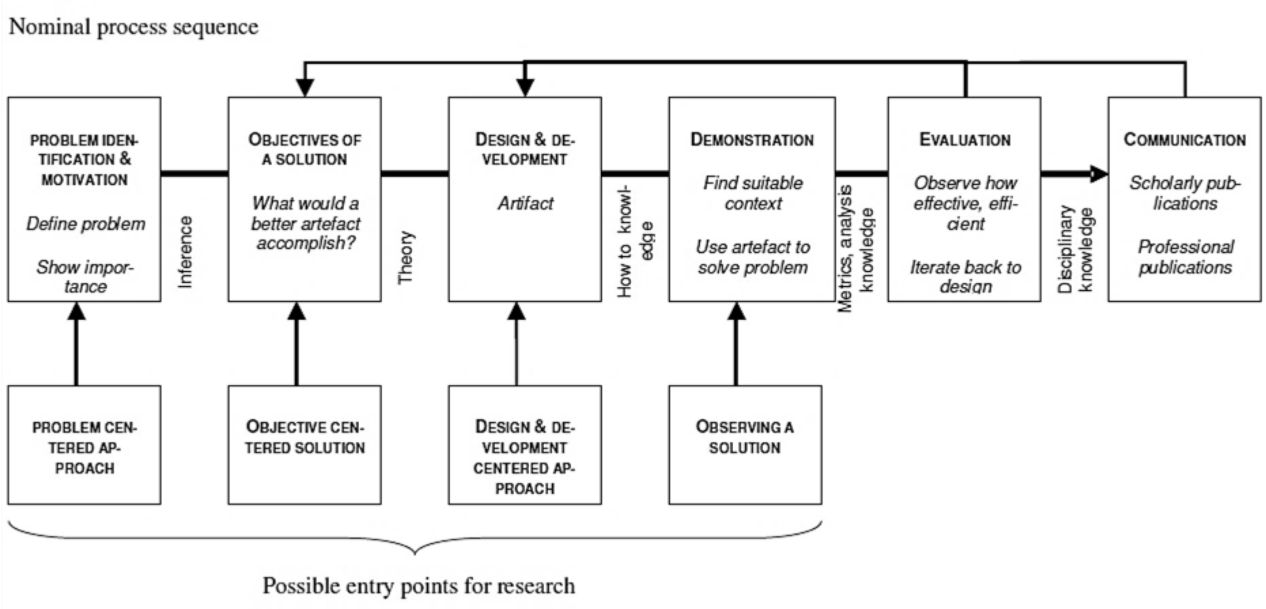

Nominal process sequence
PROBLEM IDENTIFICATION & MOTIVATION
Define problem
Show importance
Inference
OBJECTIVES OF A SOLUTION
What would a better artefact accomplish?
Theory
DESIGN & DEVELOPMENT
Artifact
How to knowledge
DEMONSTRATION
Find suitable context
Use artefact to solve problem
Metrics, analysis knowledge
EVALUATION
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM CENTERED APPROACH
OBJECTIVE CENTERED SOLUTION
DESIGN & DEVELOPMENT CENTERED APPROACH
OBSERVING A SOLUTION
Possible entry points for research

Figure 1-1: Design science research process (DSRP) model (Peffers et al., 2006)

The DSR methodology uses the process model illustrated in Figure 1-1; this process consists of six activities detailed below.

Problem Identification and Motivation: This involves defining the research problem to be solved, and motivating why the solution is relevant and valuable, based on research of the problem domain (Peffers et al., 2006).

Objectives of a Solution: Based on the problem definition, the requirements (objectives) of the solution should be clearly defined (Peffers et al., 2006). These requirements need to be based on research of the problem domain to ensure that the problem has not been solved in the manner one is planning to solve it, else one would not be contributing to the body of knowledge. The requirements should also be clear, since they are a reference point throughout the rest of the project.

Design and Development: This is the creation of artefacts based on the requirements of the solution, using the knowledge about the domain and existing systems. The artefacts can be models, constructs, methods or instantiations (Hevner et al., 2004; Peffers et al., 2006).

Demonstration: This involves demonstrating how well the artefact solves the research problem when put in a suitable context (Peffers et al., 2006).

Evaluation: This involves observing how well the artefact solves the research problem. To determine this, one must look at the solution requirements previously defined in the research process. Using relevant metrics and analytical techniques one should rigorously investigate to what extent the requirements of the solution are met by the artefact (Hevner et al., 2004). After the artefact evaluation is completed, the researchers must decide if they need to iterate back to the design and development phase to improve the artefact, or continue to the communication of their research (Peffers et al., 2006).

Communication: This is the final phase of the design science research process (DSRP) model. Communication of the problem, the research done on the problem, the rigour of that research and the artefact developed is important to build up the body of knowledge in the domain (Peffers et al., 2006).

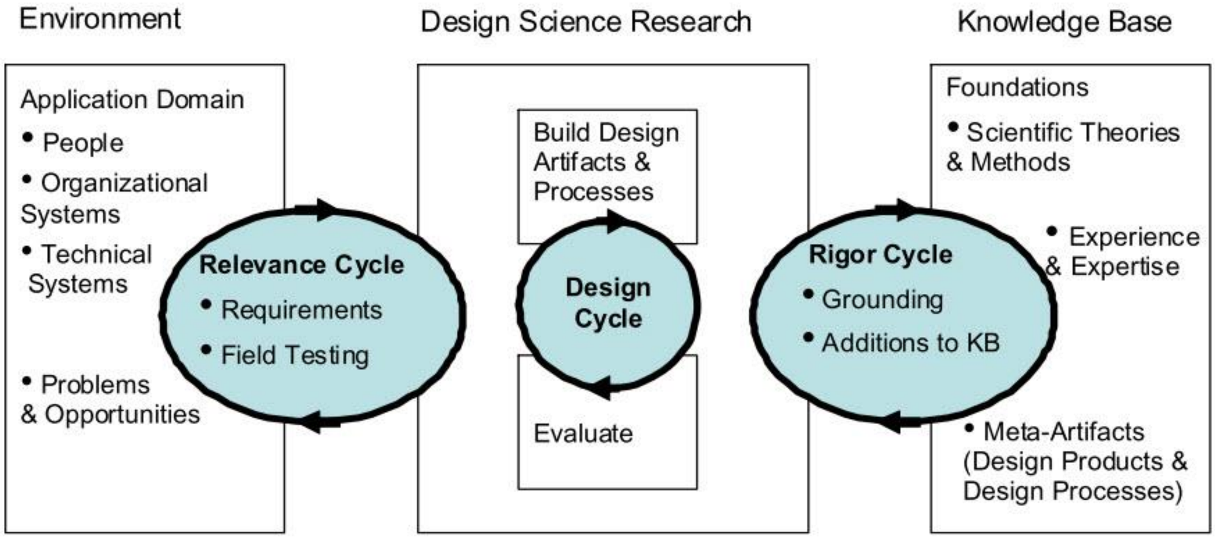


Figure 1-2: Design Science Research Cycles (Hevner, 2007)

Design Science Research can be broken down into three cycles, as illustrated in Figure 1-2. The Relevance Cycle deals with one of the main goals of Design Science Research, to improve the knowledge domain through new and innovative artefacts (Hevner, 2007; Simon, 1996). The Relevance Cycle looks at understanding the application domain, the specific problem to be solved and the requirements of that problem. One should also perform a field study of the artefact once developed to determine if the Relevance Cycle needs to be repeated due to deficiencies in the functionality.

The Rigor Cycle relies on a clear understanding of the existing scientific knowledge base related to the project. This means having expertise in the state-of-the-art research relating to the application domain, and, broad knowledge of the existing artefacts and processes in the application domain (Hevner, 2007). The purpose of the Rigor Cycle is to ensure the research project is contributing new knowledge to the domain, through a rigorous understanding of existing knowledge. Research rigor is shown by thoroughly researching and referencing the existing knowledge base throughout the project to demonstrate the research contribution (Hevner, 2007).

The Design Cycle is primarily done throughout the artefact development, artefact evaluation and evaluation feedback phases. The Design Cycle is the process of repeatedly designing an artefact and evaluating that artefact with reference to the solution requirements (from the Relevance Cycle), until a satisfactory design is achieved (Hevner, 2007; Simon, 1996). The theories used for effective design and evaluation come from the Rigor Cycle (Hevner, 2007).

The next section looks at the ethical considerations for the research. This includes who was allowed to participate in the studies completed.

1.6 Ethical Considerations

For the project, the researcher developed non-invasive sensory substitution hardware and software prototypes relating to the title of the project. Studies were conducted to evaluate the hardware and software developed. Participation in the study required approximately two hours per participant. For the study, the participants were required to navigate through unknown environments, participate in object identification tasks, and complete evaluation questionnaires. This was done while blindfolded and wearing the prototype developed. Sighted participants with hearing in both ears and the ability to hear from 200Hz to 10000Hz were eligible for participation in the evaluations. Participants had to be students of the NMU Department of Computing Sciences. In order to conduct the evaluations, ethics approval was needed from the NMU Research Ethics Committee: Human (REC-H). Approval was granted by REC-H, validating the ethical nature of the study; the ethical clearance number associated with this research was H18-SCI-CSS-002 (Appendix A).

The next section looks at how the research methodology discussed was applied to the project. Providing details about the various chapters and how they link to the methodology used.

1.7 Dissertation Structure

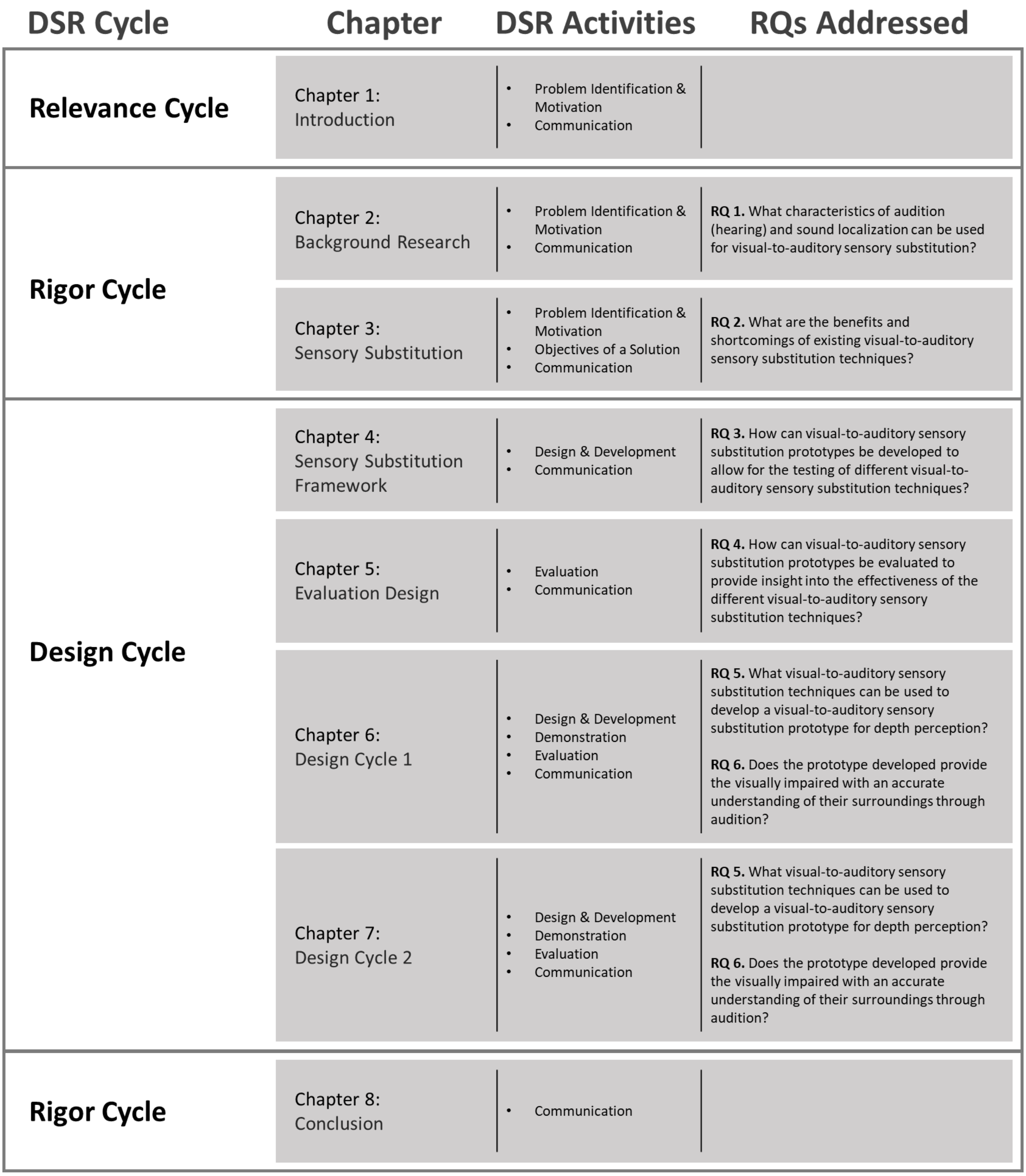
The research methodology should be used to guide the research, development and design of the project. This section gives an overview of how the DSR methodology was used to guide the project by illustrating how the DSR Cycles, the DSR Processes and the research questions link to the various chapters throughout this dissertation – this can be seen in Figure 1-3. The first column in Figure 1-3 shows which of the three DSR Cycles (Figure 1-2) are covered by which chapter. The second column specifies the actual chapter of the dissertation. The third column gives a list of the specific DSR activities (Figure 1-1) that each chapter covers. The fourth column shows which research questions (Section 1.4) are addressed in each chapter.

Chapter 1 – Introduction: Chapter one is an introduction to the research being done. It is primarily concerned with the relevance of the project and deals with the DSR Relevance Cycle. It covers the background relating to the application domain, identifies the research problem and the related research questions addressed throughout the project. It also discusses the research methodology used throughout the project.

Chapter 2 – Background Research: This chapter is part of the DSR Rigor Cycle. It is primarily concerned with the rigor of the project – in the case of this chapter, a rigorous understanding of background knowledge needed to understand visual-to-auditory sensory substitution. The aim being to provide a better understanding of how humans use audition to perceive their surroundings. This information can then be used to make informed decisions regarding the implementation of the sensory substitution prototypes.

Chapter 3 – Sensory Substitution: This chapter is also part of the DSR Rigor Cycle, and is primarily concerned with the rigor of the project. The focus of this chapter is sensory substitution. Chapter 3 discusses what visual-to-auditory sensory substitution is in more detail than the background section (Section 1.1), then it looks at some existing systems to understand the existing knowledge base.

Chapter 4 – Sensory Substitution Framework: This chapter begins with identifying objectives for the solution (prototypes) developed using the knowledge base from prior chapters. It then goes on to deal with the design and implementation of the system. The focus of this chapter is the Sensory Substitution Framework developed, and how the framework can be used to implement sensory substitution algorithms. This framework provides the design structure for the sensory substitution prototypes developed.



DSR Cycle	Chapter	DSR Activities	RQs Addressed
Relevance Cycle	Chapter 1: Introduction	• Problem Identification & Motivation • Communication	
Rigor Cycle	Chapter 2: Background Research	• Problem Identification & Motivation • Communication	**RQ 1.** What characteristics of audition (hearing) and sound localization can be used for visual-to-auditory sensory substitution?
	Chapter 3: Sensory Substitution	• Problem Identification & Motivation • Objectives of a Solution • Communication	**RQ 2.** What are the benefits and shortcomings of existing visual-to-auditory sensory substitution techniques?
Design Cycle	Chapter 4: Sensory Substitution Framework	• Design & Development • Communication	**RQ 3.** How can visual-to-auditory sensory substitution prototypes be developed to allow for the testing of different visual-to-auditory sensory substitution techniques?
	Chapter 5: Evaluation Design	• Evaluation • Communication	**RQ 4.** How can visual-to-auditory sensory substitution prototypes be evaluated to provide insight into the effectiveness of the different visual-to-auditory sensory substitution techniques?
	Chapter 6: Design Cycle 1	• Design & Development • Demonstration • Evaluation • Communication	**RQ 5.** What visual-to-auditory sensory substitution techniques can be used to develop a visual-to-auditory sensory substitution prototype for depth perception? **RQ 6.** Does the prototype developed provide the visually impaired with an accurate understanding of their surroundings through audition?
	Chapter 7: Design Cycle 2	• Design & Development • Demonstration • Evaluation • Communication	**RQ 5.** What visual-to-auditory sensory substitution techniques can be used to develop a visual-to-auditory sensory substitution prototype for depth perception? **RQ 6.** Does the prototype developed provide the visually impaired with an accurate understanding of their surroundings through audition?
Rigor Cycle	Chapter 8: Conclusion	• Communication	

Figure 1-3: Overview of the dissertation structure and how it links to the DSR methodology

Chapter 5 – Evaluation Design: A standardised evaluation design is needed to perform consistently repeatable evaluations on different sensory substitution techniques. This chapter reviews the evaluation procedure developed and used for this project. It also discusses the metrics gathered; these can be used to evaluate how well the prototypes achieved the objectives identified in Chapter 4.

Chapter 6 – Design Cycle 1: The DSR Design Cycle deals with designing, developing and evaluating an artefact; in this case a sensory substitution prototype. The design, development and evaluation are based on the objectives identified in Chapter 4. This chapter discusses the first round of the Design Cycle. First, a Baseline Prototype is implemented based on an existing system, then the first iteration from the Baseline Prototype is developed – Prototype 1. A comparative study is then performed, comparing the Baseline Prototype to Prototype 1. As a result, a set of recommendations is generated.

Chapter 7 – Design Cycle 2: This chapter discusses the second round of the DSR Design Cycle. Based on the objectives identified in Chapter 4, as well as the recommendations generated from the first Design Cycle (Chapter 6), another prototype is developed – Prototype 2. The design and development of Prototype 2 are discussed in this chapter. A comparative study is then performed, comparing the Baseline Prototype to Prototype 2. As a result, a set of recommendations is generated.

Chapter 8 – Conclusion: This chapter concludes the dissertation with an overview of the project and how the various research questions were addressed. Discussing the project's findings, what was learnt and providing suggestions for future research.

Finally, the purpose of the dissertation is to communicate the problem, the research done on the problem, and the artefact created as a result of the research. For this reason, the communication DSR activity is part of every chapter in the dissertation.

Chapter 2: Background Research

2.1 Introduction

Sensory substitution involves using one sense as a substitute for another (Lenay, Gapenne, Hanneton, Marque, & Genouëlle, 2003; Meijer, 1992; Ward & Wright, 2014). In the case of visual-to-auditory sensory substitution, it involves using audition (hearing) as a substitute for vision, allowing one to "see" through sound. Before investigating the research domain of sensory substitution, it is helpful to understand how humans perceive sound.

This chapter aims to gain a clear understanding of how the auditory system functions, covering topics such as sound localization – the ability to determine the position of sound sources in space. This will give insight into answering the *Main research question*. More specifically, addressing:

> **RQ 1**: "*What characteristics of audition (hearing) and sound localization can be used for visual-to-auditory sensory substitution?*"

This chapter is part of the DSR Rigor Cycle, as it involves understanding the research domain. The chapter's focus is on the background knowledge useful for understanding visual-to-auditory sensory substitution. Hence, throughout this chapter, a number of potentially useful characteristics of human perception are discussed.

2.2 Psychoacoustics

Sound is a mechanical wave which hits the ear and then is transformed into a series of neural action potentials. The brain then perceives these as sound. Psychoacoustics is the scientific study of sound perception. This section focuses on topics in psychoacoustics, including the anatomy of the human ear, the range of human hearing and how humans can understand where a sound is coming from (Zwicker & Fastl, 1999).

2.2.1 Anatomy of the Human Ear

Humans perceive sound through their ears, which have three main components (Figure 2-1). The outer ear, middle ear and the inner ear. The outer ear starts at the visible part of the ear – called the pinna – and ends at the eardrum (tympanic membrane). The pinna is focused on channelling the sound to the eardrum in addition to amplifying the sound. An important characteristic of the pinna is its asymmetrical shape; this results in the same sound being channelled differently depending on where the sound comes from (Zwicker & Fastl, 1999).

The middle ear is a small airtight chamber following the eardrum, this chamber contains three small bones: the malleus, incus and stapes, collectively called the ossicles. The malleus is against the eardrum, with the stapes against the oval window of the cochlea. The incus connects the malleus and stapes. The purpose of the ossicles is to transform the acoustic energy to kinetic (moving) energy, additionally working like mechanical levers to amplify the pressure delivered to the oval window of the cochlear (Zwicker & Fastl, 1999).

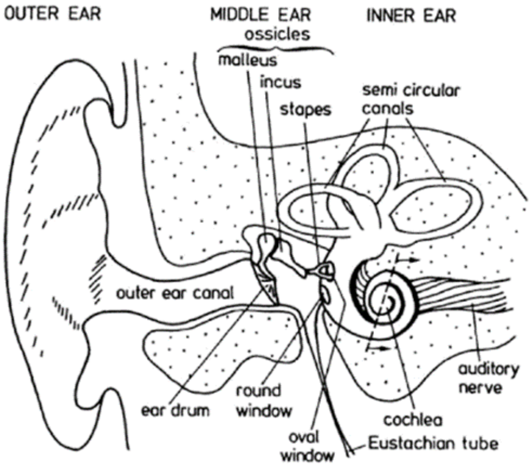


Figure 2-1: Schematic of the outer, middle and inner ear (Zwicker & Fastl, 1999)

The inner ear contains the cochlea, a fluid-filled spiral tube, which contains the organ of Corti, running the length of the cochlear duct. Inside the organ of Corti is the basilar membrane; the basilar membrane is tonotopic, meaning different sections of the basilar membrane are sensitive to different frequencies. Different frequencies of sound penetrate to different distances into the cochlea, allowing one to tell the difference between frequencies. Through vibrations in the inner ear, displacement of the cochlear fluid and movement of the hair cells, the organ of Corti produces electrochemical signals. These electrochemical signals result in the neuronal encoding of sound travelling along the auditory nerve to the brain – these are spatiotemporal patterns containing information about the sound (Hudspeth, 2014). The auditory nerve has approximately 31,000 to 32,000 nerve fibres (Spoendlin & Schrott, 1989).

In summary, pressure waves (sound) are channelled by the pinna to the eardrum, causing the eardrum to vibrate. This vibration is amplified and transferred to the cochlea by the ossicles – the small bones. Depending on the frequency of the sound, the vibration travels different distances into the cochlea. This causes displacement of the cochlear fluid and movement of the hair cells, resulting in the organ of Corti producing electrochemical signals. These electrochemical signals are then interpreted by the brain as sound.

2.2.2 Audible Range

Sound is one's perceptual experience of pressure waves hitting one's ear. It should be noted that both the brain and the ears are involved in the experience of sound perception, as without the brain perceiving the sound, there are only pressure waves (Zwicker & Fastl, 1999). Sound is comprised of two primary components, namely, frequency and amplitude. Frequency consists of the low and high tones measured in hertz (Hz), and amplitude is the intensity of the sound (loudness) measured in sound pressure level (SPL).

For amplitude, humans can hear sounds as soft as 0 dB SPL (roughly 0.00002 Pa), and as loud as 120 dB SPL (20 Pa), with 130 dB SPL (200 Pa) being the threshold for pain (Thompson, 2005). A change of 1dB SPL is generally considered the smallest average change in level that one can hear in a controlled environment (using headphones in an isolated environment). This perceivable change is known as the just-noticeable difference (JND), 3dB SPL is a more real-world JND (Thompson, 2005). It has also been shown that 8 hours of continuous listening to sound at 90 dB can cause damage to the ears, with 1 minute of a 110 dB sound causing hearing loss, and any amount of time hearing a 140 dB sound causes immediate, irreversible damage to one's hearing (Velázquez, 2010).

For frequency, humans can hear between 20Hz (Low) and 20,000Hz (High). Below 20 Hz is known as infrasound and above 20,000 Hz is known as ultrasound. One may be able to hear slightly into the infrasound and ultrasound extremes with ideal lab setups; however, these extremes are not commonly discussed. It should be noted that hearing deteriorates with age – especially the ability to hear high frequencies. The result of this is that most adults cannot hear above 16,000Hz (Thompson, 2005).

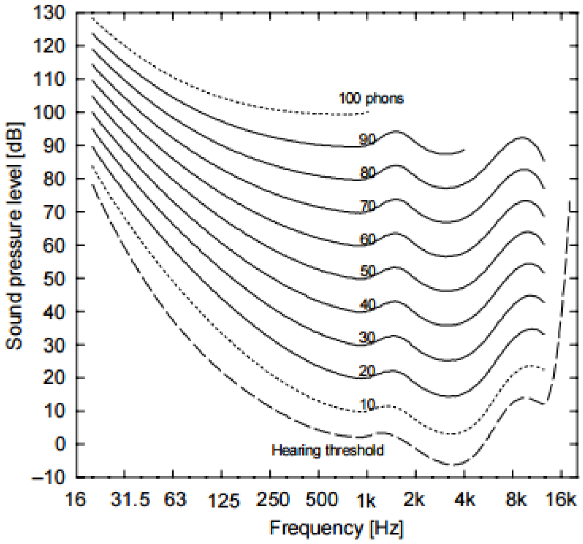


Figure 2-2: Estimated new equal-loudness contours (Suzuki, 2003)

Studies have shown that at a constant volume (loudness) as the frequency of a sound changes, how loud one perceives the sound to be (perceived loudness) also changes. Since perceived loudness changes with frequency, it is useful to have a measure of perceived loudness rather than actual loudness, this measurement is known as the phon. The phon measurement of perceived loudness is based on the reference tone of 1000 Hz, meaning x phon sounds as loud as a 1000 Hz tone at x dB. For example, 20 phon sounds as loud as a 1000 Hz tone at 20 dB (Howard & Angus, 2009; Thompson, 2005).

The equal-loudness contours (Figure 2-2) show what a listener perceives as a constant level of loudness when presented with pure tones over the frequency range of human hearing. From Figure 2-2, it can be seen that humans do not perceive loudness consistently over the frequency range. Figure 2-2 shows humans are less sensitive to low frequencies and extremely high frequencies. For example, consider the 40 phon equal-loudness contour; at 1000 Hz (1k on the graph), the perceived loudness is 40 phon and the actual loudness is 40 dB (because 1000 Hz is the reference tone), at about 20 Hz, the perceived loudness is 40 phon (because perceived loudness stays the same along the equal-loudness contour), however the actual loudness is 100 dB. This means that to perceive a 20 Hz sound as being as loud as a 1000 Hz sound at 40 dB, the sound needs to be turned up by 60 dB (100 dB – 40 dB). From this example, it can be seen that the equal-loudness contours show how much one needs to turn up or down the volume in order for a given frequency to be perceived as having the same perceived loudness as 1000 Hz reference tone at a given loudness. This perceived versus actual loudness explains why some musical instruments when set at the same level, sound softer than other instruments, for example, the bass guitar sounding softer than a normal guitar. It should be noted that the equal-loudness contours were done with tones, so one cannot simply extrapolate the graph to necessarily include more complex sounds (Howard & Angus, 2009; Suzuki, 2003; Thompson, 2005).

2.2.3 Sound Localization

Sound localization is the ability of a listener's brain to determine where a sound source is located in three-dimensional space. The localization of sound is achieved through a number of processes, starting with the sound being distorted by the head and pinna (Section 2.2.1) and ending with the brain interpreting the distortion among other features of the sound (Schiller & Brown, 2013).

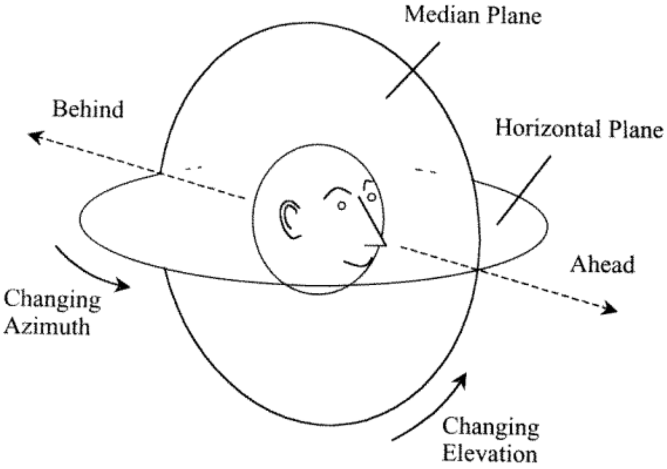


Figure 2-3: Coordinate system for sound localization (Plack, 2005)

Figure 2-3 shows the coordinate system for sound localization. Localization in the horizontal plane is measured in azimuth (angle measured from directly ahead), with positive azimuth on the left and negative azimuth on the right. For the median plane (vertical plane), the angle is measured in elevation from straight ahead, with positive elevation being upward and negative elevation being downward. Based on this coordinate system, a sound coming from straight ahead measures zero degrees azimuth and zero degrees elevation (Plack, 2005).

2.2.3.1 Azimuth Localization

Determining the azimuth is done primarily using two techniques, the interaural time difference (ITD) and the interaural level difference (ILD). The ITD and the ILD are binaural cues, meaning that they use both ears. The interaural time difference is simply the time difference between a sound hitting the left ear versus the right ear, with the JND being 0.01-0.02 ms for pure tones (Shackleton, Skottun, Arnott, & Palmer, 2003).

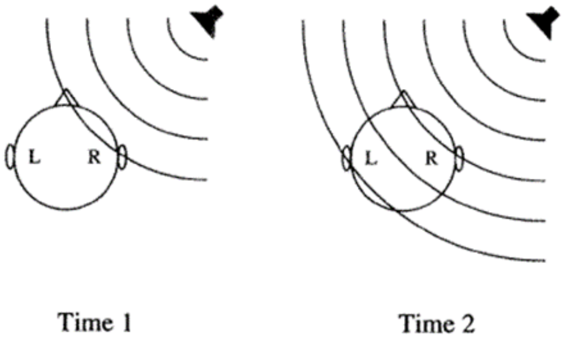


Figure 2-4: A top-down view of the head with a sound source on the right (Plack, 2005)

As illustrated in Figure 2-4, a sound coming from the right will hit the right ear before the left (Thompson, 2005). Interaural time difference is used for sounds below 1500 Hz (Shackleton et al., 2003; Smith, Price, Knudsen, Tollin, & Wagner, 2014), since higher frequency sounds do not wrap around the head as well as lower frequency sounds, forming an acoustic shadow where sounds are only heard on one side of the head (Wolfe et al., 2014).

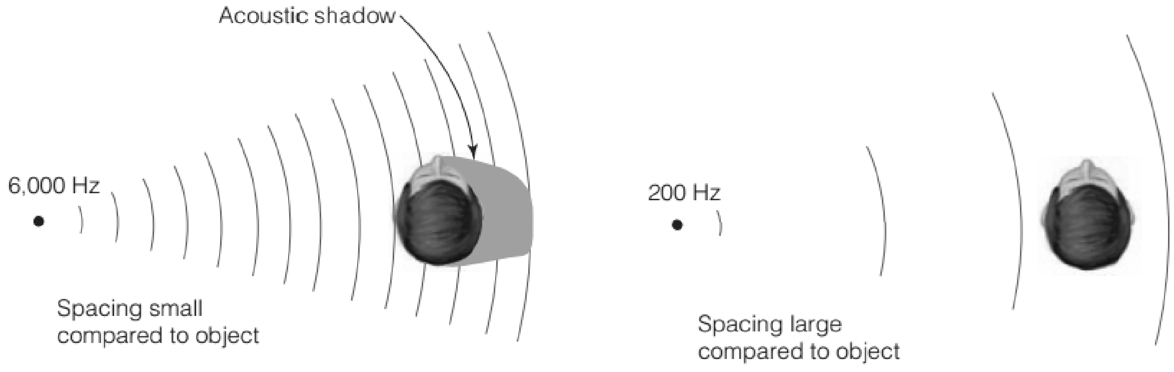

Acoustic shadow
6,000 Hz
Spacing small compared to object
200 Hz
Spacing large compared to object

Figure 2-5: Acoustic shadow for high and low-frequency sounds (modified) (Wolfe et al., 2014)

Interaural level difference (ILD) is the difference in loudness (level) of a sound between each ear. The ILD is negligible below 200 Hz since there is virtually no acoustic shadow, however, as the frequency becomes higher, the ILD increases to the point where the ILD between each ear is roughly 20 dB at 6000 Hz (Musiek & Chermak, 2013). For example, in Figure 2-5 one can see the illustrated acoustic shadow of a 6000 Hz tone coming from the left-hand side. In this example, the sound would be louder in the listener's left ear than in the listener's right ear, as a result of the acoustic shadow. Due to the increase in interaural level difference (ILD) with frequency, for sounds above 1500 Hz, the ILD is primarily used to determine a sound's azimuth (Shackleton et al., 2003; Smith et al., 2014), with 1 dB being the JND for ILD (Schiller & Brown, 2013).

The azimuth can be localized to within 1 to 2 degrees when the sound source is directly in front of the listener (zero degree azimuth); localization gets progressively worse as the azimuth approaches 90 degrees (Makous & Middlebrooks, 1990; Musiek & Chermak, 2013; Schiller & Brown, 2013). Listeners generally compensate for this loss in accuracy with increased azimuth angle by rotating their head toward the direction of the sound source (Musiek & Chermak, 2013).

2.2.3.2 Elevation Localization

To localize a sound in the median plane, and hence determine the elevation of a sound, humans primarily use spectral cues. Spectral cues are monaural (single ear) cues based on the differences in the frequency spectrum of a sound; these differences are caused by the sound waves interaction with the shoulders, head and most importantly, the pinna. As discussed in Section 2.2.1, the shape of the pinna is asymmetrical, resulting in different spectral cues as the location of a sound source changes, allowing the listener to determine the sound source's elevation (Goldstein, 2009).

Listeners can determine the elevation of a sound given it is sufficiently complex (Roffler & Butler, 1968a); this is why broadband sounds (sound containing many frequencies) are often used in sound localization studies. Listeners are not able to proficiently locate the elevation of pure tones (Roffler & Butler, 1968a). Humans do, however, have a natural tendency to associate higher pitched tones with higher vertical position, and correspondingly, lower pitched tones with lower position (Roffler & Butler, 1968b). For complex sounds, some studies suggest an elevation accuracy of 30 degrees (Schiller & Brown, 2013), while other studies found that the smallest errors averaged across subjects was 3.5 degrees (Makous & Middlebrooks, 1990). To gain a better accuracy for elevation localization, high-frequency sounds (above 4000 – 5000 Hz) are needed (Goldstein, 2009; Spagnol, 2012). Localization accuracy of the elevation was better than that of azimuth for sound sources on the periphery (Makous & Middlebrooks, 1990).

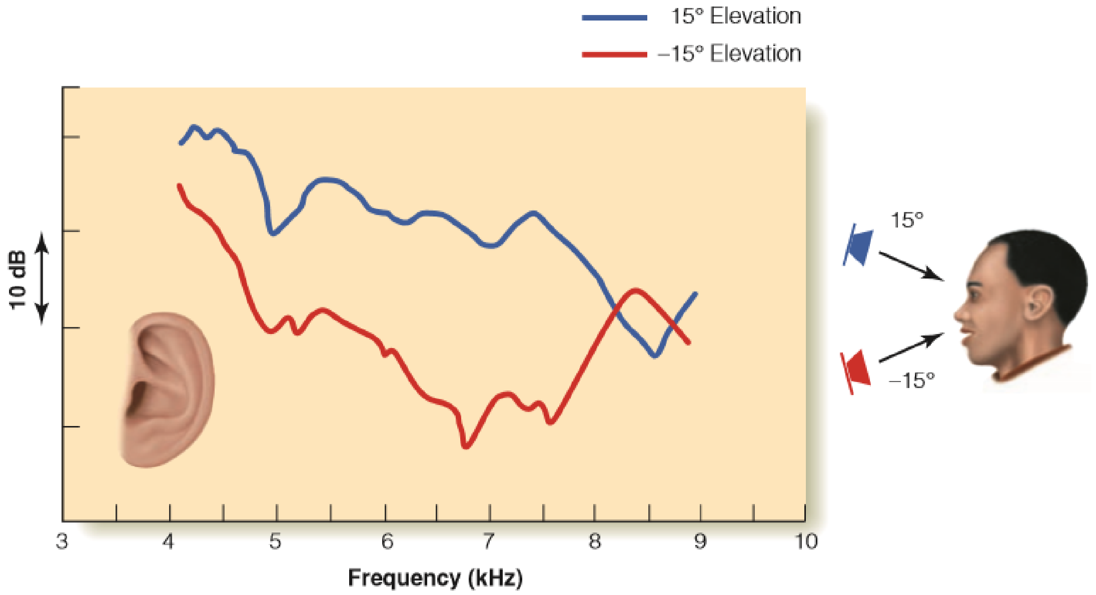


Figure 2-6: Frequency spectrum for the same sound at two elevations (Goldstein, 2009)

As shown in Figure 2-6, when a broadband sound is presented at 15 and -15 degrees elevation, the frequency spectrum for each elevation varies significantly, even though the sound remained the same. This change in the frequency spectrum is what allows the listener to determine the sound source's elevation, in addition to determining if a sound is from in front or behind.

When the pinna is distorted using moulds to change its shape, subjects immediately become less accurate in locating a sound source's elevation (Goldstein, 2009; Oldfield & Parker, 1984). These moulds do not result in an error in locating a sound source's azimuth (Oldfield & Parker, 1984).

However, given two to three weeks with the distorted pinna, detecting sound source elevation can be relearnt (Hofman, Van Riswick, & Opstal, 1998; Schiller & Brown, 2013). Once a listener had adapted to the distorted pinna, one would assume that removing the moulds (i.e. restoring the pinna's original shape) would result in a decrease in the accuracy of their elevation localization. What happens instead is that the listener is now able to locate a sound source's elevation to the original degree of accuracy with the pinna's original shape and distorted shape (Goldstein, 2009; Hofman et al., 1998). This implies that the listener retains their understanding of the new and old spectral cues for elevation.

2.2.3.3 Determining Distance

The primary cue for distance is loudness, where the azimuth and elevation provide directional information. Due to sound's attenuation with distance, generally, the softer a sound, the further away the sound source (Begault, 2000; Spagnol, 2012). However, for familiar sounds, multiple cues are used (generally visual-aural cues). For example, to the listener, an aeroplane and a car could be equally loud, but through the understanding that aeroplanes are generally louder than cars, the listener would likely conclude the aeroplane is further away than the car. For this reason, familiarity with a sound source allows one to determine distance more accurately by integrating multiple cues (Begault, 2000). However, when hearing an unfamiliar sound source, the primary cue used is loudness.

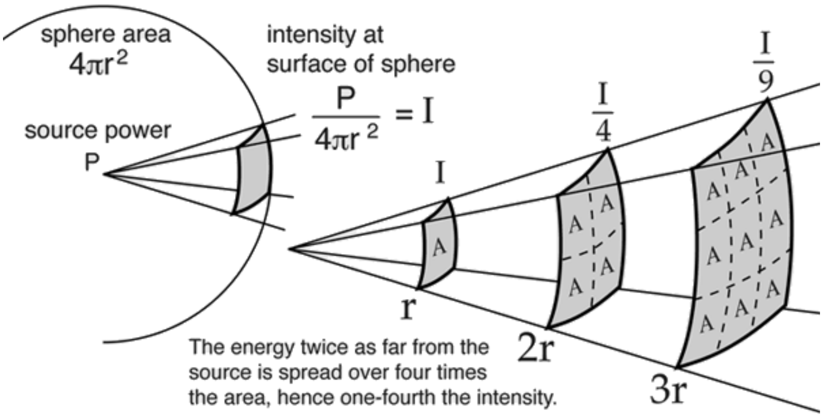

sphere area
$4\pi r^2$
intensity at surface of sphere
$\frac{P}{4\pi r^2} = I$
source power
P
I
$\frac{I}{4}$
$\frac{I}{9}$
A
r
2r
3r
The energy twice as far from the source is spread over four times the area, hence one-fourth the intensity.

Figure 2-7: Visual representation of the Inverse Square Law (Nave, 2017b)

Under anechoic (free from echo) conditions, the inverse square law can be used to understand the relationship between a sound source's distance and its loudness (Begault, 2000). The inverse square law for sound states that intensity is inversely *proportional* to the square of the distance from the source. The reason for this is that the intensity is spread over an area that is proportional to the square of the distance from the source, this can be seen visually in Figure 2-7.

The equation for the inverse square law over a spherical surface is as follows:
I : Sound intensity
r : Distance from source
P : Source power
A : Area

$$I \propto \frac{1}{r^2}$$

Equation 1

For a spherical surface, $A = 4\pi r^2$:

$$\therefore I = \frac{P}{4\pi r^2}$$

Equation 2

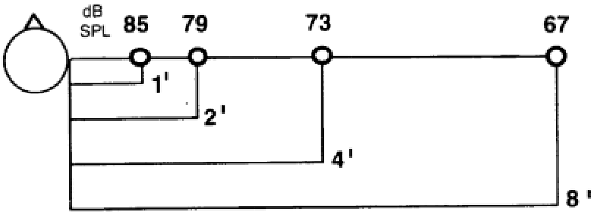


Figure 2-8: Relationship between a sound source's distance and loudness (Begault, 2000)

Using the inverse square law, it can be calculated that an omnidirectional sound source's loudness will drop by 6 dB for each doubling in distance (Begault, 2000), as shown in Figure 2-8.

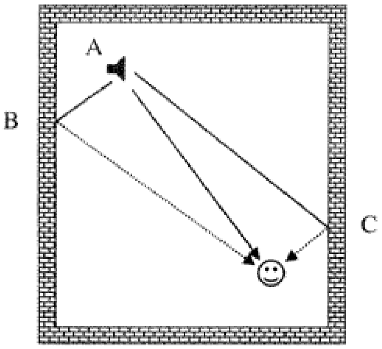


Figure 2-9: Sound source reflections (Plack, 2005)

In many scenarios, sound may be reflected off several surfaces. The problem this causes is that reflected sound can give contradictory directional information to the actual sound source, as illustrated in Figure 2-9 (Plack, 2005). The auditory system deals with this by using the precedence effect, that is, it assumes that the first sound is the original and the rest are reflections to be ignored

(Shinn-Cunningham, 2013). The precedence effect is based on the concept that the direct sound from a sound source has the least distance to travel and so will be heard first.

2.2.3.4 Head-Related Transfer Function

In Section 2.2.3.2, spectral cues were discussed, and it was noted that different positions of a sound source result in different frequency spectrums (Figure 2-6). This is caused by the sound interacting with the head, torso and pinna. A Head-Related Transfer Function (HRTF) characterises how an ear captures a sound propagating from a single point in space; this includes the transformations of the sound due to one's body, head, pinna and ear canal (Potisk & Svenšek, 2015). A Head-Related Transfer Function (HRTF) describes, for a single sound source at a specific position, the differences in the frequency spectrum of the sound heard by one's ear versus the actual sound source (Bosun, Xiaoli, Rao, & Liang, 2007; Cheng & Wakefield, 2001; Hadad, Fishman, Hadad, & Gannot, 2014; Sodnik, Umek, Susnik, Bobojevic, & Tomazic, 2004). As discussed in Section 2.2.3.2, these monaural spectral cues are primarily for determining a sound source's elevation.

To localize elevation, azimuth and distance for a single sound source at a specific point in space, two HRTFs are needed, one for each ear. Using two HRTFs allows for taking into account the many factors that influence the localization of a sound source, including spectral cues, interaural time difference (ITD) and the interaural level difference (ILD), among others (Bosun et al., 2007; Cheng & Wakefield, 2001). Since two HRTFs are linked to a sound source at a single point in space, numerous pairs of HRTFs are needed to accurately simulate sound from any point in three-dimensional space. HRTFs are very individual due to the differences in people's body, head and pinna shape. For this reason, to accurately simulate a sound source's position in three-dimensional space for a given person, one needs numerous pairs of HRTFs, and those HRTFs need to be measured specifically for that person (using interpolation between measured HRTFs to fill in the gaps). Measurements are generally performed by placing small microphones in the ears, comparing the frequency spectrum measured at the microphones with the actual sound source's frequency spectrum (Boynton, 2008).

For the most accurate three-dimensional sound simulation the HRTFs should be measured for a specific individual, but often, this is not possible. For example, if one would like to have positional sound in a game, it would be impractical to make every player obtain their HRTFs. For this reason, there are generalised HRTFs, which are commonly used in software such as games. These generalised HRTFs do not provide as accurate positional sound as personalised HRTFs do, however for many purposes, they are good enough.

2.2.4 Sound Interference

When multiple sounds from various sound sources are superimposed (combined), they interfere with one another causing constructive and destructive interference (Figure 2-10). Constructive interference is when two in-phase waves combine; this forms a wave with the combined amplitude of both the waves, resulting in a louder sound. Constructive interference is why a large band in harmony sounds much louder than a single person – when all other conditions are kept fixed. Destructive interference is when two out of phase waves combine, since their amplitudes are opposite, when the waves are combined, they cancel one another out. This produces a softer sound, and if the waves are perfectly out of phase, the result is no sound as they perfectly cancel one another out. Destructive interference is how noise-cancelling headphones work. They listen to the surrounding environment and attempt to play the exact opposite (inverse) of the sound produced by the environment, causing the sounds to cancel out through destructive interference.

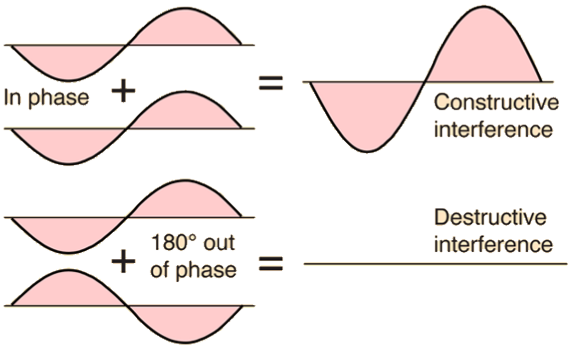


Figure 2-10: Superposition of waves seen on the right (Nave, 2017a)

When dealing with digital audio – audio generated using an electronic device such as a computer – the audio has to be within a certain upper and lower amplitude. Due to these limitations, a phenomenon known as audio clipping can occur (Figure 2-11).

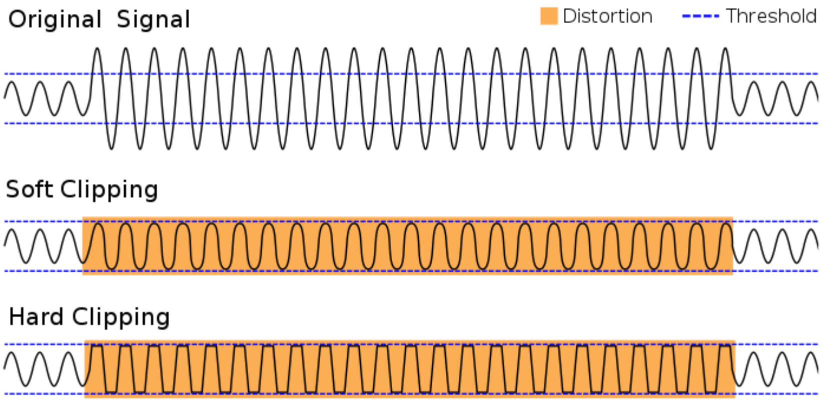


Figure 2-11: Audio Clipping

This happens when constructive interference from multiple sounds results in the amplitude exceeding those upper and lower amplitude limits. Audio clipping is when the signal peaks that exceed the threshold are completely cut off (hard clipping), i.e. resulting in signal peaks being squared off as shown at the bottom of Figure 2-11. When clipping occurs audio information is lost, and this information is not recoverable. Hard clipping generally sounds extremely harsh and unpleasant.

2.2.5 Discussion of related technologies

There are several three-dimensional positional audio libraries, which allow one to place sound sources in a three-dimensional virtual world. Software that allows for positional audio includes OpenAL (Open Audio Library) (OpenAL, 2017), Unity (Unity, 2017) and browsers such as Chrome supporting positional audio through the Web Audio API (Mozilla Developer Network, 2017). These types of software implement many of the sound localization techniques discussed in Section 2.2.3, to create accurate positional sound.

2.3 Conclusion

This chapter explored the human auditory system. It investigated different audition characteristics with the aim of finding characteristics useful for visual-to-auditory sensory substitution – addressing **RQ 1**. These characteristics included the anatomy of the human ear, how humans perceive loudness over the frequency spectrum, how humans localize sounds in space and the concept of sound interference. The next chapter looks at sensory substitution.

Chapter 3: Sensory Substitution

3.1 Introduction

The previous chapter looked at the auditory system. With the knowledge gained from the previous chapter, the focus is now turned to how these insights have been used to compensate for the loss of a sense, i.e. vision. A commonly studied compensatory technique is called sensory substitution. Sensory substitution is a non-invasive technique – meaning no surgery is required – for circumventing the loss of one sense by feeding its information through another sensory channel (Renier & De Volder, 2005). Hence, devices, which provide information about one's surroundings, typically related to one sense (e.g. vision) through stimulation of another sense (e.g. audition) are referred to as sensory substitution devices (SSDs) (Bermejo, Di Paolo, Hüg, & Arias, 2015).

Sensory substitution is the research domain for this project. The DSR Rigor Cycle requires an understanding of the existing knowledge base; this includes an investigation of the benefits and shortcomings of existing systems. In doing so, this chapter answers:

> **RQ 2**: "*What are the benefits and shortcomings of existing visual-to-auditory sensory substitution techniques?*"

This chapter considers the types of SSDs; it looks at how sensory substitution relates to the brain and investigates existing visual-to-auditory sensory substitution devices. From the existing SSDs, an algorithm is chosen as a base for developing the Baseline Prototype. The Baseline Prototype can then be used as a point of reference for improvement in later chapters.

3.2 Types of Sensory Substitution Techniques

Throughout literature, when discussing sensory substitution systems, different conventions are used by different authors. For example, auditory-to-vision sensory substitution, visual-to-auditory sensory substitution, auditory vision substitution and auditory sensory substitution – among other variations – can all be referring to the same type of sensory substitution. The convention used throughout this project is [sense being substituted]-to-[sense being used as substitute]; i.e. visual-to-auditory sensory substitution would mean the visual sense is being substituted and the auditory sense is being used as the substitute. Visual-to-auditory sensory substitution systems are generally used to give the visually impaired "sight" through sound; conveying what is normally seen visually through the auditory sense.

There are many different types of vision substitution techniques, common among these are the visual-to-tactile and visual-to-auditory techniques. Within each of these techniques, there are different ways of achieving sensory substitution. For example, visual-to-tactile sensory substitution systems could use stimulation of the tongue through electrotactile stimulators, or stimulation of one's back through vibrotactile stimulators (Kaczmarek, Webster, Bach-y-Rita, & Tompkins, 1991). One example of the electrotactile tongue stimulation is the BrainPort, which provides visual-to-tactile sensory substitution by stimulating the tongue using an electrode array (Danilov, Tyler, & Danilov, 2006).

This project will primarily focus on visual-to-auditory sensory substitution, that is, using audition as a substitute for vision, i.e. using sound to give the visually impaired "sight". The different implementations of visual-to-auditory sensory substitution systems are discussed later in this chapter.

3.3 Sensory Substitution and the Brain

The brain lies behind the ability to perceive the world. Without it, one's eyes and ears – among other sensory organs – would have little purpose. The brain has the ability to turn the feedback from these sensory organs into sensations, allowing one to experience sight, sound and touch (Beckman, 2014).

Different regions of the brain are used to process different types of sensory inputs. The visual cortex is responsible for transforming the retinal signals into the visual experience commonly referred to as sight. The auditory cortex similarly is responsible for transforming signals from the cochlea into the auditory experience commonly referred to as hearing.

Neuroimaging studies have shown that the part of the brain responsible for vision, is involved in depth perception when audition is used for vision substitution. This means that the visual cortex can be used to process depth information using senses other than vision (Renier et al., 2005). The brain is very plastic, as Robert A. Beckman stated, "We don't see with our eyes, we see with our brain" (Beckman, 2014). Because of this, there have been many devices which use audition as a vision substitution (Wright, Margolis, & Ward, 2015).

Looking at visual-to-auditory sensory substitution, it is likely also important to consider the visual and the auditory links to the brain; that is the optical nerve and auditory nerve respectively. The optical nerve and auditory nerve essentially determine the amount of visual and auditory information the brain can receive at any given moment. Considering that the optical nerve has on average around 1,158,000 fibres (Jonas, Schmidt, Müller-Bergh, Schlötzer-Schrehardt, & Naumann, 1992), where the auditory nerve has around 31,500 fibres (Spoendlin & Schrott, 1989); this suggests that the amount of information that can be transferred to the brain via the optical nerve is approximately 37 times that of the auditory nerve (Capelle, Trullemans, Arno, & Veraart, 1998). Given these information transfer limits, it is unreasonable to expect that visual-to-auditory sensory substitution systems will match that of the visual system. It is not infeasible to think that intelligent ways of getting around this limitation can be developed – for example, finding compression techniques that the brain can naturally decode. The fairly large number of auditory nerve fibres also suggests that there is likely much room for improvement on existing SSD, some of which are discussed below.

3.4 Visual-to-Auditory Sensory Substitution Devices

The general process for visual-to-auditory SSDs is to capture an image in through a camera, process the image via an algorithm, which performs an image-to-sound mapping, then output the sound through earphones or headphones. There are several existing solutions for the visually impaired, specifically for understanding their surroundings, which includes a number of wearable assistive devices (Velázquez, 2010). Due to the nature of the project, this section is focused on sensory substitution devices (SSDs), which use visual-to-auditory sensory substitution techniques. In the field of visual-to-auditory sensory substitution, there are several SSDs that have been developed over the years by a variety of researchers. This section will review algorithms commonly discussed in literature; these include the vOICe, the PSVA and the MeloSee algorithm.

3.4.1 The vOICe

The vOICe was the first SSD to do visual-to-auditory sensory substitution (Meijer, 1992; Ward & Wright, 2014). It was originally developed in 1992 and has been the subject of many sensory substitution studies. The way the vOICe works is by taking a grayscale image captured by a head-mounted camera and converting it into a soundscape (an image made of sound). The three primary components of the vOICe are a camera (usually placed on glasses) to capture the visual data, a computer to convert the image into a soundscape, and headphones to play the soundscape to the user (Figure 3-1). The default encoding used to create a soundscape is mapping horizontal space to time (from left to right over 1s – i.e. using a scanline technique), vertical space to pitch (the higher a pixel, the higher the pitch), and brightness to loudness (the louder, the brighter). Depth perception is claimed to emerge and gradually improve through experience with the system (Brown, Macpherson, & Ward, 2011; Haigh, Brown, Meijer, & Proulx, 2013), although the depth perception is likely comparable to depth perception using one eye.

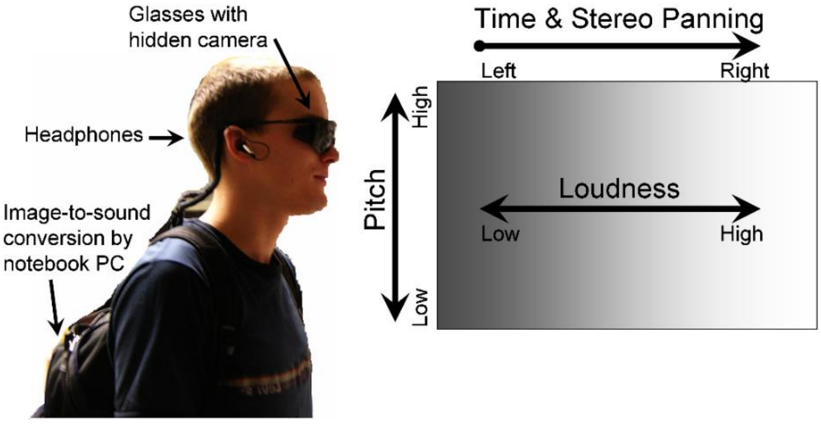


Figure 3-1: An illustration of the sensory substitution device and its conversion principles. (Proulx, Stoerig, Ludowig, Knoll, & Schnitzler, 2008)

The original resolution of the vOICe was 64 x 64 pixels (Meijer, 1992), however, this has been increased over time. More recently, the effective resolution of the soundscape is 11,264 auditory pixels (176 x 64 pixels) in the default setting (Haigh et al., 2013; Meijer, 2017). Studies have determined that participants having a relatively large amount of training (55 to 101 hours) with the vOICe, had a visual acuity of between 20/200 and 20/600 (based on the Snellen tumbling E paradigm) using a 66-degree field of view (Haigh et al., 2013). Where 20/200 vision means that one can read an eye chart from 20 feet away as well as someone with normal vision can read from 200 feet away. It is unclear if this is the maximum visual acuity achievable with the device, or a limitation based on the participant’s neural plasticity (the brain’s ability adapt to new information) and their experience with the device (Haigh et al., 2013).

The benefits of the vOICe are that it is non-invasive, cost-effective, available worldwide and research on it is ongoing (Meijer, 2017). The limitations of the vOICe are primarily in the refresh rate, which is one soundscape per second. One soundscape is equatable to one frame (still image), where for humans, showing roughly 24 frames per second creates the illusion of continuous motion seen in a video. The vOICe also does not directly deal with depth information, this in combination with the slow refresh rate, makes the vOICe less suitable for navigation. Another limitation of the vOICe is the relatively large training time (time taken to learn the system). An additional constraint is its use of only two dimensions (height and width) while humans can locate sounds in three spatial dimensions, as discussed in Section 2.2.3. A final limitation is that the playing of the soundscapes through headphones blocks out sound from the environment, something which the visually impaired rely on heavily. This final limitation can easily be overcome without any changes to the software, simply by making use of bone conduction headphones.

3.4.2 The PSVA

The Prosthesis for Substitution of Vision by Audition (PSVA) differs from the vOICe in a number of ways, the main difference being that rather than scanning the image from left to right playing one column of pixels at a time, the PSVA plays an entire image (i.e. frame) at a time. Due to the left to right scanning, the vOICe has a relatively slow refresh rate of 1Hz; this does not allow for fast sensory-motor interactions (Capelle et al., 1998). The PSVA aimed to improve this achieving refresh rates of 10Hz for the overall data-processing. As with most visual-to-auditory SSDs, the PSVA uses a head-mounted video camera, a computer to process the images captured by the camera, and a set of headphones to play the generated sound.

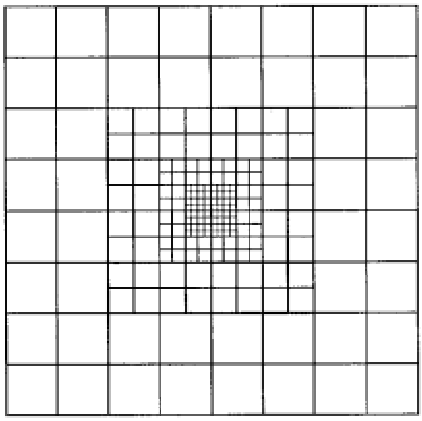


Figure 3-2: An artificial retina with four levels of resolution, amounting to a total of 208 pixels (Capelle et al., 1998)

The PSVA uses grayscale images, since colour images would increase the amount of information that needs to be encoded into sound. The PSVAs visual processing simulates two parts of the human visual system, namely lateral inhibition and graded resolution. Lateral inhibition – the idea that neighbouring neurons respond less when activated at the same time – increases the visual systems ability to detect edges on a surface. This process is simulated with an edge detection filter that is run on the image. Graded resolution – higher resolution at the fovea (the "centre" of the eye) than on the periphery – is implemented using what the authors call a "multiresolution artificial retina" (Capelle et al., 1998); this is illustrated in Figure 3-2. The "multiresolution artificial retina" is a noteworthy contribution of the PSVA system (Ward & Wright, 2014). The implementation of the PSVA provides a total of 124 pixels using the multiresolution artificial retina with two levels. The multiresolution artificial retina with two levels is an 8 x 8 pixel grid, with another a more dense 8 x 8 pixel grid covering the centre 4 pixels of the original 8 x 8 pixel grid. Hence $(8 \times 8) + (8 \times 8) - 4 = 124$ pixels.

The PSVA's auditory processing is done using a model of the human auditory system, using features such as the binaural intensity balance (ILD) and phase difference (ITD). Each pixel in the processed image is linked to a sinusoidal tone. The weighted sum of the tones then generates a single complex tone for each ear, where the intensity balance and phase difference is relative to the ear (Capelle et al., 1998). For horizontal localization, the PSVA uses intensity balance and phase difference. For vertical localization the PSVA associates higher pitch tones to the high parts of the image, and lower pitch tones to the lower parts of the image; this is based on the way humans localize sounds, associating higher pitched tones as being higher and lower pitched tones being lower as discussed in Section 2.2.3.2.

An overview of the algorithm is as follows: First, the grayscale image is passed through an edge detection filter, then the filtered image is converted into a multiresolution image (Figure 3-2). An auditory tone is assigned to each pixel in the image with the amplitude corresponding to the grey level of the associated pixel, and pitch corresponding to height. Then the weighted sum of all the tones is processed using the PSVA's model of the human auditory system, resulting in a complex sound being generated for the left and right ears approximately once every 10Hz.

The PSVA solves some of the issues of the vOICe system, namely the PSVA can convert a full image into sound in one go; rather than scanning the image left to right and converting it into sound one column at a time. This improves the system's use for navigating environments. The PSVA achieves this by making use of a broader range of human sound localization abilities. The PSVA also adds the "multiresolution artificial retina" which roughly mimics the way the human visual system functions. However, the shortfalls of the system – similar to the vOICe – include the long training period required to learn the system.

3.4.3 MeloSee

The MeloSee algorithm uses a different approach to the vOICe and PSVA. Using a grayscale image, the MeloSee algorithm uses a depth image; as stated by the title of the MeloSee paper, "Navigating from a Depth Image Converted into Sound" (Stoll et al., 2015). The depth image is downsampled using a process that mimics part of the human visual system; the result is an 8 x 8 pixel image. Based on the downsampled depth image, the output sound is generated in real-time by combining simple tones to form the overall sound.

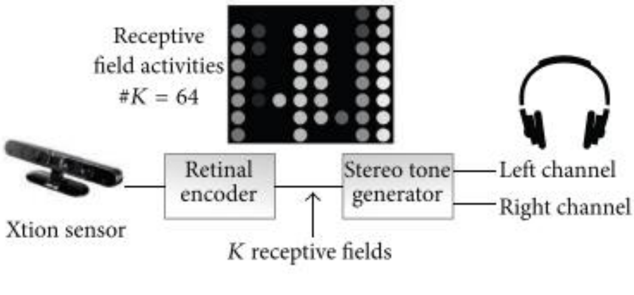


Figure 3-3: MeloSee SSD flowchart (Stoll et al., 2015)

The overview of the process is shown in Figure 3-3. The Xtion sensor is the depth camera that was used; the retinal encoder is what mimics the human visual system; the stereo tone generator converts the encoded (i.e. downscaled) depth image into sound; and the headphones play the sounds to the user through the left and right speakers.

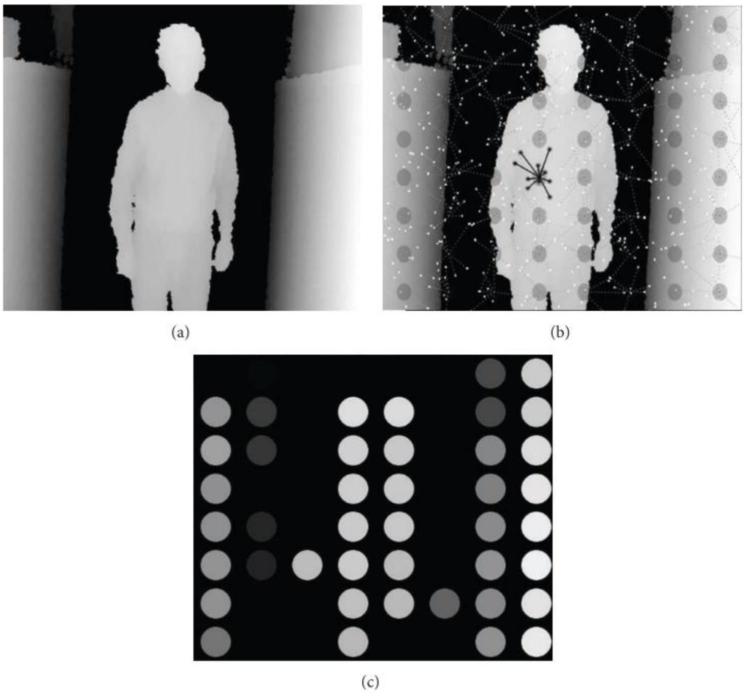


Figure 3-4: Grayscale depth image (a). Activation computation (b). RF activities (c). (Stoll et al., 2015)

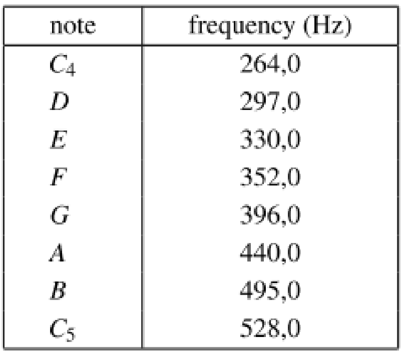
As is common with visual-to-auditory sensory substitution systems, the large amount of visual information is reduced (lossy compression) before being converted into sound. Figure 3-4 shows an overview of the visual processing part of the MeloSee algorithm that achieves this compression. The algorithm starts by receiving a depth image shown in Figure 3-4 (a), 64 receptive fields (RFs) – represented by the larger dots in Figure 3-4 (b) – are then spread out evenly across the depth image, forming an 8 x 8 grid. For each RF, $p = 10$ neighbouring pixels are randomly sampled from a 2D normal distribution. For each RF, the depth values ($l_{i,k}$) of the p samples are then averaged using Equation 3, resulting in an "activation" (Act_i) for that RF (the i^{th} RF) – this is an efficient way to approximate the mean of the pixels around a RF. The "activation" (Act_i) calculated for each RF from the retinal encoded image, as shown in Figure 3-4 (c).

$$Act_i = \frac{1}{255p} \sum_{k=1}^{p} l_{i,k}$$

Equation 3

For the sound generation aspect (Fristot, Boucheteil, Granjon, Pellerin, & Alleysson, 2012) of the MeloSee algorithm, tones on the ’just intonation’ scale are used, as shown in Table 3-1. For each pixel in the retinal encoded image the vertical position is related to pitch (C_4 to C_5 from bottom to top respectively), the horizontal position is related to left-right gain, and the distance (i.e. the depth value/”activation” of the RF) is inversely proportional to tone intensity. Hence, the sounds vary based on the “activation” (Act_i) – the approximate average depth – of the RF associated with the pixel, as well as its vertical and horizontal position of the pixel.

Table 3-1 Frequencies of the ’just intonation’ scale (Fristot et al., 2012)



note	frequency (Hz)
C_4	264,0
D	297,0
E	330,0
F	352,0
G	396,0
A	440,0
B	495,0
C_5	528,0

The sounds generated for each pixel are generated and then played in parallel; this means the whole soundscape is generated at once, rather than only part of the image being played (Fristot et al., 2012). This allows MeloSee to achieve approximately real-time image to sound conversion, with an audio update rate of 7.5 Hz (or 132ms) (Stoll et al., 2015), rather than the traditional 1 Hz (i.e. 1 fps) with algorithms such as the vOICe, which don’t convert the whole image at once (Meijer, 1992).

The benefit of using sound localization techniques for left-right positioning is that it enables the MeloSee algorithm to generate the entire soundscape in real-time. This real-time nature makes it suitable for navigation tasks. The MeloSee algorithm makes use of a depth camera rather than a standard grayscale or colour camera as most sensory substitution systems do – such as the PSVA and the vOICe. An additional benefit of the MeloSee algorithm is that it incorporates the depth information from the depth camera; this means it provides information for all three axes (x, y and z). One of the shortfalls of the MeloSee algorithm, as with sensory substitution systems in general, is the low resolution of the generated soundscape.

3.4.4 Notable Sensory Substitution Devices

EyeMusic is very similar to the vOICe, with the addition of using musical instruments to convey colour (Abboud, Hanassy, Levy-Tzedek, Maidenbaum, & Amedi, 2014). The musical instruments used are depicted in Figure 3-5 (A), and the algorithm process depicted in Figure 3-5 (B). As with the vOICe, this system also scans the image from left to right; EyeMusic additionally allows for the duration of the image scan to be adjusted by the user. This left to right scanning means that as with the vOICe, it does not allow for fast sensory-motor interactions. EyeMusic also requires a reasonable amount of training time to become familiar with the system – approximately 2-3 hours.

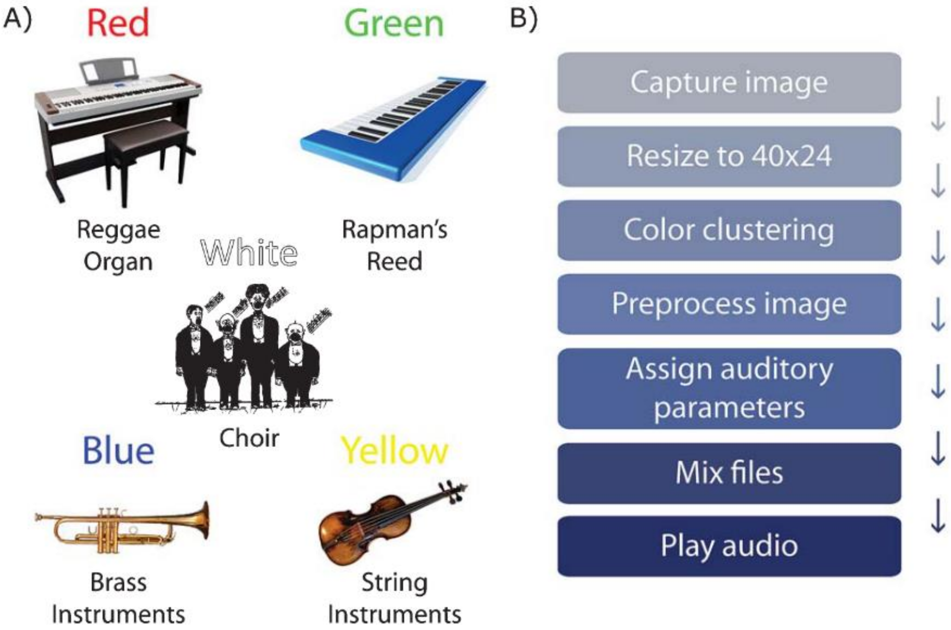


Figure 3-5: EyeMusic Overview (Abboud et al., 2014)

Another notable SSD is Project BATEYE. Project BATEYE is an affordable wearable device, which uses an ultrasonic sensor to measure the distance to the nearest object (within the 15-degree field of view of the ultrasonic sensor). The measured distance is conveyed to the user through tones ranging from 150Hz to 15,000Hz as the distance ranges from 2cm to 4m respectively (Ganguly., 2016). The benefits of this technology are its cost, simplicity and real-time feedback with essentially no training required. The limitations are that the technology only provides depth information about the nearest object within a 15-degree angle, relying on the movement of the head to gain more information about one's surroundings (Ganguly., 2016).

3.4.5 Overview of Sensory Substitution Devices

This section discusses the benefits and shortcomings of the various visual-to-auditory sensory substitution techniques shown in Table 3-2. The only consistent benefit across all the devices is that they are all non-invasive. This is valuable since it means that no invasive surgery is required in order to make use of these devices. On the other hand, a common shortcoming of these devices is their low resolution, with the highest resolution being 176 x 64 pixels.

Table 3-2: The Benefits and Shortcomings of Existing SSDs

SSD	Benefits	Shortcomings
The vOICe	• Cost-effective • Relative to other SS systems, a high resolution (176 x 64 pixel) • Research on it is ongoing • Non-invasive	• Steep learning curve (long training period) • Scans from left to right (slow refresh rate, 1Hz) • Does not directly incorporate depth information • 2D (x and y)
The PSVA	• Real-time • Multiresolution artificial retina • Non-invasive	• Steep learning curve (long training period) • Does not directly incorporate depth information • 2D (x and y) • 124 total pixels
MeloSee	• Real-time • Incorporates depth information (from depth camera) • 3D (x, y and z) • Non-invasive	• Unclear learning curve • 8 x 8 pixel resolution
Project BATEYE	• Real-time (+-70Hz refresh rate) • Easy to use • Cost-effective • 2cm near distance • Non-invasive	• Narrow field of view • Only uses 1 dimension of sound (pitch) • 1x1 resolution (only provides depth for a single point)
EyeMusic	• Conveys colour information • Non-invasive	• Steep learning curve (long training period) • Scans from left to right (slow refresh rate, +-1Hz) • Does not directly incorporate depth information • 2D (x and y) • 40 x 24 pixel resolution

Another common shortcoming of existing systems is the steep learning curve (training period) required; this is the case with the vOICe, PSVA and EyeMusic. The MeloSee algorithm also required a training period, although it is unclear how important the training period was. What is known is that with all the algorithms, including MeloSee, participants became more proficient at using the system with time. Project BATEYE is the easiest to use, however, this is also a result of its simplicity – only providing distance information for a single point (the nearest object within a 15-degree angle in front of the user).

The vOICe and EyeMusic use a scanline technique, scanning the image from left to right roughly once per second. This makes the vOICe and EyeMusic not as usable for real-time tasks such as navigating an unknown environment. The PSVA and MeloSee, on the other hand, use binaural audio techniques; this has the benefit of allowing for playback of the entire image at once, allowing them to obtain refresh rates of 10Hz, which allows for real-time movement.

The MeloSee algorithm can convert the entire image into a playable soundscape at once, and also adds depth information. This benefit is unique in that it makes MeloSee the only algorithm in Table 3-2 to provide all three dimensions of information (x, y and z) – it also does this in real-time. The other systems only incorporate x and y (the vOICe, the PSVA and EyeMusic), or only incorporate z (Project BATEYE). To provide all three dimensions of information, the MeloSee SSD uses a depth camera rather than the traditional grayscale or colour camera used by the other systems (except for Project BATEYE). The depth camera provides depth information for every point on the image. Where most of the systems then use loudness for brightness, depth images do not contain brightness nor colour information in a pixel, they only contain depth information; so, the MeloSee algorithm is free to use loudness for depth.

All the algorithms also use simple tones, except for EyeMusic, which uses simple musical notes to represent colour. However, none of the systems tested complex sounds for improving localization. Another interesting point is that all the algorithms associated pitch with height – except for Project BATEYE, which associated pitch with distance.

Finally, the EyeMusic and the PSVA algorithms both have unique benefits. The EyeMusic can convey colour information; it does this through associating a colour with a musical instrument. The addition of colour adds an extra layer of information, which is beneficial in a number of ways. The PSVA has the unique benefit of implementing a multiresolution artificial retina, roughly imitating how the eye can perceive more information in the fovea (the centre of the eye).

3.5 Evaluation Procedures for Sensory Substitution Devices

The vOICe and PSVA evaluation procedures focus on shapes. This is common in sensory substitution device evaluations. For example, the PSVA evaluation required participants to identify patterns (Capelle et al., 1998). For the evaluation, 25 of 50 basic patterns (generated from 15 base patterns – see Figure 3-6) were randomly chosen. These patterns were shown one after the other to the participants, and the participants were required to reconstruct the pattern using metallic strips. The answers were recorded as either correct or incorrect.

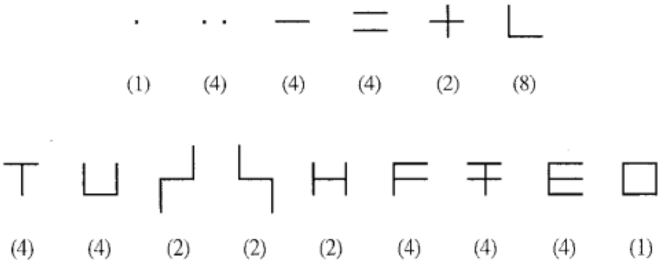


Figure 3-6: The 15 basic patterns used rotated by 45 or 90 to produce 50 images used in the PSVA evaluation (Capelle et al., 1998)

In sensory substitution evaluations it is also common to come across eye tests such as the Snellen tumbling E paradigm (Haigh et al., 2013; Reich, Maidenbaum, & Amedi, 2012; Striem-Amit, Guendelman, Amedi, Carlson, & VanMeter, 2012). These tests are used to determine the visual acuity achievable with the systems – they are essentially a variation on the shape identification evaluations. Visual acuity measured using eye tests such as the Snellen tumbling E paradigm make it easier to compare the visual acuity achieved using the system to the visual acuity of a sighted individual with normal visual acuity (Reich et al., 2012).

The Snellen tumbling E paradigm is useful for obtaining a precise score for how well the sensory substitution system performs on certain visual tasks. The limitation of using this type of evaluation is that it gives little information regarding how well the system performs on navigation tasks, e.g. navigating an unfamiliar environment. Navigation requires an understanding of one's surroundings; this includes an understanding of the distance to various objects in the scene – i.e. depth perception. Evaluations testing the practical usefulness of a sensory substitution device for navigation – specifically navigating an unfamiliar environment without physical touch – should include practical navigation tasks.

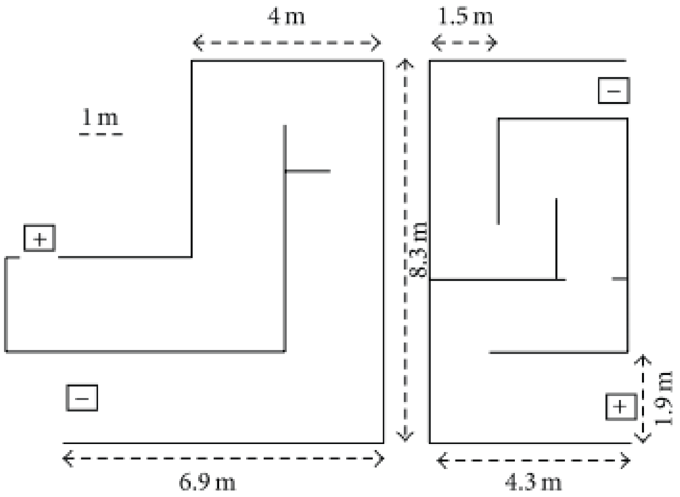


Figure 3-7: Paths A (left) and B (right) used in the MeloSee experiment (Stoll et al., 2015)

The MeloSee algorithm evaluation procedure involved participants navigating through paths (Figure 3-7). Participants were given instructions to make as little contact with walls and screens as possible (i.e. barriers blocking the left or right side of the passage). The metrics gathered for this study were: number of wall touches, number of U-turns and time on task (Stoll et al., 2015). This type of evaluation is focused on testing the usefulness of the sensory substitution system for navigation in an unfamiliar environment. This is in contrast to the type of evaluations often completed for sensory substitution systems, such as the vOICe and the PSVA; which evaluate the visual acuity of the system for the recognition of shapes (generally on a flat surface) such as letters on a board, rather than evaluating the system's usefulness for navigation.

3.6 Baseline Prototype

The Baseline Prototype is a prototype developed based on an existing system. The Baseline System is then able to be used as a starting point for developing an improved visual-to-auditory sensory substitution prototype, as well as a point of comparison. The Research Aim (Section 1.3) is:

> "*To develop an improved visual-to-auditory sensory substitution prototype – using sound localization as a sensory substitution for depth perception.*"

As it can be seen from this aim, the goal is to use sound localization as a sensory substitution for depth perception. This means that the existing system that the Baseline Prototype is based on, should approach the problem of creating a sensory substitution system in a similar way – i.e. primarily focusing on sound localization and depth perception for creating the sensory substitution algorithm. All the algorithms discussed, with the exception of the MeloSee algorithm, focus on conveying a grayscale or colour image through sound. The MeloSee algorithm focuses on conveying a depth image through sound (Section 3.4.3). It also uses sound localization principles when generating sound, produces soundscapes in real-time and is the only algorithm discussed that provides information regarding all three dimensions (x, y and z). For these reasons, the Baseline Prototype was developed based on the MeloSee algorithm.

3.7 Conclusions

This chapter focused on understanding the knowledge base for visual-to-auditory sensory substitution. First, it explained what is meant by the term visual-to-auditory sensory substitution (Section 3.2). It then went on to look at sensory substitution and the brain (Section 3.3). The next section discussed existing systems (Section 3.4) along with their benefits and shortcomings as shown in Table 3-2. It then gave an overview of existing systems evaluation procedures (Section 3.5). Next, considering the Research Aim (Section 1.3), the MeloSee algorithm was chosen as a base for developing the Baseline Prototype. This Baseline Prototype can then be used as a point of comparison when answering **RQ 6** during the DSR Design Cycle. The next chapter looks at a framework developed for implementing sensory substitution algorithms.

Chapter 4: Sensory Substitution Framework

4.1 Introduction

The previous chapters gave an understanding of how humans perceive their environment and how researchers have used that knowledge to develop sensory substitution systems in the past. In the DSR methodology, the Design Cycle is a process of development, evaluation and improvement. The next chapters will focus on the development and evaluation of new sensory substitution algorithms based on the knowledge gained from the research previously discussed. Before looking at the Design Cycle for the sensory substitution prototypes, this chapter aims to answer:

> **RQ 3**: *"How can visual-to-auditory sensory substitution prototypes be developed to allow for the testing of different visual-to-auditory sensory substitution techniques?"*

This research question is answered by creating a set of prototype development objectives, and then looking at the framework developed for implementing sensory substitution prototypes – the Sensory Substitution Framework[1]. The set of prototype development objectives are based on the knowledge gained from existing systems (Section 3.4).

4.2 Prototype Development Objectives

One of the DSR activities is to define the objectives of the solution based on the problem definition. In the case of this project, this means defining a set of prototype development objectives based on the main research question; these objectives can then be used as a reference point throughout the rest of the project. The main research question is:

> *"How can sound localization be used as a sensory substitution for depth perception, to give the visually impaired an accurate understanding of their surroundings through audition?"*

Considering the main research question and **RQ 3**, as well as how the existing systems (Section 3.4) were designed, how they function, and their benefits and shortcomings, a set of objectives were generated to guide the prototype development:

[1] https://github.com/jamesdeklerk/sensory-substitution-framework

PDO 1. Create a standardised framework for implementing (and testing) visual-to-auditory sensory substitution prototypes. The framework should allow the prototypes to be:

  a. Modular – separating the visual and auditory components; and
  b. Configurable – allowing testing of different configurations.

PDO 2. Implement the Baseline Prototype using the standardised framework.

PDO 3. Prototype Development Objectives for improving on the Baseline Prototype:

  a. Implement the prototype using the standardised framework – to allow for comparison to the Baseline Prototype;
  b. Real-time (rather than left to right scanline);
  c. Include all three dimensions – depth perception, left-right perception and up-down perception;
  d. Use sound localization principles in conjunction with a 3D audio library; and
  e. Improve spatial perception (based on a standardised evaluation procedure) – compared to the Baseline Prototype.

The first prototype development objective (PDO 1) is to create a standardised framework for implementing visual-to-auditory sensory substitution prototypes. The motivation for this PDO is given in the next section.

4.3 Motivation for Developing a Framework

The reason for developing the framework is to have a standardised, consistent and modular way of implementing, testing and comparing different algorithms. Currently if a researcher develops a new sensory substitution algorithm and wants to compare its performance to an existing algorithm, the researcher must re-implement the existing algorithm. This is the case for a large number of sensory substitution algorithms. Using a standardised framework, researchers can share their algorithms knowing there is a standard procedure for running the algorithms.

The framework also removes the need to re-implement commonly used code, such as the code used to read information from the camera. This results in code which is easier to maintain, in addition to allowing for common code to be shared across algorithms – speeding up development time. The framework also supports configuration files, which allow for rapid testing of different configurations, improving the ability to test and conduct preliminary evaluations of different combinations of algorithm subroutines.

The framework aims to improve the development and testing experience for sensory substitution researchers. Hence, it is hoped that this framework will contribute to the development of a standardised framework for developing, testing and comparing sensory substitution algorithms.

4.4 Framework Development

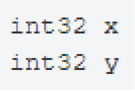
The framework developed is named the Sensory Substitution Framework (SSF). This sensory substitution framework was developed primarily with visual-to-auditory sensory substitution in mind, however, it is expected that the framework would be similarly useful for other types of SSDs. Visual-to-auditory SSDs consist of a number of hardware components; the standard components being a camera used to capture the visual information, a computer used to process the visual information and headphones to output the generated sounds. Visual-to-auditory SSDs also generally consist of a number of software components; at a fundamental level, the software for almost all visual-to-auditory sensory substitution systems can be split into two parts – the visual processing algorithm and the auditory processing algorithm (Capelle et al., 1998; Stoll et al., 2015; Veraart, 1989).

Based on the software and hardware components, careful thought was given to the structure of the framework and how its pieces would fit together. The framework needed to be flexible enough that one can develop a large variety of sensory substitution algorithms; yet, the framework also should have a fixed structure so that one can swap out the visual processing algorithm without needing to re-implement the auditory processing algorithm, or vice versa. All this while allowing for dynamic (live) swapping between algorithms and dynamic configuration of algorithms. The framework structure used to achieve this is discussed in Section 4.4.2. However, before looking at the framework structure, it is useful to understand the Robotic Operating System (ROS) discussed below.

4.4.1 Robotic Operating System (ROS)

The SSF was built on top of the Robotic Operating System (ROS). This section will give a brief introduction to ROS. ROS is an open-source modular platform (or “meta-operating system”) commonly used for the development of robotic systems (Open Source Robotics Foundation, 2018). It runs on Ubuntu 16.04 and is language independent, however, the default supported languages are Python, C++ and Lisp. The specific version of ROS used for this project was ROS Kinetic Kame, which was chosen since it is a Long Term Support (LTS) release. There are a large number of benefits to using ROS; these include that fact that ROS encourages the development of modular software (used to achieve *PDO 1.a*), it is built for real-time processing, there are many extremely useful ROS packages (e.g. for the depth camera used in this project), and it has a well-established community. ROS has even been used in the development of visual-to-tactile SSDs (Cancar, Diaz, Barrientos, Travieso, & Jacobs, 2013; Taylor, 2017).

At its core, ROS consists of nodes, messages and a ROS Master. A node is simply a program which performs some computation; in order to create modular code, generally one will have many nodes each performing a relatively specific task. Nodes communicate with one another using messages. A message is a packet of structured data, the structure of the message is called the message description. There are predefined message descriptions in ROS, and it is also possible to create a custom message description. A simple message containing two integers – x and y – is illustrated in Figure 4-1.



```
int32 x
int32 y
```

Figure 4-1: Simple ROS message description

This communication takes place using the publish-subscribe pattern (Open Source Robotics Foundation, 2019) which means that one node would publish a message to a topic, and another node would subscribe to that topic, receiving a message each time it is published (a topic is essentially a name identifying the message content published to it). This interaction is managed by the ROS Master as illustrated in Figure 4-2. Figure 4-2 illustrates that thanks to the ROS Master, the publisher (Node A) and subscriber (Node B) can be completely independent of one another. In other words, the publisher is not interested in who uses the messages it publishes, and the subscriber does not care where the messages come from. Nodes can publish on multiple topics and subscribe to multiple topics. It is also normal for multiple nodes to publish to the same topic or subscribe to the same topic.

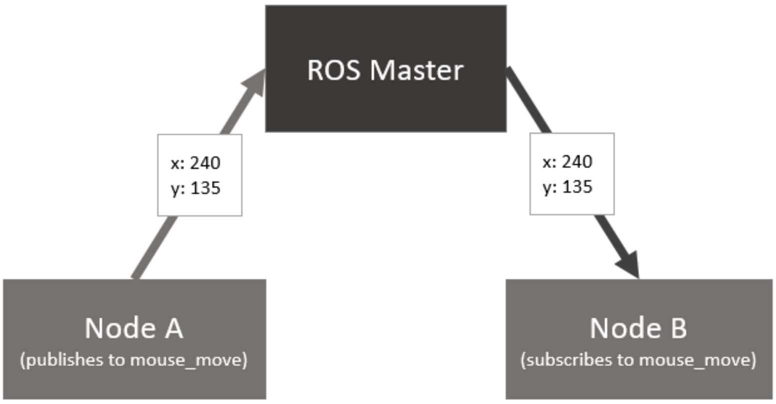


Figure 4-2: ROS Master, Node and Message Interaction

A simple example based on Figure 4-2 could be the following: Node A has information about the movement of the computer mouse. Every time the mouse moves, Node A publishes a message containing the new x and y position of the mouse to the topic “mouse_move”. The message description (i.e. format) of the message data is illustrated in Figure 4-1, with a specific instance of a message illustrated in Figure 4-2. The ROS Master manages these messages, and every time the ROS Master receives a message on a specific topic, it calls the nodes subscribed to that topic, and then passes them a copy of the message data. In this case, Node B is subscribed to the topic “mouse_move”; so, every time the mouse moves, Node A publishes a message to “mouse_move”, and subsequently the ROS Master calls Node B and passes it a copy of the message data. Node A can then do what it wants with the data, for example, print it to the console.

Nodes typically will subscribe to a topic, receive messages on that topic, process the message data (which may be images, strings etc.), form a new message with the processed data and publish the new message on a topic (generally a different topic). There will also normally be multiple nodes in the system, each performing a specific task.

The code described above would generally be put into a ROS package. A ROS package is the unit for organising code. A package commonly consists of many nodes, some configuration files (used to achieve *PDO 1.b*), some datasets and some ROS-dependent libraries. To record and playback, the data published by the various nodes ROS has what are called ROS bags. A ROS bag is a file which stores a recording of messages published to topics. When making a recording to a ROS bag, one specifies which topics to record, and ROS then records all the messages published to those topics. The ROS bag can then be played back at any stage, where playing back means publishing the messages recorded in the order they were recorded over the same amount of time they were recorded. This is extremely useful since nodes do not care where the messages come from (as previously discussed); the nodes can receive (and hence process) the recorded messages as if they were not recorded, but rather being received from sensors in real-time.

4.4.2 Sensory Substitution Framework Overview

Almost all visual-to-auditory sensory substitution systems can be split into two parts – the visual processing algorithm and the auditory processing algorithm (Section 4.4). This is the basis for the structure of the framework developed. The visual processing algorithm is referred to as the Retinal Encoder, and the auditory processing algorithm is referred to at the Sound Generator. The naming conventions of some framework components were inspired by the naming conventions used for the MeloSee system (Stoll et al., 2015).

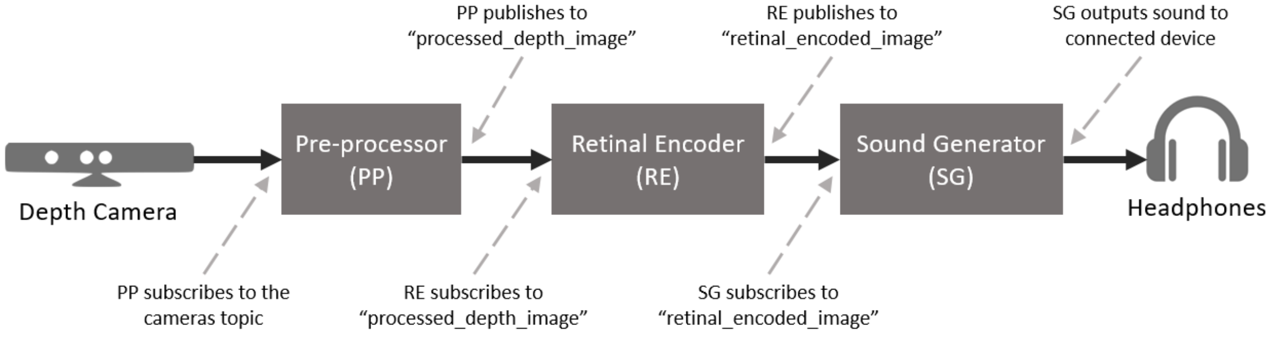
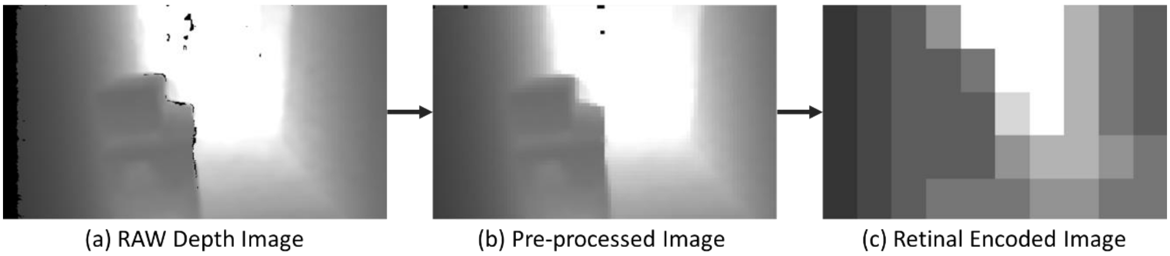

PP publishes to “processed_depth_image”
RE publishes to “retinal_encoded_image”
SG outputs sound to connected device
Depth Camera
Pre-processor (PP)
Retinal Encoder (RE)
Sound Generator (SG)
Headphones
PP subscribes to the cameras topic
RE subscribes to “processed_depth_image”
SG subscribes to “retinal_encoded_image”

Figure 4-3: Sensory Substitution Framework pipeline

As illustrated in Figure 4-3, the first stage in the pipeline is a camera; this is responsible for capturing an image in real-time – the camera could be a standard colour camera, a depth camera or a combination depending on the algorithm's needs. The next stage is the Pre-processor (PP); the Pre-processor is responsible for any pre-processing done on the image, this could include denoising the image (i.e. removing the errors) and cropping the image, among other things – see Figure 4-4 (b) for an example. The pre-processed image is then passed onto the Retinal Encoder (RE); the Retinal Encoder is responsible for compressing the image by using principles from the human visual system in a useful and efficient manner. The retinal encoded image – see Figure 4-4 (c) for an example – is then passed onto the Sound Generator (SG); the Sound Generator generates a soundscape (Section 3.4.1) from the retinal encoded image, using principles from psychoacoustics (Section 2.2). The generated sound is then played through headphones, which completes the pipeline. Depending on the speed of the algorithms, this process usually runs at a rate of 30Hz (i.e. 30 frames a second).


(a) RAW Depth Image

(b) Pre-processed Image

(c) Retinal Encoded Image

Figure 4-4: Example of a Depth Image (a), Pre-processed Image (b) and a Retinal Encoded Image (c).

The Pre-processor, Retinal Encoder and Sound Generator are all ROS nodes, which communicate via ROS messages. The way these nodes interact and the topics used are discussed next (Figure 4-3 can be used to follow this process). Although one could use both a depth camera and colour camera, for simplicity, only the depth image is considered here. The Pre-processor subscribes to the depth camera's topic (this topic is not standardised as it depends on the camera used), it then receives a depth image whenever one is published on the depth camera's topic. When the Pre-processor then receives a depth image, it processes the image received and publishes the processed image to the topic "processed_depth_image". The Retinal Encoder subscribes to that topic ("processed_depth_image"), receiving a processed depth image whenever one is published to that topic. When the Retinal Encoder receives a processed image on the "processed_depth_image" topic, it does the relevant processing on that image – generally simulating aspects of the human visual system – resulting in the retinal encoded image. The Retinal Encoder then publishes the retinal encoded image to the topic "retinal_encoded_image". The Sound Generator subscribes to the topic "retinal_encoded_image", receiving a retinal encoded image whenever one is published to that topic. When the Sound Generator receives a retinal encoded image, it generates sound based on the image, playing the sounds through the listening device attached (generally a set of headphones).

The fact that the Pre-processor, Retinal Encoder and Sound Generator are all ROS nodes – and that they subscribe and publish to a standardised set of topics – is what allows each of the components to be swapped out. This ability to swap the components out is what provides the SSF with flexibility. This means that one could develop a new Sound Generator algorithm without changing anything else in the pipeline. For example, other researchers may have implemented a Retinal Encoder that performs well, and another researcher would like to test out a new Sound Generation algorithm using the existing Retinal Encoder. With previous systems, one would need to re-implement the Retinal Encoder and determine how to make it compatible with the Sound Generator. With the SSF, because the communication is standardised between the different nodes, those complexities fall away. Using the SSF, the researcher simply focuses on implementing a Sound Generation algorithm – in fact, a Sound Generation algorithm template is even provided as a starting point – and chooses the Retinal Encoder algorithm to be used. One could just as easily write one's own Retinal Encoder or Pre-processor. Similarly, if one wanted to create a completely new visual-to-auditory sensory substitution system, one could implement a new Pre-processor, Retinal Encoder and Sound Generator. The researcher focuses on the algorithms, allowing the framework to deal with the complex interactions between the nodes, and between the hardware and software.

4.4.3 Sensory Substitution Framework Features

The SSF also provides a number of useful features (built on top of ROS). These include dashboards (Section 4.4.4) for monitoring or interacting with algorithms as they run in real-time, visualisation tools to see how sound emitters are placed in 3D space (Figure 4-5), and a set of pre-built functions (SSF Core – Section 4.4.5) that can be used across algorithms.

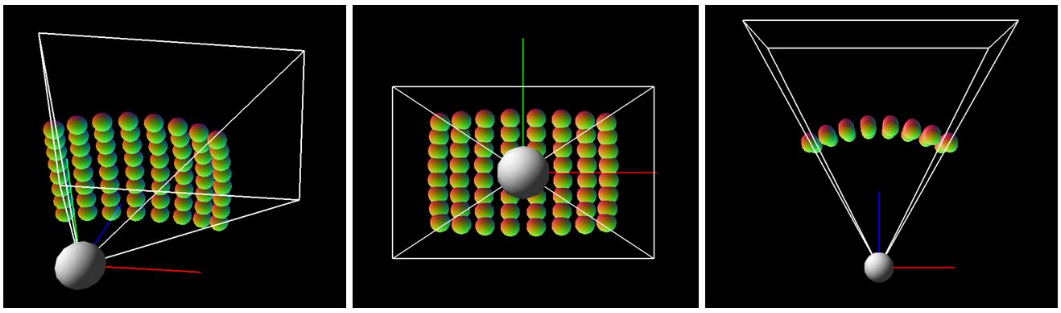


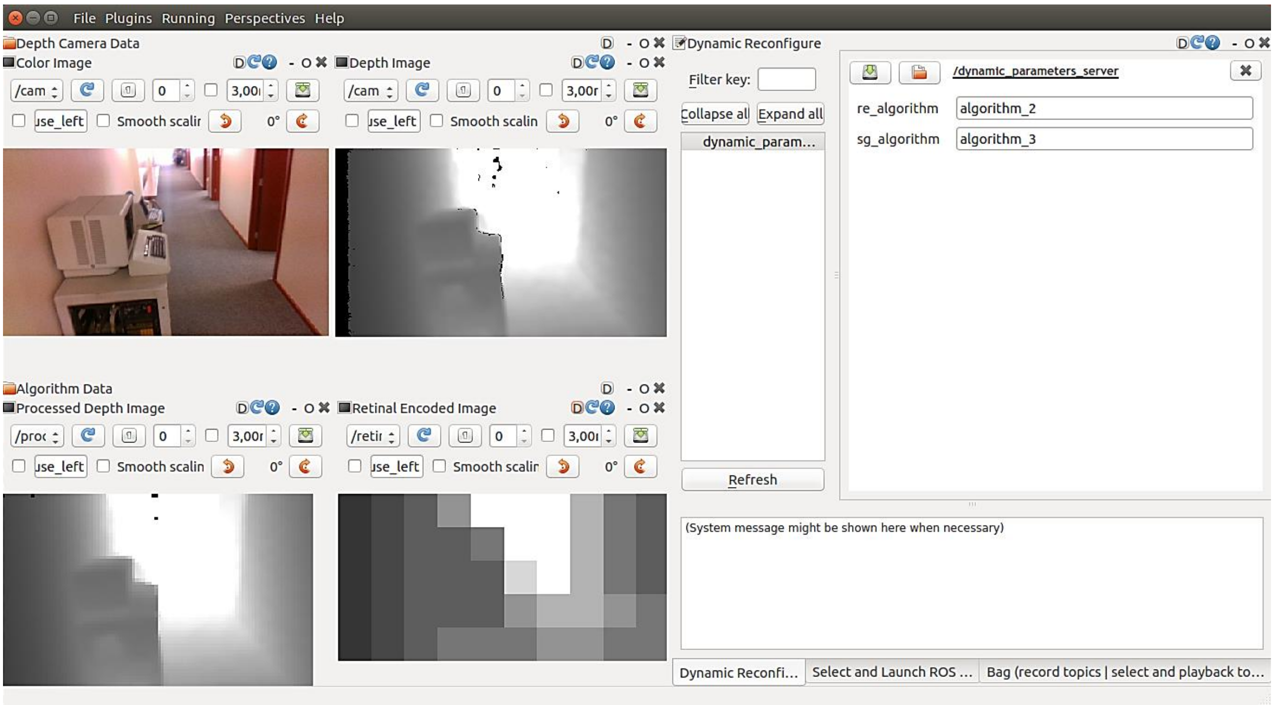
Figure 4-5: Three views of a 3D visualisation provided by the SSF. The grey sphere represents the listeners head, and the coloured spheres represent sound emitters placed in 3D space relative to the listener.

The framework allows for configuration of many parameters using configuration files. These include the camera settings, the amount of crop applied to images, the distance units used and, the default algorithms to launch, among many other parameters. The design of the framework encourages the use of configurability even when developing custom Pre-processors, Retinal Encoders and Sound Generators. The framework also supports real-time configuration (dynamic reconfiguration), meaning one can change the parameters while the algorithms are running. Due to the decoupled nature of ROS nodes, this means that one can swap out the Pre-processors, Retinal Encoders or Sound Generators in real-time. This allows one to “flick between” two different Retinal Encoders to see how they differ without having to reboot the entire system.

Since the framework is built on top of ROS, it supports ROS bags by default. As discussed in Section 4.4.1, this means that the framework can record and playback the data captured. This is extremely useful for comparing algorithms using a standard set of recorded example scenarios and for recording evaluations for future reference. For general usage of the system refer to Appendix C.

4.4.4 Dashboards

An SSF dashboard is a user interface (UI) with a number of elements used for monitoring or interacting with algorithms as they run in real-time. The framework has several built-in dashboards: one for viewing the colour image, depth image and retinal encoded image side by side. Another one is called the evaluation dashboard, which only shows the retinal encoded image, and is used to make sure the algorithm is running correctly before starting an evaluation.



re_algorithm algorithm_2

sg_algorithm algorithm_3

(System message might be shown here when necessary)

Figure 4-6: Sensory Substitution Framework prototyping dashboard

Another dashboard, called the prototyping dashboard, is shown in Figure 4-6. As can be seen from the labels on the dashboard, it gives access to dynamic configuration settings (top-right), ROS bag recording and playback features (bottom-right), and, the ability to launch other ROS nodes or packages through the UI (bottom-right). The prototyping dashboard also shows the colour image, depth image, pre-processed depth image and the retinal encoded image in a 2 by 2 grid (left). Having quick access to these features in a UI is extremely useful, as it increases the efficiency of the development process for sensory substitution algorithms.

4.4.5 SSF Core

SSF Core[2] is a Python module developed by the researcher for the SSF. This module contains core functions that may be used across the framework and across different sensory substitution algorithms. The functions and algorithms that were implemented in SSF Core include but are not limited to:

- Image cropping;
- Calculating the change in the field of view after image cropping;
- K-means clustering implementation for processing depth images;
- Applying quantisation;
- Temporal filter for denoising depth images;
- Handling 3D projections;
- Sampling pixels according to a 2D normal distribution; and
- Approximating sound frequency using a fast Fourier transform.

All of the functions and algorithms implemented in the SSF Core are optimised to ensure that they can be run in conjunction without noticeably impacting the real-time nature aimed for in SSDs. This is done through using parallelised functions wherever possible. The use of SSF Core has the potential to greatly decrease the time taken to develop sensory substitution algorithms.

4.4.6 Language Used

Python was the development language chosen for the SSF; this was based on the languages supported by ROS – one of the main languages being Python (Open Source Robotics Foundation, 2018). One advantage of using Python is that for a number of years it has been comfortably within the top five most commonly used languages according to GitHub's annual report (GitHub, 2018). Another advantage is the broad range of tools and software developed for Python, from

[2] https://github.com/jamesdeklerk/sensory-substitution-framework/tree/master/ssf_package/src/core

Tensorflow (Google, 2018) to OpenCV (OpenCV team, 2018). To improve code readability and ease of use, a Python style guide was used. The style guide followed was PEP8 (Python Software Foundation, 2018), where PEP stands for Python Enhancement Proposal. This is currently the most popular style guideline for Python.

4.5 Conclusions

The sensory substitution framework developed was successfully used for implementing the three different algorithms for this project – this is discussed in later chapters (Chapter 6 and Chapter 7). Using the framework – which allowed for relatively fast algorithm implementation, testing and comparison with other algorithms – proved valuable to the research. Other features of the framework such as recording and playback of data, configuration files and the dashboards for viewing the real-time processing of the algorithms, were also found to be valuable. In addition, thanks to the flexibility of the SSF, having the ability to easily swap out both hardware and software components made it simple to switch over to better hardware when it became available (Section 6.3.1). The framework also made it feasible to test out different combinations of Retinal Encoders and Sound Generators, which is extremely useful. *PDO 1* was thus achieved through the development of the SSF.

The framework developed will be beneficial for researchers in the field of sensory substitution; especially for researchers looking to implement and test new algorithms, then compare those algorithms to existing algorithms using the same hardware and software setup. Finally, it is hoped that this framework or a derivative of this framework will be used to improve the development experience of SSDs. The next chapter looks at the evaluation process.

Chapter 5: Evaluation Design

5.1 Introduction

The previous chapter looked at the framework used to develop new sensory substitution systems. The next stage is to look at the development and evaluation process. In the DSR methodology, the Design Cycle is a process of development, evaluation and improvement. Before looking at the Design Cycles for the sensory substitution prototypes, this chapter looks at the evaluation procedure used for evaluating the prototypes. This aims to address:

> **RQ 4**: "*How can visual-to-auditory sensory substitution prototypes be evaluated to provide insight into the effectiveness of the different visual-to-auditory sensory substitution techniques?*"

The focus of the evaluations for this project is on depth perception and spatial awareness. From Section 3.5 it was learnt that many evaluation procedures focus on the visual acuity of the sensory substitution systems by looking at Snellen eye charts. Although useful in certain regards, evaluations of sensory substitution systems using Snellen eye charts do not provide much information on how well the systems perform when navigating an unfamiliar environment. Other evaluation procedures in literature only focus on a single aspect of navigation, such as traversing a path without obstacles such as tables or chairs. No suitable evaluation procedure was found, which was able to evaluate how well the system performed for general navigation, as well as, evaluate the system on more isolated tasks; these allow for a better understanding of the specific ways in which the system performed well or poorly.

For this reason, a new evaluation procedure was designed. The evaluation procedure consists of two navigation tasks and three object detection tasks. The evaluation procedure was designed using ideas from existing evaluation procedures where applicable – such as recording wall touches for the navigation tasks, as done in the MeloSee evaluation (Section 3.5).

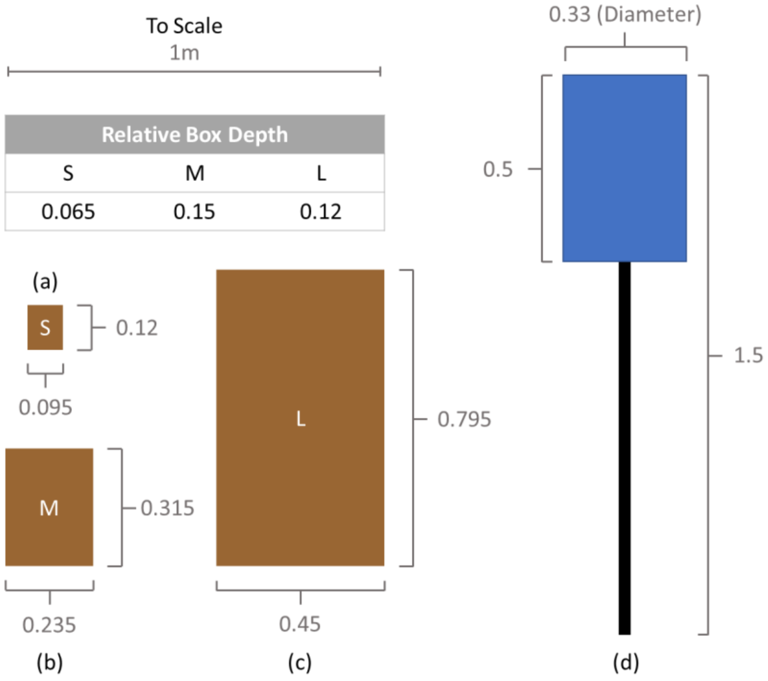
The full evaluation procedure involved a comparative study between two systems. The evaluations were done to test whether or not a prototype improved relative to the Baseline Prototype. To do this, each participant performed a set of tasks with the new prototype and performed the same set of tasks with the Baseline Prototype – the order in which the systems were evaluated was randomised. This chapter reviews the participant selection, objects used for the evaluations, the evaluation tasks, the evaluation procedure and the metrics gathered from the evaluation.

5.2 Participant Selection

Ethics clearance was given for students who were part of the NMU Department of Computing Sciences (Section 1.6). For this reason, participants were randomly selected from the NMU Department of Computing Sciences. A pre-evaluation questionnaire was completed by each potential participant to identify whether they were eligible to participate in the study. Those who had prior experience with a sensory substitution system and those who did not have the ability to hear from 200Hz to 10000Hz were not eligible for participation in the evaluations. For each of the main studies (the comparative studies) eight eligible participants were selected. For each study, a new set of participants was selected, with each new participant satisfying the same eligibility criteria set out above.

5.3 Objects Used for Evaluations

This section reviews the objects used throughout the different evaluation tasks. Four objects where used in total. Three different size cardboard boxes and one foam mat rolled up into a cylinder secured on the end of a camera monopod were used. The foam mat attached to the camera monopod will be referred to as the "quadrant task object" from here on.



Relative Box Depth		
S	M	L
0.065	0.15	0.12

Figure 5-1: Three box sizes used in the evaluation, small (a) medium (b) and large (c) together with the quadrant task object (d) – all measurements are given in meters

On the left of Figure 5-1 the three different size boxes – small, medium and large – are illustrated with their respective dimensions. Figure 5-1 (d), illustrates the quadrant task object with its respective dimensions. Since the foam mat (top) component of the quadrant task object is rolled up to form a cylinder, the horizontal dimension given is a diameter – as indicated in Figure 5-1 (d).

5.4 Evaluation Tasks

This section looks at the different tasks completed throughout the comparative study. All tasks were completed while the participants were blindfolded and wearing a visual-to-auditory sensory substitution device. For all tasks, the participants were told that moving their heads around – as one would if they were looking around – helps with the perception of their surroundings as it allows them to "see" more of their surroundings. For the tasks, there were two main task classifications, object detection tasks and navigation tasks. An orientation task was completed for each of these task classifications. The object detection tasks were completed in the evaluation room (Figure 5-4), and the navigation tasks were completed in a fixed set of passages in the NMU Department of Computing Science (Figure 5-2). For all navigation tasks, participants were told to put their elbows to their hip bones and point their forearms forward – forming an L shape. They were also told no reaching is allowed – they are to keep their arms fixed in the L position. Additionally, all doors in the passages were closed for the duration of the evaluation and passages cleared of people and objects not shown in Figure 5-2. This allowed for consistency when gathering metrics.

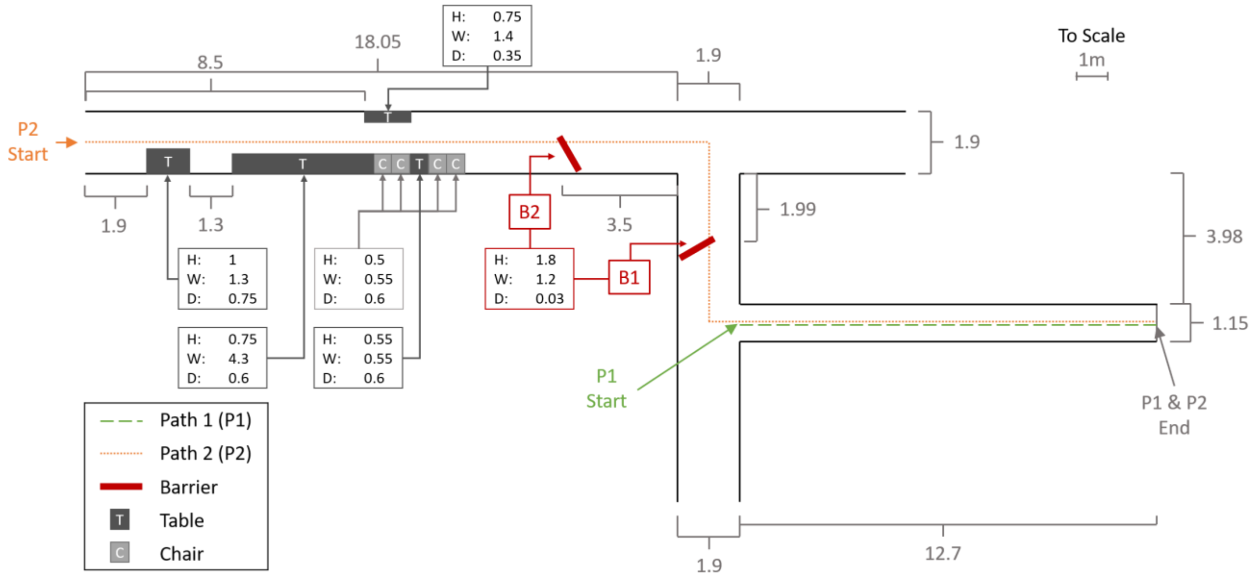


Figure 5-2: Scale diagram of the passages used in the evaluation – all measurements are in meters

Navigation Task (no obstacles): This task involved walking down Path 1 as illustrated in Figure 5-2. Path 1 is a narrow passage with no obstacles. The participant was told to walk to the end of the passage avoiding touching the end and avoiding touching any walls along the way. If the participant touched a wall, the evaluator re-centred the participant in the middle of the left and right passage walls, and the participant then continued. When the participant believed that they had reached the end of the passage – a dead end – they said “done”.

Navigation Task (with obstacles)**:** This task involved being given a set of directions, and being told to follow the directions avoiding walls and obstacles as far as possible. Path 2 (Figure 5-2) traces the directions given to the participants. Along Path 2, a barrier was randomly placed at one of two locations, B1 or B2 as illustrated in Figure 5-2. For either placement the barrier blocked the right-hand side of the passage, leaving a gap on the left – that is, when the participant was oriented along the path direction (as illustrated in Figure 5-2 using the start arrow for Path 2). For either barrier placement, the barrier was placed at a 60° angle to the wall, this left an approximately 0.85m gap on the left side of the passage.

Based on the barrier placement, the participants were told one of two things. For barrier placement B1, Path 2 was explained to the participant in the following order:

1. Walk down the passage, take the first right and carry on going.
2. Once they had taken the right, at some point, there would be a barrier blocking the right side of the passage, they were to walk around the barrier and carry on going.
3. Once they had passed the barrier, they were to take the first left.
4. After taking the first left, they would be in a passage with a dead end; it was explained that they were to walk to the end of the passage and when they believed they had reached the end of the passage, they said “done”.

For barrier placement B2, Path 2 was explained to the participant in the following order:

1. Walk down the passage, at some point, there would be a barrier blocking the right side of the passage, they are to walk around the barrier and carry on going.
2. Once they had passed the barrier, they were to take the first right and carry on going.
3. Once they had taken the right, they were to take the first left.
4. After taking the first left, they would be in a passage with a dead end; it was explained that they were to walk to the end of the passage and when they believed they had reached the end of the passage, they said “done”.

For both barrier placements (B1 and B1) along Path 2, the participants were told that after each of the four objectives – taking the right turn, after passing the barrier, after taking the left turn and after reaching the end of the passage – they were to state that they think they have completed the relevant objective. The evaluator then confirmed whether they believed they had completed the objective and confirmed the next objective with the participant. The participant was told that there may be objects along the path. The participant was also told to avoiding touching any walls and obstacles along the way. If the participant touched anything, the evaluator re-centred the participant in the middle of the left and right passage walls, and the participant then continued.

Quadrant Task: For this task, the quadrant task object discussed in Section 5.3 was held up to one of four quadrants – top-left (T-L), top-right (T-R), bottom-left (B-L) or bottom-right (B-R). The participant was then asked to point to the quadrant they believed the object was in. The Quadrant Task was completed in the evaluation room using configuration (a) as illustrated in Figure 5-4.

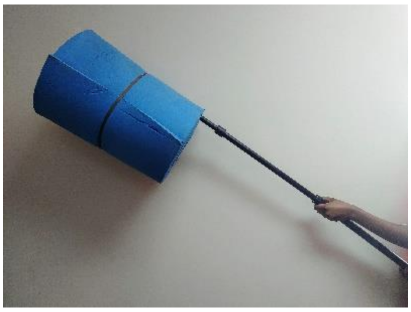


Figure 5-3: The quadrant task object being held up

For the Quadrant Task, the participant was guided to the appropriate location, given a chair to hold on to for support and asked to remain standing (see Figure 5-4 for the exact location). The evaluator then stood 2.5m in front of the participant (see Figure 5-4 for the exact location). The quadrant task object was then held up to one of the four quadrants by the evaluator (Figure 5-3). The front of the quadrant task object (the front of the foam mat) lined up with the Left Placement or Right Placement (Figure 5-4) at 0.5 (bottom) or 1.8 (top) meters high depending on the quadrant. The quadrants were randomly selected, and the participants were never told what the quadrant task object was, nor were they allowed to feel the object.

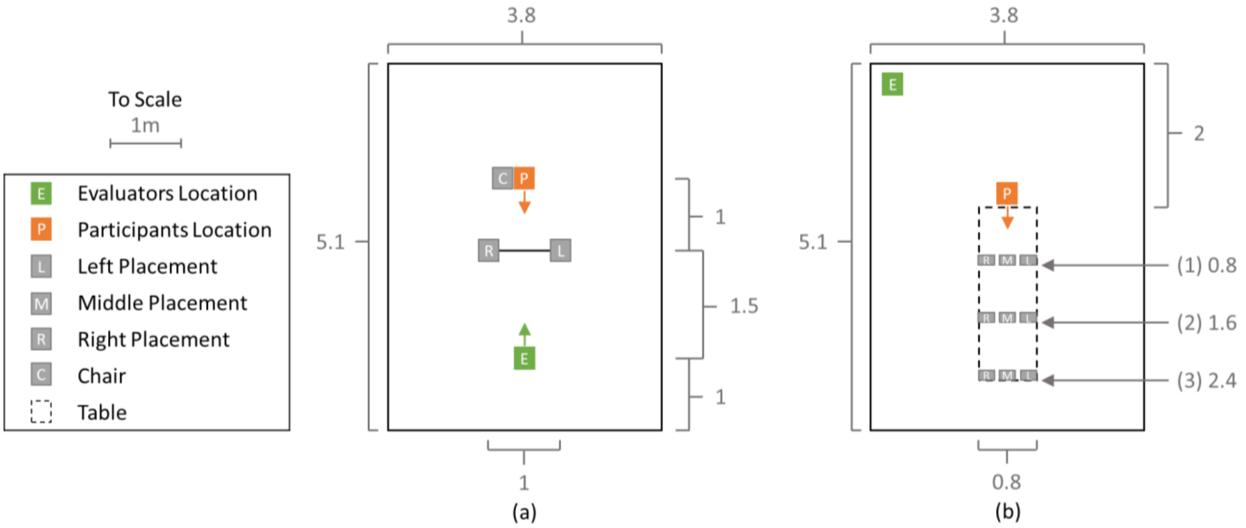


Figure 5-4: Two configurations of the evaluation room. Quadrant Task configuration (a). The Box Placement Task and Object Count Task configuration (b) – all measurements are in meters

***Box Task*:** For this task, a small, medium or large box was placed on the left or right half of a table in front of the participant. The box was placed at one of three distances and in one of two orientations. The participants were then asked the following about the box:

1. What size box did they believe was in front of them – they were given three options: small, medium or large (Figure 5-1).
2. What side did they believe the box was on – they were asked to point to the side they believed it was on and were given two options: left or right.
3. What distance did they believe the box was at – they were given three options (1) 0.8m, (2) 1.6m or (3) 2.4m (Figure 5-4).
4. What orientation did they believe the box was in – they were given two options tall (portrait) or long (landscape).

The participants were seated in a chair in front of a table in the evaluation room (see Participants Location in Figure 5-4). Configuration (b) of the evaluation room (as illustrated in Figure 5-4) was used for the Box Task. The box size, left-right position, distance and orientation were all randomised.

***Multiple Boxes Task*:** For this task, between one and three boxes were placed in front of the participant. A box could be placed on the left, middle or right. For example, a box could be placed on the left and the middle, with the right side left open. Once the boxes were placed, the participants were then asked the following:

1. Did they believe there was a box on the left?
2. Did they believe there was a box in the middle?
3. Did they believe there was a box on the right?

For the Multiple Boxes Task, all the boxes were the same size, placed at the same distance and in the same orientation. These parameters remained fixed – the parameters used were: a medium box at a distance of (1) 0.8m in the tall (portrait) orientation.

***Orientation Task (for object detection)*:** This task was performed to allow the participants to get accustomed to the system. This task was comprised of three examples, the Box Size and Distance Example, the Quadrant Example and the Person Example. For each example, the participant was told to listen for how the sounds generated changed as the evaluator described what was in front of them. Configuration (b) of the evaluation room (as illustrated in Figure 5-4) was used for the Box Size and Distance Example. The procedure for the Box Size and Distance Example was:

1. The participant was seated in front of a clear table (see Figure 5-4 for placement).
2. The participant was told that there was an empty table in front of them.
3. A large box was then placed in the horizontal-centre of the table in the long (landscape) orientation, at three different distances (see Figure 5-4 for distance markings) – (1) 0.8m, then (2) 1.6m, then (3) 2.4m.
4. A medium box was then placed in the horizontal-centre of the table in the long (landscape) orientation, at three different distances (see Figure 5-4 for distance markings) – (1) 0.8m, then (2) 1.6m, then (3) 2.4m.
5. A small box was then placed in the horizontal-centre of the table in the long (landscape) orientation, at three different distances (see Figure 5-4 for distance markings) – (1) 0.8m, then (2) 1.6m, then (3) 2.4m.

The Box Size and Distance Example was designed to give the participants an understanding of how different size boxes at different distances sound through the system when placed on a table. For the next example, the Quadrant Example, configuration (a) of the evaluation room (as illustrated in Figure 5-4) was used. In the same manner as the Quadrant Task, the quadrant task object was held up to one of the four quadrants. The procedure for the Quadrant Example was:

1. The participant was guided to stand next to a chair, holding onto the chair for stability, and the evaluator stood 2.5m in front of the participant.
2. The quadrant task object was held at the top-left quadrant, then moved to the bottom-left, then back to the top-left, then back to the bottom-left.
3. The quadrant task object was then moved to the top-right quadrant, then moved to the bottom-right, then back to the top-right, then back to the bottom-right.

The Quadrant Example was designed to give the participants an understanding of what a relatively isolated object sounds like through the system – when the object was in one of the four quadrants. The next example was the Person Example – configuration (a) of the evaluation room (as illustrated in Figure 5-4) was used. For that example, the evaluator simply stood on the participant's left, then on the participant's right – 1m in front of the participant. This meant that rather than the evaluator standing at the position illustrated in Figure 5-4 configuration (a), the evaluator stood at the Left Placement and Right Placement respectively. The Person Example was designed to allow the participant to hear what an isolated object on their left and right sounded like through the system.

***Orientation Task (for navigation)*:** This task was also performed to allow the participants to become accustomed to the system. For this orientation task, the participant was once again told to listen to how the sounds produced by the system changed as the evaluator described what was in front of them. The procedure for the Orientation Task (for navigation) for each participant was:

1. The evaluator guided the participant by the shoulders into an open space.
2. The evaluator then guided the participant to a table that was on the right; the evaluator then took the participant's wrist and guided his/her hand to touch the table.
3. The evaluator then guided the participant to a wall which was on the left; the evaluator then took the participant's wrist and guided his/her hand to touch the wall.
4. The evaluator then guided the participant to a wall which was on the right; the evaluator then took the participant's wrist and guided his/her hand to touch the wall.
5. The evaluator then guided the participant to a table which was on the left; the evaluator then took the participant's wrist and guided his/her hand to touch the table.

The Orientation Task (for navigation) was designed to allow the participants to hear how the system sounds changed as they navigated an environment. This included allowing the participants to hear how the system sounds changed as they approached lower objects such as tables.

5.5 Questionnaires

The questionnaires for the evaluation were primarily used to obtain qualitative data from the participants. There were three types of questionnaires designed for the evaluation: a Pre-Evaluation Questionnaire (Appendix D), a Post-Evaluation Questionnaire (Appendix E), and a Final Questionnaire (Appendix F). The Pre-Evaluation Questionnaire was designed to gather data before the participants had used the sensory substitution systems. The Post-Evaluation Questionnaire was designed to gather data straight after the participants had used a system – it was completed for each system used. The Final Questionnaire was designed to gather data about how the two systems compared to one another. For the questions asked and metrics gathered in the questionnaires, refer to the appendices (Appendix D, Appendix E and Appendix F) or Section 5.7.2.

5.6 Comparative Study Evaluation Procedure

This section looks at the evaluation procedure developed for a comparative study between sensory substitution systems. This procedure was followed for each participant in the study. Before the evaluation procedure began, the evaluator talked through the consent form (Appendix C) with the participants. This included explaining the aim of the study, the procedure that was to be followed for the study and the potential risks involved. The participants were then given a consent form and asked to complete it. If they gave their consent to participate in the evaluation, the evaluation procedure was started. The evaluation procedure involved completing a series of questionnaires as well as a series of tasks for two different visual-to-auditory sensory substitution systems. The purpose of the evaluation was to compare the two sensory substitution systems with one another.

The order in which the systems were evaluated was randomised on a per participant basis. The evaluation procedure was as follows:

1. The participant completed the Pre-Evaluation Questionnaire.
2. The participant completed the System Evaluation Procedure for the first system.
3. The participant completed the System Evaluation Procedure for the second system.
4. The participant completed the Final Questionnaire.

The Final Questionnaire was the last step in the evaluation procedure. The entire evaluation procedure took approximately two hours from beginning to end.

Since the purpose of the evaluation was to compare two sensory substitution systems with one another, part of the evaluation procedure was repeated – the part repeated is called the System Evaluation Procedure. The System Evaluation Procedure is a sub-procedure of the evaluation procedure. The details of this sub-procedure are discussed in Section 5.6.1 below.

5.6.1 System Evaluation Procedure

The System Evaluation Procedure was the procedure followed when evaluating a system – it formed part of the entire evaluation procedure. It started with the setup procedure, then the orientation tasks, followed by the navigation tasks and then the object detection tasks – the details of the tasks are discussed in Section 5.4. No feedback was given to the participants regarding their performance on the various tasks. The System Evaluation Procedure is described below in an ordered sequence:

Setup Procedure:

1. The participant was blindfolded.
2. A recording of the “processed_depth_image” and the “processed_color_image” topics was started, these topics were published by the Sensory Substitution Framework (Section 4.4.2). The recording was done using a ROS bag (Section 4.4.1) for review purposes.
3. The participant was equipped with the system.
4. The participant was told that moving one’s head around may make it easier to complete the tasks. That is because it allows the user to perceive more of the surroundings.

Orientation Tasks:

1. The Orientation Task (for object detection) was completed once.
2. The Orientation Task (for navigation) was completed once.

Navigation Tasks:

1. The Navigation Task (no obstacles) was completed once.
2. The Navigation Task (with obstacles) was completed once.

For the navigation tasks, the participants were asked to talk through what they were experiencing and what they thought they were perceiving – if they felt comfortable doing so. For these tasks, the evaluator was always close to the participant, ensuring that the participant did not bump into objects or fall over.

Object Detection Tasks:

1. The Quadrant Task was completed five times.
2. The Box Task was completed five times.
3. The Multiple Boxes Task was completed three times.

For the object detection tasks, the participants were asked to talk through what they were experiencing and what they thought they were perceiving – if they felt comfortable doing so. When placing a box for the Box Task and Multiple Boxes Task, random box placing noises were made using another box to confuse the participants. This was done so that the participants were not able to identify information about the box and its position using sounds not produced by the system – they were only to use the sounds generated by the system for the object detection.

Questionnaire Completion:

1. The system was taken off the participant.
2. The recording of the "processed_depth_image" and the "processed_color_image" topics was stopped.
3. The participant's blindfold was removed.
4. The participant completed the Post-Evaluation Questionnaire.

Completing the Post-Evaluation Questionnaire (Appendix E) was the last step of the System Evaluation Procedure.

5.7 Evaluation Metrics

This section discusses the metrics collected as part of the evaluation. Both quantitative and qualitative metrics were collected. This section is broken up into two parts: the metrics gathered through the evaluation tasks (Section 5.4), and the metrics gathered through the questionnaires (Section 5.5).

5.7.1 Evaluation Task Metrics

These are the metrics recorded for each of the different tasks described in Section 5.4. The quantitative metrics recorded for each task are given below. No qualitative metrics are discussed here, however, for each of the tasks, the evaluator took notes of any anomalies noticed by the evaluator or expressed by the participants. Participants were not told how any of the metrics – such as errors – were counted. All the Evaluation Task Metrics were recorded on the Evaluator's Sheet (Appendix G – note that pages 2 to 5 of the appendix were repeated for each system). On the Evaluator Sheet, the order in which the two systems were evaluated was also noted.

Table 5-1: Metrics Recorded for the Navigation Task (no obstacles)

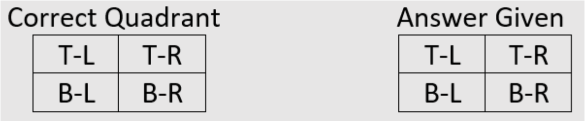
Type	Description	Unit
ToT	Time on task	Minutes and Seconds
E_{wall}	Wall touch	Integer (Error Count)
E_{ror}	Reoriented/went off course	Integer (Error Count)
E_{end}	Realized was at the end	Boolean (True or False)
E_{other}	Other errors – e.g. discrepancies between the real world and what they say they perceived	Integer (Error Count)

Navigation Task (no obstacles): The metrics recorded for that task are shown in Table 5-1. For E_{wall}, E_{ror} and E_{other} error types, the number of errors was recorded using the tally mark system (𝍸). The moment the participants made an E_{wall} or E_{ror} error, they were re-centred between the left and the right walls of the passage, and set in the correct orientation. The purpose of collecting these metrics for the Navigation Task (no obstacles) was to enable conclusions to be drawn about how well each system performed when navigating a simple environment – i.e. navigating through a straight narrow passage.

Table 5-2: Metrics Recorded for the Navigation Task (with obstacles)

Type	Description	Unit
ToT	Time on task	Minutes and Seconds
E_{wall}	Wall touch	Integer (Error Count)
E_{obj}	Object touch	Integer (Error Count)
E_{ror}	Reoriented/went off course – excluding missing turns	Integer (Error Count)
E_{t1}	Failed to make turn	Integer (Error Count)
E_{gap}	Failed to make gap	Integer (Error Count)
E_{t2}	Failed to make turn into narrow passage	Integer (Error Count)
E_{end}	Realized was at the end	Boolean (True or False)
E_{other}	Other errors – e.g. discrepancies between the real world and what they say they perceived	Integer (Error Count)

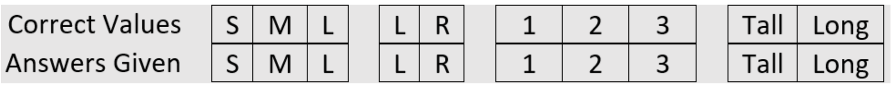
Navigation Task (with obstacles): For this task, the barrier placement (Section 5.4) was recorded. Table 5-2 shows additional metrics recorded for the Navigation Task (with obstacles). For E_{wall}, E_{obj}, E_{ror}, E_{t1}, E_{gap}, E_{t2} and E_{other} error types, the number of errors was recorded using the tally mark system (𝍸). The moment the participants made an E_{wall}, E_{obj} or E_{ror} error, they were re-centred between the left and the right walls of the passage, and their orientation was corrected when necessary. The procedure for errors E_{t1}, E_{gap} and E_{t2} during the navigation task was to allow a maximum of 2 errors per error type. After one of these error types, the participants were walked back roughly 2m, re-centred between the left and the right walls of the passage, set in the correct orientation and told that they may continue. If the participant had made two errors of a specific type (either E_{t1}, E_{gap} or E_{t2}), the participant was centred at the beginning of the relevant turn (or gap) and told to continue. The participant was considered to have failed to make the gap (E_{gap}) if they touched the barrier. The participant was considered to have failed to make the left (E_{t1}) or right turn (E_{t2}), if they walked past the turn without promptly realising they had done so, and if they made a complete turn towards the wall when the turn was further on. The purpose of collecting the metrics for the Navigation Task (with obstacles) was to enable conclusions to be drawn about how well each system performed when navigating a complex environment – i.e. a following a path with turns and obstacles.



Correct Quadrant		Answer Given	
T-L	T-R	T-L	T-R
B-L	B-R	B-L	B-R

Figure 5-5: Quadrant Task marking system

Quadrant Task: For this task, three metrics were recorded: did the participant get the left-right position correct, did the participant get the top-bottom position correct, and did the participant get the exact quadrant correct. This was done using the Quadrant Task marking system (shown in Figure 5-5) for each of the five times the Quadrant Task was completed (per system) – as part of the System Evaluation Procedure (Section 5.6.1). The Correct Quadrant – which was randomly generated – was marked on the Quadrant Task marking system prior to the evaluation (i.e. for each system, there were five copies of Figure 5-5). During the actual evaluation, the answer given was recorded on the Quadrant Task marking system under Answer Given (Figure 5-5). The purpose of collecting these metrics for the Quadrant Task was to enable conclusions to be drawn about how well each system performed at identifying the top-bottom (B/T) and left-right (L/R) position of a relatively isolated object.



Correct Values	S	M	L	L	R	1	2	3	Tall	Long
Answers Given	S	M	L	L	R	1	2	3	Tall	Long

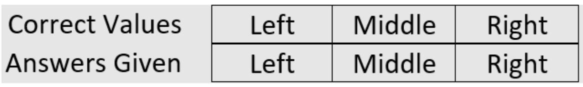
Figure 5-6: Box Task marking system

Box Task: For this task, four metrics were recorded each time the task was completed. These four metrics were:

1. The box size (S, M or L) the participant believed it was, compared to the actual box size.
2. The horizontal position (L or R) in which the participant believed the box was placed, compared to the actual horizontal position of the box.
3. The distance (1 = 0.8m, 2 = 1.6m, 3 = 2.4m) at which the participant believed the box was placed, compared to the actual box distance.
4. The orientation (Tall = portrait or Long = Landscape) in which the participant believed the box was placed, compared to the actual box orientation.

This was done using the Box Task marking system (shown in Figure 5-6) for each of the five times the Box Task was completed (per system) – as part of the System Evaluation Procedure (Section 5.6.1). The Correct Values for the box size, horizontal position, distance and orientation were marked on the Box Task marking system prior to the evaluation – these values were randomly generated. During the actual evaluation, the participant's answers were recorded on the Box Task marking system under Answers Given (Figure 5-6). The purpose of collecting these metrics for the Box Task was to enable conclusions to be drawn about how well each system performed.

Specifically, when identifying information about an object's size and position when there was interference from a larger object in a known position, e.g. a table.



Correct Values	Left	Middle	Right
Answers Given	Left	Middle	Right

Figure 5-7: Multiple Boxes Task marking system

Multiple Boxes Task**:** For this task, three metrics were recorded each time the task was completed. These three metrics were:

1. Whether the participant believed there was a box on the left, compared to whether there was.
2. Whether the participant believed there was a box in the middle, compared to whether there was.
3. Whether the participant believed there was a box on the right, compared to whether there was.

This was done using the Multiple Boxes Task marking system (shown in Figure 5-7) for each of the three times the Box Task was completed (per system) – as part of the System Evaluation Procedure (Section 5.6.1). The Correct Values for whether a box was placed on the left, whether a box was placed in the middle, and whether a box was placed on the right were marked on the Multiple Boxes Task marking system prior to the evaluation – these values were randomly generated. During the actual evaluation, the participant's answers were recorded on the Multiple Boxes Task marking system under Answers Given (Figure 5-7). The purpose of collecting these metrics for the Multiple Boxes Task was to enable conclusions to be drawn about how well each system performed when identifying how many objects there were at a specified distance when there was interference from a larger object in a known position, e.g. a table.

5.7.2 Questionnaire Metrics

Questionnaire metrics are the metrics recorded for each of the different questionnaires described in Section 5.5. The majority of metrics recorded in the questionnaires were qualitative metrics. Participants were not told how they performed in the System Evaluation Procedure for either of the two systems – neither before nor after completing the questionnaires.

Pre-Evaluation Questionnaire**:** This questionnaire gathered information about whether the participant had prior experience with SSDs, in addition to identifying whether the participant was suitable for the evaluation – i.e. are they able to hear with both ears, and are they able to hear within the required frequency range. The metrics gathered using the Pre-Evaluation Questionnaire (Appendix D) are listed below:

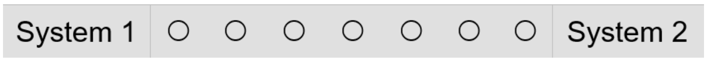
1. Whether the participant had any previous experience using sensory substitution systems. If so, the sensory substitution device and duration it was used for was recorded.
2. The participant's age range, from the options 18-20, 21-30, 31-40 or 40+.
3. The participant's gender, from the options male or female.
4. Whether the participant was able to hear with their left ear.
5. Whether the participant was able to hear with their right ear.
6. The participant's audible frequency range.

The evaluator performed the left-right ear test, as well as the test for the participant's audible frequency range while using bone conduction headphones (Section 6.3.1). These tests were done using instrumental music ("Stereo Audio Test," 2011) played separately to the left and right ears, and using a video ("20Hz to 20kHz - Human Audio Spectrum," 2012), which played the full human audible frequency range (Section 2.2.2). If the participant had previous experience with sensory substitution systems, or was not able to hear with both ears, or did not have the required audible frequency range, the evaluation was terminated as the participant did not meet the requirements for the study.

***Post-Evaluation Questionnaire*:** This questionnaire gathered subjective information about the participant's experience with a single system. The metrics gathered using the Post-Evaluation Questionnaire (Appendix E) are listed below:

1. What they enjoyed about the system – this was an open-ended question.
2. What they did not enjoy about the system (issues, difficulties, etc.) – this was an open-ended question.
3. Additional feedback about the software – this was an open-ended question.

This feedback was used to identify issues with the systems, and possible suggestions for changes to the systems.



System 1 ○ ○ ○ ○ ○ ○ ○ System 2

Figure 5-8: System Preference Rating

***Final Questionnaire*:** This questionnaire primarily gathered subjective information about the participant's experience of one system compared to the other. The metrics gathered using the Final Questionnaire (Appendix F) are listed below:

1. How much the participant preferred one system over another – this was done using the System Preference Rating shown in Figure 5-8.
2. Why the participant preferred the chosen system – this was an open-ended question.
3. Feedback regarding changes to the hardware – this was an open-ended question.
4. Any additional feedback the participant wanted to give – this was an open-ended question.

This feedback was used to identify which – of the two systems evaluated – the participant preferred, why they preferred it, and what could be improved. It was also used to identify potential issues with the hardware used.

5.8 Conclusions

This chapter reviewed the evaluation design. The metrics gathered covered a broad range of scenarios using consistent metric gathering techniques. The metrics provided a broad overview of how well the systems performed in different scenarios. This enabled the researcher to make comparisons between the two systems evaluated, drawing conclusions about the usability of the system for navigation. Additionally, due to the nature of the metrics gathered, the evaluation enabled the researcher to draw conclusions about which systems performed better, and in which ways they performed better. The next chapter looks at the implementation and evaluation of the system.

Chapter 6: Design Cycle 1

6.1 Introduction

Chapter 4 looked at the Sensory Substitution Framework, which can be used for developing sensory substitution prototypes. Chapter 5 looked at the evaluation design for evaluating sensory substitution prototypes. This chapter uses these two prior chapters as it focuses on the system design, implementation details and evaluation results. It does this with the aim of addressing:

> **RQ 5:** *"What visual-to-auditory sensory substitution techniques can be used to develop a visual-to-auditory sensory substitution prototype for depth perception?"* and;
> **RQ 6:** *"Does the prototype developed provide the visually impaired with an accurate understanding of their surroundings through audition?"*

The chapter starts off by giving a system overview, discussing the core ideas behind the prototypes. It then looks at the hardware and software used to implement the prototype visual-to-auditory sensory substitution algorithms. A breakdown of Design Cycle 1 is then given, followed by each of the phases in the first design cycle.

6.2 System Overview

The visual-to-auditory prototypes developed used the ability to localize sounds in a three-dimensional space as a substitution for vision. This works by capturing a depth map (information about the distance to surfaces in a scene) of what is in the user's field of view. Using this depth map, the system places virtual sound emitters on the surfaces in the user's field of view. Sound emitters are placed in three-dimensional space with each sound emitter relating to a point on a surface.

The idea, based on the sensory substitution literature and one's ability to localize sounds is that, over time, through neural plasticity (the brains ability to adapt to new information), one would be able to associate the placement of virtual sound emitters with points on surfaces in a scene. This then enables the formation of a type of depth map in one's mind, hence, creating spatial perception.

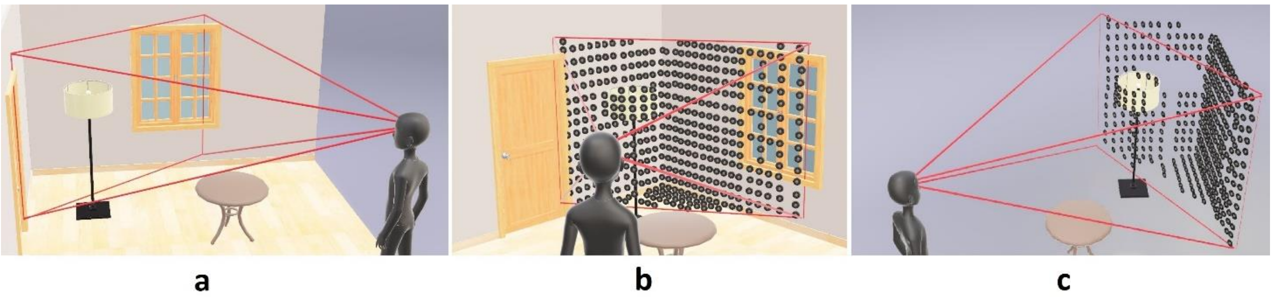


Figure 6-1: Visualisation of the virtual speaker placement in a scene, for a SSD which uses sound localization

Figure 6-1 (a) shows the user's field of view using red lines. Figure 6-1 (b) shows how the virtual sound emitters could be placed for this scene. Figure 6-1 (c) shows an alternative view (with the walls, floor, door and window removed) of where the virtual sound emitters would be placed for this scene. In Figure 6-1 (c), one can see that the virtual sound emitters were not placed on the part of the wall obscured by the lamp (relative to the user's view point) – since the user would not have been able to see that surface.

6.3 Hardware and Software Used

This section looks at the hardware and then the software used for the development and evaluation of the prototypes. The hardware design subsection looks at the hardware design as well as the specifications of the hardware used. The software subsection goes over the software used, as well as the software structure used for the prototypes.

6.3.1 Hardware Design

The final hardware design consisted of a depth camera mounted to a hard hat, a laptop strapped to the outside of a backpack and a pair of bone conduction headphones, see Figure 6-2 (a). The laptop was strapped to the outside of the backpack to allow for cooling, preventing the system from overheating or slowing down due to CPU thermal throttling. The depth camera was attached to the hard hat using a custom designed 3D printed mount shown in Figure 6-2 (b) and Figure 6-2 (d). The depth camera provided image data to the laptop via USB, the same USB cable powered the depth camera. The bone conduction headphones, shown in Figure 6-2 (c), were connected to the laptop via a standard 3.5mm audio jack.

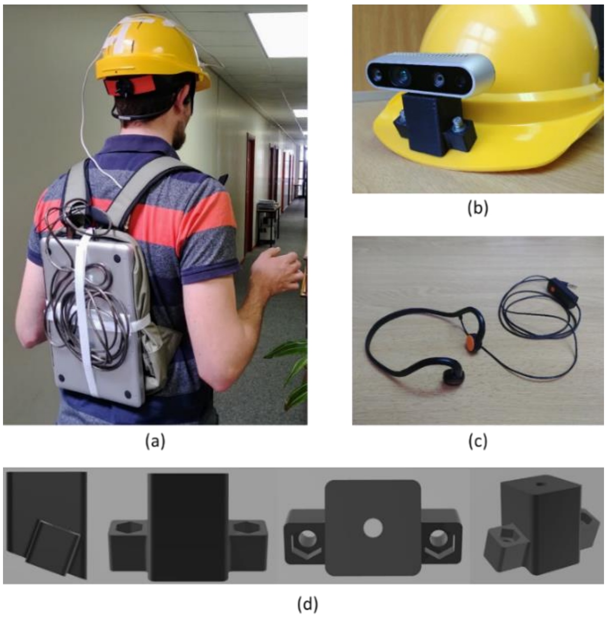


Figure 6-2: Hardware used for the prototype. The full sensory substitution system (a), a close-up of the depth camera mounted to a hard hat (b), a close-up of the bone conduction headphones (c), multiple angles of the 3D printed mount used to connect the depth camera to the hard hat (d).

The depth camera used was the Intel RealSense D435. The laptop used was an Apple MacBook, which was powered through its internal battery. The bone conduction headphones used were the Aftershokz Sportz 2 (AS320). The hardware specifications for the depth camera, laptop and headphones are listed in Table 6-1, Table 6-2, and Table 6-3 respectively.

Table 6-1: Intel RealSense D435 Specifications (Pruitt, 2018)

Intel RealSense D435 Specifications		
The Description	**Specification**	**Setting Used**
Depth Filed of View (FoV) — (Horizontal × Vertical) for HD 16:9	91.2° x 65.5° (+/- 3°)	-
Depth Stream Output Resolution	Up to 1280 x 720	480 x 270
Depth Stream Output Frame Rate	Up to 90 fps	30 fps
Minimum Depth Distance (min-z)	0.11 meters	-
Maximum Range	Approx. 10 meters (Accuracy varies depending on calibration, scene, and lighting condition)	-
RGB Sensor Resolution & Frame Rate	1920 x 1080 at 30 fps	480 x 270 at 30 fps
RGB Sensor FoV (Horizontal × Vertical)	69.4° x 42.5° (+/- 3°)	-
Camera Dimension (Length x Depth x Height	90 mm x 25 mm x 25 mm	-
Connection Type	Single USB for both power and data	-

Table 6-2: Apple MacBook Specifications

Apple MacBook Specifications	
Description	**Specification**
Operating System	Ubuntu 16.04
CPU	Intel® i5-3210M
RAM	8GB

Table 6-3: Aftershokz Sportz 2 (AS320) Specifications (Newegg, n.d.)

Aftershokz Sportz 2 (AS320) Specifications	
Description	**Specification**
Speaker Type	Bone Conduction Transducers
Sensitivity	100 +- 3dB
Frequency Response	20Hz – 20KHz
Connection Type	3.5mm Audio Jack

Three depth cameras were tested throughout the initial stages of development: the Microsoft Kinect for Windows, the Intel RealSense D415 and the Intel RealSense D435. After completing the Baseline Prototype, it was decided that the D435 would be used for further development and throughout the evaluation process.

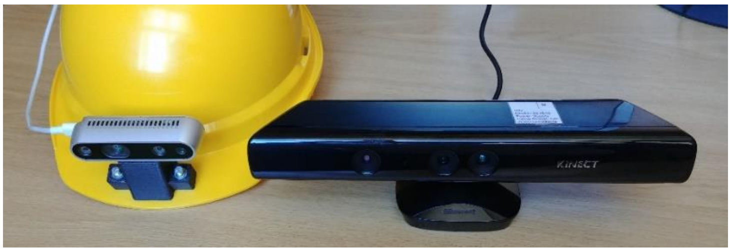


Figure 6-3: Intel RealSense D435 mounted onto a hard hat next to a Microsoft Kinect

The Intel RealSense cameras had the advantage of being more portable than the Microsoft Kinect for Windows (Figure 6-3) – this is due to their size and the fact that they are USB powered, unlike the Kinect which is powered through a wall socket. The D435 was chosen over the D415 primarily due to the D435's wider field of view – approximately 90 degrees compared to the D415's 70 degrees.

6.3.2 Software

PDO 3.a was to implement each prototype using a standardised framework. The main piece of software used for developing all the visual-to-auditory sensory substitution prototypes was the Sensory Substitution Framework (SSF), hence achieving *PDO 3.a*. The structure of the SSF – and thus, the structure for the prototypes developed – is shown in Figure 6-4. The Pre-processor, Retinal Encoder and the Sound Generator nodes are the core components (Chapter 4). The Pre-processor was primarily responsible for pre-processing the depth and colour images produced by the depth camera – the colour images were only used for review purposes, although future prototypes could use them for improving the algorithms. The Retinal Encoder was responsible for processing the images into a compressed form using principles from the human visual system. The Sound Generator was responsible for the auditory processing. The Sound Generator used psychoacoustic principles to generate the soundscape (Section 3.4.1) from the retinal encoded image; the Sound Generator was also responsible for playing the soundscape. For all prototypes, the Sound Generator used OpenAL – a positional audio library, which allows for the virtual placing of sounds in 3D space (Section 2.2.5) – to play sounds; more specifically, the PyAL (von Appen, 2017) Python wrapper for the popular OpenAL library. OpenAL uses many of the sound localization principles (this achieves *PDO 3.d*).

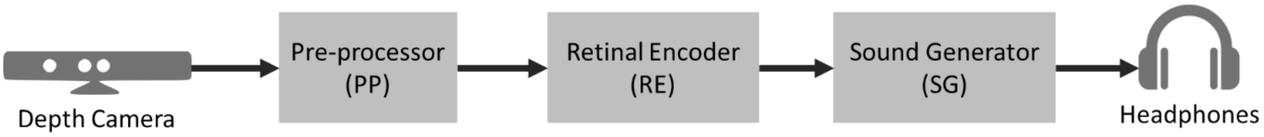


Figure 6-4: Sensory Substitution Framework Structure

The depth and colour images were obtained from the Intel RealSense D435 depth camera. The installation of the ROS Wrapper for Intel RealSense Devices was required to use this camera with the SSF. The ROS Wrapper for Intel RealSense Devices is a set of packages for using Intel RealSense cameras with ROS. The Intel RealSense D435 depth and colour cameras could be launched after installation. Using the ROS Wrapper, the D435 was able to publish the raw image data to topics to which the Pre-processor subscribed, and so, the flow of events shown in Figure 6-4 could proceed.

6.4 Design Cycle 1 Overview

Design Cycle 1 focuses on the implementation of the Baseline Prototype (P_0) and Prototype 1 (P_1). The reason it focused on both is that without P_0, a comparative study could not be completed. Figure 6-5 illustrates the different phases that make up the first design cycle. As can be seen in Figure 6-5, Design Cycle 1 consists of two sub cycles. The first sub cycle being the implementation and then preliminary study of the Baseline Prototype (P_0). In this case, a preliminary study is a small-scale study done using one or two participants – the same procedure described in Section 5.6 was followed.

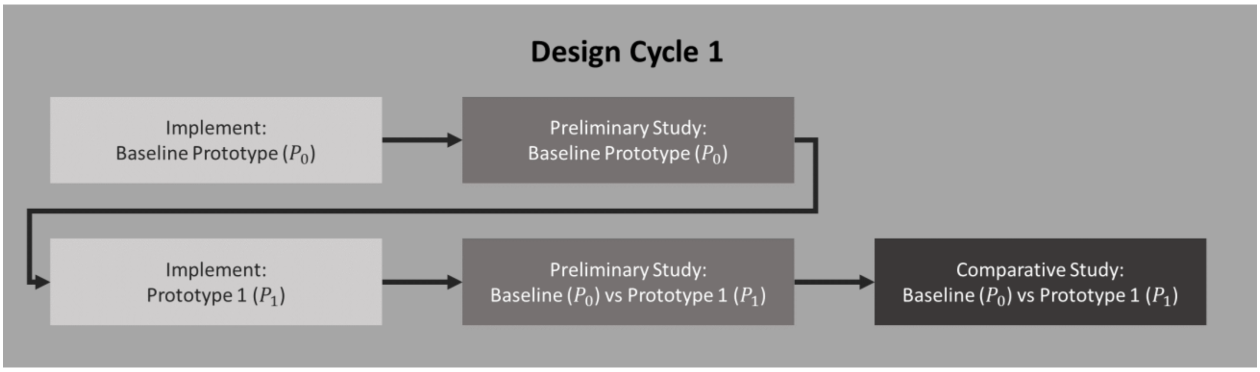

Design Cycle 1
Implement:
Baseline Prototype (P_0)
Preliminary Study:
Baseline Prototype (P_0)
Implement:
Prototype 1 (P_1)
Preliminary Study:
Baseline (P_0) vs Prototype 1 (P_1)
Comparative Study:
Baseline (P_0) vs Prototype 1 (P_1)

Figure 6-5: Overview of Design Cycle 1

The only difference between a preliminary study and the full-scale comparative study was the number of participants used. For the first sub cycle of Design Cycle 1, only one round of the System Evaluation Procedure (Section 5.6.1) was completed since there was no system to compare to P_0. The preliminary study of P_0 provided the opportunity to identify any potential issues with the software, as well as an opportunity to identify possible improvements. A set of recommendations for potential improvements was then generated based on knowledge gained from the implementation of P_0, in addition to the knowledge gained from the preliminary study.

The second sub cycle was the implementation of Prototype 1 (P_1), the preliminary study and then the comparative study between P_1 and P_0. The implementation of P_1 used the recommendations generated from the implementation and preliminary study of P_0. The preliminary study between P_1 and P_0 then gave the opportunity to fix any potential issues with the software, as well as an opportunity to make minor tweaks to P_1 before doing the more intensive comparative study. A set of recommendations for potential improvement were then generated based on knowledge gained from the implementation of P_1, and the comparative study between P_1 and P_0.

6.5 Baseline Prototype (P_0)

P_0 was the first fully functional prototype developed for this project. It was the sensory substitution algorithm used as a point of comparison for the other prototype iterations. It was also the algorithm used as a starting point for improvement (*PDO 3*). As discussed in Section 3.6, the MeloSee algorithm was chosen to be the baseline algorithm. This section discusses the researcher’s implementation of the P_0, the preliminary study done for the P_0, and the recommendations generated from the evaluation.

6.5.1 Implementation

PDO 2 was used to implement P_0 using the standardised framework. Hence, as discussed in Section 6.3.2, the SSF (Chapter 4) was used for the implementation of P_0. The main components of the SSF are: the Pre-processor, Retinal Encoder and Sound Generator. For P_0, no Pre-processor was implemented, so the RAW depth image from the camera bypasses the Pre-processor and gets used directly by the Retinal Encoder. The implementation of the Retinal Encoder and Sound Generator are discussed below.

6.5.1.1 Retinal Encoder

The Retinal Encoder for P_0 was the researcher's implementation of the retinal encoder discussed in the MeloSee papers (Fristot et al., 2012; Stoll et al., 2015). The visual processing component (the Retinal Encoder) of the MeloSee algorithm took the original depth image, and down sampled the depth image to an 8 by 8 image. As discussed in Section 3.4.3, downscaling was done by approximating the mean of the surrounding pixels for each pixel in the downsampled depth image. The approximation of the mean was done by sampling 10 neighboring pixels from the original depth image, these pixels were randomly sampled according to a 2D normal distribution – using standard deviations calculated based on the width and height of the original depth compared to the width and height of the downsampled image.

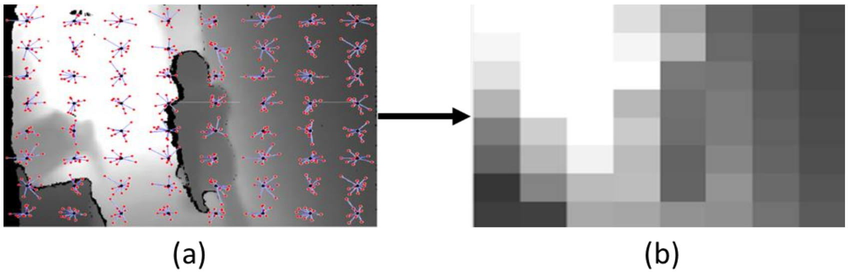


Figure 6-6: Depth Image with overlay of the sampling process (a) next to the output from the Baseline Prototype's Retinal Encoder (b)

Figure 6-6 is an illustration of the downscaling process and the resulting retinal encoded image (the downsampled image). On the left of Figure 6-6, one can see an illustration of the 10 randomly sampled pixels for each of the 64 (8 x 8) pixels in the final retinal encoded image.

6.5.1.2 Sound Generator

The Sound Generator implemented for P_0 is the researcher's implementation of the auditory processing part of the MeloSee algorithm (Section 3.4.3). The algorithm works by associating:

- Pitch with vertical position;
- Relative left-right gain (respective loudness on the left and right) with horizontal position; and
- The tone loudness to distance.

For the vertical position, the higher the pixel, the higher the pitch – humans naturally associate pitch with height (Section 2.2.3.2). The different pitch sounds used ranged from 264Hz (low) to 528Hz (high) on the just intonation scale (Table 3-1); low tones were associated with pixels at the bottom of the retinal encoded image, high tones were associated with pixels at the top of the retinal encoded image. Using these different aspects of sound means that the entire image can be converted to a soundscape and played in one go; hence, the soundscape can be updated in real-time, rather than using the scanline technique (Section 3.4.1).

The way that this was implemented was as follows. The first time the Sound Generator received the retinal encoded image from the Retinal Encoder, the setup process was run. This involved creating the virtual sound sources and attaching the appropriate sounds. Using PyAL (the 3D positional audio library), a sound source was generated for each of the 64 (8 x 8) pixels in the retinal encoded image. For each sound source, the appropriate tone was attached based on the row its related pixel was in – the tones were pre-generated and stored as .wav files, these were able to be played by PyAL. For example, if the pixel was in the 1st row of the image (i.e. the highest row), the tone attached to the related sound source would be the 528Hz tone, whereas if it were the 8th row, the 264Hz tone would be attached. Once the setup was completed, the sound sources were able to be independently and dynamically positioned using PyAL. To position a sound source one simply sets the x, y and z position of the sound source (in meters) and the audio library dynamically performed the sound processing to virtually place the sound – it did this using a number of techniques (discusses in Section 2.2.3). All positioning was relative to the listener's head.

The positioning of sound sources was used to effectively mimic how the MeloSee algorithm dealt with horizontal position and distance. The position of the sound sources was obtained from the retinal encoded image. Depending on the column the pixel was in (i.e. horizontal position), the x-value for the associated sound source was set between -1.75m (column 1) and 1.75m (column 8).

The distance of a sound source (z-value) was set based on the depth value of the associated pixel. Since the retinal encoded image was a scaled down depth image, the distance was simply the value of the pixel. The maximum distance was set to 3m; for any distance further than 3m, the associated sound source was muted. The setup process linked the vertical position to pitch, so the y-value (vertical position) was left at 0m; this again was done to mimic how the MeloSee algorithm works. The dynamic positioning of the sounds happened for every depth frame received, hence they were repositioned at a rate of 30Hz.

While implementing the P_0 Sound algorithm, the problem of audio clipping (hard clipping) was encountered (Section 2.2.4). For example, combining two sine waves, 264Hz and 352Hz each with an amplitude of 0.6 (where the maximum is 1.0) would result in clipping, since at some peaks the amplitude would reach up to 1.2 (0.6 + 0.6), this is illustrated in Figure 6-7. Additionally, in Figure 6-7 one sees both constructive and destructive interference at different points.

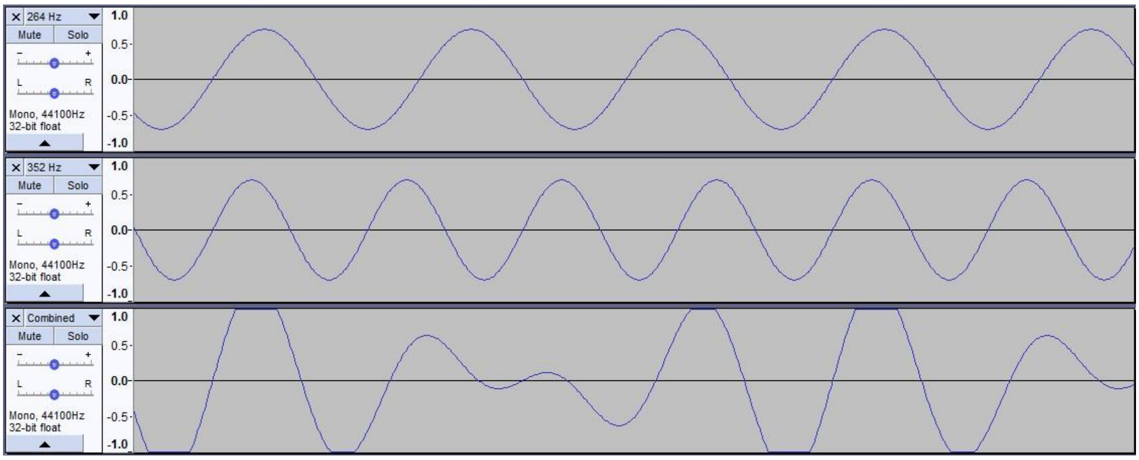


Figure 6-7: Waveforms generated in Audacity by combining 264 Hz and 352 Hz sound waves

When playing the sounds back simultaneously through PyAL, the sounds would get combined; if the resulting wave had peaks that had an amplitude greater than the system threshold, then the audio would be clipped as illustrated in Figure 6-7. To solve this issue, the amplitude of the sound files generated was set to 0.95 (where the maximum would be 1), and then the gain for each sound source was divided by the number of sound sources. This meant that a peak amplitude of 95 $percent$ could be reached.

6.5.2 Preliminary Study

Once the implementation and subjective testing had been completed by the researcher, a small-scale preliminary study was performed using two participants. Both participants were able to hear the range of sounds used in the prototypes. The preliminary study was performed to identify potential improvements to be made in future prototype iterations. Since only P_0 was being evaluated, the full comparative study (Section 5.6) was not conducted; rather, a Pre-Evaluation Questionnaire and a single System Evaluation Procedure (Section 5.6.1) was conducted. The System Evaluation Procedure involves completing all the tasks discussed in Section 5.4, as well as completing the Post-Evaluation Questionnaire. The results from the preliminary study are briefly discussed below.

6.5.2.1 Questionnaires

The qualitative feedback was gathered from the questionnaires. Additional qualitative feedback was gathered from notes the evaluator made while observing the participants. Table 6-4 shows the general positive and negative comments and notes made regarding the system. Overall, the participants felt that the system was useful for navigating, although still challenging. They felt that the system would become easier to use with practice, but even without much practice they felt the system was helpful for knowing when they were getting close to walls or large objects. Small objects on the other hand were harder to detect. This is likely due to the resolution of the retinal encoded image, in addition to the averaging technique used for generating the retinal encoded image – this appears to blur smaller objects into the background.

Table 6-4: Baseline Prototype Qualitative Feedback

	Positive	Negative
1	Close large objects were easy to detect (i.e. walls)	Floor dominated sound
2	System became easier to use with practice	Table dominated sound
3	-	Sound anomalies/artefacts (users commented they didn't know if it was intended for detection)
4	-	Low objects and small objects against a plane (table/floor/wall) were hard to detect
5	-	Distance distinction: struggled to determine how far away something was, only knew when it was right in front of them
6	-	Left-right spacing: struggled to accurately gauge distance to left and right

Participants also felt that during the navigation tasks, the floor dominated the sound, and during the object detection tasks, the table dominated the sound. This happened since the floor as well as the table covered a significant proportion of the visual scene. Finally, participants struggled with determining the distance to objects. They were only able to tell whether or not a large object (i.e. a wall) was very near, however distinguishing the precise depth was found to be challenging – even for large objects.

6.5.2.2 Tasks

The tasks completed for the preliminary study were the Navigation Task (no obstacles), the Navigation Task (with obstacles), the Quadrant Task, the Box Task and the Multiple Box Task. For the Navigation Task (no obstacles), each participant touched the wall once. For the Navigation Task (with obstacles), there were an average of 3.5 wall touches, and 2.5 object touches. During the completion of the Navigation Tasks using P_0, the researcher could see the participants avoiding larger objects such as walls as the participants came close to these objects.

For the Quadrant Task, the participants were able to determine left from right $60\ percent$ of the time, and top from bottom $40\ percent$ of the time. For the Box Task, the participants got the box size correct $70\ percent$ of the time, whether the box was to the left or right $50\ percent$ of the time, the distance to the box $30\ percent$ of the time and the orientation of the box $40\ percent$ of the time. For the Multiple Box Task, the participants were correct at determining whether there was a box on the left, middle and right, $33\ percent$, $67\ percent$ and $17\ percent$ respectively.

It should be noted that the sample size for the preliminary study was only two ($n = 2$). For all the tasks, a larger sample size would yield greater clarity in the results. Due to the very small sample size, a combination of observational data, qualitative and quantitative data from the participants, knowledge from the literature, and subjective testing by the researcher was used to gauge whether or not the system functioned as a visual-to-auditory sensory substitution system – and hence could be used as a baseline prototype. It was determined that P_0 could successfully operate as the baseline, as participants found it helpful for navigating their environment.

6.5.3 Recommendations

The recommendations for improving the system are discussed below. These recommendations are based on the Background Research, the review of Existing Systems, the experience gained from implementing P_0 (Section 6.5.1), as well as the feedback from the preliminary study (Section 6.5.2).

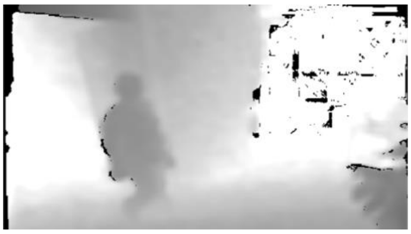


Figure 6-8: RAW Depth Image

First, it was noticed that the depth images produced by the depth camera had a fair amount of noise, as shown by the black spots in Figure 6-8. For this reason, it was recommended that a pre-processor should be implemented to clean up the depth image. Below are some recommendations for the pre-processor:

- ***Cropping:*** Cropping the depth image would get rid of the sensor's dead zones, reducing the depth image noise.
- ***Temporal Filter:*** Using data from previous frames, a temporal filter would fill in the pixels which have NaN values (i.e. points where the depth camera could not determine the depth). This would also reduce the minor depth errors, for example in the spots in Figure 6-8, one may be able to see the slight inconsistency in the walls gradient (i.e. the depth

values); this is due to the minor depth errors. The temporal filter would also reduce these errors. As a result, this would reduce the subtle crackling noise generated by the Sound Generator because of the NaN values (Table 6-4 row 3).

- ***Down Scaling:*** This means that there is less data for the algorithm to process.

One of the problems that was identified was depth accuracy (Table 6-4 row 4 and 5). With the goal of improving the depth accuracy, some recommendations for improvements to the retinal encoder are listed below:

- ***Quantization:*** Changes in depth appeared hard to identify in the testing of P_0. It was assumed that this had to do with the fact that there was a large amount of noise, and hearing subtle changes amongst that noise proved difficult. Because of this, rather than having continuous subtle changes, one could test discrete changes (as opposed to continuous). To achieve this, one could quantize the depth information. With each level getting further apart, approximating how our depth perception accuracy decreases with distance.
- ***Minimum Value Downscaling:*** Downscaling the depth image using minimum value downscaling. This means that for any given group of pixels, the nearest pixel will be used as the sample. One of the goals being to identify small objects in the scene, as small objects would potentially not be picked up.

It was noted that the participants struggled to tell whether an object was close on their left or their right, as well as the distance to the left or the right (Table 6-4 row 6). With that issue in mind, two recommendations for improving the Sound Generator are given below:

- ***Single Complex Sound:*** To improve the sound localization accuracy, especially regarding elevation (e.g. identifying a table from a roof), the literature recommends the use of complex sound (Section 2.2.3.2). One could start off with a single complex sound.
- ***Perspective Projection:*** With the implementation of P_0, sound separation on the left and right was relatively poor. To improve on this and make the movement of the sounds more aligned with reality, when placing the virtual sounds, perspective projection (along ray) could be used rather than orthogonal projection.

The researcher hypothesized that using the above recommendations would improve the random noise in the depth image, the depth accuracy, and the top-bottom left-right accuracy.

6.6 Prototype 1 (P_1)

P_1 was developed based on the recommendations generated from the implementation and preliminary study of P_0. This section reviews the implementation, preliminary study and comparative study of P_1. Based on what was learnt from the implementation and studies, a set of recommendations for potential improvement are given.

6.6.1 Implementation

As mentioned in Section 6.3.2, P_1 was implemented using the SSF (Chapter 4). The main components of the SSF are: the Pre-processor, Retinal Encoder and Sound Generator. As per *PDO 3.b*, all the components were implemented to ensure the system worked in real-time. The implementation of the Pre-processor, Retinal Encoder and Sound Generator are discussed below.

6.6.1.1 Pre-processor

Noise in RAW depth images primarily occurs when the depth camera is not able to resolve the depth for a certain point. This happens for various reasons, resulting in a NaN (Not a Number) value for the depth value that could not be determined – NaN values are rendered as black pixels as shown in Figure 6-9. The Pre-processor was responsible for cleaning up the depth image, this primarily involved techniques to clean up the noise in the image. It also involved reducing the image size to improve algorithm speed. The three techniques used were cropping, temporal filtering and then scaling – in that order.

Cropping: The first step in preprocessing the image was to crop it. As can be seen on the left of the RAW Depth Image in Figure 6-9, there are a large number of pixels for which the depth camera could not determine the distance. This “dead zone” on the left of the RAW Depth Image often occurs as a result of how depth images are generated (i.e. disparity mapping). For this reason, an equal amount was cropped off on the left and right of the RAW Depth Image to get rid of the commonly occurring noise. For the Pre-processor, $5\,percent$ was cropped off the left, and $5\,percent$ cropped off the right, the center of Figure 6-9 shows the result of this cropping. The cropping function was implemented as part of the SSF Core (Section 4.4.5).

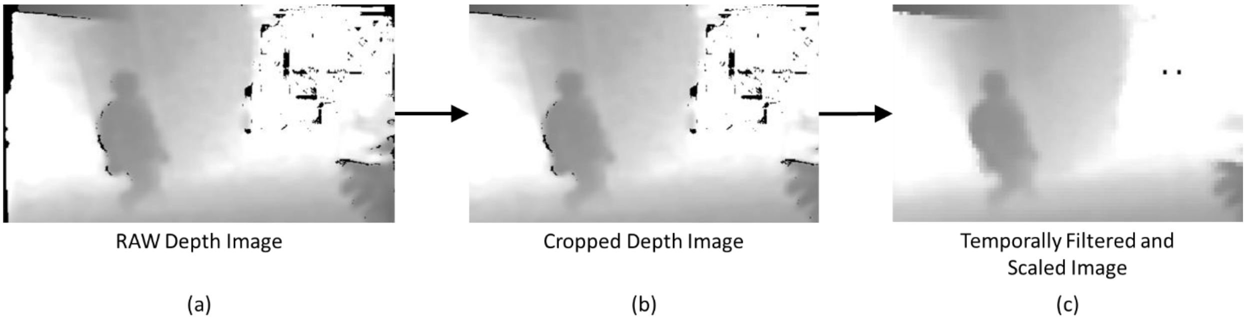


Figure 6-9: Pre-processor Event Flow

Temporal Filter: After cropping, a temporal filter was applied to reduce noise in the RAW depth image. A temporal filter is a filter that is applied over time. In this case, it was an averaging filter applied over the last $n + 1$ depth frames. This meant that if the specified number of frames is $n = 2$, and the current frame is the 235th frame, the depth values for each respective pixel of the 233rd, 234th and 235th frames would be averaged to produce the temporally filtered frame. This smoothed out the inconsistencies. The custom temporal filter algorithm implemented had the additional condition that when it encountered a NaN value, it did not include it in the averaging process. This exclusion of the NaN values in the averaging process is what made the algorithm so effective at reducing noise. This was because there would need to be n frames with the exact same pixel having a NaN value in order for the NaN value to appear in the temporally filtered frame (the output image from the temporal filter).

The temporal filter algorithm worked as follows. It had a buffer which kept the last n frames. For each new frame, the buffer was updated by dequeuing the oldest frame from the buffer and appending the current frame to the buffer. The depth values for each pixel were then averaged with the respective pixels across the frames in the buffer – excluding any NaN values from the averaging process. The result was a de-noised depth image – the temporally filtered frame. This process was repeated for each new depth frame received. For the first n frames, no temporal filtering was applied, since the buffer needed to be filled.

For the pre-processor the number of temporal filter frames was set to $n = 2$. One does not want to set n to be too large, as this would result in the image becoming blurry with fast movement. But with a small n, these unwanted effects were not noticeable and noise in the depth image was often dramatically reduced. In the temporally filtered image on the far right of Figure 6-9, one can see the spottiness of the wall was decreased, in addition to a dramatic reduction in the image noise. The temporal filter function was implemented as part of the SSF Core (Section 4.4.5).

Down Scaling: Finally, the image was downscaled. This was done to reduce the amount of information that needed to be processed in future stages of the algorithm. Reducing the amount of information to be processed reduced the amount of time taken to process that information, resulting in a faster algorithm. Many researchers in the field of machine learning use 96 x 96 pixel images for image recognition tasks (Coates, Lee, & Ng, 2011; Khorrami, Le Paine, & Huang, 2015). Some researchers use 32 x 32 images to improve algorithm performance (Le et al., 2010). Based on this, it was decided to scale the image down to be 96 pixels wide – the height was then determined to be 60 pixels based on the image ratio.

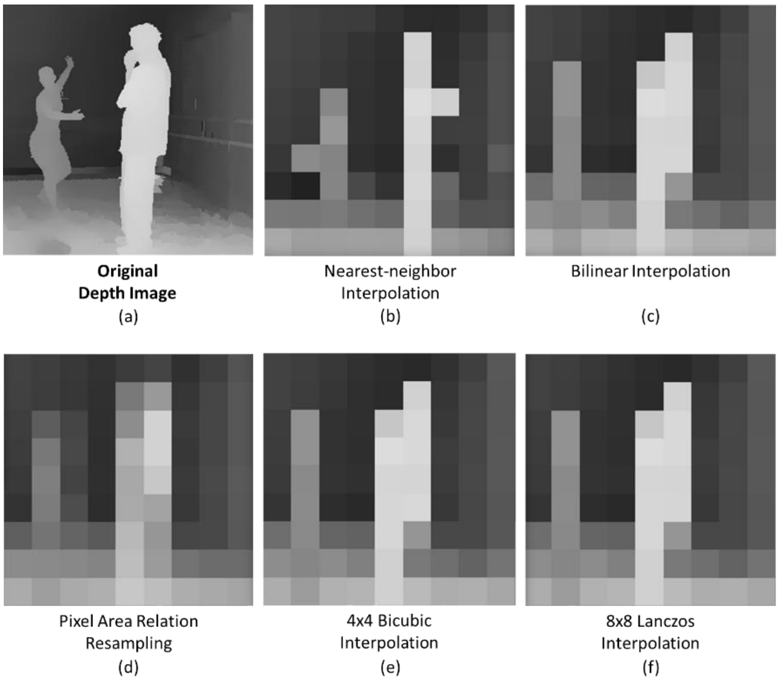


Figure 6-10: Original Depth Image (Suh, Kim, Park, & Suh, 2010) used to generate 9 x 9 scaled down depth images

Several scaling techniques were tested as shown in Figure 6-10 – where the original image was downscaled to a 9 by 9 image. What was looked for was clean depth distinction, for example, it can be seen in Figure 6-10 (d) that the Pixel Area Relation Sampling blurs the depth values in an attempt at antialiasing. This antialiasing is often desired for normal images, however, for depth images, the blurring it performs results in incorrect depth values – for example, resulting in a depth value in between a wall and the person in front of the wall, rather than the depth value for either the wall or the person. The algorithm that was chosen for scaling ended up being the Nearest-neighbor Interpolation algorithm shown in Figure 6-10 (b). Nearest-neighbor Interpolation gave the cleanest depth distinction and resulted in the least incorrect depth values. The result of scaling a depth image to a 96 x 60 image – using Nearest-neighbor Interpolation – is shown in Figure 6-9 (c). Since the images depicted in Figure 6-9 are small, the downscaling is not very apparent. The pre-processor used OpenCV to efficiently scale the colour and depth frames generated by the depth camera.

6.6.1.2 Retinal Encoder

The auditory nerve is able to carry significantly less information to the brain than the optical nerve (Section 3.3). For this reason, the depth image needs to be reduced in an effective way before being converted into sound. The Retinal Encoder was responsible for this information reduction, in addition to using intelligent processing techniques to improve the end result of generating a soundscape for a particular scene. The Retinal Encoder implemented in P_1 did this processing using a temporal filter, quantization and using a minimum value downscaling technique. The Retinal Encoders processing was done on the 96 x 60 pixel preprocessed image received from the Pre-processor; the output from the Retinal Encoder was a 10 x 5 pixel retinal encoded image.

Temporal Filter: First, a temporal filter was applied using $n = 2$. This was the same temporal filter discussed in the Pre-processor implementation section (Section 6.6.1.1). The primary difference being that in the pre-processor, the temporal filter was applied to the cropped RAW depth image, where this was applied to the fully preprocessed image. The reason for this was to further reduce noise in the image.

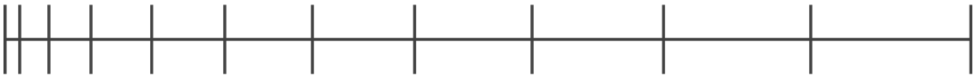


Figure 6-11: Quantization Levels

Quantization: The next step was quantization. This was implemented based on the principle that a sound's loudness decreases proportionally to the distance squared, this is known as the inverse square law (Section 2.2.3.3); hence, it was hypothesized that quantizing the distance levels – using increasing spacing (Figure 6-11) – would produce more noticeable distance changes when "seeing" through sound. The reason for quantizing the depth image was to improve depth accuracy in real-world usage.

$$S_n = \sum_{k=0}^{n-1}(a + kb) = \frac{n}{2}(2a + (n-1)b)$$

Equation 4

Since $a = 0$, and $S_{n_{max}} = d_{max} - d_{min}$:

$$\therefore b = \frac{2 \times (d_{max} - d_{min})}{n_{max}} \div (n_{max} - 1)$$

Equation 5

$$d_n = d_{min} + S_n$$

Equation 6

To simulate the increasing spacing in a useful way, a quantization function was written. The quantization function used the arithmetic sum formula (Equation 4) to calculate the increasing step size (S_n) for each level (n) – using $a = 0$, the initial step size ($b \equiv S_2$) was calculated based on the maximum distance ($d_{max} = 3$), minimum distance ($d_{min} = 0.2$) and the number of quantization levels ($n_{max} = 12$). The calculation for determining the initial step size (b) is shown in Equation 5. The initial step size (b) for P_1 ended up being $0.0\overline{42}$, using a minimum distance of $0.2m$, a maximum distance of $3m$ and 12 quantization levels; this resulted in the quantization levels d_1 to d_{12} being: $[0.2,\ 0.24,\ 0.33,\ 0.45,\ 0.62,\ 0.84,\ 1.09,\ 1.39,\ 1.73,\ 2.11,\ 2.53,\ 3]$ when rounded to two decimal places. The distance for each level was calculated using Equation 6. The quantization function now forms part of the SSF Core (Section 4.4.5).

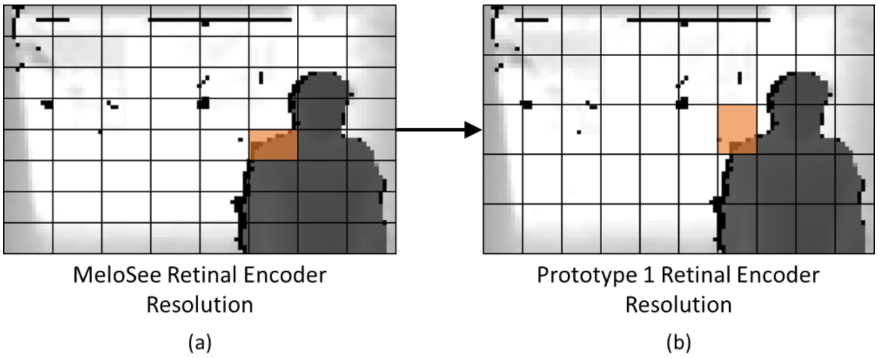

MeloSee Retinal Encoder Resolution
(a)
Prototype 1 Retinal Encoder Resolution
(b)

Figure 6-12: Grid overlaid on top of the Pre-processed depth image showing the resolution for the Baseline Prototype (a) and Prototype 1 (b) Retinal Encoders

Minimum Value Downscaling: Minimum value downscaling means creating a smaller image by choosing the minimum value from the cluster of pixels. Figure 6-12 shows (via a grid overlay) two possible ways in which the pre-processed depth image (from the Pre-processor) can be divided into clusters of pixels – a single cluster is highlighted. The number of clusters represents the number of pixels – i.e. the resolution – the retinal encoded image would have, and each cluster of pixels is used as the sample set for the respective pixel in the downscaled image.

For the P_1 Retinal Encoder, a resolution of 10 pixels across was chosen for two reasons; the first is that humans are able to locate a sound's azimuth (horizontal) more accurately than its elevation (vertical), hence having a higher horizontal pixel count. Additionally, due to the dimensions of the pre-processed depth image, 10 x 5 resolution provided pixels which were closer to being square. This meant that sound emitters covered roughly the same height as they did width.

The reason for using the minimum value was to avoid incorrect depth values. This is illustrated by the circled areas in Figure 6-13, as one can see the MeloSee Retinal Encoder (i.e. the P_0 Retinal Encoder) produced a pixel lighter than the original Pre-processed depth image and lighter than that produced by the P_1 Retinal Encoder – this means that the person's shoulder would be seen as farther away than in reality. The reason this happens is that the MeloSee algorithm approximates the average of the pixels in that cluster; in this case a little less than half the pixels were part of the background and a little less than half were part of the person's shoulder. This resulted in a depth value, which was roughly halfway between the person's shoulder and the background, even though in this case there was nothing at that position in space – hence, it is an incorrect depth value. On the other hand, the P_1 Retinal Encoder chose the closest object in that area (the minimum depth value). Figure 6-12 shows the exact cluster of pixels chosen for the respective algorithms, and Figure 6-13 shows the results of the respective processing.

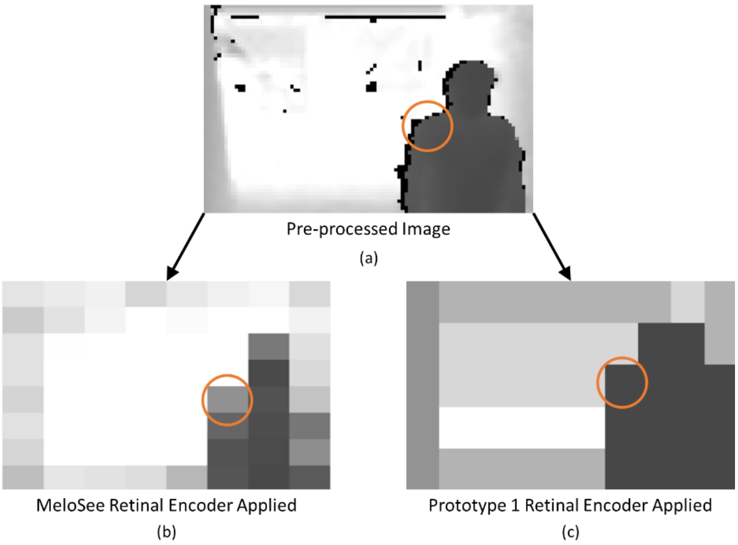

Pre-processed Image
(a)
MeloSee Retinal Encoder Applied
(b)
Prototype 1 Retinal Encoder Applied
(c)

Figure 6-13: MeloSee Retinal Encoder (b) vs Prototype 1 Retinal Encoder (c)

The results of the combination of temporal filtering, quantization and minimum value downscaling are shown in Figure 6-13. On the far left of three images in Figure 6-13 is a wall, and on the right is a person. Comparing the far left hand sides of the retinal encoded images – Figure 6-13 (b) and Figure 6-13 (c) – it can be seen that the P_1 Retinal Encoder produced a uniform column of depth values representative of the wall. The MeloSee Retinal Encoder also showed that something was on the far left, although it is less uniform. On the right side of the same respective images, it can be seen that the P_1 Retinal Encoder produced a more uniform representation of the person. This is a good representation of how the combination of quantization and minimum value downscaling was able to group objects based on their depth values – making objects more distinct.

6.6.1.3 Sound Generator

The Sound Generator was responsible for taking the retinal encoded image produced by the Retinal Encoder, and generating a soundscape from that image. The Sound Generator for P_1 built on the concepts used in the MeloSee algorithm (i.e. the P_0 Prototype's Sound Generator). That is, by associating:

- Pitch with vertical position;
- Relative left-right gain (respective loudness on the left and right) with horizontal position;
- The sound loudness to distance.

As with the MeloSee algorithm, using these associations – one for each dimension (achieving *PDO 3.c*) – means that the soundscapes can be generated in real-time (achieving *PDO 3.b*) rather than using the scanline technique (Section 3.4.1). Although P_0 and the P_1 Sound Generators are very similar, there are two core differences between them. The first is that the P_1 Sound Generator used a single complex sound, changing the pitch of the complex sound, rather than using different pitched tones. The second is that the P_1 Sound Generator used perspective projection rather than orthogonal projection for positioning the sound emitters in 3D space. Perspective projection is how one perceives the world through sight (Figure 6-14). As with the P_0 Prototype, the Sound Generator algorithm developed for P_1 uses PyAL (the 3D positional audio library). The way the Sound Generator was implemented is described below.

Single Complex Sound: When the Sound Generator received the first retinal encoded image, the setup process was run. This involved loading the complex sound (a flowing river); then, for each pixel ($10 \times 5 = 50$ pixels) in the retinal encoded image, a sound source was generated and a copy of the complex sound was attached to each sound source. The gain for each sound source was then set relative to the number of sound sources; this was done to avoid clipping (Section 2.2.4) – this gain setting was separate from the left-right loudness, and the overall loudness relative to distance, each of which were dynamically managed by PyAL based on the sound source position. Once the sound sources had been setup, each sound source was then played in a loop. Finally, depending on the row of pixels the sound source was related to, the pitch was set:

- The 1st row was set to 1.7;
- The 2nd row to 1.3;
- The 3rd row to 1.0;
- The 4th row to 0.7;
- The 5th row to 0.3.

Where the 1st row was the top row of pixels, and where set to 1.5 would mean that the sound was played at 1.5 times the speed, changing the sound's pitch – for example, a $100Hz \times 1.5 = 150Hz$. This resulted in a high-pitched sound at the top, and low-pitched sound at the bottom, as with MeloSee. This implementation mimicked how MeloSee worked, while adding the use of a complex sound; the aim of this was to improve sound localization accuracy while using the system.

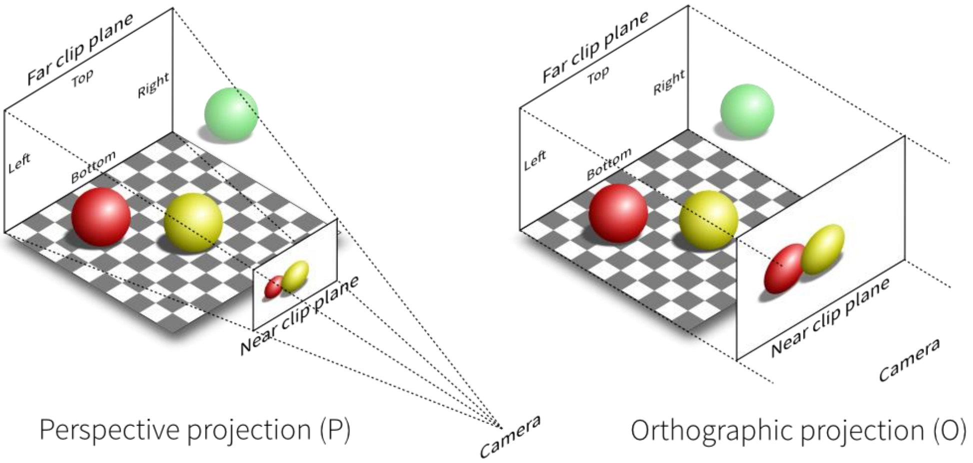


Figure 6-14: Perspective Projection ("Script Tutorials," 2018)

Unit Vector Map: Another part of the setup process was generating the unit vector map. The unit vector map contained a unit vector for each pixel – each unit vector described the perspective projection for its related pixel. This allowed for efficient calculation of the projected position of each pixel. To generate the unit vector map, one needed the pixel width (μ_w) and height (μ_h), and the retinal encoded image width (r_w) and height (r_h). To calculate the pixel width (μ_w) and height (μ_h), one needed the retinal encoded image width (r_w) and height (r_h), and the retinal encoded image horizontal (FoV_h) and vertical (FoV_v) field of view. It should be noted that the retinal encoded images horizontal field of view (FoV_h) was the depth cameras horizontal FoV cropped by $10\ percent$ since the pre-processor crops the depth image by $10\ percent$– the retinal encoded image vertical (FoV_v) was the same as the depth cameras vertical FoV. The calculations were done with respect to a *fixed* arbitrarily chosen distance (ε), P_1 used $\varepsilon = 0.5$. To calculate the pixel width (μ_w), Equation 7 was used, and for pixel height (μ_h) a very similar equation was used, Equation 8.

$$\mu_w = \frac{(2 \times \varepsilon) \times \tan(\frac{FoV_h}{2})}{r_w}$$

Equation 7

$$\mu_h = \frac{(2 \times \varepsilon) \times \tan(\frac{FoV_v}{2})}{r_h}$$

Equation 8

$$x = \begin{cases} p_x \times \mu_w, & r_w\ is\ odd \\ (p_x \times \mu_w) - (\frac{\mu_w}{2}), & r_w\ is\ even\ and\ p_x > 0 \\ (p_x \times \mu_w) + (\frac{\mu_w}{2}), & r_w\ is\ even\ and\ p_x \leq 0 \end{cases}$$

Equation 9

$$y = \begin{cases} p_y \times \mu_h, & r_h \; is \; odd \\ (p_y \times \mu_h) - (\frac{\mu_h}{2}), & r_h \; is \; even \; and \; p_y > 0 \\ (p_y \times \mu_h) + (\frac{\mu_h}{2}), & r_h \; is \; even \; and \; p_y \leq 0 \end{cases}$$

Equation 10

$$l = \sqrt{x^2 + y^2 + \varepsilon^2}$$

Equation 11

$$v_{x,y} = (v_i, v_j, v_k) = (\frac{x}{l}, \frac{y}{l}, \frac{\varepsilon}{l})$$

Equation 12

Once the pixel width and height had been calculated, the unit vector was calculated for each pixel using Equation 12, and hence Equation 11, Equation 10 and Equation 9. The perspective was from the centre of the image; which meant that the unit vectors were calculated from the centre of the image. As shown in Figure 6-15, p_x is the number of pixels from the centre of the image along the x-axis (p_x is negative to the left of the centre), p_y is the number of pixels from the centre of the image along the y-axis (p_y is negative to the bottom of the centre). Each calculated unit vector ($v_{x,y}$) was stored in a 2D array referred to as the unit vector map. The functions for calculating the pixel size as well as generating the unit vector map form part of SSF Core (Section 4.4.5).

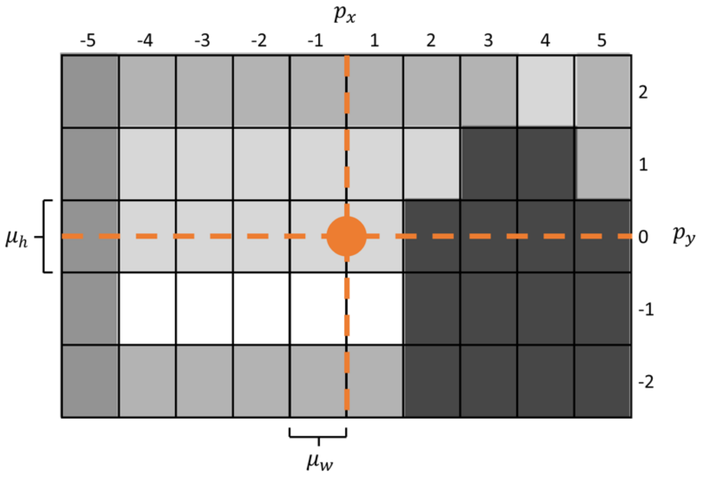


Figure 6-15: Segmentation of the image for generating the unit vector map

After the setup process was completed, the sound sources were able to be dynamically and independently positioned in virtual 3D space using PyAL. This was done by setting the x, y and z position (in meters) for a sound source; PyAL then dynamically performed the sound processing to position the sound source. The positioning was done relative to the listener’s head.

$$f(v_{x,y}, d) = (v_i \times d, v_j \times d, v_k \times d)$$

Equation 13

Perspective Projection: For the positioning of the sound sources, a perspective projection was used (Figure 6-14). Once the unit vector map had been generated, efficient calculation of the projected x, y and z values for a given depth (d) could be done using Equation 13. This was done by obtaining the unit vector (from the unit vector map), which described the specific perspective projection for the given pixel, then scaling that unit vector ($v_{x,y}$) by the depth value (d) given – hence, the projection was dependent on the pixel's placement from the centre of the image (i.e. p_x and p_y) as well as the depth (d) it was projected to. Using a perspective projection was how the depth camera captured the depth image; hence, using perspective projection for the placement of the sound emitters mimicked the actual placement of the points in space, unlike an orthogonal projection, which would have distorted the position of the points captured by the depth camera. Since perspective projection simulates the way objects are perceived by humans in the real world, the aim of implementing it for the placement of the sound emitters was to improve localization accuracy when using the system. The projection function was implemented as part of the SSF Core (Section 4.4.5).

6.6.2 Preliminary Study

A preliminary study was performed, comparing P_0 and P_1. In the first preliminary study an issue was found. The issue had to do with special cases where multiple depth values could not be determined (i.e. were NaN), for reasons such as the depth camera being too close to an object and the minimum working distance was reached – this was not catered for, and caused the program to freeze. When the bug was encountered, this was experienced in the form of a constant sound, rather than a sound that changed with the environment. It is recommended that future researchers test the edge cases for their algorithms. For example, testing how an algorithm handles walking close to a wall and receiving many undetermined depth values in the depth buffer. After fixing the issue with P_1, another preliminary study was performed to verify that everything behaved as expected. After verifying that all was working as expected, the comparative study was performed.

6.6.3 Comparative Study

This section looks at the results for the comparative study done between P_0 and P_1. The results discussed below were obtained by processing the metrics gathered – the metrics gathered are discussed in Section 5.7. For the comparative study between P_0 and P_1, there were a total of eight participants ($n = 8$), each one completing the comparative study evaluation procedure (Section 5.6). One may refer to Section 5.2 for further details on the participant selection.

The Pre-Evaluation Questionnaire (Section 5.7.2) revealed that: six of the participants were male and two were female; none of the participants had any prior experience using sensory substitution systems; all participants were between the ages of 21 and 30; all the participants had the ability to hear in their left and right ears. The highest hearable minimum frequency in the group was 190Hz, and the lowest maximum hearable frequency was 15000Hz, meaning all participants could hear between 190Hz and 15000Hz.

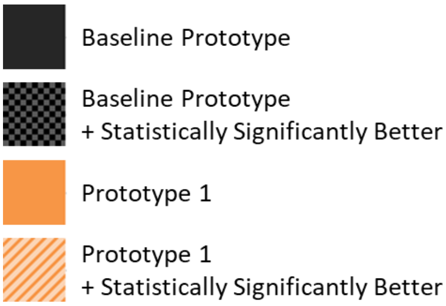

Baseline Prototype
Baseline Prototype
+ Statistically Significantly Better
Prototype 1
Prototype 1
+ Statistically Significantly Better

Figure 6-16: Graph legend

For statistical comparison of the results obtained from the study, the paired t-Test for two sample means was used; the Chi-Squared Goodness of Fit Test was also used where appropriate. For all the statistical tests, $\alpha = 0.05$ was used, and the hypotheses were:

$$H_0: \mathit{The\ two\ prototypes\ performed\ equally\ well}$$
$$H_1: \mathit{One\ prototype\ performed\ better\ than\ the\ other}$$

If the $p\text{-}value$ generated for a given t-Test was less than α ($p\text{-}value < \alpha$) that meant that the null hypothesis could be rejected; in which case it was concluded – with a $95\ percent$ confidence – that there was a statistically significant difference between the two prototypes. Figure 6-16 shows the legend used to indicate which results are for which prototype, and whether those results are statistically significant.

6.6.3.1 Questionnaires

This section summarizes the results from the Post-Evaluation Questionnaires and the Final Questionnaires, comparing P_0 to P_1. This was done primarily by looking at the qualitative data gathered. From the questionnaires (Section 5.7.2), the positive and negative feedback for P_0 was sorted and summarized into Table 6-5.

It should be noted that these tables are an overview of participants' feedback, this means that participants may have given certain feedback about one system, but not another. For example, if one participant made a comment about preferring one aspect of a certain system, and no one else made that comment; it cannot be concluded that the rest of the participants did or did not prefer a given system in that regard. The only thing that can be said is that one participant preferred a specific part of a specific system. It is with this in mind that these summary tables were compiled, and only when general trends appear are general statements made.

Table 6-5: Baseline Prototype (P_0) Qualitative Feedback from the P_0 vs P_1 study ($\mathrm{n} = 8$)

	Positive	Negative
1	A minority of participants preferred the tonal sounds of P_0 compared to the sounds used for P_1	Some participants felt the tonal sounds used cause fatigue
2	One participant felt the harmonies provided a good indication of distance	Some participants found it difficult to detect changes in the tonal sound – the changes seemed small
3	-	One participant felt the merging of the sounds made navigation difficult (the participant said the same thing about P_1)
4	-	One participant found it difficult to tell the difference between being close to the end of a passage and at the end of a passage (the participant said the same thing about P_1)
5	-	A majority of participants felt that object detection was less reliable than P_1
6	-	One participant felt the left side was slightly louder than the right
7	-	Sound anomalies/artefacts
8	-	Low objects and small objects against a plane (table/floor/wall) were hard to detect

As reflected in Table 6-5, a minority (less than $50\ percent$) of the participants enjoyed the tonal sounds, where others found the constant tonal sounds tiring. Some participants also mentioned they found it difficult to distinguishing small changes in the tonal sounds. Considering this, it makes sense that participants found it difficult to detect low objects against a plane – especially during navigation where the scene can be fairly complex. Some participants heard unexpected sound anomalies such as random crackling noises. One participant felt that the left side sounded slightly louder than the right. It was unclear what caused this.

As was done for P_0 in Table 6-5, the positive and negative feedback for P_1 was sorted and summarized into Table 6-6. This was done primarily by looking at the qualitative data gathered from the questions asked in the questionnaires (Section 5.7.2).

Table 6-6: Prototype 1 ($\boldsymbol{P_1}$) Qualitative Feedback from the $\boldsymbol{P_0}$ vs $\boldsymbol{P_1}$ study ($\mathbf{n = 8}$)

	Positive	**Negative**
1	Some participants found the sounds used were less tiring, as compared to P_0 it was more comfortable to use for longer periods of time	One participant felt the upper quadrant's feedback dominated the lower quadrant's feedback
2	Some participants found it easier than P_0 to differentiate between openings on the left and the right, making it easier to determine where the gaps were when navigating	One participant felt the merging of the sounds made navigation difficult (the participant said the same thing about P_0)
3	A majority of participants felt that object detection was more reliable than P_0	One participant found it difficult to tell the difference between being close to the end of a passage and at the end of a passage (the participant said the same thing about P_0)
4	Some participants felt that the distance accuracy was better than with P_0	Low objects and small objects against a plane (table/floor/wall) were hard to detect
5	One participant felt it was easier to identify and learn the different sounds	-

As can be seen in Table 6-6, some participants found that the sounds used in P_1 were less tiring than those used in P_0. Some participants also found that P_1 made it easier to find gaps on the left or right, and they felt the distance accuracy of P_1 was better than P_0 – this would make navigation slightly easier. Finally, the majority of participants felt that the object detection of P_1 was more reliable than P_0. One participant did however comment that they felt the upper quadrants feedback dominated the lower quadrants feedback.

With both P_0 and P_1, one participant mentioned that they felt as though the sounds merged together, making navigation difficult. Another participant mentioned finding difficulty with both systems when trying to distinguish being at the end of the passage versus being close to the end of the passage. For both P_0 and P_1 there was also a general consensus that low objects and small objects against a plane (table/floor/wall) were hard to detect – the same was said in the Preliminary Study of P_0 (Section 6.5.2). In fact, with both systems, participants mentioned throughout the study that they found the Box Task and the Multiple Boxes Task difficult – some even mentioning they felt as though they were guessing.

Table 6-7: System preference rating for P_0 vs P_1 normalized to be between -3 and 3 ($\mathrm{n} = 8$)

Evaluation	P_0 (-3) vs P_1 (3)
1	2
2	-2
3	2
4	2
5	2
6	2
7	1
8	-2
Mean	0.875
Variance	3.268

Considering that the majority of the positive feedback was given to P_1 one may conclude that in general, users preferred P_1 over P_0. To check this, a paired t-Test for two sample means was performed on the preference ratings given by the participants (Table 6-7). The resulting $p\text{-}value$ was 0.107. Since $p\text{-}value > \alpha$, H_0 was not rejected. This means that with the given data, one should not generalize and conclude that users preferred one system over another.

The majority ($75\ percent$) of the selected participants, preferred P_1 to P_0. The primary reason for this was that they felt the object detection was more reliable than P_0. Additional reasons included feeling that the distance accuracy was better than with P_0 and that with P_1 it was easier to identify and learn the sounds – as one participant put it.

Finally, from the additional feedback, it became clear that many of the participants felt that given more time to use and learn the systems, their performance using both systems would improve. Regarding the system hardware, participants felt that it could be made smaller and lighter.

6.6.3.2 Navigation Task (no obstacles)

Figure 6-17 shows the average number of E_{wall} errors (number of wall touches) made per participant while using P_0 and while using P_1 – the less wall touches, the better. P_0 had an average of 1.625 wall touches for the navigation task with no obstacles, while P_1 had 1.375 wall touches for the same task. From the results of the task, a $p\text{-}value$ of 0.258 was calculated using a paired t-Test for two sample means. Since $p\text{-}value > \alpha$, H_0 was accepted; and so, the t-Test determined there was not a statistically significant difference between the average number of errors (number of wall touches) for this task.

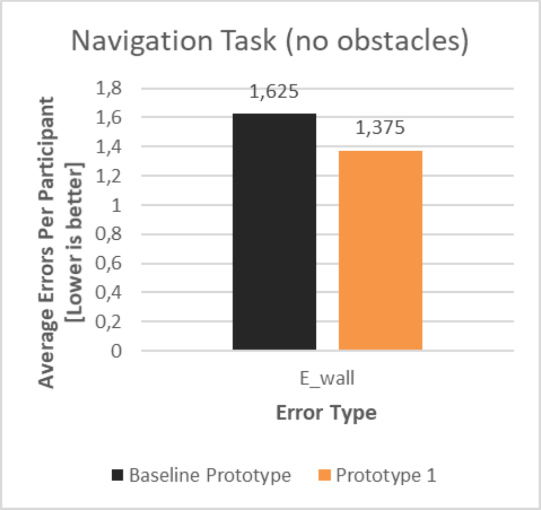


Figure 6-17: Average number of E_{wall} errors made per participant ($n = 8$) in the Navigation Task (no obstacles)

From the t-Test it was concluded that there was no significant difference between P_0 and P_1 when considering the number of wall touches while navigating a straight passage with no obstacles. This indicated that for navigating a simple environment, the two systems performed equally well.

6.6.3.3 Navigation Task (with obstacles)

Figure 6-18 shows the mean number of errors obtained – across a number of error categories (E_{wall}, E_{obj}, E_{ror}, E_{t1}, E_{gap}, E_{t2}) – while using P_0 and P_1 for the navigation task with obstacles. Initially, looking at the results shown in Figure 6-18 seems to reveal that P_1 was an improvement across almost all error categories. To check this conclusion, a paired t-Test for two sample means was run for each category of error. By comparing the resulting $p\text{-}values$ to α for each category of error, it was determined that there were no statistically significant differences for the average number of errors between P_0 and P_1, except in the case of E_{t2} (Figure 6-18).

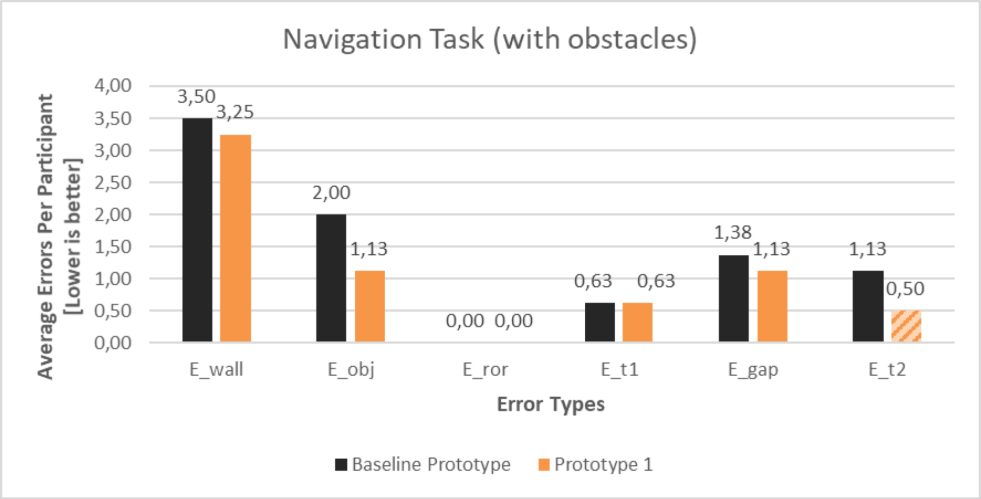


Figure 6-18: Average number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles) – grouped by error type

For the error category E_{t2} (failed to make turn into narrow passage), a $p\text{-}value$ of 0.006 was calculated, and since $p\text{-}value < \alpha$, H_0 was rejected. This means that with 95 $percent$ confidence one can say that on average, P_1 performed better than P_0 Prototype at finding narrow passages. Interestingly, this result agrees with the qualitative feedback that some participants gave; mentioning that they found it easier to differentiate between openings on the left and the right when using P_1 (Table 6-6 row 2).

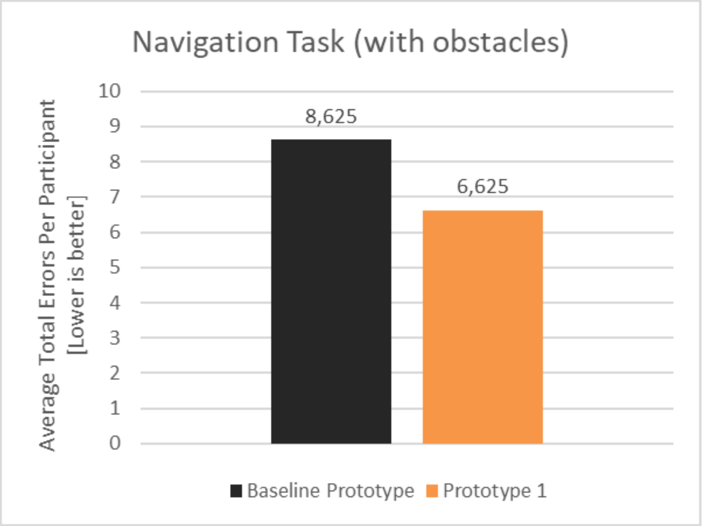


Figure 6-19: Total number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles)

Figure 6-19 shows the average number of errors made across all error categories grouped together (i.e. combining error categories E_{wall}, E_{obj}, E_{ror}, E_{t1}, E_{gap}, E_{t2}), showing this for both P_0 and P_1. A paired t-Test for two sample means was run on the overall error count. As expected from the t-Tests performed on individual error categories, it was determined that there were no statistically significant differences in the average number of overall errors between P_0 and P_1.

For the navigation task with obstacles, from looking at the graphs (Figure 6-18 and Figure 6-19), it appears that P_1 performs better than P_0 in general, the t-Tests performed show that this could not be concluded with statistical confidence – except in the case of finding narrow passages (error category E_{t2}). Hence, it was concluded that for navigating a complex environment, the two systems performed equally well; with the exception that P_1 improved on P_0 when it comes to finding narrow passages.

6.6.3.4 Quadrant Task

For the quadrant task, the data collected was first split into whether the participant got the left-right (L-R) discrimination correct and whether the participant got the top-bottom (T-B) discrimination correct. Figure 6-20 shows the total number of correct answers for L-R and T-B across the two systems (P_0 and P_1); the maximum number of correct answers was 40

($8\,participants \times 5\ = 40$). As can be seen from the results in Figure 6-20, P_1 outperformed P_0 for both vertical and horizontal position discrimination.

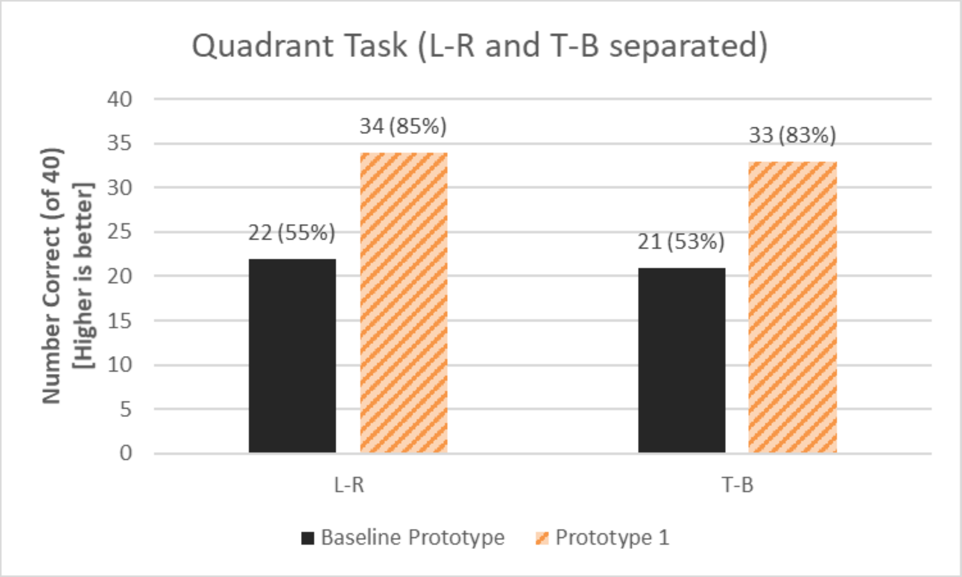


Figure 6-20: Total number of correct answers for the Quadrant Task across all participants ($n = 8$); separated into correct answers for left-right (L-R) and top-bottom (T-B)

To check whether the results were statistically significant, a Chi-Squared Goodness of Fit Test was performed independently for both the L-R and the T-B discrimination of the quadrant task. This was done to determine if there was a statistically significant difference in the accuracy of P_0 and P_1 when it comes to discriminating between left and right, and, discriminating between top and bottom; for an isolated object one meter in front of the participant.

Table 6-8: Contingency table for left-right discrimination

	Correct	Incorrect
Observed Data (P_1)	34	6
Expected Data (Baseline Prototype)	22	18

Table 6-8 is a contingency table for the L-R discrimination. Performing a Chi-Squared Goodness of Fit Test on this data generated a $p\text{-}value$ of 0.00014; since $p\text{-}value < \alpha$, H_0 was rejected. This means that with $95\,percent$ confidence one can say that, P_1 performed better than P_0 at discriminating between left and right for a close isolated object.

Table 6-9: Contingency table for top-bottom discrimination

	Correct	Incorrect
Observed Data (P_1)	33	7
Expected Data (Baseline Prototype)	21	19

Table 6-9 is a contingency table for the T-B discrimination. Performing a Chi-Squared Goodness of Fit Test on this data generated a $p\text{-}value$ of 0.00015; since $p\text{-}value < \alpha$, H_0 was rejected. This means that with $95\,percent$ confidence one can say that, P_1 performed better than P_0 at discriminating between top and bottom for a close isolated object.

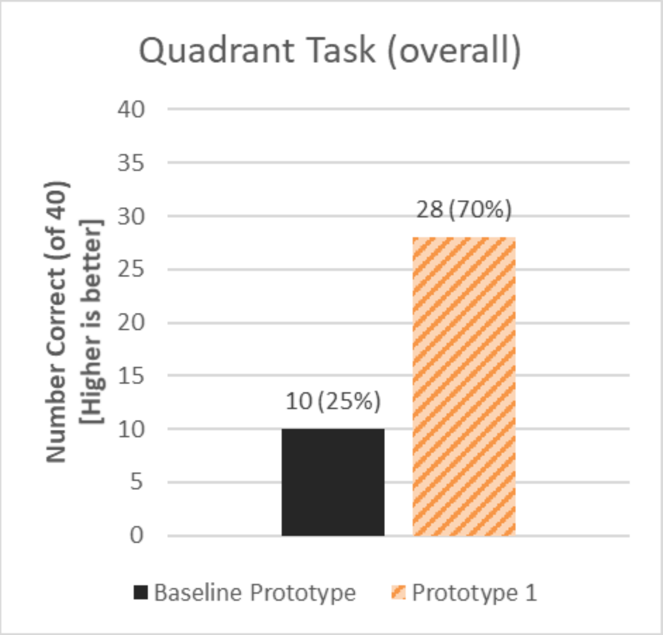


Figure 6-21: Total number of correct answers for the Quadrant Task across all participants ($n = 8$)

Figure 6-21 shows the collective number of times the participants got the quadrant correct (i.e. L-R and T-B position both correct) – this is shown for P_0 and P_1. Looking at the results in Figure 6-21 it can be seen that P_1 was a significant improvement over P_0. This was confirmed by performing a Chi-Squared Goodness of Fit Test on the data, resulting in a $p\text{-}value$ of 4.9×10^{-11}, hence H_0 was rejected. It is also interesting to note that P_0 achieved results equivalent to random guessing, with $25\,percent$ correct for the quadrant test; this could indicate the need for a longer training period to become familiar with the system – something mentioned in the qualitative feedback from Section 6.6.3.1.

From these results, it was concluded that for the given training period, P_1 was significantly more accurate than P_0 at determining the position of a relatively isolated object which is close to the user. This result matches the majority opinion seen in Section 6.6.3.1, being that participants felt that object detection was more reliable with P_1 than with P_0. The result is likely due to a combination of using perspective projection, and the use of a complex sounds.

6.6.3.5 Box Task

Figure 6-22 shows the total number of correct answers for each metric (Size, L-R, Distance and Orientation) in the Box Task; the total number of correct answers are shown for P_0 and P_1.

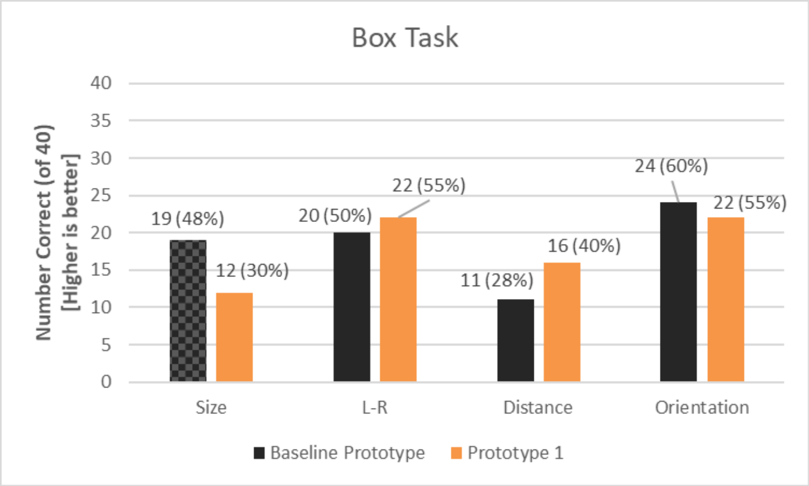


Figure 6-22: Total number of correct answers across all participants ($n = 8$) for the Box Task – grouped by metric

To check whether the results were statistically significant, a Chi-Squared Goodness of Fit Test was independently performed for the size, L-R placement, distance and orientation. For determining the L-R placement, distance and orientation, $p\text{-}value > \alpha$, so no statistically significant difference was found.

Table 6-10: Contingency table size discrimination

	Correct	Incorrect
Observed Data (P_1)	12	28
Expected Data (Baseline Prototype)	19	21

However, for box size discrimination, there was a statistically significant difference. Table 6-10 is a contingency table for the box size discrimination. Performing a Chi-Squared Goodness of Fit Test on this data generated a $p\text{-}value$ of 0.027; since $p\text{-}value < \alpha$, H_0 was rejected. This means that with 95 $percent$ confidence one can say that, P_0 performed better than P_1 at determining the size of an object when there was interference (from something such as a table).

It should be noted that for the task of determining distance, the $p\text{-}value$ calculated was 0.077. This means that with a with an alpha of 0.05 we cannot reject the null hypothesis. But if one is willing to use an alpha of 0.1, then $p\text{-}value < \alpha$, and the H_0 would be rejected. So, although it cannot be concluded with a 95 $percent$ confidence, it can be concluded with a 90 $percent$ confidence that on average, P_1 performs better when determining the distance of an object placed on a table. It should, however, be noted that the sample size is very small ($n = 8$), and that this improvement is not too far above random chance, since with P_1, 40 $percent$ of the time users correctly identified the boxes distance given the option of three distances. Where guessing would've resulted in a 33.3 $percent$ chance of guessing correctly.

The conclusion for the box placement task is that P_0 performed better at determining the box size when there was interference. It was also concluded that it is possible that P_1 performed better at determining distance to the objects when there was interference, although more testing would need to be performed, as this may have been the result of chance. Finally, based on the feedback from the questionnaires (Section 6.6.3.1), it seems that participants felt as though they were guessing with this task. It is in part for this reason that the researcher cautions the reader in the conclusions they take from the results of this task.

6.6.3.6 Multiple Boxes Task

Figure 6-23 shows the total number of correct answers for each metric in the Multiple Box Task. The metrics were the following: was the participant able to correctly determine whether there was a box on the left, a box in the middle and a box on the right. The total number of correct answers are shown for P_0 and P_1.

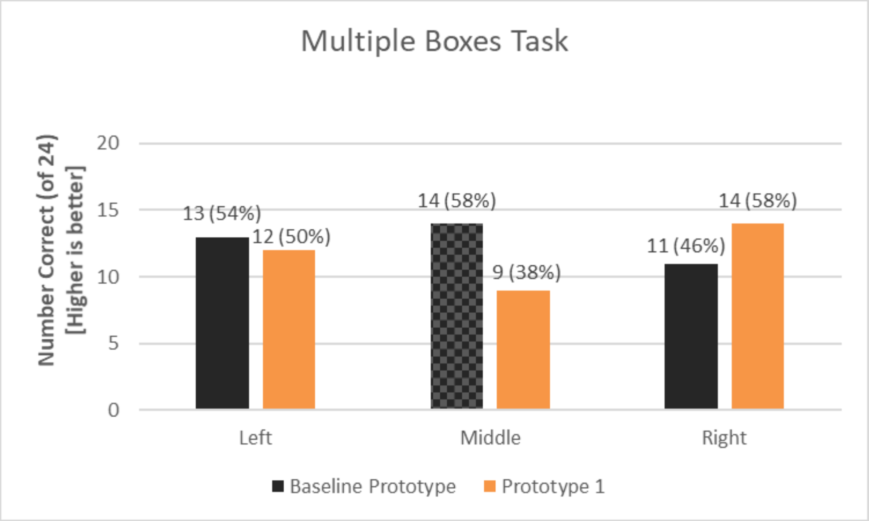


Figure 6-23: Total number of correct answers across all participants ($n = 8$) for the Multiple Box Task – grouped by metric

For boxes on the left and the right there was no statistically significant difference between the accuracy of P_0 and P_1. However, interestingly, for the box in middle, there was a statistically significant difference between the accuracy of P_0 and P_1.

Table 6-11: Contingency table for middle box identification

	Correct	Incorrect
Observed Data (P_1)	9	15
Expected Data (Baseline Prototype)	14	10

Performing a Chi-Squared Goodness of Fit Test on the data in Table 6-11, a $p\text{-}value$ of 0.038 was calculated; since $p\text{-}value < \alpha$, H_0 was rejected. This means that with 95 $percent$ confidence one can say that P_0 performed better than P_1 at helping the participant determine whether or not there was an object in the middle of a table – when the object was in close proximity to the participant.

An interesting observation is that by comparing the graphs in Figure 6-23 and Figure 6-20, one can see that the left-right accuracy for P_0 stays roughly the same – around $50\, percent$ – for the Quadrant Task and Multiple Boxes Task. But the left-right accuracy of P_1 drops from around $80\, percent$ to around $50\, percent$. This is likely an indication that interference with nearby objects such as tables and other boxes decrease the left-right accuracy of P_1. This is possibly due to the granularity of the P_1 quantization, which would be grouping objects such as the table and box together. The idea that the difficulties were due to interference would also align with the feedback from the questionnaires (Section 6.6.3.1), as participants mentioned difficulties detecting objects on tables and near walls.

6.6.4 Recommendations

From the qualitative data gathered in the preliminary study and comparative study of P_0 and P_1, it was found that with P_1 users did not experience any sound anomalies – where with P_0 some users still experienced the sound anomalies (Table 6-4 row 3). This indicated to the researcher that the pre-processor and retinal encoder implemented for P_1 successfully reduced the noise in the raw depth stream. Given that the pre-processor and retinal encoder developed for P_1 worked relatively well, the following recommendations will focus on the Sound Generation part of the algorithm, rather than the pre-processing or retinal encoding. The focus of these recommendations is to improve the sound localization accuracy using different techniques, in addition to improving learnability of the system:

- ***Multiple Complex Sounds:*** For P_1, a single complex sound was used. To improve vertical sound localization, one could try using multiple distinct complex sounds, one for each row of pixels in the retinal encoded image. As per the literature (Section 2.2.3.2), humans naturally associate higher pitched sounds with higher objects – this should be considered. For example, the top row would have a single unchanging high-pitched complex sound associated with it. The bottom would have a single unchanging low-pitched complex sound associated with it. The aim is that this would improve the ability to distinguish objects at different heights – i.e. improving the detection of objects on a table.
- ***Varying Pitch with Distance:*** P_1 used intensity (i.e. loudness) to indicate distance. The louder the sound, the closer the object. For the next iteration, a combination of loudness and varying pitch (relative to the original pitch of the sound) could be used to represent distance, providing the user with two distinct distance queues. The aim being to improve distance accuracy.
- ***Warning Beep:*** Adding a warning beep for when the user is about to bump into a wall. The aim of this being to improve learnability of the system, as well as trust in the system.

Considering that it is only the sound generator that would be updated based on these recommendations, it is not expected that issues such as interference (e.g. when a box is on a table) will be handled any differently, since the same retinal encoder and hence the same quantization, minimum value downscaling and other techniques would be used. For example, it would be expected that there would be a difference between the ability to distinguish objects on the left and right when there is interference and where there is not interference – as seen with P_0 and P_1.

6.7 Conclusion

This chapter looked at the implementation and evaluation of P_0 and P_1. The majority of participants ($75\ percent$) preferred P_1 to P_0 (Table 6-7) – although due to the variance and sample size, the participants preferences were not considered a statistically significant result. The main reason for the preference of P_1 seemed to be that participants felt the object detection was more reliable then P_0. Looking at the results from the Quadrant Task (Section 6.6.3.4), it was clear that for object detection P_1 outperformed P_0 – the improvement was statistically significant (note however that $n = 8$). Another reason participants said they preferred P_1 was that they felt the depth accuracy was better than P_0. Although participants felt this way, looking at the results from tasks like the Navigation Task (Section 6.6.3.3) and the Box Task (Section 6.6.3.5), it was unclear whether P_1 improved on P_0 with regards to depth accuracy, or whether the minor improvements were due to chance. For the navigation tasks in a simple and complex environment, the systems appeared to perform equally well. With the exception of P_1 improving on P_0 when it came to finding narrow passages (Section 6.6.3.3).

At the end of Section 6.5.3, the researcher hypothesized that the recommendations from Section 6.5.3 would improve the random depth image noise (i.e. sound anomalies), the depth accuracy and the top-bottom left-right accuracy. From the Questionnaires (Section 6.5.2.1 and 6.6.3.1), it can be seen that participants continued to experience random sound anomalies with P_0, however with P_1 no sound anomalies were reported. Regarding the top-bottom left-right accuracy, as mentioned earlier, the results from the Quadrant Task (Section 6.6.3.4) show that P_1 improved on P_0 in this regard. This indicates that for the most part, the recommendations made had the intended effect.

The prototypes were developed using the Sensory Substitution Framework (SSF), and hence *PDO 2* and *PDO 3 (a)* were achieved. P_1 also achieved the rest of *PDO 3* objectives since the system worked in real-time, included all three dimensions, used sound localization principles, and in a number of ways P_1 improved on P_0 – although there is definitely room for further development.

The next chapter implements P_2 using the recommendations generated in Section 6.6.4, with the aim of further improving the system. It then goes on to evaluate P_2 against P_0, as was done in this chapter with P_1 and P_0.

Chapter 7: Design Cycle 2

7.1 Introduction

The structure of this chapter is very similar to the previous design cycle chapter (Chapter 6), however this chapter only focuses on the implementation of Prototype 2 (P_2). P_2 is intended to be an incremental improvement over P_1. The premise of P_2 is the same as P_0 and P_1, and hence the system overview (Section 6.2) in addition to the hardware and software used (Section 6.3) are not repeated from Chapter 6. As with Chapter 6, the aim of this chapter is to address:

> **RQ 5:** *"What visual-to-auditory sensory substitution techniques can be used to develop a visual-to-auditory sensory substitution prototype for depth perception?"* and;
> **RQ 6:** *"Does the prototype developed provide the visually impaired with an accurate understanding of their surroundings through audition?"*

The chapter starts off with an overview of Design Cycle 2, followed by each of the phases in the second design cycle. The chapter concludes with a set of recommendations for potential improvements that could be made to P_2.

7.2 Design Cycle 2 Overview

Building on what was learnt from Design Cycle 1, Design Cycle 2 focuses on the implementation of Prototype 2 (P_2). Figure 7-1 illustrates the different phases of the second design cycle, and how these phases link to the first design cycle.

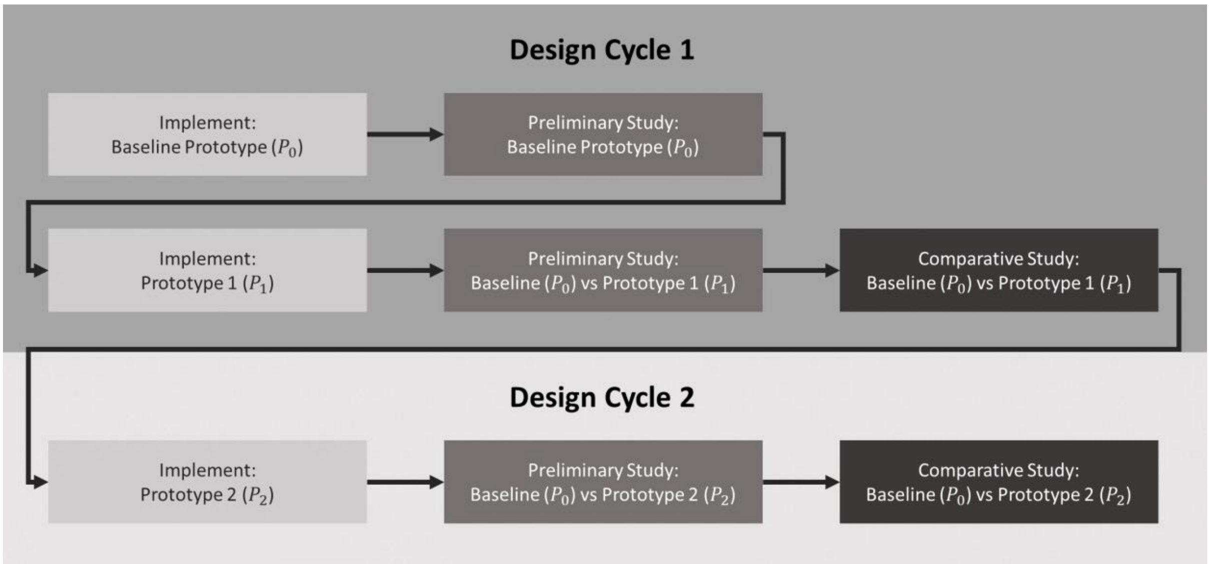

Design Cycle 1
Implement: Baseline Prototype (P_0)
Preliminary Study: Baseline Prototype (P_0)
Implement: Prototype 1 (P_1)
Preliminary Study: Baseline (P_0) vs Prototype 1 (P_1)
Comparative Study: Baseline (P_0) vs Prototype 1 (P_1)
Design Cycle 2
Implement: Prototype 2 (P_2)
Preliminary Study: Baseline (P_0) vs Prototype 2 (P_2)
Comparative Study: Baseline (P_0) vs Prototype 2 (P_2)

Figure 7-1: Overview of Design Cycle 1 and Design Cycle 2

As can be seen from Figure 7-1, Design Cycle 2 has three main phases. The implementation of P_2, followed by a preliminary comparative study between P_0 and P_2, concluding with a comparative study between P_0 and P_2. As with Design Cycle 1, the preliminary study is a small-scale study used to identify any potential issues with the software, as well as providing an opportunity to make minor tweaks to P_2 before doing the more intensive comparative study. Once the researcher was satisfied with the feedback from the preliminary study, the comparative study between P_0 and P_2 was performed. A set of improvement recommendations can then be generated from what was learnt over the course of this design cycle.

7.3 Prototype 2 (P_2)

This section follows the second design cycle, by going over the implementation of P_2, a preliminary study and then a comparative study of P_0 and P_2. It concludes with a set of improvement recommendations generated from the knowledge gained from implementing and evaluating the system.

7.3.1 Implementation

As with P_0 and P_1, the SSF (Chapter 4) was used for the implementation of P_2. The main components of the SSF being: the Pre-processor, Retinal Encoder and Sound Generator. For P_2, the focus was on creating an improved Sound Generator. Thanks to the flexibility of the node system used in the SSF – as well as the standardisation of the topics used for communication – the framework made it simple to focus on creating an improved Sound Generator without worrying about the other components of the system. It also made it simple to test the new Sound Generator against P_0 and P_1 Sound Generators. For P_2, the Pre-processor and Retinal Encoder from P_1 were used unchanged. The implementation of the Sound Generator for P_2 is discussed below.

7.3.1.1 Sound Generator

The Sound Generator for P_2 builds on the concepts used in the P_0 Sound Generator and concepts used in the P_1 Sound Generator. The focus of this implementation was to improve the sound localization accuracy using different techniques. The algorithm works by associating:

- Different complex sounds with vertical position – rather than only pitch as with P_0 and P_1;
- Relative left-right gain (respective loudness on the left and right) with horizontal position; and
- The sound loudness and sound pitch with distance – rather than solely loudness as with P_0 and P_1.

As with the P_1, using these different associations for each dimension (achieving *PDO 3.c*) means that the soundscapes can be generated in real-time (achieving *PDO 3.b*). And although P_0, P_1 and P_2 Sound Generators are very similar, the P_2 Sound Generator aimed to improve on the previous Sound Generators by using multiple complex sounds in an attempt to improve vertical sound localization accuracy; and, using pitch (as well as loudness) for distance, in an attempt to improve distance sound localization accuracy. The way the Sound Generator was implemented is as follows.

Multiple Complex Sounds: Upon receiving the first retinal encoded image, the setup process for the Sound Generator was run. This included loading the complex sounds, then generating a sound source for each of the 50 pixels ($10 \times 5 = 50$ pixels) in the retinal encoded image – as with P_1. Once the sound sources had been created, a complex sound was attached to each sound source. P_1 used a single complex sound and varied its pitch, for the P_2 Sound Generator, it was decided that multiple complex sounds would be used in a further attempt to improve localization accuracy. Since there were 5 rows of pixels, five complex sounds were carefully chosen, one for each row:

- The 1st row was bird tweets;
- The 2nd row was cricket sounds;
- The 3rd row was leaves crunching;
- The 4th row was river flowing;
- The 5th row was waterfall.

Where the 1st row of pixels was the top row. Based on literature (Section 2.2.3.2), it was decided that P_2 would use higher pitched sounds for the higher pixels, as well as using sounds logically associated with height – i.e. birds for above, and water for below. As with P_1, to avoid clipping of the sounds (Section 2.2.4), the default gain for each sound source was set relative to the number of sound sources. As with P_1, the additional measures of loudness, such as left-right loudness and loudness based on distance are dynamically managed by PyAL.

Unit Vector Map: Another part of the setup process was generating the unit vector map; as with P_1, the unit vector map contained a unit vector for each pixel in the retinal encoded image – each unit vector described the perspective projection for that specific pixel. The unit vector map is an efficient way of calculating 3D projection placements for pixels given a specific depth value. The exact same functions used in the P_1 Sound Generator were used to generate the unit vector map – having these functions as a part of SSF Core is what made this possible. For implementation details, see Section 6.6.1.3. Once the sound sources had been setup and the unit vector map had been generated, the setup process was complete; from there, the sound sources could be dynamically and independently positioned, additional adjustments such as varying the pitch could also be done dynamically.

Perspective Projection: As with the P_1 Sound Generator, once the unit vector map had been generated, the projected x, y and z values for a given depth could be efficiently calculated. The exact same projection functions used in P_1 were used in the P_2 Sound Generator – this was possible thanks to SSF Core. For implementation details see Section 6.6.1.3. Once the projected 3D placement had been calculated, the sound source was positioned in virtual 3D space using PyAL – this was done independently for each sound source.

Varying Pitch with Distance: The benefit of using different complex sounds for different heights was that it freed up the ability to use pitch for indicating something other than height. For the P_2 Sound Generator it was decided that pitch (as well as loudness) would be used to indicate depth – with the aim of improving depth accuracy. For each sound source, the pitch was set based on the depth (i.e. the distance) of the related pixel in the retinal encoded image. As discussed in the P_1 Retinal Encoder, $d_{min} = 0.2$ (minimum distance) and $d_{max} = 3$ (maximum distance), these values were fixed. The depth percentage ($d_{percentage}$) is a measure of where in that depth range the given depth value (d) falls (each pixel has its own depth value), i.e. if $d = d_{min}$ then $d = 0\, percent$, where if $d = d_{max}$ then $d = 100\, percent$. The depth percentage was calculated using Equation 14.

$$d_{percentage} = \frac{d - d_{min}}{d_{max} - d_{min}}$$

Equation 14

$$p = p_{max} - ((p_{max} - p_{min}) \times d_{percentage})$$

Equation 15

Once the depth percentage ($d_{percentage}$) had been calculated for a given pixel (i.e. sound source), the pitch (p) for that sound source could be calculated – this was done using Equation 15. Equation 15 calculates the pitch (p) by inversely varying the pitch between a fixed maximum and minimum pitch ($p_{max} = 1.5$ and $p_{min} = 0.3$ respectively), the varying of the pitch is based off the depth percentage. For example, if $d_{percentage} = 0\, percent$ then $p = 1.5$ and if $d_{percentage} = 100\, percent$ then $p = 0.3$. The closer something was, the higher the pitch of the sound source.

Warning Beep: A warning feature was added to this algorithm, which warns the participant if they were about to bump into something (e.g. a wall) on their left, or on their right. The way this worked is that whenever a participant was too close to an object on their left, a beep would go off in their left ear, and whenever a participant was too close to an object on their right, a beep would go off in their right ear. A distance of $0.3m$ was set as the minimum depth; hence, when the participant was closer than $0.3m$ the beeping sound would play on a loop until they moved further than $0.3m$ from the object. The aim of adding the warning beep was to improve the learnability of the system.

7.3.2 Preliminary Study

A preliminary study was performed, comparing P_0 and P_2. After the preliminary study was completed the beeping sounds were discussed with the participant. It was realised that the participant relied heavily on the beeping to tell when they were close to a wall – relying less on the actual variations in sound, i.e. relying less on the actual algorithm, and more on the warning feature (the beeping) added to the system. Since the warning beeps were a very dominant sound, and based on the discussions with the preliminary study participant, the researcher decided that the warning beeps were likely to skew the results; hence, making it difficult to determine whether the change in performance for P_2 was due to the changes – such as using multiple complex sounds and changing pitch with depth – or whether it was due to the warning beeps. For this reason, the warning beeps were disabled for the formal evaluations. In the long term, it is unclear whether having warning beeps – in a system such as the prototypes developed – would increase the learnability of the system, or become a crutch for the user, in essence decreasing the learnability of the actual system. Further than the warning beep removal, P_2 functioned as expected and the comparative study was performed.

7.3.3 Comparative Study

This section looks at the results of the comparative study between P_0 and P_2. As with the comparative study done between P_0 and P_1 (Section 6.6.3), for the comparative study between P_0 and P_2, there were a total of eight participants ($n = 8$). These were not the same participants as from the first comparative study (Section 6.6.3) – as per the section on participant selection (Section 5.2). Each participant completed the comparative study evaluation procedure discussed in Section 5.6.

The Pre-Evaluation Questionnaire (Section 5.7.2) revealed that: All the participants were male; none of the participants had any prior experience using sensory substitution systems; all participants, bar one, were between the ages of 21 and 30, with the one being between the ages of 31 and 40; all the participants had the ability to hear in their left and right ears. The highest hearable minimum frequency in the group was 200Hz, and the lowest maximum hearable frequency was 12000Hz, meaning all participants could hear between 200Hz and 12000Hz.

As was the case in Section 6.6.3, for the results obtained from the study, the paired t-Test for two sample means was commonly used for statistical comparison. The Chi-Squared Goodness of Fit Test was also used where appropriate. For all the statistical tests throughout this section, $\alpha = 0.05$ was used, and the hypotheses were:

$$H_0\text{: } The\ two\ prototypes\ performed\ equally\ well$$
$$H_1\text{: } One\ prototype\ performed\ better\ than\ the\ other$$

The graphs throughout this section use the same graph legend as in Section 6.6.3 (Figure 6-16). This section will discuss the results of the different metrics gathered.

7.3.3.1 Questionnaires

This section summarizes the results from the Post-Evaluation Questionnaires and the Final Questionnaires comparing P_0 to P_2. The results are primarily qualitative data obtained from the questionnaires. The feedback collected for P_0 was sorted and summarized into positive and negative feedback shown in Table 7-1.

Table 7-1: Baseline Prototype (P_0) Qualitative Feedback from the P_0 vs P_2 study ($\mathrm{n} = 8$)

	Positive	Negative
1	Some participants preferred the sound compared to P_2, feeling that the sound from P_0 was less cluttered	Some participants felt it was difficult to detect the distance to obstacles since the changes in sound were too small
2	One participant felt that using P_0 it was easy to detect walls and gaps	One participant felt that compact spaces such as corridors were difficult to navigate since constant noise was produced
3	One participant said it felt easier using P_0 than P_2 to make turns and find the passage	Two participants felt the sound was slightly unbalanced
4	-	Sound anomalies/artefacts
5	-	Low objects and small objects against a plane (table/floor/wall) were hard to detect

Table 7-1 shows that some participants preferred the sound of P_0 compared to P_2, stating that it felt less cluttered. Some participants however found the sound to have too little distinction, making it difficult to detect obstacles since the changes were harder to detect than with P_2. With regards to navigation, one of the participants said that they found it easy to detect walls and gaps using P_0 – although they felt they did equally well in the navigation tasks with P_0 and P_2. Another participant said that it felt easier to find the passage with P_0, however the same participant said that they felt that P_2 gave them a better understanding of the overall environment, compared to P_0. As was the case in the first comparative study (Section 6.6.3), there were sound anomalies since P_0 does not have a Pre-processor that reduces the issue. From Table 7-1 it can also be seen that two participants felt the sound was slightly unbalanced – as with the previous comparative study where one participant found this to be the case (Section 6.6.3.1).

In Table 7-2, from the questionnaires, the qualitative feedback for P_2 is summarized into positive and negative feedback. Looking at Table 7-2, some participants felt that the sounds produced by P_2 were overwhelming, with one participant saying that less environmental sounds should be used. Other participants preferred the sounds from P_2 over the sounds produced by P_0.

Table 7-2: Prototype 2 (P_2) Qualitative Feedback from the P_0 vs P_2 study ($\mathrm{n} = 8$)

	Positive	**Negative**
1	The majority of participants felt that P_2 performed better than P_0 at distinguishing the distance to obstacles	Some participants felt that the composition of the sounds was overwhelming
2	Some participants felt that the different sounds made it easier than P_0 to navigate environments with barriers and open spaces	One participant felt that compact spaces such as corridors were difficult to navigate since constant noise was produced
3	Some participants felt that the different sounds used in P_2 were more distinguishable and had a larger range	Low objects and small objects against a plane (table/floor/wall) were hard to detect
4	The majority of participants found P_2 to be more accurate in determining the quadrant of an object	-
5	Some participants found the sounds to be pleasant and less monotonous than those produced by P_0	-
6	Some participants felt that P_2 was more intuitive than P_0	-

Regarding the sounds produced by P_2, one participant said, “I feel like some people would be annoyed by it, but I enjoyed it”. Sound preferences aside, the majority of participants felt that P_2 was more accurate in a number of areas, such as distance discrepancy and determining the quadrant of an object. One participant said that it might be the variety of sounds that made it easier. Another participant said that even though they preferred sounds from P_0, they felt as though P_2 was more accurate and more intuitive – a number of participants felt P_2 was more intuitive.

For both systems one participant felt that they struggled in compact spaces such as corridors, since both systems produced a large amount of noise in that environment. As was the case in the comparative study between P_0 and P_1 (Section 6.6.3.1), participants found the Box Task and Multiple Boxes Task difficult when using P_0 and P_2 in this study. Participants generally found low objects and small objects close to a plane (such as a table) difficult to detect.

Table 7-3: System preference rating for P_0 vs P_2 normalized to be between -3 and 3 ($n = 8$)

Evaluation	P_0 (-3) vs P_2 (3)
1	3
2	1
3	3
4	1
5	2
6	3
7	2
8	-2
Mean	1.625
Variance	2.839

Table 7-3 shows which system the participant preferred. As can be seen from Table 7-3, 7 of 8 participants preferred P_2 over P_0. To check the statistical significance of these results, a paired t-Test for two sample means was performed on the preference ratings. The resulting $p\text{-}value$ was 0.015. Since $p\text{-}value < \alpha$, H_0 was rejected. This means that with a 95 $percent$ confidence, one can conclude that in general, users preferred P_2 over P_0.

In summary, participants who preferred the sound from P_0 liked that it was not as overwhelming. Participants who preferred the sound from P_2 liked that the sounds were more distinct and less monotonous. In general, people felt that they had a better understanding of their environment with P_2 compared to P_0; with the majority of participants saying they found P_2 more accurate with distance and determining the quadrant an object was in. For navigation, some participants preferred P_0 with others preferring P_2. Participants also found it difficult to distinguish low objects and small objects close to a plane (such as a table), making that a constant issue across P_0, P_1 and P_2.

Finally, from the additional feedback it again became clear – as with the previous comparative study – that participants felt that given more time to use and learn the systems, they would have performed better. Regarding the hardware, as with the previous comparative study, participants said that the hardware could be made smaller and lighter. One participant said that they liked the fact that the earphones used (Section 6.3.1) did not impair sounds from the environment. Another participant made a helpful suggestion, saying that it might be beneficial to modify the system so that the camera is at eye level (rather than above on the helmet).

7.3.3.2 Navigation Task (no obstacles)

Figure 7-2 shows the average number of wall touches (E_{wall}) for P_0 and P_2 while completing the navigation task with no obstacles. P_0 had an average of 1.875 wall touches – similar P_0's 1.625 wall touches seen from Section 6.6.3.2 in the study comparing P_0 and P_1, with P_1 having 1.375 wall touches. Comparatively, P_2 had an average of only 0.625 wall touches for the same task.

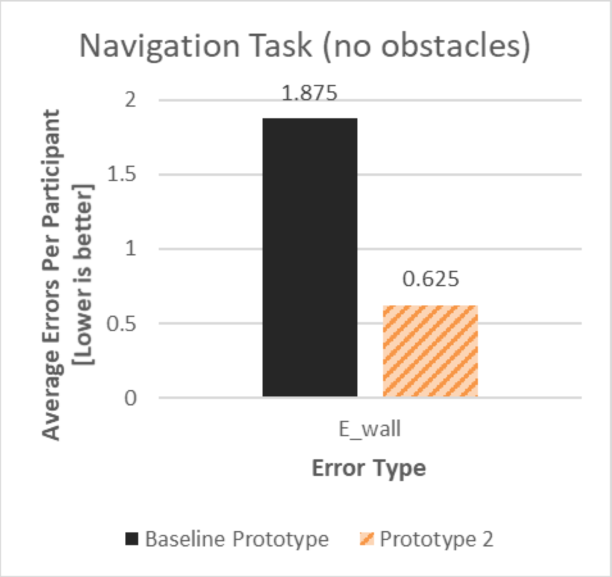


Figure 7-2: Average number of E_{wall} errors made per participant ($n = 8$) in the Navigation Task (no obstacles)

A $p\text{-}value$ of 0.006 was calculated using a paired t-Test for two sample means. Since $p\text{-}value < \alpha$, H_0 was rejected. From the t-Test, it was concluded with a 95 $percent$ confidence that while navigating a straight passage with no obstacles, participants would touch the walls less on average when using P_2 rather than P_0. This aligns with the feedback from the questionnaires, where participants generally felt as though they had a better understanding of their environment with P_2, and that P_2 was more accurate at determining distance.

7.3.3.3 Navigation Task (with obstacles)

Figure 7-3 shows the average number of errors made per participant for the navigation task (with obstacles) when using P_0 and when using P_2. The errors are grouped into the subcategories listed in Section 5.7.1.

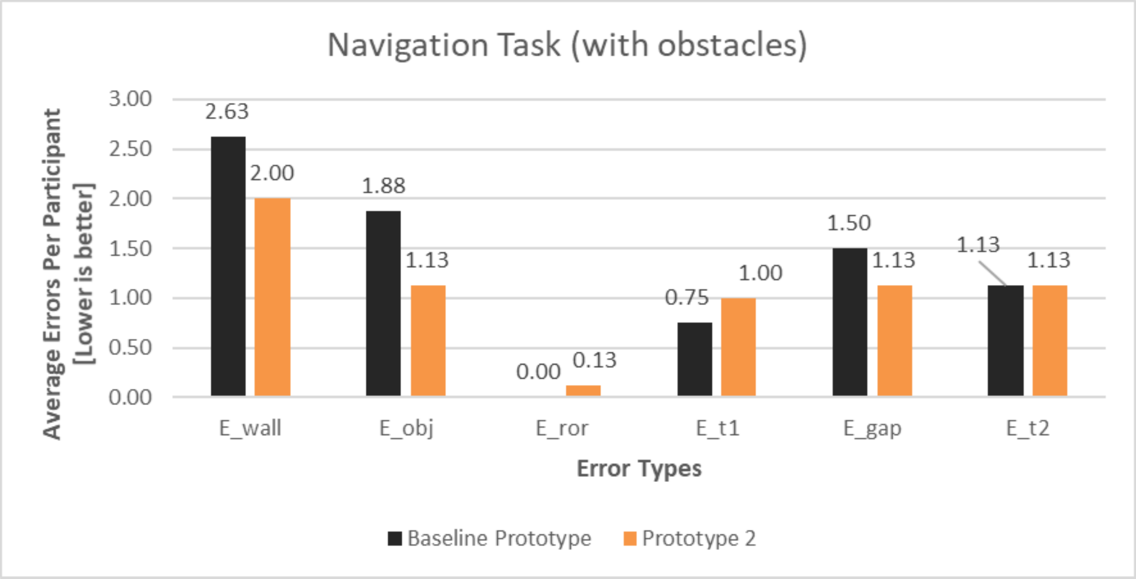


Figure 7-3: Average number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles) – grouped by error type

For each of the error types, a paired t-Test for two sample means was performed. For each of these groups their respective $p\text{-}value$'s were greater than α, which meant that H_0 was not rejected and hence no statistically significant difference was found.

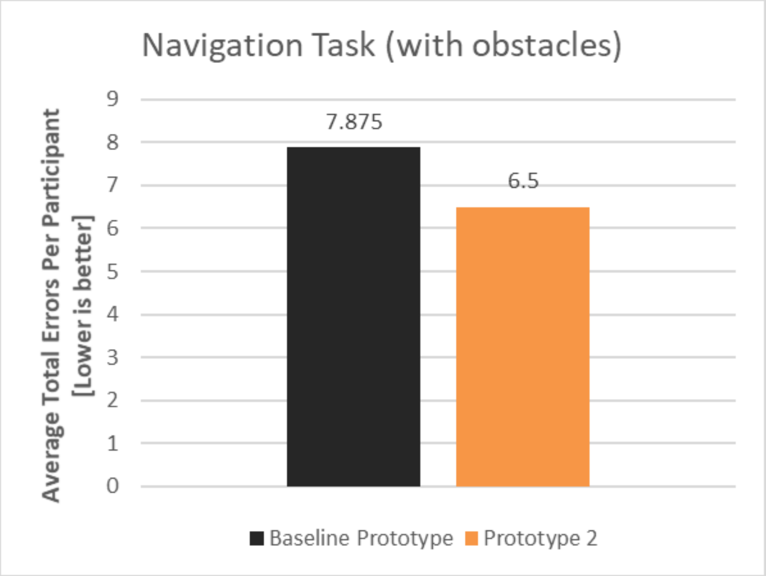


Figure 7-4: Total number of errors made per participant ($n = 8$) in the Navigation Task (with obstacles)

Figure 7-4 shows the total number of errors made per participant on average for the navigation task with no obstacles – comparing P_0 prototype to P_2. A paired t-Test for two sample means was run on the overall error count. A $p\text{-}value$ of 0.06 was calculated, hence for $\alpha = 0.05$, $p\text{-}value > \alpha$ and so H_0 was not rejected. This means that when using each system, no statistically significant difference was found regarding the average number of errors per participant. It should however be noted that if one was willing to use an $\alpha = 0.1$, then $p\text{-}value < \alpha$ and hence H_0 is rejected. With $\alpha = 0.1$, one could conclude with a $90\, percent$ confidence that on average participants make less errors when using P_2 to navigate a complex environment than when using P_0.

7.3.3.4 Quadrant Task

For the quadrant task, as with Section 6.6.3.4, since there were 8 participants, and since each participant was asked to determine the quadrant 5 times, the totals are out of 40. Figure 7-5 shows the total number of times the participants discriminated left from right correctly, and top from bottom correctly.

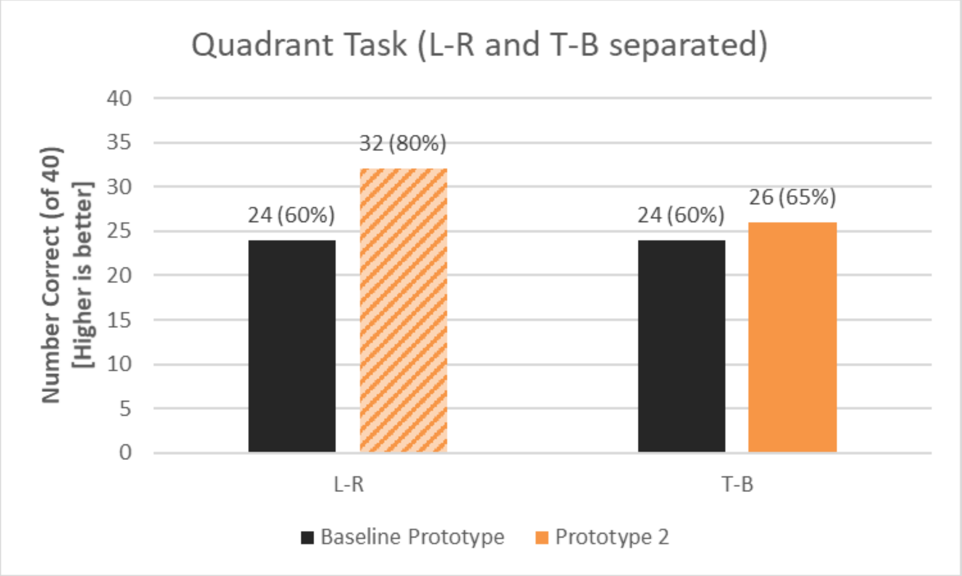


Figure 7-5: Total number of correct answers for the Quadrant Task across all participants ($n = 8$); separated into correct answers for left-right (L-R) and top-bottom (T-B)

In order to determine whether the results were statistically significant, it was decided to generate contingency tables, and perform Chi-Squared Goodness of Fit Tests on the data. This is the same procedure that done in Section 6.6.3.4.

Table 7-4: Contingency table for left-right discrimination

	Correct	Incorrect
Observed Data (P_2)	32	8
Expected Data (Baseline Prototype)	24	16

Table 7-4 is a contingency table for how many times participants got the left-right discrimination correct while using P_0 and P_2. Using the table, a Chi-Squared Goodness of Fit Test performed on this data generated a $p\text{-}value$ of 0.01; since $p\text{-}value < \alpha$, H_0 was rejected. This means that with $95\, percent$ confidence one can say that, P_2 performed better than P_0 at discriminating between left and right for a close isolated object.

Table 7-5: Contingency table for top-bottom discrimination

	Correct	Incorrect
Observed Data (P_2)	26	14
Expected Data (Baseline Prototype)	24	16

Table 7-5 is a contingency table for how many times participants got the top-bottom discrimination correct while using P_0 and P_2. Using the table, a Chi-Squared Goodness of Fit Test generated a $p\text{-}value$ of 0.517; since $p\text{-}value > \alpha$, H_0 was not rejected. This means that no statistically significant difference was found between P_0 and P_2 when discriminating whether an object is in the top or bottom quadrant.

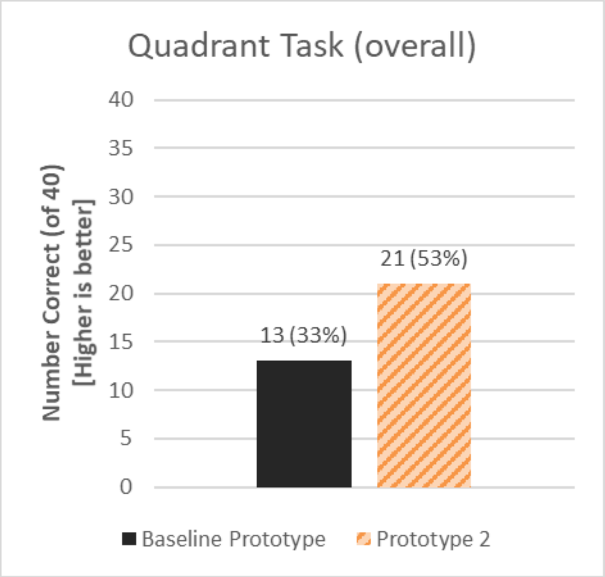


Figure 7-6: Total number of correct answers for the Quadrant Task across all participants ($n = 8$)

For each prototype, P_0 and P_2, Figure 7-6 shows the number of times the participants got the quadrant correct – meaning both whether the object is on the left or right, and whether the object is at the top or the bottom, getting either incorrect means the answer was incorrect. Using this data, a Chi-Squared Goodness of Fit Test was performed to determine whether participants performed better when using P_0 or P_2 when determining the quadrant an object was in. A $p\text{-}value$ of 0.007 was calculated, and hence $p\text{-}value < \alpha$, this meant that H_0 was rejected. With this result, one can conclude with 95 $percent$ confidence that P_2 performs better than P_0 at determining the quadrant of an object.

Looking at the individual results, an interesting observation was made. When using P_2 for the quadrant task, and choosing between top or bottom, two of the eight participants chose the opposite of where the object was 100 $percent$ of the time – i.e. choosing top when it was at the bottom. The consistency with which these two participants were incorrect stood out quite prominently to the researcher, both observationally during the study (where the researcher actually noted the consistency, promptness and confidence with which the two participants answered incorrectly), and analytically looking at the resulting data. It was realized that this may be a result of the sounds chosen to represent high and low objects. In that case, what would happen for these two participants, is that whenever they would hear a sound that was intended to be associated with the top, they would associate it with the bottom, and vice versa. It was decided to correct for this anomaly and investigate the results. To correct the anomaly, it was decided to take the average number of correct top-bottom answers from the other six participants when using P_2. This resulted in assigning $\frac{4}{5}$ to the two participants for the top-bottom part of the quadrant test when using P_2. Since these two participants had originally answered all the questions incorrectly ($\frac{0}{5}$), one could have also assigned $\frac{5}{5}$ as this would be the opposite. However, it was decided that $\frac{4}{5}$ would be used. Below are some graphs of the corrected data, together with a discussion.

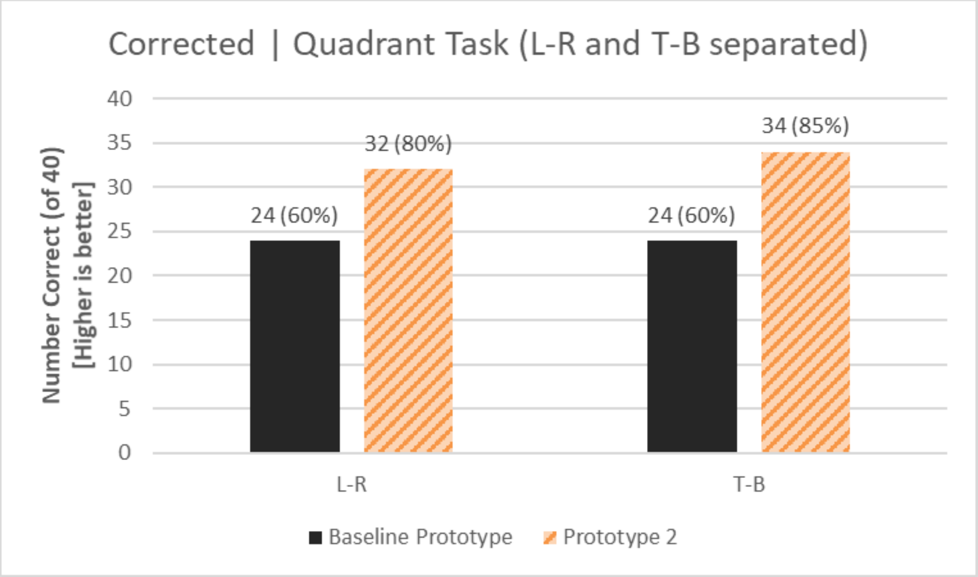


Figure 7-7: The corrected graph of the total number of correct answers for the Quadrant Task across all participants ($n = 8$); separated into correct answers for left-right (L-R) and top-bottom (T-B)

Figure 7-7 is the corrected version of Figure 7-5. Since it is only the top-bottom results for P_2 that were corrected, top-bottom results for P_0 and the left-right results for P_0 and P_2 remain exactly the same between Figure 7-7 and Figure 7-5. For the updated results, the number correct for the top-bottom discrimination changed from 26 to 34 for P_2. Table 7-6 is the updated contingency table, with the original being Table 7-5.

Table 7-6: Corrected contingency table for top-bottom discrimination

	Correct	Incorrect
Observed Data (P_2)	34	6
Expected Data (Baseline Prototype)	24	16

Performing the same Chi-Squared Goodness of Fit Test on the corrected data in Table 7-6, a $p\text{-}value$ of 0.001 was generated; since $p\text{-}value < \alpha$, H_0 was rejected. This means that assuming the correction made was valid, it can be concluded with a 95 $percent$ confidence that P_2 performs better than P_0 at discriminating between the top and bottom quadrants for nearby objects.

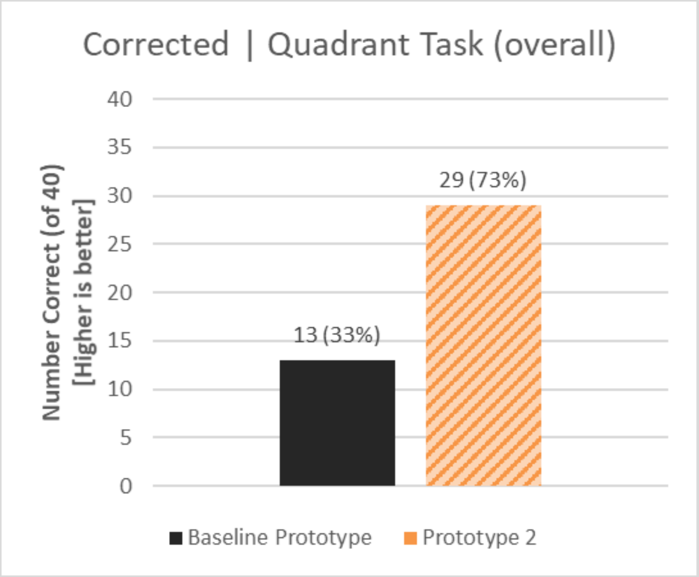


Figure 7-8: The corrected graph of the total number of correct answers for the Quadrant Task across all participants ($n = 8$)

Figure 7-8 shows the corrected totals (based on the corrections made above). The totals for the P_0 remain the same as in Figure 7-6, however, the totals for P_2 changed from 21 to 29 due to the corrections. Performing a Chi-Squared Goodness of Fit Test on the corrected data in Figure 7-8, a $p\text{-}value$ of 6.62×10^{-8} was calculated. Since $p\text{-}value < \alpha$, H_0 was rejected. This is the same result as seen for Figure 7-6 – one can conclude with 95 $percent$ confidence that P_2 performs better than P_0 at determining the quadrant of an object.

From the corrected data, one can see the same trend with P_2 as was seen with P_1; where P_2 significantly outperformed P_0. It is also interesting to note that P_1 and P_2 performed similarly on this task with P_1 having an overall of 70 $percent$ correct and P_2 having an overall of 73 $percent$ correct.

In conclusion, it appears that whether the results are corrected or not, overall P_2 outperforms P_0 at determining the quadrant in which a nearby object is. It is also possible to conclude that for a minority of people, the complex sounds chosen to represent higher and lower objects (Section 7.3.1.1) in P_2 had the opposite effect to what was intended. The researcher assumes that it is highly likely that with more training these participants would come to understand top from bottom when using P_2. It is also reasonable to assume that for both P_0 and P_2, with more training the results would further improve. Finally, assuming the corrections made to the top-bottom P_2 results were valid, it can be concluded that P_1 and P_2 performed equally well on the quadrant task – both outperforming P_0. Based on the questionnaire feedback from Section 6.6.3.4 and Section 7.3.3.4, this result is consistent with the general participants view on how well they did on the Quadrant Task when using P_1 compared to P_0, and P_2 compared to P_0. It is left for the reader to decide for themselves whether or not the researcher made the correct assumption with how the given data should be handled – based on the qualitative and quantitative data gathered.

7.3.3.5 Box Task

Figure 7-9 shows the total number of correct answers – grouped by metric – for the box task. It compares the results of P_0 and P_2.

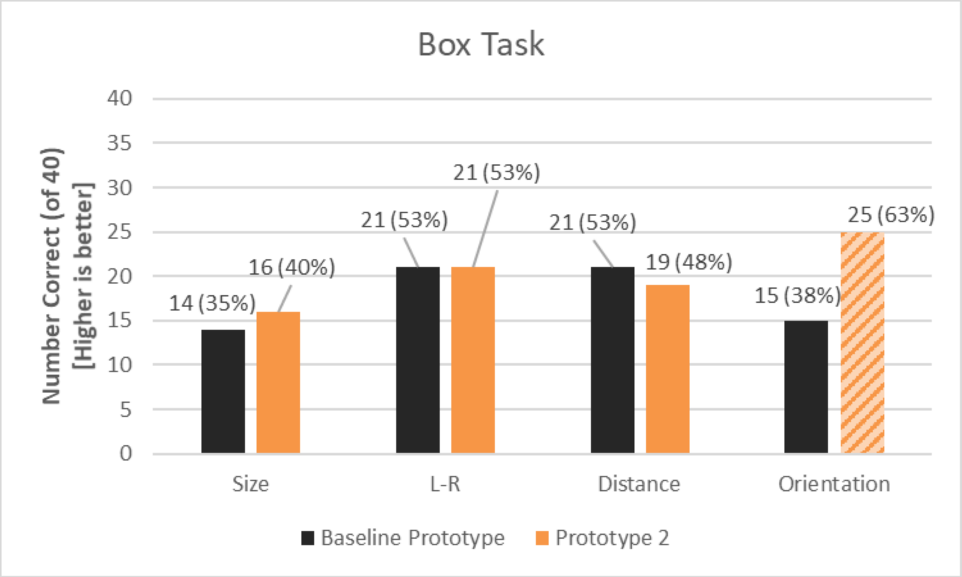


Figure 7-9: Total number of correct answers across all participants ($n = 8$) for the Box Task – grouped by metric

As was done in Section 6.6.3.5, a Chi-Squared Goodness of Fit Test was independently performed for the size, L-R placement, distance and orientation. For all metrics except the orientation metric, no statistically significant difference was found between P_0 and P_2. For the orientation, a $p\text{-}value$ of 0.001 was calculated, hence $p\text{-}value < \alpha$ and so H_0 was rejected. This means that P_2 performed statistically significantly better than P_0 when identifying the orientation of a box.

It should however be noted that, specifically comparing the distance and orientation results in Section 6.6.3.5 to the distance and orientation results seen here (Figure 7-9), there is a large amount of variance in the results for P_0. Using the same box task, with new participants, under the same conditions and using the same baseline prototype, for the first comparative study the participants got the distance correct 28 $percent$ of the time, where in the second the participants got the distance correct 53 $percent$ of the time. Similarly, a large discrepancy was seen for the box orientation – for the first comparative study the participants got the orientation correct

$60\ percent$ of the time, where in the second the participants got the distance correct $38\ percent$ of the time. The only difference between the tests was the prototype P_0 was compared against – however the order in which the systems were used was randomized for both studies, hence this should not have had an effect on the results. This could indicate that a larger sample size is needed for this test to yield clear results.

It is also possible that for the box task, none of the current iterations of the system are able to confidently locate an object which is randomly sized, orientated and positioned on a table. This is somewhat unsurprising since this is a complex problem. Additionally, looking at the questionnaire feedback in Section 6.6.3.4 and Section 7.3.3.4, it seems that the participants would agree that the current systems are not capable of reliably distinguishing objects on tables and in similar situations. If that is the case, with a larger sample size, one would expect the results to tend to the equivalent of random guessing. That being $33.3\ percent$ for size, $50\ percent$ for L-R, $33.3\ percent$ for distance and $50\ percent$ for orientation. Looking at the results in Figure 6-22 and Figure 7-9, this seems feasible.

The reason for stating this is to caution the reader regarding the results of the box task, since, as indicated, a larger sample size may be needed for this task.

7.3.3.6 Multiple Boxes Task

Figure 7-10 shows the results of the multiple box task when comparing baseline prototype and P_2. As explained in Section 5.4, for each different region – left, middle and right – the participant had to identify whether or not a medium size box was present. For each metric – left, middle and right – a Chi-Squared Goodness of Fit Test was performed. No statistically significant differences were found between P_0 and P_2.

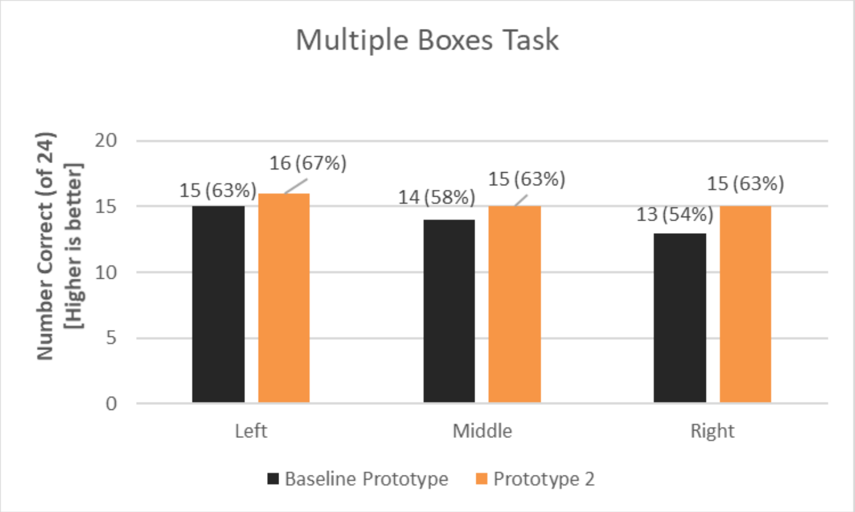


Figure 7-10: Total number of correct answers across all participants ($n = 8$) for the Multiple Box Task – grouped by metric

Comparing the results from the study between P_0 and P_1 (Section 6.6.3.6 and Figure 6-23) to the results from the study between P_0 and P_2, the baseline results seem to be similar. Overall, comparing the results from P_1 to P_2, P_2 appears to be more consistent across the left, middle and right. For the most part there is no statistically significant difference between the P_1 and P_2, with the exception that the P_2 now performs on par with P_0 for determining whether or not there is an object placed in the middle. Finally, for distinguishing whether there was a box on the left and on the right in the Multiple Box Task, P_2 did not do as well as it did in the Quadrant Task. However, as mentioned in Section 6.6.4, this was expected, since it was not the focus for improvement on this iteration. It should also be mentioned that as with the Box Task, participants generally felt that it was difficult to reliably distinguish objects on a table, at times mentioning they felt as though they were guessing.

7.3.4 Recommendations

In Section 6.6.4 of the previous chapter it was recommended that one make use of multiple complex sounds, one for each row of pixels in the retinal encoded image. The aim being that this would improve the vertical sound localization accuracy. Practically this would mean an improved ability to distinguish an object from the table on which it is placed. From the qualitative (Section 7.3.3.1) and quantitative results of the Box Task (Section 7.3.3.5) and Multiple Boxes Task (Section 7.3.3.6), it does not seem as though the complex sounds greatly improved the participants' ability to detect low objects – if at all. This could be due to a number of factors, such as the need for more training time on these finer grained detection challenges, or it could even be that the systems simply are not able to detect these changes due to the resolution of the retinal encoded image being too low.

An interesting comment was made by one participant, stating that to improve the system, one could put the camera at eye level (Section 7.3.3.1). This would improve the line of sight since rather than the line of sight going out from above the participants head, it would go out from their eye level – which is of course a more natural place for the cameras line of sight to be. The closer an object is, the worse an offset line of sight (i.e. camera mounted on top of the helmet) would affect the localization. For example, if one's actual line of sight is directed at object close to them, then the offset camera's line of sight might be looking just over the object. Since detecting an object on a table is generally something that would be done when a participant is relatively close to the object, this error of parallax would likely be relatively significant. Correcting this, by putting the camera at eye level could potentially improve participants' performance on the box tasks with P_0, P_1 and P_2.

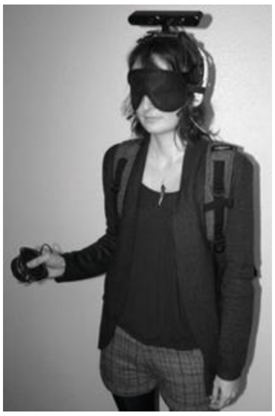


Figure 7-11: A participant wearing the MeloSee hardware (Stoll et al., 2015)

Often SSD designs that one would come across – including MeloSee as seen in Figure 7-11 – placed the camera atop a helmet on one's head, or on ones forehead in some or other way (Danilov et al., 2006; Hoffmann, Spagnol, Kristjánsson, & Unnthorsson, 2018; Stoll et al., 2015). From this, it seemed clear that the camera should be mounted on top of the helmet as seen in Figure 6-2 (b). Details such as placing the camera at eye level are obvious in retrospect, but initially, basing the hardware design on a number of other systems, this was not something that was considered. Future researchers should be encouraged to place the camera at eye level, where possible.

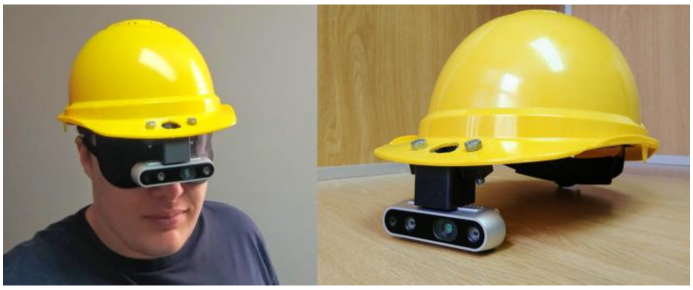


Figure 7-12: Camera mounted at eye level

In Section 6.6.4 of the previous chapter it was also recommended that one use a varying pitch with distance – which was implemented in P_2. From the participants feedback (Section 7.3.3.1) in addition to the results of things such as the Navigation Task (Section 7.3.3.2), it can be seen that the distance discrimination for P_2 improved on P_0.

Based on these results, it is recommended that some parts such as the varying pitch with distance are kept, while other parts change. Below are the recommendations for future prototype implementations, starting with a hardware recommendation:

- ***Eye level Camera:*** As per the feedback given by the participant, and the discussion above, the camera should be put at eye level – rather than on top of the helmet. This can be done by mounting the camera upside-down (Figure 7-12), then in software, one can use OpenCV to flip the image to be orientated correctly. Mounting the camera at eye level will align the camera’s line of sight with one’s normal line of sight. This will likely improve one’s ability to locate objects, especially nearby objects, since one does not need to account for an offset line of sight. This will mean that no mental transposing of the object’s location needs to be done. Additionally, this should improve learnability of the system, since the line of sight is more natural.

The next recommendations are for the Sound Generator:

- ***HRTFs:*** To further improve sound localization accuracy, Head-Related Transfer Functions (HRTFs) could be used. As discussed in Section 2.2.3.4, HRTFs are functions, which change the sound based on a number of factors such as how sounds interact with the shape of one’s pinna. The transformations applied by HRTFs provide spectral cues – these cues are, especially helpful in vertical sound localization (Section 2.2.3.2). Applying HRTFs could be challenging, since different people have unique HRTFs. It is, however, known from literature, that humans have the ability to adapt to deformed pinna (Section 2.2.3.2). This indicates that it is highly likely that given sufficient training time, participants would be able to adapt to generic HRTFs. The aim of using HRTFs would be to improve the sound localization accuracy, in addition to improving the learnability of the system. To improve the learnability of the system, a fairly universal set of HRTFs would need to be used.
- ***Unique Complex Sounds:*** The reason for using complex sounds is that humans are better at localizing complex sounds (Section 2.2.3.2). Additionally, without the complex sounds, the HRTFs would not be able to generate the spectral cues since there would only be one frequency. The combination of spectral cues from the HRTF and using a unique complex sound per receptive field (i.e. per pixel) – rather than just one per row – would likely allow for greater localization accuracy both in the vertical and horizontal directions. It is also possible, however, that using a unique complex sound per receptive field would be overwhelming. To get around this, one could possibly mirror the left and right sounds. Using the unique complex sounds per receptive field could then be compared to the use of a single complex sound per row in the retinal encoded image as implemented in P_2.

The recommendations given aim to improve the accuracy of localizing low objects and small objects against a plane, in addition to improving the learnability of the system by continuing to build on the use of how humans localize sounds.

7.4 Conclusion

This chapter looked at the implementation of P_2 and then the results from the study comparing P_0 and P_2. From the results it was concluded that participants preferred P_2 over P_0 (Table 7-3). The majority of participants felt that they had a better understanding of their environment when using P_2 compared to P_0. As with P_1, participants felt that P_2 was more reliable than P_0 for object detection (this achieves *PDO 3.e*). This was especially true of the Quadrant Task, where based on the corrected results, participants got the exact quadrant correct $73\ percent$ of the time when using P_2, whereas with P_0 they only got it correct $33\ percent$ of the time (Section 7.3.3.4).

For distance discrimination participants also felt that they were more accurate when using P_2. Looking at the results in Section 7.3.3.2 from the Navigation Task (no obstacles), P_2 improved on P_0 and P_1 – indicating that participants were able to tell the distance to walls more accurately. This improvement was likely due to a combination of varying the pitch with distance, in addition to using a wider variety of complex sounds (Section 6.6.4). It should also be mentioned that when comparing P_0 and P_2, some participants felt that the different sounds from P_2 made it easier to navigate environments with barriers, and where open spaces needed to be found, some participants also found that P_2 was more intuitive than P_0 – i.e. improved learnability. However, with the more complex navigation task (Section 6.6.3.3), no significant difference was found between P_0 and P_2, as was the case with the study comparing P_0 and P_1 (Section 6.6.3.3). That is, with the exception of P_1 outperforming P_0 and P_2 when it comes to finding narrow passages (Figure 6-18 and Figure 7-3).

For all three of the systems (P_0, P_1 and P_2) it is clear that participants found it difficult to detect low objects and small objects close to a plane – as with the comparative study between P_0 and P_1; this made participants feel as though they were guessing during the box tasks. As mentioned at the end of Section 6.6.4, this was expected. Due to the qualitative and quantitative results for the box tasks, the researcher felt that no conclusions should be made in this section regarding the results of the Box Task and the Multiple Boxes Task. It also became clear that for all the different tasks, participants felt that their performance would have improved if they were given more time to become accustomed to the systems.

Chapter 8: Conclusion

8.1 Introduction

This chapter forms part of the DSR Rigor Cycle, by reflecting on the research completed and its contributions. The chapter begins by looking at the contributions made throughout the research, focusing on the techniques identified for improving the visual-to-auditory sensory substitution; techniques that would give the visually impaired an improved understanding of their surroundings – as per the main research question (Section 1.4). The chapter then looks at the research achievements by reviewing the various chapters and how they addressed the research questions in Section 1.4. It then goes on to discuss the challenges and limitations of the research. The chapter concludes with a summary.

8.2 Research Contributions

Based on the background research (Chapter 2) and a review of several existing systems (Section 3.4), three visual-to-auditory sensory substitution prototypes were created over the course of two Design Cycles (Chapter 6 and Chapter 7). The prototypes developed were P_0 (Section 6.5), P_1 (Section 6.6) and P_2 (Section 7.3), each prototype building on what was learnt from the previous prototype. From each implementation, a number of techniques were identified for improving the visual-to-auditory sensory substitution prototypes, and presented as recommendations (Section 6.5.3, 6.6.4 and 7.3.4). Each set of recommendations was implemented and tested in the form of another prototype – all with the aim of providing the visually impaired with an accurate understanding of their surroundings through audition. The main techniques, which proved useful in the studies (Section 6.5.2, 6.6.3 and 7.3.3) are summarised below:

- ***Techniques for improving depth accuracy:*** The researcher found that quantization of the depth values into discrete distances (Section 6.6.1.2) provided a more distinct and identifiable change in distance, which users preferred (Section 6.6.3.1). It was also found that varying the pitch with distance (Section 7.3.1.1) provided more accurate depth perception in certain instances (Section 7.3.3.2), compared to not associating pitch with distance as with P_0. Using perspective projection (Section 6.6.1.3 and 7.3.1.1) also contributed to the fact that participants felt P_1 and P_2 had better depth accuracy than P_0 (which did not use perspective projection).
- ***Techniques for improving quadrant discrimination accuracy:*** Both P_1 and P_2 successfully used perspective projection to improve localization accuracy – especially regarding the left-right separation of sounds. The use of various complex sounds was also successful in improving

localization accuracy – specifically with regard to the elevation of an object. From the Quadrant Task results (Section 6.6.3.4 and 7.3.3.4), it can be seen that using these techniques greatly improved the quadrant discrimination accuracy when compared to P_0.

- ***Techniques for cleaning up and smoothing out the depth image:*** For visual-to-auditory sensory substitution systems using depth cameras, it is important to ensure the depth data is clean. It is recommended that one identify and crop out any "dead zones" in the depth images generated by the camera. It is also recommended that one use a temporal filter (Section 6.6.1.1) to smooth out the inconsistencies in the depth data provided as output by many depth cameras, hence reducing the image noise.

The recommendations focused on improving the previous prototypes, hence none of the baseline prototypes techniques were mentioned above. Further than the techniques mentioned above, when developing a visual-to-auditory sensory substitution system, it is recommended that one use an accurate lightweight depth camera with a wide field of view. One should also aim to mount the camera at eye level (Section 7.3.4) to provide an accurate line of sight, rather than an offset line of sight.

Additionally, it is recommended that one use a framework such as the Sensory Substitution Framework (Chapter 4), which represents another contribution. Using such a framework allows for faster iteration and testing of prototypes, since it allows the researcher to focus on the algorithm being implemented rather than things such as how to get data from the depth camera. It also breaks the system into reusable components such as the Pre-processor, Retinal Encoder and Sound Generator, which helps when it comes to testing, iterating and improving the system since the parts are modular and interchangeable (Section 4.5). Finally, from the recommendations in Section 7.3.4, it is believed that using HRTFs (Section 2.2.3.4) would also improve on the sound localization accuracy and the learnability of the system.

8.3 Research Achievements

The Design Science Research methodology was followed throughout the research process (Section 1.7). First, the DSR Relevance Cycle was completed, this was done in Chapter 1 to ensure the relevance of the research. From this cycle, the aim of the research was refined and presented in Section 1.3, namely "*To investigate and develop visual-to-auditory sensory substitution techniques – using sound localization as a sensory substitution for depth perception*". In order to achieve this aim, a set of research questions were formulated (Section 1.4).

The research questions are shown below:

RQ 1. What characteristics of audition (hearing) and sound localization can be used for visual-to-auditory sensory substitution?

RQ 2. What are the benefits and shortcomings of existing visual-to-auditory sensory substitution techniques?

RQ 3. How can visual-to-auditory sensory substitution prototypes be developed to allow for the testing of different visual-to-auditory sensory substitution techniques?

RQ 4. How can visual-to-auditory sensory substitution prototypes be evaluated to provide insight into the effectiveness of the different visual-to-auditory sensory substitution techniques?

RQ 5. What visual-to-auditory sensory substitution techniques can be used to develop a visual-to-auditory sensory substitution prototype for depth perception?

RQ 6. Does the prototype developed provide the visually impaired with an accurate understanding of their surroundings through audition?

Each of these research questions were addressed throughout various chapters (Figure 1-3). This section looks at how these research questions were addressed, and whether they were addressed successfully.

Chapter 2 reviewed the literature on audition and sound localization by looking at the field of psychoacoustics. The chapter reviewed the literature on topics such as the anatomy of the human ear (Section 2.2.1), the audible range of humans (Section 2.2.2) and sound localization (Section 2.2.3). From this a number of useful characteristics of audition and sound localization were discovered. These include the interaural time difference (ITD) and the interaural level difference (ILD) which are used for azimuth (horizontal) localization; the fact that the elevation of a complex sound is easier to identify than the elevation of a tone, due to spectral cues; and, that the primary cue for determining the distance of an unfamiliar sound is loudness. In identifying these characteristics, **RQ 1** was successfully answered by Chapter 2.

Using the background research from Chapter 2, Chapter 3 discussed sensory substitution – focusing on visual-to-auditory sensory substitution. The chapter looks at a number of visual-to-auditory sensory substitution devices (Section 3.4), discussing the techniques used by these devices. It then discussed the benefits and shortcomings of existing visual-to-auditory sensory substitution techniques in Section 3.4.5, with a summary in Table 3-2. It was identified that certain techniques allowed for the use of real-time feedback by using binaural audio to provide three-dimensional sound localization information at the same time. It was also identified that none of the systems tested the use of complex sounds for improving localization – although one SSD used complex sounds (musical instruments) to represent colour. Chapter 3 successfully answered **RQ 2**.

Chapter 4 then discussed the Sensory Substitution Framework (SSF). It was noted in Section 4.5 that the SSF was successfully used to implement, test and evaluate all three prototypes. It provides a standardised structure for developing and testing visual-to-auditory sensory substitution techniques and algorithms. Hence, Chapter 4 answers **RQ 3**.

Chapter 5 details the evaluation design. It discusses the various evaluation tasks (Section 5.4), in addition to the evaluation procedure followed (Section 5.6). It also covered the evaluation metrics gathered for the tasks, together with the types of conclusions that could be drawn from the metrics (Section 5.7.1). Chapter 5 answered **RQ 4**.

Chapter 6 and Chapter 7 covered the Design Cycles in the DSR methodology. The Design Cycles followed an iterative process of implementation followed by evaluation. The evaluations resulted in a set of recommended techniques for improving the visual-to-auditory sensory substitution prototypes. With the intention that each set of recommendations can be used in the next implementation (Figure 7-1). The first prototype implemented was P_0, which was the implementation of a chosen existing system called MeloSee (Section 3.4.3). P_0 was used as a baseline for comparison. Based on what was learnt from P_0, a set of recommendations were generated (Section 6.5.3), P_1 was implemented based on these recommendations (Section 6.6.1). A comparative study was then performed between P_0 and P_1, generating a new set of recommendations (Section 6.6.4). P_2 was then implemented based on the recommendations generated from the comparative study between P_0 and P_1. A comparative study was then performed between P_0 and P_2, generating another set of recommendations (Section 7.3.4). From the Design Cycle process followed in Chapter 6 and Chapter 7, a number of visual-to-auditory sensory substitution techniques were generated – as shown in Section 8.2. Hence, **RQ 5** was successfully answered. It was also concluded that in a number of ways, P_0, P_1 and P_2 provided an accurate understanding of one's surroundings. With P_1 and P_2 improving on P_0 in many regards (Section 6.7 and 7.4) – this is especially true of the Quadrant Tasks, where P_1 and P_2 both achieved an accuracy of around $70\ percent$. In other regards such as with small objects against a table, all three prototypes performed poorly. So although the prototypes provided a fairly accurate understanding of one's surroundings, there is still room for improvement. Hence, **RQ 6** was successfully answered.

This means that over the course of a number of chapters, the research successfully addressed all of the research questions posed in Section 1.4. However, throughout the research there were challenges and limitations, which were discussed in the next section.

8.4 Challenges and Limitations

Over the course of the completing this research, a number of challenges and limitations were encountered. One of the challenges was to ensure the various techniques (Section 8.2) used were implemented in a performant manner. This is because many of the functions and algorithms used needed be applied in real-time, going from a raw depth image to sound in a few milliseconds. This included needing to quantize the incoming depth image and temporally filter it in real-time.

A limitation of the study was the performance of the prototypes with regard to low objects and small objects against a plane such as a table. For example, as was mentioned by the participants (Section 6.6.3.1 and 7.3.3.1), the Box Task and Multiple Boxes Task proved difficult. This is likely due a combination of the short training period used, in addition to the low resolution of the prototypes – meaning they did not have the granularity to distinguish objects in these scenarios. It is recommended that future researchers experiment with longer training periods, in addition to investigating ways of increasing the resolution used.

A final limitation of the study was the small sample size of participants used per study ($n = 8$). Due to the small sample size, for certain results it was difficult to determine whether the general trend of marginally improved results was down to chance, or that given a larger sample size the trend would be seen to be an actual improvement. With a larger sample size, it is likely that more trends would become clear, giving the researcher an improved understanding of the strengths and weaknesses of the various visual-to-auditory sensory substitution techniques. It is for this reason that it is recommended that future researchers use a larger participant sample size.

8.5 Summary

The Design Science Research methodology was successfully followed throughout the research (Section 1.7). This resulted in three artefacts being developed, namely the three visual-to-auditory sensory substitution prototypes. The first was the researcher's implementation of an existing system (Section 6.5); the other two being novel implementations based on learnings throughout the research (Section 6.6 and 7.3). Each prototype used sound localization as a sensory substitution for depth perception. The research performed provided a number of contributions, including several techniques for visual-to-auditory sensory substitution (Section 8.2), and a framework for implementing and testing sensory substitution algorithms (Chapter 4). Despite challenges and limitations, the research questions were successfully addressed over the course of the various chapters (Section 8.3). In addition, over the course of these chapters, the main research aim, "To investigate and develop visual-to-auditory sensory substitution techniques – using sound localization as a sensory substitution for depth perception." was achieved.

References

20Hz to 20kHz - Human Audio Spectrum. (2012). Retrieved November 28, 2018, from https://www.youtube.com/watch?v=qNf9nzvnd1k

Abboud, S., Hanassy, S., Levy-Tzedek, S., Maidenbaum, S., & Amedi, A. (2014). EyeMusic: Introducing a "visual" colorful experience for the blind using auditory sensory substitution. *Restorative Neurology and Neuroscience*, *32*(2), 247–257. https://doi.org/10.3233/RNN-130338

American Foundation for the Blind. (2017). *Getting Around*. Retrieved from http://www.afb.org/ProdBrowseTaskResults.aspx?TaskID=274%7B&%7DSpecID=26

Beckman, R. A. (2014). *Interview on BrainPort (3) - YouTube*. Retrieved from https://www.youtube.com/watch?v=JAruu9xp8-A

Begault, D. R. (2000). *3-D Sound for Virtual Reality and Multimedia*. National Aeronautics and Space Administration.

Bermejo, F., Di Paolo, E., Hüg, M. X., & Arias, C. (2015). Sensorimotor strategies for recognizing geometrical shapes: A comparative study with different sensory substitution devices. *Frontiers in Psychology*, *6*(MAY). https://doi.org/10.3389/fpsyg.2015.00679

Bosun, X., Xiaoli, Z., Rao, D. &, & Liang, Z. (2007). *Head-related transfer function database and its analyses*. *50*(3), 267–280. https://doi.org/10.1007/s11433-007-0018-x

Boynton, G. (2008). *Sound Localization (Ch 12)*. The University of Washington.

Brown, D., Macpherson, T., & Ward, J. (2011). Seeing with Sound? Exploring Different Characteristics of a Visual-to-Auditory Sensory Substitution Device. *Perception*, *40*(9), 1120–1135. https://doi.org/10.1068/p6952

Cancar, L., Diaz, A., Barrientos, A., Travieso, D., & Jacobs, D. M. (2013). Tactile-Sight: A sensory substitution device based on distance-related vibrotactile flow regular paper. *International Journal of Advanced Robotic Systems*, *10*. https://doi.org/10.5772/56235

Capelle, C., Trullemans, C., Arno, P., & Veraart, C. (1998). A real-time experimental prototype for enhancement of vision rehabilitation using auditory substitution. *IEEE Transactions on Biomedical Engineering*, *45*(10), 1279–1293. https://doi.org/10.1109/10.720206

Cheng, C. I., & Wakefield, G. H. (2001). Introduction to Head-Related Transfer Functions (HRTFs): Representations of HRTFs in Time, Frequency, and Space. *Journal of the Audio Engineering Society*, *49*(4), 231–249. Retrieved from http://www.aes.org/e-lib/browse.cfm?elib=10196

Coates, A., Lee, H., & Ng, A. Y. (2011). *An Analysis of Single-Layer Networks in Unsupervised Feature Learning* (Vol. 15). Retrieved from http://proceedings.mlr.press/v15/coates11a/coates11a.pdf

Dakopoulos, D., & Bourbakis, N. G. (2010). Wearable Obstacle Avoidance Electronic Travel Aids for

Blind: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *40*(1), 25–35. https://doi.org/10.1109/TSMCC.2009.2021255

Dandona, L., & Dandona, R. (2006). Revision of visual impairment definitions in the International Statistical Classification of Diseases. *BMC Medicine*, *4*, 7. https://doi.org/10.1186/1741-7015-4-7

Danilov, Y., Tyler, M., & Danilov, Y. ; (2006). BrainPort: an alternative input to the brain. *ARTICLE in JOURNAL OF INTEGRATIVE NEUROSCIENCE*. https://doi.org/10.1142/S0219635205000914

Elli, G. V, Benetti, S., & Collignon, O. (2014). Is There a Future for Sensory Substitution Outside Academic Laboratories? *Multisensory Research*, *27*(5–6), 271–291. https://doi.org/10.1163/22134808-00002460

Fristot, V., Boucheteil, J., Granjon, L., Pellerin, D., & Alleysson, D. (2012). *Depth - Melody substitution*. 1–5. Retrieved from https://hal.archives-ouvertes.fr/hal-00731099

Ganguly., D. (2016). DEVELOPING AN ECONOMIC SYSTEM THAT CAN GIVE A BLIND PERSON BASIC SPATIAL AWARENESS AND OBJECT IDENTIFICATION. *International Journal of Advanced Research*, *4*(11), 2003–2008. https://doi.org/10.21474/IJAR01/2304

GitHub. (2018). The State of the Octoverse. Retrieved November 19, 2018, from https://octoverse.github.com/projects#languages

Goldstein, E. B. (2009). *Sensation and Perception 8th Edition*. Wadsworth, Cengage Learning.

Google. (2018). TensorFlow. Retrieved November 22, 2018, from https://www.tensorflow.org/

Hadad, E., Fishman, D., Hadad, E., & Gannot, S. (2014). A study of 3D audio rendering by headphones. *2014 IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI)*, 1–4. https://doi.org/10.1109/EEEI.2014.7005862

Haigh, A., Brown, D. J., Meijer, P., & Proulx, M. J. (2013). How well do you see what you hear? The acuity of visual-to-auditory sensory substitution. *Frontiers in Psychology*, *4*, 330. https://doi.org/10.3389/fpsyg.2013.00330

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2), 87–92. https://doi.org/http://aisel.aisnet.org/sjis/vol19/iss2/4

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105. https://doi.org/10.2307/25148625

Hoffmann, R., Spagnol, S., Kristjánsson, Á., & Unnthorsson, R. (2018). Evaluation of an Audio-haptic Sensory Substitution Device for Enhancing Spatial Awareness for the Visually Impaired. *Optometry and Vision Science*, *95*(9), 757–765. https://doi.org/10.1097/OPX.0000000000001284

Hofman, P. M., Van Riswick, J. G. A., & Opstal, A. J. Van. (1998). Relearning sound localization with new ears. *Nature Neuroscience*, *1*(5), 417–421. https://doi.org/10.1038/1633

Howard, D. M. (David M., & Angus, J. (2009). *Acoustics and psychoacoustics*. Focal.

Hudspeth, A. J. (2014). Integrating the active process of hair cells with cochlear function. *Nature Reviews Neuroscience*, *15*(9), 600–614. https://doi.org/10.1038/nrn3786

Jonas, J. B., Schmidt, A. M., Müller-Bergh, J. A., Schlötzer-Schrehardt, U. M., & Naumann, G. O. (1992). Human optic nerve fiber count and optic disc size. *Investigative Ophthalmology & Visual Science*, *33*(6), 2012–2018. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/1582806

Kaczmarek, K. A., Webster, J. G., Bach-y-Rita, P., & Tompkins, W. J. (1991). Electrotactile and vibrotactile displays for sensory substitution systems. *IEEE Transactions on Biomedical Engineering*, *38*(1), 1–16. https://doi.org/10.1109/10.68204

Khorrami, P., Le Paine, T., & Huang, T. S. (2015). *Do Deep Neural Networks Learn Facial Action Units When Doing Expression Recognition?* Retrieved from https://github.com/ifp-uiuc/anna

Le, Q. V, Ngiam, J., Chen, Z., Chia, D., Koh, W., & Ng, A. Y. (2010). *Tiled convolutional neural networks*. Retrieved from http://papers.nips.cc/paper/4136-tiled-convolutional-neural-networks.pdf

Lenay, C., Gapenne, O., Hanneton, S., Marque, C., & Genouëlle, C. (2003). Sensory Substitution : Limits and Perspectives. *Touching for Knowing Cognitive Psychology of Haptic Manual Perception*, *19*, 275–292. Retrieved from https://web.archive.org/web/20061122061515/http://www.utc.fr/gsp/publi/Lenay03-SensorySubstitution.pdf

Makous, J. C., & Middlebrooks, J. C. (1990). Two-dimensional sound localization by human listeners. *The Journal of the Acoustical Society of America*, *87*(5), 2188–2200. https://doi.org/10.1121/1.399186

Meijer, P. (1992). An experimental system for auditory image representations. *IEEE Transactions on Biomedical Engineering*, *39*(2), 112–121. https://doi.org/10.1109/10.121642

Meijer, P. (2017). *The vOICe website*. Retrieved from https://www.seeingwithsound.com/

Mozilla Developer Network. (2017). PannerNode. Retrieved May 14, 2017, from https://developer.mozilla.org/en-US/docs/Web/API/PannerNode

Musiek, F. E., & Chermak, G. D. (2013). *Handbook of central auditory processing disorder. Volume 1, Auditory neuroscience and diagnosis*.

Nave, C. R. (2017a). Interference of Sound. Retrieved December 4, 2018, from http://hyperphysics.phy-astr.gsu.edu/hbase/Sound/interf.html

Nave, C. R. (2017b). Inverse Square Law for Sound. Retrieved October 18, 2018, from http://hyperphysics.phy-astr.gsu.edu/hbase/Acoustic/invsqs.html#c2

Newegg. (n.d.). Aftershokz Sportz 2 AS320. Retrieved November 29, 2018, from https://www.newegg.com/Product/Product.aspx?Item=0TH-002R-00002

Oldfield, S. R., & Parker, S. P. A. (1984). Acuity of Sound Localisation: A Topography of Auditory

Space. II. Pinna Cues Absent. *Perception*, *13*(5), 601–617. https://doi.org/10.1068/p130601

Open Source Robotics Foundation. (2018). ROS. Retrieved November 21, 2018, from http://www.ros.org/about-ros/

Open Source Robotics Foundation. (2019). ROS Concepts. Retrieved February 28, 2019, from http://wiki.ros.org/ROS/Concepts

OpenAL. (2017). OpenAL. Retrieved May 14, 2017, from https://openal.org/

OpenCV team. (2018). OpenCV library. Retrieved November 22, 2018, from https://opencv.org/

Peffers, K., Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., … Bragge, J. (2006). The Design Science Research Process: A Model for Producing and Presenting Information Systems Research. *IN: 1ST INTERNATIONAL CONFERENCE ON DESIGN SCIENCE IN INFORMATION SYSTEMS AND TECHNOLOGY (DESRIST*, 83--106. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.469.2936

Plack, C. J. (2005). *The Sense of Hearing* (2nd Editio). Routledge.

Potisk, T., & Svenšek, D. (2015). *Head-Related Transfer Function*. Retrieved from https://www.semanticscholar.org/paper/Head-Related-Transfer-Function-Potisk-Svenšek/239714fced9554364fb1dfdd2cd071a89f72bcd5

Proulx, M. J., Stoerig, P., Ludowig, E., Knoll, I., & Schnitzler, A. (2008). Seeing 'Where'' through the Ears: Effects of Learning-by-Doing and Long-Term Sensory Deprivation on Localization Based on Image-to-Sound Substitution.' *PLoS ONE*, *3*(3), e1840. https://doi.org/10.1371/journal.pone.0001840

Pruitt, B. (2018). Choosing an Intel® RealSense™ Depth Camera. Retrieved November 29, 2018, from https://realsense.intel.com/compare/

Python Software Foundation. (2018). PEP 8 - Style Guide for Python Code. Retrieved November 22, 2018, from https://www.python.org/dev/peps/pep-0008/

Reich, L., Maidenbaum, S., & Amedi, A. (2012). *The brain as a flexible task machine: implications for visual rehabilitation using noninvasive vs. invasive approaches*. https://doi.org/10.1097/WCO.0b013e32834ed723

Renier, L., Collignon, O., Poirier, C., Tranduy, D., Vanlierde, A., Bol, A., … De Volder, A. G. (2005). Cross-modal activation of visual cortex during depth perception using auditory substitution of vision. *NeuroImage*, *26*(2), 573–580. https://doi.org/10.1016/j.neuroimage.2005.01.047

Renier, L., & De Volder, A. G. (2005). Cognitive and brain mechanisms in sensory substitution of vision: a contribution to the study of human perception. *Journal of Integrative Neuroscience*, *4*(4), 489–503. https://doi.org/10.1142/S0219635205000999

Roffler, S. K., & Butler, R. A. (1968a). Factors That Influence the Localization of Sound in the Vertical Plane. *The Journal of the Acoustical Society of America*, *43*(6), 1255–1259. https://doi.org/10.1121/1.1910976

Roffler, S. K., & Butler, R. A. (1968b). Localization of tonal stimuli in the vertical plane. *The Journal of the Acoustical Society of America*, *43*(6), 1260–1266. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/5659494

Schiller, P. H., & Brown, M. C. (2013). 9.04 Sensory Systems. *Massachusetts Institute of Technology: MIT OpenCourseWare*. Retrieved from https://ocw.mit.edu/courses/brain-and-cognitive-sciences/9-04-sensory-systems-fall-2013/%7B#%7D

Script Tutorials. (2018). Retrieved August 3, 2018, from https://www.script-tutorials.com/webgl-with-three-js-lesson-9/

Shackleton, T. M., Skottun, B. C., Arnott, R. H., & Palmer, A. R. (2003). Interaural Time Difference Discrimination Thresholds for Single Neurons in the Inferior Colliculus of Guinea Pigs. *Journal of Neuroscience*, *23*(2). Retrieved from http://www.jneurosci.org/content/23/2/716.long

Shinn-Cunningham, B. (2013). Auditory Precedence Effect. In *Encyclopedia of Computational Neuroscience* (pp. 1–3). https://doi.org/10.1007/978-1-4614-7320-6_101-5

Simon, H. A. (Herbert A. (1996). *The sciences of the artificial*. MIT Press.

Smith, R. C. G., Price, S. R., Knudsen, P. F., Tollin, D. J., & Wagner, H. (2014). Modelling of Human Low Frequency Sound Localization Acuity Demonstrates Dominance of Spatial Variation of Interaural Time Difference and Suggests Uniform Just-Noticeable Differences in Interaural Time Difference. *PLoS ONE*, *9*(2), e89033. https://doi.org/10.1371/journal.pone.0089033

Sodnik, J., Umek, A., Susnik, R., Bobojevic, G., & Tomazic, S. (2004). Representation of head related transfer functions with principal component analysis. *Proceedings of the Annual Conference of the Australian Acoustical Society, NSW.*

Spagnol, S. (2012). Are spectral elevation cues in head-related transfer functions distance-independent? *Proceedings of the 19th CIM, Trieste, November 21-24, 2012*, 166–171. Retrieved from https://notendur.hi.is/~spagnols/pubbl/CIM_2012a.pdf

Spoendlin, H., & Schrott, A. (1989). Analysis of the human auditory nerve. *Hearing Research*, *43*(1), 25–38. https://doi.org/10.1016/0378-5955(89)90056-7

Stereo Audio Test. (2011). Retrieved November 28, 2018, from https://www.youtube.com/watch?v=4bJ0dvAl98k

Stoll, C., Palluel-Germain, R., Fristot, V., Pellerin, D., Alleysson, D., & Graff, C. (2015). Navigating from a Depth Image Converted into Sound. *Applied Bionics and Biomechanics*, *2015*, 1–9. https://doi.org/10.1155/2015/543492

Striem-Amit, E., Guendelman, M., Amedi, A., Carlson, S., & VanMeter, J. (2012). 'Visual'' Acuity of the Congenitally Blind Using Visual-to-Auditory Sensory Substitution.' *PLoS ONE*, *7*(3), e33136. https://doi.org/10.1371/journal.pone.0033136

Stronks, H. C., Nau, A. C., Ibbotson, M. R., & Barnes, N. (2015). The role of visual deprivation and experience on the performance of sensory substitution devices. *Brain Research*, *1624*, 140–152. https://doi.org/10.1016/j.brainres.2015.06.033

Suh, D.-Y., Kim, K.-Y., Park, G.-H., & Suh, D.-Y. (2010). *Adaptive Depth-Map Coding for 3D-Video*. https://doi.org/10.1587/transinf.E93.D.2262

Suzuki, Y. (2003). *Precise and Full-range Determination of Two-dimensional Equal Loudness Contours*. Retrieved from https://web.archive.org/web/20070927210848/http://www.nedo.go.jp/itd/grant-e/report/00pdf/is-01e.pdf

Taylor, E. (2017). An Obstacle Avoidance System for the Visually Impaired Using 3-D Point Cloud Processing. *All Theses and Dissertations*. Retrieved from https://scholarsarchive.byu.edu/etd/6614

Thompson, D. M. (2005). *Understanding Audio: Getting the Most Out of Your Project or Professional Recording Studio*. Berklee Press.

Unity. (2017). Unity - Audio Listner. Retrieved May 14, 2017, from https://docs.unity3d.com/Manual/class-AudioListener.html

Velázquez, R. (2010). Wearable assistive devices for the blind. *Lecture Notes in Electrical Engineering*, *75 LNEE*, 331–349. https://doi.org/10.1007/978-3-642-15687-8-17

Veraart, C. (1989). Neurophysiological approach to the design of visual prostheses: a theoretical discussion. *Journal of Medical Engineering & Technology*, *13*(1–2), 57–62. https://doi.org/10.3109/03091908909030196

von Appen, M. (2017). *PyAL Documentation Release 0.2.0*. Retrieved from https://media.readthedocs.org/pdf/pyal/latest/pyal.pdf

Ward, J., & Wright, T. D. (2014). Sensory Substitution Devices as Advanced Sensory Tools. Retrieved October 8, 2018, from https://www.researchgate.net/publication/266315050_Sensory_Substitution_Devices_as_Advanced_Sensory_Tools

WHO | Visual impairment and blindness. (2014). *Visual Impairment and Blindness*. Retrieved from http://www.who.int/mediacentre/factsheets/fs282/en/

Wolfe, J. M., Kluender, K. R., Levi, D. M., Bartoshuk, L. M., Herz, R. S., Lederman, S. J., & Merfeld, D. M. (Daniel M. . (2014). *Sensation &amp; perception*.

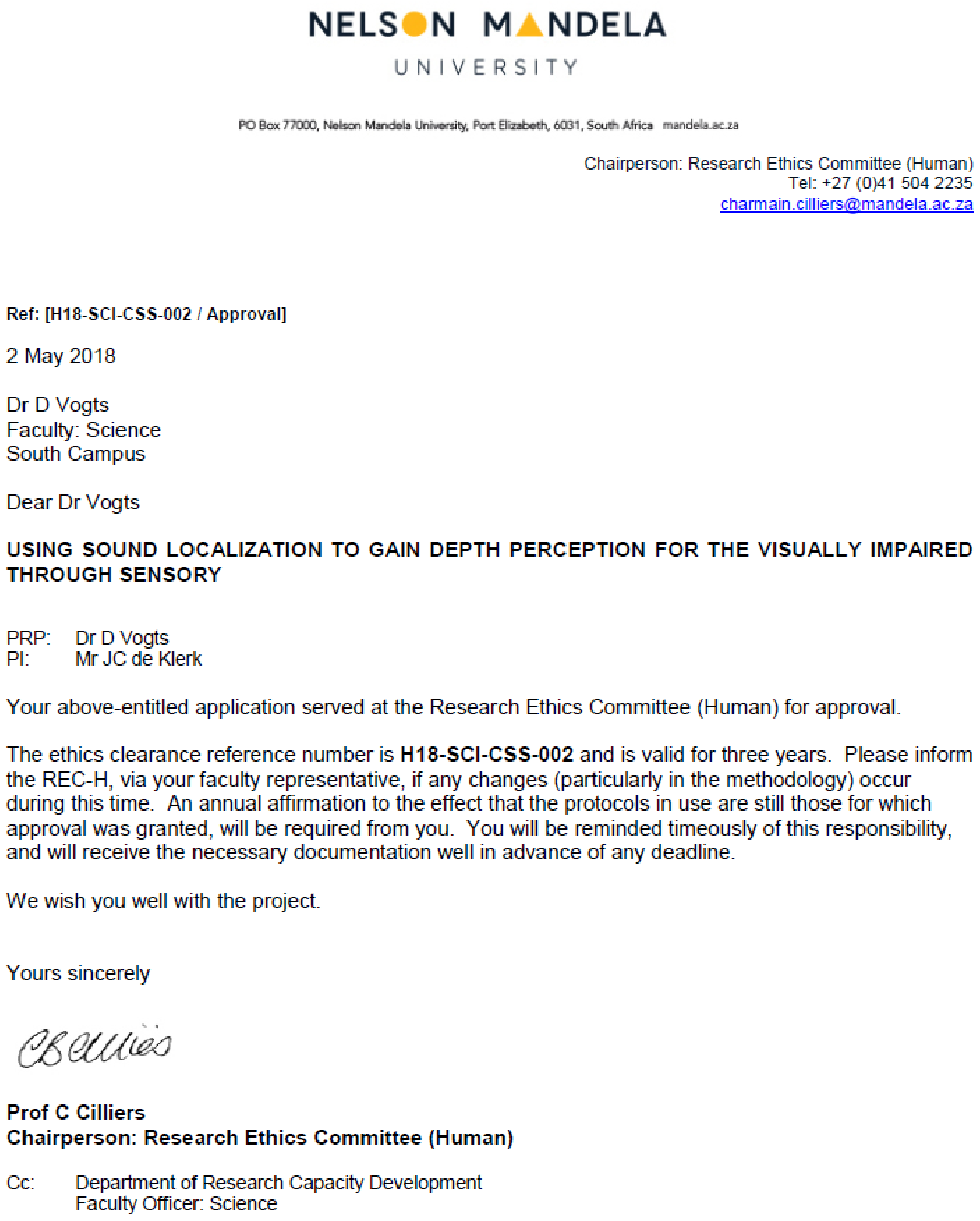
World Health Organization. (2016). ICD-10 Version:2016. Retrieved December 3, 2017, from http://apps.who.int/classifications/icd10/browse/2016/en#/H53-H54

Wright, T. D., Margolis, A., & Ward, J. (2015). Using an auditory sensory substitution device to augment vision: evidence from eye movements. *Experimental Brain Research*, *233*(3), 851–860. https://doi.org/10.1007/s00221-014-4160-1

Zwicker, E., & Fastl, H. (Hugo). (1999). *Psychoacoustics: Facts and Models* (3 edition). Springer.

Appendices

Appendix A: Ethics Approval



NELSON MANDELA
UNIVERSITY

Chairperson: Research Ethics Committee (Human)
Tel: +27 (0)41 504 2235
charmain.cilliers@mandela.ac.za

Ref: [H18-SCI-CSS-002 / Approval]

2 May 2018

Dr D Vogts
Faculty: Science
South Campus

Dear Dr Vogts

USING SOUND LOCALIZATION TO GAIN DEPTH PERCEPTION FOR THE VISUALLY IMPAIRED THROUGH SENSORY

PRP: Dr D Vogts
PI: Mr JC de Klerk

Your above-entitled application served at the Research Ethics Committee (Human) for approval.

The ethics clearance reference number is **H18-SCI-CSS-002** and is valid for three years. Please inform the REC-H, via your faculty representative, if any changes (particularly in the methodology) occur during this time. An annual affirmation to the effect that the protocols in use are still those for which approval was granted, will be required from you. You will be reminded timeously of this responsibility, and will receive the necessary documentation well in advance of any deadline.

We wish you well with the project.

Yours sincerely

CBCilliers

Prof C Cilliers
Chairperson: Research Ethics Committee (Human)

Cc: Department of Research Capacity Development
Faculty Officer: Science

Appendix B: Written Information

Department of Computing Sciences
Faculty of Science
Nelson Mandela University
Tel: +27 41 504 2322

Date: .. … ….

Ref: H18-SCI-CSS-002
Contact persons: Dr. Dieter Vogts and James Carmichael de Klerk

Dear Participant

You are being asked to participate in a research study titled *"USING SOUND LOCALIZATION TO GAIN DEPTH PERCEPTION FOR THE VISUALLY IMPAIRED THROUGH SENSORY SUBSTITUTION"*. The researcher has developed non-invasive sensory substitution hardware and software relating to the title of the project. The purpose of the study is for the researcher to evaluate the hardware and software developed for visual to auditory sensory substitution. Participation in the study will require approximately 2 hours of your time.

We will provide you with the necessary information to assist you to understand the study and explain what would be expected of you (participant). Please feel free to ask the researcher to clarify anything that is not clear to you.

To participate, it will be required of you to provide written consent that will include your signature, date and initials to verify that you understand and agree to the conditions. It is important that you are aware of the fact that the ethical integrity of the study has been approved by the Nelson Mandela University Research Ethics Committee: Human (***H18-SCI-CSS-002***).

You have the right to query concerns regarding the study at any time using the telephone numbers below:

- Researcher: +27 76 728 6305 (James Carmichael de Klerk)
- Supervisor: +27 41 504 2089 (Dr. Dieter Vogts)
- REC-H: +27 41 504 3140

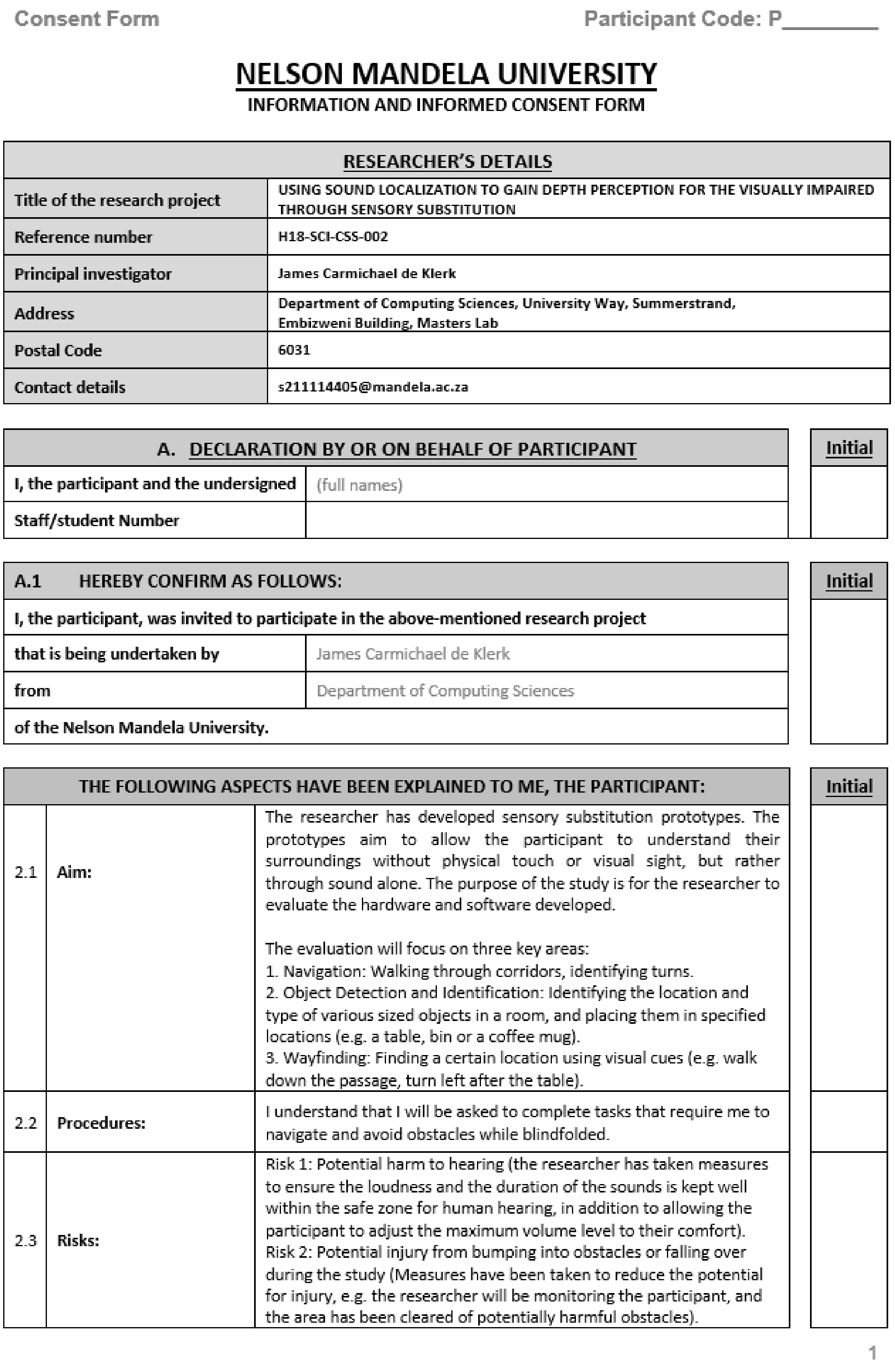
If you participate, you have the right to withdraw at any given time during the study without penalty or loss of benefits. Your identity will at all times remain confidential. The results of the research study may be presented at scientific conferences or in specialist publications.

This informed consent statement has been prepared in compliance with current statutory guidelines.

Yours sincerely
James Carmichael de Klerk

MSc Computer Science and Information Systems
Department of Computing Sciences
Nelson Mandela University

Appendix C: Consent Form



Consent Form Participant Code: P________

NELSON MANDELA UNIVERSITY

INFORMATION AND INFORMED CONSENT FORM

RESEARCHER'S DETAILS	
Title of the research project	USING SOUND LOCALIZATION TO GAIN DEPTH PERCEPTION FOR THE VISUALLY IMPAIRED THROUGH SENSORY SUBSTITUTION
Reference number	H18-SCI-CSS-002
Principal investigator	James Carmichael de Klerk
Address	Department of Computing Sciences, University Way, Summerstrand, Embizweni Building, Masters Lab
Postal Code	6031
Contact details	s211114405@mandela.ac.za

A. DECLARATION BY OR ON BEHALF OF PARTICIPANT		Initial
I, the participant and the undersigned	(full names)	
Staff/student Number		

A.1 HEREBY CONFIRM AS FOLLOWS:		Initial
I, the participant, was invited to participate in the above-mentioned research project		
that is being undertaken by	James Carmichael de Klerk	
from	Department of Computing Sciences	
of the Nelson Mandela University.		

THE FOLLOWING ASPECTS HAVE BEEN EXPLAINED TO ME, THE PARTICIPANT:			Initial
2.1	Aim:	The researcher has developed sensory substitution prototypes. The prototypes aim to allow the participant to understand their surroundings without physical touch or visual sight, but rather through sound alone. The purpose of the study is for the researcher to evaluate the hardware and software developed. The evaluation will focus on three key areas: 1. Navigation: Walking through corridors, identifying turns. 2. Object Detection and Identification: Identifying the location and type of various sized objects in a room, and placing them in specified locations (e.g. a table, bin or a coffee mug). 3. Wayfinding: Finding a certain location using visual cues (e.g. walk down the passage, turn left after the table).	
2.2	Procedures:	I understand that I will be asked to complete tasks that require me to navigate and avoid obstacles while blindfolded.	
2.3	Risks:	Risk 1: Potential harm to hearing (the researcher has taken measures to ensure the loudness and the duration of the sounds is kept well within the safe zone for human hearing, in addition to allowing the participant to adjust the maximum volume level to their comfort). Risk 2: Potential injury from bumping into obstacles or falling over during the study (Measures have been taken to reduce the potential for injury, e.g. the researcher will be monitoring the participant, and the area has been cleared of potentially harmful obstacles).	

		Note: Should an injury occur during the evaluation, the South Campus Health Service (041 504 2174) will be consulted.			
2.4	**Possible benefits:**	There are no direct benefits for participants, however participation in the study may lead to benefits for the visually impaired in the future.			
2.5	**Confidentiality:**	My identity will not be revealed in any discussion, description or scientific publications by the investigators.			
2.6	**Voluntary participation / refusal / discontinuation:**	My participation is voluntary	**YES**	**NO**	
		My decision whether or not to participate will in no way affect my present or future academic performance / development / care / employment / lifestyle	**TRUE**	**FALSE**	

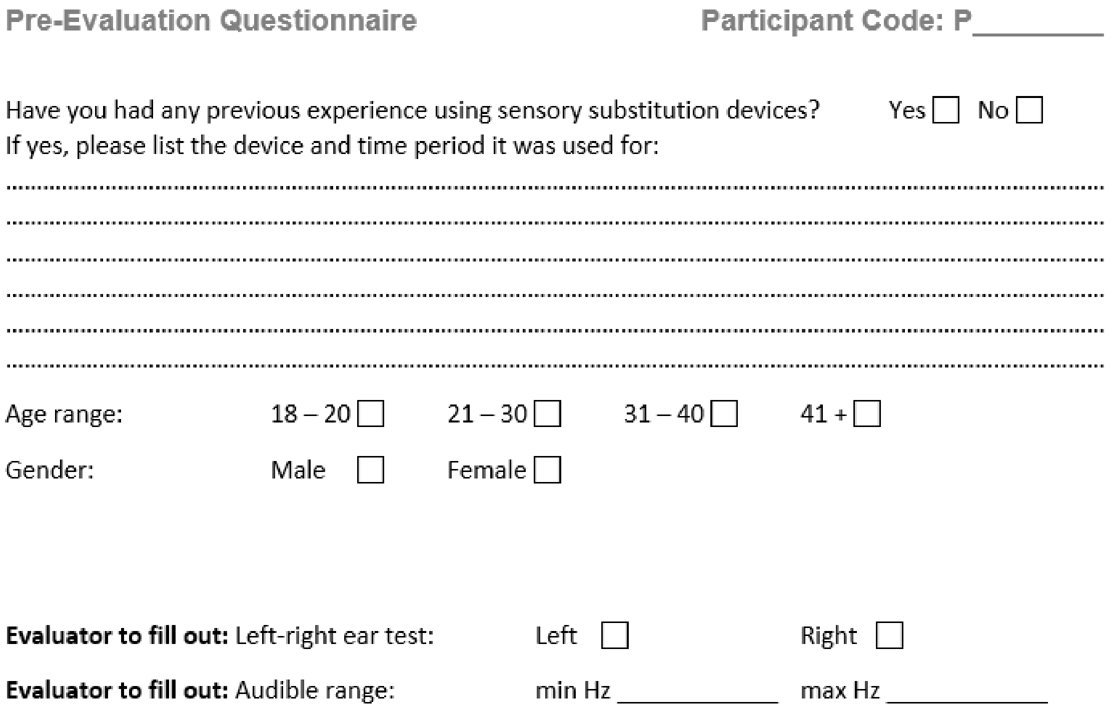
3. **THE INFORMATION ABOVE WAS EXPLAINED TO ME/THE PARTICIPANT BY:**	**Initial**
James Carmichael de Klerk, in English	
I was given the opportunity to ask questions and all these questions were answered satisfactorily.	

4.	No pressure was exerted on me to consent to participation and I understand that I may withdraw at any stage without penalisation.	

5.	Participation in this study will not result in any additional cost to myself.	

A.2 **I HEREBY VOLUNTARILY CONSENT TO PARTICIPATE IN THE ABOVE-MENTIONED PROJECT:**	
Signed/confirmed at on 2018	
Signature of participant	Signature of witness: Full name of witness:

Appendix D: Pre-Evaluation Questionnaire



Pre-Evaluation Questionnaire **Participant Code: P________**

Have you had any previous experience using sensory substitution devices? Yes □ No □

If yes, please list the device and time period it was used for:

……

……

……

……

……

……

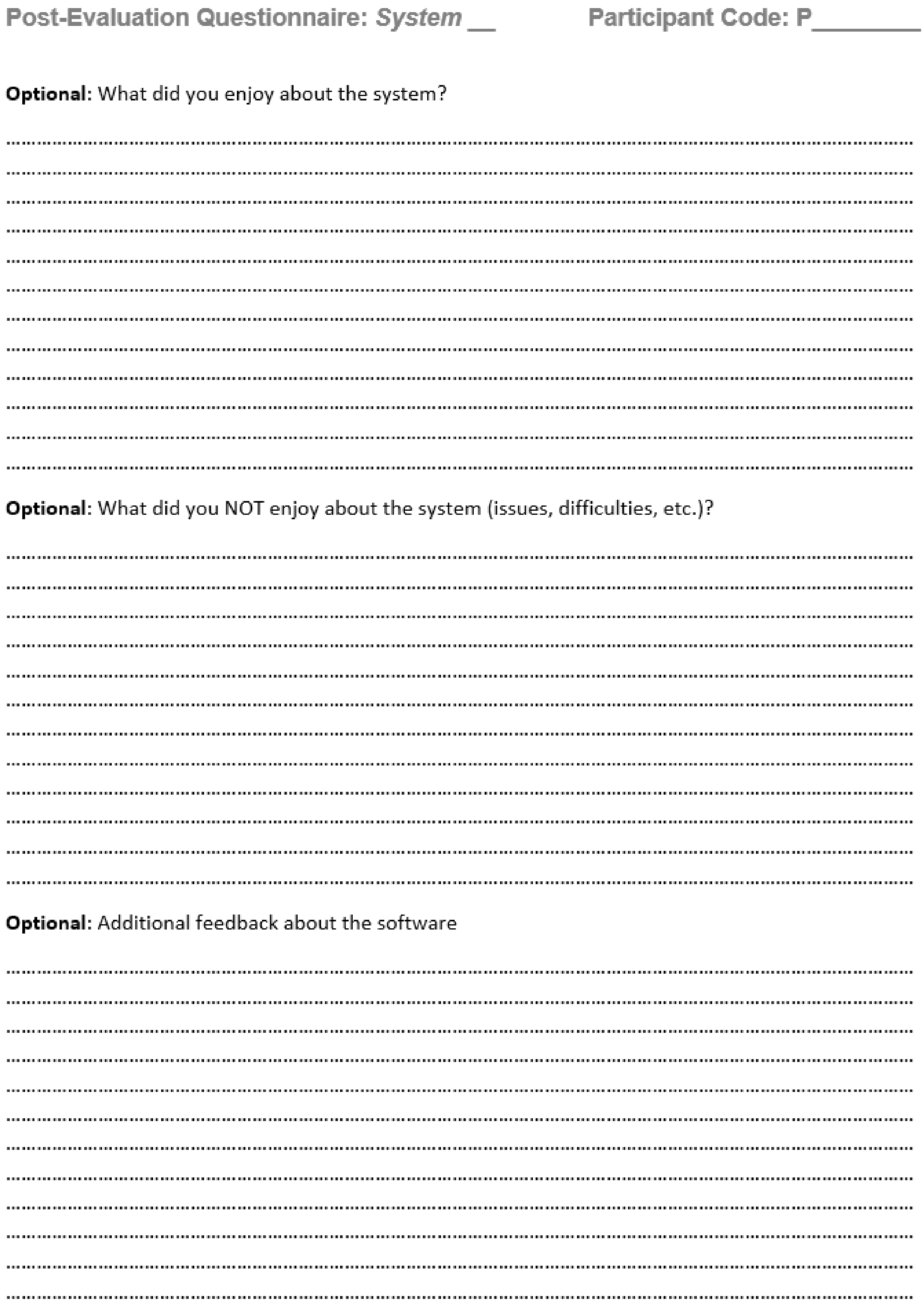
Age range: 18 – 20 □ 21 – 30 □ 31 – 40 □ 41 + □

Gender: Male □ Female □

Evaluator to fill out: Left-right ear test: Left □ Right □

Evaluator to fill out: Audible range: min Hz ____________ max Hz ____________

Appendix E: Post-Evaluation Questionnaire



Post-Evaluation Questionnaire: *System* __ **Participant Code: P________**

Optional: What did you enjoy about the system?

……

……

……

……

……

……

……

……

……

……

……

……

Optional: What did you NOT enjoy about the system (issues, difficulties, etc.)?

……

……

……

……

……

……

……

……

……

……

……

……

Optional: Additional feedback about the software

……

……

……

……

……

……

……

……

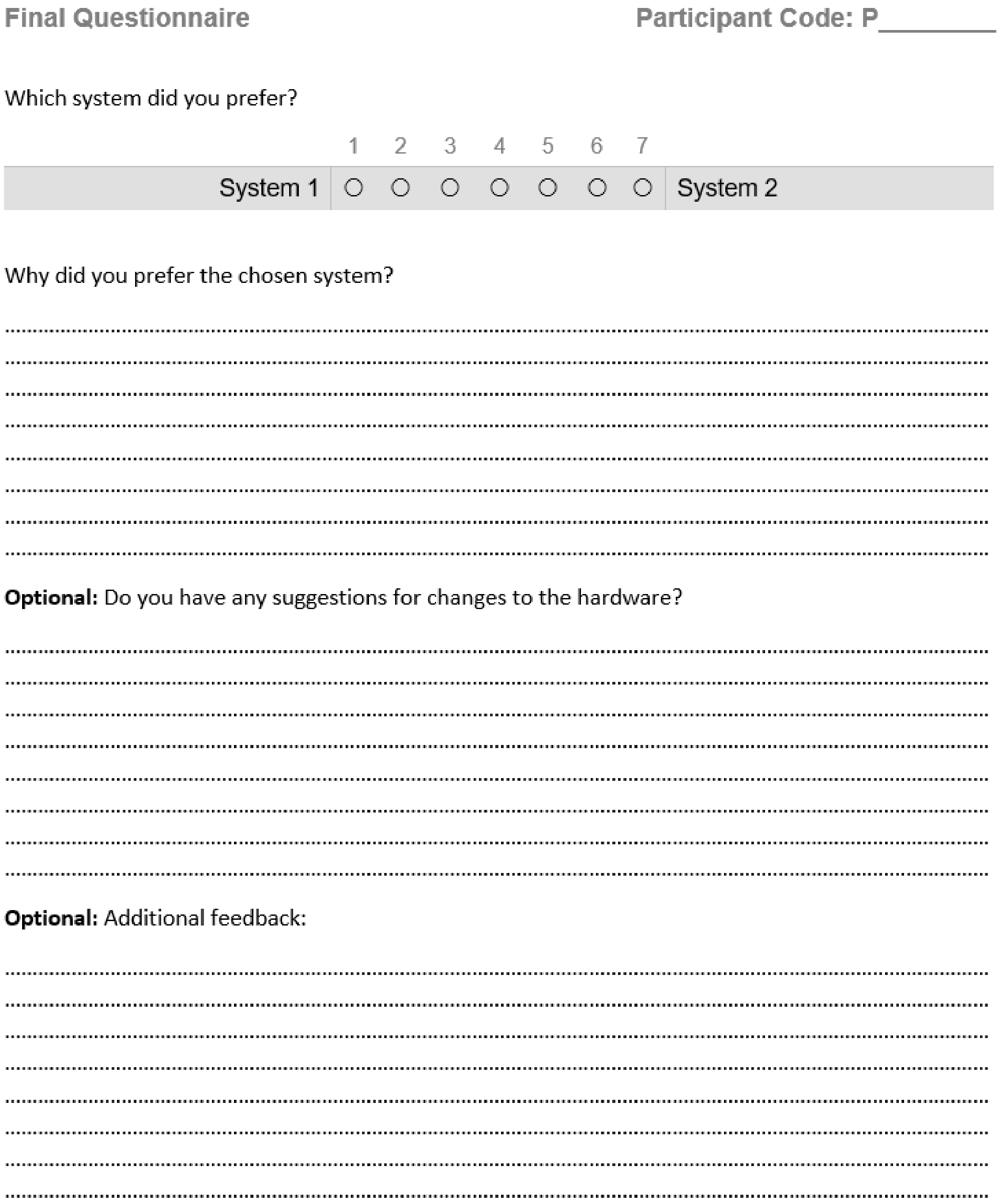
……

……

……

……

Appendix F: Final Questionnaire



Final Questionnaire **Participant Code: P________**

Which system did you prefer?

	1	2	3	4	5	6	7	
System 1	○	○	○	○	○	○	○	System 2

Why did you prefer the chosen system?

....

....

....

....

....

....

....

....

Optional: Do you have any suggestions for changes to the hardware?

....

....

....

....

....

....

....

....

Optional: Additional feedback:

....

....

....

....

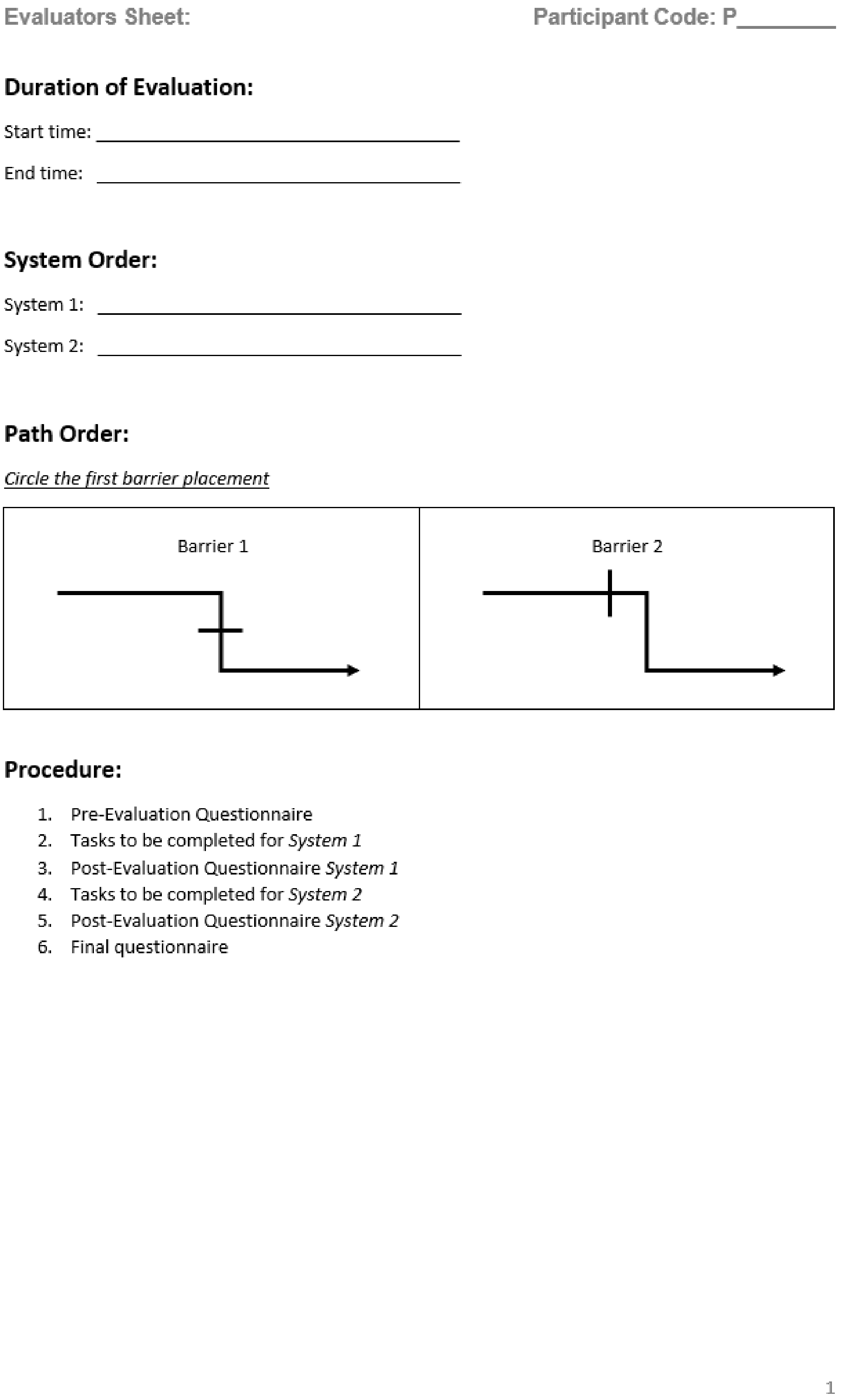
....

....

....

....

Appendix G: Evaluators Sheet



Evaluators Sheet: Participant Code: P________

Duration of Evaluation:

Start time: ____________________________________

End time: ____________________________________

System Order:

System 1: ____________________________________

System 2: ____________________________________

Path Order:

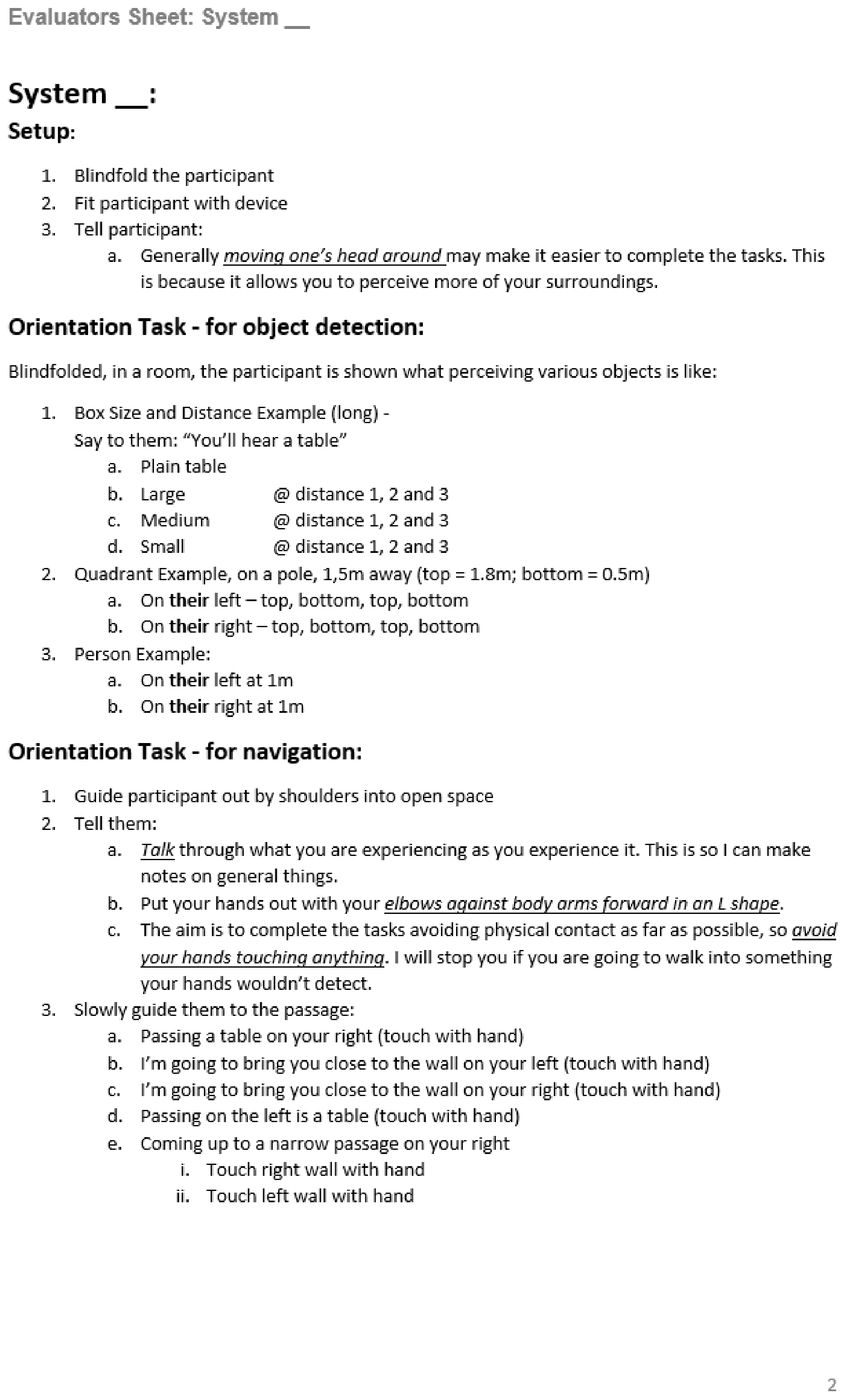
Circle the first barrier placement

Barrier 1	Barrier 2

Procedure:

1. Pre-Evaluation Questionnaire
2. Tasks to be completed for *System 1*
3. Post-Evaluation Questionnaire *System 1*
4. Tasks to be completed for *System 2*
5. Post-Evaluation Questionnaire *System 2*
6. Final questionnaire

1



Evaluators Sheet: System __

System __:

Setup:

1. Blindfold the participant
2. Fit participant with device
3. Tell participant:
    a. Generally *moving one's head around* may make it easier to complete the tasks. This is because it allows you to perceive more of your surroundings.

Orientation Task - for object detection:

Blindfolded, in a room, the participant is shown what perceiving various objects is like:

1. Box Size and Distance Example (long) -
   Say to them: "You'll hear a table"
    a. Plain table
    b. Large @ distance 1, 2 and 3
    c. Medium @ distance 1, 2 and 3
    d. Small @ distance 1, 2 and 3
2. Quadrant Example, on a pole, 1,5m away (top = 1.8m; bottom = 0.5m)
    a. On **their** left – top, bottom, top, bottom
    b. On **their** right – top, bottom, top, bottom
3. Person Example:
    a. On **their** left at 1m
    b. On **their** right at 1m

Orientation Task - for navigation:

1. Guide participant out by shoulders into open space
2. Tell them:
    a. *Talk* through what you are experiencing as you experience it. This is so I can make notes on general things.
    b. Put your hands out with your *elbows against body arms forward in an L shape*.
    c. The aim is to complete the tasks avoiding physical contact as far as possible, so *avoid your hands touching anything*. I will stop you if you are going to walk into something your hands wouldn't detect.
3. Slowly guide them to the passage:
    a. Passing a table on your right (touch with hand)
    b. I'm going to bring you close to the wall on your left (touch with hand)
    c. I'm going to bring you close to the wall on your right (touch with hand)
    d. Passing on the left is a table (touch with hand)
    e. Coming up to a narrow passage on your right
        i. Touch right wall with hand
        ii. Touch left wall with hand

Evaluators Sheet: System __

Navigation Task - NO obstacles:

The participant is guided to the start of a passage. Then told:
In front of you is a clear passage, i.e. no obstacles. Walk to the end of the passage, avoid physical contact with walls, *say "done" when you think you've reached the end*.

- Time on task: ____________ min ____________ sec
- It is recommended that errors, are counted using the tally mark system (卌)

E_{wall}	Wall touch (re-centre)		
E_{ror}	Reoriented/went off course (re-centre)		
E_{end}	Realized was at the end	T	F
E_{other}	Other errors, e.g. discrepancies between the real world and what they say they perceived		

NOTE:

Procedure for errors E_{t1} E_{gap} and E_{t2} during the wayfinding task: allow a maximum of 2 errors per error type.
Upon one of these types of errors, walk them back roughly 2m and get them to continue. If the participant has made two errors of a specific type, centre them at that turn/gap and get them to continue onto the next one.

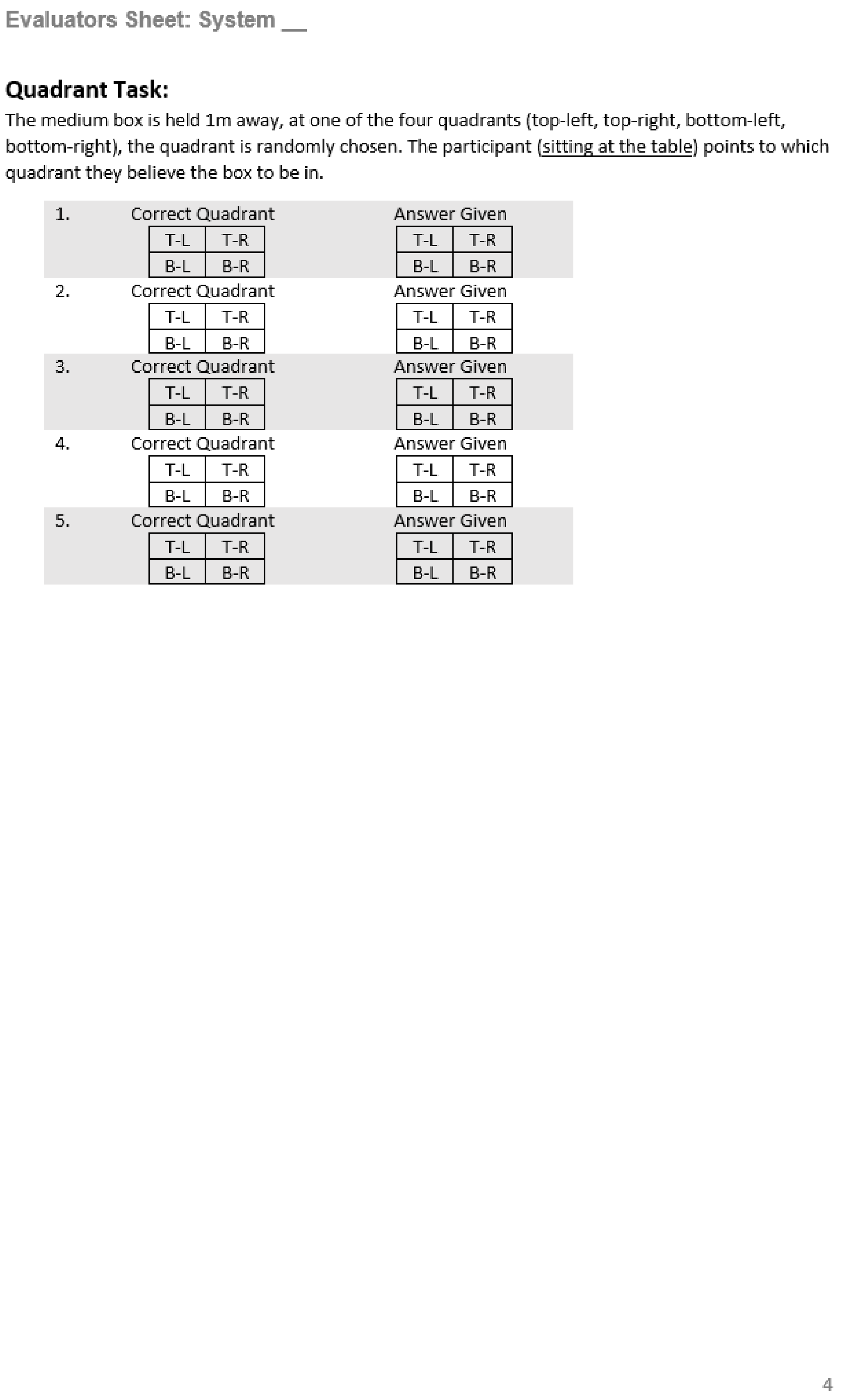
Navigation Task - WITH Obstacles:

The participant is guided to the start of a new passage. Given directions and asked to follow those directions. Avoid physical contact with walls. *Say "done" when you think you've reached the end*.

The participant must tell you after they believe they have completed turn 1, the gap and turn 2.

- Time on task: ____________ min ____________ sec
- It is recommended that errors, are counted using the tally mark system (卌)

E_{wall}	Wall touch (re-centre)		
E_{obj}	Object touch (re-centre)		
E_{ror}	Reoriented/went off course -not including missing turns (re-centre)		
E_{t1}	Missed turn 1		
E_{gap}	Missed gap		
E_{t2}	Missed turn 2 (narrow passage)		
E_{end}	Realized was at the end	T	F
E_{other}	Other errors, e.g. discrepancies between the real world and what they say they perceived		



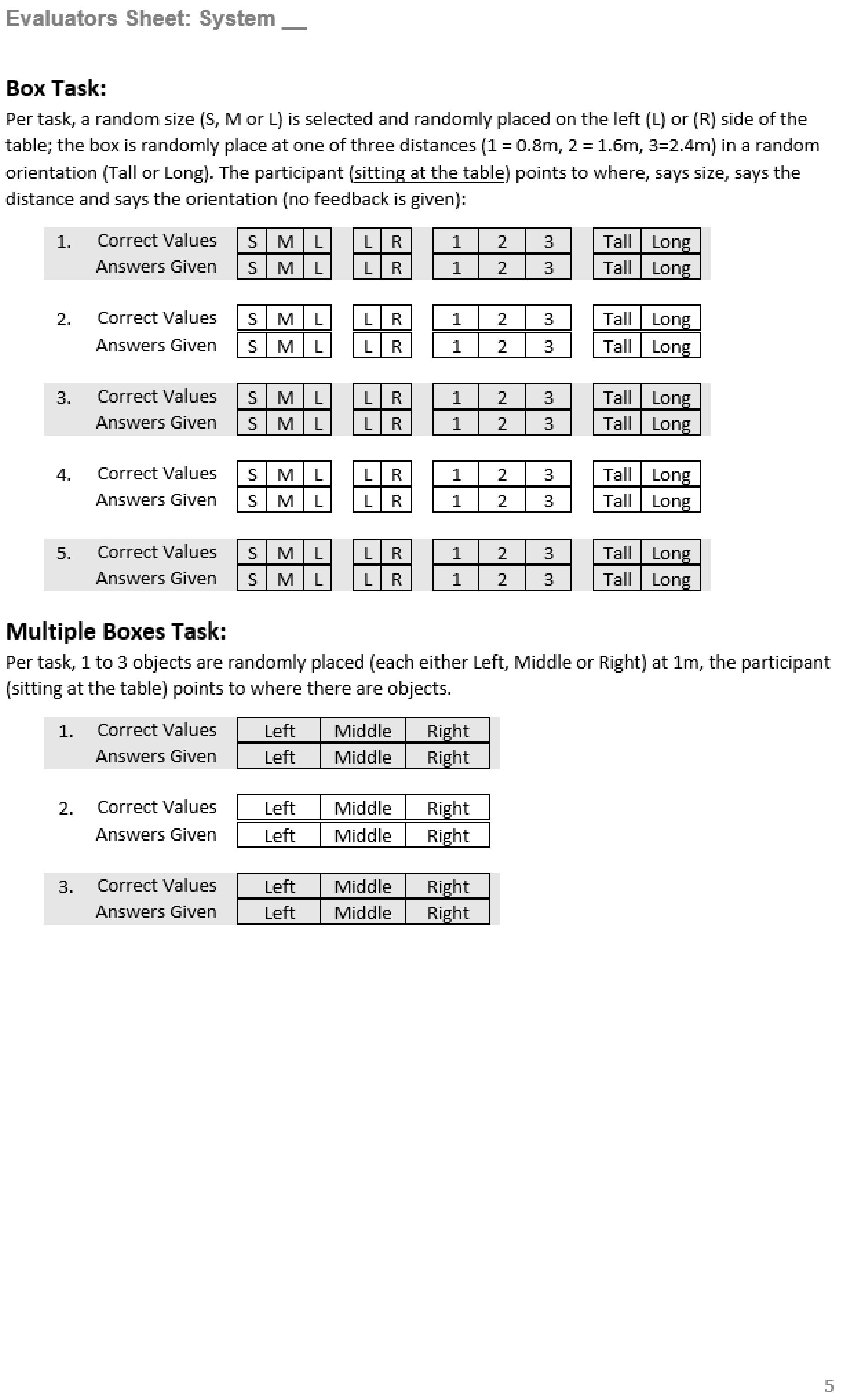
Evaluators Sheet: System __

Quadrant Task:

The medium box is held 1m away, at one of the four quadrants (top-left, top-right, bottom-left, bottom-right), the quadrant is randomly chosen. The participant (sitting at the table) points to which quadrant they believe the box to be in.

1.	Correct Quadrant		Answer Given	
	T-L	T-R	T-L	T-R
	B-L	B-R	B-L	B-R
2.	Correct Quadrant		Answer Given	
	T-L	T-R	T-L	T-R
	B-L	B-R	B-L	B-R
3.	Correct Quadrant		Answer Given	
	T-L	T-R	T-L	T-R
	B-L	B-R	B-L	B-R
4.	Correct Quadrant		Answer Given	
	T-L	T-R	T-L	T-R
	B-L	B-R	B-L	B-R
5.	Correct Quadrant		Answer Given	
	T-L	T-R	T-L	T-R
	B-L	B-R	B-L	B-R

4



Evaluators Sheet: System __

Box Task:

Per task, a random size (S, M or L) is selected and randomly placed on the left (L) or (R) side of the table; the box is randomly place at one of three distances (1 = 0.8m, 2 = 1.6m, 3=2.4m) in a random orientation (Tall or Long). The participant (sitting at the table) points to where, says size, says the distance and says the orientation (no feedback is given):

1.	Correct Values	S	M	L	L	R	1	2	3	Tall	Long
	Answers Given	S	M	L	L	R	1	2	3	Tall	Long
2.	Correct Values	S	M	L	L	R	1	2	3	Tall	Long
	Answers Given	S	M	L	L	R	1	2	3	Tall	Long
3.	Correct Values	S	M	L	L	R	1	2	3	Tall	Long
	Answers Given	S	M	L	L	R	1	2	3	Tall	Long
4.	Correct Values	S	M	L	L	R	1	2	3	Tall	Long
	Answers Given	S	M	L	L	R	1	2	3	Tall	Long
5.	Correct Values	S	M	L	L	R	1	2	3	Tall	Long
	Answers Given	S	M	L	L	R	1	2	3	Tall	Long

Multiple Boxes Task:

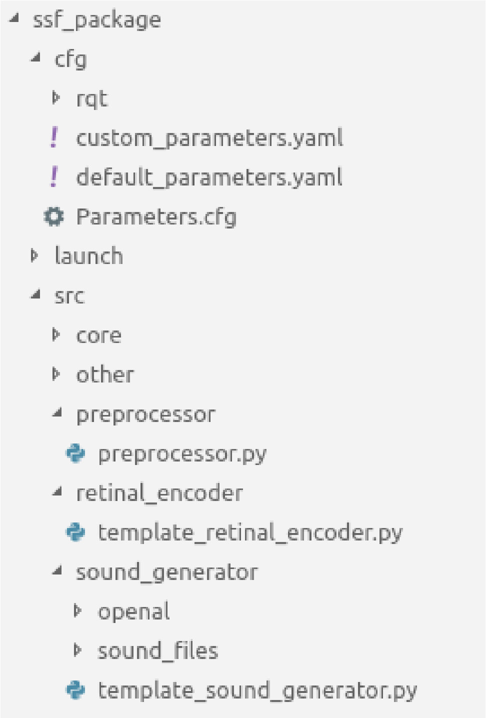
Per task, 1 to 3 objects are randomly placed (each either Left, Middle or Right) at 1m, the participant (sitting at the table) points to where there are objects.

1.	Correct Values	Left	Middle	Right
	Answers Given	Left	Middle	Right
2.	Correct Values	Left	Middle	Right
	Answers Given	Left	Middle	Right
3.	Correct Values	Left	Middle	Right
	Answers Given	Left	Middle	Right

Appendix H: SSF Usage

This appendix aims to give a brief introduction to using the SSF. The frameworks core components are the configuration files, SSF Core, the Pre-processor, the Retinal Encoder and the Sound Generator. Looking at Figure 0-1, one can see this reflected in the folder structure – with the folder names *cfg*, *core*, *pre-processor*, *retinal_encoder* and *sound_generator* respectively.

The SSF provides templates for the Retinal Encoder (*template_retinal_encoder.py*) and the Sound Generator (*template_sound_generator.py*). To create a new Retinal Encoder, one would make a copy of the file *template_retinal_encoder.py* to the same location (i.e. the *retinal_encoder* folder). Then one would rename the file to <algorithm_name>*_retinal_encoder.py*. For example, if one wanted to create a new Retinal Encoder algorithm called "algorithm_2", the name of the main file containing the algorithm should be *algorithm_2_retinal_encoder.py* and the file should be placed in the *retinal_encoder* folder. The same procedure is followed for a creating a new Sound Generator, using "*sound_generator*" in place of "*retinal_encoder*" and placing it in the *sound_generator* folder. The Pre-processor generally remains the same for all algorithms as it is not intended to do much aside from cleaning up the raw depth and colour images. If one wanted to create a new Pre-processor or edit the existing one, they would edit or replace the file *preprocessor.py*.

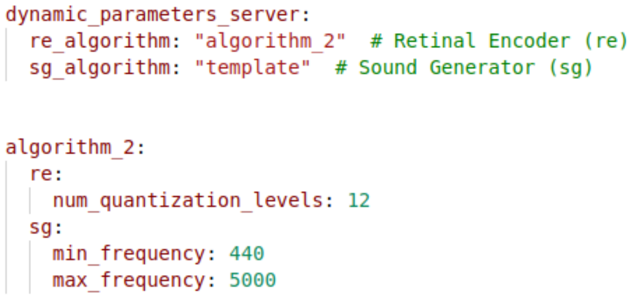


```
◢ ssf_package
  ◢ cfg
    ▷ rqt
    ! custom_parameters.yaml
    ! default_parameters.yaml
    ⚙ Parameters.cfg
  ▷ launch
  ◢ src
    ▷ core
    ▷ other
    ◢ preprocessor
        preprocessor.py
    ◢ retinal_encoder
        template_retinal_encoder.py
    ◢ sound_generator
      ▷ openal
      ▷ sound_files
        template_sound_generator.py
```

Figure 0-1: The Sensory Substitution Frameworks (SSFs) folder structure

SSF Core is found in the *core* folder. SSF Core contains a number of algorithms and functions (Section 4.4.5) used across the framework and can be used in custom Retinal Encoder or Sound Generator algorithms. If one wants to add algorithms or functions that would be used across multiple Retinal Encoders or Sound Generators, SSF Core is a good place to put them. Using SSF Core in a custom Retinal Encoder or Sound Generator is done in the same way one would use a standard Python module – once the module is imported, one simply calls the functions needed.

The *cfg* folder houses two files, *custom_parameters.py* and *default_parameters.py*. The default parameters for the camera, Pre-processor, Retinal Encoder and Sound Generator are all set in *default_parameters.py*. For each algorithm developed (whether a PP, RE or SG) one can have custom parameters set up; these are set in *custom_parameters.py* – an example is shown in Figure 0-2 where the algorithm is named "algorithm_2". In the example, "algorithm_2" has custom parameters for its Retinal Encoder and Sound Generator components – if for example "algorithm_2" was only a Retinal Encoder, the custom Sound Generator parameters could be left blank.



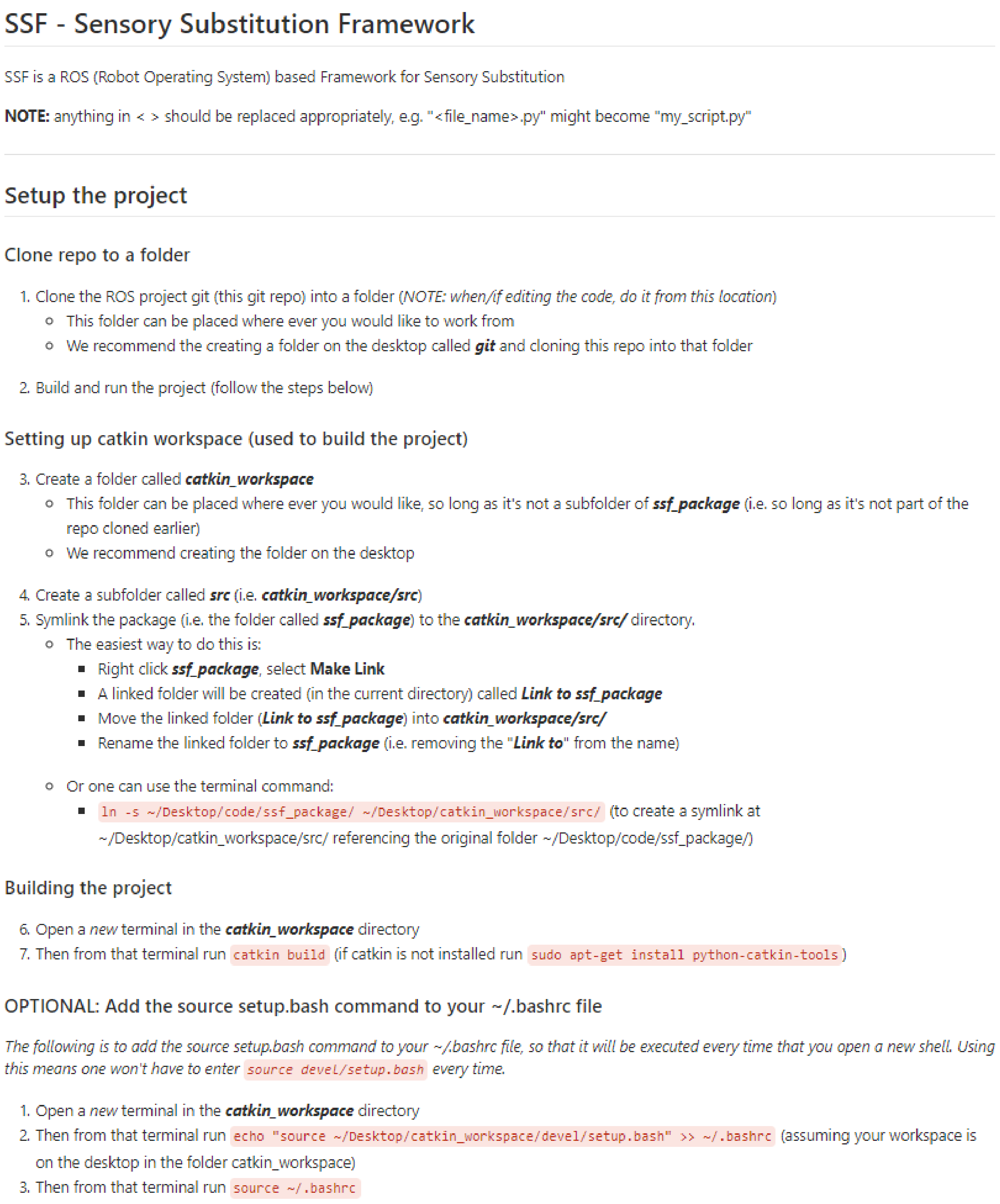
```
dynamic_parameters_server:
  re_algorithm: "algorithm_2"  # Retinal Encoder (re)
  sg_algorithm: "template"  # Sound Generator (sg)


algorithm_2:
  re:
    num_quantization_levels: 12
  sg:
    min_frequency: 440
    max_frequency: 5000
```

Figure 0-2: Example of custom parameters in *custom_parameters.py*

In *custom_parameters.py* (shown in Figure 0-2) one will also see "*dynamic_parameter_server*"; this is where one would specify the default Retinal Encoder and Sound Generator algorithms to be launched. In the case of the example in Figure 0-2, the Retinal Encoder from "algorithm_2" is launched with the "template" Sound Generator algorithm. When the program is started, the framework looks at the "*dynamic_parameter_server*" settings to find the Retinal Encoder and Sound Generator algorithms to be used. The framework then looks up the related filenames and launches those as ROS nodes. In this case the files launched are *algorithm_2_retinal_encoder.py* and *template_sound_generator.py*.

For a detailed guide to setting up and using the SSF, as well as how to launch the algorithms, how to record information to ROS bags and more, please see Appendix I.

Appendix I: Sensory Substitution Framework – README



SSF - Sensory Substitution Framework

SSF is a ROS (Robot Operating System) based Framework for Sensory Substitution

NOTE: anything in < > should be replaced appropriately, e.g. "<file_name>.py" might become "my_script.py"

Setup the project

Clone repo to a folder

1. Clone the ROS project git (this git repo) into a folder (*NOTE: when/if editing the code, do it from this location*)
   - This folder can be placed where ever you would like to work from
   - We recommend the creating a folder on the desktop called ***git*** and cloning this repo into that folder
2. Build and run the project (follow the steps below)

Setting up catkin workspace (used to build the project)

3. Create a folder called ***catkin_workspace***
   - This folder can be placed where ever you would like, so long as it's not a subfolder of ***ssf_package*** (i.e. so long as it's not part of the repo cloned earlier)
   - We recommend creating the folder on the desktop
4. Create a subfolder called ***src*** (i.e. ***catkin_workspace/src***)
5. Symlink the package (i.e. the folder called ***ssf_package***) to the ***catkin_workspace/src/*** directory.
   - The easiest way to do this is:
     - Right click ***ssf_package***, select **Make Link**
     - A linked folder will be created (in the current directory) called ***Link to ssf_package***
     - Move the linked folder (***Link to ssf_package***) into ***catkin_workspace/src/***
     - Rename the linked folder to ***ssf_package*** (i.e. removing the "***Link to***" from the name)
   - Or one can use the terminal command:
     - `ln -s ~/Desktop/code/ssf_package/ ~/Desktop/catkin_workspace/src/` (to create a symlink at ~/Desktop/catkin_workspace/src/ referencing the original folder ~/Desktop/code/ssf_package/)

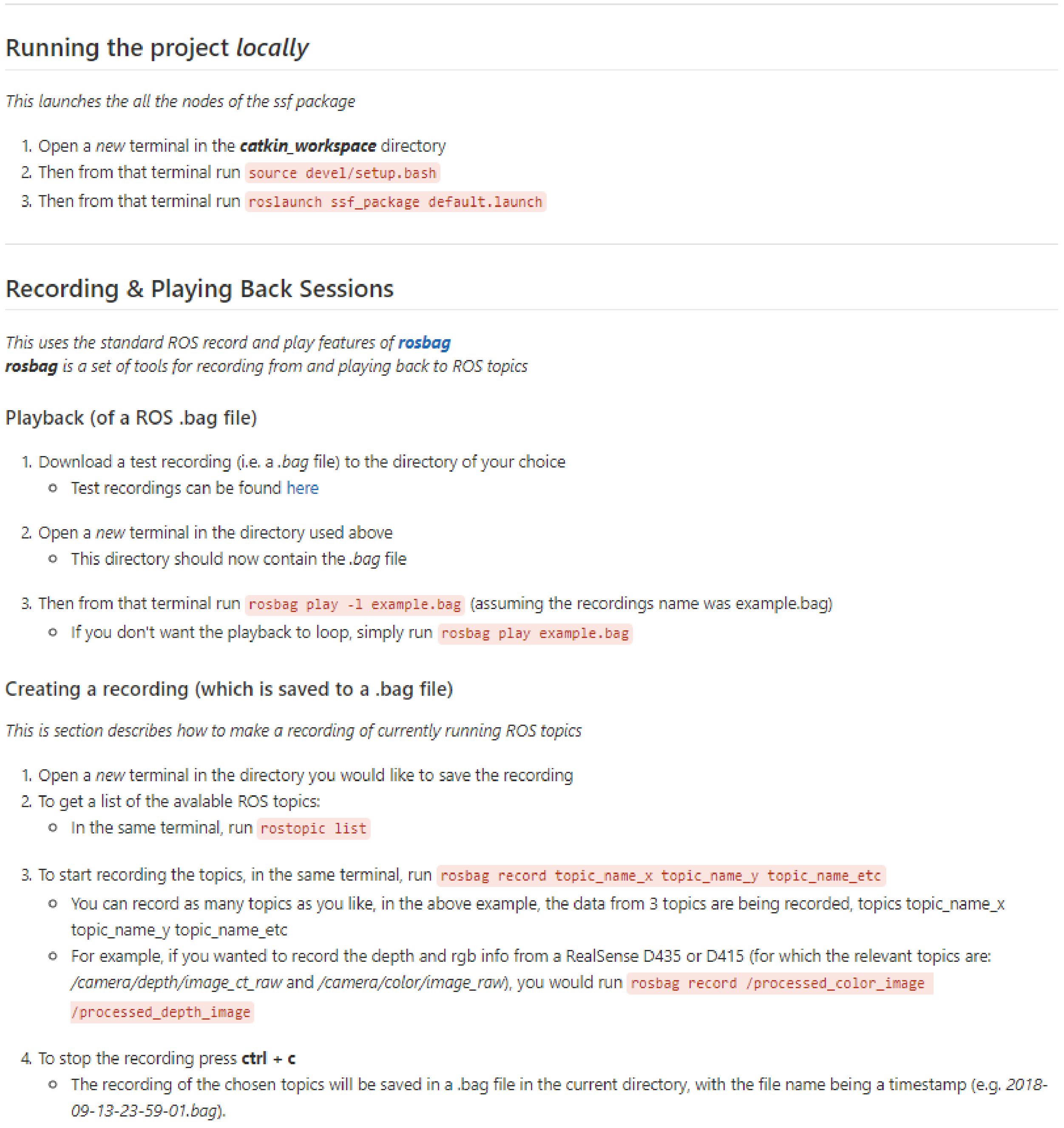
Building the project

6. Open a *new* terminal in the ***catkin_workspace*** directory
7. Then from that terminal run `catkin build` (if catkin is not installed run `sudo apt-get install python-catkin-tools`)

OPTIONAL: Add the source setup.bash command to your ~/.bashrc file

The following is to add the source setup.bash command to your ~/.bashrc file, so that it will be executed every time that you open a new shell. Using this means one won't have to enter `source devel/setup.bash` *every time.*

1. Open a *new* terminal in the ***catkin_workspace*** directory
2. Then from that terminal run `echo "source ~/Desktop/catkin_workspace/devel/setup.bash" >> ~/.bashrc` (assuming your workspace is on the desktop in the folder catkin_workspace)
3. Then from that terminal run `source ~/.bashrc`



Running the project *locally*

This launches the all the nodes of the ssf package

1. Open a *new* terminal in the ***catkin_workspace*** directory
2. Then from that terminal run `source devel/setup.bash`
3. Then from that terminal run `roslaunch ssf_package default.launch`

Recording & Playing Back Sessions

*This uses the standard ROS record and play features of **rosbag***
***rosbag** is a set of tools for recording from and playing back to ROS topics*

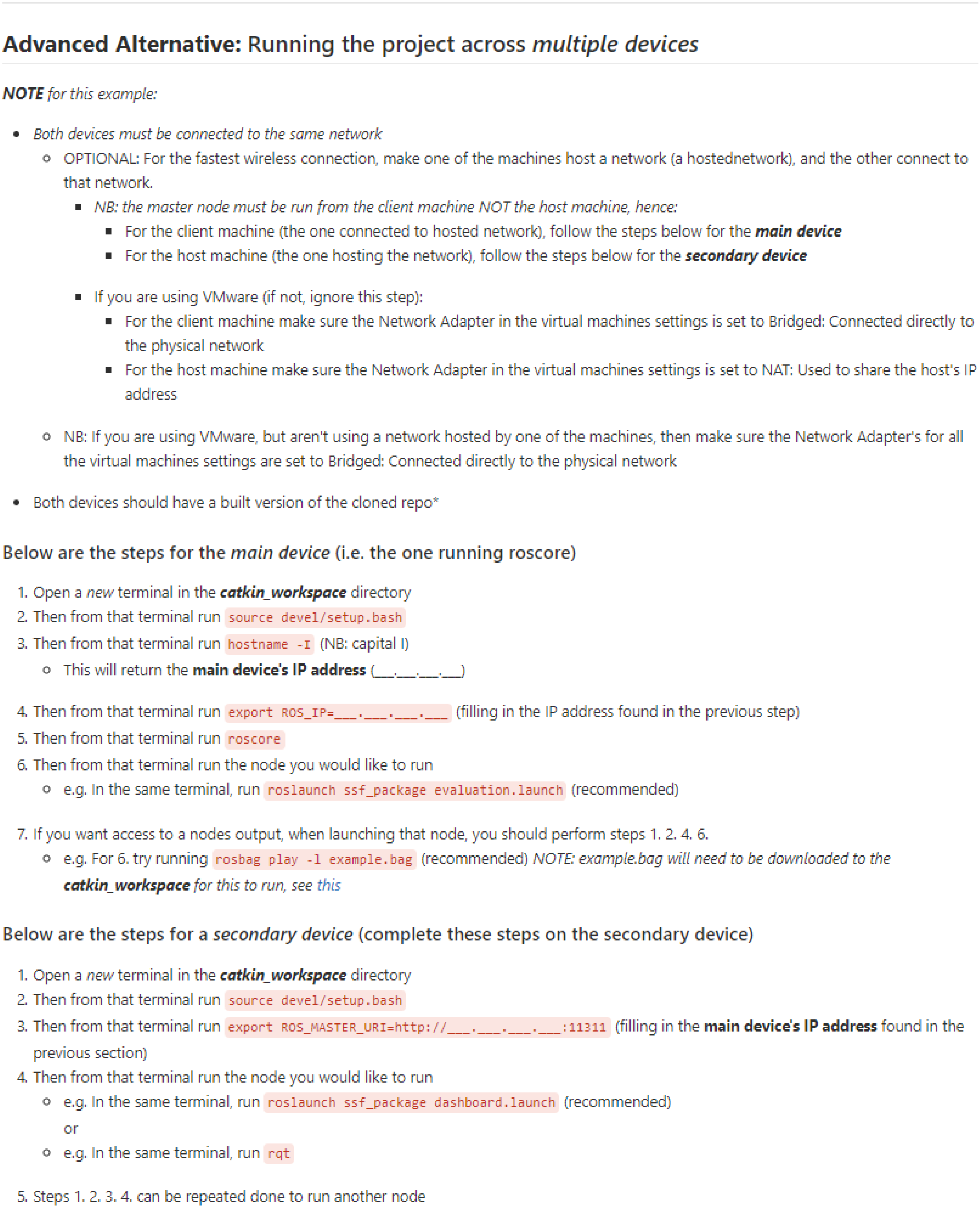
Playback (of a ROS .bag file)

1. Download a test recording (i.e. a *.bag* file) to the directory of your choice
   - Test recordings can be found here
2. Open a *new* terminal in the directory used above
   - This directory should now contain the *.bag* file
3. Then from that terminal run `rosbag play -l example.bag` (assuming the recordings name was example.bag)
   - If you don't want the playback to loop, simply run `rosbag play example.bag`

Creating a recording (which is saved to a .bag file)

This is section describes how to make a recording of currently running ROS topics

1. Open a *new* terminal in the directory you would like to save the recording
2. To get a list of the avalable ROS topics:
   - In the same terminal, run `rostopic list`
3. To start recording the topics, in the same terminal, run `rosbag record topic_name_x topic_name_y topic_name_etc`
   - You can record as many topics as you like, in the above example, the data from 3 topics are being recorded, topics topic_name_x topic_name_y topic_name_etc
   - For example, if you wanted to record the depth and rgb info from a RealSense D435 or D415 (for which the relevant topics are: */camera/depth/image_ct_raw* and */camera/color/image_raw*), you would run `rosbag record /processed_color_image /processed_depth_image`
4. To stop the recording press **ctrl + c**
   - The recording of the chosen topics will be saved in a .bag file in the current directory, with the file name being a timestamp (e.g. *2018-09-13-23-59-01.bag*).



Advanced Alternative: Running the project across *multiple devices*

NOTE *for this example:*

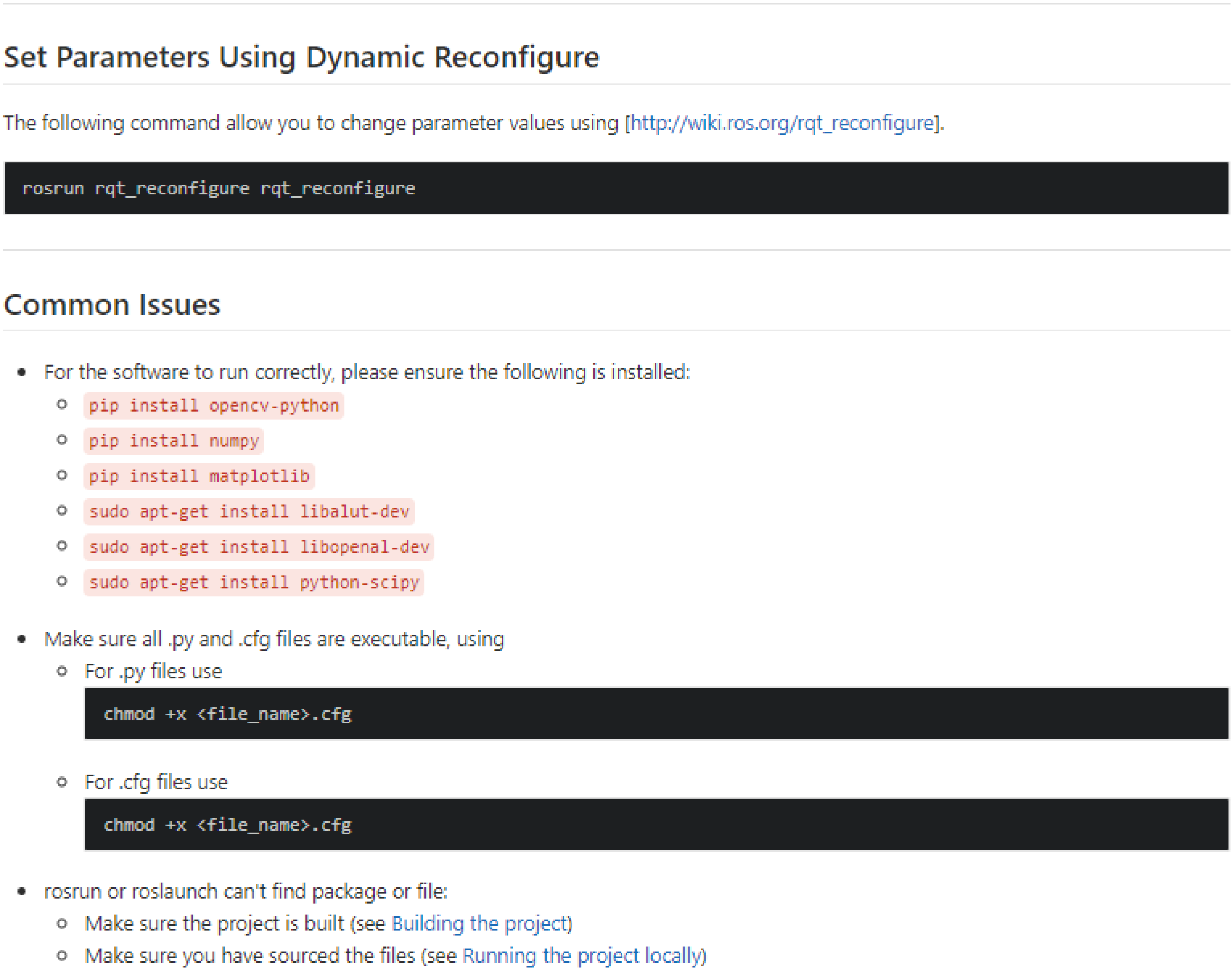
- *Both devices must be connected to the same network*
  - OPTIONAL: For the fastest wireless connection, make one of the machines host a network (a hostednetwork), and the other connect to that network.
    - *NB: the master node must be run from the client machine NOT the host machine, hence:*
      - For the client machine (the one connected to hosted network), follow the steps below for the ***main device***
      - For the host machine (the one hosting the network), follow the steps below for the ***secondary device***
    - If you are using VMware (if not, ignore this step):
      - For the client machine make sure the Network Adapter in the virtual machines settings is set to Bridged: Connected directly to the physical network
      - For the host machine make sure the Network Adapter in the virtual machines settings is set to NAT: Used to share the host's IP address
  - NB: If you are using VMware, but aren't using a network hosted by one of the machines, then make sure the Network Adapter's for all the virtual machines settings are set to Bridged: Connected directly to the physical network
- Both devices should have a built version of the cloned repo*

Below are the steps for the *main device* (i.e. the one running roscore)

1. Open a *new* terminal in the ***catkin_workspace*** directory
2. Then from that terminal run `source devel/setup.bash`
3. Then from that terminal run `hostname -I` (NB: capital I)
   - This will return the **main device's IP address** (___.___.___.___)
4. Then from that terminal run `export ROS_IP=___.___.___.___` (filling in the IP address found in the previous step)
5. Then from that terminal run `roscore`
6. Then from that terminal run the node you would like to run
   - e.g. In the same terminal, run `roslaunch ssf_package evaluation.launch` (recommended)
7. If you want access to a nodes output, when launching that node, you should perform steps 1. 2. 4. 6.
   - e.g. For 6. try running `rosbag play -l example.bag` (recommended) *NOTE: example.bag will need to be downloaded to the* ***catkin_workspace*** *for this to run, see this*

Below are the steps for a *secondary device* (complete these steps on the secondary device)

1. Open a *new* terminal in the ***catkin_workspace*** directory
2. Then from that terminal run `source devel/setup.bash`
3. Then from that terminal run `export ROS_MASTER_URI=http://___.___.___.___:11311` (filling in the **main device's IP address** found in the previous section)
4. Then from that terminal run the node you would like to run
   - e.g. In the same terminal, run `roslaunch ssf_package dashboard.launch` (recommended)
     or
   - e.g. In the same terminal, run `rqt`
5. Steps 1. 2. 3. 4. can be repeated done to run another node



Set Parameters Using Dynamic Reconfigure

The following command allow you to change parameter values using [http://wiki.ros.org/rqt_reconfigure].

```
rosrun rqt_reconfigure rqt_reconfigure
```

Common Issues

- For the software to run correctly, please ensure the following is installed:
  - `pip install opencv-python`
  - `pip install numpy`
  - `pip install matplotlib`
  - `sudo apt-get install libalut-dev`
  - `sudo apt-get install libopenal-dev`
  - `sudo apt-get install python-scipy`
- Make sure all .py and .cfg files are executable, using
  - For .py files use
    ```
    chmod +x <file_name>.cfg
    ```
  - For .cfg files use
    ```
    chmod +x <file_name>.cfg
    ```
- rosrun or roslaunch can't find package or file:
  - Make sure the project is built (see Building the project)
  - Make sure you have sourced the files (see Running the project locally)