



**Roberto Gaspar Vieira**

**AudioNavigator: áudio como suporte a sistemas de orientação com GPS**

**AudioNavigator: using audio cues to support GPS-enabled guidance systems**





**Roberto Gaspar Vieira**

**AudioNavigator: áudio como suporte a sistemas de orientação com GPS**

**AudioNavigator: using audio cues to support GPS-enabled guidance systems**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática (M.I.E.C.T.), realizada sob a orientação científica do Professor Doutor José Maria Amaral Fernandes, Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e do Professor Doutor João Paulo Trigueiros da Silva Cunha, Professor Associado com Agregação do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto



**o júri / the jury**

presidente / president

**Prof. Doutor Joaquim João Estrela Ribeiro Silvestre Madeira**

Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

**Prof. Doutor António Miguel Pontes Pimenta Monteiro**

Professor Auxiliar do Departamento de Engenharia Informática da Faculdade de Engenharia da Universidade do Porto

**Prof. Doutor José Maria Amaral Fernandes**

Professor Auxiliar do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)

**Prof. Doutor João Paulo Trigueiros da Silva Cunha**

Professor Associado com Agregação do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto (co-orientador)



**agradecimentos /  
acknowledgements**

Agradeço à minha família pelo apoio incondicional e incansável dedicação ao longo destes anos, em especial à minha mãe, que sempre fez mais do que lhe cabia.

Aos meus amigos de sempre e para sempre.

Aos meus amigos e colegas que me acompanharam nesta memorável aventura académica, em especial aos Lendários, que tantas vivências e momentos únicos me proporcionaram.

Ao Prof. José Fernandes e ao Prof. José Vieira pelo apoio e disponibilidade fulcrais na elaboração deste trabalho.





## Palavras-chave

Orientação, Navegação, Áudio, Dificuldade Visual, GPS, Computação Móvel

## Resumo

A oferta de dispositivos móveis para navegação e orientação tem aumentado ao longo dos últimos anos devido à "revolução móvel", impulsionada pelo acesso generalizado a smartphones com GPS integrado e outras tecnologias semelhantes que fornecem informação geográfica, visando diferentes cenários de utilização tais como condução de veículos ou navegação pedestre. Porém, a forma como estes dispositivos interagem com o utilizador não é apropriada para indivíduos com dificuldades visuais nem para situações onde o utilizador não possa utilizar a visão para se orientar (e.g. equipas de primeiros socorros em missão de salvamento num local onde há fumo, nevoeiro ou noutras condições de visibilidade reduzida). Nos últimos anos, os sistemas de navegação áudio têm sido apontados como uma possível solução para este problema.

Nesta dissertação abordamos esta temática desenvolvendo um sistema - AudioNavigator - que fornece instruções via áudio. AudioNavigator é um sistema (aplicação) para dispositivos móveis Android capaz de converter sinais de áudio mono em sinais de áudio espacial que são percebidos pelo utilizador como provenientes de uma determinada localização, orientando-o até à localização alvo. O objectivo principal do AudioNavigator é guiar um indivíduo equipado com um dispositivo móvel com GPS até um determinado local recorrendo apenas a instruções áudio.

O sistema foi desenvolvido com base num modelo "head-related transfer function" (HRTF) que modula o tempo e intensidade das instruções áudio de forma a gerar uma percepção 3D do som para os utilizadores que envergam um par de auriculares.

O AudioNavigator foi testado quanto à percepção espacial de 10 participantes que conseguiram identificar o ângulo correcto de determinado som em 63% das vezes, com uma margem de 30 graus. Além disso, o sistema foi testado em duas séries em campo aberto onde dois percursos, um composto por 12 e outro composto por 25 pontos geográficos, foram percorridos por 5 participantes, tendo estes apenas as orientações áudio do sistema como suporte. Os 5 participantes completaram os 10 caminhos com sucesso.

Estes resultados demonstram o actual potencial dos dispositivos móveis para serem utilizados em cenários de navegação sem recurso a visão. O AudioNavigator consiste numa prova de conceito para a utilização de sistemas de navegação áudio neste tipo de cenários.



**Keywords**

Guidance, Navigation, Audio, Visual Impairment, GPS, Mobile Computing

**Abstract**

Navigation and orientation assistive devices availability has been increasing over the years due to the "mobile revolution", propelled by the generalized accessibility to smartphone integrated GPS systems and other similar technologies that provide location information, targeting different usage scenarios such as car driving or pedestrian navigation. Nevertheless, the way these consumer devices interface with the users are not appropriate for visually impaired individuals nor to critical situations where users cannot rely on vision for navigation (e.g. first responders involved in rescue procedures in dense smoke presence or other reduced visibility conditions). Over the past years, audio guidance systems have been pointed in the literature as a possible solution to this problem.

In this thesis we address this topic by developing a system - AudioNavigator - that provides navigation via audio signatures. AudioNavigator is an Android mobile application that is able to convert monaural audio signals into spatial audio signals that are perceived by the user as coming from a certain direction, thus guiding him towards the target destination. The main objective of AudioNavigator is to guide an individual equipped with a GPS-enabled mobile device to a given destination through solely audio cues.

The system was developed based on a "head-related transfer function" (HRTF) model that modulates time and intensity of audio pitch cues to generate 3D direction perceptions to users wearing a set of headphones.

AudioNavigator was tested for angle perception with 10 volunteers that were able to select the correct angle based on the audio cues in 63% of the times within a 30 degrees margin. Furthermore, the system was tested in two open field trials where two paths, composed of 12 and 25 waypoints, were tested by 5 users, solely relying on the audio guidance provided by the system. The 5 users completed all 10 paths with success.

These results demonstrate the current potential of mobile devices to be used in non-visual navigation scenarios. AudioNavigator provides a proof of concept for the usage of audio navigation systems in such scenarios.



# Contents

<b>Contents</b>	<b>i</b>
<b>List of Figures</b>	<b>iii</b>
<b>List of Tables</b>	<b>v</b>
<b>List of Acronyms</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and context . . . . .	1
1.2 Objectives . . . . .	1
1.3 Dissertation structure . . . . .	2
<b>2 State of the Art</b>	<b>3</b>
2.1 Guidance systems . . . . .	3
2.2 Audio guidance . . . . .	3
2.3 Spatial audio . . . . .	5
2.4 Binaural model . . . . .	5
2.4.1 Panning model . . . . .	6
2.4.2 HRTF model . . . . .	6
2.4.3 Signal filtering using HRTF . . . . .	7
2.4.4 FIR filter . . . . .	7
<b>3 AudioNavigator</b>	<b>9</b>
3.1 AudioNavigator overall architecture . . . . .	9
3.1.1 Routing module . . . . .	10
3.1.2 Visual module . . . . .	10
3.1.3 Audio module . . . . .	10
3.2 Overall implementation of the system . . . . .	10
3.2.1 Routing module . . . . .	10
3.2.2 Visual module . . . . .	11
3.2.3 Audio module . . . . .	11
Native code . . . . .	12
NativeBridge . . . . .	12
NativeAudio . . . . .	12
MIT HRTF lib . . . . .	12
FIR filter . . . . .	13

Audio . . . . .	13
AudioTrack . . . . .	14
Player . . . . .	14
<b>4 AudioNavigator evaluation</b>	<b>15</b>
4.1 AudioNavigator spatial cueing accuracy . . . . .	15
4.1.1 Spatial audio evaluation . . . . .	15
4.1.2 Results . . . . .	17
4.2 Overall system performance in guidance accuracy . . . . .	21
4.2.1 Overall system evaluation . . . . .	21
4.3 Discussion on the results . . . . .	23
4.3.1 Spatial audio evaluation . . . . .	23
4.3.2 Overall system evaluation . . . . .	24
<b>5 Conclusions</b>	<b>25</b>
5.1 Accomplishments . . . . .	25
5.2 Issues . . . . .	26
5.3 Future Work . . . . .	26
5.3.1 Sound perception - further research . . . . .	26
5.3.2 Routing module . . . . .	26
5.3.3 Audio playback . . . . .	26
5.3.4 Stereophones vs. scenario . . . . .	26
<b>References</b>	<b>29</b>

# List of Figures

2.1	Features and implementation details of the displays . . . . .	4
2.2	Block-diagram structure for a third-order FIR filter . . . . .	8
3.1	AudioNavigator architecture. . . . .	9
3.2	AudioNavigator’s GUI. . . . .	11
4.1	Spatial audio test - GUI for the participants (on the left), GUI for the tester (on the right) and test flow. . . . .	16
4.2	Test results compared to ideal results (the 45° line) - without “learning” part. . . . .	17
4.3	Histogram of deviations from ideal results (in degrees). . . . .	18
4.4	Deviations (in degrees) from the ideal test values for each subject (‘S’). . . . .	19
4.5	Comparison between response time and answers deviation. . . . .	20
4.6	User wearing stereophones and holding the mobile device. . . . .	21
4.7	Test results (colored lines) compared to ideal path (red dots) - path 1. . . . .	22
4.8	Test results (colored lines) compared to ideal path (red dots) - path 2. . . . .	23





# List of Tables

2.1	Guidance system categories and features. . . . .	4
4.1	Set of 35 sound directions (in degrees) used in spatial audio test. . . . .	16
4.2	Answer deviation <i>vs.</i> time to answer - for each subject. . . . .	20



# List of Acronyms

<b>Acronym</b>	<b>Description</b>
<b>CMU</b>	Carnegie Mellon University
<b>FIR</b>	Finite Impulse Response
<b>GLONASS</b>	Global Navigation Satellite System
<b>GPS</b>	Global Positioning System
<b>GUI</b>	Graphical User Interface
<b>HPI</b>	Haptic Pointer Interface
<b>HRIR</b>	Head-Related Impulse Response
<b>HRTF</b>	Head-Related Transfer Function
<b>IEETA</b>	Instituto de Engenharia Electrónica e Telemática de Aveiro
<b>IT</b>	Instituto de Telecomunicações
<b>JNI</b>	Java Native Interface
<b>KEMAR</b>	Knowles Electronics Mannequin for Acoustics Research
<b>MIT</b>	Massachusetts Institute of Technology
<b>NDK</b>	Native Development Kit
<b>PCM</b>	Pulse-Code Modulation
<b>POI</b>	Point Of Interest
<b>SDK</b>	Software Development Kit
<b>UI</b>	User Interface



# Chapter 1

## Introduction

### 1.1 Motivation and context

The availability of navigation and orientation assistive devices and systems has been increasing over the years, propelled by the free accessibility to GPS and other similar technologies (e.g. GLONASS) that provide location information, targeting different usage scenarios such as car driving or pedestrian navigation. In general [1], these technological assistive solutions provide navigation through visual representations of a map and other relevant information while playing audio guidance instructions in the form of speech. This approach might not be suitable when the user is visually impaired (i.e. individuals partly or totally deprived of their sight) or is unable to rely on sight to navigate (e.g. a firefighter in a rescue mission may not be able to check the guidance device in a smoke-filled area, given the reduced visibility conditions). Moreover, even if being able to use vision to navigate, there might be situations when a user cannot check the navigation device, such as while doing something that keeps both hands occupied. In this context, there is a need to implement a solution for these scenarios where individuals cannot rely on sight to get navigation instructions. This thesis addresses scenarios where visual guidance is not possible or suitable by developing a solution - the AudioNavigator - that provides guidance instructions through audio signature. Considering that audition plays an important role in human perception of location and direction, the use of audio cues seems to be an interesting solution in this scenario to substitute or complement the vision. Just by hearing, an individual can sense the proximity and direction of a certain sound source [2]. This capacity is particularly valued in the absence of sight, which is the primary human sense for sighted people in terms of surrounding environment awareness [3].

### 1.2 Objectives

The main goal of this thesis is to explore navigation support namely for visually impaired individuals using audio signature to support existing visual navigation that may not be the ideal or a viable option. We aim to explore the audio solutions already used in other scenarios concerning guidance based on audio instructions [4, 5, 1, 6, 7] to study the employed methods and best ways of representing the audio instructions in order to achieve high user perception of direction with minimal cognitive effort. Upon collecting and comparing relevant information from such solutions, we will implement a guidance tool to orient the individuals

to a given target location taking into account the existing routes and avoidance of obstacles (e.g. physical, dangerous locations). Note beforehand that the routing solution is not the focus of the present work but the implementation of a reliable audio cueing solution to direct individuals in a given route. This thesis will focus on the implementation of a reliable audio cueing system, considering several parameters (e.g. what sort of audio is ideal and how it should be generated) to provide audio support, besides visual navigation support.

### 1.3 Dissertation structure

This dissertation is divided into the following chapters, excluding this one:

- **Chapter 2 - State of the Art**, presents an overview of the existing literature, technologies and solutions within this thesis scope, in an attempt to gather relevant information to design and implement the AudioNavigator knowing the strategies, assumptions and capabilities of the existent approaches to similar scenarios.
- **Chapter 3 - AudioNavigator**, describes the AudioNavigator architecture, its main modules and their implementation and role in the system.
- **Chapter 4 - AudioNavigator evaluation**, presents the evaluation parameters, the usability tests developed to evaluate the system performance and the obtained results. An analysis of the results for each test is also presented.
- **Chapter 5 - Conclusions**, presents some issues found during the development of this work, the achieved accomplishments and proposes further work improvements.

# Chapter 2

## State of the Art

In this chapter we will address some areas that are relevant to this work, presenting concepts and previous works to understand what has been done and what results and conclusions that have been reached. We will focus on the following subjects:

- Guidance systems - an overview on existing guidance systems.
- Audio guidance systems - guidance systems that use audio as spatial cue.
- Spatial audio - the models used to produce the audio cues that support the guidance systems.

### 2.1 Guidance systems

Guidance systems are present in many contexts of people's lives, being it a simple system embedded in a mobile phone or a more complex system incorporated in an airplane. Since the days where a map and a simple compass were among the few available tools in which a person could rely, many new guidance mechanisms have shown up. From inertial, radio or radar to satellite navigation systems such as GPS and GLONASS [8], a diverse set of technologies are available nowadays. Moreover, not only the number of guidance methods has increased, but also the accuracy, reliability and availability, such as the GPS technology in mobile devices (e.g. smartphones) [9] that people can easily carry and use. Based on GPS technology, different types of guidance systems can be implemented, as it is discussed next. GPS-enabled devices support guidance systems providing visual and/or audible instructions. Table 2.1 presents some existing categories and features.

### 2.2 Audio guidance

Among the numerous areas where guidance mechanisms can be applied, the visual impairment issue has received much attention over the past years [7]. The visual impaired people have been directly affected by the evolution of audio guidance devices and techniques, providing the chance of a more independent life-style. The systems studied in this thesis address different ways of providing audio instructions, such as:

- Audio modulation, either spatial audio or simple stereo;

- Speech, including (or not) spoken cues with route information (direction, time, distance);
- Audio icons, allowing to match sounds with objects or guidance instructions.

	Visual			Audio			
	Graphical		Textual	Spatial		Non-Spatial	
	2D	3D		Speech	Non-speech	Speech	Non-speech
Experimental Procedure	Map; Next turn arrow; Compass-like arrow	Map with 3D imagery lanes and signposts; orthophotomap; full-textured 3D landmarks [10]	Distance; Speed; Street name; Next turn street name	Utterance giving distance to next waypoint such as "24 meters" [11]	Short tones like "beeps" [11] or continuous sounds [7]	"Turn a little bit to the left." describing a 20 degree turn [5]	Short sounds pre-associated to directions or objects [5]

Table 2.1: Guidance system categories and features.

Among several studies there is a particularly interesting one [12] that addresses several audio guidance displays, each one based on different types of audio cues. It describes and compares many of the existing techniques that can be used to provide audio guidance, considering this thesis' context, as presented next. Loomis [12] performed tests with 15 blind users who had to walk 50 meters in a path with turns, defined by several waypoints, using five different displays corresponding to different sets of guidance equipment, as shown in Figure 2.1.

Features	Type of Display				
	Virtual speech	Virtual tone	HPI tone	HPI speech	Body pointing
Auditory source	Headphones	Headphones	Shoulder-mounted speaker	Shoulder-mounted speaker	Shoulder-mounted speaker
Compass location	Head	Head	Handheld pointer	Handheld pointer	Torso
Nonspeech acoustic signal	None	Tone (5 per second); swept tone (2.3 per second)	Tone (5 per second)	None	Tone (5 per second)
Acoustic off-course indicator	None	Yes; type of tone	Yes; tone if pointer is accurate; bearing if off by more than 90 degrees	Speech; straight left/right bearing if off by more than 90 degrees	Yes; tone if pointer is accurate; bearing if off by more than 90 degrees
Acoustic directional spatialization	Yes	Yes	None	None	None
Acoustic distance spatialization	Intensity	Intensity	Intensity	None	Intensity
Haptic pointer	None	None	Handheld	Handheld	Torso

Figure 2.1: Features and implementation details of the displays [12].

Guidance was provided based on the user's distance and orientation to the next waypoint. The measured functional attributes were:

- Speech versus non-speech (tone) information;
- Spatialized (virtual) versus non-spatialized auditory output;
- Presence or absence of acoustic distance cues;
- User's hand orientation versus body orientation.



Distance cues (when present) were issued as speech indicating the distance to the next waypoint and the sound intensity of speech and tones increased as the user got closer to the waypoint. "Virtual speech" and "Virtual tone" displays played audio that was spatialized by virtual acoustic software to include direction cues, so that the sound would seem to come from the direction of the next waypoint. The other displays provided guidance instructions playing audio, through a shoulder-mounted speaker, based on the direction between a pointer held by (or attached to) the user and the next waypoint. That is, if the user pointed to the direction of the next waypoint (within a 10 degrees interval), the "HPI tone" and "Body pointing" displays would play a sequence of beeps to indicate alignment, whilst the "HPI speech" display would present the word "straight".

As concluded after a series of tests, the virtual displays provided the best results by leading to the fastest mean travel times, which favours spatial audio systems. Also, speech (spoken) instructions are considered a viable solution to guide a blind individual [3]. Yet, such instructions can be distracting or hard to hear when on-the-go [4], increasing the cognitive load necessary to handle simultaneous tasks that are characteristic of everyday navigation [13].

## 2.3 Spatial audio

Audition allows humans to percept the location and direction of a sound source, as referred in section 1.1. In the context of the present thesis we are interested in what ways audio can complement or replace the visual guidance in 3D navigation - in other words, in 3D or spatial audio solutions. This type of audio can be achieved using models that describe how it is produced. This section presents the following models:

- Binaural model.
- Panning model.
- Head-related transfer function (HRTF) model.

The panning and HRTF models are a type of binaural model. In the following section previous studies where such models have been used are referred and a comparison between models is performed.

## 2.4 Binaural model

Binaural, as the name indicates, refers to both ears and has to do with hearing perception of sound. A binaural model is a way to make the sound seem like coming from a certain direction in a three-dimensional (3D) space [14]. By using two channels, one for each ear, a sound can be played with different intensity and time delay between the ears, so that an individual can locate the direction of the sound - a sound coming from the left side of the head reaches the left ear first and with higher intensity than the right ear. Time and intensity differences are the most important clues to achieve the 3D sound effect. As binaural models, the panning model and the HRTF model have been widely used in this context.

### 2.4.1 Panning model

Panning is a technique to create a sound from a source signal that is spread (panned) to each of the channels in order to add spatial features like sound direction. It is a simple and computationally cheap method to achieve the illusion of 3D sound, using time and intensity differences together so as to do it. The time delay resultant from the difference between the distances from each ear to the sound source gives a clue about its direction; if the angle between the head and the sound source is different from  $0^\circ$  and  $180^\circ$  (clockwise), the torso will hide one of the ears, which will perceive the sound with lower intensity than the other ear - this is known as shading effect. These intensity differences are particularly difficult to define between back and front of the head [14]. The following projects use a panning model to render 3D sound.

Holland, Morse, and Gedenryd [4] addressed the widely used GUI-based interaction style present in mobile devices to propose an alternative user interaction based on audio, aimed at keeping the necessary user interaction with the device while reducing the amount of user attention needed when performing a task. The developed prototype indicates the bearing to the target location by means of panning a sound source, which can provide clear distinct sounds from straight ahead, extreme left, extreme right and some intermediate points. Though panning method is referred to be not as capable as a more precise HRTF-based method, the front-back confusion (difficulty in distinguishing if a sound is located in front or behind the listener) seems to affect both methods in many systems. To avoid this problem, two different sound sources were used to represent the destination bearing - one for the semicircle in front of the user and the other for the semicircle behind the user.

Jeffrey, Mathieu and Jeremy [1] used spatial audio to provide blind people a sense of environmental awareness by indicating points of interest (POI) information located nearby the user, resorting to speech or "audio icons" - short elucidative sounds. Though relying on an "iphone 4" smartphone with a built-in library ("OpenAL") to generate 3D audio, it is referred that "OpenAL" is limited, appearing to perform only simple stereo panning, lacking the robustness of a full HRTF implementation. They concluded that the currently available smartphones' capabilities make it possible to explore an environment, but cannot provide high-accuracy navigation.

Zwinderman, Zavialova, Tetteroo and Lehouck [7] implemented a 3D audio navigation system for bicyclists. The system is based on a pair of headphones and an Android smartphone ("HTC Hero"), attached to the headphones, to which they developed a software application that plays a continuous signal (music), balancing its intensity between left and right ear, according to the user's location and orientation. They evaluated the implemented system and considered that the concept of navigation based on spatial distribution of audio for bicyclists facilitates efficient and accurate navigation.

### 2.4.2 HRTF model

The HRTF model [15] is more accurate but computationally more expensive than the panning model. This model uses functions (HRTFs) that capture relevant spatial cues - sound intensity, time delays between the two ears and spectrum of the sound, which depend on the interaction between the sound wave and the listener's torso, head and outer ear shape. These cues are recorded as a head-related impulse response (HRIR - time domain) and represented by its Fourier transform (HRTF - frequency domain) [16]. This is possible by placing microphones

in the ears of a person or a dummy head, while playing one of the sound sources evenly distributed around the subject's head. The HRIRs can be used as filters in spatial audio systems, since a convolution of a HRIR with a sound converts the sound to what a listener would hear if the sound was played at the sound source location. Each recording is produced using two different filters - one for each ear [14]. The following projects' summaries regard 3D audio guidance using HRTF model.

In [17], tests comparing the use of a (physical) map with the use of a headset providing spatial audio guidance through a bell sound with direction cue and proximity cue (volume change), in a controlled area, show that the difference between these two approaches is minimal. It is referred that the use of 3D audio guidance is suitable in rescue work performed by firefighters as well as in other groups like persons with visual impairment.

Vazquez-Alvarez and Brewster [18] tested the built-in HRTF implementation for rendering spatial audio of a "Nokia N95" mobile phone and concluded that it can provide clear localization discrimination for 5 sources in the front 180° of the user, hence showing that mobile phones are capable of running 3D audio interfaces.

Jens [19] took on a project addressing the lack of commercial breakthroughs on binaural 3D audio technology, implementing an Android application to render this type of audio. A comparison between stereophones shows that stereophones normally distributed with Android devices can be used for binaural 3D audio, even though they are not as accurate as larger stereophones conventionally used for such audio. The resulting implemented system can filter mono signals in real time but does not support binaural 3D audio, leaving the project's conclusion open.

### 2.4.3 Signal filtering using HRTF

In order to spatialize an audio signal in a certain direction, it is necessary to apply the HRTF coefficients correspondent to that direction using a filter. A filter is a system that is designed to remove or modify some characteristic of a signal [20]. Considering the use of finite-length (discrete-time) signals, we will address a discrete-time system - a computational process for transforming one sequence (input signal) into another sequence (output signal) - known as FIR filter.

#### 2.4.4 FIR filter

A FIR filter is a system that transforms a discrete-time signal by computing a weighted running average of consecutive numbers of the sequence, forming a new sequence of the average values, where the weights are the coefficients. Such behaviour is defined by the general FIR filter equation

$$y[n] = \sum_{k=0}^M b_k x[n - k] \quad (2.1)$$

$b_k$  is the set of filter coefficients. The  $M$  parameter is the order of the FIR filter, which is equal to the number of filter coefficients (known as the filter length) minus one.  $x[n]$  is the finite-length input signal and  $y[n]$  is the finite-length output signal. Figure 2.2 shows a graphical representation of a block diagram for a third-order FIR digital filter, a particular case of the FIR filter general definition 2.1.

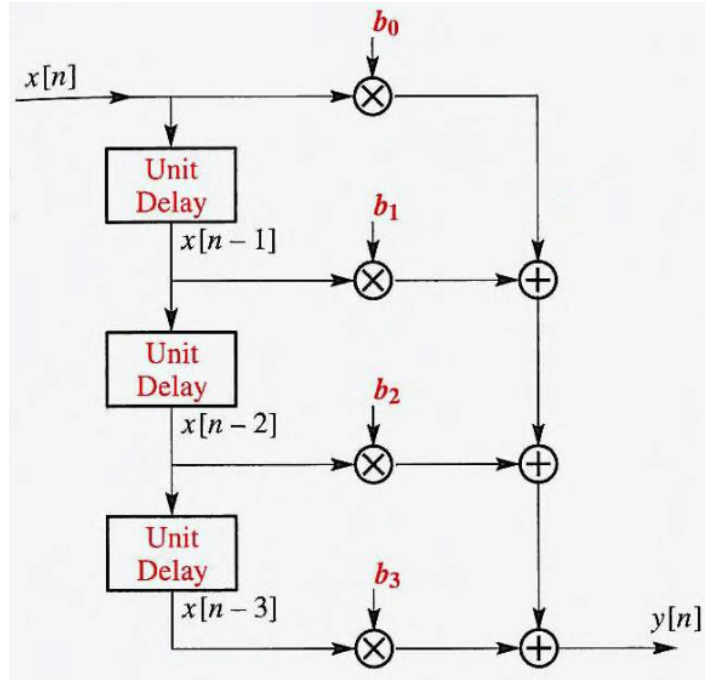


Figure 2.2: Block-diagram structure for a third-order FIR filter [20].

This general expression 2.1 for the FIR filter's output can be derived in terms of impulse response, replacing the filter coefficients  $b_k$  by the impulse response sequence  $h[k]$ , obtaining the following convolution sum

$$y[n] = \sum_{k=0}^M h[k] x[n - k] \quad (2.2)$$

Thus, the output is obtained by convolving the input ( $x[n]$ ) with the impulse response ( $h[k]$ ). An impulse response is a finite sequence, hence the name finite impulse response (FIR). This is where the HRIRs (represented by their Fourier transform - HRTFs) are used, since these are impulse responses that describe how a sound source is perceived at the left and right ears of a listener.

## Chapter 3

# AudioNavigator

In this chapter we will describe the basic architecture and technical options made in AudioNavigator, an audio based navigation system.

### 3.1 AudioNavigator overall architecture

AudioNavigator is a software application developed to face the need of a guidance solution, based on audio. It is meant to be deployed in mobile devices running Android operating system and equipped with GPS receiver, accelerometer and magnetometer. The application receives the location to where the individual must be guided and uses the device's GPS and sensors to navigate towards it. The visual interface is minimalist, with an arrow similar to a compass pointing to the intended direction. As for the audio interface, it produces sounds that are perceived by the user as if coming from a certain point of the three-dimensional space (surrounding the individual). An architecture overview of AudioNavigator is presented in Figure 3.1.

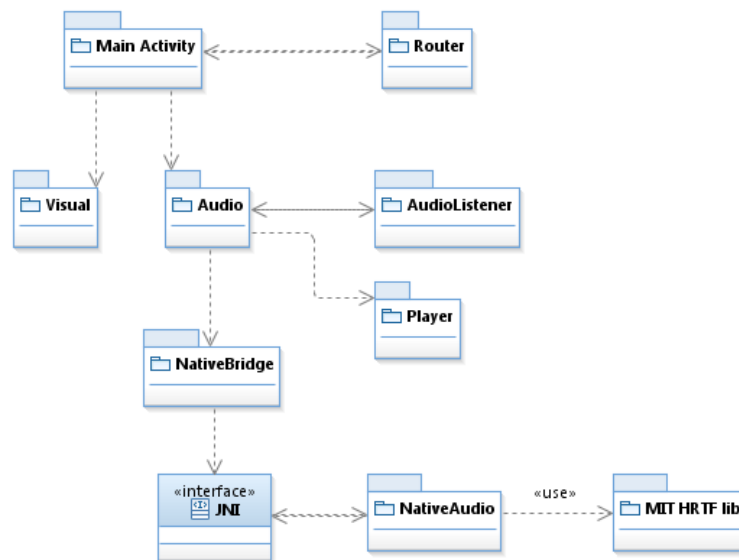


Figure 3.1: AudioNavigator architecture.

The core module of this system is the main activity module, which initiates the application and performs several tasks such as gathering and managing GPS and sensors data, calculating bearing angles, switching information with and between other modules. Besides the main activity module, the application has three other main modules:

- Routing
- Visual
- Audio

More on these modules is presented in the next sections. Though represented as standalone components, the remaining modules are inserted in the audio module logic, thus being approached in the context of such module, later in an implementation overview.

### **3.1.1 Routing module**

The routing module is responsible for establishing the next location (waypoint) where the system must guide the user to, based on the current location and the path to destination. The path is composed of several geographical coordinates (locations) that are loaded by the routing module as a set of waypoints. The goal is to navigate successive waypoints until the final destination.

### **3.1.2 Visual module**

This module provides the visual interface of the system, under the form of an arrow that points the direction the user has to take, besides some textual information. By interacting with the main activity module, this module obtains the value of direction to target location from the routing module periodically, and then performs the necessary calculations and procedures to rotate the arrow to the corresponding direction (angle) in the device's visual interface.

### **3.1.3 Audio module**

This is the crucial module of this work, since it is responsible for generating and playing the spatial audio cues to provide audio guidance to the user, which is the focus of this thesis. This module interacts with the main activity to acquire the direction it must reflect in the spatial audio instructions. The audio module is composed of several components, from classes to external libraries that interact with each other, as shown above in Figure 3.1, to generate and playback spatial audio cues.

## **3.2 Overall implementation of the system**

### **3.2.1 Routing module**

The routing module enables the AudioNavigator to navigate on a certain path based on waypoints. When this module is initialized, the next location variable (`nextLoc`) is defined, based on the location at the time relatively to the path. Each time the routing module is called, it compares the user's current location with `nextLoc` by calculating the distance

between both; if the distance is less than a predefined value (due to GPS error), nextLoc's value is updated to the actual next location.

Since routing is not the focus of the present work, the routing module is implemented to load a file containing a set of previously acquired geographical coordinates that constitute the path to be used in the application.

### 3.2.2 Visual module

This module is responsible for the information displayed in AudioNavigator's visual interface. That information contains the number of waypoints that compose the path to the target location, the distance to the next waypoint of the path (i.e., the waypoint to where the user is heading) and a compass-like arrow pointing the direction to the next waypoint, as shown in Figure 3.2. To point this arrow to the right direction, this module receives the bearing value from the main activity module and rotates the arrow from the previous direction to the new one. The distance to the next waypoint is also received from the main activity module, coming from the routing module, and is updated every 5 seconds, approximately (a time constraint ensured by a timer).

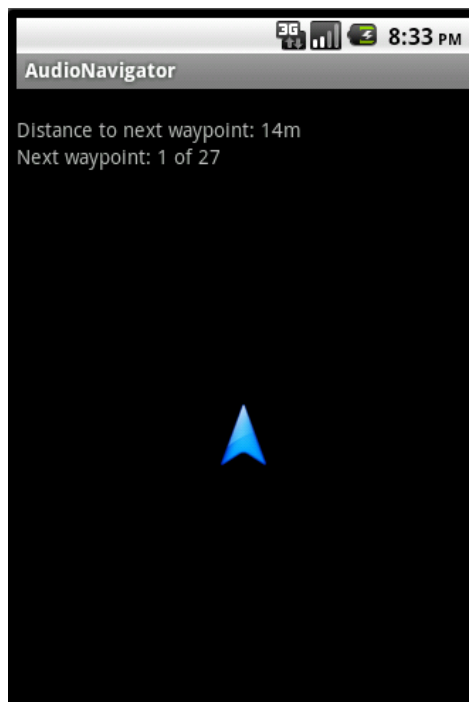


Figure 3.2: AudioNavigator's GUI.

### 3.2.3 Audio module

The audio module is the focus of the AudioNavigator system implementation given its importance and dimension. This module is responsible for the transformation of a simple monaural audio signal into a spatial audio signal. A description of this module's several components and the type of interaction between them follows.

## Native code

Android is a software stack composed of an operating system, middleware and key applications [21]. Although Android applications are usually developed in Java programming language using the Android Software Development Kit (SDK), it is possible to develop an application or parts of it in C or C++ programming languages, which are native-code languages, using the Android Native Development Kit (NDK). Developing native code benefits applications that, for example, reuse external libraries or perform operations such as signal processing [21], which is the case of the audio module here discussed.

The interaction between Java code and native code is defined and ensured by the Java Native Interface (JNI) [22]. This interface maps native data types to or from Java data types, both primitive (e.g. the JNI data type `jshort` corresponds to the native data type `short`) and others such as objects or arrays (e.g. `GetDoubleArrayElements()` returns a pointer to an array of `joubles`). JNI exposes the developed native code through dynamic shared libraries pre-compiled and provided by the Android NDK so that these libraries' code can be loaded by the Java code, thus becoming available for the application to use.

## NativeBridge

This class presents the available native functions that can be accessed inside the AudioNavigator application context. It contains the native functions declaration i.e. the function name, the required parameters and the return type. Note that the load and link of the libraries containing the code of the native functions into the application is done in the main activity class, not in `NativeBridge`.

## NativeAudio

This component holds the code of the native function exposed by JNI and presented in `NativeBridge`, combining C code with JNI's primitive types and methods to interact with the Java code, as mentioned above. Such function performs the filtering of a monaural audio signal in order to obtain spatial audio. As parameters, it receives the input signal - an array of integer values (short primitive data type) to be filtered, the sample rate of the signal and two arrays that act as delay lines for each of the channels (left and right) during the filtering process. Loading the HRTF coefficients to be used in the filtering process and the filtering process itself are the two core steps executed in this component. The HRTF coefficients and the input signal are sets of integers ranging between  $[-32768, 32767]$  that are mapped to  $[-1, 1]$  range (division by short's maximum value and cast to double) prior to the filtering process, to match its accepted parameters. The inverse mapping is performed after the filtering process finishes, in order to restore the signal's original data type, to match the audio data type accepted by the class responsible for audio playback - the `AudioTrack`.

## MIT HRTF lib

The MIT HRTF lib is a C library [23] that enables the access to the MIT KEMAR HRTF data set [16] - an extensive set of HRTF measurements of a KEMAR dummy head [24] microphone that consist of the left and right ear impulse responses (HRIR) from a loudspeaker mounted near the KEMAR head. This data set has 16-bit samples represented as integer numbers and provides different measures of 128, 140, 256 and 279 taps per channel at sampling



rates of 44100, 48000, 88200 and 96000Hz, respectively. The data is sorted by elevation (vertical plane), ranging from -40 to 90 degrees, and for each elevation there is a different set of azimuth values (horizontal plane) that can go from 0 to 180 degrees (e.g. 90 degree elevation has only a 0 degree azimuth) with a maximum resolution of 5 degrees interval; each data entry of a certain elevation and bearing has a pair of coefficient arrays corresponding to the left and right ear. Representing the other half of the horizontal plane (from -180 to 0 degrees) is a simple matter of switching the right and left coefficient sets.

This library is included in this module as a source of HRTF coefficients that are necessary to filter the audio signal according to the target bearing and elevation. To the purpose of this work, elevation is always equal to 0.

### **FIR filter**

As mentioned before in section 2.4.4, a FIR filter is a discrete-time system that converts an input audio signal into an output signal by convolving it with an impulse response given by the filter coefficients. A basic FIR filter implementation written in C programming language is used in this work, taking the MIT HRTF lib coefficients that match certain directions as the filter coefficients.

The FIR filter implementation was validated using a *MATLAB* implementation as a reference. Using the same set of values as input argument for the FIR filter function implementation already existent in the *MATLAB*, the results of both executions were compared and proven to be equal, validating the FIR filter implementation. The next step was to include this implementation's code in the NativeAudio and test it against *MATLAB* filter function. This was a necessary step due to the primitive data type conversions and other language differences that affect the interaction between Java and C code. A set with 29 double-precision values was filtered with the *MATLAB* filter function and with the filter function included in the Android application. The results obtained from the filter functions are basically the same, as evidenced by the resulting correlation equal to 1. The largest error between values from the resultant datasets was in the order of  $10^{-11}$ .

### **Audio**

The Audio module is a class directly connected to the main activity module with which it shares all information and resources regarding audio in this application. This class performs the load of the sound sources used in AudioNavigator - two WAVE [25] (.wav) audio files containing the same pluck sound but with different pitch, intended to solve the front-back confusion that frequently affect the users of this type of audio systems (each sound represents a semicircle, one located in front of the user and the other behind the user). However, these sound sources cannot be directly played nor manipulated by the application mechanisms responsible to do so, since these WAVE files contain more information besides the raw audio data (e.g. header); this raw audio data is a type of data known as Pulse Code Modulation (PCM) data. As so, the Audio class gathers the PCM data and other relevant information from the audio files using two other classes:

- Converter: parses the .wave file separating the header information from the PCM data, both accessible through a call of the appropriate functions.
- WavInfo: stores the relevant header information returned from the Converter class (data size, sample rate, bits per sample and number of channels).

The PCM data is loaded into an array of shorts, becoming available to be manipulated and played by the application. There are a few classes that can playback audio in the Android platform, but considering the need to manipulate (filter) audio data before playing it, the `AudioTrack` is the most appropriate one.

## **AudioTrack**

`AudioTrack` is a module implemented as a class available on the “`android.media`” package that allows the streaming of audio buffers to the hardware for playback [21], being only accepted buffers with byte or short data. This class has two operation modes: static and streaming. In static mode, the sound is loaded once and the size of that sound defines the maximum size the `AudioTrack` instance can play, thus being better suited to applications that play the same sounds often, which is not the case of this work. In streaming mode, the application can write a continuous stream of data to the `AudioTrack` through a blocking function that writes the data from the Java layer to the native layer and queues it for playback. Compared to the static mode, this operation has a bigger latency (delay) associated, but it is a more appropriate mechanism to implement the periodic audio filtering and consequent audio playback intended for this work’s addressed system.

Several parameters are required to instantiate an object of this class, such as the sample rate and the number of bits per sample values that were obtained from the source sounds. After initialized, the `AudioTrack` instance is available to receive the audio streams written to it, but given the blocking behaviour that characterizes this functionality in `AudioTrack`, this instance is executed in a separate thread - an instance of the `Player` class, to prevent the running thread from getting hanged.

## **Player**

As referred, the `Player` module was implemented as a Java Thread i.e. a concurrent unit of execution that runs concurrently with other application threads. A player thread is initialized with two arguments: the `AudioTrack` instance and the pre-filtered audio buffer that will be written to it. This writing is done when the player thread is run. But before running such thread, a listener (`AudioListener`) is associated to the `AudioTrack` instance so that the playback end is detected. This enables the application to be aware of the playback state (still playing or finished) and thus the `AudioTrack` instance state, to control if the stream writes can be performed or not at the moment and prevent the playback methods and underlying hardware from clogging. This mechanism seems unnecessary given the `AudioTrack`’s blocking function to write audio data that prevents new data writes while the previous write is not finished, but this feature failed sometimes while the application was run and there were unexpected buffer overwrites and application crashes.

## Chapter 4

# AudioNavigator evaluation

In this chapter we present the evaluation of the AudioNavigator system concerning its performance and ability to accomplish the goal of guiding using audio cues across a geolocalized path.

The AudioNavigator was evaluated according to:

- the AudioNavigator spatial cueing accuracy;
- the overall system performance in guidance accuracy.

### 4.1 AudioNavigator spatial cueing accuracy

The first step to evaluate the AudioNavigator system is to test if the user has a correct perception of the spatial audio it generates i.e., if the sound played by the system representing a certain direction is perceived by the user as a sound coming from that same direction.

#### 4.1.1 Spatial audio evaluation

Spatial audio is perceived as coming from a certain direction. In order to evaluate the users' perception of the AudioNavigator's spatial audio, we need to play sounds to the users and compare its direction with the direction perceived by them. To do so, a system to play a sound and record the users' answer about the direction of the sound they heard had to be implemented, consisting of two separate applications - one based on the AudioNavigator's audio module that plays spatial audio corresponding to a certain direction in degrees ( $^{\circ}$ ) and also records the users' response time, that is to be run on an Android device and is destined to be used by the tester; the other is to be used by the users in order for them to indicate their answers and record them (Figure 4.1).

The tester operates an extremely simple Graphical User Interface (GUI) based on two buttons, to play a sound and to record the time the user takes to answer which direction the played sound was coming from, respectively. The sound direction ranges from  $0^{\circ}$  to  $355^{\circ}$  in intervals of  $5^{\circ}$ , which is the maximum resolution of the spatial audio system used.

The time taken by a user to answer is measured comparing the time the sound playback finishes with the time the tester presses the "Time" button; the response times are written in a file and stored in the device. This response times can then be compared with the answers given

by the users to ascertain if there is a correlation between longer response times (hesitation) and direction errors (deviation).

"Learning" part	270		300		330		0		30		60		90	
Random part	100	180	315	0	210	45	335	260	315	60	120	0	150	310
	30	180	230	270	340	75	30	180	0	245	220	340	70	200

Table 4.1: Set of 35 sound directions (in degrees) used in spatial audio test.

The device played the sounds that were presented to the user. The sound sequence was fixed and hardcoded in the program to avoid any unintended change during the evaluation. A total of 35 sounds, representing 35 directions, compose the test set, as presented in Table 4.1. Besides these, 4 sounds are played before every test, corresponding to front ( $0^\circ$ ), extreme right ( $90^\circ$ ), back ( $180^\circ$ ) and extreme left ( $270^\circ$ ) of the user, acting as reference directions. Note that the sounds located behind the user (from  $90^\circ$  to  $270^\circ$ , exclusive) have a different pitch from the other sounds, to avoid front-back confusion as referred earlier in this thesis. The test set comprehends a "learning" part of 7 sounds that go from a direction of  $270^\circ$  to  $90^\circ$ , in  $30^\circ$  intervals, relatively to the direction faced by the user, as in a clockwise representation (pattern) of the sounds present in the virtual semicircle located in front of the user - from the extreme left to the extreme right of the user; the rest of the sounds were randomly chosen, to infer the user capability to identify the direction of sounds without any pattern associated. None of the sounds of the test set is repeated i.e. played again during the test.

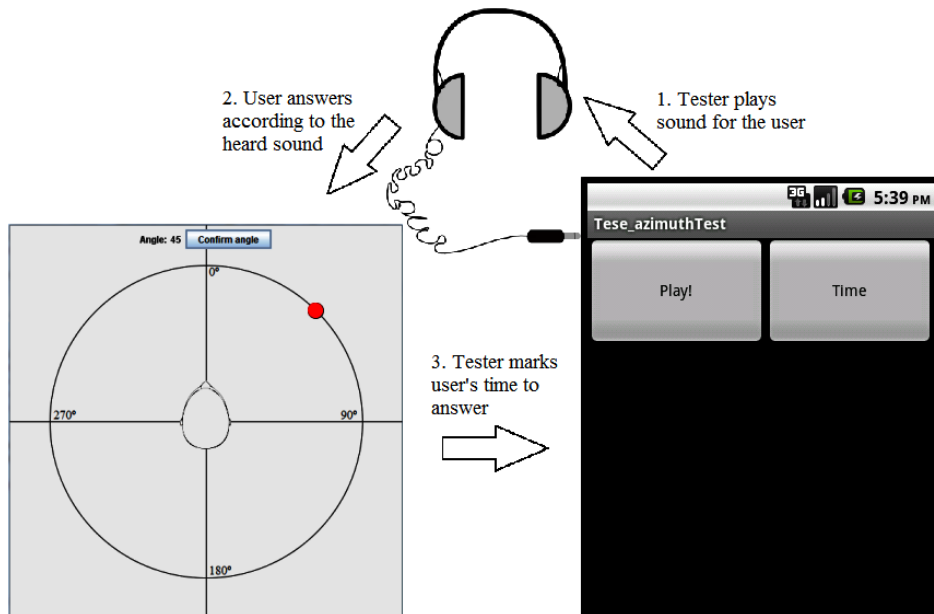


Figure 4.1: Spatial audio test - GUI for the participants (on the left), GUI for the tester (on the right) and test flow.

The users interact with a GUI that illustrates the virtual circle located around the user where the sounds come from, and where the user can point the direction from where a sound was played. Using the mouse, the user can click in a location and a red ball will point that

location and the angle between that location and the direction of the user’s head will be displayed, as shown in Figure 4.1. The user can then check the indicated angle and confirm the selected answer by clicking the “Confirm” button or rectify the answer. These answers are then written and saved in a file with the user’s name, a parameter initially passed to the program.

The test was performed with 10 subjects, one at a time, with ages between 22 and 25 years old, who assured to have no history of auditory problems / disorders. It took place in a silent room, to avoid noise that could interfere with the subjects’ auditory perception. The same Android device and the same computer to answer the test were used in every test, and all subjects wore the same pair of full-size headphones (Philips SHP1900) during the test, to minimize the differences between user test conditions other than the individual hearing perception.

#### 4.1.2 Results

Based on the answers of the 10 subjects about their perception of the 35 played sounds, a mean between the answered angle values was calculated for each of the sounds and then compared with the actual angle values. Figure 4.2 presents the test results.

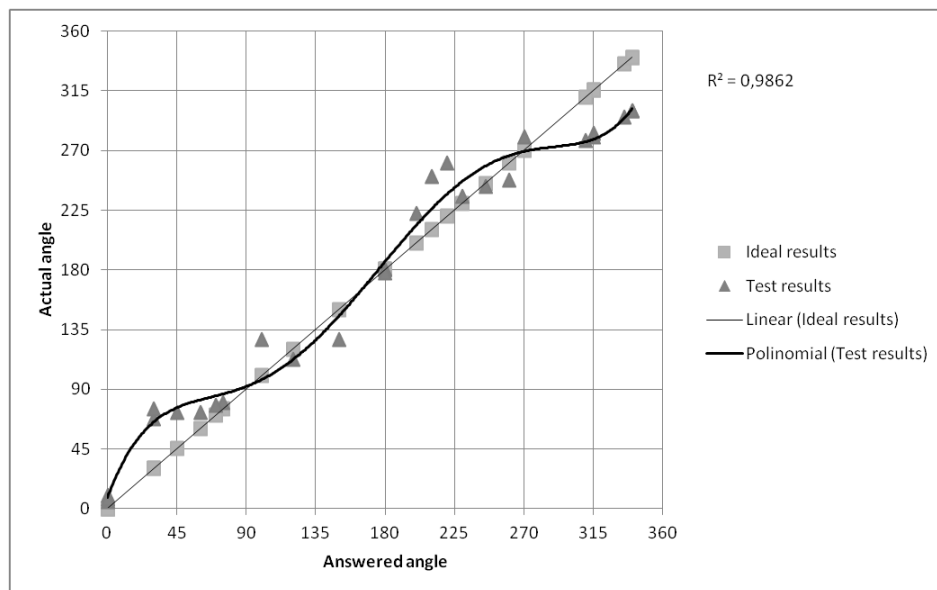


Figure 4.2: Test results compared to ideal results (the 45° line) - without “learning” part.

Among all the subjects the performance was not particularly affected by the initial familiarization with the system as supported by similar results in test with and without the “learning” part - the initial 7 sounds. The answers’ average error (deviation) was 28.5°. The tendency line of the test results shows that the answers relative to the 0°, 90°, 180° and 270° angles tend to the ideal results, which is an expected outcome since these angles correspond to the reference directions of the subjects - front, extreme right, back and extreme left, respectively.

Although slight, there is a noticeable difference in the performance of the subjects when

perceiving the sounds coming from their front ( $[270^\circ, 0^\circ]$  and  $[0^\circ, 90^\circ]$ ) compared to the sounds coming from behind them ( $[90^\circ, 270^\circ]$ ). One might say that this occurs possibly due to the weight of the “learning” part of the test, since this part is composed of sounds located in front of the user, but the difference is also present in the results without the “learning” part (Figure 4.2), which excludes this possibility. The explanation resides in the pitch difference between the sounds played as coming from the user’s front (higher tone) and the sounds coming from behind the user (lower tone). This pitch difference aims to avoid the front-back confusion that occurs in this type of audio systems, as referred before, but as it turns out it also affects the perception accuracy of the user.

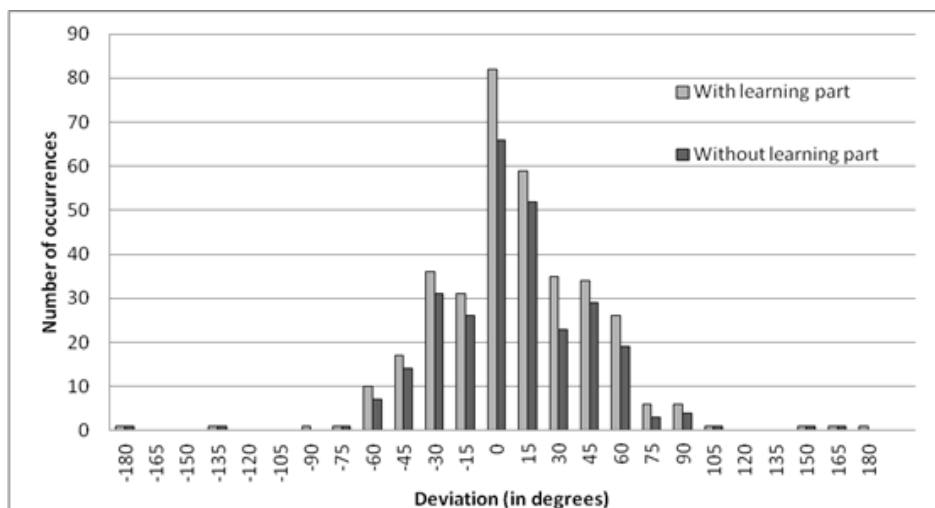


Figure 4.3: Histogram of deviations from ideal results (in degrees).

Figure 4.3 presents an histogram that matches the deviations of the users answers (relatively to the ideal results) with the number of occurrences of such deviations. The deviation values are presented in intervals of 15 degrees, which seem an appropriate resolution given the average deviation of the results ( $28.5^\circ$ ); these values range from  $-180^\circ$  to  $180^\circ$ , with positive and negative values representing deviation in clockwise and anticlockwise direction, respectively. The highest occurrences lay between  $0^\circ$  and  $30^\circ$ , which is consistent with the  $28.5^\circ$  average deviation. Nevertheless, many  $45\text{-}60^\circ$  deviations in clockwise direction occurred during the test, suggesting a certain difficulty in perceiving the correct location of the sounds produced by the AudioNavigator. Comparing the results’ deviations with and without “learning” part we can realize that these two differ in almost every deviation intervals, thus the subjects’ familiarization period with the system does not match the highest deviations (errors), contrary to what would be expected.

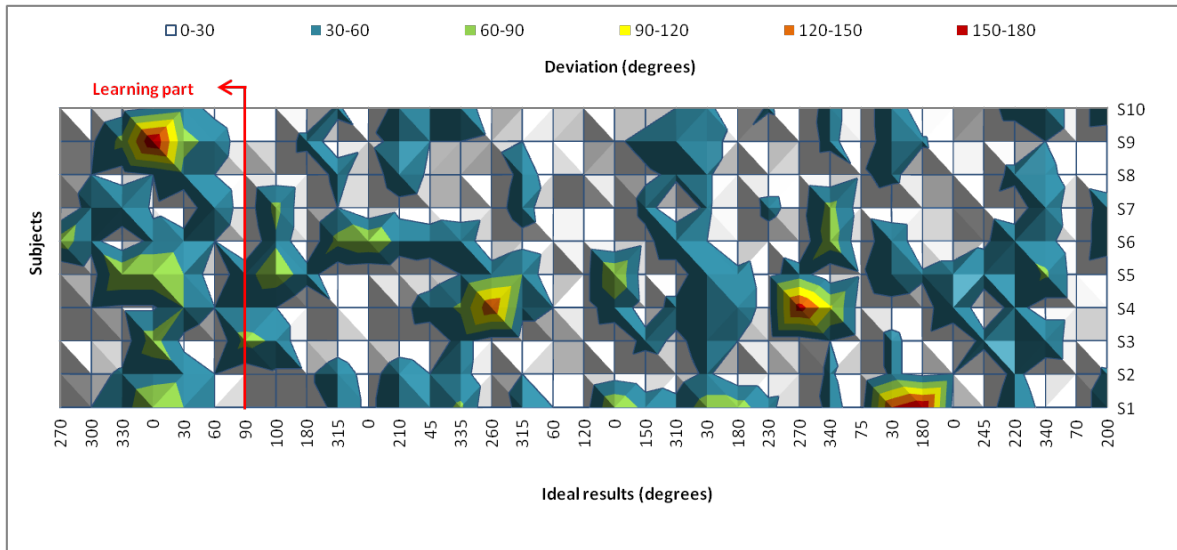


Figure 4.4: Deviations (in degrees) from the ideal test values for each subject ('S').

Figure 4.4 establishes the relation between deviations (different colors), ideal results (horizontal axis) and each subject (vertical axis). The goal is to analyze the chart and unveil if certain subjects tend to deviate to the same angles or performed the test with a regular deviation. Analyzing the chart regarding the “learning” part, the fourth value (0°) obtained high deviations, especially from subject 9 (150-180° deviation) and subjects 1, 3 and 5 (60-90° deviation), while subjects 2, 6, 8 and 10 deviated 30-60° and subjects 4 and 7 deviated 0-30° from the ideal values. There were fewer deviations in the other occurrences of the 0° value, with subjects 1, 5 and 6 with 60-90° deviation and subjects 4 (twice) and 5 with 30-60° deviation, while the rest deviated 0-30°. In particular, subject 4 answers highly deviated from the ideal values for two times during the test (for sounds coming from 260° and 270°), but for similar values the answers barely deviated, suggesting a left-right confusion or a lack of attention from the user. The answers of subjects 1 (during “learning” part) and 9 highly deviated for 0°, 30° and 180° suggesting the occurrence of front-back confusion, even though the sounds coming from the front and back of the user have different pitch to avoid this sort of confusion, as explained before.

As shown in Figure 4.5 and visible in Table 4.2, the gradient of the lines relating the values of deviation with the users’ response time is positive in 80% of the cases i.e. higher response times corresponded to higher deviations from the actual angle, which might indicate that the users answer more incorrectly when they hesitate to answer. The remaining cases took less time to answer more incorrectly i.e. lower response times corresponded to higher deviations, which might have occurred due to some rush in answering. Nevertheless, the obtained results are very sparse and the resultant line gradients are very lean, preventing us from taking irrevocable conclusions.

Subject	Deviation / error (degrees)		Time to answer (seconds)		Gradient
	Average	Std.deviation	Average	Std.Deviation	
1	42,57143	39,87797	4,05945	1,53657	0,0055
2	21,85714	15,53392	4,27556	1,221962	0,0043
3	22,14286	19,93688	3,06992	1,307724	0,0098
4	37	35,77298	2,68745	1,192704	0,0008
5	36,57143	31,03062	4,68023	2,636903	0,0113
6	30,85714	26,85708	4,28572	2,746226	0,0145
7	22,57143	22,30706	4,94029	3,276315	0,0467
8	21,14286	14,50587	2,95002	1,14723	-0,0205
9	30	32,31281	5,10589	2,236701	0,0147
10	22	19,78562	3,30273	1,035626	-0,0108

Table 4.2: Answer deviation *vs.* time to answer - for each subject.

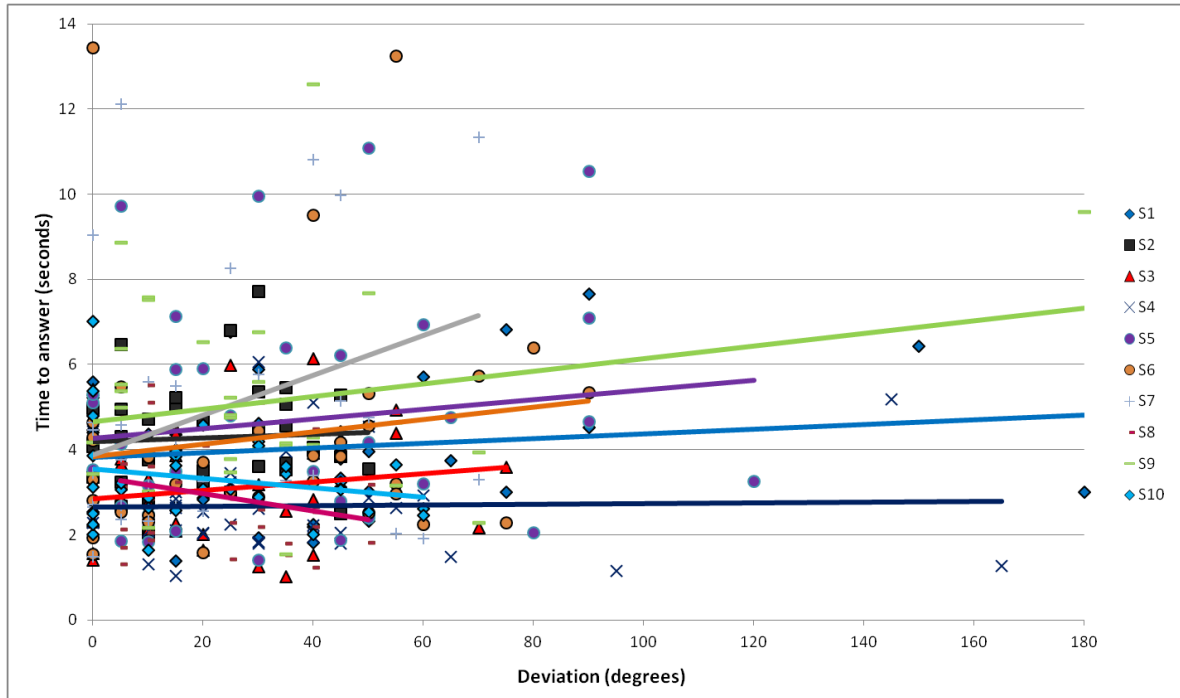


Figure 4.5: Comparison between response time and answers deviation.



## 4.2 Overall system performance in guidance accuracy

Our first objective was to test the AudioNavigator efficiency to guide an individual towards a target location. The purpose is to measure how accurate are the user decisions when based on the AudioNavigator audio cues (supported by the routing module) to “point” the direction to a target location, particularly when the user has to decide which path to take when facing a junction, to realize if the user follows the suggested way or not.

This test is a field test i.e., it is meant to be held outside, where the sounds coming from the surrounding environment may interfere with the AudioNavigator audio cues making it harder for the user to correctly distinguish its direction, as an approach to a real life situation.

### 4.2.1 Overall system evaluation

Using the same Android device as in the first test and the earphones included with the device (instead of the full-size headphones used in the first test), this test relies on the AudioNavigator system to provide spatial audio guidance to the user. The goal is to determine if a user can navigate to a target location only with audio guidance support. To ensure the user is guided only by the audio cues, the compass-like arrow present in the visual interface was removed, remaining the information relative to the total number of waypoints and the distance to the next waypoint. The user holds the mobile device in one hand, aligned with the body, as shown in Figure 4.6. This setup was found to be adequate for the test purpose, without the need to fix the device to the user’s body or head. The test consists of two separate paths the user has to walk, one at a time, guided by the audio cues provided by the system. AudioNavigator keeps track of the user location during each test, allowing the comparison of the user path with the ideal path, afterwards.



Figure 4.6: User wearing stereophones and holding the mobile device.

The user's location is obtained from the GPS receiver embedded in the Android device and processed by the AudioNavigator's routing module that determines the direction to the next waypoint of the path the user shall take. This comprehends technology constrains such as the error inherent to the device's GPS receiver, which accuracy was found to be 5 meters in the best case (obtained in a wide, open space with clear sky). It is therefore necessary to compensate such inaccuracy by establishing a radius of 5 meters around each waypoint, to minimize the need for the user to be "looking back and forth" for the exact waypoint coordinates, when the idea is simply to guide him through a defined path. During the test the user location is tracked and the geographical coordinates are written to a file and saved in the mobile device. This data is then inserted and visualized using Google Fusion Tables [26], a data visualization web application that accepts simple text files with geographical coordinates, aligned as in a table, and enables its representation in a map.

The test was performed with 5 subjects, with ages between 23 and 25 years old, which also participated in the first test, thus being already familiarized with the system. Every subject performed the test in the same conditions, travelling the same two paths without knowing the target location or the route to take.



Figure 4.7: Test results (colored lines) compared to ideal path (red dots) - path 1.

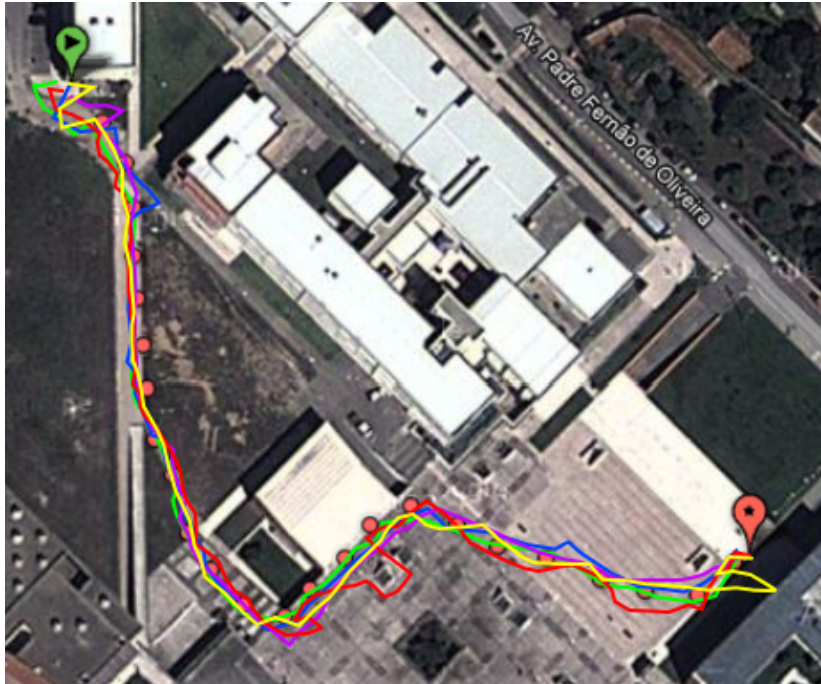


Figure 4.8: Test results (colored lines) compared to ideal path (red dots) - path 2.

Regarding the guidance test results, the first point to mention is that every subject reached the target destination successfully, as illustrated in Figures 4.7 and 4.8. Even with (at least) a 5m inaccuracy in GPS data and the noise surrounding the subjects (people talking, wind blowing), they navigated according to the reference route, i.e. the route used to guide the users, with minor deviations along the majority of the path. In almost every test there was a noticeable deviation from the reference route between the starting point and the first points, corresponding to the period in which the device is acquiring the GPS signal to obtain the user's location with the best possible accuracy. The remaining major deviations occur in turning points or near buildings, which suggests that the routing module might not be describing the turn points as accurately and/or smoothly as desirable and that the buildings interfere with the GPS signal reception, decreasing its accuracy.

## 4.3 Discussion on the results

### 4.3.1 Spatial audio evaluation

As expected, regarding the spatial audio evaluation results, it is noticeable that each user has a different perception of sound location and also react differently (e.g. hesitate vs. rush to answer). In spite of the "learning" part that was included in the evaluation and the previous explanation about the different sounds, some of the users confused sounds represented in their front with sounds coming from behind and vice-versa, a situation that reveals a certain lack of attention and not just a misinterpretation of the sound location. By the time of the preliminary tests (during the AudioNavigator development) the margin of error was considered to be around  $30^\circ$ , a value that was then confirmed to be close to the average deviation of  $28.5^\circ$  obtained from the results, almost the same as the  $26.74^\circ$  average

deviation considered in [19] as a reasonable measure for categorizing a correct localization in the horizontal plane; this seems to be an acceptable value considering that the reference directions (front, extreme right, back and extreme left) differ  $90^\circ$  from each other, which leaves two intermediate directions per each of the four quadrants besides the reference ones. Surprisingly, even though the following happened during the “learning” part, the angle about which the users answered with higher deviation was  $0^\circ$ , an angle that was expected to be well perceived from the start.

As for the response time, although slight, the values’ tendency suggests that, for the majority of the users, the longer they take to respond, the greater the deviation from the ideal answer. Using the stereophones included with the mobile device, as in this work, the standard deviation of the results obtained in [19] for a similar test was  $51.0139^\circ$ , higher than the  $39.8790^\circ$  corresponding to the highest standard deviation obtained in this work (Table 4.2).

### 4.3.2 Overall system evaluation

This field evaluation demonstrates the potential of the AudioNavigator, with all subjects reaching the final location guided only by the provided audio cues. The largest difference between the users’ paths is noticeably at the beginning of each test, with some users navigating to one direction while the others go to the opposite direction, a behaviour caused by the “warm-up” period in which the GPS receiver of the mobile device is trying to acquire the most precise geographical coordinates possible. The other significant deviations from the original path take place in turning points, suggesting that the routing module might not be describing steeper direction changes as accurately and / or smoothly as it should, and near to buildings, where the GPS signal reception is decreased. While the deviations from the ideal path are undesirable, they show that the AudioNavigator is able to correct the user and guide him back on the right path, particularly when the next location is behind the user causing AudioNavigator to play a sound with different pitch (front-back disambiguation), a noticeable and effective change that enables the quick orientation correction by the user, much like what happens in [5] where the sound timbre is changed to emphasize that the waypoint is now behind the listener.

# Chapter 5

## Conclusions

In this chapter we identify the accomplishments achieved with AudioNavigator and describe related issues that may be addressed as future work.

### 5.1 Accomplishments

With this work, we approached the thematic of guidance with the support of audio cues, making use of the new technologies regarding mobile devices. As proof-of-concept, the developed software application - AudioNavigator - proves that the current widely available mobile phones (smarphones) are capable of providing audio guidance with acceptable performance, as shown in the obtained results and supported in [18] about the mobile phones capability of running 3D audio interfaces. AudioNavigator is capable of guiding an individual towards the target destination solely relying on the audio guidance cues it provides. As concluded in [19], the stereophones distributed with the Android device and used during the overall AudioNavigator evaluation (section 4.2.1) are suitable for binaural 3D audio playback, although they are not as accurate as the full-size headphones used during the spatial audio evaluation (section 4.1.1). The results in this work also confirm results from literature: the location of short periodic non-speech spatialized sounds can be correctly perceived by the listener while maintaining his awareness of the surrounding noises, instead of continuous synthesized spatialized speech [1][13].

Regarding the spatial audio provided, AudioNavigator is comparable to the system implemented in [19], since AudioNavigator obtained similar average error but significantly lower standard deviations of the error, which indicates that the variation of the error between answers was lower in the case of AudioNavigator.

Regarding the overall system, although using different binaural models, AudioNavigator provides a reliable audio guidance solution much like the system developed in [7], but with the advantage of including a front-back disambiguation mechanism.

Nevertheless, in spite of the AudioNavigator capability to guide a user to a certain location by providing spatial audio cues based on the use of HRTFs, it is not a fully-functional 3D audio system since there was a need to implement such front-back disambiguation mechanism, in order to avoid front-back confusion.

## 5.2 Issues

With the development of the Audio module, it was made clear that the Android OS version 2.2 that was used lacks a powerful mechanism that allows audio manipulation. The chosen mechanism to be used in AudioNavigator - the `AudioTrack` class - was found to be the most adequate, but still, it accepts only one audio format (raw audio data), it inserts high latency in audio playback, which affects the periodicity of audio playback, and its playback head position listener does not provide a notification for all the positions it should.

Also, the geographic coordinates obtained from the GPS receiver are inaccurate and tend to be dispersed from one another, a critical issue when the objective is to match the current location with the next waypoint that is part of the ideal route, even though the routing module was not the focus of this work.

In the end, among other limitations, this work attempts to overcome the referred ones resorting to timing mechanisms to force a certain periodicity of audio playback and performs a comparison of the received geographic coordinates to obtain a mean location closer to the real one.

## 5.3 Future Work

This section proposes further research on this thesis' context and further developments that can improve the performance of AudioNavigator.

### 5.3.1 Sound perception - further research

Further research on whether the perception of each subject highly differs from each other or not and more detailed perception evaluation could help improve the performance of AudioNavigator.

### 5.3.2 Routing module

The routing module is undoubtedly the module that has more potential for further development, since it is currently implemented simply to show the performance of the audio module. Improving the behaviour of this module when the GPS receiver starts receiving signal, implementing a more robust navigation mechanism to follow a certain route other than based on waypoints or improving the way the current location is compared with the waypoints, and smoothing the subtle direction changes are some of the proposals for the further development of this module.

### 5.3.3 Audio playback

A possible strategy to reduce the latency of audio playback would be to develop a mechanism for passing the audio signal filtered in the native layer directly to the audio hardware, avoiding the need to pass it through an `AudioTrack` instance.

### 5.3.4 Stereophones vs. scenario

Even though two different types of stereophones (earphones and full-size headphones) were used during the evaluations performed in this work, a comparison between the use of

different stereophones in several different scenarios might provide relevant results about the audio perception of the users in different conditions and setups.





# References

- [1] Jeffrey R. Blum, Mathieu Bouchard, and Jeremy R. Cooperstock. What's around me? spatialized audio augmented reality for blind users with a smartphone. In *Proceedings of the 8th annual international conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, MobiQuitous '11, 2011.
- [2] William G. Gardner. Spatial audio reproduction: Towards individualized binaural sound. *Frontiers Of Engineering*, 34:113–122, 2004.
- [3] Naizatul Akmal Karizaman. Teaching human senses to standard one students' using cognitive theory. <http://eprints.ptar.uitm.edu.my/1711/>, 2006. Bachelor's thesis.
- [4] Simon Holland, David R. Morse, and Henrik Gedenryd. AudioGPS: spatial audio navigation with a minimal attention interface. *Personal and Ubiquitous Computing*, 6(4):253–259, September 2002.
- [5] J. Wilson, B.N. Walker, J. Lindsay, C. Cambias, and F. Dellaert. SWAN: system for wearable audio navigation. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 91–98, 2007.
- [6] Christina Dicke, Katrin Wolf, and Yaroslav Tal. Foogues: eyes-free interaction for smartphones. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, MobileHCI '10, pages 455–458, 2010.
- [7] Matthijs Zwinderman, Tanya Zavialova, Daniel Tetteroo, and Paul Lehouck. Oh music, where art thou? In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 533–538, 2011.
- [8] P.D. Groves. *Principles of GNSS, inertial, and multi-sensor integrated navigation systems*. GNSS technology and applications series. Artech House, 2008.
- [9] Juan C. Herrera, Daniel B. Work, Ryan Herring, Xuegang (Jeff) Ban, Quinn Jacobson, and Alexandre M. Bayen. Evaluation of traffic data obtained via GPS-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies*, 18(4):568–583, August 2010.
- [10] Mário Tiago Pereira Vasconcelos Freitas. A novel visualisation paradigm for three-dimensional map-based mobile services. Master's thesis, February 2012.

- [11] Shiri Azenkot, Sanjana Prasain, Alan Borning, Emily Fortuna, Richard E. Ladner, and Jacob O. Wobbrock. Enhancing independence and safety for blind and deaf-blind public transit riders. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 3247–3256, 2011.
- [12] Jack M. Loomis, James R. Marston, Reginald G. Golledge, and Roberta L. Klatzky. Personal guidance system for people with visual impairment: A comparison of spatial displays for route guidance. *Journal of visual impairment & blindness*, 99(4):219–232, 2005.
- [13] Roberta L. Klatzky, James R. Marston, Nicholas A. Giudice, Reginald G. Golledge, and Jack M. Loomis. Cognitive load of navigating without vision when guided by virtual sound versus spatial language. *Journal of Experimental Psychology: Applied*, 12(4):223–232, 2006.
- [14] M. Linell. *Comparison between two 3d-sound engines of the accuracy in determining the position of a source*. Master’s thesis, Lule U. of Technology, 2011.
- [15] Wen Zhang, R.A. Kennedy, and T.D. Abhayapala. Efficient continuous HRTF model using data independent basis functions: Experimentally guided approach. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(4):819–829, May 2009.
- [16] Bill Gardner and Keith Martin. HRTF measurements of a KEMAR Dummy-Head microphone. Technical Report 280, MIT Media Lab Perceptual Computing, 1994.
- [17] Samuel Sandberg, Calle Hkansson, Niklas Elmqvist, Philippas Tsigas, and Fang Chen. Using 3D audio guidance to locate indoor static objects. *Human Factors and Ergonomics Society Annual Meeting Proceedings*, 50(16):1581–1584, 2006.
- [18] Yolanda Vazquez-Alvarez and Stephen Brewster. Investigating background & foreground interactions using spatial audio cues. In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, CHI EA '09, pages 3823–3828, 2009.
- [19] Jens Nornberg Paaske. *3D audio rendering on the Android platform*. PhD thesis, Aalborg Universitet, May 2011.
- [20] J.H. McClellan, R.W. Schafer, and M.A. Yoder. *Signal Processing First*. Pearson Education International, 2003.
- [21] Android developers. <http://developer.android.com/index.html>.
- [22] Java native interface. <http://docs.oracle.com/javase/1.5.0/docs/guide/jni/>.
- [23] Aristotel Digenis. mit\_hrtf\_lib - MIT kemar HRTF c library. [http://www.digenis.co.uk/mit\\_hrtf\\_lib/index.html](http://www.digenis.co.uk/mit_hrtf_lib/index.html).
- [24] M. D. Burkhard and R. M. Sachs. Anthropometric manikin for acoustic research. *The Journal of the Acoustical Society of America*, 58(1):214–222, 1975.
- [25] WAVE soundfile format. <http://ccrma.stanford.edu/courses/422/projects/WaveFormat>.
- [26] Google drive. <http://www.google.com/fusiontables/Home/>.