Active Vision-Based Guidance with a Mobile Device for People with Visual Impairments

by

Jacobus C. Lock

Dissertation presented for the degree of Doctor of Philosophy in Computer Science in the School of Computer Science at The University of Lincoln

Supervisor:	Dr. N. Bellotto
Co-supervisor:	Dr. G. Cielniak

April 2020

The financial assistance of Google Inc. towards this research is hereby acknowledged. Any opinions expressed in this document are those of the author and can not necessarily be attributed to Google.

Declaration

By submitting this dissertation electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by The University of Lincoln will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: July 1, 2020

Abstract

Active Vision-Based Guidance with a Mobile Device for People with Visual Impairments

JC Lock

School of Computer Science,
University of Lincoln,
Brayford Way, Brayford Pool,
Lincoln LN6 7TS, United Kingdom.

Dissertation:

April 2020

The aim of this research is to determine whether an active-vision system with a human-in-the-loop can be implemented to guide a user with visual impairments in finding a target object. Active vision techniques have successfully been applied to various electro-mechanical object search and exploration systems to boost their effectiveness at a given task. However, despite the potential of intelligent visual sensor arrays to enhance a user's vision capabilities and alleviate some of the impacts that visual deficiencies have on their day-to-day lives, active vision techniques with human-in-the-loop remains an open research topic. In this thesis, an active guidance system is presented, which uses visual input from an object detector and an initial understanding of a typical room layout to generate navigation cues that assist a user with visual impairments in finding a target object. A complete guidance system prototype is implemented, along with a new audio-based interface and a state-of-the-art object detector, onto a mobile device and evaluated with a set of users in real environments. The results show that an active guidance approach performs well compared to other unguided solutions. This research highlights the potential benefits of the proposed active guidance controller and audio interface, which could enhance current vision-based guidance systems and travel aids for people with visual impairments.

Acknowledgements

I would like to thank my supervisors, Dr. Nicola Bellotto and Dr. Grzegorz Cielniak, for their sound guidance and advice throughout my PhD. I would also like to thank all of the participants that volunteered for the experiments we conducted, as well as the people that were so helpful in the recruitment process, with a special thanks extended to Malcolm Swinburn, from the South Lincolnshire Blind Society, and Allison Williams, from the Lincolnshire Sensory Services. I would like to thank all of my colleagues who deepened my knowledge and made me a more well-rounded researcher, as well as my friends who make me a more well-rounded person. Finally, I would like to acknowledge and thank Google for their financial support for this research.

Dedications

Vir Christian en Jan-Louis, my grootste aanhangers wie hopelik nog trots is op my, en my ouers, Koos en Elzahn, wie my nuuskierigheid aangemoedig het en my gedryf het na nuwe hoogtes.

To Emma, my English Rose.

Contents

Declaration i

Abstract ii

Acknowledgements iii

Dedications iv

Contents v

List of Figures vii

List of Tables x

Nomenclature xi

1 Introduction 1
1.1 Motivation 1
1.2 Research Problem 2
1.3 Proposed Solution 3
1.4 Research Contributions 5
1.5 List of Publications 5
1.6 Thesis Layout 6

2 Relevant Work 8
2.1 Electronic Travel Aids 8
2.2 Human-Machine Interfaces for ETAs 13
2.3 Active Vision Systems 15
2.4 Background Concepts and Methods 17
2.5 Conclusion 23

3 Active Guidance System Design 24
3.1 System Description 24
3.2 Active Guidance 27
3.3 Human-Machine Interface Module 28

3.4 Guidance Control Module 29
3.5 Conclusion 30

4 Audio-based User Interface 32
4.1 Hardware Components 33
4.2 Audio Interface Design 34
4.3 Experiments 36
4.4 Results 41
4.5 Conclusion 53

5 Visual Target Search 54
5.1 Active Vision System 55
5.2 Human Control Module 56
5.3 Experiments 64
5.4 Results 67
5.5 Conclusion 72

6 Mobile Guidance System 73
6.1 Extending the Human Control Module 74
6.2 Object Detector 77
6.3 Experiments 80
6.4 Results 83
6.5 Conclusion 90

7 Conclusions 91
7.1 Summary of Research Contributions 91
7.2 Current Limitations 92
7.3 Future Research Opportunities 93

Appendices 95

A 96

B Guidance App Implementation 98

List of References 105

List of Figures

1.1 A typical feedback control loop. 3
1.2 A visual representation of the proposed system. 4
1.3 A visual representation of the system's internal structure. 4

2.1 An example of a colour marker. 9
2.2 Pictures of the GuideCane and the Drishti prototype. 10
2.3 Images of some of the SLAM-based guidance systems discussed here. 12
2.4 Figures of showing some of the ETAs and their guidancee interfaces discussed here. 15
2.5 A figure showing an exmaple of a recycling example robot. 18
2.6 An example of a MDP modelling a cliff traversal robot. 20
2.7 The network diagrams of the SSD and YOLO networks. 23

3.2 The latest Tango-enabled mobile device. 26
3.3 The original Tango developer kit. 26
3.4 The AfterShokz bone-conduction headphones. 26
3.5 A high-level block diagram representation of the guidance system. 27

4.1 A person with visual impairments during an experiment. 33
4.2 A schematic of the different components of the proposed audio interface. 35
4.3 One of the pitch gain function used to convey the target's elevation angle. 36
4.4 A diagram of the individual system components and their communication pipelines. 37
4.5 Histograms of the participants' guesses of the tone locations. . . . 42
4.6 Histograms of the participants' guesses of which tone was higher pitched. 43
4.7 Distributions of the median cut-off frequency thresholds. 44
4.8 Histogram distributions of the participants' 75% cut-off thresholds. 45
4.9 Distributions of the angular errors in the pan and elevation dimensions for the three different pitch gradient settings. 46
4.10 Box-plots of the median pan and elevation errors for each setting. . 47

4.11 Distributions of the absolute angular errors in the pan and elevation dimensions for the three different pitch gradient settings. 48
4.12 Plots showing the Fitts relationship between the time it took the participants to find a target and as a function of the target's index of difficulty. 50
4.13 Box plots showing the participants' indices of performance. 51

5.1 A diagram of the internal structure of the proposed guidance controller. 55
5.2 An example of an action policy that the MDP generates to guide a user. 57
5.3 Examples of office environments taken from the OpenImage dataset. 61
5.4 Examples of the spatial relationships between the desk, keyboard and mouse objects. 62
5.5 A figure showing the 6 × 6 grid that simulates the controller's world. 63
5.6 A screenshot of the Android app with the guidance controller implemented. 65
5.7 A picture of the environment used for the experiments and a schematic grid representation of the environment. 66
5.8 The cumulative distribution of the participants' number of waypoints generated to find a target object. 69
5.9 The TAR for each of the objects within $\mathbf{\Omega}$. 70
5.10 The cumulative distribution of the participants' time taken to find a target object. 71
5.11 Box plots of the mean angular and linear displacements for the target search experiment. 71

6.1 The system control loop, including the HMI, guidance controller and feedback noise, ε. 73
6.2 A sample of the 'office' dataset, showing pictures taken from two different points of view. 79
6.3 The internal pipeline of the guidance system showing the flow of information. 79
6.4 Graphical representations of the environment layouts used in the guided and unguided experiments. 81
6.5 A participant with visual impairments during an experiment run. . 83
6.6 A set of box plots showing the TAR results for Group *G1* and *G2* for the guided and baseline experiments. 85
6.7 Cumulative distribution functions of the time to target results for each group and experiment. 85
6.8 Boxplots of the time to target results for each group and experiment. 86
6.9 Box plots showing the total device displacement in the pan, elevation and linear dimensions for each search experiment and group. 87

B.1 A class diagram of the object search guidance system as it is implemented in an Android app. 99
B.2 A diagram of the Metrics module responsible for recording the device's and targets' pose information. 100
B.3 A diagram of the sound generation module, responsible for generating the audio-based guidance signals. 101
B.4 A diagram of the module responsible for looking up the optimal action to execute from the policy file. 102
B.5 A diagram of the module that collects the visual information from the device's camera. 103
B.6 A diagram of the object detection module that extracts object information from a still image collected from the camera. 104

List of Tables

3.1 A summary of the design choices made for the proposed guidance system. 30

4.1 A summary of the participant demographics. 41
4.2 The average target acquisition error in the pan and elevation dimensions. 45
4.3 A summary of the *p*-values for the comparisons between the different settings. 49
4.4 The average target acquisition error in the pan and elevation dimensions for each participant group. 52

5.1 The MDP's reward function values. 58
5.2 A table containing each participant's results for the TAR, median number of waypoints and median time to target. 68
5.3 A table containing the results for the mean device displacement that was recorded during the target search experiments. 68

6.1 The reward function for the POMDP model. 76
6.2 A summary of the participant demographics. 82
6.3 A summary of the experiment results for the blindfolded and blind participants. 84
6.4 A table containing the results for each metric's performance comparison. 89

A.1 A table containing the full set of results for the experiments conducted in Chapter 6. 97

Nomenclature

Abbreviations

AI	Artificial Intelligence
API	Application Programming Interface
CDF	Cumulative Distribution Function
DCNN	Deep Convolutional Neural Network
DoF	Degrees of Freedom
ETA	Electronic Travel Aid
GPS	Global Positioning System
HMI	Human-machine Interface
HOG	Histogram of Orientated Gradients
HRTF	Head-related Transfer Function
ILD	Inter-aural Level Difference
IMU	Inertial Measurement Unit
ITD	Inter-aural Time Difference
MDP	Markov Decision Process
OCR	Optical Character Recognition
PBVI	Point-based Value Iteration
POMDP	Partially Observable Markov Decision Process
PVI	Person/People with Visual Impairments
RFID	Radio-frequency Identification
RNIB	Royal National Institute of Blind People
SARSA	State-action-reward-state-action
SIFT	Scale-invariant Feature Transform
SLAM	Simultaneous Localisation and Mapping
SSD	Single-shot Detector
SVM	Support Vector Machine
TAR	Target Acquisition Rate
TF	Tensorflow

Control Signals

G	Interface
H	Human-in-the-loop
K	Controller
P	Controlled process
e	Error (difference between output and reference)
p	Controller output
r	Reference value (i.e. desired output)
u	Process input
y	Output value
ε	An error signal

Parameters

$\mathbf{A}$	A set of possible PO/MDP actions
$\mathbf{B}$	A set of belief MDP states
$\mathbf{O}$	A POMDP observation matrix
$\mathbf{R}$	A PO/MDP reward function
$\mathbf{S}$	A set of possible PO/MDP states
$\mathbf{T}$	A PO/MDP state transition matrix
ID	Fitts's Index of Difficulty
IP	Fitts's Index of Performance
Q	The calculated quality of an PO/MDP state
V	The calculated value of an PO/MDP state
a	An action in **A**
b	A belief state in **B**
n	The number of waypoints generated
o	An observation made by the device
r	A reward from **R**
s	A state in **S**
v	Binary value to show repeated waypoint
w_e	Welford's effective target width
$\mathbf{Z}$	A set of possible POMDP observations

$\mathbf{\Omega}$	The encoded objects encoded into the PO/MDP
α	The PO/MDP learning learning rate
γ	The PO/MDP discount factor
δ	Standard deviation
π	A policy produced by an MDP
ρ	The reward function over the POMDP belief states
τ	The POMDP belief state transition function
ζ	An observation in $\mathbf{Z}$

Variables

f	Audio frequency	[Hz]
t	Time	[s]
θ	Elevation angle	[rad]
ϕ	Pan Angle	[rad]

Chapter 1

Introduction

1.1 Motivation

Worn or handheld cameras could potentially enhance or even replace human vision, which would particularly benefit people with visual impairments (PVI). Members of this group rely on a number of aids, such as a walking cane and optical character recognition (OCR) devices, to help them navigate and interact with the world. A mobile device is a good platform to integrate some of these aids onto, especially with the increasing amount of support for accessibility from major device and software manufacturers, such as Apple[1] and Google[2]. The UK's Royal National Institute of Blind People (RNIB) have also prioritised solutions enabling PVI to more effectively use common services, such as public transport and cellphones (RNIB, 2016). OCRs have already been implemented onto mobile phones and researchers have experimented with a number of methods to enable low-vision users to exploit existing navigation tools, such as Google Maps. People with limited or no vision are very adept at navigating within well-structured and familiar environments, such as their own home, but have issues with navigating in unfamiliar and dynamic environments (e.g. a street with pedestrians, a new shop, etc.) (Quinones *et al.*, 2011; Passini & Proulx, 1988). For unfamiliar environments, they often rely on assistance from a friend or carer. An estimated half a billion people worldwide live with mild to severe sight impairments or with total blindness and this number is expected to drastically rise with the ageing population (Bourne *et al.*, 2017). A significant amount of people with visual impairments would therefore benefit from a solution that would allow them to navigate in unknown environments more independently.

Macro-navigation tools, such as car SatNav systems and Google Maps, have largely been adapted to use audio signals to guide a user. Indeed, this has been helpful to people with and without visual impairments. However, replicating

[1] `https://www.apple.com/accessibility/`

[2] `https://support.google.com/accessibility/android/answer/6006564?hl=en`

such a guidance approach in an indoor environment remains a challenge. One approach is to augment or effectively replace a user's vision with a handheld mobile camera that can tell the user what it is seeing. Such an approach allows the user to make their own decisions on where to point the device to find a target object or landmark, based on the device's feedback. However, there is immense variance in building and room layouts, which makes this an inefficient and unreliable method to find the desired object. This is because without an external guidance component, the user would be forced to randomly scan an unknown environment with the device camera until the desired object or landmark falls within the camera's view. Therefore, the research in this thesis addresses the need for an automatic mobile guidance system that is able to use the surroundings' information to provide a PVI with instructions to lead them to the target object in a reasonable amount of time.

1.2 Research Problem

The work in this thesis largely contributes to the body of research in active vision. Aloimonos *et al.* (1988) and Bajcsy (1988) describe active observers as agents that are able to control their sensory apparatus such that the readings' quality are improved. In a computer vision context, this means that a camera is able to control its own geometric parameters or viewing direction (with pan/elevation actuators, for example) such that it maximises the information it gathers and its environmental understanding. Active vision techniques have been widely implemented in various forms, particularly in the fields of machine vision and robotics, where a system is tasked with manipulating its sensors to achieve some task (e.g. object detection and classification, grasping, etc.).

The classic active vision paradigm can be thought of as a simple feedback control loop, where the controller generates a signal for an electro-mechanical actuator to manipulate the camera's orientation and drive its output to some reference input (see Figure 1.1 for an example of a typical control loop). However, a challenge arises when the electro-mechanical actuator is replaced with a human in the loop. In this case, the controller has to generate a signal to control the human's actions *in addition to* the camera's viewing direction. A simple analogy is that the human replaces the pan/elevation actuator and manipulates the camera according to the controller's output. However, the added complexity of human-in-the-loop is non-trivial and requires a careful design of the controller, such that it takes human variability and performance inconsistencies into account. Furthermore, the actual control signal must be transmitted to the human through an interface that maximises the amount of information they can extract from it. The current state-of-the-art does not provide an adequate answer for the question: "can active vision techniques be applied to a system with a human in the control loop?"

The research presented in this thesis addresses this question by implement-

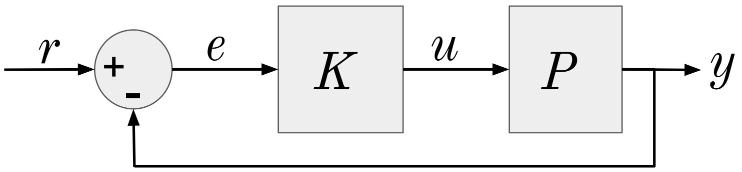


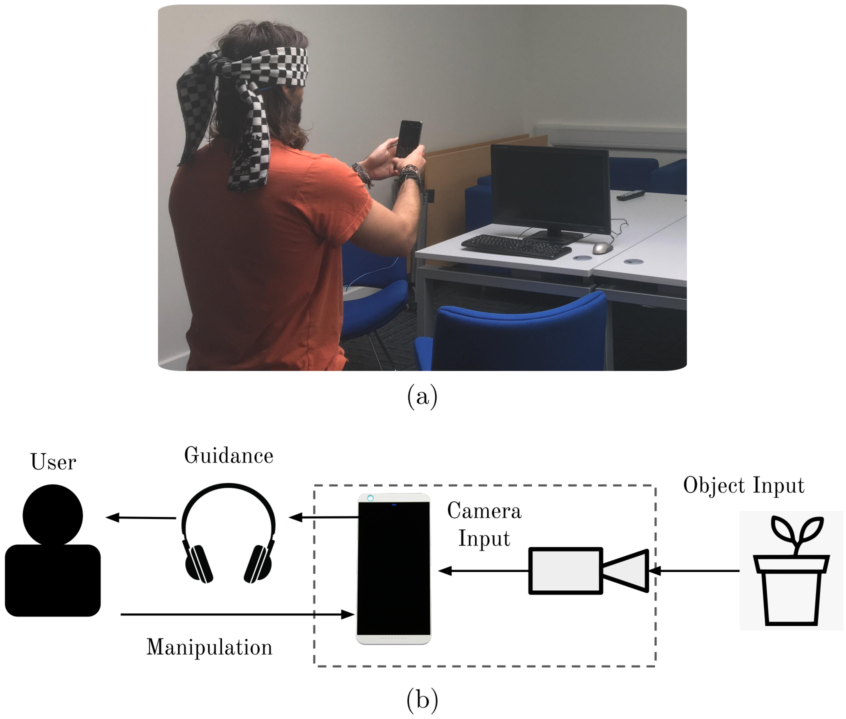
Figure 1.1: A typical feedback control loop, including the reference, error, control and output signals (r, e, u, y respectively), and the controller (K) manipulating some process (P).

ing a vision-based object detection system for a hand-held camera that *actively* guides the user towards the desired object. Such a system can be used by people with visual impairments to find objects or regions of interest in unknown indoor environments, such as chairs or doors in an office. In particular, this research consists of developing an effective audio interface and a controller that is able to perform vision-based object search with a tablet or smartphone camera in real-time, and to experimentally evaluate the proposed solutions with groups of blindfolded participants, as well as participants with visual impairments.

1.3 Proposed Solution

As shown in Figure 1.2, the solution to the research problem addressed in this thesis is based on a mobile device with a colour camera and a set of gyroscopes and accelerometers that are able to track the device's 6-dimensional movements. Furthermore, pan/elevation actuation is performed by a human-in-the-loop that manipulates the mobile device. While the user points the device at different locations, camera images are collected to detect and classify any observed objects. These observations then update an internal model of the environment, which the system uses to select a new location for the user to explore, maximising the probability of finding the target object. The coordinates of the new locations are communicated to the user by an audio-based human-machine interface (HMI). If the target object is not found, the process is repeated. The internal data flow on the device is presented in Figure 1.3.

The final system includes three separate modules, the first of which is an audio interface that provides guidance instructions for the user. The second one is the controller that selects the actual instructions based on the device's current orientation and the observed objects. Finally, the object classifier is responsible for processing camera images and passing object information to the control module. All of these components, introduced in the following chapters, are integrated into a single mobile device and run concurrently in real-time.


(a)

(b)

Figure 1.2: A visual representation of the proposed system in use (top) and as a schematic (bottom). The camera provides image data to the mobile device as it is moved and reorientated by the user, which the device then uses to generate guidance instructions in real-time. The top image was taken during an initial set of experiments with blindfolded participants (Lock *et al.*, 2019*d*).

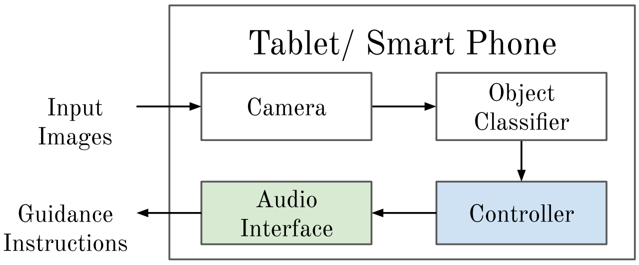


Figure 1.3: A visual representation of the guidance system's internal structure as it is integrated onto a mobile device. The device uses image data to detect and classify all objects seen by its camera. This information is then used to generate guidance waypoints, which are translated into instructions for the user.

1.4 Research Contributions

This thesis presents several research contributions. The main ones are listed here.

1. A new audio interface for people with visual impairments that uses spatialised sounds and bone-conduction headphones to provide guidance instructions without blocking other ambient sounds. The interface's effectiveness is demonstrated and evaluated in a set of experiments with a large number of participants. The solutions and design choices adopted in this research can be applied to other interfaces that use similar headphones.

2. A new probabilistic controller that generates guidance instructions for a user with visual impairments to complete an object search task. This controller uses a data-based transition model trained from a large object dataset to learn the spatial relationships between multiple objects in typical indoor environments. These inter-object spatial relationships and the method of their extraction can be used by other researchers to model an indoor environment in simple terms. The proposed solution was evaluated in simulated and real object search scenarios with blindfolded participants and participants with visual impairments.

3. A fully-functioning pipeline for active visual search that integrates a spatialised audio interface, a probabilistic guidance controller and real-time object detection to assist people with visual impairments find objects in unknown environments. The integrated system is implemented on a mobile device and evaluated against an unguided object detector that resembles other apps currently available on the market.

4. A final, technical contribution is provided by the source code of the full guidance system implementation, including the audio interface, the controller and the vision-based object detector. The app is not available on the Play store, but can be freely downloaded and compiled to run on any Android platform[3].

1.5 List of Publications

The research presented in this thesis generated the following peer-reviewed publications:

[3] https://github.com/yassiezar/POMDPObjectSearch

J. C. Lock, G. Cielniak and N. Bellotto (2017). A Portable Navigation System with an Adaptive Multimodal Interface for the Blind. In: *Proceedings of the AAAI Spring Symposium*, p395 — 400. AAAI.

J. C. Lock, I. D. Gilchrist, G. Cielniak and N. Bellotto (2019). Bone-Conduction Audio Interface to Guide People with Visual Impairments. In: *Proceedings of the International Conference on Smart City and Informization*, p542 — 553. Springer.

J. C. Lock, G. Cielniak and N. Bellotto (2019). Active Object Search with a Mobile Device for People with Visual Impairments. In: *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)*, p476 — 485. Springer.

J. C. Lock, A. G. Tramontano, S. Ghidoni and N. Bellotto (2019). ActiVis: Mobile Object Detection and Active Guidance for People with Visual Impairments. In: *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, p649 — 660. Springer.

J. C. Lock, I. D. Gilchrist, G. Cielniak and N. Bellotto (2020). Experimental Analysis of a Spatialised Audio Interface for People with Visual Impairments. Submitted to *ACM Transaction on Accessible Computing* (accepted, pending revisions).

M. Terreran, A.G. Tramontano, J.C. Lock, S. Ghidoni and N. Bellotto (2020). Real-time Object Detection using Deep Learning for helping People with Visual Impairments. Submitted to the *International Conference on Pattern Recognition (ICPR).*

1.6 Thesis Layout

The remainder of this thesis is organised as follows.

First, Chapter 2 describes the state-of-the-art in mobile guidance systems, active vision techniques, and their implementations. Various principles, concepts and frameworks used in subsequent chapters, such as Markov Decision Processes (MDP) and object detection algorithms, are also introduced here.

Chapter 3 then introduces the overall system design, including the specific concepts and ideas that the proposed solution is based on. The various modules, as well as a high-level representation of the active vision system with human-in-the-loop, are also discussed.

Following this, Chapter 4 presents the audio interface that communicates the guidance instructions to the users. This interface uses a set of bone-conduction headphones that do not interfere with ambient sound to accommodate users with visual impairments who rely on the latter. The audio interface's implementation is followed by a set of experiments conducted with blindfolded and visually impaired participants to evaluate its performance.

In Chapter 5, an MDP-based controller, which guides a human by actively generating suitable instructions, is introduced. The proof-of-concept developed in this chapter simulates an object detector with a QR code scanner and generates guidance instructions to guide a user to find a target object, represented by a QR code. The controller is implemented on a mobile phone and evaluated with a set of user-based experiments.

A full implementation of the active vision system that includes the audio interface, an improved guidance controller and a real object detector, is discussed in Chapter 6. The design of this complete guidance system is explained and then implemented in an Android app. It is finally evaluated with a set of experiments to determine its effectiveness in assisting blindfolded and visually impaired participants in an object-search task.

The thesis concludes with Chapter 7, which summarises the research contributions and their current limitations. Open research questions and possible avenues for future research are also discussed.

Chapter 2

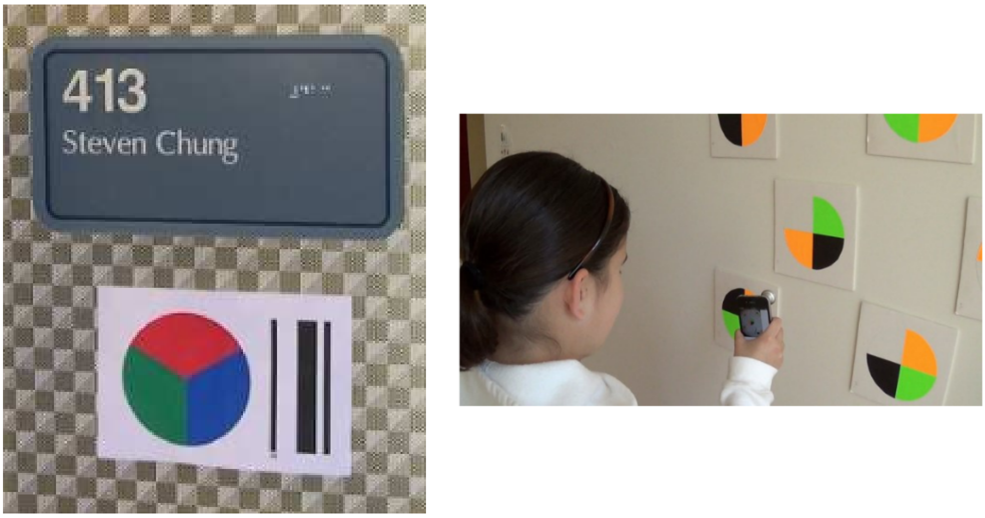
Relevant Work

Much research has been done in the past regarding different guidance systems for people with visual impairments, as well as different human-machine interfaces and active vision strategies. Furthermore, different AI-based decision algorithms and object detection networks have become well-established in the literature and are suitable for the work conducted in this research. This chapter presents the state-of-the-art in these areas, providing a detailed review of the literature in the first three sections on electronic travel aids, their user interfaces and relevant active vision systems, followed by an overview of some AI concepts and methods used in later chapters of this thesis.

2.1 Electronic Travel Aids

Navigation aids are a common tool for people with visual impairments to help them navigate through the world more independently and increase their autonomy. The white walking cane has become a near universal aid to provide early warning of upcoming obstacles within the user's vicinity. With improvements to embedded sensors and digital technology, researchers have proposed newer solutions for safer navigation and obstacle avoidance experiences by implementing electronic alternatives — so-called electronic travel aids (ETA).

A simple ETA solution consists of installing electronic or visual markers in the environment, and then use a device that can easily find and extract useful information from those markers. Information such as the distance to a dangerous object (e.g. open manhole), the direction to different locations, and upcoming changes to the environment (e.g. stairs) could be very useful to people with visual impairments. Many different types of markers have been proposed and tested, including Bluetooth (Kim *et al.*, 2016; Ahmetovic *et al.*, 2016), RFID (Faria *et al.*, 2010; Ramón *et al.*, 2012; Willis & Helal, 2005) and visual markers, which includes both 2D barcodes (Iannizzotto *et al.*, 2005; Nicholson *et al.*, 2009) and colour markers (Coughlan *et al.*, 2006; Al-Khalifa, H.S., 2008). All of these are available as stickers and can quickly be applied in

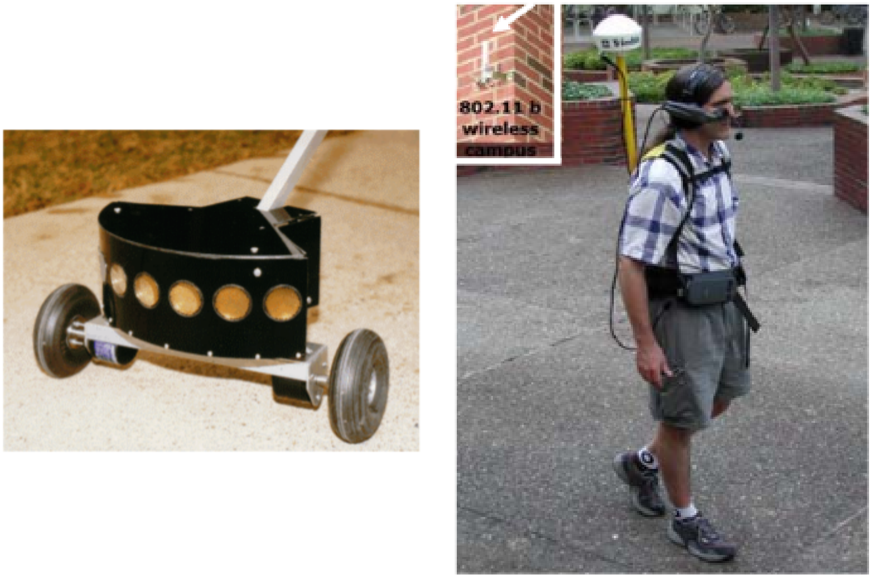


(a) (b)

Figure 2.1: An example of a colour marker described by Coughlan *et al.* (2006) (left) and similar markers being used in an experiment (right) (Manduchi & Coughlan, 2014).

different locations (see Figure 2.1 for examples of vision-based colour markers). Bluetooth and RFID markers work well beyond the range of a typical walking cane and are suitable for passive applications where a user is informed of a nearby marker's content without actively prompting it. Conversely, barcodes and visual markers need to be in a camera's line-of-sight, making it harder to use. However, all of these systems suffer from the same issue of maintenance: while it is easy to label the environment with these simple markers, it is time-consuming and potentially costly to manually update and replace them when their information becomes out of date. This issue is amplified for very dynamic environments.

Early attempts at improving the existing guidance systems for people with visual impairments involved equipping the walking cane with an array of sensors that can detect upcoming obstacles and either warn or help the user to avoid them. For example, the GuideCane and BatCane proposed by Borenstein & Ulrich (1997) (see Figure 2.2a), and Hersh & Johnson (2008), respectively, are wheeled canes equipped with sonars that scan the front of the device for oncoming obstacles. The GuideCane is also equipped with a set of motor controllers that turns the cane's wheels to avoid such obstacles. The Drishti system, implemented by Ran *et al.* (2004) (shown in Figure 2.2b), and the iSonar, proposed by Vorapatratorn & Nambunmee (2014), are two other worn systems using embedded sonars to detect and warn the user of upcoming obstacles. All of these systems performed quite well in their experimental settings, but their bulkiness have proven to be major hurdles for large-scale adoption. This issue is evident in the surveys conducted by Golledge *et al.* (2004) and Yusif *et al.*



(a) (b)

Figure 2.2: Pictures of the GuideCane (Borenstein & Ulrich, 1997) (left) and the Drishti prototype (Ran *et al.*, 2004) (right).

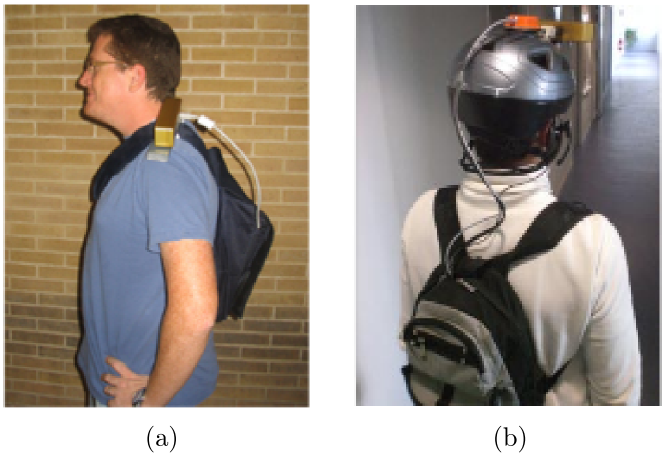
(2016), which show that people with visual impairments are concerned about "social stigma". Indeed, they "strongly agree" that, even if a system worked well, they would not use it if it affected their public appearance (Golledge *et al.*, 2004).

An interesting and more recent approach to ETAs has been the use of mobile phones as walking canes to perform sensing and warning tasks. For example, the virtual cane concept proposed by Vera *et al.* (2014) replaces the walking cane with a laser pointer and a smartphone. In this case, the laser pointer acts as a walking cane and the mobile device vibrates depending on the distance to the pointed at obstacle. This distance is calculated as a function of the time the reflected laser beam takes to reach the mobile phone's camera. Mocanu *et al.* (2016) proposes a device that combines a mobile phone and its camera with a sonar. The resulting object detection device is able to not only detect any oncoming obstacles, but also robustly classify objects and appropriately warn the user of potential danger (e.g. slow moving pedestrian requires less intense warning compared to a fast moving car). These recent mobile device-based systems are a step in the right direction for ETAs, when compared to their bulky predecessors. However, they still only provide basic obstacle avoidance functionality and are unable to guide a user towards a pre-selected destination or location.

Other researchers have improved upon previous designs by exploiting prior knowledge of the environment to not only improve obstacle detection, but also provide intelligent guidance instructions to a destination that potentially avoids obstacles entirely. Indeed, an updated version of the Drishti system uses GPS to direct a user along a predefined side walk (Ran *et al.*, 2004). A simple solution to track a user is to equip them with an inertial measurement unit (IMU) and compass. This movement can then be used to localise the user within a predefined indoor or outdoor map that may be pre-programmed to contain all of the environment's landmarks, features and potential obstacles. The system proposed by Hesch & Roumeliotis (2010) uses a set of WiiMotes and their built-in IMUs and infrared sensors to localise the user, while the TANIA system proposed by David *et al.* (2014) uses a dedicated IMU sensor to do the same. The former is able to scan for familiar corridor corners (used to correct drift), while the latter is able to detect changes to the user's movement (e.g. walking, stumbling, etc.) and adjust the position estimate accordingly. This process is similar to the one used in the system of Apostolopoulos *et al.* (2012), where the authors used a mobile phone's built-in IMU to localise a user, while continuously adjusting for the user's step length to minimise drifting errors. It is also capable of generating the shortest path to the target destination and guide the user accordingly. These systems use innovative processes to maximise usability in terms of cost, complexity and size, while minimising localisation errors caused by sensor drift. However, these solutions are bound to known and mapped environments and it is unclear how they can be applied in unknown ones.

A possibly more robust solution is to use visual data from the sensor array to build a map of the environment in real-time, which the guidance system can use to generate navigation instructions. The works by Sáez & Escolano (2008), Rodríguez *et al.* (2012) (pictured in Figure 2.3a), Schwarze *et al.* (2015), Pradeep *et al.* (2010), and Katz *et al.* (2012) (showed in Figure 2.3b) all use so-called simultaneous localisation and mapping (SLAM) techniques to build a 3D map of the environment, which is used to generate instructions for the user to safely move through it. The authors of these works report encouraging results, although their experiments did not include a sufficient number of participants. Lee & Medioni (2015), instead, conducted a set of experiments with both blindfolded and visually impaired participants. In addition to building a 3D map of the environment with no prior information, their proposed system creates a 2D probabilistic occupancy map for efficient traversability analysis and path planning. In their experiments with four participants with visual impairments, they report a 47% improvement to the participants' mobility when the guidance system was used alongside a white cane, suggesting that a hybrid solution could be useful. These vision-based systems are potentially helpful in unknown and dynamic environments, but it is not clear how scalable such solutions are for mapping and storing many different areas.

A different approach to the guidance problem is to implement an efficient object or landmark-detection system, which acts as the user's eyes and helps to identify the user's current location (e.g. which room). This feedback can also be used to provide guidance instructions based on the relative position of



(a) (b)

Figure 2.3: Images of the SLAM-based guidance systems from Rodríguez *et al.* (2012) (left) and Katz *et al.* (2012) (right).

an object. For example, the systems from Schauerte *et al.* (2012), Tian *et al.* (2013), and Ou *et al.* (2020) use a number of different object detection techniques to classify various landmarks and objects, such as doors, pedestrian crossings, stairs, etc., and provide their relative positions through different feedback modalities. Although their results are promising, they lack substantial experimental data of participants with visual impairments. Vázquez & Steinfeld (2012), instead, have conducted experiments with a reasonable number of participants with visual impairments, testing a system that helps them take 'good' pictures with a camera using audio guidance signals. Their results with participants with healthy eyesight, as well as partially and totally blind participants, showed a strong improvement when guided by audio tones and vocal instructions. The Headlock system, presented by Fiannaca *et al.* (2014), uses feature detection algorithms to search for known landmarks in an indoor environment (e.g. door) and then guides the user towards them using audio cues. It is designed for the Google Glass platform and used alongside a white cane that is used to detect any immediate floor-level obstacles. These audio cues are continuously updated as a function of the user's heading to minimise veering from the optimal path. Compared to using a white cane only, the authors found that participants were able to find a doorway faster and with less veering. Indeed, when using only the cane, the participants relied the wall to lead them to the door, as opposed to moving directly to the latter with the Headlock guidance system. The VizWiz::LocateIt system from Bigham *et al.* (2010) uses a different approach. Instead of performing object detection on the device, the user sends a picture of the work area to an external Amazon Mechanical Turk worker, who visually searches for the target object. The worker then sends instructions to the blind user on how to reach the target object. In their experiments, the authors found that participants managed to find the correct objects in approximately 92s, which is similar to the performance of a barcode-based approach used as baseline.

The strengths of object detection-based guidance systems are numerous. Firstly, they can be implemented onto modern mobile and wearable devices, requiring minimal additional hardware (e.g. a set of headphones) to guide a user. Secondly, object detection is an active research topic in computer vision and it is likely that performance and efficiency of these algorithms will increase over time. Furthermore, no modification or annotation of the user's environment are required, as long as the key objects are clearly visible. Indeed, Kunhoth *et al.* (2019) found that their object detector-based guidance approach outperformed a Bluetooth marker-based approach by more than 30%. However, the major drawback to all of the systems described here are that they are *passive* and rely on the user to find the target object with the camera, before they can actually be guided towards it. This process could potentially be improved by using other non-target objects as an additional input to the guidance system, but further research into such systems is needed.

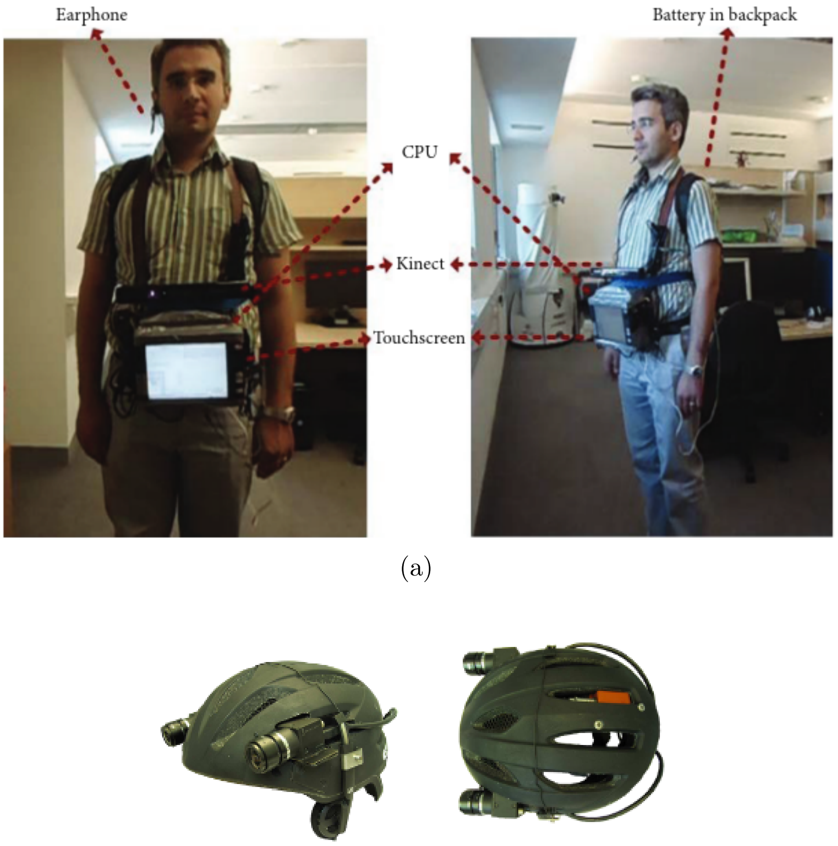
2.2 Human-Machine Interfaces for ETAs

An important part of any system involving human interaction is an effective human-machine interface (HMI). In the case of a guidance system for people with visual impairments, an HMI will be responsible for generating simple signals that a user can easily and accurately interpret to execute an action and move in the right direction. Popular feedback media are vibration, sound and voice commands, each with their own set of advantages and drawbacks.

Two surveys were conducted by Golledge *et al.* (2004), and Arditi & Tian (2013) to gauge the HMI media preferences of people with visual impairments. In particular, Golledge *et al.* (2004) found that their participants had a strong preference for vocal feedback, followed by tonal feedback and then vibration. They also prefer non-covering or single headphones (e.g. a single in-ear headphone) instead of headphones that fully cover their ears. In the other survey, Arditi & Tian (2013) found a similar strong preference for vocal feedback, followed by tonal and vibration feedback. They also show that people prefer to prompt the system for feedback, rather than receiving constant feedback.

Many of the ETAs discussed in Section 2.1 have implemented some form HMI using either one or a combination of vibration, tonal and vocal feedback media. For example, the systems from Mocanu *et al.* (2016), Chessa *et al.* (2016) and Kanwal *et al.* (2015) rely on vocal feedback, while the ones from Schwarze *et al.* (2015), Rodríguez *et al.* (2012) and Katz *et al.* (2010) use pure audio signals (see Figure 2.4 for images of the systems proposed by Kanwal *et al.* (2015), and Schwarze *et al.* (2015)). Instead, the systems from Rivera-Rubio *et al.* (2015), Lee & Medioni (2015) and Xiao *et al.* (2015) use vibro-tactile haptic feedback. All of these works have shown that their guidance systems, including their HMIs, are capable of guiding a participant to a target location. However, given the survey results from Golledge *et al.* (2004), and Arditi & Tian (2013), vibro-tactile feedback is not a desirable modality and requires a significant amount of extra hardware to achieve a reasonable level of guidance resolution. Vocal feedback seems to be the best modality among the three. However, guiding a person to a small target object would require high-resolution guidance instructions and many device adjustments, creating significant cognitive load for the user. In this regard, tonal feedback would be more desirable, given its reduced level of cognitive effort compared to vocal feedback (Klatzky *et al.*, 2006), although some care must be taken to avoid unpleasant tones and listener fatigue. Furthermore, these tones may be less obvious to understand than vocal commands, so some training would be required to work effectively. One extreme example is ‘The Voice’, a system that uses a set of sinusoidal sound waves with different frequencies and amplitudes to translate camera images and requires more than 15 hours of training to be used effectively (Ward & Meijer, 2010). However, totally blind users, who have used the system for long periods of time, have been able to perceive depth, edges, movement and even colour.

To reduce the amount of time required to use an assistive system effectively, researchers have investigated more intuitive methods that convey guidance instructions via simple audio tones. One possibility is to apply a head-related transfer function (HRTF) to spatialise an audio signal (Xie, 2013). In other words, a monaural sinusoidal audio wave can be mathematically transformed into a binaural signal, making a listener believe that the sound originates from some arbitrary 3D position. This position is fully controlled by the HRTF parameters, exploiting humans’ natural hearing pathways and mechanisms (more details can be found in Section 4.2). However, each person’s hearing characteristics are unique, which makes it difficult to create a truly generic HRTF with consistent localisation performance. Nevertheless, there are widely used models, such as the MIT’s KEMAR mannequin (Gardner & Martin, 1995), that are based on a generic human frame and over-ear headphones. Several researchers, such as Geronazzo *et al.* (2016), Wilson *et al.* (2007), Katz *et al.* (2010), and Blum *et al.* (2013), have implemented some of these spatialisation techniques in their interfaces to provide the user with navigation waypoints, with good results when applied to over-ear headphones or external speakers. However, deviating from the aforementioned HRTF constraints can significantly affect the performance of these interfaces. Indeed, research has showed that different playback devices, including in-ear and bone-conduction headphones, lead to diminished localisation performance compared to over-ear headphones (Schonstein *et al.*, 2008; Stanley & Walker, 2006). This problem seem mostly limited to the elevation dimension (i.e. vertical direction), as stated by Stanley & Walker (2006), who show that an interface with bone-conduction headphones and a well-tuned HRTF can achieve similar results to the over-ear solution.



(a)

(b)

Figure 2.4: Figures of the guidance system prototypes used in the works by Kanwal *et al.* (2015) (top) and Schwarze *et al.* (2015) (bottom), including their feedback devices (headphones).

2.3 Active Vision Systems

An active vision system is one where the observer is able to adjust its sensors to gather more information or to maximise its understanding of the environment (Bajcsy *et al.*, 2018). This is a well-understood and common occurrence in biological systems (Findlay *et al.*, 2003), where, for example, a human would determine their current location by moving their head and eyes to maximise the visual information and match it against known or familiar locations. Similarly, researchers in computer vision and robotics have incorporated some of these techniques into electro-mechanical and digital systems that change the view of a perception agent in order to gather more environmental information (Chen *et al.*, 2011), for object detection and tracking tasks, for example.

Object detectors often rely on dividing an input image into smaller windows, searching for as many objects as possible in these smaller search spaces (more details in Section 2.4.3) until the target object is found. Researchers in active vision, however, have tried more efficient object search strategies for the window selection and proposal scheme. Such an active search strategy was implemented by Gonzalez-Garcia *et al.* (2015). In their work, each of the objects detected in a window is used, alongside the current window's position, by a purpose-built classifier that outputs the most likely location containing the target object. The next window is then sampled from this location. With this approach, they report a significant reduction in the number of windows that need to be processed compared to a random window sampling approach. Caicedo & Lazebnik (2015) proposes a similar active search strategy, but their system uses a Markov Decision Process (MDP) to select the next best window, achieving significant improvements in terms of precision and recall. The state-of-the-art Faster R-CNN network (Ren *et al.*, 2015) also uses an active strategy based on a separate neural network to select windows, although they require a large number of window proposals to achieve good results.

These active object search strategies have proven useful when used on still images and have subsequently also been applied to electro-mechanical platforms that capture video data and manipulate their viewing parameters to maximise the information it gathers. Such systems include cameras manipulated by simple pan-tilt servos (Giefing *et al.*, 1992), which can possibly be implemented on more complex robotic platforms with high degrees of freedom (DoF). For example, the systems proposed by Radmard & Croft (2017) manipulates a camera attached to a static robotic arm with 7 DoF to collect visual information. Rasolzadeh *et al.* (2010), instead, use static stereo cameras with a robotic arm that manipulates the object to maximise the visual information it gathers, such as colour, object shape, etc. (also known as interactive perception).

Active vision-based search processes have also been implemented on fully mobile (i.e. wheeled) robot platforms by Shubina & Tsotsos (2010), Aydemir *et al.* (2013) and Ye *et al.* (2018*a*,*b*), for example. These robots can traverse a 3D indoor environment while searching for a target object placed in an unknown location. Each of these systems use different methods to generate the optimal actions to reach the target object, including neural networks that generate navigation policies (Ye *et al.*, 2018*a*,*b*), search space optimisation (Shubina & Tsotsos, 2010) and probabilistic search methods using partially observable MDP (POMDP) models (Aydemir *et al.*, 2013). All the robots were tested either in simulation or in real experiments. Shubina & Tsotsos (2010) showed that their optimisation process leads to a high success rate (91%) and low search time for finding the target object when partial information about the object's location is provided (e.g. on top of a table). With their POMDP-based system, Aydemir *et al.* (2013) reports clear improvement in a searching task, compared to the greedy policy, and is almost comparable to a normal human search. The difference between the human and the POMDP search performance is reduced even further when additional semantic information about the environment is provided to the system. Similar conclusions were reached by Ye *et al.* (2018*a*,*b*), significantly improving their robots' search capabilities with a neural network that generates opportune action policies.

All of these works implemented active search systems on complex electromechanical platforms, reporting significant improvements compared to passive approaches. However, as highlighted by Bajcsy *et al.* (2018), the question of whether such techniques can be useful for systems including human interactions remains an open research area.

2.4 Background Concepts and Methods

Various well-established tools and frameworks are used for this research to implement an active vision guidance system. This section introduces some state-of-the-art AI and computer vision methods for decision-making and object classification, respectively. Some of these concepts are used and implemented in Chapter 4, Chapter 5 and Chapter 6.

2.4.1 Markov Decision Processes

A Markov decision Process (MDP) is a mathematical framework that models an agent's decision-making process where the outcomes are partly random and affected by the agent's actions. The agent's goal is to reach some target state, maximising the total utility during the state transitions that take place during this process (Puterman, 2014).

MDPs are an extension of Markov Chains, which describe a stochastic sequence of state transitions where the next state transition depends only on the current and previous states. However, MDPs also include actions, which model the agent's transition choices and rewards, giving the agent some motivation to execute a specific action. A typical MDP is modelled by the tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R} \rangle$, where $\mathbf{S}$ is a finite set of states the agent can reach, $\mathbf{A}$ is a finite set of actions it can execute from a state $s \in \mathbf{S}$. $\mathbf{T}$ is an $\mathbf{S} \times \mathbf{A} \times \mathbf{S}$ transition matrix that contains the probability of transitioning from state s to state s' after executing action $a \in \mathbf{A}$. This matrix is constrained by the agent's environment (e.g. the agent cannot move through a wall). Finally, $\mathbf{R}$ is an $\mathbf{S} \times \mathbf{A}$ reward matrix that determines the reward (or punishment) an agent receives for executing action a when transitioning from state s to s'.

After the MDP's parameters have been determined — typically heuristically or through experimentation — it can be used to produce the optimal set of actions for an agent to reach a goal state, s_g, from any initial state, s_0. A state-action mapping scheme that leads the agent to the goal state is known as a policy, $\boldsymbol{\pi}$. The optimal policy, $\boldsymbol{\pi}^*$, maximises the cumulative reward the agent receives while traversing from s_0 to s_g. The $\mathbf{R}$ and $\mathbf{T}$ matrices are particularly important to find a good policy, since a policy generated with a weak $\mathbf{R}$ (a not sufficiently motivated agent) or malformed $\mathbf{T}$ (a badly modelled environment) may not converge to an optimum or not reach s_g at all. Figure 2.5

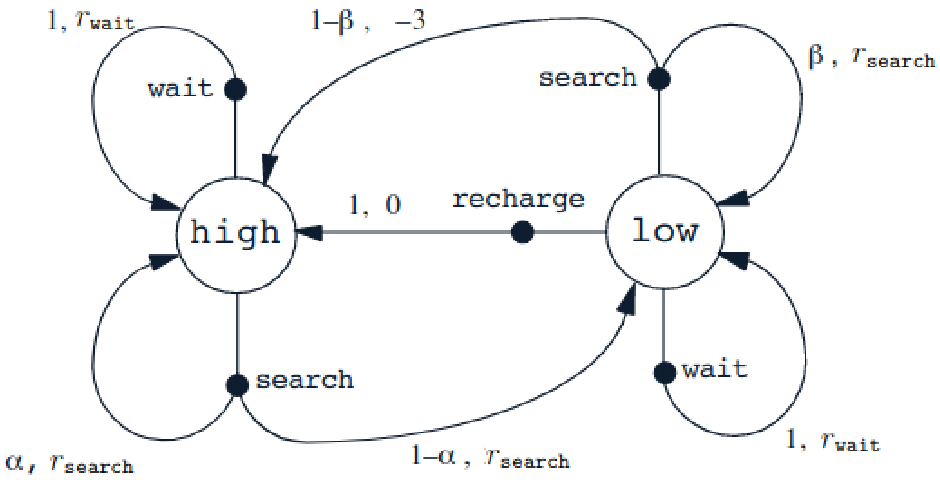


Figure 2.5: A figure depicting an example of a recycling example robot (Sutton & Barto, 1998, p. 69). It can transition between two high and low battery states with probabilities paramaterised by α and β, while being rewarded for each action it takes (search, wait, recharge or flat).

shows an example MDP (Sutton & Barto, 1998) that models a recycling robot, which can either be in a high or low battery state, and can execute a search, wait or recharge action. The transitions probabilities depend on the parameters α and β, and the rewards by r_{wait}, r_{search}, $r_{recharge}$ and r_{flat} (-3 in the figure). The optimal policy $\boldsymbol{\pi}^*$ will be highly influenced by these parameters, since the robot is more likely to execute a recharge action if there is a high probability that the battery runs flat and if the penalty for it is harsh.

The policy $\boldsymbol{\pi}^*$ can be determined using dynamic programming techniques (Bellman, 1957) by maximising the agent's expected long-term cumulative reward. There are a number of algorithms that are able to solve MDPs and find $\boldsymbol{\pi}^*$, including value iteration (Bellman, 1957), policy iteration (Howard, 1960), Q-Learning (Watkins, 1989) and state-action-reward-state-action (SARSA) (Rummery & Niranjan, 1994). For example, the value iteration algorithm solves an MDP by calculating the discounted cumulative reward that the agent can expect to receive when starting from state s_0. It then assigns a so-called value $V(s)$ for each state s it reaches:

$$V(s) = \sum_{s'} \mathbf{T}_{\boldsymbol{\pi}(s)}(s, s')(\mathbf{R}_{\boldsymbol{\pi}(s)}(s, s') + \gamma V(s')), \tag{2.4.1}$$

where $\gamma \in [0, 1.0]$ is a weighting factor that prioritises long-term or short-term reward and is typically set close to 1.0 to prioritise long-term reward. The policy $\boldsymbol{\pi}^*$ can then be determined by selecting the action that maximises a state's value:

$$\boldsymbol{\pi}^*(s) = \operatorname*{argmax}_a \sum_{s'} \mathbf{T}(s, a, s')\mathbf{R}(s, a, s') + \gamma V(s'), \tag{2.4.2}$$

In many cases, however, the environment is difficult to explicitly model, so $\mathbf{T}$ is determined by experimentation or simulation, leading to so-called "model-free" algorithms. Q-Learning is a popular reinforcement learning algorithm that 'learns' $\mathbf{T}$ through iteration. In this process, each state-action pair is given a quality score, Q, as defined by

$$Q^{new}(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q(s_{t+1}, a)), \tag{2.4.3}$$

which is iteratively updated with the action that maximises the next state's expected quality. α is the learning rate, which determines the weight of new values added to the previous quality score. After enough iterations, a state-action pair's quality, or Q-value, approaches its optimal value and can then be added to $\boldsymbol{\pi}^*$.

The SARSA algorithm is another model-free alternative to the value iteration algorithm. While similar to Q-Learning, it differers in how it updates its Q-value at each training iteration. Specifically, Q-Learning is known as an "off-policy" process, which refers to the fact that for each Q-value update during the training process, the agent selects an action that maximises the expected future value. Instead, the on-policy approach updates the Q-value based on the policy learned up to that point and can only select the action given by $\boldsymbol{\pi}(s)$. This difference is shown in the update step of the SARSA algorithm, where the maximum lookup operation is replaced by a_{t+1}:

$$Q^{new}(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1})). \tag{2.4.4}$$

Under the same conditions, the main practical difference between on- and off-policy training is the speed at which they converge to the optimal policy, with Q-Learning typically being faster. However, the SARSA algorithm is preferable in cases where the agent's performance during the training process is important, or if the problem requires a more exploratory learning approach. To illustrate this point, consider the example from Sutton & Barto (1998) in Figure 2.6, where a battery-powered robot is placed on a cliff's edge and tasked with reaching the other side without falling. Each grid cell represents a state that the robot can reach by moving either up, down, left or right towards the goal cell, G, trying to minimise energy consumption. With a negative reward punishing each movement and falling off of the cliff (-1 and -100, respectively), Q-Learning produces the optimal, shortest route that traverses the cliff as closely as possible. Instead, SARSA generates a safer path that reaches the goal further away from the cliff, but with a longer route. Naively, it would seem that Q-Learning outperforms SARSA. However, in many real-world application, for example with an expensive robot, the safer route produced by SARSA seems preferable.

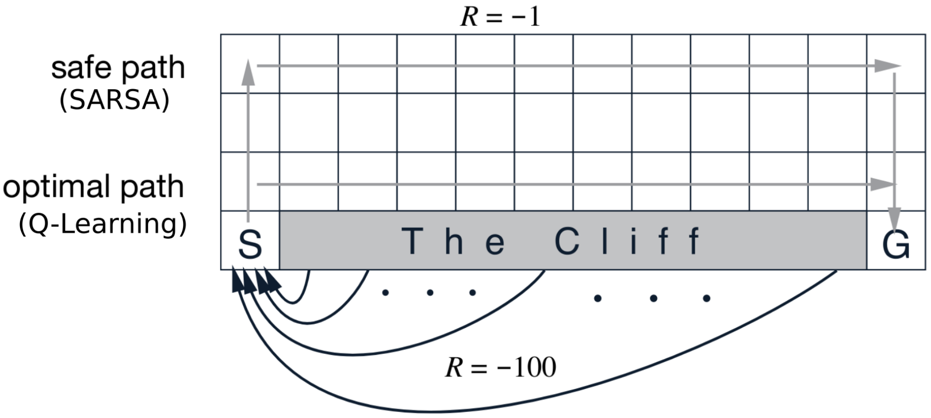


Figure 2.6: An example of an MDP modelling a simple robot moving around a cliff's edge (Sutton & Barto, 1998, p. 132). Here R is the negative reward given for moving one square or falling off of the cliff. S and G are the initial and goal states, respectively.

2.4.2 Partially Observable Markov Decision Process

MDP models operate under the assumption that the agent's state is perfectly observable. This assumption is convenient for simulated systems and simple applications, but not to model real systems with sensors and actuators that are affected by noise and errors. An agent can therefore never be certain about its true current state. To compensate for the reality of imperfect sensors and state observations, an MDP model can be replaced with a partially observable MDP (POMDP), represented by the tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R}, \mathbf{Z}, \mathbf{O} \rangle$. Since the true state cannot directly be measured, it introduces observation set, $\mathbf{Z}$, and an $\mathbf{S} \times \mathbf{A} \times \mathbf{Z}$ observability function $\mathbf{O}$, which contains the probability of making observation $\zeta \in \mathbf{Z}$ after the agent executes action a from state s.

The addition of uncertainty to the model is non-trivial, since the agent's actual state is no longer known, unless it is tracked from the start of the process, violating the Markov assumption and making the problem intractable. However, if the state size is known, it is possible to restructure a POMDP to resemble an MDP, which can be solved using existing and well-understood methods, such as value iteration. In particular, a POMDP can be reduced to an MDP by adding a new belief meta-state that maintains a probability distribution over all the states, effectively tracking the state history over time. For such a belief-MDP, the current state can be inferred from the probabilistic belief state, b, which is updated for every discrete observation as follows:

$$b'(s') = \frac{\mathbf{O}(s', a, \zeta) \sum_{s \in \mathbf{S}} \mathbf{T}(s, a, s') b(s)}{\nu(\zeta, a, b)}, \tag{2.4.5}$$

where ν is the normalisation factor

$$\nu(\zeta, a, b) = \sum_{s' \in \mathbf{S}} \mathbf{O}(s', a, \zeta) \sum_{s \in \mathbf{S}} \mathbf{T}(s, a, s') b(s). \tag{2.4.6}$$

A belief-MDP is given by the tuple $\langle \mathbf{B}, \mathbf{A}, \tau, \rho \rangle$, where $\mathbf{B}$ is a set of belief states over the underlying POMDP's states ($b \in \mathbf{B}$), $\mathbf{A}$ are the actions from the underlying POMDP, τ is the belief state transition function and ρ is the reward function over the belief states (Sutton & Barto, 1998). The functions τ and ρ are derived from the base POMDP parameters using

$$\tau(b, a, b') = \sum_{\zeta \in \boldsymbol{Z}} \nu(\zeta, a, b) \tag{2.4.7}$$

and

$$\rho(b, a) = \sum_{s \in \mathbf{S}} b(s) \mathbf{R}(s, a). \tag{2.4.8}$$

After a POMDP has been transformed into its equivalent belief-MDP, it looks similar to a normal MDP, since the belief state is now fully observable. However, a key difference between the two is that a belief-MDP has a probabilistic policy, as opposed to the deterministic policy produced by a normal MDP (i.e. each state has exactly one optimal action assigned to it).

Existing solutions for MDPs can be used to solve POMDPs transformed into their equivalent belief-MDPs, but need to be modified to account for random belief states and actions. For example, the value iteration algorithm can be modified to find the optimal sequence of actions, given some initial belief state, by using

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi}}{\operatorname{argmax}} \sum_{t=0}^{\infty} \gamma^t E[\mathbf{R}(s_t, a_t), b_0, \boldsymbol{\pi}]. \tag{2.4.9}$$

This provides an exact solution to the belief-MDP, assigning a unique action to each state. However, this is a computationally expensive process and often impractical for solving POMDPs with a sizeable state-space. In such cases, approximate solutions, such as point-based value iteration (PBVI) (Pineau *et al.*, 2003), are more practical. Examples include the works by Boger *et al.* (2005) and Hoey *et al.* (2010), who implemented a POMDP-based controller to assist people with dementia to wash their hands. Their work showed promising results and indicates that this decision making approach can be used to guide users in performing a sequence of tasks to complete a specific task.

2.4.3 Object Detection and Classification

An important goal in computer vision is to autonomously detect objects of a certain class within a digital image. Historically, this research has focused on solutions that detect a single or a few classes well, such as human faces and pedestrians (Zhang & Zhang, 2010). However, more recent developments in computing power, dataset availability and algorithms have contributed to the development of more sophisticated methods for detecting multiple different, and potentially unrelated, object classes (Borji *et al.*, 2014). A major breakthrough in this effort was the work by Krizhevsky *et al.* (2012), which introduced the concept of Deep Convolutional Neural Networks (DCNN), called 'AlexNet'. Since this publication, many other deep-learning networks have been proposed, such as R-CNN (Ren *et al.*, 2015), YOLO (Redmon *et al.*, 2016) and ResNet (Szegedy *et al.*, 2017), and great progress has been made in terms of detection speed and accuracy.

The aforementioned works use different techniques and processes to extract information from an input image. However, each of these processes can approximately be described by the following three steps:

1. Region Selection – In this step, the algorithm determines areas of interest within an image in an attempt to separate objects from the background. The entire image is scanned (potentially many times), since an object can appear in any location with different sizes and visual features (e.g. colour). These regions are marked by multiple windows with different aspect ratios (known as a "multiscale sliding windows" process). This is a computationally intensive process, given the sheer number of possible object positions and locations that need to be marked. However, several techniques have been proposed to achieve more reasonable computation performance (Ren *et al.*, 2015).

2. Feature Extraction – After some areas of interest have been extracted from the input image, the algorithm starts searching for actual objects. To do this, it extracts a set of features from each region to provide a robust representation of the latter. Objects can be described by the classic SIFT (Lowe, 2004), HOG (Dalal & Triggs, 2005) or Haar-like (Lienhart & Maydt, 2002) feature descriptors. However, many DCNN architectures uses their own descriptors learned from large datasets, which do not need to be manually designed.

3. Classification – In this third and final step, a classifier, such as an SVM (Cortes & Vapnik, 1995) or another DCNN, uses the feature set for each window and matches it against a set of learned features. This is then used as output for the object classifier, which provides both the region and the label of the object located within an image.

Most deep network-based object detectors can be described as either single or two-stage detection models, the major difference being how they perform the region selection step described above. Two-staged models, such as R-CNN (Girshick *et al.*, 2014), follows a process similar to the three-step one

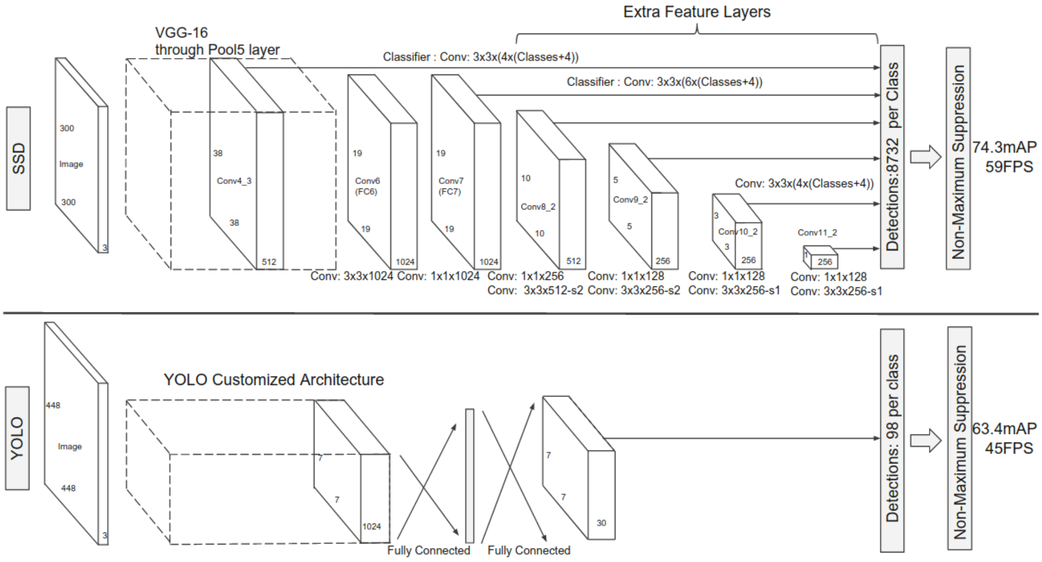


Figure 2.7: The network diagrams of the SSD (top) and YOLO (bottom) networks (Liu *et al.*, 2016).

described above, using an external region proposal network to find regions of interest. These networks achieve high accuracy levels, but their complex structure is computationally expensive and difficult to train, since two separate networks need to be optimised simultaneously (Liu *et al.*, 2018). Conversely, single-stage models consist of a single, fully enclosed network that performs region selection, feature extraction and classification. Models such as SSD (Liu *et al.*, 2016) paired with MobileNet (Howard *et al.*, 2017) are less accurate than some two-stage models, but typically have significantly lower computing requirements (see Figure 2.7 for diagrams comparing the SSD and YOLO networks). This makes the latter an attractive option for low-powered devices, such as the mobile devices used in this thesis.

2.5 Conclusion

In this chapter, an extensive review of the state-of-the-art of ETAs, HMIs and Active Vision is presented. A number of shortcomings in the body of knowledge were identified. These include the limited number of participants with visual impairments typically used in experiments, the lack of research into bone-conduction headphones with navigation tasks, and the challenges of an active vision system with human-in-the-loop. Furthermore, a number of relevant methods and tools were introduced to provide background information on MDPs, POMDPs, and object detection systems. While the latter are not the topic of this research, they are useful to understand some of the concepts presented in the following chapters.

Chapter 3

Active Guidance System Design

This thesis attempts to determine whether it is possible to use active vision techniques with a human in the system loop. The issue of accessibility for people with vision impairments (PVI) is used as a platform and use-case for this investigation. In particular, an object detection and searching aid is implemented that will benefit PVI within unknown environments. Such a system could be a significant boon for increasing PVI's independence and for enabling them to play a more active role in modern society (RNIB, 2016). There are systems available (academic and commercial) that are able to find and guide a person towards a target object, but these typically use a passive detection approach that relies on chance for the target object to fall within a sensor's range (Bigham *et al.*, 2010; Schauerte *et al.*, 2012). The research in this thesis attempts to address this problem by implementing a complete object detection and guidance system, that *actively* guides a user to a target object, which is not necessarily within a sensor's range. Such a system draws knowledge from the fields of human-machine interfacing (HMI), active vision and mobile computing.

This chapter describes the overall design of the proposed guidance system and its individual components. It begins with a description and a conceptual design of the system in Section 3.1 and Section 3.2 respectively, followed by descriptions of the proposed subsystems in Section 3.3 and Section 3.4. The chapter concludes in Section 3.5 with a summary of the proposed design solutions, which are presented in detail and evaluated in the coming chapters. Part of the solutions proposed in this chapter was previously published as a conference paper (Lock *et al.*, 2017).

3.1 System Description

This research investigates a new concept of active vision with human-in-the-loop, building an active object search and guidance system for PVI. The final system should be fully mobile and should not require any additional ex-

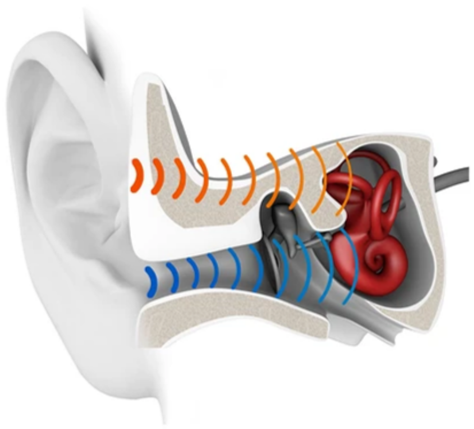


Figure 3.1: An illustration showing the differences between the air- and bone-conduction pathways[2].

pensive hardware or data connections to external services via the internet, for example. These design choices address the issue of low acceptance observed among PVI towards existing commercial guidance systems Arditi & Tian (2013); Golledge *et al.* (2004). The reason for the indifference PVI typically display towards existing solutions, often preferring the simple walking cane instead, can be attributed to these systems' clunkiness and price, often being a costly advertisement of the users' disability, rather than an effective guidance aid (Golledge *et al.*, 2004). Furthermore, existing commercial products are often prohibitively expensive, with a walking cane augmented with a sonar costing as much as £590 (e.g. the GuideCane (Borenstein & Ulrich, 1997) and the UltraCane[1] at the time of writing).

For these reasons, the guidance system proposed in this thesis will be based on a mobile phone or a small tablet which are common personal devices nowadays and will therefore not make the user stand out from the crowd. The only additional hardware requirement is a set of inexpensive bone-conduction headphones to transmit the guidance instructions to the users. Typical over-ear headphones conduct sound via air vibrations that are carried to the cochlea (the inner ear) via the ear canal and ear drums. Bone-conduction headphones, however, are placed on the user's cheekbones (or elsewhere on their head) and conduct sound through the skull directly into their cochlea and therefore do not impede a user's normal, air-conducted hearing function. Figure 3.1 illustrates the differences between the two audio signal transmission media.

Specifically, an Asus Zenphone AR[3] handsets and a Sandstone Tango development kit (each pictured in Figure 3.2 and Figure 3.3) were used to develop the guidance system prototypes. Both of these devices are equipped with a

[1] `www.ultracane.com/`

[2] `help.aftershokz.com/hc/en-us/articles/115002337953-How-They-Work`

[3] `www.asus.com/Phone/ZenFone-AR-ZS571KL/`

Figure 3.2: The latest Tango and ARCore compatible device from Asus: the Zenphone AR.

Figure 3.3: The original Tango developer's kit tablet device.

Figure 3.4: The AfterShokz bone-conduction headphones used in this project.

set of depth-perception cameras and are enabled with Google's experimental Tango SDK[4] and its more recent ARCore[5] toolkit, which provide sophisticated tracking and depth perception APIs. In addition, a pair of AfterShokz Sportz Titanium[6] bone-conduction headphones, pictured in Figure 3.4, were used to develop and evaluate the proposed guidance interface.

[4] en.wikipedia.org/wiki/Tango_(platform)

[5] developers.google.com/ar

[6] https://aftershokz.com/collections/wired/products/sportz-titanium

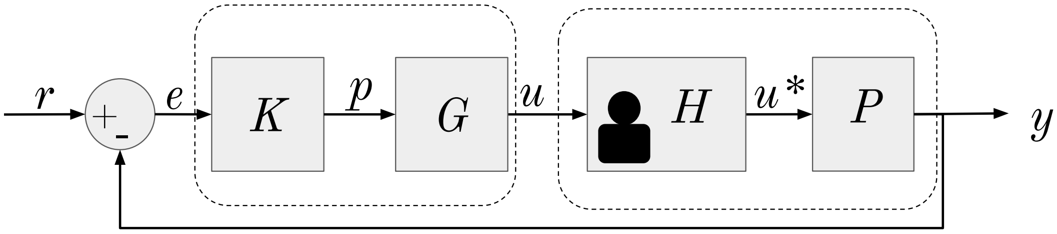


Figure 3.5: A high-level block diagram representation of the guidance system.

3.2 Active Guidance

The proposed system is based on one of the aforementioned devices, e.g. a tablet or mobile phone, and will have access to their onboard hardware, including inertial measurement units (IMU) and high-resolution cameras. The system will take a user's desired object as input and will output instructions to guide them towards it. In this setup, the user forms a central component of the guidance system, acting as the manipulator that adjusts the device's viewing angle, and is therefore included in the control loop.

A high-level representation of the proposed guidance system is given in Figure 3.5. Conceptually, this is similar to the classical closed-loop control problem shown in Figure 1.1, where the difference between the desired and actual output of the process is used to generate a signal that controls the process state. For the guidance system, the process tries to drive the output, y, to the goal (a target object), r, by using active vision concepts to generate the control signal, u. However, instead of an electro-mechanical servo, u is driving a human user, represented by the block H. The latter transforms u to the signal u^* (i.e. the actual camera manipulation), reflecting the variability between different users in perceiving different sensory stimuli. Therefore, it is important to design the HMI, G, and the controller K's signal, p, such that u tracks u^* as closely as possible. As the user interprets u^* and manipulates the camera sensor, P, the difference, e, between the current and desired states, as well as the controller output, change accordingly.

The challenge in designing a good guidance system for the proposed task is two-fold. Firstly, K must be independent of the environment and the user, meaning that objects can be placed in different unknown positions, and potentially large divergences between u and u^* by different user behaviours must be handled appropriately. Secondly, the HMI must provide clear and easily interpretable instructions that will allow the user to execute the correct action and move the device to the intended position, as instructed by the guidance system. These considerations are taken into account in the initial design of the audio interface and the controller, as discussed in the following chapters.

3.3 Human-Machine Interface Module

This section describes the HMI module, specifically motivating the design choices that were made to implement an active guidance system for PVI, and the different aspects that may affect its performance.

3.3.1 Feedback Modality

The purpose of the guidance system presented in this thesis is to generate navigation instructions and to effectively and accurately communicate them to a user with visual impairments. Traditional navigation systems, such as Google Maps and GPS devices, use a simple 2D topographic map of the user's surroundings and overlay it with the recommended route, providing periodic vocal instructions to supplement the visual data. In this case, a non-visual interface should be implemented for the guidance system to accommodate the needs of this research's intended audience of people with visual impairments. In particular, the guidance is limited to two dimensions, in the lateral (pan angle) and median (elevation angle) planes, since only the direction of an object is considered, and not its distance from the user.

Common solutions are haptic, audio and vocal feedback Flores *et al.* (2015); Marston *et al.* (2019) and haptic feedback via vibration signals were originally considered. To avoid additional hardware requirements, the mobile device's on-board vibration actuators could be used. However, these actuators are not sufficient to transmit directional guidance instructions. Vocal instructions use discrete, turn-by-turn guidance signals that are well-suited for macro-navigation tasks, such as navigating to a certain building in a city using Google Maps, for example. However, for a search task focussed on the user's immediate surroundings (i.e. a micro-navigation search task as described by Petrie *et al.* (1997)), an adjustable audio signal is a more appropriate feedback mode, given the increased guidance resolution it provides. The advantage of audio over vocal feedback for a micro-navigation task is further highlighted by researchers that recorded users' general dissatisfaction when vocal feedback is used for such a task (Arditi & Tian, 2013; Lewis *et al.*, 2015; Golledge *et al.*, 2004). Furthermore, this feedback mode has the added benefit of requiring little additional hardware to transmit high-bandwidth information and is also flexible enough to better suit users' preferences. Audio feedback was therefore selected as the feedback medium for this research.

3.3.2 Audio Interface Design

Audio signals can be split into two separate channels (e.g. one each for the pan and elevation dimensions) and transmitted via the same device, minimising the risk of overwhelming the user. The audio cues can be manipulated to transmit different data in many ways, including adjusting the signal's spectral signature, amplitude and periodic behaviour (beeping or a continuous tone). When an audio signal is transmitted via a set of speakers or headsets, it can also be transformed to mimic the characteristics of a natural, external sound source, thereby tricking the brain into believing the sound is being played from some arbitrary position. Such a transformation can be done using a head-related transfer function (HRTF) and a pair of stereo headsets or speakers. In this thesis, the guidance interface has been designed to spatialise the audio signal in both the pan and elevation dimensions, thereby giving the user an external reference point users can localise and search for. This approach was inspired by the 'earcons' concept described by Frauenberger & Noisterig (2003), which involved so-called virtual audio realities (the audio-based version of more well-known virtual reality concepts). Spatialised audio signals are well-suited to the task, displaying similar levels of performance to vocal feedback, but with less cognitive load and higher resolution (Klatzky *et al.*, 2006). However, since bone-conduction signals bypass the outer ear structure, it does not perform well in the elevation dimension. A simple linear adjustment to the signal's pitch as a function of the elevation angle is therefore proposed instead. The pan angle can still be conveyed by transforming the audio signal with an HRTF, and indeed it has been found that this dimension is unaffected by using bone-conduction headphones (Schonstein *et al.*, 2008; MacDonald *et al.*, 2006; Stanley & Walker, 2006). The proposed audio interface is discussed in detail and evaluated in Chapter 4.

3.4 Guidance Control Module

In addition to the audio interface, the guidance system must generate step-by-step instructions that, when executed, will move the user from an initial to the final target position. For example, commercial guidance systems (Garmin, Google Maps, etc.) use GPS technology to localise a user on a road map and generate discrete navigation instructions that are a function of the user's current position and the target destination. Navigation instructions are typically triggered by significant events, such as when the user reaches the end of a road or a landmark. A chain of such events forms a set of waypoints that will lead the user to their desired destination. The key challenge here is to generate the optimal guidance instruction every time the user reaches a new waypoint. In this research, the destination is an object instead of a physical location. The goal of the guidance system is therefore to generate the optimal set of instructions so that the user points a camera towards the target object with the least amount of effort, i.e. the fewest possible waypoints.

With the stochastic, discrete and chain-like sequence of states and actions (e.g. 'go right when at the fountain'), the problem can be modelled as a Markov Chain and solved as a Markov decision process (MDP). An MDP is a control process and mathematical framework for modelling the decision-making pro-

Table 3.1: A summary of the design choices made for the proposed guidance system.

Component	Selected Item or Framework	Motivation
Hardware	Asus Zenphone AR, Tango Developer Kit	Project Tango/ARCore compatibility, small form-factor, available libraries, drivers and documentation, familiarity
Hardware	AfterShokz bone-conduction headphones	Low cost, non-interference with ambient sounds, compact form-factor
Interface	2D Audio signals	Low cognitive load, flexibility, favourable hardware requirements
Guidance	PO/MDP-based controller	Suitability for a sequential search task, well-established algorithms and modelling methods, flexibility

cess of Markov systems where the decision outcome is either fully or partly under the control of the decision-making agent. A Markov system has the so-called Markov property, which refers to the memoryless nature of a stochastic process, meaning that the system's next state is only dependant on the current state and not any state that preceded it. However, an MDP assumes that all of the states are fully observable by the agent, i.e. the agent knows with 100% certainty its current state at any given moment. Of course this is a naive assumption given that sensors are imperfect and contain errors. This issue can be addressed with a partially observable MDP (POMDP), which includes a so-called observation matrix that compensates for sensor noise and other errors. Section 2.4.1 and Section 2.4.2 explain how MDPs and POMDPs are modelled and discuss popular algorithms to make complex decisions with them.

In this research, the guidance control module's objective is to take the camera's view as input and generate guidance instructions for the user to point the camera towards the target object. An MDP- or POMDP-based solution was devised to provide such discrete guidance instructions, and evaluated on a mobile device. These implementations and experiments are presented in Chapter 5 and Chapter 6 for the MDP and POMDP models respectively.

3.5 Conclusion

This chapter laid out the general system design, introducing the individual sub-components that will form the core of the active guidance solution. These are the audio interface that communicates the guidance instructions to the user, and the control module that is responsible for generating such instructions. Some key design choices were introduced in this chapter, which are summarised in Table 3.1. Subsequent chapters discuss these components in greater detail. In particular, Chapter 4 presents the audio-based guidance interface for people with visual impairments, while Chapter 5 proposes the MDP-based controller used for active visual search, and Chapter 6 discusses the final POMDP-based guidance system for object search in a real-world scenario.

Chapter 4

Audio-based User Interface

As discussed in Chapter 3, traditional visual guidance interfaces are not suitable for this research's target demographic of people with visual impairments (PVI). An audio-based interface was therefore implemented to accommodate the largest number of people while minimising the cognitive load and effort required by the user. This human-machine interface (HMI) should be effective in guiding a person to the target with reasonable accuracy and time, be non-intrusive, and not overwhelm the user's senses or otherwise confuse them.

Humans are naturally able to determine the 3D position of a sound source and by exploiting this ability, real-time guidance instructions can be interpreted without posing a significant cognitive load (Klatzky *et al.*, 2006). A sound source can be spatialised by adjusting a tone's intensity (distance), spectral signature (elevation angle), time delay and level differences (pan angle). In this case, only the pan and elevation positions are transmitted to guide them to point the camera towards a target object or visual feature. However, the bone-conduction headphones used in this research bypass the outer ear structure and their spectral profile can therefore not be properly interpreted. The target's elevation angle can be conveyed by adjusting the tone's pitch instead, as explained in the next sections.

This chapter discusses the design and implementation of this new audio interface for a mobile assistive device in Section 4.1 and Section 4.2. A set of experiments were carried out to test its efficacy at directing a user towards a point in space, which are discussed in Section 4.3, with their results presented in Section 4.4. Finally, Section 4.5 concludes the chapter, discussing the results and how the interface will be used in the coming chapters. Parts of the research work presented in this chapter was published in a conference paper (Lock *et al.*, 2019*b*) and submitted to a journal (Lock *et al.*, 2019*c*).

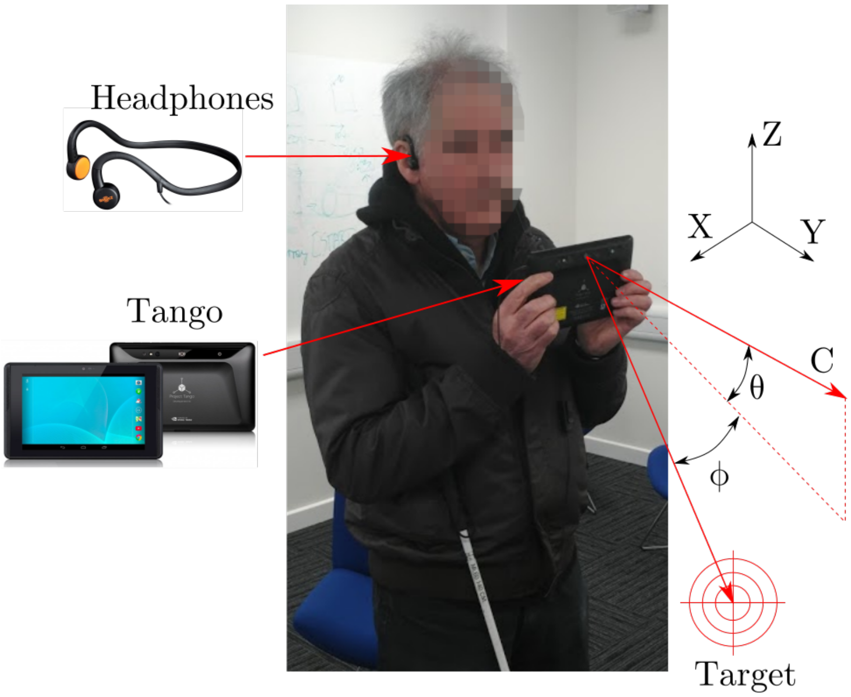


Figure 4.1: A person with visual impairments during an experiment. The hardware components used are shown, along with the reference coordinate system used to describe the angular adjustments required to point the camera (denoted by its surface normal C) at a target.

4.1 Hardware Components

The work for this chapter is based on a mobile Android device (the Google Project Tango tablet pictured in Figure 3.3 and Figure 4.1). As a result, the HMI is limited to input/output options, drivers and libraries that are available to this operating system.

A set of bone-conduction headphones are used as the audio transmission medium. These headphones are placed on a user's cheekbones and conduct the audio signals into the inner ear through the skull, instead of the auricle (the outer part of the ear) like typical over-ear headphones. This has the benefit of allowing the user access to ambient sounds and does not impede a user's ability to detect oncoming vehicles and people, for example (Lichenstein *et al.*, 2012). Alternative headphones that allow ambient noise to pass through, such as open-back headphones, still interfere with the incoming sound and were therefore disregarded. A set of AfterShokz bone-conduction headphones, pictured in Figure 3.4, were ultimately used.

4.2 Audio Interface Design

Humans can localise a sound source in three dimensions by considering cues recorded in one ear (monaural cues) and comparing cues received at both ears (binaural cues) (Blauert, 1997, 1969). Binaural cues include inter-aural time and level differences (ITD and ILD respectively) that help to determine a source's location on the lateral plane. ITD is the perceived time delay between the signal reaching both ears, while the ILD is the perceived volume difference in the signal. For example, a sound that comes from an individual's right will hit the right ear first with a slightly higher volume compared to the left ear. Monaural cues are produced by the interaction between the sound signal and the listener's anatomy (e.g. head, shoulders, outer ear) which modifies the signal's spectral profile before it enters the ear canal. When the modified audio signal finally enters the inner ear, the user is able to subconsciously analyse the frequency response and accurately determine the position of the sound source on the median plane. The distance to the source is simply derived as the intensity, or volume, of the source, i.e. a louder sound would appear closer to the user than a softer one.

These cues can be artificially generated with a head-related transfer function (HRTF) and played back to the user to simulate sounds originating at different locations. However, given the limitations of bone-conduction in conveying a spatialised audio signal's elevation component'(discussed in more detail in Section 3.3.2), a simple linear adjustment to the signal's pitch as a function of the elevation angle is proposed instead to convey the target's elevation angle. The pan angle can still be conveyed by applying an HRTF to the audio signal, since this dimension is unaffected by the use of bone-conduction headphones (Schonstein *et al.*, 2008; MacDonald *et al.*, 2006; Stanley & Walker, 2006).

A schematic of the proposed audio interface, adapted from block G in Figure 3.5, is shown in Figure 4.2. The target's pose, p, enters the interface module, from which the pan and elevation angles are extracted and sent to separate transformation modules. The elevation angle is sent to the signal generator w, which selects the correct pitch and generates the audio signal as a pure sine wave. This audio wave is then sent an HRTF block and spatialised using the pan angle value. The details of the pan and elevation angle transmission mechanisms are discussed in detail next.

4.2.1 Pan

The audio signal is based on a pure sinusoidal wave, transformed using an HRTF. People typically have trouble localising a tone without a sufficiently rich spectral profile. However, the ITD and ILD are the dominant perception mechanisms in this dimension and are independent of the tone's spectral signature, while the elevation angle is given through a different mechanism. This

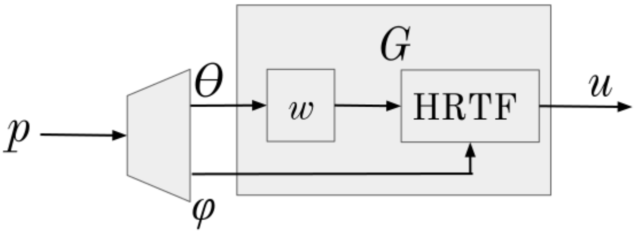


Figure 4.2: A schematic of the different components of the proposed audio interface. The target's elevation angle (θ) is extracted from the pose signal, p, sent from the controller and is used by the signal generator w to generate an audio wave with the correct pitch, after which it is spatialised with an HRTF and the target's pan angle, ϕ.

is the reason why perception accuracy in the pan dimension is largely unaffected by headphone choice. A pure sine wave is therefore suitable to convey the target's pan angle. The default HRTF provided by the OpenAL library[1], based on the MIT's KEMAR dataset (Hiebert, 2005), was applied to the audio signal to spatialise its source according to the user and target's poses.

4.2.2 Elevation

Applying a generic HRTF to an audio signal that is played back via bone-conduction headphones is not very effective in conveying a sound source's elevation angle (MacDonald *et al.*, 2006; Schonstein *et al.*, 2008). To compensate for this, the target's elevation is communicated to the user by adjusting the tone's pitch (i.e. the sine wave's frequency) as a function of its elevation angle relative to the camera's surface normal (the angle θ and vector C in Figure 4.1, respectively). When the camera vector is at the correct elevation (i.e. $\theta = 0$), the tone's pitch is set to the reference frequency, whereas the pitch (i.e. the audio wave's frequency) is increased and decreased when the target is above or below the camera vector, respectively. This high/low assignment scheme is motivated by humans' natural association of high-pitched sounds with elevated sound sources, and low-pitched sounds with source's below an individual's ear (Pratt, 1930; Blauert, 1997). An octave- and semitone-based function is used to adjust the tone's pitch to ensure perceptible changes while keeping the timbre almost constant (Shepard, 1964).

The pitch is updated at a rate of 10 Hz as the user moves the device. It changes as a linear function of the elevation angle between the camera surface normal and the target (see Figure 4.3) and the gradient is determined by setting the angle and pitch limits. For the interface, the field of view is limited to a range of ±90°, or $[-\frac{\pi}{2}, \frac{\pi}{2}]$ radians, in both the pan and elevation dimensions. After practical tests with the interface, the reference pitch that

[1] https://www.openal.org/

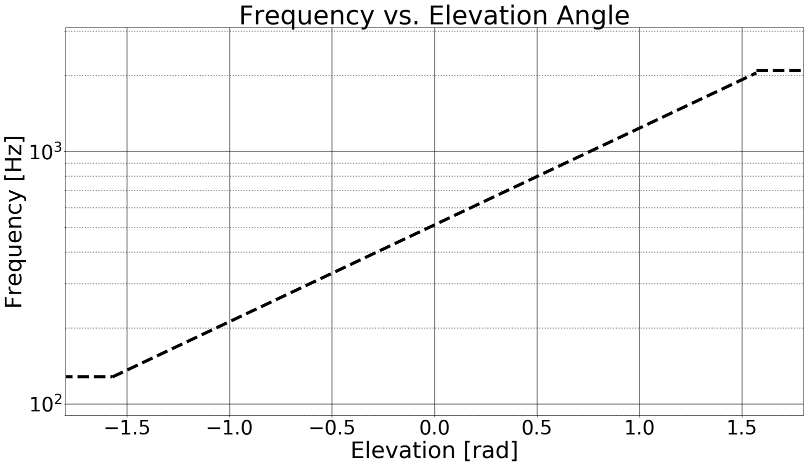


Figure 4.3: One of the pitch gain function used to convey the target's elevation angle. Note the logarithmic scale of the frequency axis.

the audio interface emits when the camera vector is on-target is set to 512 Hz, which is comfortably audible and allows for a large number of upper and lower pitch limits to be selected. These pitch limits are set at a predefined number of octaves above and below the reference pitch that indicates the camera is at the correct elevation. For example, upper and lower limits of one octave around this reference point means doubling the reference frequency to reach the upper limit and halving it to reach the lower limit, leading to upper and lower frequency bounds of 1024 Hz and 256 Hz respectively.

4.3 Experiments

A set of experiments were conducted to evaluate the interface and determine how effective it is in a pointing task where the user adjusts the pan and elevation angles of a camera to search for a target. Furthermore, a set of pre-screening experiments were also conducted to characterise each participant's hearing and determine their perception limits in the respective audio dimensions. This section describes each of these experiments and their data generation and capturing processes.

4.3.1 Interface Implementation

A diagram of the experimental system pipeline is shown in Figure 4.4, where the arrows indicate the direction of information flow. When the user taps anywhere on the device's screen, a new virtual target is generated and its coordinates are sent to the audio generation module, along with the device's current position and orientation. The audio generator then produces a tone

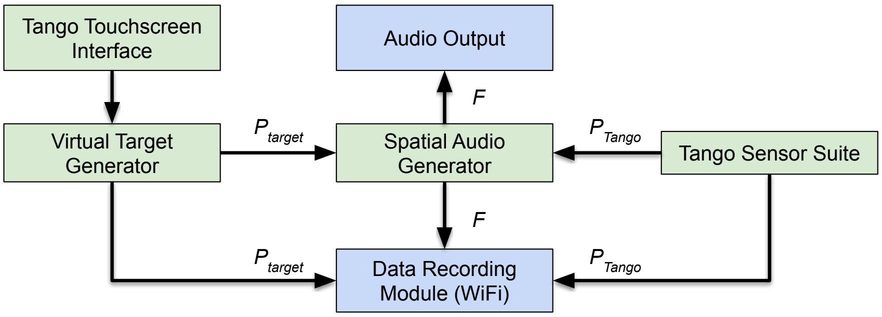


Figure 4.4: A diagram of the individual system components and their communication pipelines. F indicates a feedback signal and P a pose signal.

based on the difference between the device camera's viewing direction and the target's position. The tone is sent to the audio output channel, which plays it back to the user. The user is not explicitly informed when they have successfully found a target and have to instead rely on the audio signal and their subjective judgement determine whether they are on target or not. A WiFi recording module is constantly monitoring the device's pose, the targets' positions and the audio pitch and records it all in a remotely stored datafile.

4.3.2 Participant Characterisation

A preliminary set of experiments were conducted to characterise the participants' hearing characteristics. The measured characteristics were each participant's audio localisation ability on the lateral plane, as well as their ability to discriminate between tones with different frequencies. These results will provide context to the subsequent target search experiment as well as additional insight on any possible biases or limitations.

Lateral Sound Localisation

This experiment was conducted to evaluate the participants' abilities to determine the lateral direction a sound is coming from. To do this, a continuous 512 Hz sinusoidal tone, transformed with an HRTF to place it to the participant's left or right, was played through the bone-conduction headphones. The participant then had to select the direction the sound came from. OpenAL's default HRTF (Hiebert, 2005) was used to perform the signal transformation. The longer the experiment lasted and the more correct guesses the participant made, the closer the source moved to the centre-front of the participant, making it increasingly harder to localise.

For this progressive increase in difficulty, a '2-up, 1-down' step process was used (Wetherill & Levitt, 1965; Levitt, 1971), meaning that for every two correct answers, the distance to the centre was halved. Conversely, the task became easier for each incorrect answer by doubling the sound source's distance from the centre. Furthermore, two different step sequences, one starting at a large angular distance (45°) from the user's front and the other at the minimum distance (approximately 1°), was used, giving an 'easy' and a 'hard' progression respectively. The terminating condition for the experiment was when the two sequences converged to a direction within two intervals of one another for three consecutive guesses. For example, the experiment terminated when one step sequence was positioned at 11.25° and the other between 2.8° or 45° for three consecutive guesses. This gave an angular distance band where the participant is capable of localising the sound source. Each participant performed this experiment three times.

Pitch Discrimination

Being able to differentiate between different sounds' pitches (i.e. the ability to tell if one tone is higher or lower pitched than another) is an important mechanism for this interface. The following experiment was conducted to determine how well the participants can perform this task. These results are also used to find any potential hidden biases among the participant population and provide additional context to other experimental results.

Two pure sinusoidal tones with different frequencies were played to the participant through bone-conduction headphones. They had to select whether they perceived the second tone as higher or lower pitched than the first tone. The first tone was randomly generated, while the second tone was generated by adding or subtracting some value from the first one. These values were a function of the participant's performance. Similar to the sound localisation experiment, a '2-up, 1-down' step process was used: for every two consecutive correct answers, the pitch difference between the tones was halved, while it was doubled for every incorrect answer. Two step sequences were again used here, with one initialised with a large pitch difference ($f_h = 2^9 = 512\,\text{Hz}$) between the tones and the other with a small difference ($f_l = 2^1 = 2\,\text{Hz}$). The termination condition was when the two-step sequences were within one octave of each other (i.e. $\log_2 \frac{f_h}{f_l} = 2$) for three consecutive answers. For example, the experiment would terminate when one step sequence was set to 64 Hz and the other between 32 Hz or 128 Hz for three consecutive guesses. Pitch differences were measured in semitones, which can be obtained with the relation

$$\Delta f = 12 \log_2 \frac{f_0}{f_1}, \tag{4.3.1}$$

where f_0 and f_1 are the frequencies of the first and second tone respectively. Each participant performed this experiment twice.

4.3.3 Target Search

To test the interface's effectiveness at guiding the user in a pointing task, a set of experiments were conducted to capture the difference between the targets' actual and perceived angular positions. The participants were given a Tango tablet running an app that implements the experimental setup in Figure 4.4. The app generates a set of virtual targets and presented them to each participant through the audio interface, one at a time. The targets were set at a constant distance from the participant and their pan and elevation angles were uniformly generated across the four quadrants of the pan-elevation plane to avoid clustering. Each target's angular position was adjusted and communicated in real-time as the participant points the device around. When the participants were confident that the device was on-target, i.e. hearing the audio front-on at 512 Hz, they tapped the screen, marking the location and generating the next target. The targets' positions were all set relative to the device's coordinate system, which was tracked using the Tango hardware and localisation API. A total of 28 targets were generated per participant.

Part of this experiment's goal is to evaluate how changing the gradient of the pitch function (visualised in Figure 4.3) affects target acquisition performance, e.g. does a steeper pitch gain as a function of the elevation angle improve accuracy or decrease the search time? Pitch limits of one, two and three octaves above and below the neutral tone were then set for the so-called *lo*, *med* and *hi* pitch gradient settings, respectively. The pitch limits are given by the following intervals:

$$\begin{aligned} f_{lo} &\in [256\,\mathrm{Hz}, 1024\,\mathrm{Hz}] \\ f_{med} &\in [128\,\mathrm{Hz}, 2048\,\mathrm{Hz}] \\ f_{hi} &\in [64\,\mathrm{Hz}, 4096\,\mathrm{Hz}]. \end{aligned} \tag{4.3.2}$$

After 28 targets were marked, the experiment run was ended and the pitch gain rate was changed. The entire process was repeated two more times, giving a total of three experiment runs for the three different pitch gains and 84 targets per participant. Furthermore, to minimise any speed-accuracy trade-offs and suppress any competitive tendencies among the participants, they were asked to approach the experiment as naturally as possible and avoid emphasising time or accuracy. The order in which the pitch gain was changed for each participant was randomised in order to minimise any learning effects. Prior to the experiment, the participants were allowed to familiarise themselves with the interface and its behaviour by pointing the camera in different directions and hearing what the 512 Hz on-target tone sounds like.

4.3.4 Performance Metrics

Two different metrics are used to compare the three different pitch gradient settings: the acquisition accuracy and search time. The accuracy is given by the angles ϕ and θ in Figure 4.1, where small values for each indicate improved accuracy performance in the pan and elevation dimensions, respectively. The results are separated between pan and elevation in order to see how the different pitch gradients affect a participant's pointing accuracy.

The performance of the three pitch gradient settings in Equation (4.3.2), in terms of target finding accuracy and the time it took each participant to find a target, are also compared. However, since each participant was presented with a different, randomly generated set of targets, a direct time comparison is not appropriate. Instead, Fitts's Law (Fitts, 1954), which is a predictive model of human movement in a 2D pointing task, can be used to generate a common performance metric between the settings' search behaviours using each settings' average target estimation error and search time. The trends predicted by this model have also been observed in works with non-visual and sound-based pointing tasks (Wu *et al.*, 2010; Marentakis & Brewster, 2006; Ahmaniemi & Lantz, 2009). To accommodate the uncertain target sizes (the targets are effectively a single point in space in these experiments) and noisy data, MacKenzie's modified version of Fitts's Law (MacKenzie, 1992), is used in this analysis to determine the time performance. Both of these methods state that there is a relationship between the time it takes to find a target and the ratio between the distance to the target and its width, i.e. its so-called 'index of difficulty'. It also provides an 'index of performance' that can be used as a metric to compare the results between the three pitch gradient settings in Equation (4.3.2).

Fitts's Law is given by the following equation:

$$t = a + bID, \tag{4.3.3}$$

where t is the time it takes to find a target, a and b are constants determined through regression and is ID the target's index of difficulty, given as a logarithmic ratio between the distance to the target and its width. In this case, the targets have no width, since they are points in space, and MacKenzie's modified form for ID is therefore used instead. Here, ID is given by

$$ID = \log_2 \left(\frac{\theta_d}{w_e} + 1 \right), \tag{4.3.4}$$

where θ_d is the angular distance between subsequent target centres and w_e is the targets' effective angular width (Welford, 1968). This effective width is given by

$$w_e = \sigma\sqrt{2\pi e} = 4.133\sigma, \tag{4.3.5}$$

where σ is the standard deviation of the error data, taken as the angular difference between the participant's target selection and its actual angular position. Finally, Fitts's index of performance, IP, can be calculated using the relation

Table 4.1: A summary of the participant demographics.

	Group *G1*	Group *G2*
Gender [M/F]	10/32	7/3
Age [years]	20 ± 2	61 ± 17
Degree of Vision Impairment	N/A	7 totally blind, 3 with very limited light perception
Experience with ETAs	None	None

$$IP = \frac{ID}{t}. \tag{4.3.6}$$

4.3.5 Experiment Procedure

Two groups of participants were recruited for the experiments on a volunteer basis. Group *G1* consisted of 42 undergraduate students (10 male, 32, female) with normal eyesight who were blindfolded for the experiments (mean age: 20 ± 2 years). Group *G2* contained 10 people (7 male, 3 female) with severe visual impairments (mean age: 61 ± 17 years). Of the latter group, 3 are congenitally blind, while the rest were classified as severely sight impaired later in life. Of these, 3 participants still have limited light perception with no ability to reliably discern shapes and objects (the rest had no light perception). Nevertheless, they were asked to close their eyes during the experiment. None of the participants reported any significant prior experience with electronic navigation aids and none had any hearing or other disabilities that could have influenced their performance in the experiments. These demographics are summarised in Table 4.1.

Each participant performed three sets of experiments each, with the two characterisation experiments preceding the final target-search experiment. Both groups were given some time before the target search experiment to familiarise themselves with the system, the audio signal's behaviour and the 512 Hz on-level tone. Furthermore, to minimise any potential speed/accuracy biases, we asked the participants to focus on finding the targets without worrying about the time it took to complete the task.

4.4 Results

This section contains the results collected during the experiments. Each experiment's results are first presented and analysed individually, and then summarised and discussed together in the last section.

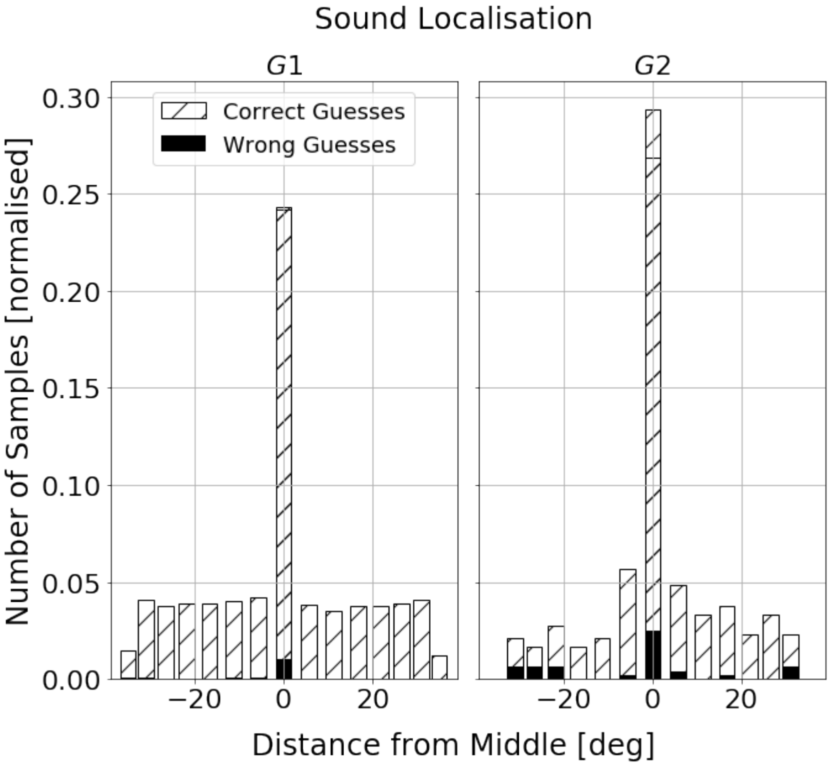


Figure 4.5: Histograms of all of the participants' guesses of the tone locations. Each bin shows the correct and incorrect guesses.

4.4.1 Characterisation of Sound Localisation

Figure 4.5 shows the results captured from the sound localisation experiment in which the participants had to select the direction (left or right) where the tone was played from. This histogram plot was generated by summing the number of correct and incorrect guesses for each participant. The 2-up-1-down experiment procedure used here led to an unequal distribution on total guesses for each participant, since some participants reached the equilibrium faster than others. The data were therefore normalised for each participant before being normalised for the total number of participants to avoid unfairly weighting the participants with more guesses.

It can be seen that the vast majority of guesses for both groups were correct. For Group *G1*, most of the errors were made at the minimum distance from the centre, i.e. the most difficult to guess correctly, which is the expected behaviour. This indicates that the participants in *G1* consistently progressed through the distance intervals and it can therefore be concluded they had little difficulty determining the correct sound direction. The seemingly higher proportion of incorrect guesses for Group *G2* could be an artefact from the smaller sample size compared to Group *G1*.

Group *G2* also displays a concentration of erroneous guesses in the centre

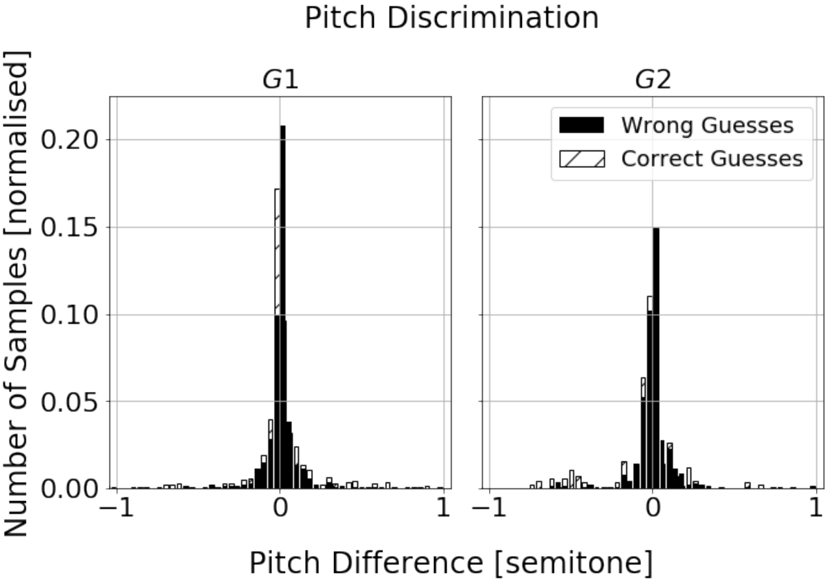


Figure 4.6: Histograms of all of the participants' guesses of which tone was higher pitched. Each bin shows the correct and incorrect guesses.

interval. However, it also shows more errors in other distance intervals and a more even progression towards the centre. This could indicate that, instead of terminating the experiment as described in Section 4.3.2, there was more switching back and forth between the three central intervals.

These results show that both participant groups are capable of determining a sound source's location with a reasonable level of consistency and accuracy. Indeed, these results are in line with previous literature (Schonstein *et al.*, 2008; MacDonald *et al.*, 2006; Stanley & Walker, 2006), confirming that humans are very adept at localising a sound source, particularly on the lateral plane.

4.4.2 Characterisation of Pitch Discrimination

The results of the pitch discrimination experiment are shown in Figure 4.6, where the bar plots show the proportion of correct and incorrect guesses while choosing which tone was higher pitched for different tone difference intervals. This plot was generated in a similar way to that of Figure 4.5, where the total number of correct and incorrect guesses were normalised for each participant and then normalised across the number of participants. For Group *G1*, it is observed that the guesses are normally spread around the 0 semitone-difference interval and the highest proportion of incorrect guesses occur in the $[-0.25, 0.25]$ semitone-difference interval. The guesses from Group *G2* are more concentrated around the centre and the majority of incorrect guesses also occur in the $[-0.25, 0.25]$ semitone-difference interval.

Assuming these differences are normally spread, a cumulative distribution function (CDF) is fit over each participant's set of results for their correct

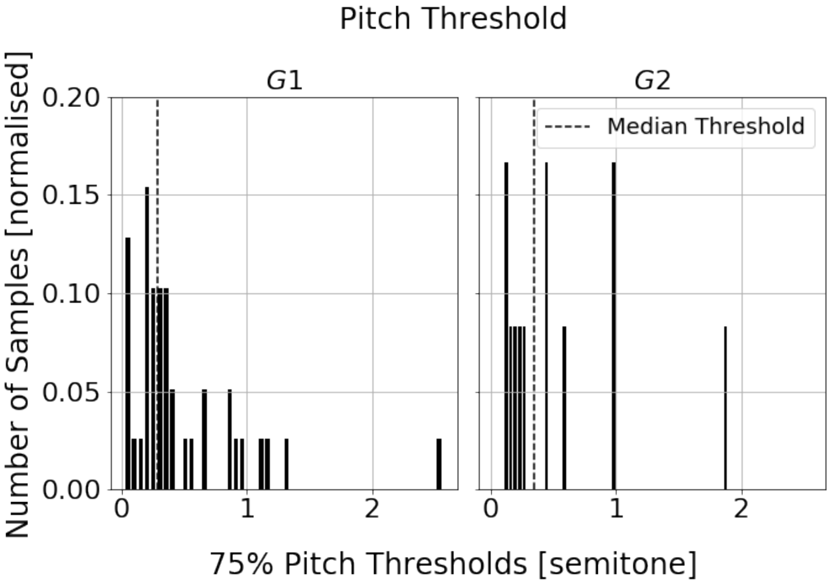


Figure 4.7: Distributions of the median cut-off frequency thresholds along with the median 75% cut-off thresholds.

guesses. Each CDF's parameters are then used to determine a frequency cut-off threshold where the participant could no longer reliably tell tones apart, which is set to contain 75% of each participant's correct guesses. The median of these threshold values can then be used to estimate the frequency difference at which the entire participant population can no longer tell the difference between two tones. It can also be used to improve the interface's pitch gain profile (e.g. Figure 4.3) and performance. Figure 4.7 shows the threshold distribution, along with the median value, which was found to be approximately 0.4 semitones for each group. Figure 4.8 contains histograms of the cut-off frequency thresholds for each setting, which show that the results for each respective setting can be grouped together.

4.4.3 Target Search

The results from the target search experiment are shown in the 2D histograms in Figure 4.9, where the angular errors in the pan and elevation dimensions are plotted against each other. A set of box-plots of the errors are also given in Figure 4.10 for each audio setting. The results are summarised in Table 4.2.

The Shapiro-Wilk test for normality reveals that none of these distributions are normally spread. Therefore the Pearson test is used to investigate the correlation between the actual target locations and participants' selected locations. These results are included in Table 4.2. The Pearson correlation scores for Group *G1* indicate a moderate to strong positive correlation between the target and the selected locations ($r_{pan} \in [0.72, 0.77], p < 0.001; r_{elevation} \in [0.36, 0.49], p < 0.001$), showing that both the pan and elevation cues gener-

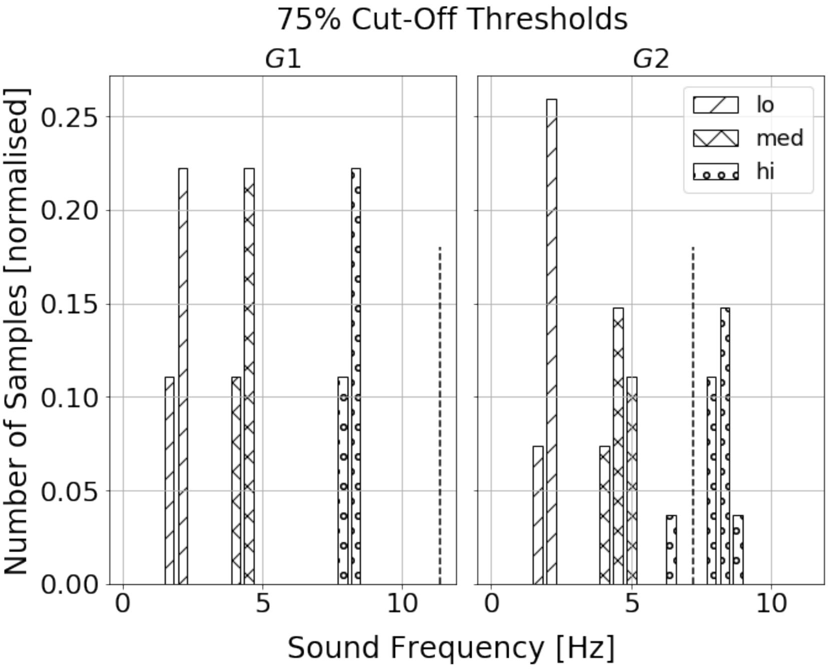


Figure 4.8: Histogram distributions of the participants' 75% cut-off thresholds. The linear Hz scale is used here to more clearly show the separation between the three settings.

Table 4.2: The average target acquisition error in the pan and elevation dimensions for each participant group, as well as their correlation scores.

		Setting	Mean Angle Error [rad]	Mean Absolute Angle Error [rad]	Pearson Correlation
G1	Pan	*lo*	-0.02 ± 0.37	0.25 ± 0.27	$0.75, p < 0.001$
		med	-0.01 ± 0.37	0.26 ± 0.27	$0.77, p < 0.001$
		hi	-0.03 ± 0.39	0.26 ± 0.29	$0.72, p < 0.001$
	Elevation	*lo*	-0.12 ± 0.51	0.42 ± 0.31	$0.36, p < 0.001$
		med	-0.11 ± 0.41	0.44 ± 0.24	$0.49, p < 0.001$
		hi	-0.15 ± 0.44	0.36 ± 0.29	$0.48, p < 0.001$
G2	Pan	*lo*	-0.01 ± 0.37	0.48 ± 0.31	$0.10, p = 0.03$
		med	0.04 ± 0.53	0.45 ± 0.27	$0.13, p = 0.01$
		hi	0.03 ± 0.48	0.36 ± 0.22	$0.21, p < 0.001$
	Elevation	*lo*	-0.30 ± 0.59	0.49 ± 0.39	$0.03, p = 0.48$
		med	-0.42 ± 0.45	0.42 ± 0.33	$0.31, p < 0.001$
		hi	-0.37 ± 0.43	0.36 ± 0.32	$0.40, p < 0.001$

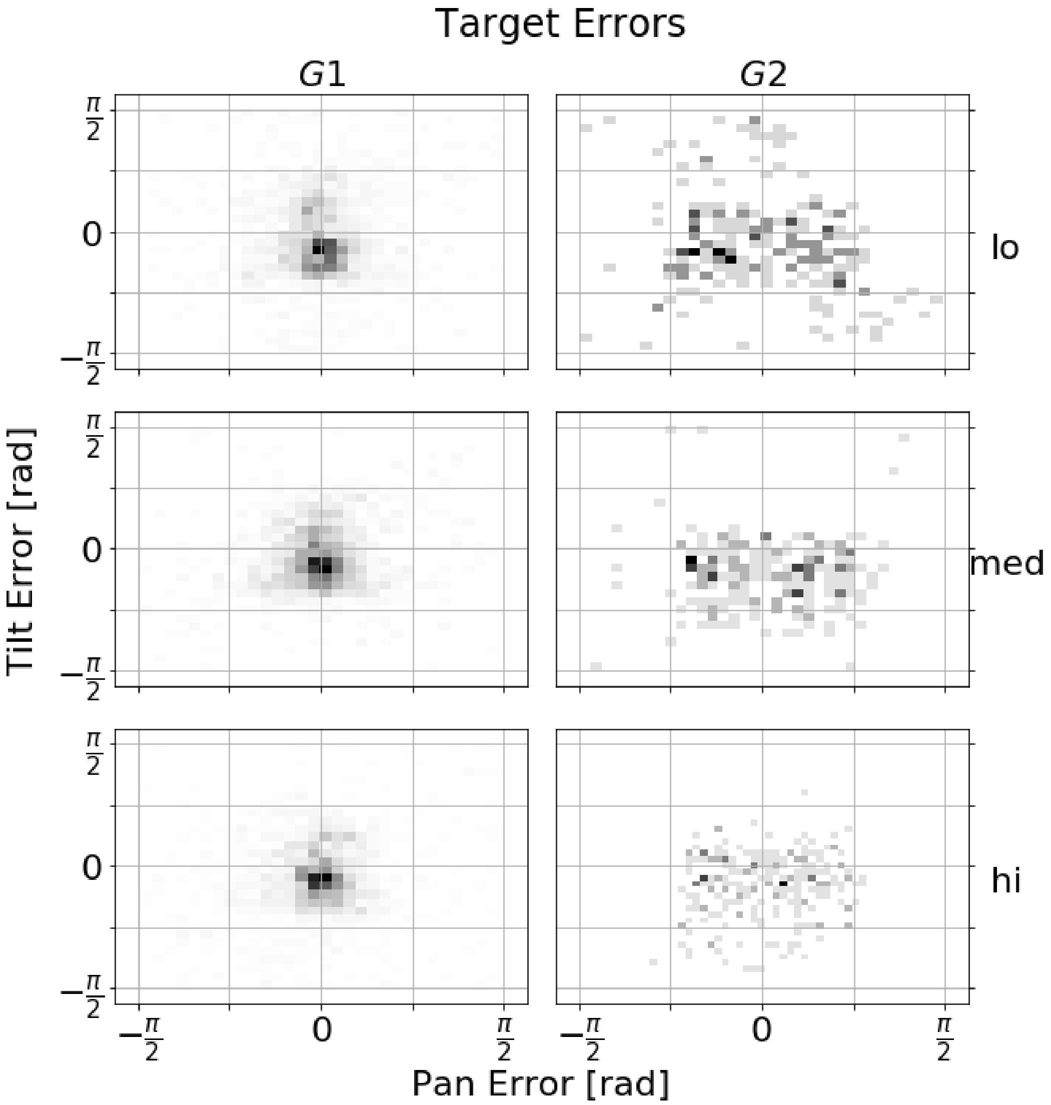


Figure 4.9: Distributions of the angular errors in the pan and elevation dimensions for the three different pitch gradient settings.

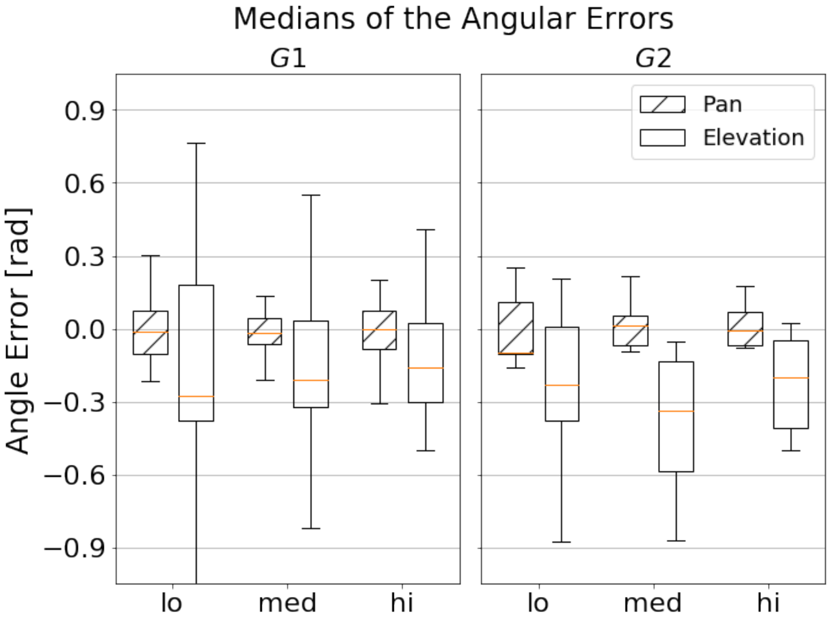


Figure 4.10: Box-plots of the median pan and elevation errors for each setting.

ally worked as expected. However, the correlation scores for Group *G2* are significantly weaker, with a pan angle correlation of $r_{pan} \in [0.1, 0.21], p < 0.03$. With the exception of the *lo* setting ($p_{lo} = 0.48$), the elevation correlation is generally stronger, with $r_{elevation} \in [0.31, 0.40], p < 0.001$.

The repeated-measures procedure that was used for these experiments requires the data for each participant to be grouped together for each setting. The medians of these data groupings are then used as individual samples that represent an individual participant's performance for each setting. Figure 4.10 shows these median data collected from each participant as a set of box-plots, while Figure 4.11 shows the collection of absolute errors.

The box-plots in Figure 4.10 show that the error in the pan dimension is approximately centred around 0 rad for both groups, with some divergence between the groups for the different settings. However, using the Friedman test for repeated measures on the medians of absolute errors, these divergences are found to not be significant ($p_{G1} = 0.17, p_{G2} = 0.09$), showing that spatial perception and accuracy are not affected by changes in the tone's pitch. This is further demonstrated by the box-plots in Figure 4.11, which demonstrate relatively consistent error levels in the pan dimension for both groups and across all three settings.

Regarding the errors in the elevation dimension, shown in Figure 4.10 for Group *G1*, a narrowing distribution is observed between the *lo*, *med* and *hi* settings respectively, as well as a median error gradually approaching 0 rad. A similar trend is observed for Group *G2*, but the improvement across the settings are more subtle and not as linear as for Group *G1*. Figure 4.11 shows

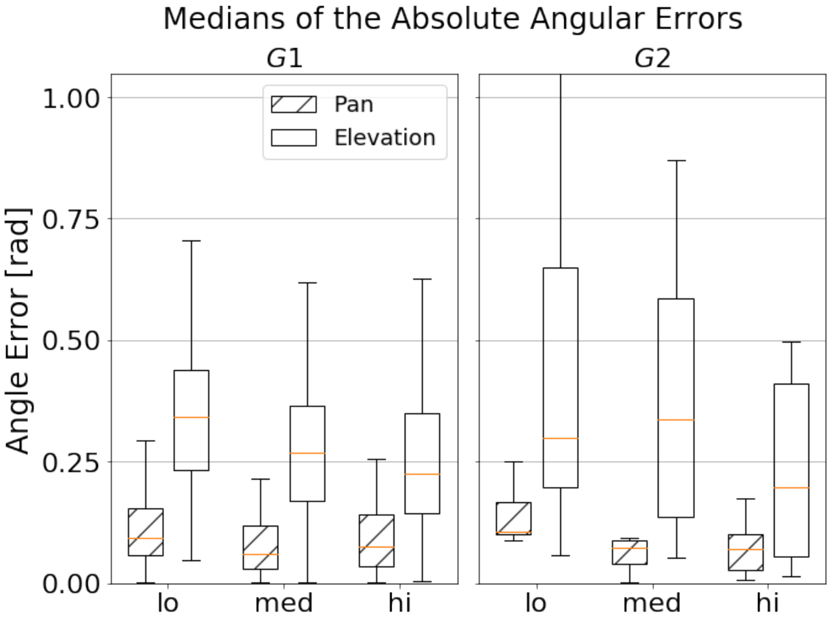


Figure 4.11: Distributions of the absolute angular errors in the pan and elevation dimensions for the three different pitch gradient settings.

a clearer improvement, i.e. errors approaching 0 rad, for the elevation data between the three settings in both groups, with the *hi* setting producing the smallest error in both cases. Further analysis of the medians of the absolute elevation error with the Friedman test reveals that the results for the different settings are significantly different from one another only in Group *G1* ($p_{G1} = 0.002, p_{G2} = 0.32$).

A post-hoc analysis using the Wilcoxon signed rank test, with a Holm-Bonferroni correction applied to the commonly used 0.05 threshold, was used to investigate the setting relationships more closely. This analysis reveals that there is a significant difference between the errors generated by the *lo* and *med* settings, as well as the *lo* and *hi* settings, for Group *G1* ($p_{lo-med} = 0.003, p_{lo-hi} < 0.001$), showing that the *lo* setting clearly produces the highest error. However, it is not clear which of the *med* and *hi* settings are best for Group *G1*. Based on the current data, it is impossible to conclude which setting produces the smallest angular error for Group *G2*, but this may be caused by the relatively small sample size for each setting. It is also noted that there is a significant negative error bias in the elevation error data for all of the settings and both groups, possibly caused by a cognitive constraint introduced by the floor, below which the participants believed a target could not appear. A similar trend was observed by Stanley & Walker (2006). Since this bias seems to be constant, it could easily be removed by adjusting its frequency parameters to shift the bias upwards by some offset.

The average absolute pan errors from Group *G1* falls within the ranges ob-

Table 4.3: A summary of the p-values for the comparisons between the different settings' and groups' error data in both the pan and elevation dimensions.

	Pan	Elevation
lo	$p = 0.18$	$p = 0.90$
med	$p = 0.86$	$p = 0.34$
hi	$p = 0.28$	$p = 0.38$

served by MacDonald *et al.* (2006) and Schonstein *et al.* (2008) of $[0.16, 0.38]$ radians and $[0.17, 0.26]$ radians, respectively. Indeed, the errors for each setting fall within the latter, more conservative range. However, Group *G2* demonstrates a wider spread in their error data and higher average error than *G1* ($[0.25, 0.26]$ vs. $[0.36, 0.48]$ radians). The results from Katz & Picinali (2011) and Zwiers *et al.* (2001) reported a similar trend.

The elevation estimation performance for both groups deteriorated when compared to their pan estimation results, which is expected given previous experimental results on human audition (Barfield *et al.*, 1997). The mean absolute error ranges for of two groups are $[0.36, 0.44]$ radians and $[0.36, 0.49]$ radians for Groups *G1* and *G2* respectively. These results are more similar than the pan results are. Comparing these results to those from Schonstein *et al.* (2008) for bone-conduction headphones with a signal spatialised in the elevation dimension, this method increases the performance by approximately 57–144% for group *G1* and 41–105% for group *G2*. Indeed, the performance is comparable to that of open-back and high-quality in-ear headphones.

Comparing the distributions for each setting between the two groups with the Kruskal-Wallis test for non-parametric data, it can be seen that the differences between the distributions for all three settings are not significantly different for either group and for both pan and elevation (the p-values are summarised in Table 4.3). These results confirm that the performance of the blindfolded participants and those with severe vision impairments are statistically similar, and that groups from a different population can reasonably be expected to produce similar errors, under similar experimental conditions. Consequently, it is concluded that the *hi* setting, which generates the smallest elevation error, is the best audio pitch level to guide a user in a pointing task, and that the pan error is completely independent of such setting choice.

4.4.4 Time to Target

To investigate if the interface generates a Fitts-like response from the participants, the time to find the target as a function of the targets' indices of difficulty, as defined by Equation (4.3.4), is plotted. The data are binned in intervals of the effective target width (w_e) as given by Equation (4.3.5) and are plotted for each gradient setting. A logarithmic line is fitted through the bins'

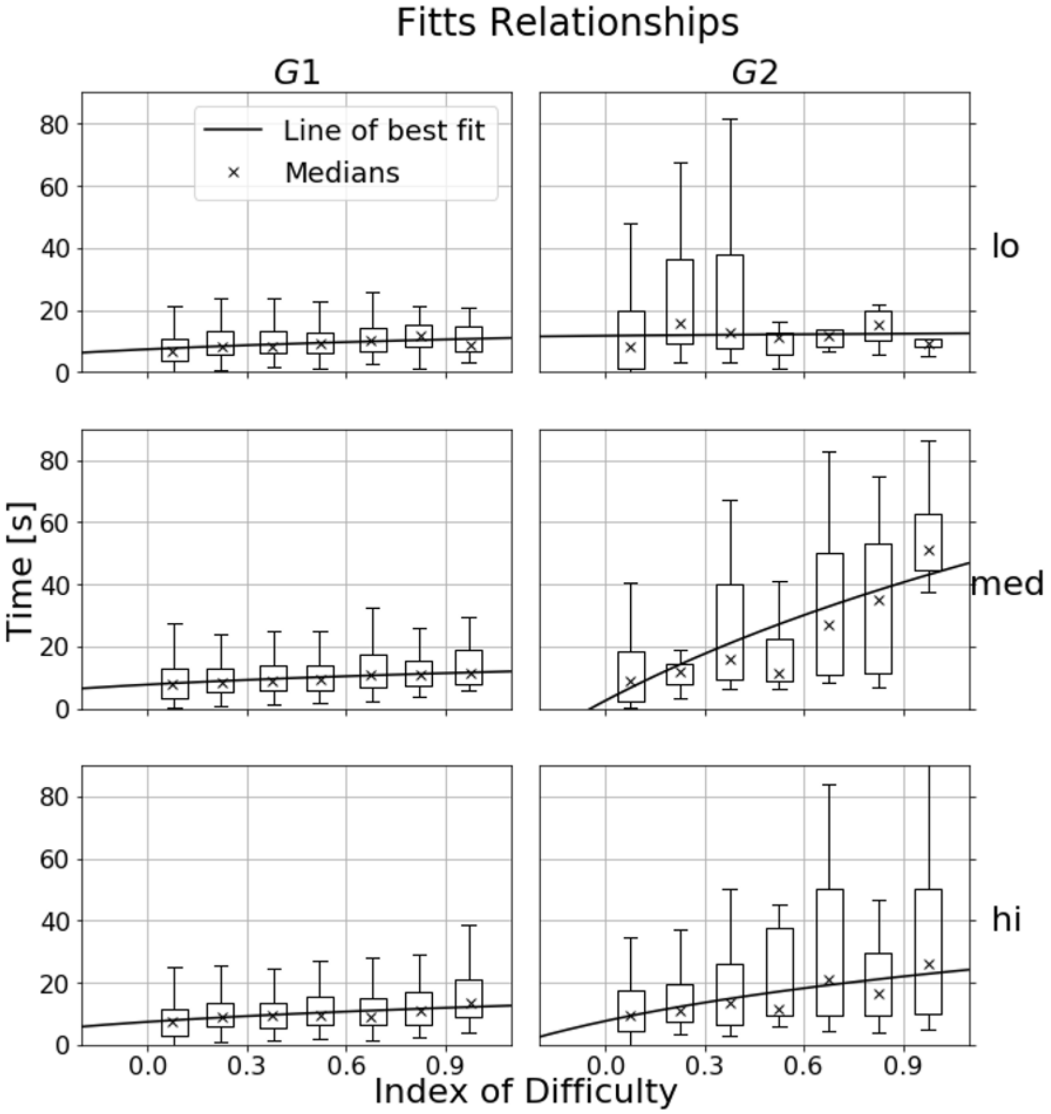


Figure 4.12: Plots showing the Fitts relationship between the time it took the participants to find a target and as a function of the target's index of difficulty.

median values by regression and all the results are presented in Figure 4.12.

For Group *G1*, a Fitts relationship can be observed and the logarithmic line of best fit closely approximates the median values of the binned data for all three settings. This is confirmed by strong Pearson correlation scores for each setting ($r_{lo} = 0.76, p_{lo} = 0.045; r_{med} = 0.96, p_{med} < 0.001; r_{hi} = 0.86, p_{hi} = 0.013$). Regarding Group *G2*, larger spreads for each binned dataset are observed, indicating less consistency in the time-to-target results for participants with severe vision impairments. This could be due to each participant's result being taken as a single datum and to the smaller population size in Group *G2*. Nevertheless, the lines of best fit for the *med* and *hi* settings exhibit strong Pearson correlation scores ($r_{med} = 0.88, p_{med} = 0.008; r_{hi} = 0.84, p_{med} =$

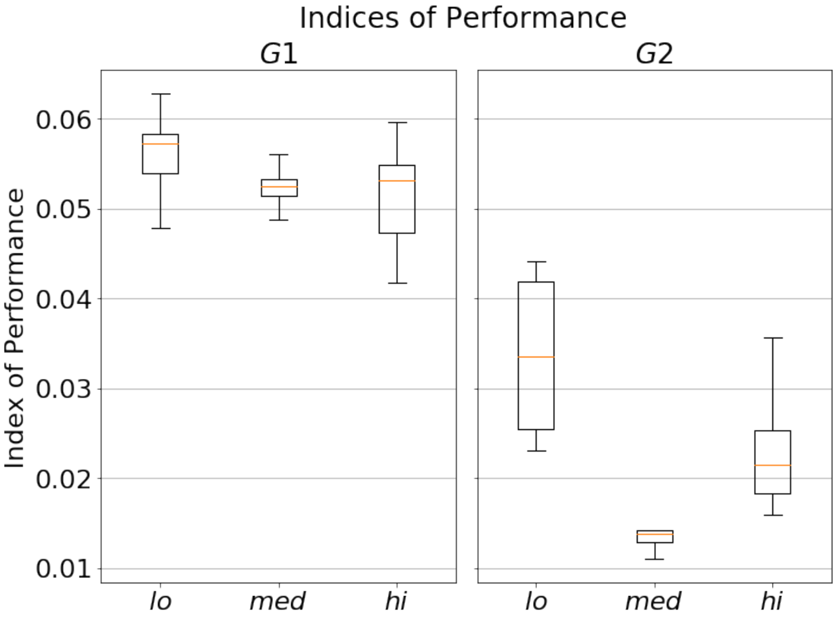


Figure 4.13: Box plots showing the participants' indices of performance.

0.017), while the results for the *lo* setting does not produce a statistically significant correlation ($r_{lo} = 0.08, p_{lo} = 0.85$).

These results allows for the indices of performance to be calculated by Equation (4.3.6), and plotted for each setting in Figure 4.13. Their results are summarised in Table 4.4. For Group *G1*, there is a fairly consistent level of performance between the three settings, with *lo* producing the highest indices of performance overall (i.e. the participants found the targets with the smallest error in the least about of time). This is supported by the results from the Friedman test, showing that there is a significant difference in performance between the settings ($p < 0.001$), as well as post-hoc Wilcoxon tests with Holm-Bonferroni corrections, which show that the *lo* setting is significantly different to the *med* and *hi* settings ($p_{lo-med} < 0.001, p_{lo-hi} < 0.001$). The *med* and *hi*, instead, are not significantly different from each other ($p_{med-hi} = 0.85$). The results for Group *G2* show generally lower and inconsistent indices of performance for each setting, which is expected given the increased times to target observed in Figure 4.12. Again, from the Friedman test, the *lo* setting produces the highest performance by a large margin ($p < 0.001$), compared to the *med*, setting with the Wilcoxon test with Holm-Bonferroni corrections ($p_{lo-med} = 0.01$), followed by the *hi* and *med* settings' results, respectively. This seems to indicate that, for both groups, the *lo* setting produces the highest level of performance, followed by the *hi* setting.

Figure 4.13 shows a significant difference between the indices of performance for each group's respective settings, with *G2* producing significantly lower indices of performance. This is further supported by the Kruskal-Wallis test, revealing that each setting's distribution is indeed significantly different

Table 4.4: The average target acquisition error in the pan and elevation dimensions for each participant group.

	Setting	Mean IP
	lo	0.056 ± 0.005
G1	*med*	0.053 ± 0.008
	hi	0.051 ± 0.007
	lo	0.034 ± 0.009
G2	*med*	0.014 ± 0.002
	hi	0.022 ± 0.006

from its counterpart in the other group ($p_{lo} < 0.001, p_{med} < 0.001, p_{hi} < 0.001$). The difference between the blindfolded group and the group with severe vision impairments seems to indicate that the latter require significantly more time to find the target. However, it is unclear whether there is a systematic cause or simply a difference in search strategy between the two groups, e.g. *G2* may prefer, on average, a slower and more methodical approach.

4.4.5 Discussion

The results for the accuracy and time performance with the proposed audio interface shows an interesting contrast. That is, the *hi* pitch setting produces the lowest target acquisition error, followed by the *med* and *lo* settings respectively. However, this trend is almost completely reversed in the time-to-target results from the Fitts model, where the *lo* setting gives the highest level of performance, followed by the *hi* and *med* settings, respectively. Since Fitts's model takes the angular error into account, one might reasonably expect that the results for both datasets would follow a similar trend. However, the Fitts model does not account for changing stimuli and different movement strategies. It is therefore hypothesised that the reason for this divergence in performance is due to the increased resolution of the *hi* setting, which allows for finer adjustments of the device's orientation, getting it closer to the correct target, but at the cost of a higher average time-to-target. This seems to indicate a speed/accuracy trade-off in finding the targets. With the Fitts model discussed here, future versions of the audio interface can be modified to prioritise different metrics and produce the desired output.

Regarding target acquisition, the progressive improvement from the *lo*, *med* and *hi* settings (see Table 4.2) seems to indicate that that simply increasing the pitch gradient will lead to better target-pointing performance. However, Figure 4.8 shows that the frequency difference between the 'on-target' tone and the selected on in the *hi* setting is approaching the cut-off frequency of Group *G1*, indicating an inflection point where increasing the gradient reduces the final performance. Indeed, the participants from Group *G2* seem to go beyond this threshold and reach a saturation point where they can no longer reliably distinguish different tones.

A final observation is that there seems to be a general performance difference between the groups, with the blindfolded group outperforming the group with visual impairments. The data do not make it clear why there is such a difference, but it can be hypothesised that the demographic differences between the groups, particularly the median age difference, may have had an effect. Indeed, it does not seem unreasonable that the more elderly members of Group *G2* may have been less familiar with the mobile technology that was used in these experiments, which may have hampered their performance. However, more data are required to determine whether the observed performance differences are an artefact of the demographic differences between the sample groups, or whether there is an underlying difference in the way people with visual impairments localise sound.

4.5 Conclusion

This chapter investigated the implementation and use of a new spatialised audio interface with varying pitch to guide a user with severe visual impairments during a target pointing task. It was found that the blindfolded participants and those with severe visual impairments performed similarly in localising sound sources and differentiating between different tones. It was also found that both groups are able to find a randomly generated, uniformly distributed set of virtual targets with similar levels of accuracy. However, the blindfolded group outperformed the other in terms of mean time-to-target. Different pitch gradient settings (Equation (4.3.2)) were also tested and it was found that the user performance in the pan dimension, based on spatialised cues, is independent of such settings. Moreover, a speed/accuracy trade-off between the settings was noticed, where a higher pitch setting produces a smaller angular error, but at the cost of reducing the time performance (i.e. more time to reach the target). These results, together with a Fitts's Law analysis of the audio interface, provide a useful baseline to improve and refine the latter in future applications, prioritising speed or accuracy to produce the desired output.

In this case, the experiments determined that the *hi* setting produces the best results in terms of accuracy and search time. In conclusion, with these results, the audio interface block G of Figure 3.5 is validated and is ready to be integrated in the guidance system presented in Chapter 6.

Chapter 5

Visual Target Search

In addition to the audio interface described in Chapter 4, the guidance system generates instructions for the user based on the device's current location and state. These instructions are used to guide the user towards a target object. To achieve this, an intelligent sensor and control module are implemented using techniques and ideas from active vision, drawing inspiration from the exploratory work by Bellotto (2013). Regarding active vision, Bajcsy *et al.* (2018) state that

> An agent is an active perceiver if it knows why it wishes to sense, and then chooses what to perceive, and determines how, when and where to achieve that perception.

Active vision enables an electro-mechanical system to intelligently gather useful visual information on its environment in order to efficiently accomplish a task. In this research, the guidance control module is an active perception agent (as referred to in the quote above) that gathers and processes information about its environment and generates the optimal guidance instructions for the user. This control problem was schematically introduced in Figure 3.5, in which the guidance control module, K, acts as the agent. Its internal structure is illustrated in more detail in Figure 5.1. The controller K uses sensor feedback from various sources to determine the system's current state, s, which allows it to select an action, a, that determines where to generate a guidance waypoint. The details of these components are discussed and described in more detail in later sections.

This chapter provides an in-depth analysis and evaluation of the active vision controller responsible for generating guidance instructions. The chapter begins with a discussion on the controller's framework in Section 5.1. Section 5.2 then explains the controller's design and the procedure it uses to generate the guidance path. This is followed by the experiments that were conducted to evaluate the controller in Section 5.3, including a description on the controller implementation. The results are presented and discussed in

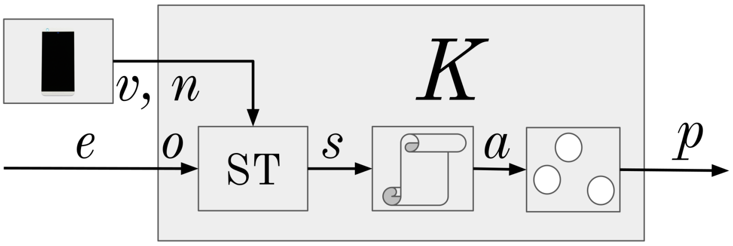


Figure 5.1: A diagram of the internal structure of the proposed controller. The mobile phone tracks the state parameters, which are used to derive the state s. The latter is used to determine the optimal action a from the policy file and to generate a waypoint location. This location is sent to the audio interface as the input signal p.

Section 5.4. Finally, Section 5.5 concludes the chapter with a short summary and conclusion. Part of the work presented in this chapter has been published in a conference paper (Lock *et al.*, 2019*a*).

5.1 Active Vision System

The closed-loop system in Figure 3.5 is conceptually similar to other classical control problems, where the difference between the desired and actual state of a process is used to generate a control signal that changes the process itself. In this case, the reference, r, is the object the user wishes to capture with the mobile device's camera. The goal of the control block, K, is to generate human interpretable instructions, u, to guide the user towards the target object. The process to be controlled involves a human, H, who interprets an instruction and executes a physical action, u^*, to actually manipulate the device's camera, P. A new observation, y, from the camera is then fed back to the loop and the error, e, is updated accordingly.

This chapter focusses in particular on the implementation of the control module K. Two important points are considered in the design of this controller. Firstly, K must be scenario-agnostic, meaning that objects can be placed in different unknown positions without affecting search performance. Secondly, since each person can interpret the instruction u differently (i.e. different transformation block H), the controller must be robust enough to handle incorrect interpretations. For example, one person might interpret and execute an 'UP' instruction correctly (i.e. $u \simeq u^*$), while another might not. This risk can be mitigated by the use of clear and simple instructions.

5.2 Human Control Module

Actual guidance is done by generating a set of waypoints, one after another, which need to be observed by the device camera, tracing a path that should eventually lead to the target object. The true position of the target object is initially unknown to the system, so it should guide the user towards its most likely locations to find it. These locations are based on an internal knowledge of the spatial relationships between objects (e.g. a computer monitor is more likely to be on top of a desk than below it). The path to the most likely object position is generated one waypoint at a time and is updated after every non-target object observation captured by the camera, or after re-orientation of the latter beyond a certain angle. The discrete transitions and probabilistic nature of the problem is well-suited to be structured as a Markov decision process (MDP) and can be solved using existing solutions for MDPs. In the subsequent sub-sections, the structure of the MDP, its parameters, how it is solved and how its output is used to guide a user are described.

5.2.1 MDP For Human Control

As explained in Section 2.4.1, an MDP is a mathematical framework that models the decision-making process of an agent, minimising some cost function while attempting to reach some goal state (Bellman, 1957). For each discrete time step, the agent finds itself in a state s and may execute any action, a, that is available to it. This transitions the agent into a new state, s', where it is rewarded or penalised by some scalar value, r. The cumulative reward — the total reward the agent receives from the environment for entering different states en route to the goal state — is the cost function that the agent is optimising for. With an MDP, it is crucial to structure the reward function appropriately so the agent avoids failure states and efficiently transitions towards the goal state. Solving an MDP produces a so-called policy that maps any state to an optimal action.

These principles are applied to the proposed guidance system to take the camera's view from an initial state to the target state (i.e. a camera view containing the target object) and minimising the effort required by the user. The policy is used by the agent (the guidance controller in this case) to generate the waypoint positions. To illustrate this process with an example, consider the scene in Figure 5.2 which contains a number of simple, distinct object observations in the red boxes, including the target observation (the mug in the bottom-left). The agent should guide the user towards the latter by inferring the current state and executing the correct actions, i.e. generating waypoints that lead the user to the target object. As the camera points to new waypoints and gathers more information on the environment, the guidance controller is able to direct the user toward the target position and terminate the search as soon as it is reached.

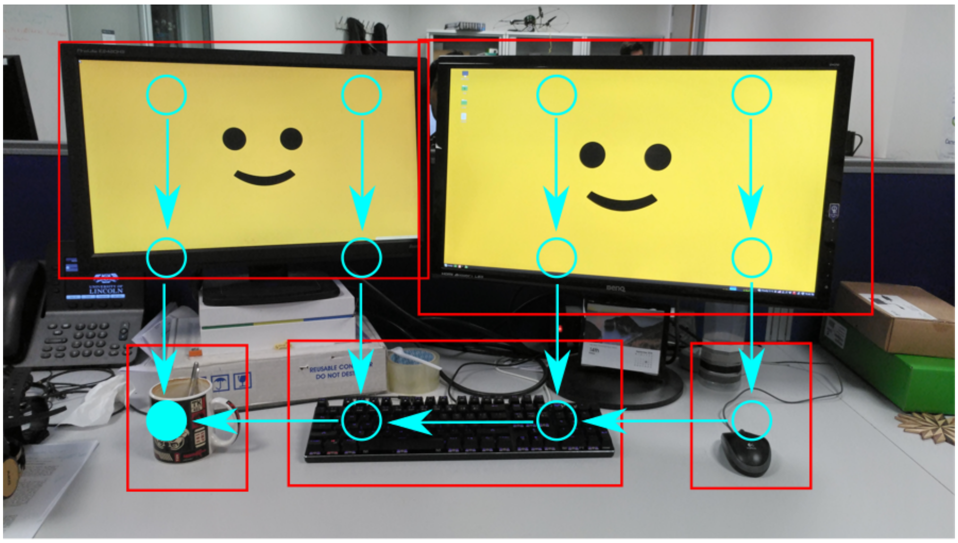


Figure 5.2: An example action policy generated by an MDP to guide the user in pointing the camera from a random starting object (e.g. monitor) to a target object (mug).

Formally, an MDP is represented by the tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R}, \gamma \rangle$, where $\mathbf{S}$ is a set of possible agent states, $\mathbf{A}$ is a set of actions an agent can take in any given state, $\mathbf{T}$ is a set of state transition probabilities that define the probability of going from state s to state s' after executing action $a \in \mathbf{A}$, with $s, s' \in \mathbf{S}$. $\mathbf{R}$ is the function that defines the reward, $r \in \mathbf{R}$, the agent receives for reaching state s' after executing action a in state s. The scalar γ is a discount factor which prioritises immediate over long-term rewards. The following sections describe how each of these parameters are defined for implementing the guidance system controller.

States

The state is a combination of parameters that define the agent's world and influences its decision process. The state vector, s, is defined as $s = \langle o, n, v \rangle$, where o is the object currently within the camera's view, n is the number of waypoints that have been generated since the search started, and v is a binary variable that tracks whether a waypoint has been generated in the same position before. o is received from the output in the feedback loop in Figure 5.1, while v and n are tracked with the device's on-board sensors. n is simply incremented for each new waypoint that is generated, while v is set to either *true* or *false*, depending on whether the device has explored a specific region before (v is initialised to *false*).

Table 5.1: The MDP's reward function values.

$r(o = o_{target})$	10000
$r(v = true)$	-10
$r(n > n_{\max})$	-10
otherwise $r(\cdot)$	-1

Actions

For any state, there is an optimal action that will maximise the agent's cumulative reward. In this case, the action is the direction of the next waypoint relative to the device's current viewing direction, and is given by the set in **A**:

$$\mathbf{A} = \{UP, DOWN, LEFT, RIGHT\} \tag{5.2.1}$$

The example in Figure 5.2 shows the linear actions that the MDP can generate. An action is considered completed, and therefore a new state reached, when the camera has rotated past a predefined angle or a new object is detected.

State Transitions

The state transition matrix, **T**, defines the probability of the agent moving from state s to state s' after executing action a, i.e. the probability of observing object o' after object o due to a pan or elevation rotation of the camera beyond an angular threshold. Therefore, **T** represents the spatial relationships between the different objects in the environment model. These spatial relationships are learned from a dataset during an initial training process, which is discussed in more detail in Section 5.2.2.

Reward Function

The reward function, **R**, defines the immediate reward, r, that the agent receives after transitioning from state s to state s'. The agent's goal is to maximise its cumulative reward and **R** is therefore an important design parameter for producing an effective policy. In order to encourage the agent to find the target object as fast as possible, a relatively large positive reward should be assigned for successfully reaching the goal state, and a smaller negative one in any other case.

The reward function is hand-crafted and the parameters were empirically selected. These values are listed in Table 5.1. The rewards punishes the agent for every waypoint it generates that does not lead to the target object and becomes increasingly negatives when a waypoint threshold is exceeded ($n > n_{\max}$) or when a waypoint is generated in the same position more than once ($v = true$) during the same search. Conversely, a significant positive reward is given when the target object and goal state are reached.

5.2.2 Policy Generation

MDPs can be solved to determine the optimal state-action mapping, or policy. This is generated through an iterative training process, where the agent is allowed to explore the entire state-action space and incrementally improve its decision function to reach the target state, maximising its cumulative reward. In this particular setup, the agent is unaware of the environment. That is, the transition probabilities and reward function are not known to the agent, but is learned through interaction with the environment during the aforementioned iterative learning process. Well-understood methods to solve MDPs and produce the optimal policy are available, with prominent solutions including Q-Learning (Watkins & Dayan, 1992), Value Iteration (Bellman, 1957) and State-Action-Reward-State-Action (SARSA) (Rummery & Niranjan, 1994).

In this implementation of the MDP controller, 8 objects are initially encoded, including a 'nothing' instance indicating that nothing of note is observed. In particular, this implementation considers a simple office desk scenario that contains the objects given by $\mathbf{\Omega}$:

$$\begin{split} o \in \mathbf{\Omega} = \{&monitor, mouse, keyboard, window, \\ &mug, stationary, desk, nothing\}. \end{split} \tag{5.2.2}$$

The spatial relationships between the objects in $\mathbf{\Omega}$ were extracted from the OpenImage dataset (Kuznetsova *et al.*, 2018), which consists of 1.74M images with 14.6M manually drawn and labelled bounding boxes around objects (see Figure 5.3 for two examples from the dataset). Pixel coordinates of the bounding boxes are provided with the dataset. Iterating over all the images that contain two or more of the objects in Ω and using the bounding box coordinates for each object in the image, it was possible to extract their positional relationship in terms of the basic linear directions specified in $\mathbf{A}$. The total number of instances for each position can then be normalised by the total number of instances between the objects to provide a probability distribution for each object combination (e.g. a desk is below, above, to the left and right of a keyboard in 5%, 75%, 10%, 10% instances respectively). The relatively simple action space compensates for the dataset's lack of absolute position information (e.g. it is only possible to determine that object 1 is above object 2, but not how far above), while maintaining the desired property of generating clear and simple user instructions.

This dataset is primarily aimed towards researchers to benchmark their object detection and classification algorithms, so the absolute distances between the objects in the scene are not included. Furthermore, camera perspective information and absolute object dimensions were not available for this dataset, so it was not possible to estimate objects' locations. While distance information is not required in this case, it would be beneficial in allowing future iterations of the controller to provide not only a pointing direction, but also to what extent the pointing adjustment should be made (e.g. look left 20°). At the time this research was conducted, an appropriately rich dataset containing the absolute positions of different objects relative to one another was not available.

Figure 5.4 shows the spatial relationship of a subset of $\mathbf{\Omega}$ (desk, keyboard and mouse) in terms of the probability of finding a certain object after executing an action. For example, when the agent is in state $s = \langle o = mouse, n, v \rangle$ and is searching for object $o_{target} = keyboard$, there is a strong probability that the latter is on the mouse's $LEFT$. The MDP of course considers all of the objects' spatial relationships when generating the optimal policy.

The agent's goal state is defined as any state where $s = \langle o = o_{target}, n, v \rangle$, which gives a total of 14 terminal states for a specific object target, assuming $n = 1$ (7×2, 'nothing' cannot be a goal). The target can be found in a location that has been marked as searched before or has not been searched at all (i.e. $v = true|false$), thereby doubling the total number of possible terminal states beyond the original 7. Each potential target object has its own unique policy file since each of them define different terminal states.

The MDP is set to generate a maximum of 11 (inclusive) waypoints before being punished (i.e. $n_{\max} = 11$). For this particular setup, 11 grid cells is the longest possible route from the initial state to the goal state (more details about the grid are given in Section 5.2.3). More than 11 waypoints may be required or generated during the transition to the target state, but the MDP considers 11 as the maximum threshold, after which the agent starts receiving additional punishment. Is is also convenient for maintaining a manageable state-space size and to simplify the reward function. Consequently, the MDP has a total of 154 reachable states ($k_{states} = 11 \times 7 \times 2$).

The Q-Learning algorithm was initially used to generate the optimal policies. Unfortunately, the lack of absolute spatial information in the OpenImage dataset generates ambiguities (e.g. a coffee mug is roughly equally likely be on the left or right hand side of a computer monitor) which led to the Q-Learning algorithm not converging in a reasonable amount of time. The SARSA algorithm was therefore used instead, since it was found that its on-policy design and more exploratory approach led to a close-to-optimal policy being found. It has an additional exploration parameter, α, used during training to control the agent's exploration vs. exploitation behaviour. This is an important parameter for training, since too much exploration may cause the model not to converge, or do so very slowly. However, too little exploration may cause the model to get stuck in a local minima. A careful balance must be struck between these two training strategies. Having direct control over α makes it easier to find a good policy, although this is not guaranteed to be optimal (Rummery & Niranjan, 1994).

The MDP is trained until it converges to the optimal, or close-to-optimal, policy for a maximum of approximately 17 million episodes. The parameter α maximises the exploration focus when it is set to 1 and exploitation when set to 0. For this solution, α is set to the exponential function

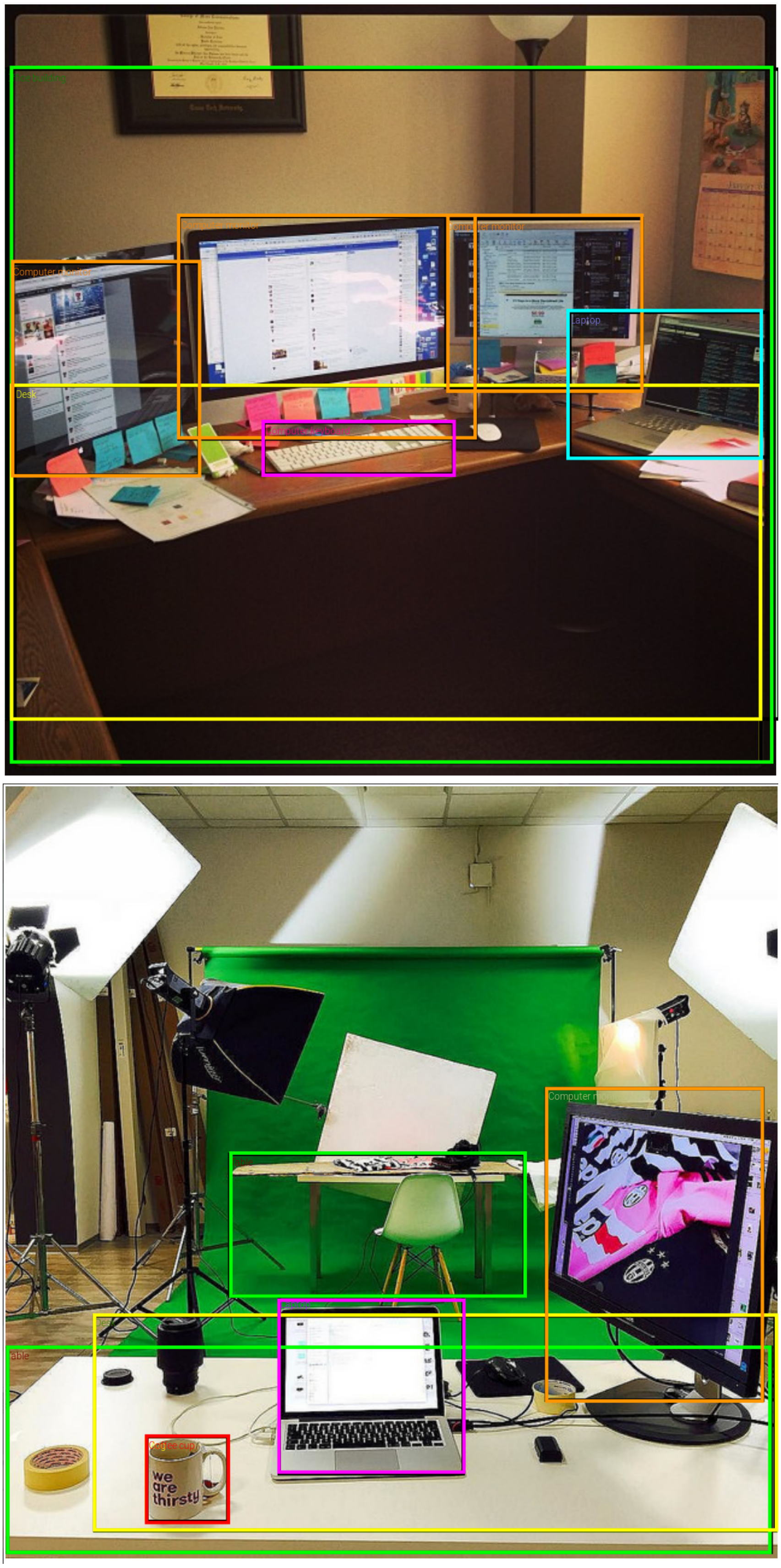


Figure 5.3: Two examples of an office environment taken from the OpenImage dataset. Both images contain some objects in $\mathbf{\Omega}$ (Kuznetsova *et al.*, 2018).

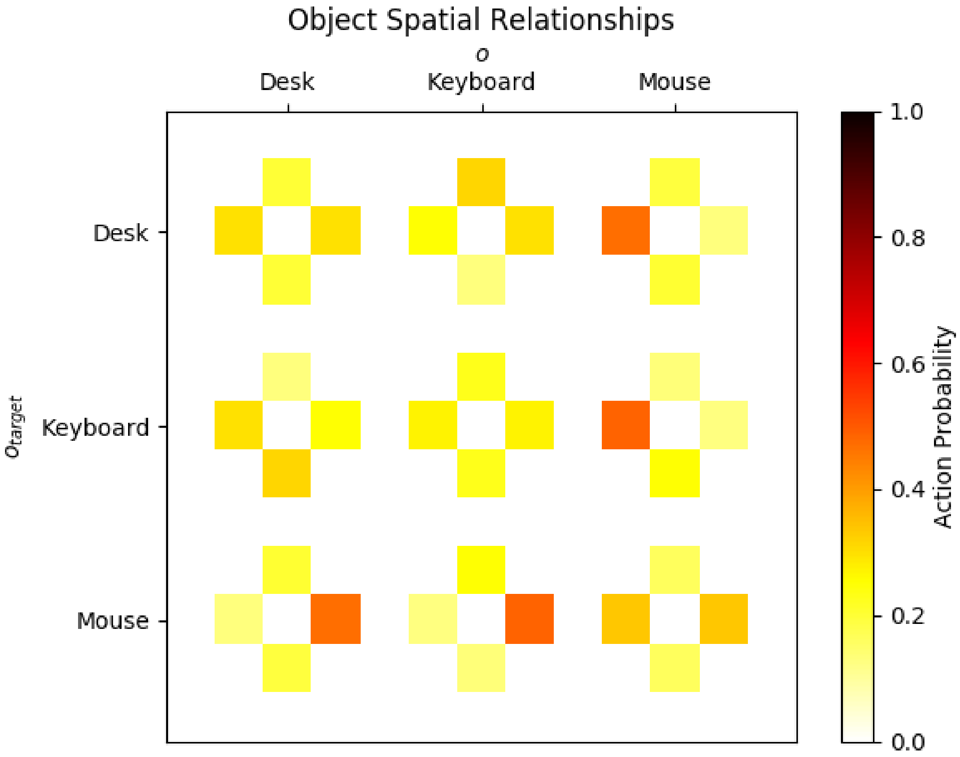


Figure 5.4: Examples of the spatial relationships between the desk, keyboard and mouse objects. Each square corresponds to the probability of executing an action (top square for UP, left square for $LEFT$, etc.)

$$\alpha = \exp\left(\frac{-i}{10\ k_{states}}\right) - 0.001, \tag{5.2.3}$$

which was heuristically selected as a function of the number of training episodes, i. It is initialised with a high exploration value, exponentially decreasing to switch focus to exploitation as training progresses. Finally, the parameter γ is set to a constant 0.95 to prioritise long-term cumulative rewards over short-term ones. The SARSA algorithm used here is implemented in the AI-Toolbox library[1].

The MDP has a relatively small state-action space, so a set of 7 policies (one for each object) were generated in a reasonable amount of time. However, it should be noted that due to the state size's multiplicative nature, adjusting the angle interval between waypoint positions or adding more actions or objects can easily lead to an intractable state-space size, where different assumptions or training algorithms might be required to generate a policy.

[1] `https://github.com/Svalorzen/AI-Toolbox`

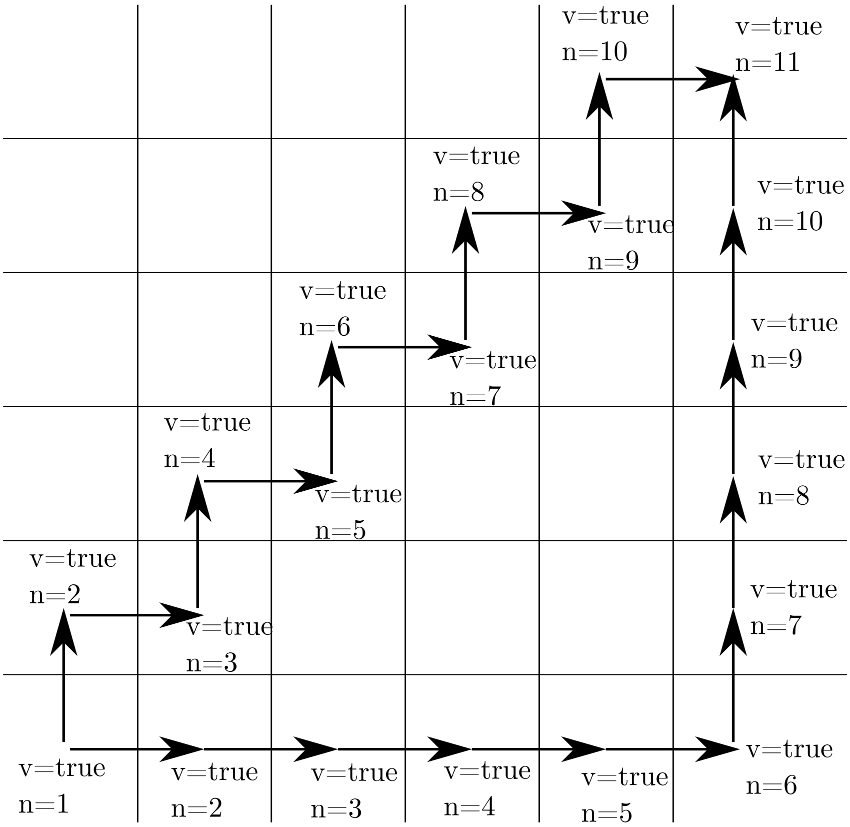


Figure 5.5: A figure showing the 6 × 6 grid that simulates the controller's world. Notice the state variables n and v being set as the controller guides the user across the grid (v for each square is set to *false* until a waypoint is generated in that square).

5.2.3 Waypoint Generation

The system uses a 6×6 discretised and wrapped radial grid to enable the waypoint tracing and generation processes. The grid spans 120° (approximately 2.1 rad) in the pan and elevation dimensions, giving a resolution of 20° (appropriately 0.35 rad) per grid cell, and wraps around the user in a semi-cylindrical fashion. Wrapping the grid (i.e. if the location of a waypoint exceeds the 120° limit, the same waypoint is moved to the opposite side of the grid) effectively limits the search space to a 120°×120° area. With the 6×6 grid, 11 waypoints (indexing from 1) is the longest possible direct route that can guide the user towards a target object, as shown in Figure 5.5. Note also the state variable v being set to *true* as the controller guides the user across the grid.

An action sampled from the policy is converted by the system into a new search waypoint centred on a cell of the radial grid (e.g. an '*UP*' action will generate a waypoint one grid cell above the camera's current orientation). Note that this cell is not part of the MDP's state and the grid is only used to discretise the camera's pan-elevation movements to guarantee minimum angular variation between subsequent actions, which ensures reliable state transitions with reasonable movement sizes. Also, the policy actions, and waypoints by extension, are relative to the current camera's pan-elevation orientation. The audio interface can then then transform use the waypoint's location to guidance instructions (i.e. u in Figure 3.5).

5.3 Experiments

The controller was implemented on a mobile device and a set of experiments were conducted to evaluate the proposed MDP-based guidance controller's performance. This section describes the initial system's implementation details, as well as the experiments that were conducted with it.

5.3.1 Controller Implementation

The MDP guidance controller and policies were integrated into an Android app (see Figure 5.6 for a screenshot) on an Asus ZenPhone AR smartphone[2] running Android 7.0 with Google's augmented reality toolkit ARCore[3], which provides the device's 3D pose. No further software or hardware modifications were required. This app is responsible for generating the guidance instructions and estimating the pose of the camera sensor (K and P blocks in Figure 3.5) in real-time. Knowing the camera's pose allows the app to infer the current state and sample the next optimal action from the policy.

The system determines the state values for n (number of waypoints generated so far) and v (waypoint already visited or not) described in Section 5.2 by recording the previous search and waypoint locations. The camera provides the ID of the object currently within view, which is assigned to the state variable o. For these experiments, a real object detector was not used. Instead, the objects were simulated with 7 different QR codes, one for each object class, and a camera-based QR code scanner from Android's machine learning API[4]. This simplification guarantees full observability of the state and allows the experiments to focus on the performance of the MDP-based guidance controller. Moreover, to speed up processing and avoid scanning multiple QR codes simultaneously, only the central 300×300 pixel area of the camera's frame is used to scan for codes. This choice also defines the precision required in pointing the camera towards the object (see the white box in Figure 5.6).

In a real application for people with visual impairments, a waypoint's position would be communicated to the user by the audio interface described in Chapter 4. However, since the scope of these experiments is mainly to evaluate the control algorithm and not the interface (evaluate K and not G), the

[2] `https://www.asus.com/Phone/ZenFone-AR-ZS571KL/Tech-Specs/`

[3] `developers.google.com/ar/`

[4] `https://developers.google.com/ml-kit/`

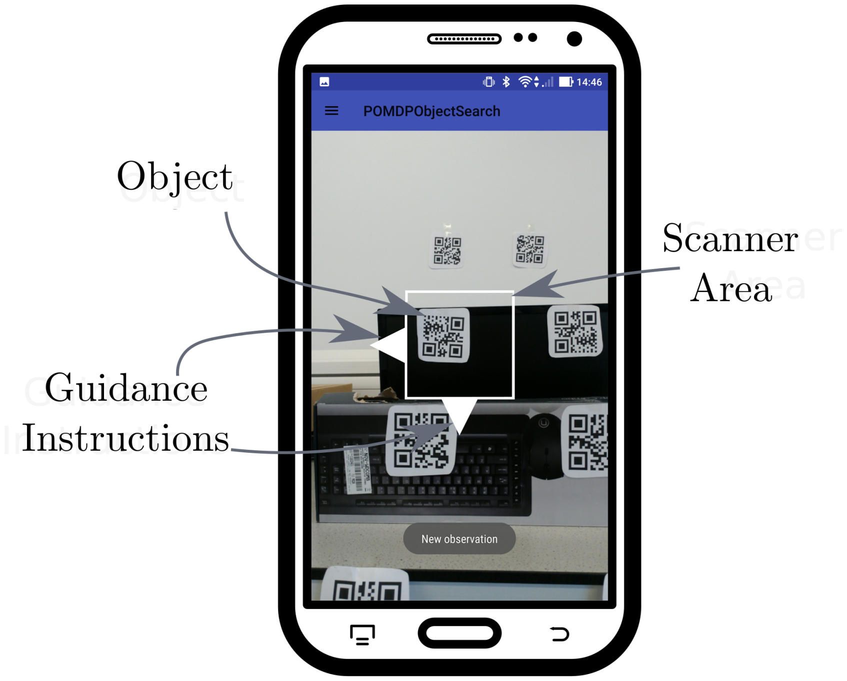


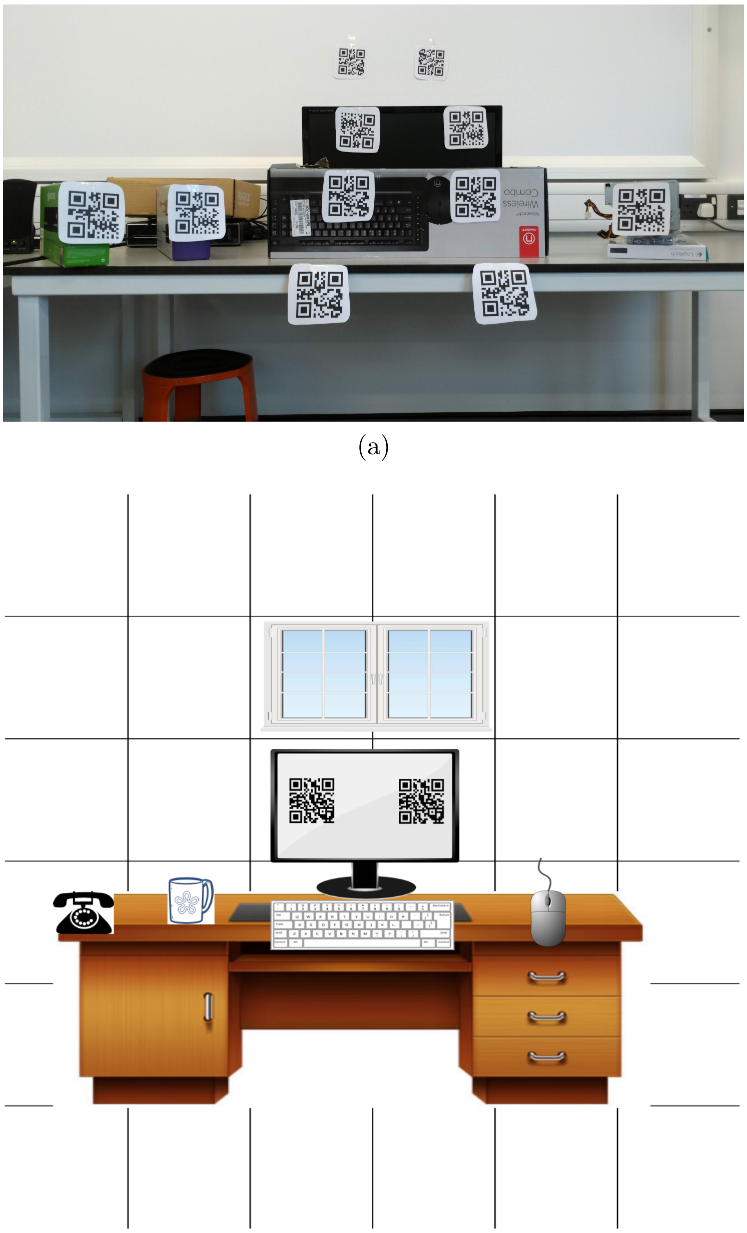
Figure 5.6: A screenshot of the Android app with the guidance controller implemented. The interface shows the QR code scanner area, as well as an example of the visual instructions used to guide a user towards a waypoint.

app provides guidance instructions with four on-screen arrows (see Figure 5.6). This visual interface is helpful for debugging and for the experimental evaluation of the controller (in Chapter 6, it will be replaced with the audio interface).

5.3.2 Experiment Design

The goal of the experiments is to determine the feasibility and effectiveness of the MDP controller that guides users in pointing the mobile device's camera towards a target object (i.e. a QR code). As explained in Section 5.3.1, this phase of the research focusses on the active vision controller, and not the human's performance in a real object-search task. Therefore, the experiments include visual markers and instructions and were performed by people with healthy eyesight. The experimental environment mimicked a typical office desk layout, containing 7 different objects (i.e. QR codes), one of which was selected as the target for each experiment run. See Figure 5.7a for a picture of the actual experiment scenario and the environment it simulates in Figure 5.7b.

For each experiment, the participant was placed approximately 1 m from the closest barcode and was asked to remain on the same spot during the experiment. The participant started the experiment by pressing a button on the app, which then guided the user towards the target object. Since the participants were allowed to use the device's display, the target was randomly



(a)

(b)

Figure 5.7: A picture of the environment used for the experiments and a schematic grid representation of the environment. Each QR code in (a) represents an object that corresponds to the schematic in (b).

selected by the app without informing the participant what they were searching for, at least until the object was found. This prevented the participants from learning the objects' locations between subsequent experiment runs by associating QR codes to them.

To avoid pointing at uncluttered edges of the search space, where the system had difficulty guiding the user back to the centre, a waypoint-limit of 15 was set. A search run therefore ended when the participant either successfully found the target object by pointing the device camera to it and scanning the barcode, or exceeded the 15-waypoint limit. After this, the participant restarted from the central position, selecting a new random target object and repeated the experiment.

Each participant performed 10 searches per object, giving a total of 70 searches recorded per participant. A total of 12 different participants were recruited, none having any disabilities or handicaps that could affect their performance. This gives a dataset with a total of 840 samples.

5.4 Results

Four metrics are used to evaluate the system's performance for the experiments: the number of waypoints to the target, the target acquisition rate (TAR), the total time it took to find a target object and the total linear and angular device displacement for each search. The results for each individual participant is presented in Table 5.2 and Table 5.3, which include the mean performances for the entire participant group. A number of simulations were also performed in an environment mimicking the experiment setup with a virtual agent that perfectly executes the policy (i.e $u = u^*$). This provides a baseline measurement to compare some of the experimental results. The TAR and the number of waypoints to target results for the simulation are included in Table 5.2. However, the simulation does not include the time delay and the time to target is therefore not included (a computer simulation will of course execute a command faster than a human). Number of waypoints, TAR, time to target and device displacement results are discussed in more detail in the following sub-sections.

5.4.1 Number of Waypoints to Target

The number of waypoints represents the total waypoints that were generated by the system to guide the participant to the target object. It gives an indication of system performance, where less waypoints means more efficient target acquisition. Figure 5.8 illustrates the cumulative distribution of the number of waypoints to the target for all participants. Approximately 75% of searches ended with the target object successfully found with 15 waypoints or less. Furthermore, Figure 5.8 shows that the majority of targets were found with

Table 5.2: A table containing each participant's results for the TAR, median number of waypoints and median time to target, including the group's means for each of these metrics. Results for a simulated 'perfect' participant are also included.

Participant	Num. Waypoints	TAR [%]	Time [s]
s1	5	79	6.41
s2	6.5	73	10.2
s3	4	80	7.88
s4	5	78	9.96
s5	7	70	13.7
s6	15	52	15.3
s7	10	64	10.6
s8	3.5	89	9.42
s9	3	82	9.65
s10	11.5	57	21
s11	5	91	9.98
s12	5	79	9.95
Participant Means	6.7 ± 3.5	74 ± 11	11.2 ± 3.71
Simulation	3.8	99.7	—

Table 5.3: A table containing the results for the mean device displacement that was recorded during the target search experiments. This includes the participant group's mean values.

Participant	Pan [rad]	Elevation [rad]	Linear [m]
s1	1.27	1.05	0.36
s2	1.28	1.46	0.18
s3	0.81	0.47	0.24
s4	0.98	0.94	0.37
s5	1.10	0.95	0.20
s6	1.93	2.17	0.11
s7	0.64	0.59	0.10
s8	0.79	0.76	0.16
s9	1.03	0.95	0.36
s10	1.31	1.40	0.47
s11	0.79	0.52	0.18
s12	1.42	1.23	0.30
Participant Means	1.11 ± 0.34	1.04 ± 0.46	0.25 ± 0.11

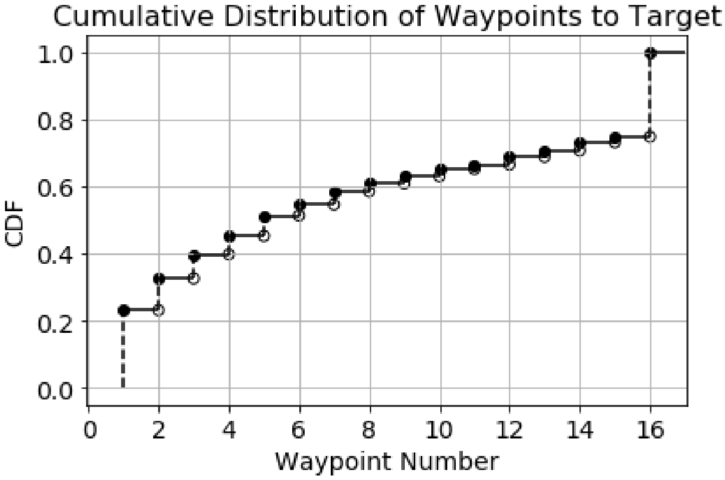


Figure 5.8: The cumulative distribution of the participants' number of waypoints generated to find a target object.

less than 6 waypoints, which is equal to the width and height of the grid cell environment. This indicates that the majority of targets were found without needing to explore the entire search environment.

5.4.2 Target Acquisition Rate

The TAR measures the proportion of searches where each participant successfully found the target object within the 15 waypoint limit. Such a measure gives an indication of how effective the system is at directing a user towards the desired object within a reasonable waypoint threshold. Table 5.2 shows that the inter-participant spread ($\sigma = 11\%$) is fairly significant, perhaps indicating that the participant's search behaviour affects the overall target acquisition performance. However, with a mean TAR of 74%, it is clear that the system successfully finds the target object during the majority of searches.

Figure 5.9 shows the TAR for each individual target object in $\mathbf{\Omega}$. There are variations to the TAR for different objects, with the QR codes representing physically smaller objects being the hardest to find. Most failure cases were typically caused by the system entering a no-recovery state where the user was directed into dead-space with no spatial information (e.g. ceiling or wall section). In this case the system could not observe useful clues to intelligently guide the user. Possible improvements for future versions of the algorithm would be to implement some fall-back method that can detect a no-recovery state (e.g. exceeding a set number of steps/time without any new object observation) and guide the user back to a position to restart the search.

It should be noted that the 15-waypoint threshold selection may affect the TAR performance measure. However, given the gradual tapering off in the TAR observed in Figure 5.8 and the fact that many searches ended in unre-

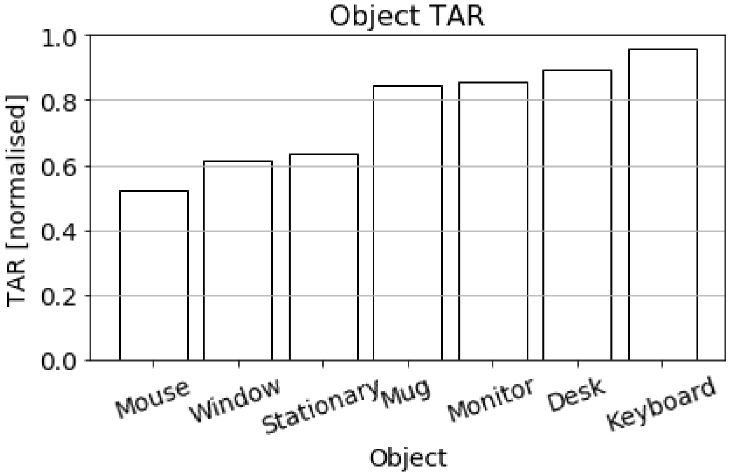


Figure 5.9: The TAR for each of the objects within $\mathbf{\Omega}$.

coverable fail-states, it is unlikely that the threshold applies a positive performance bias. Indeed, it is instead more likely that increasing the threshold would only *increase* the TAR, eventually reaching 100%. A waypoint threshold, that triggers a reset of the guidance system, would be useful to avoid getting stuck in areas with little to no visual features.

5.4.3 Time to Target

The time to target indicates how fast the participants were guided toward a target object. In this case, only successful searches are included. A cumulative distribution of the search times is shown in Figure 5.10. The distribution of these results has a mean of 11.4s and standard deviation of 4.01s, as shown in Table 5.2. Furthermore, Figure 5.10 indicates that the majority of targets were found in less than 15s. In comparison to the remotely-assisted VizWiz system (Bigham *et al.*, 2010) (mean 92s, standard deviation 37.7s), these results look very encouraging, although there might be variations in the case of participants with visual impairments.

5.4.4 Mean Device Displacement

The mean device displacement is a measure that indicates the total radial and linear movement of the device between subsequent target searches. Less angular and linear displacement indicates reduced physical exertion from the participant, which is a desirable outcome. The mean pan, elevation and linear displacement per target search are given in Table 5.3 and shown in Figure 5.11. Each dimension's mean value is 1.11 rad, 1.04 rad and 0.25 m for the pan, elevation and linear displacements respectively.

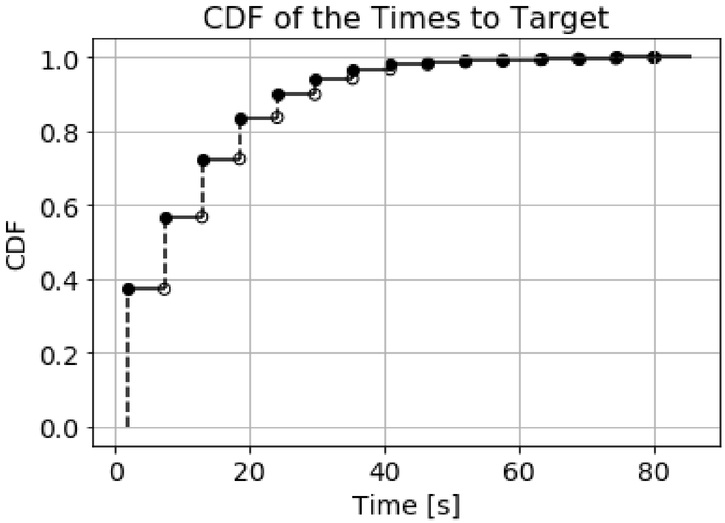


Figure 5.10: The cumulative distribution of the participants' time taken to find a target object.

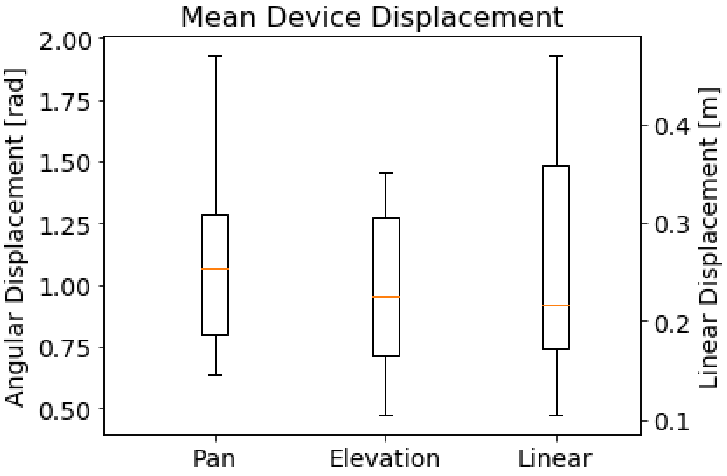


Figure 5.11: Box plots of the mean angular and linear displacements for the target search experiment.

In Figure 5.11, it can be seen that the mean pan displacement has a greater spread than the elevation dimension's results. This is expected, given that the object layout for the experiment is wider than it is tall (5 consecutive grid cells in the pan dimension vs. 4 in the elevation, see Figure 5.7b). Furthermore, the mean angular device displacement is well below the experiment environment's angular height and width of 2.1 rad (approximately 120°) each, which is an encouraging result. At this stage of the research, the significance of the linear displacement result is less clear, since there is no baseline to compare it to. However, it will prove useful for future comparisons against an updated active guidance system in Chapter 6.

5.4.5 Discussion

The results shown in Figure 5.8 show that approximately 75% of all the target searches ended with the participant finding the target object using at most 15 waypoints. Furthermore, the majority of the targets were found within 15s of the search start. On average, approximately 7 waypoints were generated per search, while the device was moved 0.25 m and rotated by approximately 1.11 rad and 1.04 rad in the pan and elevation dimensions respectively. These results are comparable to existing research works (Bigham *et al.*, 2010) and therefore validated the proposed guidance approach.

5.5 Conclusion

This chapter proposed and evaluated a new MDP-based system to guide a person towards a target object with no prior knowledge of the environment, using a mobile device camera. The MDP guidance controller uses the spatial relationships between objects, learned from the OpenImages dataset, to generate waypoints where the target object is most likely to be located. A successful policy was generated for this MDP using the SARSA algorithm and applied by the guidance controller to generate the optimal waypoint based on the camera's current and past viewing locations, as well as any observed objects. The system was implemented on an Android mobile phone and tested with several participants to determine its effectiveness.

The work presented in this chapter serves as an initial investigation into whether an MDP can effectively generate guidance instructions. A set of experiments with a simplified guidance interface and simulated object detector (QR code scanner) were conducted and indeed, the results show that the proposed solution produces effective guidance instructions, comparable in performance to similar existing systems. However, further work is needed to assist users with visual impairments in more realistic scenarios, where a real object detector and the audio interface are to be used. These aspects are investigated in the next chapter.

Chapter 6

Mobile Guidance System for Object Detection

The goal of this research is to determine whether a mobile guidance system with human-in-the-loop can effectively guide a person with visual impairments (PVI) to find target object object. To this end, an updated diagram of the proposed guidance system is shown in Figure 6.1. This includes a guidance control module and human-machine interface that communicates audio instructions, respectively. Both of these sub-components were designed, implemented and evaluated in Chapter 4 and Chapter 5.

The active guidance control module introduced in Chapter 5 is based on a Markov decision process (MDP) framework that assumes perfect state observability. In other words, it assumes that the sensor's inputs contain no noise or errors and that the agent (the guidance controller in this case) can perfectly estimate its state. This initial controller was implemented and tested with QR codes and a QR code scanner. Of course, to be a true guidance system, the QR codes and scanner must be replaced with real objects and an object detector. However, the perfect state observability assumption will no longer hold true when a real object detector is used, given that they are prone to detection and classification errors (represented by the signal ε in Figure 6.1). A vast amount of research focuses on reducing these errors, some of which are

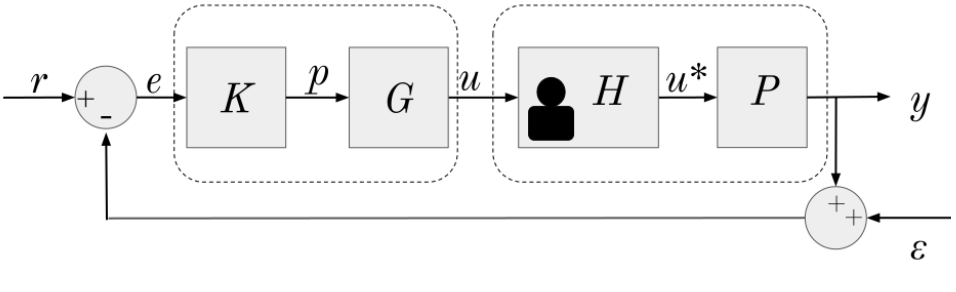


Figure 6.1: The system control loop, including the HMI, guidance controller and feedback noise, ε.

discussed in Chapter 2, but it is outside the scope of this thesis.

In this chapter, the existing MDP-based guidance controller is extended and updated to take sensor noise (i.e. object detection errors) into account. This is done in Section 6.1, after which the new, improved controller is integrated into a full Android app for a mobile phone, along with the bone-conduction audio interface and a state-of-the-art object detector in Section 6.2. The complete guidance system is then evaluated through experimentation with blindfolded and visually impaired participants in Section 6.3. This is followed by the experiment results, as well as a discussion on their significance in Section 6.4, before the chapter is concluded with a short summary and final remarks in Section 6.5. Part of the work presented in this chapter has been published in a conference paper (Lock *et al.*, 2019*d*). The source code used to create the app, object detector and audio interface has also been made freely available to the public[1].

6.1 Extending the Human Control Module

In this chapter, a computer vision-based object detector is integrated into the guidance system to replace the QR code scanner that was used previously. However, these detectors introduce noise and errors into the feedback loop that the controller must take into account during the waypoint generation process. In Figure 6.1, the object detector's error, ε, is added to the feedback loop as a noise signal that influences K's output.

To take this additional error into account, the MDP implemented in Chapter 5 is replaced by a partially observable MDP (POMDP, described in more detail in Section 2.4.2). POMDPs attempt to minimise some cost function by executing the optimal action in any given state, to reach a certain goal state. The cost is defined by a POMDP's reward function, which it tries to maximise. During a training process, an agent can learn the optimal action to take for any reachable state and produce a policy that captures this information. These principles are applied to the guidance controller to take the camera's view from some initial state to the target state, while minimising the effort required by the user. To do this, the POMDP controller follows a process similar to the MDP-based controller, using a grid- and waypoint-based guidance strategy to discretise the world and generate user instructions. The guidance controller uses the POMDP policy to select the next waypoint. Since the controller does not initially know the target object's true position, this waypoint is placed in a location that maximises the probability of the user pointing to it, moving increasingly closer to the target object. The audio interface uses this waypoint location to generate guidance instructions for the user to point the camera.

This guidance system shares many similarities with the previously-implemented MDP-based controller. However, the major differences here are the

[1] `https://github.com/yassiezar/POMDPObjectSearch`

use of a real object detector and a new POMDP implementation that removes the perfect state observability assumption. The following sub-sections describe how the previous MDP is replaced by the POMDP and how it was trained and implemented in the final active guidance system.

6.1.1 POMDP for Human Control

A POMDP model is described by the tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R}, \mathbf{Z}, \mathbf{O}, \mathbf{b}, \gamma \rangle$. Similar to an MDP, $\mathbf{S}$ represents a finite set of discrete states, $\mathbf{A}$ is a set of discrete actions, $\mathbf{T}$ is a matrix containing the probabilities of transitioning from state s to state s' (where $s, s' \in \mathbf{S}$) after executing action $a \in \mathbf{A}$, and $r \in \mathbf{R}$ is the reward the agent receives for executing a and reaching s'. The major difference between MDPs and POMDPs are the additional $\mathbf{Z}$ and $\mathbf{O}$ matrices. These elements respectively define the possible observations and their emission probabilities after transitioning to state s' with action a. Finally, $\mathbf{b}$ is the belief vector containing the state probability distribution and γ is a discount factor that prioritises long-term over short-term rewards, which affects the model's convergence rate.

As described in Section 2.4.2, a POMDP maintains a probability distribution over all the possible states. However, given its continuous nature, determining, updating and tracking this distribution can become computationally expensive and potentially intractable for moderately-sized state-spaces. The states must therefore be carefully selected to avoid unsolvable problems. If the model is well-designed, however, the POMDP can be transformed into an equivalent, so-called 'belief-MDP', where the probability distribution is transformed into a quasi-state parameter, called the 'belief'. This new state is fully observable by the agent at any time step. Note that the belief state reflects the agent's best guess about its current state, not necessarily its true state.

The possible states, actions and their transition probabilities are the same as the MDP in Chapter 5. However, new reward function and observation probabilities have to be implemented for this POMDP.

Reward Function

A similar rationale to Chapter 5 was used for designing the reward function for the POMDP. Specifically, the agent is punished for any action that does not lead the user to the target (progressively increasing the punishment after some threshold) and for leading the user to the same location more than once. The reward values were empirically determined, similar to the initial MDP implementation (see Table 6.1). In the new POMDP, the penalty given for every waypoint that does not lead to the target was significantly increased in order to make the model more reactive.

Table 6.1: The reward function for the POMDP model.

Reward condition	Reward
$r(o = o_{target})$	10000
$r(v = \text{true})$	-75
$r(n > n_{\max})$	-75
otherwise $r(\cdot)$	-100

Observations

The quantities $\zeta \in \mathbf{Z}$ represent the observations that an agent can make from state s. An observation in this context can simply be considered as a sensor reading. Since the only reading for this controller comes from the object detector, the observation is $\zeta = \langle o \rangle$, where $o \in \mathbf{\Omega}$ is one of the possible objects that are detectable by the system.

The observation matrix, $\mathbf{O}$, defines the probability of the agent making observation ζ after it has transitioned to state s' with action a. For example, a perfect, zero-noise sensor would produce an observation $\zeta = \langle o' \rangle$, where o' is the true object in state s'. Of course, this ideal case rarely applies in reality and the entries in the observation matrix encapsulates this uncertainty.

For the current implementation, the values in $\mathbf{O}$ were determined from the detection and classification errors of the object detector, as evaluated by Tramontano (2019). The implementation and performance evaluation of the detector are summarised in Section 6.2.

6.1.2 Policy Generation

The POMDP's optimal policy — which contains the optimal state-action mapping — is determined through an iterative training procedure where the agent is allowed to explore the entire state-action space. During this training process, the agent explores the state-space and iteratively improves the policy such that it maximises the cumulative reward it receives. For the current implementation, 16 objects were encoded into the POMDP, including a 'nothing' instance for cases where the camera does not observe anything of interest:

$$\begin{aligned}\mathbf{\Omega} = \{&nothing, monitor, keyboard, mouse, desk, laptop, mug, window,\\ &couch, lamp, backpack, chair, plant, telephone, whiteboard, door\}.\end{aligned} \tag{6.1.1}$$

A 6×6 grid, similar to the one implemented in Chapter 5, was used to discretise the agent's world and simplify the state tracking and transition processes. The maximum number of waypoints, $n_{\max}$, is also set to 11, which is the longest traversable path of waypoints in the 6×6 grid. When n_{max} is exceeded, the agent is penalised for every additional waypoint generated that does not lead to the target object, as defined by the reward function in Table 6.1. This results in a total of 352 reachable states ($k_{states} = 16 \times 11 \times 2$), with any state containing the target object being a terminal state ($15 \times 2 = 30$ in this case). The number of terminal states is double the number of objects, because the latter can be found in locations that the camera has visited before, but failed to detect the target object.

Since the belief vector is fully observable by the agent, the problem becomes a belief-MDP and can be solved with the existing solutions described in Section 2.4.2. However, given the belief vector's continuous nature and the potentially infinite state-observation combinations, finding an exact policy within a reasonable amount of time and with a practical file size is often not very efficient. Indeed, this was the case even for the current moderately-sized state space, ruling out the Value Iteration algorithm. Approximate algorithms are a viable alternative that offers a reasonable and potentially optimal solution for the POMDP considered here. In this case, the Point-Based Value Iteration (PBVI) (Pineau *et al.*, 2003) algorithm is used[2]. The PBVI algorithm is based on the familiar Value Iteration algorithm, but it tracks and updates only the values of a smaller representative subset of belief points. Using this algorithm, a total of 15 policies were generated, one for each object given in $\boldsymbol{\Omega}$.

6.2 Object Detector

A state-of-the-art object detector (Tramontano, 2019; Terreran *et al.*, 2020) was implemented and integrated into the guidance system to provide the controller with object observations. This was designed specifically for mobile devices, which typically have limited computing power. Furthermore, the detector works with (near) real-time processing performance and enables the implementation of a useful guidance system.

The image-based object recognition system uses SSD-Lite, which is a state-of-the-art single-stage object detection and classification network based on the SSD architecture and implements MobileNetV2 (Sandler *et al.*, 2018). It is a lightweight network specifically designed for mobile platforms, requiring relatively little memory to perform inference tasks. The increased computation efficiency from the SSD-Lite architecture is a product of a novel depth-wise separable convolution procedure, which reduces the total number of model parameters without affecting its effective depth when compared to a model that uses normal convolution operations (e.g. the standard SSD network). The reduced number of parameters does affect the model's classification accuracy, however. In their work, Tramontano (2019) compared initial implementations of SSD-Lite, Tiny-DSOD and YOLOv3 on a mobile platform and found that SSD-Lite outperformed YOLOv3 and Tiny-DSOD in terms of classification

[2] AI-Toolbox library implementation: `https://github.com/Svalorzen/AI-Toolbox`

accuracy and frame rate. For example, with their initial implementations, the YOLOv3 only achieved 2 frames per second, compared to SSD-Lite's 25, and Tiny-DSOD achieved a mAP of 6.3%, compared to SSD-Lite's 16 (this has since been fine-tuned to achieve a mAP of 33% on a subset of the OpenImage dataset (Terreran *et al.*, 2020)).

The model was trained with a total of 10,000 objects per class in $\mathbf{\Omega}$. Training with a smaller subset of object classes would make the model substantially smaller, speeding up object inference and increasing overall frames-per-second performance. However, this comes at the cost of generality. Each training sample was taken from the OpenImages dataset (Kuznetsova *et al.*, 2018), with a 60–20–20% split for training, validation and testing, respectively.

After the object detector was trained, its performance was further tested on a self-made 'office' dataset in order to evaluate its performance in a real environment that more accurately represents the experimental scenario for the guidance system. To generate this dataset, pictures of the 15 objects in $\mathbf{\Omega}$ were taken with the mobile phone's built-in camera. One instance was considered for each object class, e.g. the same laptop was used in all of the 'laptop' pictures, within a generic office environment. However, the pictures were taken from different perspectives, orientations and positions to simulate the challenges of detecting objects using a hand-held mobile camera.

Firstly, pictures were taken from three different points of view (see Figure 6.2), selected within the office environment to capture typical positions that the guidance system may be used from (e.g. from the door, from behind a desk, etc.). Then, for each point of view, pictures of the object were taken from various body-related positions to simulate different use-cases, such as when the mobile phone is held at pelvis, chest or head height. Finally, for each point of view and camera position, three different device orientations were adopted to simulate different device-holding habits. These orientations are $-45°$, $0°$ and $45°$, as measured from the vertical axis. This test set therefore includes 27 pictures per object, for a grand total of 405 images.

The POMDP's observation matrix, $\mathbf{O}$, was populated by the probability of the object detector correctly classifying a given object (taken as the number of correct classifications over the total number of samples in the reduced OpenImages set). The object detector was then converted into an Android-compatible Tensorflow-Lite[3] (TF-Lite) model. TF-Lite models are approximately equivalent to full TensorFlow models, but have been optimised specifically for object inference on mobile platforms and is compatible with the latest versions of Android. The TF-Lite API has the added benefit of being modular with regard to the TF-Lite model, which means that a new object detector can be implemented by simply loading a different pre-trained model file. This model was finally integrated into the mobile device, providing the POMDP guidance controller with object observations at approximately 20 frames per second.

[3] https://www.tensorflow.org/lite/

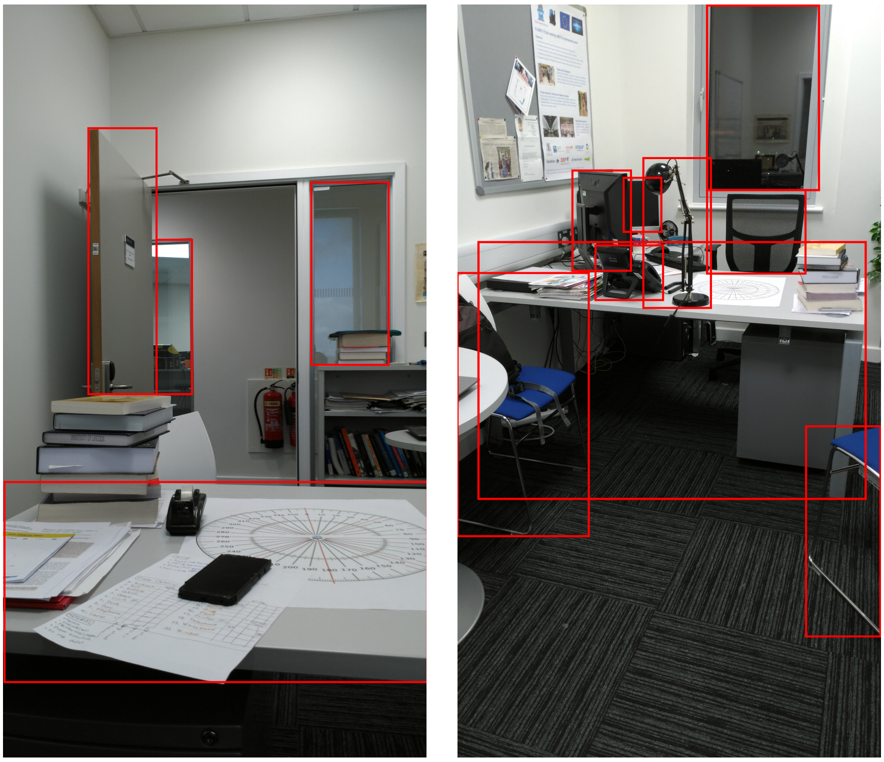


Figure 6.2: A sample of the 'office' dataset, showing pictures taken from two different points of view (Tramontano, 2019).

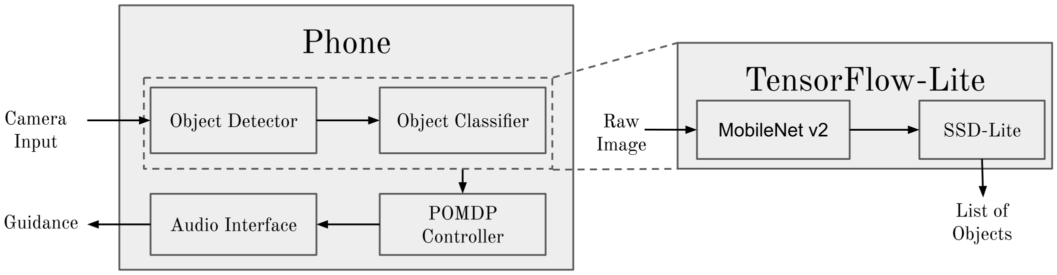


Figure 6.3: The internal pipeline of the guidance system showing the flow of information. The camera captures image data, which the object detector uses to produce a list of detected objects. The guidance controller then generates a waypoint, which the audio interface uses to generate an audio signal to guide the user.

Figure 6.3 shows the object detector's internal structure, including the MobileNetV2 and SSD-Lite models enclosed in a TF-Lite package and where it is implemented in the app pipeline.

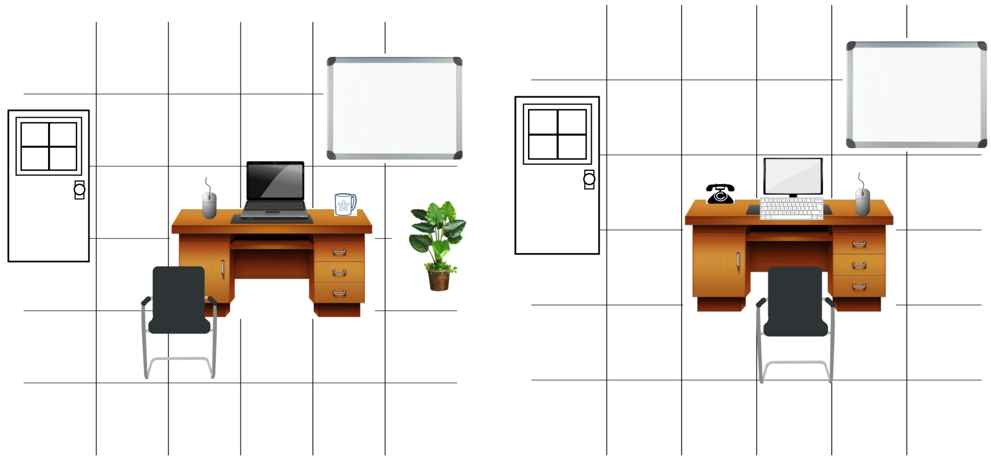
6.3 Experiments

The proposed active guidance system was implemented onto a mobile phone and evaluated with a set of experiments with blindfolded and visually impaired participants to gather performance data. In the following sub-sections, the details of the guidance system's implementation are discussed along with the experiment design.

6.3.1 Guidance System Implementation

A complete active guidance system was implemented by integrating the audio interface described in Chapter 4, with the object detector and POMDP guidance controller presented in this chapter. The audio interface conveys a target location using a spatialised audio signal and varying pitch through a set of bone-conduction headphones. The *hi* pitch gradient was used for its desirable performance characteristics (see Section 4.4). The waypoints are generated using the same approach used in Chapter 5, where the environment is discretised into a 6×6 grid to simplify the state estimation and waypoint generation processes. Target waypoints are generated at the centre of each grid cell, and in this case represent a 35° angular rotation, giving a 210° field of view. Moving into a new cell and reaching the waypoint triggers the generation of a new waypoint. The object detector discussed in Section 6.2 was implemented into a TF-Lite model. Its detection parameters were empirically tweaked to achieve the desired object detection and guidance characteristics.

These components were integrated and implemented in an Android app (the full data pipeline can be seen in Figure 6.3) for the Asus ZenPhone AR, using the ARCore API to track the device's pose. No further software was required for this app and the only additional hardware used is the set of AfterShokz bone-conduction headphones to transmit audio signals without blocking ambient noises (see Figure 3.4 for a picture). All the tasks (object detection, action lookup, guidance signal generation, etc.) were performed by the mobile device in real-time. The different components were integrated alongside each other as separate modules that interact with one another through standard communication pipelines implemented in the Android framework. This modular approach allows for modifications to be made to individual components, without affecting the overall system's functionality. For example, the object detection model can be replaced by any other TFLite-compatible model file and the app will start up and use the new model without any other modifications. A detailed class diagram of the full guidance system implementation can be found in Appendix B, which shows how the different components interact with one another.



(a)

(b)

Figure 6.4: Graphical representations of the environment layouts used in the guided and unguided experiments.

6.3.2 Experiment Design

To evaluate the complete system's effectiveness in guiding a PVI towards a target object, a set of experiments were conducted in a static environment. Additional experiments with an unguided version of the system provide a baseline for the active guidance results.

The environments for each experiment, guided and unguided, were modelled on a typical office scenario and care was taken to randomise layouts and object placements. If the same objects were present in both sets, they were placed in different positions. However, some of the larger, static objects (e.g. door, desk, etc.) appeared in both experiments. Note that the same policy for a given object can be used across both environments. The objects in the experiment with the guided version of the object search system are $\boldsymbol{\Omega}_g = \{\textit{door, desk, chair, whiteboard, mouse, laptop, backpack, mug}\}$. The objects with the experiment using the unguided object search system are $\boldsymbol{\Omega}_u = \{\textit{door, desk, chair, whiteboard, mouse, monitor, telephone, keyboard}\}$. See Figure 6.4a and Figure 6.4b for a graphical representation of the guided and unguided experiment environments, respectively.

In total, 16 participants were recruited for these experiments, including 10 blindfolded people recruited mostly from the School of Computer Science (Group *G1*, 8M, 2F, 33±12 years). The remaining 6 (Group *G2*, 6M, 44±6.7) general member of the public with visual impairments who are classified as severely sight impaired and legally blind.Of the latter group, 4 are congenitally blind, while the other 2 lost their sight later in life. These demographics are summarised in Table 6.2.

A time limit of 45 seconds was set for each experiment run, which ended either by finding the target object or by reaching the time limit. This limit

Table 6.2: A summary of the participant demographics.

	Group *G1*	Group *G2*
Gender [M/F]	8/10	6/0
Age [years]	33 ± 12	44 ± 6.7
Degree of Vision Impairment	N/A	4 totally blind, 2 with limited light perception
Experience with ETAs	None	None

was set following the results observed in Chapter 5, where more than 95% of the targets were found within 45 seconds, and limited the total time for each experiment to approximately one hour. There was one experiment run per target object, giving 8 experiment runs for each of the guided and the unguided scenarios.

6.3.3 Guidance System Experiment

In this experiment, the guidance system's performance at directing a user to a target object was evaluated. Here, all perception and guidance decisions are made by the mobile device, with the participant acting as the 'actuator', interpreting control signals from the system and moving the device accordingly (as shown in Figure 6.1). In this case, the participants were not told what the target object was, but they were informed that the object was found when the device vibrated. This helped to focus only on the performance of the guidance system, reducing possible biases due to user common-sense and a-priori knowledge of typical object placement locations. An experiment run ended when the target object was detected by the camera, or the 45 second time limit was reached. Figure 6.5 shows a participant with impaired vision participating in one of the experiments.

6.3.4 Baseline System Experiment

For this experiment, an unguided object detector told the user which objects were within the camera's view, using vocal feedback, when they tapped the device's screen. It was left to the participants to decide how to manipulate the camera and find the target object without any additional guidance. In this case, the baseline system is similar to other commercially available apps, such as SeeingAI[4] and TapTapSee[5], which only read out the current objects upon user's request. However, given these apps' fundamental differences to the guidance system, including the lack of control over its object detection process,

[4] http://www.microsoft.com/en-us/ai/seeing-ai

[5] http://taptapseeapp.com

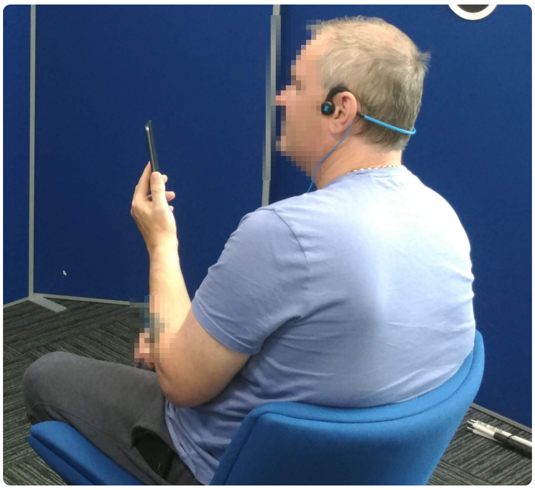


Figure 6.5: A participant with visual impairments during an experiment run. Note that this participant was congenitally totally blind and was therefore not blindfolded.

these apps were not suitable to be used as a baseline for these experiments. For the object search system used here as baseline, the device vibrated when the target object was detected and correctly classified to inform the participant that the target was found and concluding the experiment run. Of course, in this case the participants knew in advance which objects they were looking for and were therefore able to exploit their prior knowledge of typical object placements.

6.4 Results

Similar to Chapter 5, a number of important metrics are used to determine the performance of the guidance and baseline systems. These are the target acquisition rate (TAR), the time to target and the total device displacement. Each of the results collected from both experiments (guided and unguided) and groups are listed in Table 6.3, alongside statistical test scores that describe the statistical significance of the comparison. The Wilcoxon test is the most appropriate statistical test to use in this case, given the repeated measures experiment process used and the non-normal distributions of the data (see Appendix A for Shapiro-Wilk test results). The commonly used p value of 0.05 was used to evaluate the data's statistical significance. The following sub-sections present the results collected according to these metrics.

6.4.1 Target Acquisition Rate

The TAR is a measure of how successful each guidance system is at directing a participant towards a target. Similar to the values calculated in Chapter 5, the

Table 6.3: A summary of the experiment results for the blindfolded and blind participants (Groups *G1* and *G2*, respectively), as well as the two group's consolidated results. Except for the TAR, the values listed here are the median of each participant's mean.

	Test	**Guided**	**Unguided**	**Wilcoxon Statistic**
G1	TAR [%]	50 ± 19.7	50 ± 17	0.68
	Time to target [s]	12.4 ± 7.12	13.9 ± 6.92	0.44
	Pan Displacement [rad]	3.16 ± 6.1	9.21 ± 2.6	0.11
	Elevation Displacement [rad]	9.29 ± 6.4	10.8 ± 2.5	0.11
	Linear Displacement [m]	2.42 ± 1.5	2.60 ± 0.5	0.51
G2	TAR [%]	41.4 ± 15.7	26.7 ± 22.5	0.35
	Time to target [s]	11.9 ± 9.59	10.8 ± 13.5	0.35
	Pan Displacement [rad]	6.22 ± 6.7	9.12 ± 4.1	0.60
	Elevation Displacement [rad]	12.5 ± 6.2	13.6 ± 7.0	0.75
	Linear Displacement [m]	2.17 ± 0.6	2.77 ± 1.2	0.75
G1 + G2	TAR [%]	44.4 ± 18.6	37.5 ± 20.8	0.38
	Time to target [s]	12.4 ± 8.20	13.4 ± 10.1	0.21
	Pan Displacement [rad]	4.21 ± 6.5	9.21 ± 3.3	0.11
	Elevation Displacement [rad]	8.57 ± 6.5	11.6 ± 5.0	0.16
	Linear Displacement [m]	2.28 ± 1.3	2.60 ± 0.9	0.60

TAR is taken as the proportion of successful object searches completed within 45 seconds. In this case, a high value indicates a large number of targets found. Figure 6.6 shows the TAR distribution for each experiment and group.

As seen in Table 6.3, the median TAR for Group *G1* (blindfolded) is evenly split between the two experiments (i.e. 50% vs. 50% for the guided and unguided experiments, respectively). The results for Group *G2* (the legally blind group) show a more significant improvement of approximately 14.7% in favour of the guidance system (41.4% vs. 26.7% for the guided and unguided experiments, respectively). However, the Wilcoxon test ($F = 6.0, p = 0.35$) shows that this difference is not statistically significant. When considering the results for the entire sample (*G1* + *G2*), the guidance system's improvement is recorded as 6.9% ($F = 44.5, p = 0.38$). The data currently available do not conclusively show which system is better, given the lack of statistical significance. However, the results for the guidance system are encouraging, with moderate improvements in at least one group and increasing significance with the entire sample.

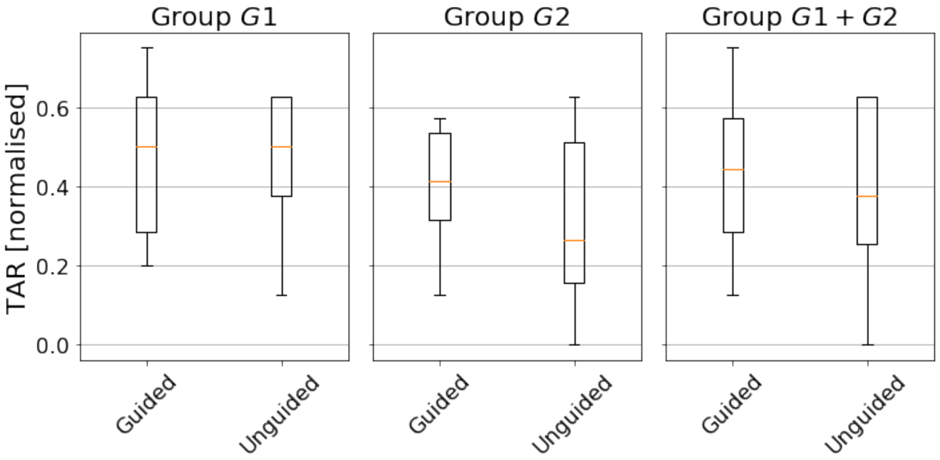


Figure 6.6: A set of box plots showing the TAR results for Group *G1* and *G2* for the guided and baseline experiments. A distribution that contains both groups is also included.

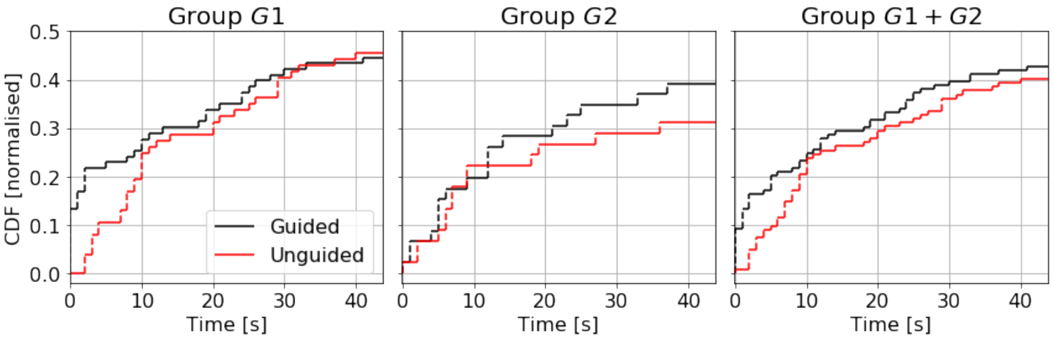


Figure 6.7: Cumulative distribution functions of the time to target results for each group and experiment.

6.4.2 Time-to-target

The time it takes to find a target object is an important indicator of system performance. These time-to-target results are presented in Figure 6.7, which shows the cumulative distributions for each group and experiment for all searches. The results show that the guidance system improved the participants' overall time-to-target performance by a moderate margin. Group *G1* shows a consistent improvement, particularly before 10s, after which it tapers off and becomes equivalent to the unguided system after 30s. However, Group *G2* shows a more gradual increase in performance as time goes on, being similar to the unguided system until approximately 12s, at which point the guidance system starts outperforming the unguided one. For the overall sample including both groups, the guidance system consistently outperforms the unguided baseline across the entire timespan.

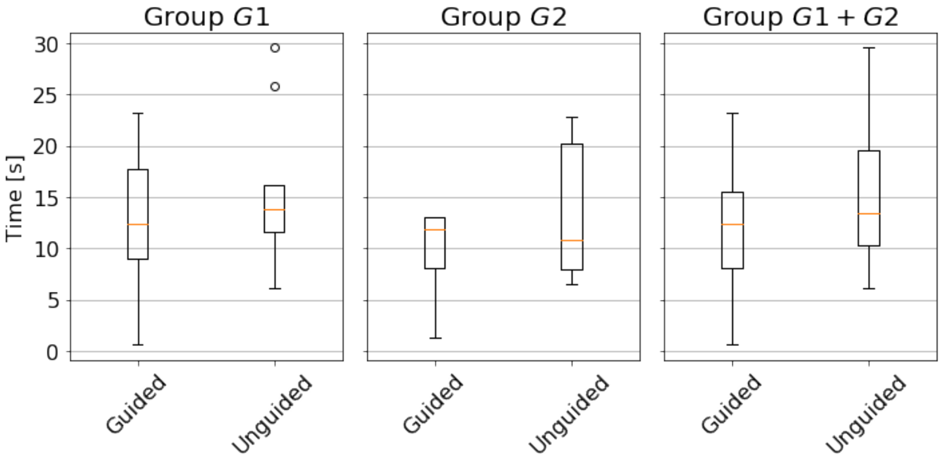


Figure 6.8: Box plots of the time to target results for each group and experiment. These distributions include only the cases where participants found the target objects.

When considering only successful searches where the target was found within 45 seconds, the performance differences become clearer (see Figure 6.8). The participants in Group *G1* managed to find the targets with a median time of 12.5s vs. 13.9s for the guided and unguided systems, respectively (Wilcoxon $F = 16.0, p = 0.44$). For Group *G2*, this trend seems reversed with medians of 11.9s vs. 10.8s ($F = 6.0, p = 0.35$) for the guided and unguided experiments, respectively. However, there is a large spread in the results, with long tails to the upper and lower ends of the time scale. This observation, in addition to the improvement of the combined results for Groups *G1* and *G2* with the guidance system (12.4 vs. 13.4, $F = 38.0, p = 0.21$), suggest that the guided system could potentially outperform the unguided one even for Group *G2*, if a larger sample of participants with visual impairments were available.

6.4.3 Device Displacement

The device's displacement data indicate the effort required to find each target. Less displacement is desirable, since it means less physical exertion is required for the user to find an object. The total displacement in the angular (pan, elevation) and linear dimensions for each search is given in Figure 6.9.

The box plots for Group *G1* show a consistent, albeit non-statistically significant, reduction in the angular displacement per target search with the guidance system for both the pan (group medians of 3.16 rad vs. 9.21 rad, Wilcoxon statistic $F = 9.0, p = 0.11$) and elevation dimensions (9.29 rad vs. 10.8 rad, Wilcoxon statistic $F = 9.0, p = 0.11$). There is some improvement in the linear dimension as well, with median displacements of 2.42 m vs. 2.60 m (Wilcoxon statistic $F = 17.0, p = 0.51$). Group *G2*'s displacement performance also improved when the guidance system was used (median pan and

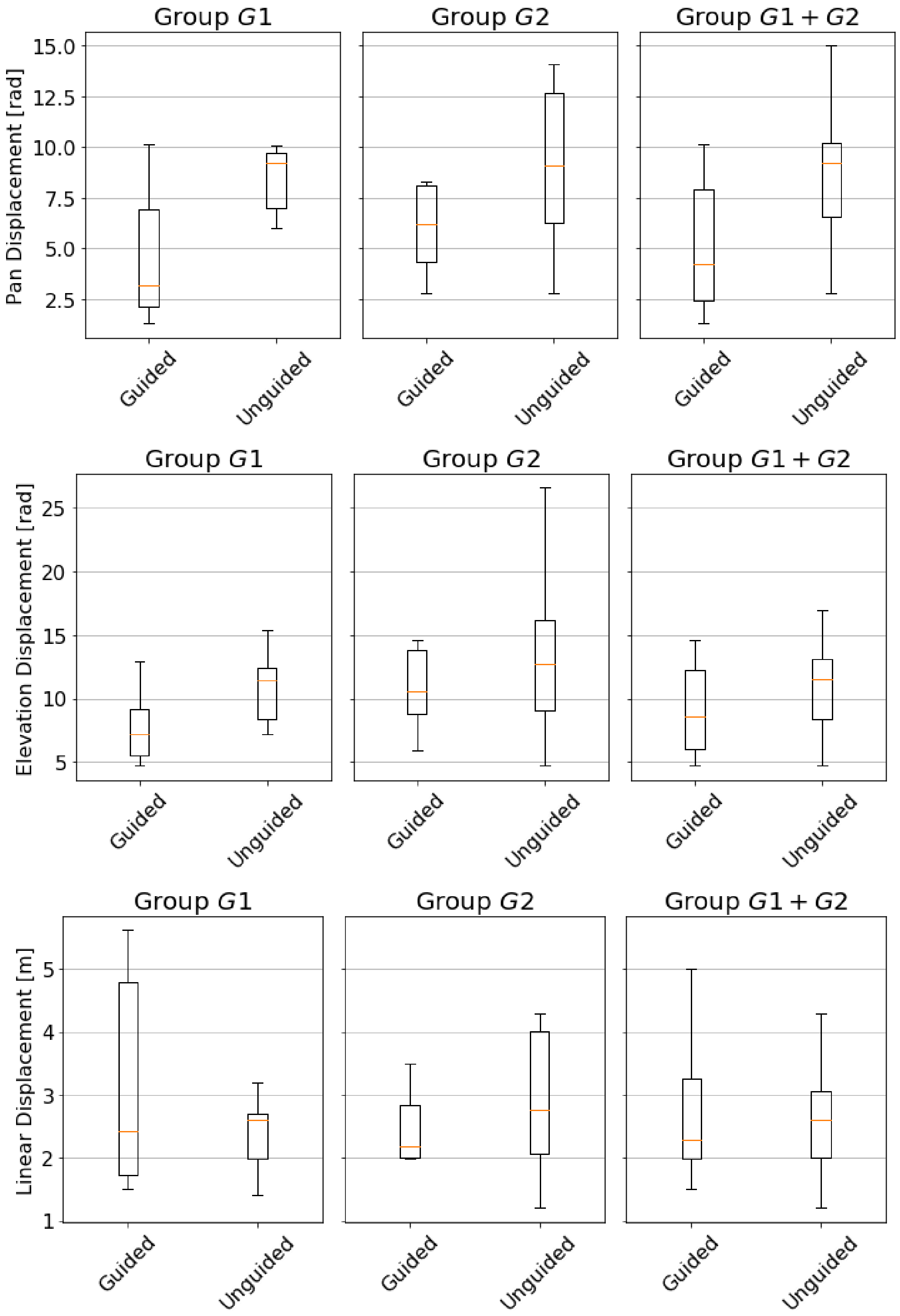


Figure 6.9: Box plots showing the total device displacement in the pan, elevation and linear dimensions for each search experiment and group.

elevation displacements of 6.2 rad vs. 9.1 rad and 12.5 rad vs. 13.6 rad for the guided and unguided experiments, respectively, linear displacement medians of 2.17 m vs. 2.77 m). The results for the medians of the entire sample with Groups *G1* and *G2* combined shows an improvement across all three displacement dimensions. These results are 4.21 rad vs. 9.21 rad, 8.57 rad vs. 11.6 rad, 2.28 m vs. 2.60 m for the pan, elevation and linear dimensions for the guided and unguided experiments, respectively. Although not statistically significant, the use of the guidance system shows a general trend of improvement in total device displacement with respect to the unguided case.

6.4.4 Discussion

The results from these experiments show that, except for the time-to-target with Group *G2*, the guidance system performed better than the unguided one for both groups. These performance differences are not extreme and have been observed in other works (see Section 4.4 and Passini & Proulx (1988)), but it is not clear why there are any differences. One possible explanation is the age difference between the groups and the general level of familiarity and comfort with mobile technology between the groups (Group *G1* was recruited from a technically qualified population). Considering the entire sample, the guidance system gives better performance for all the metrics. Unfortunately, some of these comparisons are not statistically significant, potentially due to the small sample size caused by the impossibility of recruiting a larger number of participants with visual impairments. This limitation is further analysed by considering the effect size and the proportion of improvement shown in Table 6.4 and discussed next.

Effect size is a quantitative measure of the magnitude of the difference between two groups, where a larger number indicates a stronger effect from an external stimulus. In this case, the stimulus is the guidance system that provides the user with navigation instructions. If the two distributions are normally spread, the effect size, d, can be determined as follows:

$$\begin{aligned} d &= \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2+\sigma_2^2}{2}}} \cdot \kappa, \\ \kappa &= \frac{N-3}{N-2.25} \cdot \sqrt{\frac{N-2}{N}}, \end{aligned} \tag{6.4.1}$$

where μ and σ represent each group's mean and standard deviation and κ is a correction factor for small sample sizes (N in this case). However, if the data are non-normal, then the Wilcoxon test statistic,

$$d = \frac{F}{\sqrt{N}}, \tag{6.4.2}$$

Table 6.4: A table containing the results for each metric's performance comparison.

	Metric	**Effect Size**	**Proportion Improved**
	TAR	−0.10	55.5%
	Time	0.68	66.7%
G1	Pan	0.53	88.8%
	Elevation	0.44	88.9%
	Linear	−0.75	44.4%
	TAR	−0.62	66.7%
	Time	0.38	66.7%
G2	Pan	0.21	66.7%
	Elevation	0.24	50%
	Linear	0.62	50%
	TAR	0.29	60%
	Time	0.33	60%
G1 + *G2*	Pan	0.41	80%
	Elevation	0.32	73.3%
	Linear	0.13	46.7%

is used to calculate the effect size (F is the test statistic value). In the experiments, the effect sizes range from weak (0.1 for Group *G1*'s TAR and *G1* + *G2*'s linear displacement) to strong (0.75 for *G1*'s elevation). The mean effect sizes for each group is 0.5 and 0.41 for *G1* and *G2* respectively, while the entire sample's mean effect size is 0.31. These values are typically considered as moderate effect sizes (Keppel, 1991).

In addition, the repeated measures process used for these experiments (i.e. the same participant performed both the guided and unguided experiment) allows a simple proportional measure to be used to indicate the significance of the guidance system's effect on the performance measures. This proportion is simply taken as the number of participants that improved their performance with the guidance system, listed in Table 6.4 (the complete results, including the normality test results, is given in Appendix A). It can be seen that the guidance system outperformed the unguided one in 9 out of 10 comparisons for Groups *G1* and *G2*, and 4 out of 5 for *G1* + *G2*.

These considerations suggest that further experiments with a larger sample size (which was unfortunately unavailable at the time of this research) could yield similar, but more statistically significant results. In particular, using the effect sizes list in Table 6.4 and the power tables generated by Cohen (2013), a minimum of 20 participants with visual impairments is recommended to achieve more significant results at the recommended 80% power level.

6.5 Conclusion

In this chapter, a POMDP-based guidance controller is proposed that improves upon the previous MDP-based solution, which is capable of handling uncertain state observations. This controller was implemented alongside the audio interface of Chapter 4 and a state-of-the-art object detector to create a fully-integrated mobile object search and guidance prototype. This implementation was evaluated through a set of real object search experiments with blindfolded participants and PVI. For these experiments, the guidance system was compared to an unguided one that is functionally similar to apps currently available on the market. It was shown that the guidance system improved the participants' target finding performance, in particular for the group of PVI. Although a larger sample size would benefit the statistical significance for the results of future studies.

Chapter 7

Conclusions

This thesis investigated the feasibility and effectiveness of an active vision system with human-in-the-loop that guides a user in a pointing task. To address this research problem, an active guidance system was developed to assist people with visual impairments (PVI) in finding objects within an unknown indoor environment. The system was implemented on a mobile phone, along with a bone-conduction audio interface and a vision-based object detector. Several experiments with blindfolded participants and PVI showed that such a solution compares favourably to approaches used by alternative object search apps that are currently available for PVI, although further studies with a larger sample size of PVI would provide more definitive results.

This chapter presents a summary of the research conducted for this thesis, including the main findings and contributions in Section 7.1. The current limitations are discussed in Section 7.2, alongside recommendations on how to address them and potential future research prospects in Section 7.3.

7.1 Summary of Research Contributions

Current ETAs for PVI are typically limited to well-defined and labelled areas and are often too bulky or unappealing to its target demographic to achieve significant market penetration. A number of different guidance interfaces have been implemented, with spatialised audio being particularly suitable given its relatively high bandwidth, flexibility and reduced cognitive load. Previous research into active vision has yielded promising results, but existing works in this area are still relatively sparse.

Two significant limitations identified in the literature are the lack of user acceptance and generality of current ETAs. In this thesis, an active guidance system is proposed that attempts to address these issues. User acceptance issues largely stem from a fear of public ridicule (i.e. ‘standing out from the crowd’). The proposed system addresses this with a guidance system implemented on a mobile phone, which uses its camera to gather environmental information and generate audio instructions. Furthermore, the proposed guidance system is generalised (i.e. useable in different unknown environments), providing a framework to actively control the behaviour of a person in a challenging perception task; in this case helping a PVI to find an object. The proposed system design was published in Lock *et al.* (2017).

The guidance interface uses audio instructions transmitted via bone-conduction headphones. The audio signals are spatialised in the lateral plane to convey a target location's pan angle relative to the device, while the elevation is conveyed by adjusting the tone's pitch. This new interface was implemented and its effectiveness evaluated with a set of experiments with a large number of users, showing good performance with respect to other audio interfaces with over-ear headphones. The research contributions of this new spatialised audio interface with bone-conduction headphones were published in Lock *et al.* (2019*b*) and accepted, pending revisions, for publication in the ACM Transaction of Accessible Computing journal (Lock *et al.*, 2019*c*).

An active-vision based guidance controller is proposed that uses the mobile phone's camera feed as input and generates guidance instructions that lead to the target object. An initial version of this guidance system was implemented with an Markov decision process-based (MDP) controller and tested on objects simulated by QR codes through a number of user-based experiments. The research behind this new active controller for visual object search was published in Lock *et al.* (2019*a*).

The guidance system was finally completed by replacing the QR code scanner with a real object detector and by integrating the audio interface. This required a redesign of the active controller to take into account the uncertainty of the object observations with a partially observable MDP (POMDP). The two new components (i.e. object detector and POMDP controller) were integrated alongside the bone-conduction audio interface into an app for mobile phones. A set of experiments were then conducted with a group of blindfolded participants and PVI. It was found that the active guidance system generally performed better than a simple unguided object search. The research contributions of this guidance system was published in Lock *et al.* (2019*d*).

7.2 Current Limitations

The research described in this thesis delivered an active guidance system for PVI. The evaluation experiments showed that users performed better with the guidance system than with the unguided one. However, the results lack robust statistical significance. Further analysis suggests that a larger sample size could remedy this issue, in particular by repeating the final experiments with approximately 20 or more PVI. Unfortunately, despite the involvement of and support from local charity organisations, particularly the South Lincolnshire

Blind Society[1] and the Lincolnshire Sensory Institute[2], recruiting significant numbers of such participants has proven extremely difficult, especially due to privacy and safety concerns.

The object detector implemented in the guidance system is based on a state-of-the-art neural network for mobile platforms with limited computing power. However, while it gave almost real-time detections with adequate classification performance in a controlled experimental environment, its low precision is not ideal for real-world applications. This type of object detector is currently the topic of much research in machine vision and will likely improve over time. Thanks to the modular design of the proposed solution, these improved detection algorithms could easily be re-integrated into the guidance system and significantly boost its performance.

7.3 Future Research Opportunities

The guidance system is currently based on a hand-held mobile phone. However, egocentric wearable camera systems, such as Google Glass[3] and Microsoft's Hololense[4] systems, would also be ideal platforms to implement such a system (Damen *et al.*, 2016). These systems are worn like glasses and will therefore free up the user's hand. Furthermore, they could potentially make guidance simpler and more intuitive, since all of the instructions are effectively given relative to the user's head instead of their hand. However, despite efforts to make these wearables more discreet and appealing, there are still concerns about user acceptability, which indeed have hindered their widespread use among the general public.

In addition to wearable cameras, it would be helpful to improve the audio interface transmitting guidance instructions to a PVI. However, there is a great deal of variability within this population in terms of level of vision impairment, ranging from limited light perception to complete blindness, which naturally leads to different needs from the guidance interface. It would therefore be helpful to have a configurable interface that allows the user to select, for example, the sine wave used in this research or a richer, more natural one, such as a musical instrument. An interesting evolution of these improvements would be to automate the entire interface configuration process. For example, if the guidance system was able to record and automatically evaluate its performance over time, it could change its own interface parameters to reach some optimal configuration. The Fitts's Law relationship discussed in Chapter 4 provides a good performance metric and can therefore be used to optimise such a co-adaptation process. A possible scenario would be a person

[1] `http://www.blind-society.org.uk/`

[2] `http://www.lincolnshiresensoryservices.org.uk/`

[3] `https://www.google.com/glass/start/`

[4] `https://www.microsoft.com/en-us/hololens`

with extreme near-sightedness who can find nearby objects: in this case the interface could gradually reduce the guidance intensity as the user gets closer to them. This concept of co-adaptation would extend the idea originally proposed by Gallina *et al.* (2015) and apply it to active mobile technologies, such as the one presented in this thesis.

Finally, the existing object detector and POMDP models could be improved to include a larger number of objects and tested in a dynamic environment with moving people or objects. It would also be useful to integrate an additional classifier to recognise the current location based on all of the objects found during a search session. This could provide the initial POMDP with supplementary meta-observations to refine its state estimation and subsequent decisions. For example, if the guidance controller believes it is currently in a kitchen, it would be able to more reliably disregard incorrect object classifications, such as a car (unlikely to appear in a kitchen). This would provide the guidance system with a high level, context-based mechanism for object detection, making it more effective and robust in different environments.

Appendices

Appendix A

Table A.1: A table containing the full set of results for the experiments conducted in Chapter 6.

	Metric	**Shapiro-Wilk**	**Effect Size**	**Proportion Improved**
	TAR	$G : F = 0.92, p = 0.42$ $U : F = 0.85, p = 0.08$	-0.10	55.5%
	Time	$G : F = 0.96, p = 0.89$ $U : F = 0.89, p = 0.19$	0.68	66.7%
G1	Pan	$G : F = 0.71, p = 0.002$ $U : F = 0.86, p = 0.12$	0.53	88.8%
	Elevation	$G : F = 0.69, p = 0.001$ $U : F = 0.93, p = 0.48$	0.44	88.9%
	Linear	$G : F = 0.87, p = 0.23$ $U : F = 0.95, p = 0.79$	-0.75	44.4%
	TAR	$G : F = 0.92, p = 0.52$ $U : F = 0.93, p = 0.64$	-0.62	66.7%
	Time	$G : F = 0.87, p = 0.21$ $U : F = 0.78, p = 0.04$	0.38	66.7%
G2	Pan	$G : F = 0.76, p = 0.02$ $U : F = 0.95, p = 0.74$	0.21	66.7%
	Elevation	$G : F = 0.87, p = 0.23$ $U : F = 0.95, p = 0.79$	0.24	50%
	Linear	$G : F = 0.81, p = 0.07$ $U : F = 0.91, p = 0.45$	0.62	50%
	TAR	$G : F = 0.96, p = 0.72$ $U : F = 0.88, p = 0.05$	0.29	60%
	Time	$G : F = 0.94, p = 0.48$ $U : F = 0.83, p = 0.009$	0.33	60%
G1 + G2	Pan	$G : F = 0.75, p < 0.001$ $U : F = 0.96, p = 0.73$	0.41	80%
	Elevation	$G : F = 0.78, p = 0.002$ $U : F = 0.88, p = 0.04$	0.32	73.3%
	Linear	$G : F = 0.78, p = 0.002$ $U : F = 0.88, p = 0.04$	0.13	46.7%

Appendix B

Guidance App Implementation

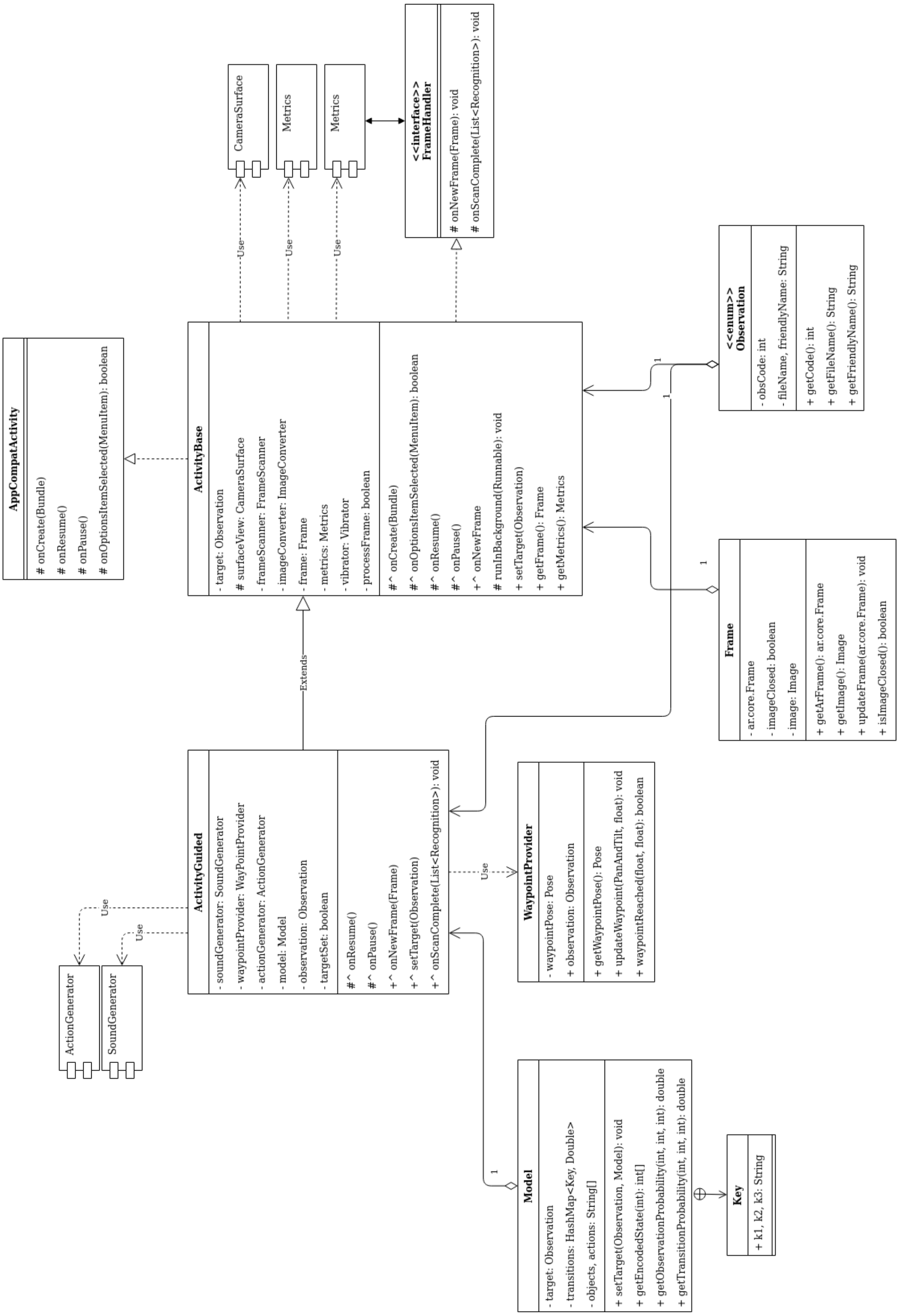

AppCompatActivity
# onCreate(Bundle)
# onResume()
# onPause()
# onOptionsItemSelected(MenuItem): boolean
ActivityBase
- target: Observation
# surfaceView: CameraSurface
- frameScanner: FrameScanner
- imageConverter: ImageConverter
- frame: Frame
- metrics: Metrics
- vibrator: Vibrator
- processFrame: boolean
#^ onCreate(Bundle)
#^ onOptionsItemSelected(MenuItem): boolean
#^ onResume()
#^ onPause()
+^ onNewFrame
# runInBackground(Runnable): void
+ setTarget(Observation)
+ getFrame(): Frame
+ getMetrics(): Metrics
CameraSurface
Metrics
Metrics
Use
Use
Use
<<interface>>
FrameHandler
# onNewFrame(Frame): void
# onScanComplete(List<Recognition>): void
<<enum>>
Observation
- obsCode: int
- fileName, friendlyName: String
+ getCode(): int
+ getFileName(): String
+ getFriendlyName(): String
Frame
- ar.core.Frame
- imageClosed: boolean
- image: Image
+ getArFrame(): ar.core.Frame
+ getImage(): Image
+ updateFrame(ar.core.Frame): void
+ isImageClosed(): boolean
1
1
1
Extends
ActivityGuided
- soundGenerator: SoundGenerator
- waypointProvider: WayPointProvider
- actionGenerator: ActionGenerator
- model: Model
- observation: Observation
- targetSet: boolean
#^ onResume()
#^ onPause()
+^ onNewFrame(Frame)
+^ setTarget(Observation)
+^ onScanComplete(List<Recognition>): void
Use
Use
ActionGenerator
SoundGenerator
Use
WaypointProvider
- waypointPose: Pose
+ observation: Observation
+ getWaypointPose(): Pose
+ updateWaypoint(PanAndTilt, float): void
+ waypointReached(float, float): boolean
Model
- target: Observation
- transitions: HashMap<Key, Double>
- objects, actions: String[]
+ setTarget(Observation, Model): void
+ getEncodedState(int): int[]
+ getObservationProbability(int, int, int): double
+ getTransitionProbability(int, int, int): double
1
Key
+ k1, k2, k3: String

Figure B.1: A class diagram of the object search guidance system as it is implemented in an Android app.

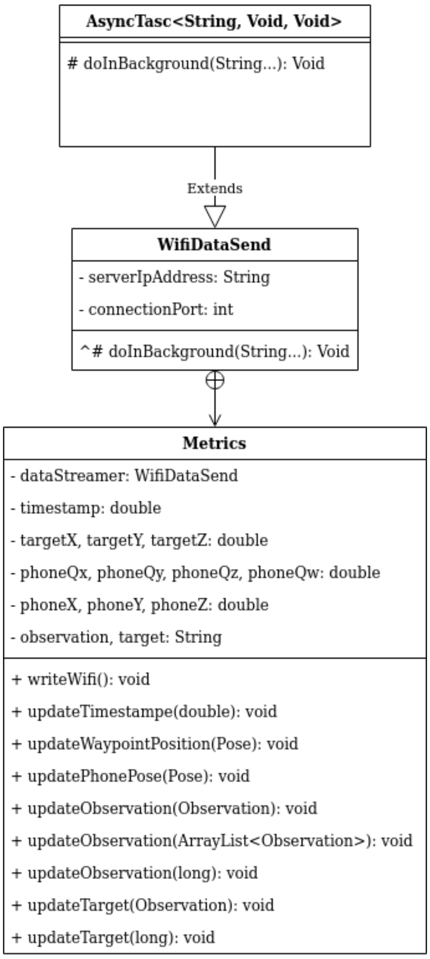

AsyncTasc<String, Void, Void>
# doInBackground(String...): Void
Extends
WifiDataSend
- serverIpAddress: String
- connectionPort: int
^# doInBackground(String...): Void
Metrics
- dataStreamer: WifiDataSend
- timestamp: double
- targetX, targetY, targetZ: double
- phoneQx, phoneQy, phoneQz, phoneQw: double
- phoneX, phoneY, phoneZ: double
- observation, target: String
+ writeWifi(): void
+ updateTimestampe(double): void
+ updateWaypointPosition(Pose): void
+ updatePhonePose(Pose): void
+ updateObservation(Observation): void
+ updateObservation(ArrayList<Observation>): void
+ updateObservation(long): void
+ updateTarget(Observation): void
+ updateTarget(long): void

Figure B.2: A diagram of the Metrics module responsible for recording the device's and targets' pose information.

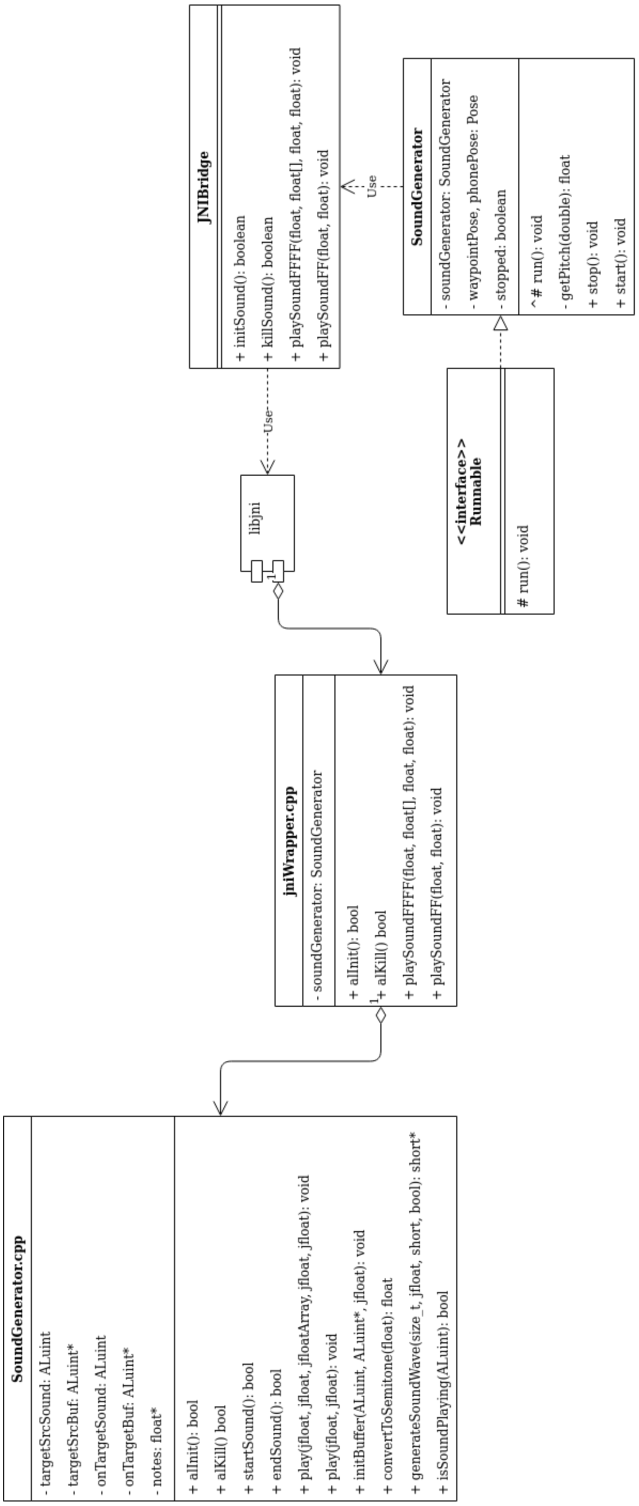

JNIBridge
+ initSound(): boolean
+ killSound(): boolean
+ playSoundFFFF(float, float[], float, float): void
+ playSoundFF(float, float): void
Use
SoundGenerator
- soundGenerator: SoundGenerator
- waypointPose, phonePose: Pose
- stopped: boolean
^# run(): void
- getPitch(double): float
+ stop(): void
+ start(): void
Use
libjni
<<interface>>
Runnable
# run(): void
jniWrapper.cpp
- soundGenerator: SoundGenerator
+ alInit(): bool
+ alKill() bool
+ playSoundFFFF(float, float[], float, float): void
+ playSoundFF(float, float): void
1
SoundGenerator.cpp
- targetSrcSound: ALuint
- targetSrcBuf: ALuint*
- onTargetSound: ALuint
- onTargetBuf: ALuint*
- notes: float*
+ alInit(): bool
+ alKill() bool
+ startSound(): bool
+ endSound(): bool
+ play(jfloat, jfloat, jfloatArray, jfloat, jfloat): void
+ play(jfloat, jfloat): void
+ initBuffer(ALuint, ALuint*, jfloat): void
+ convertToSemitone(float): float
+ generateSoundWave(size_t, jfloat, short, bool): short*
+ isSoundPlaying(ALuint): bool

Figure B.3: A diagram of the sound generation module, responsible for generating the audio-based guidance signals.

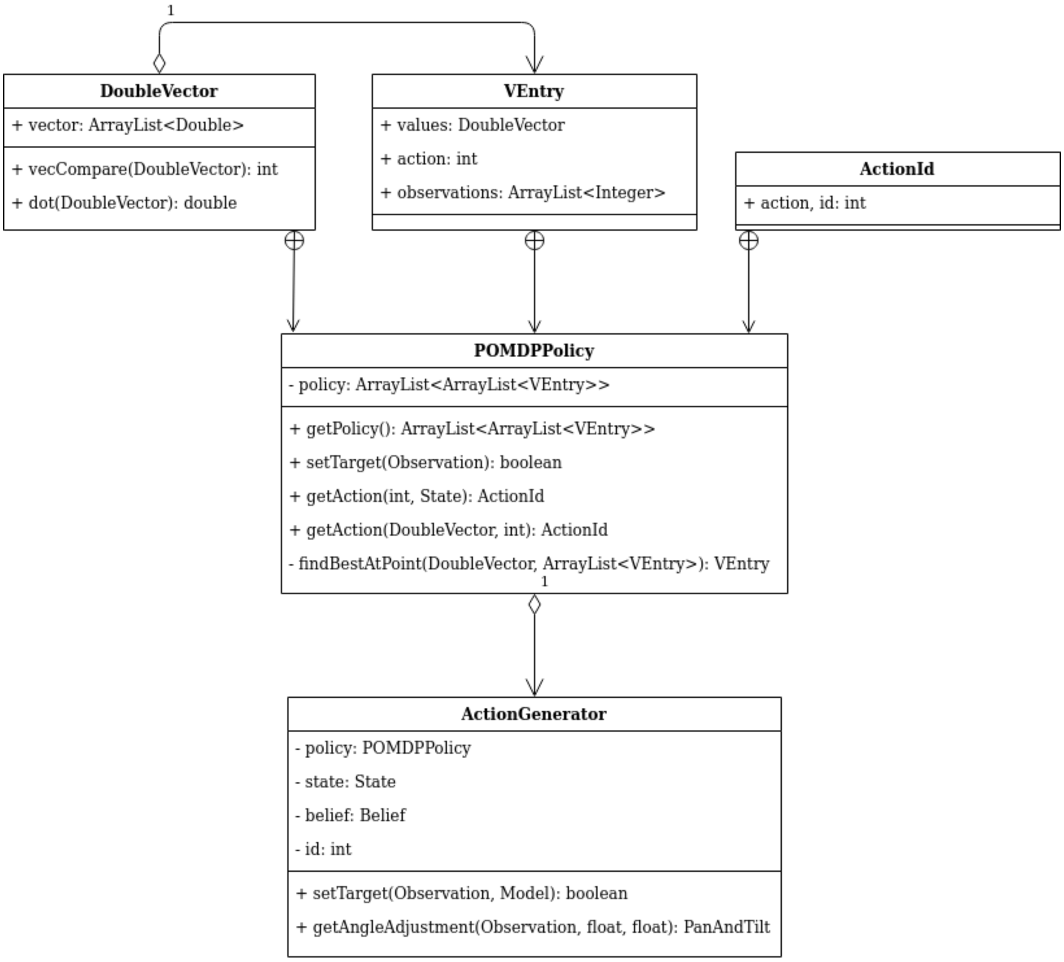

1
DoubleVector
+ vector: ArrayList<Double>
+ vecCompare(DoubleVector): int
+ dot(DoubleVector): double
VEntry
+ values: DoubleVector
+ action: int
+ observations: ArrayList<Integer>
ActionId
+ action, id: int
POMDPPolicy
- policy: ArrayList<ArrayList<VEntry>>
+ getPolicy(): ArrayList<ArrayList<VEntry>>
+ setTarget(Observation): boolean
+ getAction(int, State): ActionId
+ getAction(DoubleVector, int): ActionId
- findBestAtPoint(DoubleVector, ArrayList<VEntry>): VEntry
1
ActionGenerator
- policy: POMDPPolicy
- state: State
- belief: Belief
- id: int
+ setTarget(Observation, Model): boolean
+ getAngleAdjustment(Observation, float, float): PanAndTilt

Figure B.4: A diagram of the module responsible for looking up the optimal action to execute from the policy file.

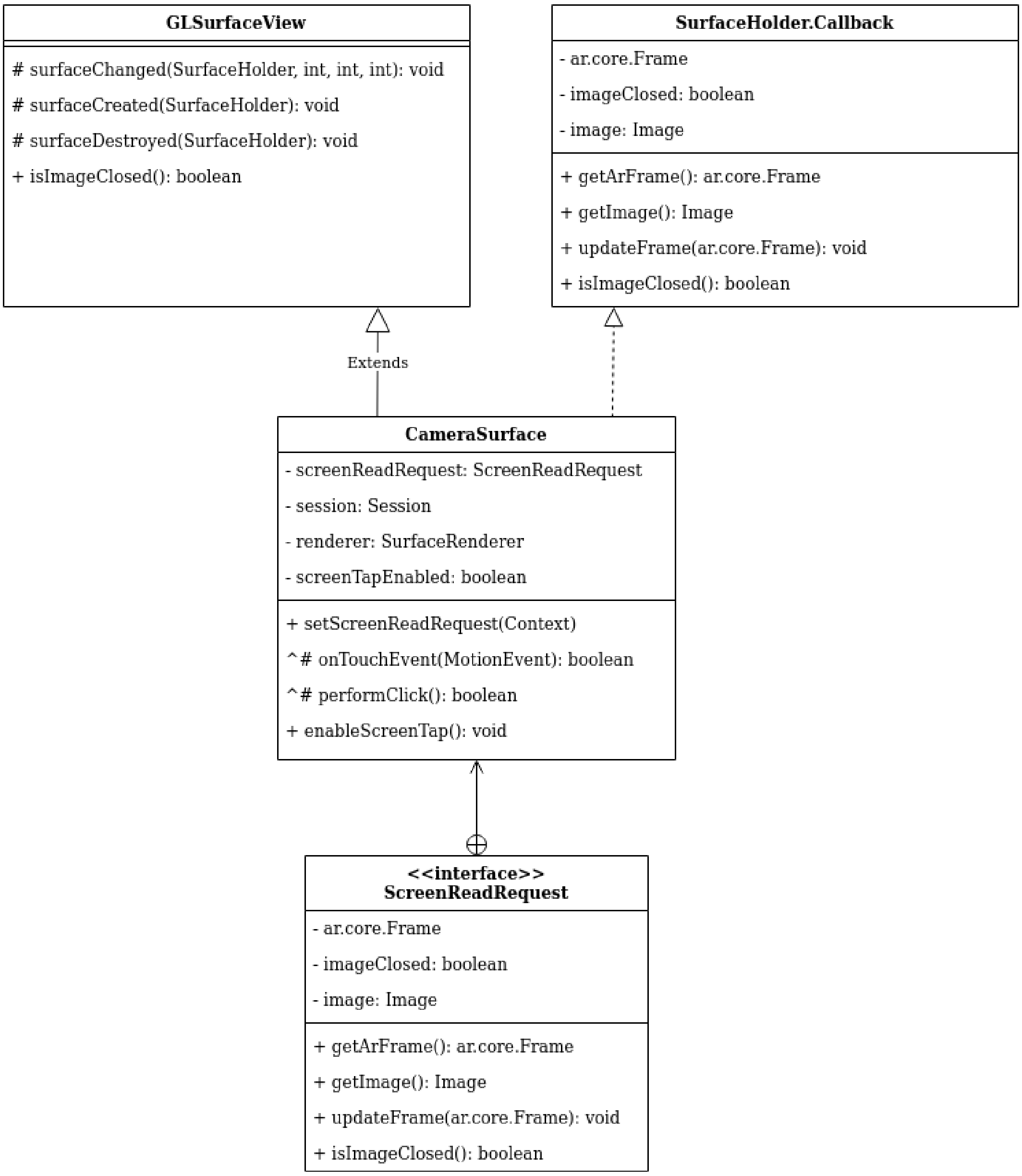

GLSurfaceView
# surfaceChanged(SurfaceHolder, int, int, int): void
# surfaceCreated(SurfaceHolder): void
# surfaceDestroyed(SurfaceHolder): void
+ isImageClosed(): boolean
SurfaceHolder.Callback
- ar.core.Frame
- imageClosed: boolean
- image: Image
+ getArFrame(): ar.core.Frame
+ getImage(): Image
+ updateFrame(ar.core.Frame): void
+ isImageClosed(): boolean
Extends
CameraSurface
- screenReadRequest: ScreenReadRequest
- session: Session
- renderer: SurfaceRenderer
- screenTapEnabled: boolean
+ setScreenReadRequest(Context)
^# onTouchEvent(MotionEvent): boolean
^# performClick(): boolean
+ enableScreenTap(): void
<<interface>>
ScreenReadRequest
- ar.core.Frame
- imageClosed: boolean
- image: Image
+ getArFrame(): ar.core.Frame
+ getImage(): Image
+ updateFrame(ar.core.Frame): void
+ isImageClosed(): boolean

Figure B.5: A diagram of the module that collects the visual information from the device's camera.

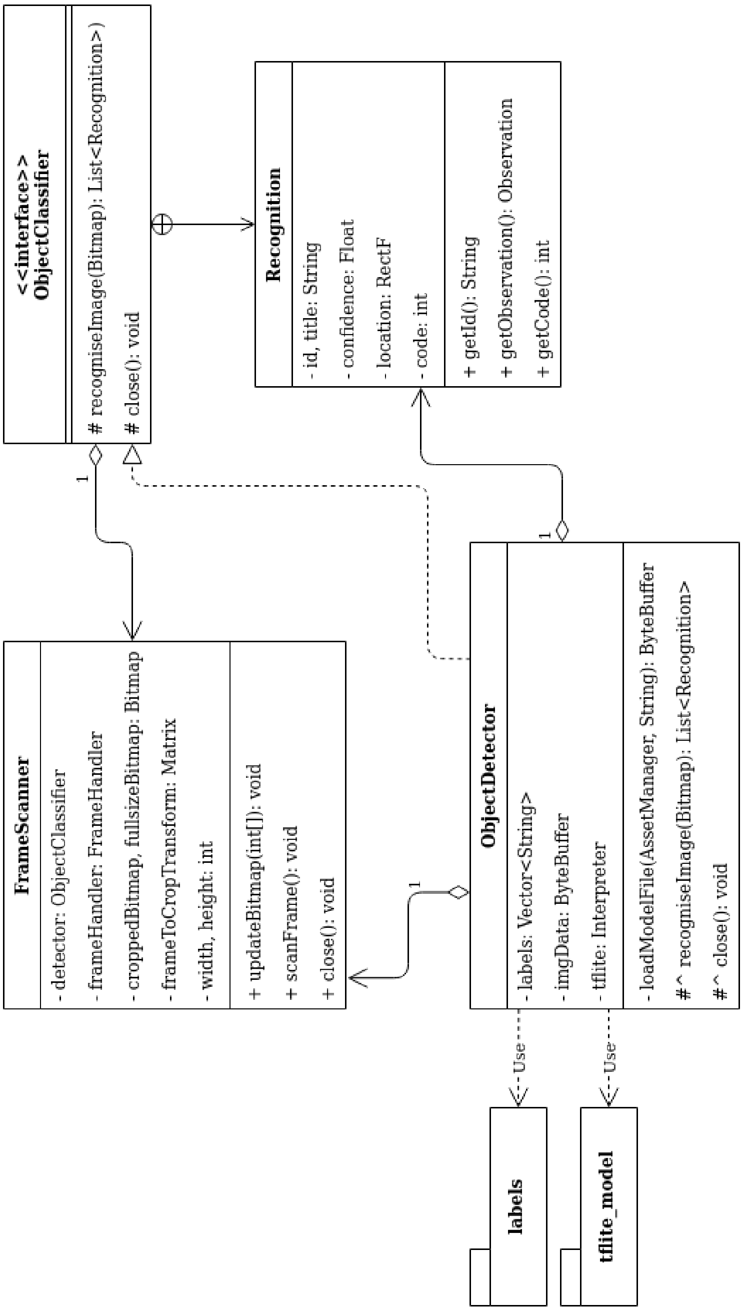


Figure B.6: A diagram of the object detection module that extracts object information from a still image collected from the camera.

List of References

Ahmaniemi, T. and Lantz, V. (2009). Augmented reality target finding based on tactile cues. In: *Proceedings of the 2009 International Conference on Multimodal Interfaces*, pp. 335–342. ACM.

Ahmetovic, D., Gleason, C., Ruan, C., Kitani, K., Takagi, H. and Asakawa, C. (2016). Navcog: a navigational cognitive assistant for the blind. In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 90–99. ACM.

Al-Khalifa, H.S. (2008). Utilizing QR Code and Mobile Phones for Blinds and Visually Impaired People. In: *Computers Helping People with Special Needs*, pp. 1065–1069. Springer.

Aloimonos, J., Weiss, I. and Bandyopadhyay, A. (1988). Active vision. *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356.

Apostolopoulos, I., Fallah, N., Folmer, E. and Bekris, K.E. (2012). Integrated online localization and navigation for people with visual impairments using smart phones. *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, no. 4, pp. 1322–1329.

Arditi, A. and Tian, Y. (2013). User interface preferences in the design if a camera-based navigation and wayfinding aid. *Journal of Visual Impairment & Blindness*, vol. 107, no. 2, pp. 118–129.

Aydemir, A., Pronobis, A., Göbelbecker, M. and Jensfelt, P. (2013). Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 986–1002.

Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005.

Bajcsy, R., Aloimonos, Y. and Tsotsos, J.K. (2018). Revisiting active perception. *Autonomous Robots*, vol. 42, no. 2, pp. 177–196.

Barfield, W., Cohen, M. and Rosenberg, C. (1997). Visual and auditory localization as a function of azimuth and elevation. *International Journal of Aviation Psychology*, vol. 7, no. 2, pp. 123–138.

Bellman, R. (1957). A Markovian decision process. *Journal of Mathematics and Mechanics*, pp. 679–684.

Bellotto, N. (2013). A multimodal smartphone interface for active perception by visually impaired. In: *IEEE SMC International Workshop on Human Machine Systems, Cyborgs and Enhancing Devices.* IEEE.

Bigham, J.P., Jayant, C., Miller, A., White, B. and Yeh, T. (2010). Vizwiz:: LocateIt-enabling blind people to locate objects in their environment. In: *Computer Vision and Pattern Recognition Workshops*, pp. 65–72. IEEE.

Blauert, J. (1969). Sound localization in the median plane. *Acta Acustica United with Acustica*, vol. 22, no. 4, pp. 205–213.

Blauert, J. (1997). *Spatial hearing: the psychophysics of human sound localization.* MIT press.

Blum, J., Bouchard, M. and Cooperstock, J.R. (2013). Spatialized audio environmental awareness for blind users with a smartphone. *Mobile Networks and Applications*, vol. 18, no. 3, pp. 295–309.

Boger, J., Poupart, P., Hoey, J., Boutilier, C., Fernie, G. and Mihailidis, A. (2005). A decision-theoretic approach to task assistance for persons with dementia. In: *International Joint Conference on Artificial Intelligence*, pp. 1293–1299. IEEE.

Borenstein, J. and Ulrich, I. (1997). The GuideCane - a computerized travel aid for the active guidance of blind pedestrians. In: *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1283–1288. IEEE.

Borji, A., Cheng, M.-M., Hou, Q., Jiang, H. and Li, J. (2014). Salient object detection: A survey. *Computational Visual Media*, pp. 1–34.

Bourne, R.R.A., Flaxman, S.R., Braithwaite, T., Cicinelli, M.V., Das, A., Jonas, J.B., Keeffe, J., Kempen, J.H., Leasher, J., Limburg, H., Naidoo, K., Pesudovs, K., Resnikoff, S., Silvester, A., Stevens, G.A., Tahhan, N., Wong, T.Y., Taylor, H.R. and Vision Loss Expert Group (2017). Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: A systematic review and meta-analysis. *The Lancet Global Health*, vol. 5, no. 9, pp. 888–897.

Caicedo, J.C. and Lazebnik, S. (2015 December). Active object localization with deep reinforcement learning. In: *The IEEE International Conference on Computer Vision.* IEEE.

Chen, S., Li, Y. and Kwok, N.M. (2011). Active vision in robotic systems: A survey of recent developments. *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377.

Chessa, M., Noceti, N., Odone, F., Solari, F. and Sosa-García, J. (2016). An integrated artificial vision framework for assisting visually impaired users. *Computer Vision and Image Understanding*, pp. 209 – 228.

Cohen, J. (2013). *Statistical power analysis for the behavioral sciences.* Routledge.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, vol. 20, no. 3, pp. 273–297.

Coughlan, J., Manduchi, R. and Shen, H. (2006). Cell phone-based wayfinding for the visually impaired. *International Workshop on Mobile Vision*, pp. 1–15.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893. IEEE.

Damen, D., Leelasawassuk, T. and Mayol-Cuevas, W. (2016). You-Do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance. *Computer Vision and Image Understanding*, vol. 149, pp. 98–112.

David, S., Schmitt, S., Utz, J., Hub, A. and Schlicht, W. (2014). Navigation within buildings: Novel movement detection algorithms supporting people with visual impairments. *Research in Developmental Disabilities*, vol. 35, no. 9, pp. 2026–2034.

Faria, J., Lopes, S., Fernandes, H., Martins, P. and Barroso, J. (2010). Electronic white cane for blind people navigation assistance. In: *World Automation Congress*, pp. 1–7. IEEE.

Fiannaca, A., Apostolopoulous, I. and Folmer, E. (2014). Headlock: a wearable navigation aid that helps blind cane users traverse large open spaces. In: *Proceedings of the International Conference on Computers & Accessibility*, pp. 19–26. ACM.

Findlay, J.M., Findlay, J.M., Gilchrist, I.D. *et al.* (2003). *Active vision: The psychology of looking and seeing.* 37. Oxford University Press.

Fitts, P. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, vol. 47, no. 6, p. 381.

Flores, G., Kurniawan, S., Manduchi, R., Martinson, E., Morales, L. and Sisbot, E. (2015). Vibrotactile guidance for wayfinding of blind walkers. *IEEE Transactions on Haptics.*

Frauenberger, C. and Noisterig, M. (2003). 3D Audio Interface for the Blind. In: *Proceedings of the International Conference on Auditory Displays*, pp. 280 – 283. Department of Computer and Information Sciences, University of Newcastle-upon-Tyne.

Gallina, P., Bellotto, N. and Di Luca, M. (2015). Progressive co-adaptation in human-machine interaction. In: *International Conference on Informatics in Control Proceedings*, pp. 2362 – 368. Springer.

Gardner, W.G. and Martin, K.D. (1995). HRTF measurements of a KEMAR. *The Journal of the Acoustical Society of America*, vol. 97, no. 6, pp. 3907–3908.

Geronazzo, M., Bedin, A., Brayda, L., Campus, C. and Avanzini, F. (2016). Interactive spatial sonification for non-visual exploration of virtual maps. *International Journal of Human Computer Studies*, vol. 85, pp. 4–15.

Giefing, G.-J., Janssen, H. and Mallot, H. (1992). Saccadic object recognition with an active vision system. In: *Proceedings of the International Conference on Pattern Recognition*, pp. 664–667. IEEE.

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587. IEEE.

Golledge, R.G., Marston, J.R., Loomis, J.M. and Klatzky, R.L. (2004). Stated preferences for components of a personal guidance system for nonvisual navigation. *Journal of Visual Impairment & Blindness,*, vol. 98, no. 3, pp. 135–147.

Gonzalez-Garcia, A., Vezhnevets, A. and Ferrari, V. (2015). An active search strategy for efficient object class detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3022–3031. IEEE.

Hersh, M. and Johnson, M. (2008). *Assistive Technology for Visually Impaired and Blind People.* Springer-Verlag London.

Hesch, J.A. and Roumeliotis, S.I. (2010). Design and analysis of a portable indoor localization aid for the visually impaired. *The International Journal of Robotics Research*, vol. 29, no. 11, pp. 1400–1415.

Hiebert, G. (2005). *OpenAL 1.1 specification and reference.*

Hoey, J., Poupart, P., von Bertoldi, A., Craig, T., Boutilier, C. and Mihailidis, A. (2010). Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 503–519.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861.*

Howard, R.A. (1960). *Dynamic programming and Markov processes.* John Wiley.

Iannizzotto, G., Costanzo, C., Lanzafame, P. and La Rosa, F. (2005). Badge3D for visually impaired. In: *IEEE Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 29–29. IEEE.

Kanwal, N., Bostanci, E., Currie, K. and Clark, A.F. (2015). A navigation system for the visually impaired: a fusion of vision and depth sensor. *Applied Bionics and Biomechanics*, vol. 2015.

Katz, B. and Picinali, L. (2011). Spatial audio applied to research with the blind. *Advances in Sound Localization.*

Katz, B.F., Kammoun, S., Parseihian, G., Gutierrez, O., Brilhault, A., Auvray, M., Truillet, P., Denis, M., Thorpe, S. and Jouffrais, C. (2012). NAVIG: Augmented reality guidance system for the visually impaired. *Virtual Reality*, vol. 16, no. 4, pp. 253–269.

Katz, B.F.G., Truillet, P., Thorpe, S.J., Jouffrais, C. and Jouffrais (2010). NAVIG: Navigation assisted by artificial vision and GNSS. In: *Workshop on Multimodal Location Based Techniques for Extreme Navigation.*

Keppel, G. (1991). *Design and analysis: A researcher's handbook.* Prentice-Hall, Inc.

Kim, J.-E., Bessho, M., Kobayashi, S., Koshizuka, N. and Sakamura, K. (2016). Navigating visually impaired travelers in a large train station using smartphone and bluetooth low energy. In: *Proceedings of the Symposium on Applied Computing*, pp. 604–611. ACM.

Klatzky, R.L., Marston, J.R., Giudice, N.A., Golledge, R.G. and Loomis, J.M. (2006). Cognitive load of navigating without vision when guided by virtual sound versus spatial language. *Journal of Experimental Psychology: Applied*, vol. 12, no. 4, pp. 223–232.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105. MIT Press.

Kunhoth, J., Karkar, A., Al-Maadeed, S. and Al-Attiyah, A. (2019). Comparative analysis of computer-vision and ble technology based indoor navigation systems for people with visual impairments. *International Journal of Health Geographics*, vol. 18, no. 1, p. 29.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J.R.R., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T. and Ferrari, V. (2018). The OpenImages Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982.*

Lee, Y. and Medioni, G. (2015). RGB-D camera-based wearable navigation system for the visually impaired. *Computer Vision and Image Understanding*, pp. 3 – 20.

Levitt, H. (1971). Transformed up-down methods in psychoacoustics. *The Journal of the Acoustical Society of America*, vol. 49, no. 2B, pp. 467–477.

Lewis, L., Sharples, S., Chandler, E. and Worsfold, J. (2015). Hearing the way: Requirements and preferences for technology-supported navigation aids. *Applied Ergonomics*, vol. 48, pp. 56–69.

Lichenstein, R., Smith, D.C., Ambrose, J.L. and Moody, L.A. (2012). Headphone use and pedestrian injury and death in the United States: 2004–2011. *Injury Prevention*, vol. 18, no. 5, pp. 287 – 290.

Lienhart, R. and Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In: *Proceedings of the International Conference on Image Processing*, vol. 1. IEEE.

Liu, L., Ouyang, W., Wang, X., Fieguth, P.W., Chen, J., Liu, X. and Pietikäinen, M. (2018). Deep learning for generic object detection: A survey. *International Journal of Computer Vision*.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S.E., Fu, C.-Y. and Berg, A.C. (2016). SSD: Single shot multibox detector. In: *Proceedings of the European Conference of Computer Vision*. Springer.

Lock, J., Cielniak, G. and Bellotto, N. (2017). Portable navigations system with adaptive multimodal interface for the blind. In: *AAAI Spring Symposium*. AAAI.

Lock, J., Cielniak, G. and Bellotto, N. (2019*a*). Active object search with a mobile device for people with visual impairments. In: *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 476–485.

Lock, J., Gilchrist, I., Cielniak, G. and Bellotto, N. (2019*b*). Bone-conduction audio interface to guide people with visual impairments. In: *International Conference on Smart City and Informatization*. Springer.

Lock, J., Gilchrist, I., Cielniak, G. and Bellotto, N. (2019*c*). Experimental analysis of a spatialised audio interface for people with visual impairments. Submitted January 2020.

Lock, J., Tramontano, A., Ghidoni, S. and Bellotto, N. (2019*d*). ActiVis: Mobile object detection and active guidance for people with visual impairments. In: *International Conference on Image Analysis and Processing*, pp. 649–660. Springer.

Lowe, D.G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110.

MacDonald, J.A., Henry, P.P. and Letowski, T.R. (2006). Spatial audio through a bone conduction interface. *International Journal of Audiology*, vol. 45, no. 10, pp. 595–599.

MacKenzie, I. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, vol. 7, no. 1, pp. 91–139.

Manduchi, R. and Coughlan, J. (2014). The last meter: blind visual guidance to a target. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3113–3122. ACM.

Marentakis, G. and Brewster, S. (2006). Effects of feedback, mobility and index of difficulty on deictic spatial audio target acquisition in the horizontal plane. In: *Proceedings of the Conference on Human Factors in Computing Systems*, pp. 359–368. ACM.

Marston, J.R., Loomis, J.M., Klatzky, R.L. and Golledge, R.G. (2019). Nonvisual route following with guidance from a simple haptic or auditory display. *Journal of Visual Impairment & Blindness*, vol. 101, no. 4, pp. 203–211.

Mocanu, B., Tapu, R. and Zaharia, T. (2016). When ultrasonic sensors and computer vision join forces for efficient obstacle detection and recognition. *Sensors*, vol. 16, no. 11.

Nicholson, J., Kulyukin, V. and Coster, D. (2009). Shoptalk: independent blind shopping through verbal route directions and barcode scans. *The Open Rehabilitation Journal*, vol. 2, no. 1.

Ou, S., Park, H. and Lee, J. (2020). Implementation of an obstacle recognition system for the blind. *Applied Sciences*, vol. 10, no. 1, p. 282.

Passini, R. and Proulx, G. (1988). Wayfinding without vision: An experiment with congenitally totally blind people. *Environment and behavior*, vol. 20, no. 2, pp. 227–252.

Petrie, H., Johnson, V., Strothotte, T., Raab, A., Michel, R., Reichert, L. and Schalt, A. (1997). Mobic: An aid to increase the independent mobility of blind travellers. *British Journal of Visual Impairment*, vol. 15, no. 2, pp. 63–66.

Pineau, J., Gordon, G., Thrun, S. *et al.* (2003). Point-based value iteration: An anytime algorithm for POMDPs. In: *International Joint Conference on Artificial Intelligence Organization*, pp. 1025–1032. Elsevier.

Pradeep, V., Medioni, G. and Weiland, J. (2010). Robot vision for the visually impaired. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pp. 15–22. IEEE.

Pratt, C. (1930). The spatial character of high and low tones. *Journal of Experimental Psychology*, vol. 13, no. 3, p. 278.

Puterman, M.L. (2014). *Markov decision processes: discrete stochastic dynamic programming.* John Wiley & Sons.

Quinones, P.-A., Greene, T., Yang, R. and Newman, M. (2011). Supporting visually impaired navigation: a needs-finding study. In: *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, pp. 1645–1650.

Radmard, S. and Croft, E.A. (2017). Active target search for high dimensional robotic systems. *Autonomous Robots*, vol. 41, no. 1, pp. 163–180.

Ramón, A., Ruiz, J., Granja, F.S., Carlos, J., Honorato, P., Rosas, J.I.G., Jimenez Ruiz, A.R., Seco Granja, F., Prieto Honorato, J.C. and Guevara Rosas, J.I. (2012). Accurate pedestrian indoor navigation by tightly coupling foot-mounted IMU and RFID measurements. *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 178–189.

Ran, L., Helal, S. and Moore, S. (2004). Drishti: an integrated indoor/outdoor blind navigation system and service. In: *Proceedings of the IEEE Conference on Pervasive Computing and Communications*, pp. 23–30. IEEE.

Rasolzadeh, B., Björkman, M., Huebner, K. and Kragic, D. (2010). An active vision system for detecting, fixating and manipulating objects in the real world. *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 133–154.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788. IEEE.

Ren, S., He, K., Girshick, R.B. and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Transactions on Pattern Analysis and Machine Intelligence*, pp. 1137–1149.

Rivera-Rubio, J., Arulkumaran, K., Rishi, H., Alexiou, I. and Bharath, A.A. (2015). An assistive haptic interface for appearance-based indoor navigation. *Computer Vision and Image Understanding*, vol. 149, no. Assistive Computer Vision and Robotics, pp. 126–145.

RNIB (2016). UK vision strategy. Tech. Rep., RNIB. Accessed: 19-07-2016.

Rodríguez, A., Bergasa, L., Alcantarilla, P., Yebes, J. and Cela, A. (2012). Obstacle avoidance system for assisting visually impaired people. In: *Proceedings of the IEEE Intelligent Vehicles Symposium Workshops*, vol. 35, p. 16. IEEE.

Rummery, G.A. and Niranjan, M. (1994). *On-line Q-learning using connectionist systems*, vol. 37. University of Cambridge, Department of Engineering Cambridge, England.

Sáez, J.M. and Escolano, F. (2008). Stereo-based Aerial Obstacle Detection for the Visually Impaired. In: *Workshop on Computer Vision Applications for the Visually Impaired*. IEEE.

Sandler, M.B., Howard, A.G., Zhu, M., Zhmoginov, A. and Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the of Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520.

Schauerte, B., Martinez, M., Constantinescu, A. and Stiefelhagen, R. (2012). An assistive vision system for the blind that helps find lost things. In: *International Conference on Computers for Handicapped Persons*, pp. 566–572. Springer.

Schonstein, D., Ferré, L. and Katz, B.F. (2008). Comparison of headphones and equalization for virtual auditory source localization. *The Journal of the Acoustical Society of America*, vol. 123, no. 5, pp. 3724–3724.

Schwarze, T., Lauer, M., Schwaab, M., Romanovas, M., Bohm, S. and Jurgensohn, T. (2015). An intuitive mobility aid for visually impaired people based on stereo vision. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 17–25. IEEE.

Shepard, R. (1964). Circularity in judgments of relative pitch. *The Journal of the Acoustical Society of America*, vol. 36, no. 12, pp. 2346–2353.

Shubina, K. and Tsotsos, J.K. (2010). Visual search for an object in a 3D environment using a mobile robot. *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 535–547.

Stanley, R.M. and Walker, B.N. (2006). Lateralization of sounds using bone-conduction headsets. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, pp. 1571–1575. Sage.

Sutton, R.S. and Barto, A.G. (1998). *Introduction to reinforcement learning*, vol. 135. MIT Press.

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In: *AAAI Conference on Artificial Intelligence*. AAAI.

Terreran, M., Tramontano, A., Lock, J., Ghidoni, S. and Bellotto, N. (2020). Real-time object detection using deep learning for helping people with visual impairments. In: *Proceedings of the International Conference on Pattern Recognition.*

Tian, Y., Yang, X., Yi, C. and Arditi, A. (2013). Toward a Computer Vision-Based Wayfinding Aid for Blind Persons to Access Unfamiliar Indoor Environments. *Machine Vision and Applications*, pp. 521 – 535.

Tramontano, A.G. (2019). *Deep Learning Networks for Real-time Object Detection on Mobile Devices.* Master's thesis, University of Padova.

Vázquez, M. and Steinfeld, A. (2012). Helping visually impaired users properly aim a camera. In: *Proceedings of the International Conference on Computers and Accessibility*, pp. 95–102. ACM.

Vera, P., Zenteno, D. and Salas, J. (2014). A smartphone-based virtual white cane. *Pattern Analysis and Applications*, vol. 17, no. 3, pp. 623–632.

Vorapatratorn, S. and Nambunmee, K. (2014). isonar: an obstacle warning device for the totally blind. *Journal of Assistive, Rehabilitative & Therapeutic Technologies*, vol. 2, no. 1.

Ward, J. and Meijer, P. (2010). Visual experiences in the blind induced by an auditory sensory substitution device. *Consciousness and Cognition*, pp. 425 – 500.

Watkins, C.J.C.H. (1989). *Learning from delayed rewards.* Ph.D. thesis, King's College, Cambridge.

Watkins, C.J.C.H. and Dayan, P. (1992 May). Q-learning. *Machine Learning*, vol. 8, no. 3, pp. 279–292.

Welford, A. (1968). *Fundamentals of skill.* Methuen.

Wetherill, G. and Levitt, H. (1965). Sequential estimation of points on a psychometric function. *British Journal of Mathematical and Statistical Psychology*, vol. 18, no. 1, pp. 1–10.

Willis, S. and Helal, S. (2005). RFID information grid for blind navigation and wayfinding. In: *Proceedings of the IEEE International Symposium on Wearable Computers*, pp. 34–37. IEEE.

Wilson, J., Walker, B., Lindsay, J., Cambias, C. and Dellaert, F. (2007). Swan: System for wearable audio navigation. In: *IEEE International Symposium on Wearable Computers*, pp. 91–98. IEEE.

Wu, J., Yang, J. and Honda, T. (2010). Fitts' law holds for pointing movements under conditions of restricted visual feedback. *Human Movement Science*, vol. 29, no. 6, pp. 882–892.

Xiao, J., Joseph, S.L., Zhang, X., Li, B., Li, X. and Zhang, J. (2015). An assistive navigation framework for the visually impaired. *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 5, pp. 635–640.

Xie, B. (2013). *Head-related transfer function and virtual auditory display.* J. Ross Publishing.

Ye, X., Lin, Z., Lee, J.-Y., Zhang, J., Zheng, S. and Yang, Y. (2018*a*). GAPLE: Generalizable approaching policy learning for robotic object searching in indoor environment. *IEEE Robotics and Automation Letters.*

Ye, X., Lin, Z., Li, H., Zheng, S. and Yang, Y. (2018*b*). Active object perceiver: Recognition-guided policy learning for object searching on mobile robots. In: *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 6857–6863. IEEE.

Yusif, S., Soar, J. and Hafeez-Baig, A. (2016). Older people, assistive technologies, and the barriers to adoption: A systematic review. *International Journal of Medical Informatics*, vol. 94, pp. 112–116.

Zhang, C. and Zhang, Z. (2010 June). A survey of recent advances in face detection. Tech. Rep., Microsoft.

Zwiers, M., Van Opstal, A. and Cruysberg, J. (2001). A spatial hearing deficit in early-blind humans. *Journal of Neuroscience*, vol. 21, no. 9.