2013

An Artificial Intelligence Approach to Concatenative Sound Synthesis

Mohd Norowi, Noris

AN ARTIFICIAL INTELLIGENCE APPROACH TO CONCATENATIVE SOUND SYNTHESIS

by

NORIS MOHD NOROWI

A thesis submitted to the University of Plymouth

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

Faculty of Arts

July 2013

Abstract

An Artificial Intelligence Approach to Concatenative Sound Synthesis

Noris Mohd Norowi

Technological advancement such as the increase in processing power, hard disk capacity and network bandwidth has opened up many exciting new techniques to synthesise sounds, one of which is Concatenative Sound Synthesis (CSS). CSS uses data-driven method to synthesise new sounds from a large corpus of small sound snippets. This technique closely resembles the art of mosaicing, where small tiles are arranged together to create a larger image. A 'target' sound is often specified by users so that segments in the database that match those of the target sound can be identified and then concatenated together to generate the output sound.

Whilst the practicality of CSS in synthesising sounds currently looks promising, there are still areas to be explored and improved, in particular the algorithm that is used to find the matching segments in the database. One of the main issues in CSS is the basis of similarity, as there are many perceptual attributes which sound similarity can be based on, for example it can be based on timbre, loudness, rhythm, and tempo and so on. An ideal CSS system needs to be able to decipher which of these perceptual attributes are anticipated by the users and then accommodate them by synthesising sounds that are similar with respect to the particular attribute. Failure to communicate the basis of sound similarity between the user and the CSS system generally results in output that mismatches the sound which has been envisioned by the user. In order to understand how humans perceive sound similarity, several elements that affected sound similarity judgment were first investigated. Of the four elements tested (timbre, melody, loudness, tempo), it was found that the basis of similarity is dependent on humans' musical training where musicians based similarity on the timbral information, whilst non-musicians rely on melodic information. Thus, for the rest of the study, only features that represent the timbral information were included, as musicians are the target user for the findings of this study.

Another issue with the current state of CSS systems is the user control flexibility, in particular during segment matching, where features can be assigned with different weights depending on their importance to the search. Typically, the weights (in some existing CSS systems that support the weight assigning mechanism) can only be assigned manually, resulting in a process that is both labour intensive and time consuming. Additionally, another problem was identified in this study, which is the lack of mechanism to handle homosonic and equidistant segments. These conditions arise when too few features are compared causing otherwise aurally different sounds to be represented by the same sonic values, or can also be a result of rounding off the values of the features extracted. This study addresses both of these problems through an extended use of Artificial Intelligence (AI). The Analysis Hierarchy Process (AHP) is employed to enable order dependent features selection, allowing weights to be assigned for each audio feature according to their relative importance. Concatenation distance is used to overcome the issues with homosonic and equidistant sound segments.

The inclusion of AI results in a more intelligent system that can better handle tedious tasks and minimize human error, allowing users (composers) to worry less of the mundane tasks, and focusing more on the creative aspects of music making.

In addition to the above, this study also aims to enhance user control flexibility in a CSS system and improve similarity result. The key factors that affect the synthesis results of CSS were first identified and then included as parametric options which users can control in order to communicate their intended creations to the system to synthesise. Comprehensive evaluations were carried out to validate the feasibility and effectiveness of the proposed solutions (timbral-based features set, AHP, and concatenation distance). The final part of the study investigates the relationship between perceived sound similarity and perceived sound interestingness. A new framework that integrates all these solutions, the query-based CSS framework, was then proposed. The proof-of-concept of this study, *ConQuer,* was developed based on this framework.

This study has critically analysed the problems in existing CSS systems. Novel solutions have been proposed to overcome them and their effectiveness has been tested and discussed, and these are also the main contributions of this study.

Table of Contents

List of Figures

List of Tables

Abbreviations

AHP	Analysis Hierarchy Process
AI	Artificial Intelligence
AIFF	Audio Interchange File Format
ASCII	American Standard Code for Information Interchange
BMP	Beat Per Minute
CD	Complex-Domain Distance
CI	Consistency Index
CR	Consistency Ratio
CSP	Constraint Satisfaction Programming
CSS	Concatenative Sound Synthesis
FT	Fourier Transform
GA	Genetic Algorithm
HFC	High Frequency Content
IS	Intelligent System
KL	Kullback-Liebler Distance
KNN	K-Nearest Neighbour
MARSYAS	Music Analysis, Retrieval and Synthesis for Audio Signal
MFCC	Mel-Frequency Cepstral Coefficient
MIDI	Musical Instrument Digital Interface
MP3	MPEG-1 or MPEG-2 Layer III
MPEG	Moving Pictures Expert Group
MPEG-7	Multimedia Content Description Interface
PCA	Principal Component Analysis
PD	Phase Deviation
PSOLA	Pitch Synchronous Overlap Add Method

PPMC	Pearson Product-Moment Correlation Coefficient
RI	Random Consistency Index
RMS	Root Mean Square
SD	Spectral Differences
SDIF	Sound Description Interchange Format
SoX	Sound eXchange
STFT	Short Term Fourier Transfer
TTS	Text-to-Speech
WAV	Waveform Audio File Format
ZCR	Zero Crossing Rate

Acknowledgements

At this moment of accomplishment, I would like to express my heartfelt gratitude towards my Director of Study, Professor Eduardo R. Miranda for believing in me and continuously providing me with support, encouragement and invaluable suggestions during this research. I would also like to include my gratitude towards Dr. John Matthias and Dr. Geoff Cox for the constructive comments and contributions that they have provided me with. I could not be prouder of my academic roots and hope that I can in turn pass on the research values and the dreams that they have given to me.

I am also thankful towards my examiners, Professor Stephen Davismoon and Dr. David Bessell for the constructive criticism and valuable comments of my work.

I take this opportunity to sincerely acknowledge and thank the Malaysian Ministry of Higher Education (MOHE) and my employer, Universiti Putra Malaysia for the scholarship and financial aid that were awarded to me in order to enable this research to be carried out.

I thank my ICCMR members, especially Alexis Kirke, Hanns Holger Rutz (and wife Naya), Jaime Serquera, Anna Troisi, Antonino Chiaramonte, Leandro Costalonga, Marcelo Gimenes, Joao Martins, Duncan Williams, and Marianna Blosche for sharing the helpful discussions and friendship. Also a warm thank you to the Malaysian community in Plymouth who had been there during the times we needed help most, in particular Zali, Wak, Mail, Kak Mazni, Shafie, Dr. Azli, Bard, Shue, Zarul, Afham, Kak Ad and their families. My appreciation also goes towards my closest friends – Che Fadh, Eti, Iezma, Mas, Jia, Lia, Azri and Shakir _ for their help and support, both in technical and emotional terms.

Heartiest thanks also towards these wonderful group of people that I had privilege of meeting and befriending – Mummy friends whom I've made at my daughter's school ground (Maya, Lisa, Andrea and Zahraa); the cleaning ladies at Theatre Royal Plymouth (Rhona, Debbie, Lynn, Pat, June and the gang); and the cleaning team at the University of Plymouth (Kim, Barbara, Jean, John and the rest) where both my husband and I had worked part-time in order to make ends meet during my financially challenging time as a graduate student;

thank you for the shared jokes and laughter. With them around, life has been less stressful, for they have made me momentarily forget the burden that was constantly at the back of my head and to also realise that there are more important things in life other than my thesis!

My sincerest thank you goes to the staff of Graduate School, University of Plymouth – Dr. Cristina Rivas, Anne Treeby, Sarah Kearne, Julia Crocker, Anna Jonson, Rosie Beck, Rebecca Rose, Sarah Carne, Susan Matheron, Tim Batchelor, Lucy Cheetham, Carole Watson, Catherine Johnson, and the rest – for not only being there for me in their capacity as Graduate School officers, but also as friends that provided me with encouragement and pointed me in the right direction. Their service has truly been above and beyond what is expected. My family and I are forever indebted to the Dean of the Graduate School, Professor Mick Fuller for the trust he has given my husband to undertake a 15-month paid attachment programme at the University. Without it, life as a PhD student for me, I believe, would not have ended as smoothly as it has.

Finally I want to thank my large family ‒ my Dad, Dr. Mohd Norowi Hamid whom I have always looked up to, for his consoling advice and support; my Mom, Azizah Ismail, for her love and prayers; my in-laws (Maskan Basiran and Siti Ali) for their understanding and well-wishes; and my siblings for their care. To my beloved husband, my best friend, my mentor, Saiful “Sam” Maskan, I can never thank you enough for all the love and encouragement provided during this challenging journey. Without you by my side, this quest would never have ended as victoriously. The same goes for my sweet daughter Aisyah, who has been my most powerful source of joy and inspiration, who taught me more about life than any degree possibly could. I am also thankful that at the most difficult part of the study (the final six months of thesis writing), I had been blessed with a loyal companion inside me who had never failed to provide me with amusing kicks, cheering me to keep going on, especially during the wee hours of morning when the bulk of this thesis was written – my son, Zayd. Born exactly four weeks after my viva, he is the epitome of the term ‘thesis baby’ and has been the positive catalyst from within me to race towards the finishing line (belly first!).

Above all, I want to thank God for it is His grace and blessings that has made it possible for me to be where I am and achieved what I have today, *Alhamdulillah…*

Author's Declaration

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Graduate Committee.

This study was financed with the aid of scholarships from the Ministry of Higher Education Malaysia and the Universiti Putra Malaysia.

A record of activity is attached with this thesis detailing the papers which have been published and other research activities that were undertaken during the course of this study are included in Appendix F (p.238).

Word count of main body of thesis: 45054 words

Signed……

Date……..

Chapter 1: Introduction

This first chapter introduces the motivation, background and the gaps and challenges that exist in this study. The objectives of the study comprised of proposed solutions to overcome the problems are also presented. At the end of the chapter, an outline of thesis structure is given.

1.1 Motivation

From a young age, I have always found the folk music of Malaysia to be fascinating. Despite it being a very peculiar music of choice for youngsters in Malaysia at the time due to the cultural shift that gave way to the more westernise musical genres such as pop and rock, my home was never quiet from the sounds of different traditional Malaysian music playing in the background, especially that of *Dikir Barat* and *Wayang Kulit* genres. The musical preferences of my parents had somewhat influenced my taste in music. The passion grew stronger as I got older, and for my Masters, I had proposed, argued the need for and developed an automated system that could classify traditional Malaysian music into one of eight genres, namely *Dikir Barat*, *Etnik Sabah*, *Gamelan*, *Inang*, *Joget*, *Keroncong*, *Tumbuk Kalang*, *Wayang Kulit* and *Zapin* (Norowi *et al.*, 2005). Further reading on the subject of traditional Malaysian music can be found in Nasuruddin's work (Nasuruddin, 2003).

Out of the eight genres, I have a special interest in *Gamelan*, as I took a much formal path in studying and performing the art as an elective course in the third year of my degree programme. Malay gamelan is different than that of Javanese or Balinese gamelan, not so much in the instruments included in the ensemble, but in the way the music is played. Malay gamelan is missing the intricately locked parts that are found in both Javanese and Balinese gamelan. Instead, all its instruments play the melody, which translates into a much simpler

play (Ahmad, 1997). The gamelan was first brought over to the state of Pahang in Malaysia from Riau-Lingga (islands from the Indonesian archipelago) circa the early 1800s. It then spread over to the neighbouring state, Terengganu, through the royal marriage. Of the many original songs brought over, only twelve were notated and regularly performed today (Ariffin, 1990). Like any other traditional Malaysian music, Malay gamelan pieces are passed aurally from generation to generation, and are often carried to their graves by the original players. The influence of western music further de-emphasises its appeal to the average Malaysian listeners.

I had thought of how wonderful it would be if these 'missing' songs could be recreated from the original pieces that survived. Perhaps this would help revive the interest in gamelan for the younger generation of Malaysia. However, I quickly realised that a rule-based composition was not the way forward, seeing that the number of surviving pieces are too small to generate the rules for which new sounds would be composed from. Instead, I thought of approaching this differently, rather than recreating something that was missing, I could experiment composing new gamelan pieces from small cut up segments of existing pieces, or even using the original gamelan songs as targets to compose new gamelan-like sounds from a corpus of different other sounds. This approach is known as data-driven sound synthesis.

It then struck me that the idea of creating new sounds using a set sound from a specific corpus as a target should not be restricted to only Malay gamelan (which had a rather small-sized dataset to begin with), but could be extended to other sounds as well. This is especially useful as obtaining the Malay gamelan dataset whilst I was physically abroad can be a cumbersome process, as little preservation of the surviving pieces is done in the digital format. As is the case with most traditional Malay music, these pieces are disseminated non-

commercially, and when performed, they are typically played by persons who are not highly trained musical specialists, resulting in variants of the original pieces. Thus, expanding the dataset to other sounds could open up an endless possibility for sound creation. I began to experiment with several combinations of target and source sounds, some of which can be referred to in Appendix A1.

This method of sound creation had previously been used before as seen in Concatenative Sound Synthesis (CSS) or Music Mosaicing. Although the idea itself is not something new, the field itself is still in its infancy. CSS had been inspired by the art of mosaicing. Mosaics are designs and pictures formed from a process of putting bits and pieces (called tesserae or tiles) made of cubes of marbles, stones, terracotta or glass of different range of colours to create larger, whole images (Figure 1). These images are typically seen in many decorative paraphernalia and are also applied to the design of many significant cultural and spiritual erections. It is so widely dispersed in time and place that the evidence of its existence is seen across many cultures and periods, including Greco-Roman, early Christian, Byzantine, Islamic, post-Renaissance and even in contemporary art today (Dierks, 2004). Further reading on the background of mosaic can be found in the works of several notable experts such as in Bowersock (2006), Chavarría, (1999) and Ling (1998).

Through the same concept of rearranging small tiles together to produce larger pieces, more meaningful artwork, mosaicing has been applied to digital image synthesis and digital audio synthesis, and is referred to as ‘photomosaicing’ and ‘musaicing’ (musical mosaicing) respectively. In photomosaicing, small tiles of images are assembled together to compose a bigger, overall picture (Tran, 1999), as illustrated in Figure 2. Likewise, musical mosaicing assembles a large number of unrelated sound segments together according to specifications

given by an example sound to form a larger, more coherent sound framework. In any case, the creation of beautiful mosaic art is reliant upon the creativity of the artist.

Figure 1: Roman mosaic, Tripoli Museum, Libya
Source: Creative Commons License

Figure 2: Photomosaic of Mark Zuckerberg, founder of Facebook, using icons from the site
Source: Creative Commons License

In general, traditional-looking mosaic follows several basic properties (Di Blasi and Gallow, 2005):

1) each tile has a uniform colour,
2) tiles may change in size and shape, but must be within reasonable ranges and are generally convex, and
3) empty spaces between tiles should be reduced to a minimum and serves as graphical element to strengthen borders, lines and edges.

In photomosaic, images are synthesised using information such as size, shape, colour and orientation, and also discrete primitives such as pixels. As the very same concept holds true for musical mosaic, I wanted to identify the properties or factors that would affect the end product of synthesised sounds. Not only will these factors serve as a guideline by which concatenation of sound tiles can occur, but by allowing these factors or properties to be altered to suit user's specifications, user control flexibility could be enhanced. For instance, if the size of the sound segment is found to affect sound synthesis via CSS, then allowing users to set different segment sizes (500 ms or 1 sec) will enable users to generate wider range of sounds.

More importantly, I wondered if the selection or activation of these properties could somehow be automated, or at least partially-automated to assist in the process of sound creation. Automation would enable the process to be carried out more efficiently, faster and with lesser effort, without compromising the synthesis result that is closeness to target. If this was possible, I questioned if the new sounds generated automatically would resemble the target and if users agree that they are indeed, perceptually similar.

These were some of the questions that became the basis of my research. I anticipated that the inclusion of some artificially intelligent methods would be able to provide the solution to the task at hand. Although the bulk of my research has shifted slightly from the earlier idea of recomposing Malay gamelan pieces, this was the starting point that moved me towards CSS. The following section discusses the principles of CSS in more detail.

1.2 Introduction

The impact of digital technology has brought many forwarding changes in the music field, especially in the generation of sounds. The increase in processing power, storage capacity, and improved accessibility of data helped the sound collection to grow, whilst the network bandwidth and advances in audio compression technology have made the distribution and sharing of these digital files easier. Facilitated by the advancement in the field of Artificial Intelligence (AI), the possibilities to manipulate and re-create sounds are endless.

One such area of sound creation that benefited from the rise of these technological advancements is Concatenative Sound Synthesis (CSS). CSS is an art of producing new sounds from a composite of many small snippets of audio. The basic framework of a CSS system involves taking in a sound, decomposing it into smaller sound segments, analysing its spectral and other auditory content, before searching into a database of other sound segments for a matching pair. The selected segments are then concatenated together in sequence, and are then resynthesised to produce new sounds that are based on the original sound entered. Figure 3 illustrates the general mechanism of a CSS system.

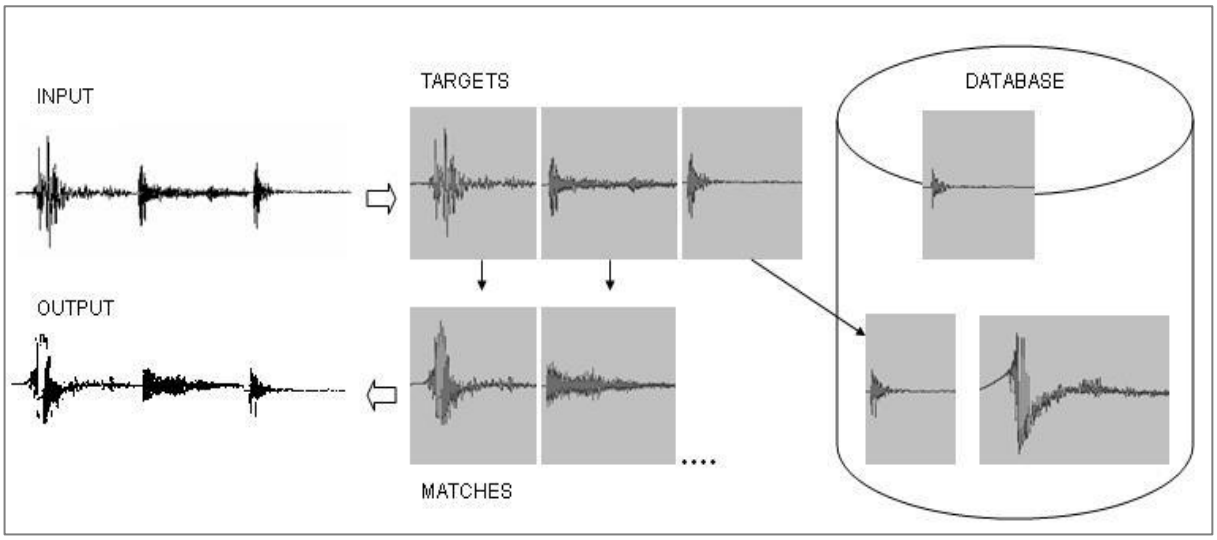


Figure 3: General mechanism of a CSS system

A typical CSS system has two major components: analysis and synthesis. During the analysis phase, both the original sounds (target) and the sounds in the database (source) are segmented into smaller sound snippets. Following segmentation, relevant information from these sound snippets is then extracted. In the synthesis phase, sound snippets in the database that match closely with the targets are selected and concatenated together forming a long string of sound, which are then synthesised (Figure 4).

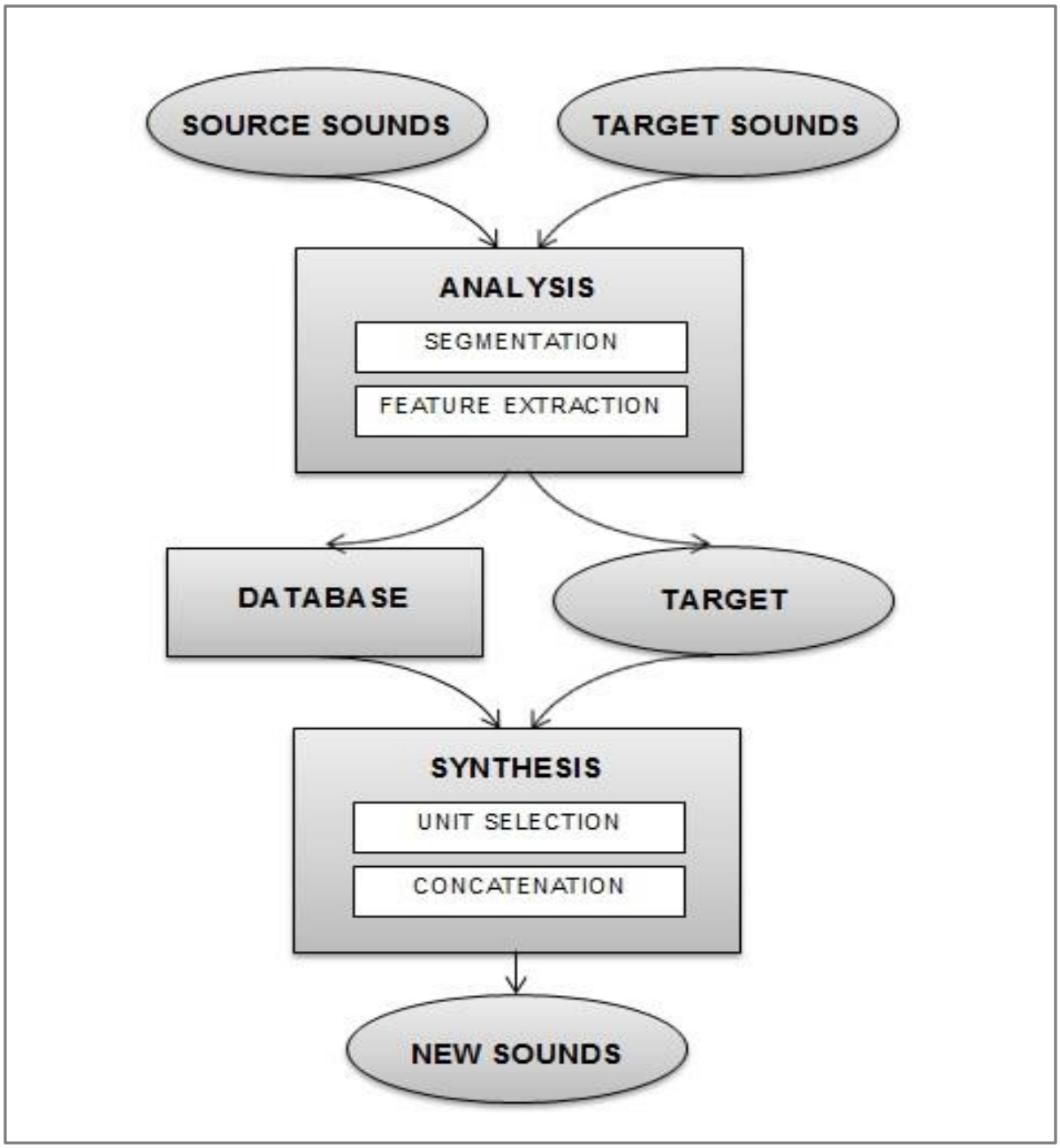


Figure 4: Data flow model of a basic CSS system

Currently, many commercial applications have made use of the technology derived from the utilisation of CSS. Its use is already commonly embedded in many communication aid devices that rely on text-to-sound synthesis or voice synthesis such as screen readers, talking watch, time announcement software and voice output communication aid. Whilst these examples prove that CSS can be a very useful technology to aid the development of many assistive devices, CSS can also be used as a creative medium (audio making tools such as *CataRT and Soundspotter*).

For instance, CSS has been used to generate soundtracks from sound libraries to suit the characters or the mood at a particular moment in a film (Cardle *et al.*, 2003); to create sound effects library for computer games (Farnell, 2007); to replace original audio recording with the sound of a different singer while keeping the same musical or phonetics structure in singing voice re-synthesis (Fonseca *et al.*, 2011); and even in motion-based sound synthesis, where sound synthesis are controlled via spatial information (Jensenius and Johnson, 2010).

Sounds in movies, computer games and graphic animations typically use pre-recorded sounds for pre-defined events that occur such as explosions, grunts and shots, resulting in the same sound to be repeated for every event occurrence. This quickly becomes monotonous, especially if similar scenes or actions occur several times over the entire course of the movie or game. CSS can be used to overcome this situation by segmenting the audio into smaller tiles and analysing the content of each tiles. Variations of the sound from the original recorded sounds can then be generated by finding sounds that match the detailed motion. Parameters that governed the audio tiles match are based on the input received from the users in real time, such as location, surface, material, height and volume. This provides more sound realism in movies and games, as unique sounds are generated for each event.

Through the same principal, singing voice resynthesis allows users to control the voice synthesiser using his or her own voice and the synthesiser will replicate the input voice based on the melody, phonetic sequences and musical performance of this voice using the set corpus in the database. Useful applications include replicating performance of deceased singers and as voice transformation tool, e.g. gender transformation (male to female), age transformation (adult to child) and number of voices transformation (solo to choir).

Although CSS has been found useful in many of the above-mentioned applications, perhaps its most popular use in the creative media still remains as a compositional aid that helps sound composers and sound designers in creating and manipulating auditory experiences. With the invention of gramophone in the late 19th Century, and then the invention of magnetic tapes not long after, it was already envisaged by several audio critics and composers of the time then that *".. perhaps the time is not far off when a composer will be able to represent through recording, music specifically composed for the gramophone*" (Battier, 2007).

Therefore when Pierre Schaeffer cut and spliced tapes together to compose his piece *Études aux Chemins de Fer* (1948) in the 1940s, it marked the beginning of an exciting possibility in sound making. Shortly after, in the early 1950s, Karlheinz Stockhausen began experimenting with the same concept and composed *Études des mille collants* (1952). Other notable pieces created using the same idea are *William's Mix* (1953) by John Cage, *Analogique A et B* (1958/1959) by Iannis Xenakis, and of more recent, *Plunderphonics* (1993) by John Oswolds, *Dedication to George Crumb* (2004) by Bob Sturm, and *Trowel and Seal* (2007) by Diemo Schwarz; to name a few.

These earlier works were the inspiration for CSS. Over the years, the interest on the subject grew steadily, and by year 2000, most of the manual processes involved in a typical CSS system have been automated and higher level of control is offered to users. A basic chronology in the advancement of CSS systems are as follows: Caterpillar (Schwarz, 2000), *Musaic* (Zils and Pachet, 2001), *MoSievius* (Lazier and Cook, 2003), *MATConcat* (Sturm, 2004), *CataRT* (Schwarz, 2005), *GrainStick* (Leslie *et al.*, 2010) and *EarGram* (Bernardes *et al.*, 2012).

Although the fundamental structures and functioning of these systems are similar, they do differ in several ways such as their segmentation approaches, feature selections and unit selection methods. Newer systems tried to improve the limitations of previous systems, some systems enabled audio segmentation to be done on-the-fly (Schwarz, 2005); others shifted the use of low-level features to using context-based and high-level descriptors (Zils and Pachet, 2001; Lazier and Cook, 2003); and several others focused their work on achieving synthesis in real-time, allowing live concerts to be performed (Casey, 2005; Schwarz, 2005).

Despite the steady show of interest and enhancements made over time in the field of CSS, there are still gaps and challenges in existing systems that could be further improved, specifically the development of a more 'intelligent' CSS system. The term 'intelligence' is defined as the ability to comprehend; to understand and to profit from experience. An Intelligent System (IS) is therefore a system that can manage data gathering which is then processed and interpreted to provide reasoned judgment to decision makers as a basis for action.

The computer's ability to perform tasks that were typically thought to require human intelligence is made possible through the advancement in AI. AI is the study of man-made computational devices which can be made to act in an intelligent manner. This field of study was first introduced in the early fifties through the work of a British mathematician, Alan Turing, in which he discussed the conditions that would qualify machines as intelligent. Subsequently, he designed the Turing test which observed if and how a machine was successful enough to imitate a human's reaction through a teletype. In short, the relationship between intelligence and AI can be summed such that intelligence comprises the mechanisms in order to perform a task, whereas AI research has discovered how to embed these mechanisms in computers so that they can perform the very same task. In the context of this study, the mechanisms that are involved during the synthesis of similar sounding segments via CSS needed to be understood so that they can be transcribed and replicated into the system to produce a more intelligent CSS system than those already available.

An intelligent CSS system is needed for several reasons. For instance, the task of synthesising sounds manually is labour-intensive, but when the process is somewhat automated, it becomes more efficient as it requires fewer resources and is completed in lesser time too. An intelligent system can also be more competent than humans, especially in tasks that can get too stressful or exhaustive such as searching the entire database for a matching sound segment. In addition, when designed and developed appropriately, an intelligent CSS system is less likely to make errors in judgment-related tasks such as determining sound similarity. Since the functions of intelligent systems are infinite, this vastly aids the creative process of music making. The ways in which AI has helped shape and improve sound synthesis will be further discussed in the next chapter.

Therefore, CSS systems can no longer remain stagnant as the simple arts tool that relies on random re-synthesis of sound segments to generate new sounds, but must become sufficient and adept at deciphering the needs and demands of composers. A system that can generate news sounds that are in line with the composers' interpretation is highly sought after. This can be achieved by extending user control in CSS systems, and through the enhancement of the AI elements in CSS systems.

1.3 Research Objectives

The objectives of this study are constructed in view of the challenges that exist in developing a framework for an intelligent CSS system. To achieve this, the following research questions are addressed:

a) *What are the factors that affect the resulting sound generated from a CSS system?*

Many existing CSS systems offer some form of user control flexibility to its users. For example, users can select different audio features to be included as the basis of similarity between target and source sounds, or be provided with options to alter the pitch or loudness, or given the flexibility to set the similarity threshold between the target and the sound segments in the source database. However, with the exception of features selection, most of these control options are offered post-unit selection, i.e. after the segments are already selected and synthesised by the system. Post-unit selection transformation often means that re-selection of the sound segments to conform to the last minute adjustments entered by users. If these criteria were made clear before the selection of sound segments takes place, it is possible that the resulting sound will match the target sound more closely. This change will not only minimise the transformation needed, but also saves time as any ambiguities can be eliminated from the start. In depth elaboration on the basic processes involved in a CSS system is described in Chapter 2 (Technical Overview of CSS, p.36). Thus, identifying the factors that affect the synthesis result and including them in the system as options that users can control are the key factors to ensuring that the demands of users are communicated to the system.

b) *Would extending some aspects of the AI implementation in a CSS system enhance user control flexibility and improve similarity result of the sounds composed?*

A CSS system with good control allows users to provide a clearer description on what needed to be searched. This provides users with the opportunity to fine tune their parameters and constraints with regards to the sounds they intend to compose. But once the information has been relayed, the backbone of the search mechanism lies heavily on the AI approaches implemented. The more recent CSS systems have already assimilated some forms of AI in their working algorithm. However, the use of AI should not be restricted to the search and selection processes only as they currently are but to further embed AI to other stages that occur in typical CSS systems. Potential extension of AI in CSS includes training the system to intelligently distinguish the sound segment that is more relevant to the target when several of the sound segments with same magnitude exist in the database, concatenating the sound smoothly from one segment to the next, and judging whether the user is more interested in the interestingness or the preciseness of the sound generated from a given target. In addition, more innovative CSS systems that encourage qualitative input from users who are assumed to possess some level of expertise in composition are needed e.g. by allowing users to assign orders and weights for the features selected (order dependent feature selection). The limitations of existing CSS systems will be investigated in this study and later, the possible solutions to overcome these limitations through the use of AI will be described.

c) *When determining whether two sounds are similar, what elements of sound play a major role in humans?*

There are several different ways that sounds are characterised such as through their melody, timbre, tempo, dynamics and rhythm. Determining which of these sound elements are more dominantly engaged by humans during the process of determining the similarity of sounds, and applying it to the CSS framework could play an important role in ensuring that the system generates sounds that are in line with the expectation of its users.

Therefore, the aim of this study is as follows:

To propose a novel framework to address the issues in existing CSS systems and to improve sound similarity of composed sounds by exploiting the AI approaches derived from the understanding of the human's sound cognitive domain.

In light of the above, the following needs to be thoroughly understood, analysed and developed:

i) Identify the parametric factors that affect synthesis results.

ii) Establish the need for an order-dependent audio feature selection process which prioritises the match between target and source segments according to the weights assigned for individual features, and propose a solution to this.

iii) Demonstrate the challenges in existing CSS systems during the unit selection process, and propose a robust new approach to counter this.

iv) Understand better the sound cognition domain, particularly the way it affects sound similarity deductions in humans, with respect to the similarity deduced between the target sound and the sound composed by the CSS system.

v) Design and propose a novel framework for CSS system that stresses the importance of inserting a ‘query’ stage in the workflow of general CSS systems.

1.4 Thesis Structure

This thesis is divided into six main chapters, including this Introduction chapter. The remaining of the thesis is organised as follows:

Chapter 2 presents the principles of CSS, beginning with the different sound synthesis approaches, which then delves into the sub-areas of CSS which covers speech, singing voice and music syntheses. The technical overview of a CSS system is also described, with focus given on each of the stages involved, i.e. audio segmentation, audio feature extraction, search algorithms applied in similarity matching of the sound units, and the similarity measurements used to determine the distance between target and source sounds.

Chapter 3 reviews the state-of-the-art of existing CSS systems, and discusses the issues that are still present in the context of the degree of analysis, unit selection level, concatenation quality and real-time capabilities. The discussion on the problems is then concentrated into a smaller scope in which this study is intended to solve. A preliminary listening test which had been conducted to discern the dominant perceptual audio elements in humans is also described, and results obtained from this initial experiment is then presented and discussed.

Chapter 4 presents the framework of this study, the 'Query-based CSS Model'. It revisits the problems that were raised in Chapter 3 and delivers the rationale for the new framework. The novel approaches proposed to overcome the earlier problems are also explained in detail here, with stronger emphasis on the parametric factors affecting CSS output, the order-dependent feature selection approach, and an original solution for the search and selection method, which are the main contributions of this study.

In order to validate the approaches mentioned in the previous chapter, series of experiments that were performed in four phases are described in **Chapter 5**, including one

listening test that compares the correlation between sound similarity and interestingness level in humans. The consistency of humans' judgment on sound similarity is also conducted and elaborated in this chapter. Results from these experiments are also analysed and discussed.

Chapter 6 concludes this thesis by highlighting the contribution to knowledge introduced in the thesis and also gives recommendations for future advancements in the field.

Additionally, this thesis also includes a number of appendices, which contain various additional information that support the body of discussion in the thesis, such as detailed results, samples of sounds and a number of peer-reviewed publications from this study.

Chapter 2: Principles of Concatenative Sound Synthesis

This chapter reviews the literature that is referred to in this thesis to give better insight to the principles of this study. It covers the arguments between two sound synthesis approaches, followed by a survey of past literature on the application of concatenative synthesis in the field of speech, singing voice and music syntheses. The technical overview of a typical CSS system is also revisited, but with further emphasis on the components involved in the process, such as audio segmentation, audio feature extraction, search algorithms and similarity measurements.

2.1 Sound Synthesis

A very broad definition of sound synthesis is given as 'the process of generating streams of audio samples by algorithmic means (Roads, 1996). Loosely, the general usage of the term refers to the process of synthesising sounds is taken as designing a sound 'from scratch'. There are many techniques that can be applied to synthesise a sound, one of which is through CSS, where sound segments that are similar to the example or target segments are searched within an audio collection using a sound matching algorithm. New sounds are synthesised by concatenating the matching segments back together. This methodology is not exclusively restricted to creating music composition, but is also applied in other tasks such as audio matching. However, the latter is more fixated towards finding (with the intention of eliminating) sound pieces in the database that are redundant or descendants of the target sound such as same piece with different artist or same piece with different arrangement.

There are many motivations behind synthesising sounds, but one of the most common reasons is to enable the emulation of existing sounds. For instance, sound synthesis allows the replication of sounds that are difficult to capture, e.g. in the case of a human performance, replacing the need of a human performer. In addition to producing usual, everyday sounds, it is also useful in producing 'new' or 'unheard' sounds. Sound synthesis is seen extensively employed by many sound designers in the production of films depicting various sci-fi or fantasy characters, particularly in scenes where unworldly growls, roars and explosions are involved. Some examples of such sounds include the sound of dinosaurs in the movie *Jurassic Park*, or the notorious sound of laser weapons (Lightsabre) blasted in the movie *Star Wars*. Moreover, sound synthesis can also mix life-like sounds and physically impossible sounds together, providing composers with endless possibilities of creating different range of sounds.

There are many ways in which sounds can be synthesised, ranging from combining basic waveforms together to formulating complex mathematical algorithms in reconstructing a sound's physical attributes. These include syntheses that are derived through spectral or Fourier-based techniques (subtractive synthesis, additive synthesis and wavetable synthesis), modulation techniques (amplitude, frequency or based modulations), wave shaping synthesis (distorting an input waveform using a transfer function), time modeling (granular synthesis, re-synthesis by fragmentation) and physical modeling (modal synthesis). A more thorough dissection of the strengths, weaknesses and suitability of each of these techniques can be found covered by several experts in the area (Pellman, 1994; Tolonen *et al*., 1998; Miranda, 1998a ;Chafe, 2001; Cook, 2002; Russ, 2012).

Despite the many different sound synthesis techniques available, the techniques above are mostly considered to be of low-level. This is because sound syntheses using these techniques

are carried out by attempting direct emulation of the intended sound, which typically involves basic analysis of the sound, followed by addition or elimination of different parameters until the replication of desired sound is achieved. Several shortfalls are seen in the synthesis via these techniques, namely that these techniques result in difficult and laborious task of configuring and re-configuring numerical input into the sound synthesis system until the synthesis of the anticipated sound is reached. The problems with these low-level approaches have been eloquently expressed by Miranda (1998b):

> A composer can set the parameters for the production of an immeasurable variety of sounds, but this task is still accomplished unnaturally by inputting streams numerical data specified manually. Even if composers knew the role played by each single parameter for synthesising a sound, it is both very difficult and tedious to ascertain which values will synthesise the sound they want to produce. Moreover, composers often need to master a sound synthesis programming language in order to communicate with the computer. Even if they master this language, the design of an instrument is not a straightforward task. In such situation, higher processes of inventive creativity and abstraction become subsidiary and time-consuming, non-musical tasks. Composers need better working environment. (p.2)

In addition to being physically demanding and time consuming, low level sound synthesis techniques do not take into account any qualitative input from composers. Miranda (1998b) further proposed that the situation can be improved by combining these sound synthesis techniques with AI techniques. This is seen achieved in approaches such as the rule-based sound synthesis and data-driven sound synthesis. Synthesis using rule-based model includes the use of a set of assertions or 'rules' that are constructed from the collective knowledge of composers, which specify the actions or solutions when certain conditions are met. Data-driven model, on the other hand, does not involve rules to create sound, but instead utilises

sound corpus to re-create sounds. Its intelligence lies in the selection algorithm which it employs to select the string of sound units that most closely matches the input specifications. These two approaches are explained further in the following sub-sections.

2.1.1 Rule-based Model

Any system or technique that is rule-based uses human expert knowledge to find a solution to the real world problems that would normally require human intelligence to solve (Abraham, 2005). It does so by capturing the knowledge of an expert in a specialised domain, and exploiting that knowledge to devise series of IF-THEN rules, which are useful in making deductions or choices.

Before being applied to the field of music making, rule-based model has long been used in other areas of AI, for instance in the field of natural language processing (NLP) and expert decision systems (EDS). In NLP, rule-based systems perform lexical analysis to compile or interpret computer programmes, or to clear disambiguation of prepositional phrases based on the different contextual cues (Brill, 1992). The diagram in Figure 5 illustrates the use of rule-based model in NLP.

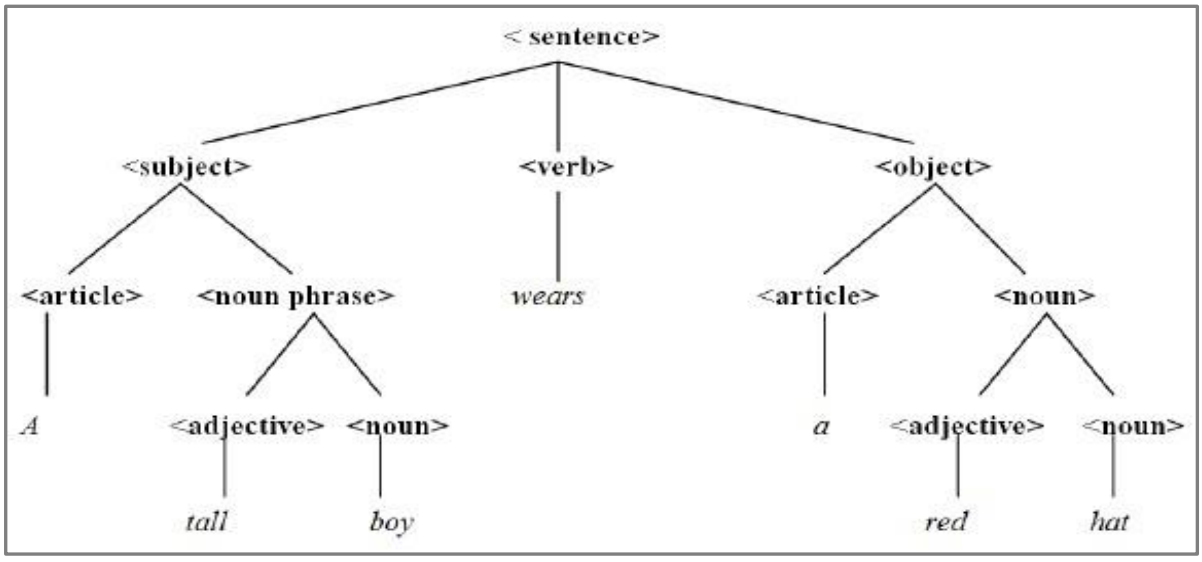


Figure 5: Rule-based model used to parse a sentence in NLP

Similarly, EDS uses rules that are derived from relevant knowledge and relationships obtained from human experts, but rather than using the rules as the blueprint to constructing statements that are syntactically correct, it links certain conditions to specific outcomes, as demonstrated in Figure 6. This model is particularly useful in diagnostic and risk assessment tasks.

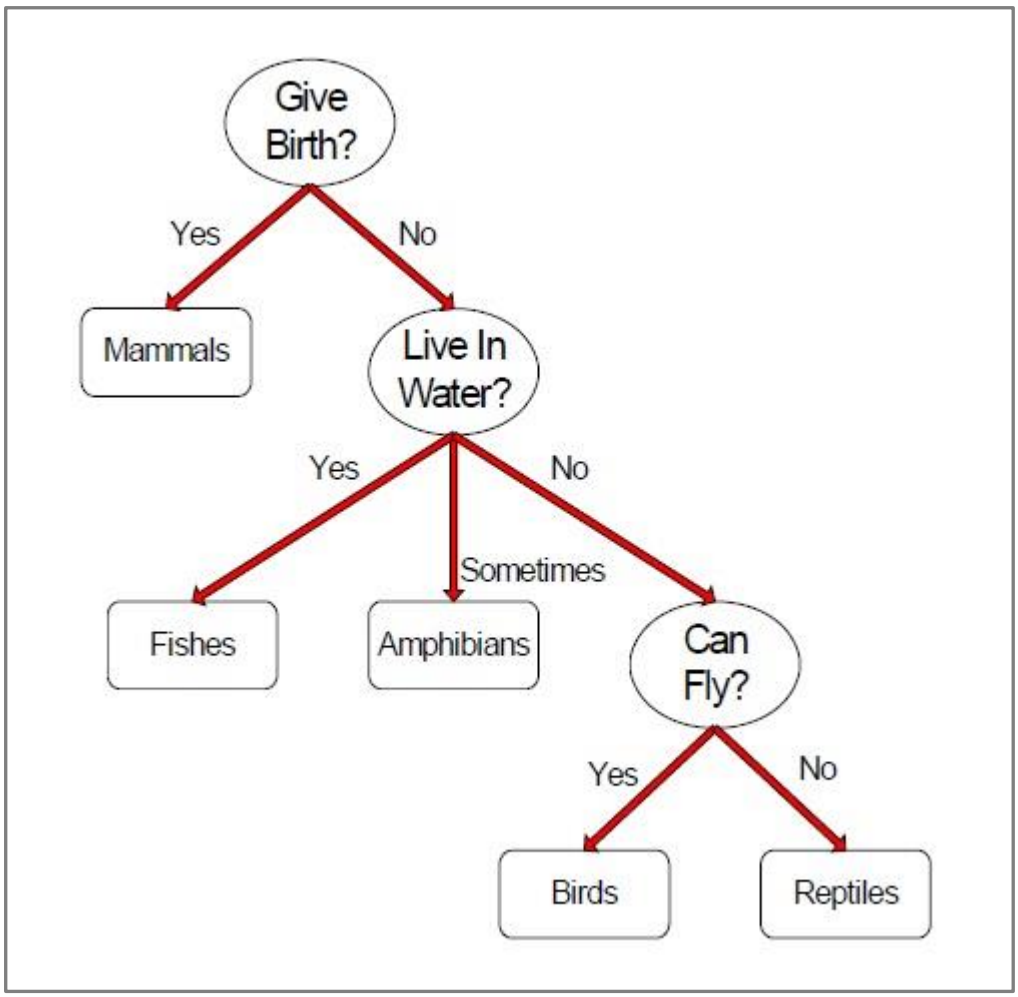


Figure 6: Use of rule-based model in EDS to classify animals into classes

The idea for a rule-based system transpired in parallel with the budding field of AI research circa the fifties, but early efforts were found to be too ambitious, owing to the fact that the scale of the problems was too large and difficult to tackle at the time. It was not until a decade later when researchers began to concentrate on smaller, more specific problems that rule-based proved to be a more sound AI approach. This was exhibited in successful

earlier projects such as DENDRAL, MYCIN and PROSPECTOR, which are all rule-based expert systems that performed chemical analysis, infectious blood diseases diagnosis and mineral exploration respectively (Negnevitsky, 2005). The rule-based approach continues to be in use in the present times, as can be evidently observed in modern systems such as the NHS Direct Adviser system and many online assistance systems.

The rule-based model later found its way into the field of algorithmic composition. As was the case with the systems from other fields, a sequence or set of rules for solving a particular task is set, by which the compositional process must behave once it is put into motion. In this case, the task directly refers to the act of combining musical parts into a whole composition (Papadopoulus and Wiggins, 1999).

This simple notion of embedding musicological rules into computational procedures to produce music has been adopted by many. One of the systems known to have an almost complete rule of harmonisation covered is developed through the work of Kemal Ebcioğlu in his programme CHORAL, which could generate four-part chorales in the style of J S Bach, using over 350 rules that he had designed (Ebcioğlu, 1984). In the same year, William Schottstaedt developed an automatic species counterpoint programme, which engaged over 75 IF-ELSE rules and a series of penalties assigned for every occurrence of a rule break during composition. Several more works have advanced since then, among them is one that includes the construction of grammars for the generation of jazz chord progression (Johnson-Laird, 1991), construction of grammar-based music composition using L-systems; a method formerly used for the modeling of curves, biological systems and morphogenesis (McCormack, 1996), the use of combinatorial rules to deduce a sequence in elements of surprise or unexpected jazz harmonic progression (Pachet, 1999) and the use of probabilistic grammars to automatically generate convincing jazz melodies (Keller and Morrison, 2007). A

more complete view on the grammars in music can be found in the intensive discussion by Roads and Wieneke (1979).

As the years progressed, the basic rule-based sound synthesis systems later evolved into a more advanced compositional system, incorporating complex methods such as stochastic approaches, neural network, genetic algorithm and other models such as fractals, cellular automata and swarm. The motivation towards this is the prospect of encouraging computational creativity. Materials on these subjects, in the order that they appear above, can be found more intensively discussed by Blackwell (2003), Chapel (2003), Jones (1981), Miranda (1995), and Todd and Loy (1991).

Although a moderate-sized rule-based model can be easily developed, the main drawback of using this approach is that it demands heavy cost of authoring and maintaining the rule sets. Furthermore, there may be brittleness in the rules. This is a situation where some conditions had not been covered when the rule sets were first designed causing some loop holes in the system, or in situation where one of the rules antecedents are absent causing a breakdown in the rule. Consider the situation where a machine will only release its valve under two conditions: the temperatures are cool for both air coming from the engine, and air moving to the engine. If one of the temperature sensors that read the temperature is faulty, then the sensor will read the temperature as ‘False’ (hot), thus disabling the release of the valve, causing the system breakdown at the face of sensor failure. This is an example of the brittleness in rule-based model. In addition to heavy maintenance cost and brittleness, rule-based model is also computationally expensive during synthesis, as there are many complex calculations involved. The data-driven model is therefore proposed to overcome these challenges.

2.1.2 Data-driven Model

The term data-driven implies that the flow of a system is determined by specific factors via external data. It is based on the analysis of the data about a system, in particular finding connections between the system state variables without explicit knowledge of the physical behaviour of the system (Solomatine *et al.*, 2008). With regards to data-driven sound synthesis, Diemo Schwarz described the model as "*synthesising sounds through the rules that are induced from the data itself, as opposed to the rule based model which supplies the rules which have been constructed through careful thinking*" (Schwarz, 2000).

Data-driven model is not only restricted to sound synthesis, but applies to many other systems that are critically dependent on data to work. For instance, all systems that depend on the ability to store, acquire and present vast amount of information such as search engines, are based on this model. The basic mechanism of such model is presented in Figure 7 below.

In data-driven sound synthesis, new sounds are created by segmenting the sounds into smaller sound snippets and rearranging the sounds based on certain parameters of existing sounds that have been modified; a process also known as re-synthesis.

Figure 7: The mechanism of a data-driven model

Unlike the previously described rule-based approach, data-driven approach does not involve complex calculations in synthesising its output. It performs computations in an order; it is dictated by data dependencies which suggest that the rules are induced from the data itself. In the case of CSS, the target sounds are primary source of information by which the rules are deduced from (Schwarz, 2003).

The obvious advantage of using the data-driven model in the synthesis of music is that it preserves the fine details of the sound. This is because the output is generated using actual recordings, as opposed to generating a synthesised sound from scratch using a model. The use of actual sounds also means that it is easier to materialise sounds that have been envisaged in the minds of composers, a feat that is otherwise extremely difficult to perform with the rule-based approach. The only down side to this approach is that it may require a larger storage space compared to the rule-based synthesis. Nevertheless, it is an ideal solution when naturalness is a priority and space is not an issue. In general, the larger the size of the database, the more likely an exact matching sound is to be found, hence greatly reducing the need to apply transformation on the sounds from a data-driven CSS system.

Further use and applications of data-driven CSS systems, along with examples of sound synthesis systems developed based on the data-driven model are described in the next section.

2.2 Sub-areas of Data-driven Concatenative Sound Synthesis

This section discusses the applications of data-driven CSS in the following sub-areas of sound syntheses: (1) speech, (2) singing voice and (3) music.

2.2.1 Concatenative Speech Synthesis

Between the three sub-areas of CSS, the research that has been carried out on speech synthesis appears to be the most prominent. Such is expected, as the advancement in speech synthesis is roughly ten years ahead of other forms of sound syntheses (Schwarz, 2006). It is therefore, unsurprising that many approaches in other forms of sound syntheses are heavily inspired and influenced by the methods applied in speech synthesis. Based on this fact, it is worth reviewing the general area of concatenative speech synthesis before delving specifically into other areas of concatenative sound syntheses.

Speech synthesis is an artificial production of human speech. It can be created in two ways, as previously presented, synthetically via a synthesiser to model the human vocal tract (rule-based model), or concatenatively (data-driven model). Regardless of which method is adopted, a good speech synthesis system should be able to conventionally display high intelligibility and naturalness in the sounds generated. 'Intelligibility' refers to the system's proficiency in understanding the language, i.e. how relevant is the answer synthesised with respect to the context of things, for example is the word 'lead' synthesised as (l̄ d) - an act of showing the way by going in advance; or (l̆d) – a dense metallic element. Equally important is naturalness, which refers to the human-like quality of the speech, i.e. how closely the speech sounded like humans, as opposed to sounding robot-like. In short, an ideal concatenative speech synthesis system should be able to deliver comprehensible sentences (intelligent), through a human-like voice (naturalness), to its audience (Schwarz, 2006).

As its name suggests, concatenative speech synthesis systems generate speech from actual recordings of human speaker. Speech synthesised this way is more natural-sounding than that generated from rule-based synthesis systems. However, the speech may contain some glitches and distortions in the output due to the automatic segmentation and waveform techniques that are applied in the process. Even so, it remains a popular synthesis approach of choice, as the use of original recordings retain the quality of sounds better (Hunt and Black, 1996).

Concatenative speech synthesis can be further divided into three different sub-types: diphone synthesis, unit selection synthesis and domain-specific synthesis. In diphone synthesis, the segments can be concatenated at the diphone unit only. To simplify matters, human speech recordings are usually carried out in a monotonous pitch. During synthesis, the diphones are concatenated together and sound is generated through signal processing techniques. The advantage to this approach is that it is smaller in size, but suffers from sonic glitches during concatenation and can sometimes appear to sound more 'robotic', owing to the signal processing techniques applied prior to synthesising the sound.

In contrast, unit selection synthesis does not limit segmentation of recorded words in the database by diphones only, but can include many different unit sizes such as phones, diphones, half-phones, syllables, words, or even as large as whole sentences. Although unit selection synthesis gives greater size flexibility, it also means a much larger database of sounds at varying unit sizes is needed in order for it to work. The basic sound information of each of these units is analysed, e.g. pitch, duration, and neighbouring phones. . During run-time, concatenation of several unit sizes are created, and through specially weighted tree, the best chain is selected and synthesised (Figure 8). If well-matching units are found in the database, and no signal processing is necessary, the results are much more natural-sounding

speech compared to those produced via diphones synthesis alone. However, when there are no appropriate units found, the concatenation results can be very bad (Schroder, 2001).

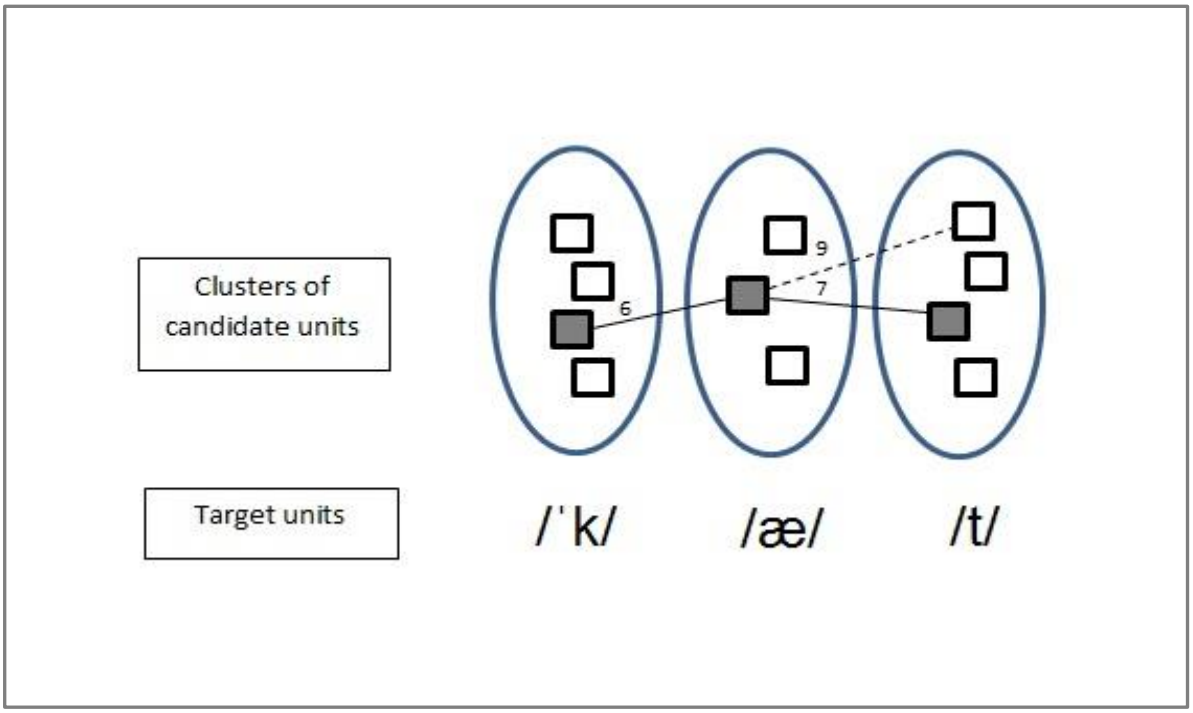


Figure 8: The word 'cat' is synthesised by concatenating relevant phonemes in the database. In the case where multiple phoneme units are present, they are clustered together and using the weighted decision tree, the best chain based with the least concatenation cost is selected (compare the solid lines and dotted line in this example)

In a similar fashion as the unit-selection synthesis approach works, domain-specific synthesis approach uses pre-recorded whole words or phrases into complete utterances, but its uses are highly limited to one particular domain, for example the weather, sports and time announcements. Hence, it is the simplest form of concatenative speech synthesis, yet sounds very natural. Since its collections of words are very contained, it only requires a very small database. Domain-specific synthesis is already widely and successfully used for many commercial applications, among them are talking watches, talking calculators, Public Address (PA) announcements, automatic ticketing and queue calling system.

One of the earlier concatenative speech systems developed was the ATR v-Talk, which was the research product of ATR. It embraces the very basic unit selection algorithm where units with the least acoustic distance measured between the target and sounds in the database were selected for concatenation (Sagisaka, 1992). CHATR then added the prosodic features like duration and intonation to target specification to allow the system to choose more appropriate units in terms prosody (Hunt and Black, 1996). Following CHATR, Next-Gen further improved the existing unit selection algorithm by allowing units to be compared on half-phone basis (Syrdal *et al.*, 2000). The IBM Trainable Speech Synthesis System advanced further and used decision trees to decide on the appropriate unit sizes (Donovan *et al.*, 2001).

Development of concatenative speech synthesis systems is not limited to the English language only, but are equally well-researched in other languages of the world including Japanese (Sagisaka, 1992), Hindi (Kishore and Black, 2003), Turkish (Sak, 2004) and Mongolian (Davaatsagaan and Paliwal, 2008).

2.2.2 Concatenative Singing Voice Synthesis

Concatenative singing voice synthesis refers to the production of human-like singing voice which is produced by a computer. It is a mixture of both speech synthesis and music synthesis, although its methods are more closely related to the former. On top of the intelligibility and naturalness factors, singing voice synthesis must also consider properties such as vocal aesthetics and music quality (Rodet, 2002). An example of its greatest application is creating voices which humans are unable to do, for example the castrato voice in the film *Farinelli* (1994) by Gérard Corbiaud.

Like all other forms of concatenative sound syntheses, singing voice synthesis stores short speech units in its inventory and units with the smallest distance from the target units are selected to be concatenated. The units are then modified in duration, melody or other properties such as vibrato, timbre, pitch, or energy of the sounds to achieve prosody of natural utterances and to ensure a smooth concatenation result. This is typically performed through signal processing techniques such as PSOLA. Since the inventories need to be very large and constructed from specifically recorded sounds that need to be mostly indexed, it is no surprise that there are only very few of such database available, one of which is the Lyricos system (Macon *et al.*, 1997). Further reading on the different synthesis methods, control strategies and learning techniques that are uniquely related to singing voice synthesis can be found in an intensive review by Xavier Rodet (Rodet, 2002).

2.2.3 Concatenative Synthesis for Music

Sound synthesis approach that specifically focuses on music production started to appear around the forties onwards, where experimental artists such as Pierre Schaeffer, Karlheinz Stockhausen, John Cage and others began recomposing sounds by cutting and pasting segments of sounds that could be played from ordinary gramophones and tapes, to produce interesting new sounds.

The process that was first conducted manually became available digitally in the seventies. This became apparent in the use of the digital sampler player or simply referred as the 'sampler'; a tool resembling synthesisers that can generate new sounds through imitation and manipulation of existing sounds. However, unlike synthesisers, samplers use recordings of sampled sounds that are loaded onto the machine by users and then played back by a keyboard sequence or other controlling devices to create music, rather than through sound synthesis methods.

Following the widespread use of computers in the nineties, it became easier to perform digital sampling as it required nothing more than highlighting a section of already-recorded music and clicking the 'duplicate' icon to create loops on a personal computer. This form of sampling is termed as 'phrase sampling' and is still extensively used in the production of hip-hop and R&B music. The history of digital sampling can be found in the literature written by authors Julius O Smith, Hugh Davies and Henry Self (Smith, 1991;Davies, 1996; Self, 2001).

Another form of music synthesis that is based on the same idea of cutting musical sounds into smaller pieces and rearranging them again is called 'granular synthesis'. Granular synthesis is defined as the process of combining basic grains of sounds to form larger sound events (Miranda, 1995). Granular synthesis has very short durational units that are micro in size, ranging anything from 10 – 100 milliseconds long. Many well-known composers have composed many interesting pieces through granular synthesis, including Curtis Road's *Klang-1* (1974), Barry Truax's *The Wing of Nike* (1987) and Eduardo Miranda's *Olivine Trees* (1994). These example pieces are the first to have been implemented using granular synthesis digitally, in real-time, and by means of cellular automata.

Although both granular synthesis and CSS involve reassembling small sound segments to compose larger musical pieces, there are several differences that set them apart. For instance, the segment size for granular synthesis is typically very small and of uniform length, whereas in CSS, the segments are longer and can have varying lengths, especially if an event-based segmentation is used. The concatenation rules in granular synthesis are also generally more flexible, as synthesis happens in a more unrestricted manner (free synthesis). This means that new sounds can be generated either by sampling a portion of the sound and replicating it many times; or by selectively sampling in different parts of sounds in the same source and concatenating them back together. In comparison, CSS only allows segments that

have satisfied the features or descriptors set based on the target sample provided to be synthesised.

As with the previously discussed concatenative speech synthesis, the basic principles of concatenative synthesis for music are fairly similar, for example new sounds are produced from the re-synthesis of an original sound. However, there are several characteristics that set speech and music syntheses apart. One such attribute is phonemes. In concatenative speech synthesis, phonemes are the basis unit for segmentation, whereas for music, units are usually segmented according to musical notes or events. The second attribute that is time, is crucial for music synthesis as time is needed to ensure that the rhythm is in place, but has very little effect with speech. Finally, as concatenative synthesis for music is more artistically-perceived, in general it allows more space for creation, as it does not need to take into account the intelligibility or naturalness of utterance as concatenative speech synthesis. Nor does it require the high syntax-semantics quality as expected in concatenative speech synthesis, in order for it to be understood by its audience. An in-depth review on the state-of-the-art concatenative sound synthesis systems for music is covered in Chapter 3.

2.3 Technical Overview of Concatenative Sound Synthesis

Previously in Chapter 1, the model of a basic CSS system was presented and it was briefly described to have been made up of several components, i.e. database, target unit and source unit (refer to Figure 4). These components and the technical overview of CSS systems are further discussed below.

2.3.1 Database

The database of a CSS system stores a collection of audio files, or is also called the 'corpus' that will be used in the generation of new sounds. In addition to storing the actual audio files, it can also save the source files, references, unit descriptors and the relationships between all entities in it. The actual synthesis of the sound is also generated from the database.

Up until the early nineties, the majority of the corpus in the database was kept in an analogue format. However, this has changed and most data are now accessible on digital media. There are a number of issues surrounding digital signals such as the size of the data, its resolution and legal procurements of data. Fortunately, there are several large audio databases that have been made accessible to the public, allowing computer music-related research to be carried out, such as the Free Music Archive (FMA)[1], Creative Commons Mixter (CCMixter)[2] and Magnatune[3].

There are many other sites that fit the same purpose but the selection of musical databases is usually influenced by users' preferences of musical genre and language, the size of the audio collection, the format of the audio (wav, aiff, mp3), the length of the song (whole

[1] http://freemusicarchive.org
[2] http://ccmixter.org
[3] http://magnatune.com

length, 5-seconds long) and the costs involved in obtaining the material. For example, the choice of genre may be based on the user's intended sound output, e.g. it may be more of an obvious choice for a user to include sounds of the classical genre as opposed to genres such as pop or rock, if the intended piece needs to sound like it is composed with a lot of string instruments in it (though sometimes interesting results can happen with corpus that are not so obvious). Likewise, the length of the segments depend on whether the user intends to compose more granular-like sounds (very small segment length), or to imitate the melody of the target sound where the segments need to be much larger in order to have enough melodic information to be captured. Also, the fee charged by some sound archives is another factor that affects composers' choice of sound to be included in the database.

2.3.2 Target Unit

The target unit is the piece of audio that is supplied as an input into a CSS system, so that a matching unit can be found from the database and played back concatenatively as the output. The target unit can be supplied to the system in several ways, but the most typical form is by submitting an 'exemplary' sound file into the system that the system can imitate by searching the nearest sounding sound segments in the database and synthesising them. Other methods include providing a short piece of sound to the system by humming through a microphone, or via a MIDI keyboard or guitar in place of the sound file. Some systems such as *Audio Analogies* (Simon *et al.*, 2005) accept symbolic information such as the MIDI score that can be fed directly into the system.

2.3.3 Segmentation

Before any processing can take place, the audio files in the database must first be segmented into smaller sound units. This takes place by marking audio streams at its boundaries, which can happen through automatic alignment of the musical score, spectral

change or arbitrary segmentation. However, segmentation can be classified into two basic categories: time-based or event-based.

Time-based segmentation is performed by segmenting a sound stream at an evenly spaced time interval, for example for every 500 milliseconds, resulting in homogeneous units of sound. This method generally takes no consideration of the musical activity that goes on, and is the most straightforward form of segmentation, as there is no complex detection methods involved. Despite its simplicity, it is the most useful choice of segmentation mode if all the sound units need to be of uniform length. Basic sound editing tools such as Audacity[4] and Garage Band[5] are perfectly adequate to perform time-based segmentation.

In contrast, event-based segmentation produces heterogeneous (non-uniform) units of sound. This is because segmentation takes place when a characteristic change in the audio stream is detected, e.g. the entrance of a guitar solo or a change from spoken words to music. One of the ways to perform event-based segmentation is by separating the musical signals at the boundaries of audio objects, i.e. where the note starts (onset) or where it finishes (offset). Onset and offset segmentation is particularly useful for the modeling of attacks, as it helps localising the beginning of a note (Brossier, 2006). It is therefore unsurprising that the onset detection method has been employed in segmentation for many different applications such as music classification, characterisation of rhythmic pattern and tempo tracking, for example.

The onset of a signal is described as the 'perceived beginning of a discrete event, determined by a noticeable increase in intensity of by a sudden change of pitch or timbre' (Brossier *et al.*, 2006). Onset can be further divided into two types, percussive and tonal. Generally,

[4]http://audacity.sourceforge.net/
[5]http://apple-garageband.en.softonic.com/

percussive onsets detect sharp attacks and sudden increase in energy, and are more suitable to segment audio pieces that inherit these characteristics such as drums. On the other hand, tonal onsets are good in detecting more subtle changes or smooth transitions, and are better suited in segmenting pieces with singing voice or string instruments. It is therefore justified to come to a conclusion that for a corpus of sound that is broad in nature, a robust segmentation system should be able to perform both types of onsets to ensure the best result possible. This has been demonstrated in Paul Brossier's work on temporal segmentation (Brossier, 2006), where he ran a test on five onset detection functions – High Frequency Content (HFC), Kullback-Liebler Distance (KL), Spectral Differences (SD), Phase Deviation (PD) and Complex-Domain Distance (CD). Towards the end of his experiment, he found that KL worked best for highly percussive music, whilst for harmonic music, SD seemed to be a more fitting option. Brossier's findings are also supported by other researchers of the same field, where each onset detection functions are better equipped to serve different purpose (FitzGerald, 2010; Stowell and Plumbley, 2010).

2.3.4 Audio Feature Extraction

Each of the segmented sound unit has unique characteristics that can be extracted from the segment itself. These descriptors are sometimes interchangeably referred as features, and can be generated from either the audio signal, their spectral, acoustical, perceptual, instrumental or harmonic properties, or symbolic score (Schwarz, 2006). There are many different features that can be extracted, and they may be extracted based on their acoustical properties are such as pitch, loudness, energy and formants. For example, a task that requires a system to classify whether a sound is grouped under 'speech' or 'music' might make use of the loudness feature. Since music tends to have higher energy than speech does, all sounds with energy under a certain threshold can be classified as 'speech'.

Features are normally extracted automatically in a process known as audio feature extraction – a process of computing a compact numerical representation that can be used to characterise a segment of audio (Tzanetakis and Cook, 2002). Usually, the use of one feature is not enough for any unique deductions to be made about a sound; therefore it is common that several features are combined into feature vectors. Feature vectors list all features for a single point in time. Figure 9 depicts a *d*-dimensional feature vector from the combination of d features. The *d*-dimensional space defined by the feature vector is also known as the 'feature space' and the floating points in the feature space are sound characteristics.

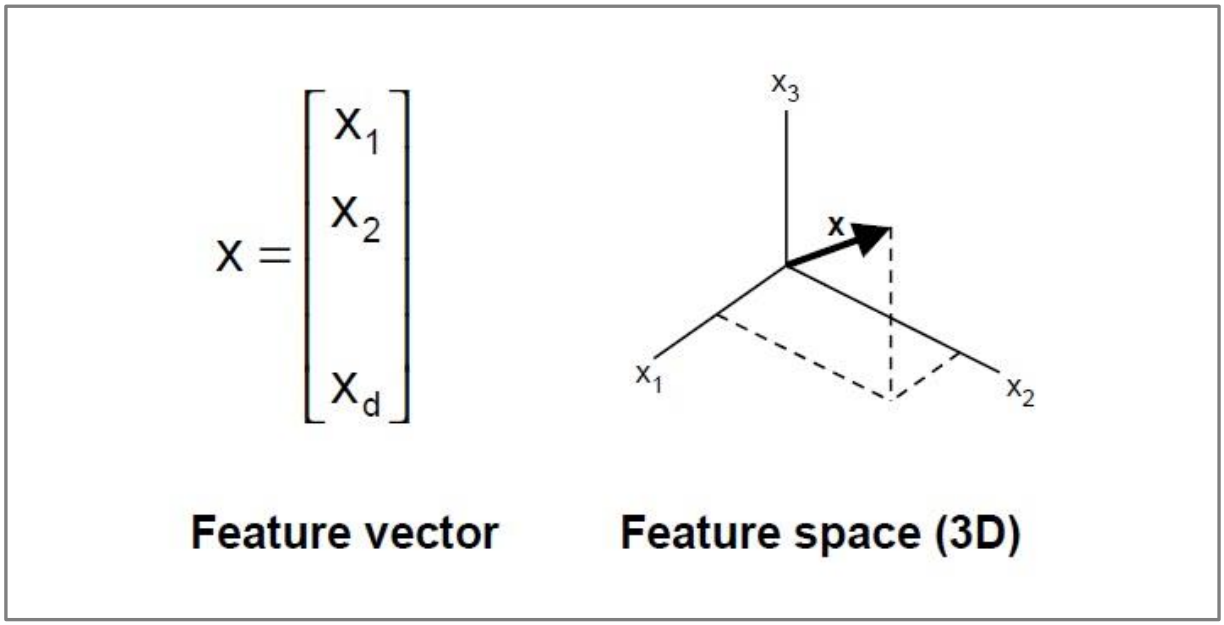


Figure 9: Feature vector and corresponding feature space

It is worth mentioning that in some cases, when several features are extracted together, a step called normalisation of the feature vector is required. Normalising a vector is done by dividing a norm of the vector, for example to make Euclidean length of the vector equal to one. It is often referred as scaling by a minimum and range of the vector, to make all elements lie between 0 and 1. This is similar to the process of converting a data that

contains a mixture of Fahrenheit, Kelvin and Celsius units to a standardised Celsius unit to ensure that the values used in all calculations are standard.

There are two basic approaches to calculate the feature vector that represents a sound; (1) trajectory approach and (2) single feature vector approach (Tzanetakis and Cook, 2002). In the first approach, the audio file is broken into fixed, small segments in time called analysis windows (20 – 40 milliseconds long) and a feature vector is computed for each window, resulting in a time series of feature vectors that can be seen as trajectory points across the feature space (Figure 10). This approach is most useful when information from the sound needs to be updated in real time, such as during a live audio streaming and interactive human-computer performance.

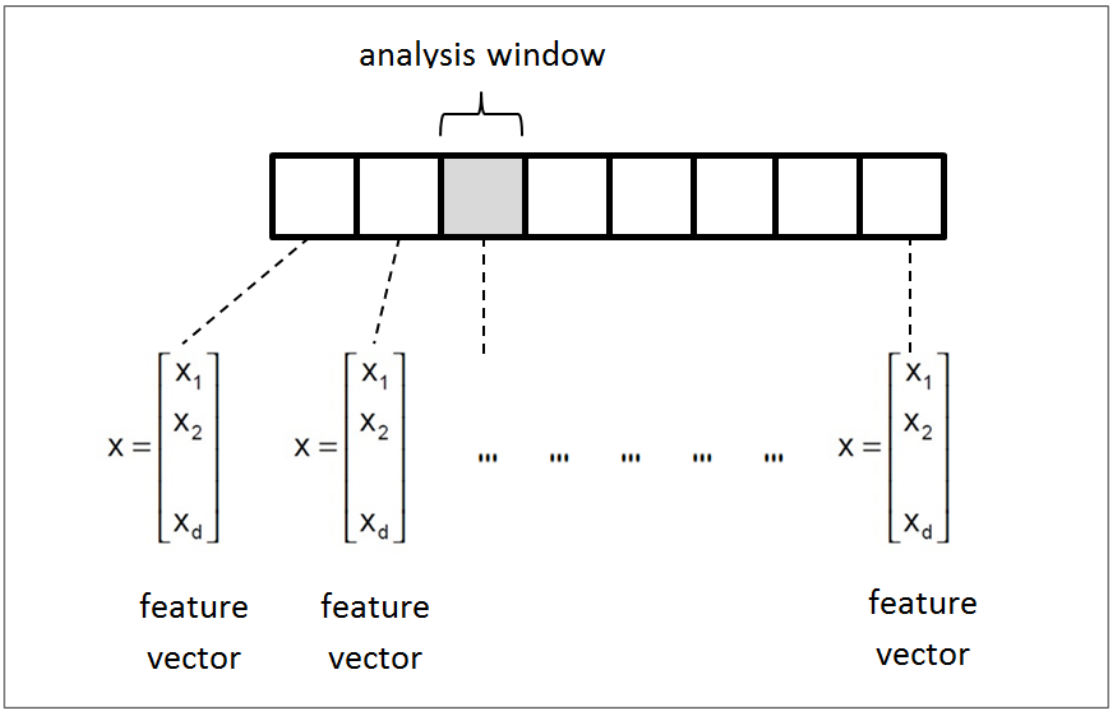


Figure 10: Trajectory approach

On the other hand, single feature vector approach summarises all the information on the audio file into one single feature vector (Figure 11). The reduction in the information analysed reduces the time and computational load to process the audio, and this approach is better suited when a gist or signature of the sound is required.

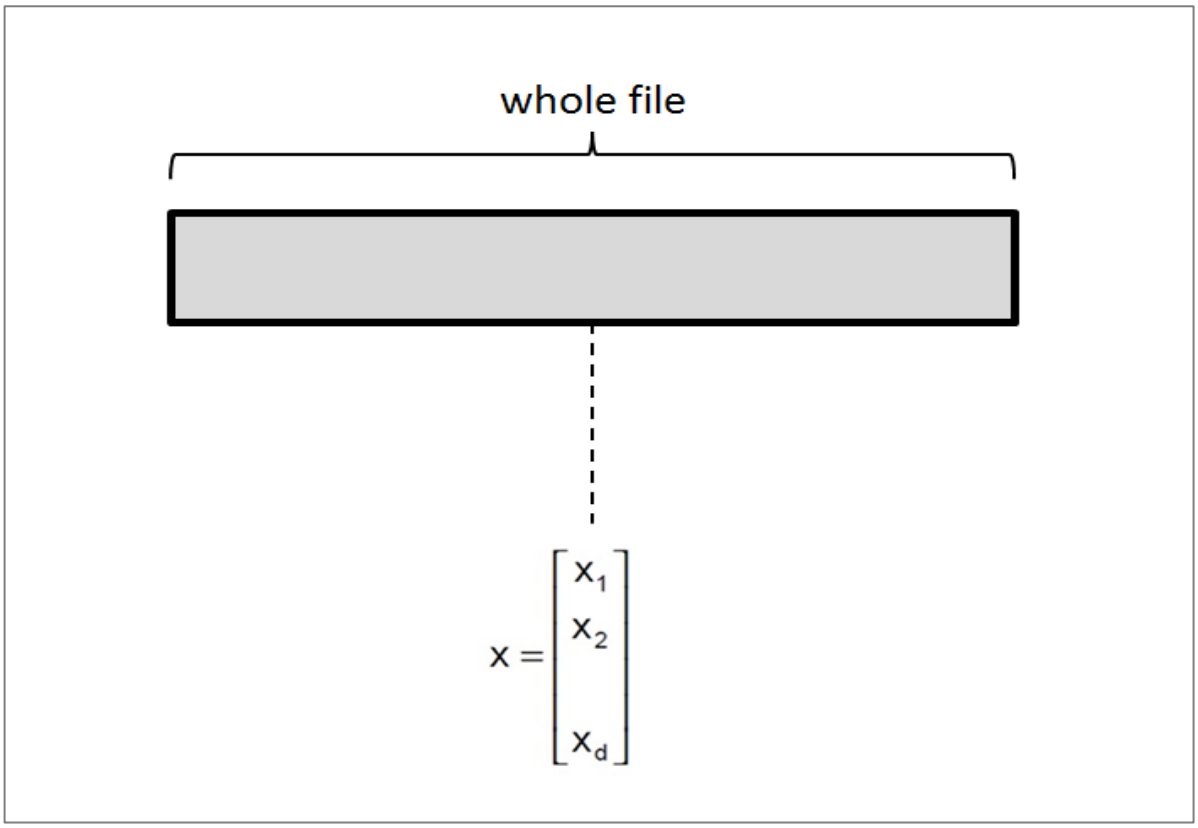


Figure 11: Single feature vector

All digital sound files are made up of audio signals that contain information on those particular files. When sound characteristics are obtained from the sound signal, the information is said to be derived from low-level audio features, which is the rawest information that a sound can contain. At the lowest level, audio signals can correspond to several different domains such as time domain, frequency domain, and time-frequency domain, which all give off different features.

The most basic of the three domains is the time domain. It represents the audio signal as amplitudes against time and can also show the sign changes that happen within the signal with respect to time (Figure 12). Examples of audio features derived from this domain are the Root Mean Square Amplitude (RMS) and the Zero Crossing Rate (ZCR). The former feature returns the average of various frequencies of the bandwidth being used, whilst the latter calculates the rate of sign changes along the signal, i.e. the number of times that a sign changes from positive to negative and vice versa. Both features can be used in speech/music classification (Saunders, 1996; El-Maleh *et al.*, 2000; Panagiotakis and Tziritas, 2005), as speech generally has lower average energy and higher zero crossing rate than music, in account of the pauses in conversation. In addition, ZCR is also used in classification of voiced or unvoiced speech, where a higher ZCR rate between two signals suggests unvoiced speech as unvoiced signals oscillate faster along the time axis.

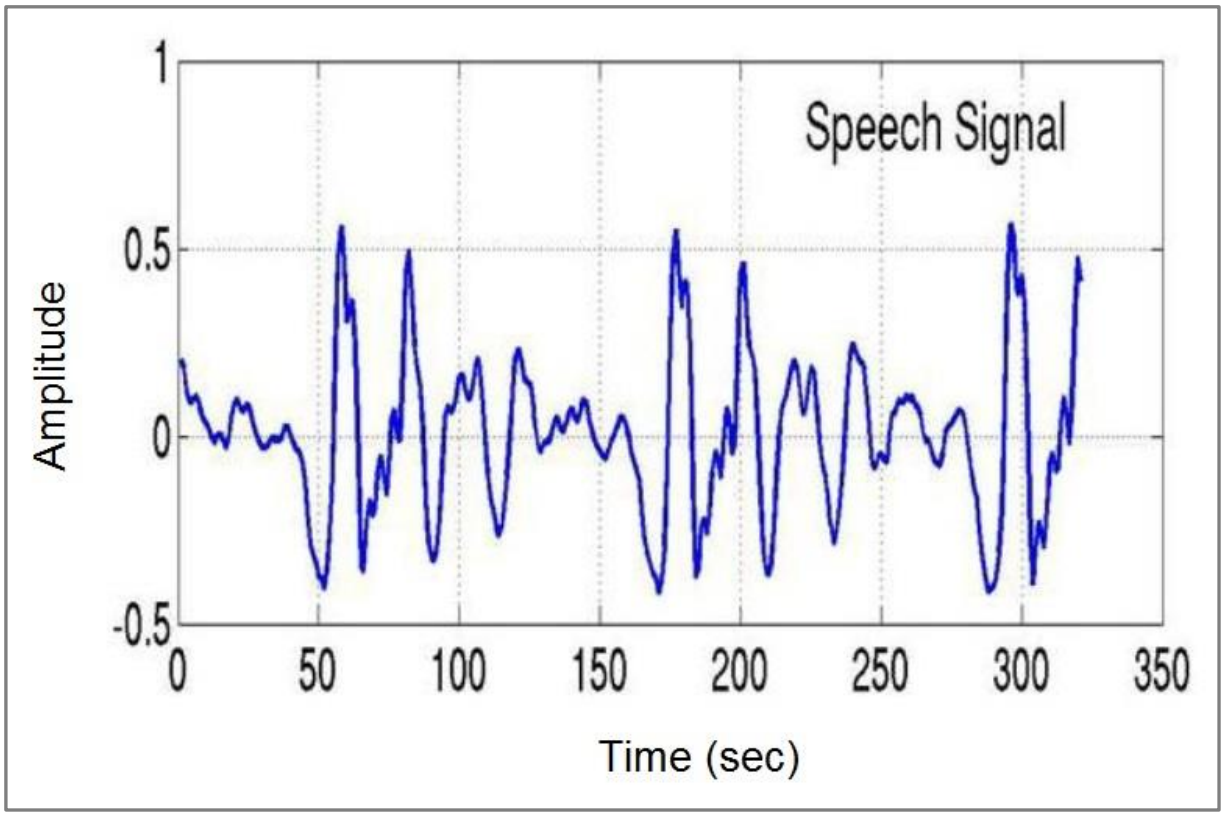


Figure 12: Time domain representation
Source: Creative Commons License

The frequency domain shows the frequency components and frequency distribution of a signal, attributes that cannot be shown through time-domain representation alone (Figure 13). One way to obtain the frequency information from the time-domain signals is through Fourier Transform (FT), a process that decomposes any signal into its frequency components. Harmonicity is an example of audio feature generated from the frequency domain. It distinguishes periodic signals (harmonic sounds) and non-periodic signals (in harmonic sounds and noise) by determining if the frequencies of dominant components are of multiples of the fundamental frequency (Mitrovic *et al.*, 2010). Frequency peaks indicate that the audio signal may be music, whereas random frequency peaks may suggest that the sound is noise or speech. Again, this feature would prove to be useful in speech or music classification, or instruments classification, e.g. between violins (instruments with high harmonicity) and drums (percussive instruments with little or no harmonicity).

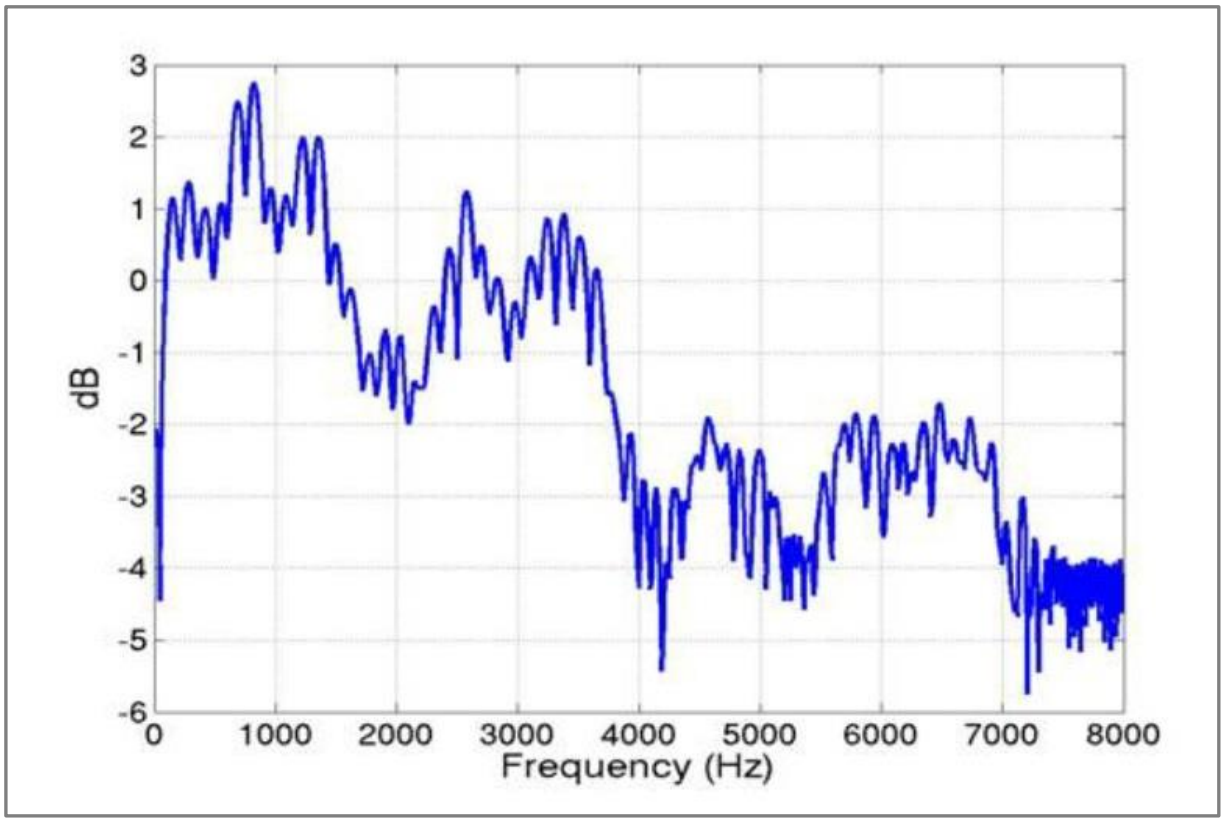


Figure 13: Frequency domain representation
Source: Creative Commons License

Although the frequency domain features are more useful than the basic features extracted from the time domain, the resulting features only reveal the occurrences for each of the frequency that exists, but lose out on the time information as to when these frequencies happen. As audio signals are non-stationary, there can be times when both frequency and time information are needed simultaneously, which can be solved through the conversion to time-frequency domain (Figure 14).

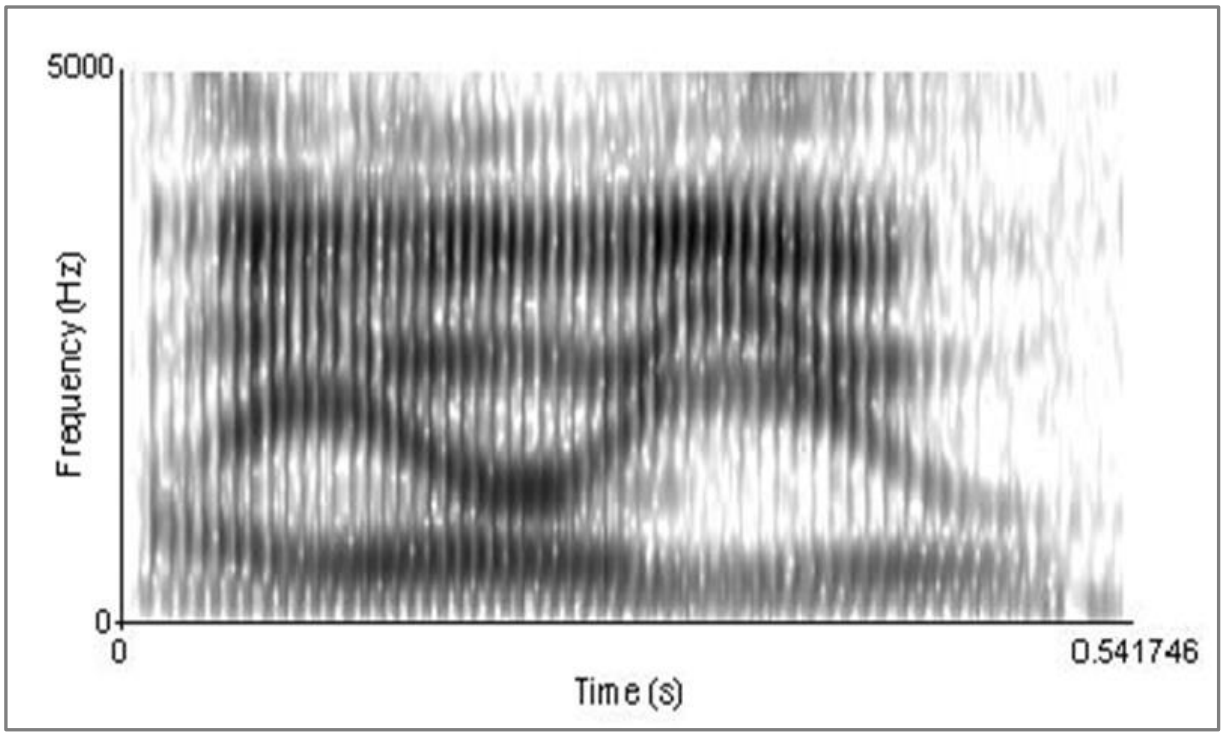


Figure 14: Time-Frequency domain representation
Source: Creative Commons License

This representation can be obtained by transforming the frequency domain signals through Short-Time Fourier Transform (STFT). STFT is a powerful general-purpose tool for audio signal processing which specify complex amplitude versus time and frequency for any signal (Allen and Rabiner, 1977). STFT works by dividing the signals into small portions so that FT can then be applied to each of the small portions. Smaller portions are achieved by changing the width of the window function and each window is shifted and multiplied with signal. The

use of STFT enables many spectral-based features to be extracted. Spectral features are particularly robust, hence widely used in many audio-related tasks (Scheirer and Slaney, 1997; Aucouturier and Pachet, 2002; Tzanetakis and Cook, 2002; McKinney and Breebaart, 2003).

One of the more important audio characteristics that researchers have been trying to extract is 'timbre'. Timbre refers to the colour of sound and is typically divorced conceptually from pitch and loudness (Wessel, 1979). Perceptual research on timbre has demonstrated that the spectral energy distribution and temporal variation in this distribution provide the acoustical determinants of human's perception of sound quality (Grey, 1975). Many researchers believe that the timbral quality of brightness correlated with increased power at high frequencies. For example, a note played at a high pitch generally has a higher spectral centroid than when it is played at a lower pitch, even when the note is played on the same instrument. Thus, spectral-based features may be able to help timbre-related audio tasks. The following features are commonly used to extract timbral-related information from an audio:

i) *Spectral Centroid*

The spectral centroid is defined as the centre gravity of the magnitude spectrum of STFT. It gives off the general spectral shape and is commonly used to approximate the brightness of a sound (Li and Tzanetakis, 2003, Tzanetakis, 2002). Sounds with higher centroid values indicate having higher frequencies present in the signal, and can be interpreted as having 'brighter' sound textures. Music with higher frequency noise, such as percussive sounds, typically have higher spectral mean.

ii) Spectral Rolloff

The spectral rolloff is another measure of the spectral shape. It shows the skewness of the spectral shape. The spectral rolloff point is the N% (N is usually 85% or 95%) percentile of the power spectral distribution, where the power spectrum is concentrated (Scheirer and Slaney, 1997). The rolloff point increases as the bandwidth of a signal increases. As the bandwidth is larger in music than it is in speech, it is also a most commonly low-level spectral feature used to distinguish between an audio file made of speech or music. It is also useful in music genre classification, if the difference in bandwidth signals between two audios of different genre is comparable, e.g. classical and rock.

iii) Spectral Flux

The spectral flux is another feature that can be used to determine the timbre of an audio signal (Grey, 1975). It measures how quickly the power spectrum of a signal is changing, by calculating the frame-to-frame spectral difference, i.e. power spectrum of one frame against the power spectrum from the previous frame.

iv) Pitch

The pitch feature typically refers to the fundamental frequency of a monophonic sound signal. Pitch itself is a subjective property of sound that can be used to order sounds from low to high, in the sense associated with musical melodies. Pitch can be calculated using various different techniques, such as autocorrelation method, cepstrum method and data reduction method. A more extensive comparison study on the different methods of extracting pitch can be found in studies by Hess (1983), Rabiner *et al.* (1976), and Tan and Karnjanadecha (2003).

The abovementioned features are all comprised of low-level audio features, where only sonic aspects of the sounds are encoded and extracted, with little or no input from human perception. The high-level features, on the other hand, are features that go beyond the raw spectral and cepstral information of a sound, by using methods that extract audio features by synchronising audio and metadata (Macrae, 2008).

Commonly, high-level audio features extract the metadata of a sound file, giving access to information such as the song name, artist, album, year released, label and possibly genre. The extraction of these metadata is made easier with the rise of MPEG-7, a multimedia content description standard which made the information to be readily available. High-level audio features can also capture symbolic data found in the likes of MIDI data such as notes, velocity and duration; as well as perceptual information such as pitch, texture, rhythm and tempo, among many.

Inclusion of symbolic data such as MIDI provides direct access to relevant score parameters, making it easier for high-level audio features to be translated by musicologists, due to the close relation to musical expressions (Abeßer *et al.*, 2009). It is therefore easily understood why high-level audio features have become as important as low-level features (if not more important than), with regards to feature extraction. However, the computational and memory load of extracting high-level audio features can be very high as a result of the extremely complex and sophisticated method that is required to perform the extraction. Hence, an intelligent decision must be made on whether or not the inclusion of high-level audio features will really bridge the gap of the performance, or whether the low-level audio features provide enough structure of the sound reliably.

Many studies have been carried out and many more are continually being conducted on the study of audio feature extraction itself, as it is the fundamental process in fields such as audio recognition, content-based audio classification and retrieval, and automatic musical genre classification (Tzanetakis and Cook, 2002; Klapuri, 2004; Mierswa and Morik, 2005). Audio recognition recognises speech in several ways: recognition of the spoken language, recognition of the speaker and finally recognition of the speaker's emotions (Eronen and Klapuri, 2000). While exact match is expected between the target sound and the sound in the database in audio recognition, content-based audio retrieval, on the other hand, functions by searching the database to retrieve sounds that are similar to that of the queried sound. An audio retrieval system normally returns a list of several similar sounds which are presented to the users in a rank where sounds appearing on the top are those with closer similarity to the target sound. Users can then select the most relevant sound from the list (Wold *et al*., 1996; Zhang and Kuo, 1998).

Audio retrieval process is first subjected to analysis and classification of sounds. Sounds can be classified into several categories such as speech, music, environmental sound, silence and so on. This is the basis of another field that involves the extraction of the audio features in order to function, which is audio classification. Work on audio classification has then been extended to include hierarchical classification, where sounds are separated into much finer classification (Tzanetakis and Cook, 2002; Klapuri, 2004; Mierswa and Morik, 2005). An example of this hierarchical classification is seen in musical genre classification where music is further classified into respective genres (Figure 15).

The works on feature extraction in music genre classification are not only limited to western music, but to other forms of music across the world, as evident from the works of Petri Toiviainen (Toiviainen and Eerola, 2001), Noris Mohd Norowi (Norowi *et al.*, 2005) and

Aniruddha Ujlambkar (Ujlambkar and Attar, 2012). Their works had been based on the classification of Chinese folk's songs, traditional Malaysian music and Indian popular music respectively.

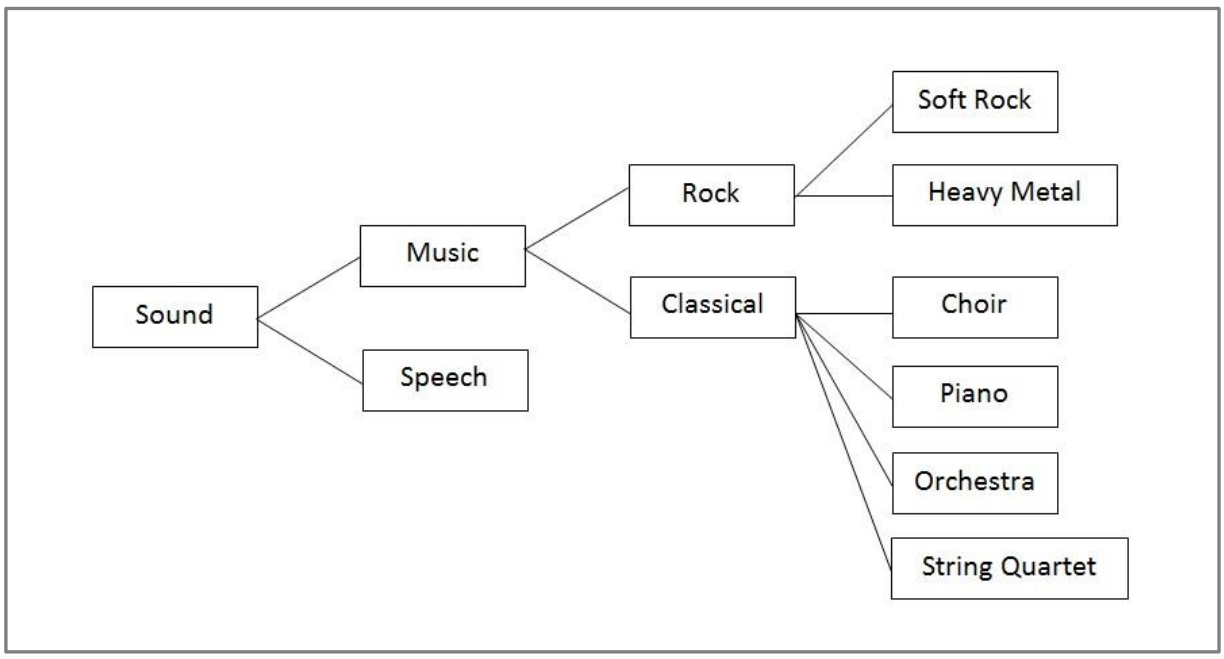


Figure 15: Music genre classification hierarchy

In order for the tasks discussed above (audio analysis, classification, retrieval and recognition) to work, audio signals need to be subjected to feature extraction first. The raw information obtained from the extraction, coupled with the AI embedded in the system, enable the analysis of the sound content to be made and further generate the output.

It is clear that audio features play a crucial part in any task involving the analysis of a sound. However, there are still many challenges in this area, the biggest being which audio features to extract. Audio features can be based on temporal (ZCR, energy), cepstral (MFCC), perceptual (loudness, pitch), physical (STFT, auto regression) and psycho acoustical model (signal-to-mask ratio). To include all of these existing audio features is impractical (and almost impossible); therefore the features extracted must be meaningful and show high variation across the audio classes. Ideally, the number of features included needs to be kept

at a minimum to reduce the computational load and run time. In addition, some forms of calibration are needed so that the features are not too sensitive to the noise or slight fluctuations in the signal, as this can result in flawed analysis.

2.3.5 Unit Selection

In unit selection, all search methods have been developed with one basic notion: to find the optimal match between a target unit and the source units in the database, within the least amount of time and using the fewest possible resources. However, there are differences in the way some algorithms are tuned to perform the search task, which is why in many cases, performing the same search on a different method will often produce dramatically different results.

In some circumstances, a search method performs better than others because of its execution design. In order to utilise the full potentials of these search methods, it is important to understand their strengths and weaknesses. For instance, the basic brute force algorithm is actually the only search method that can guarantee an optimal solution, but this method carries a huge overhead that increases exponentially as the dataset size increases (Korf, 1985). Therefore, this method is only suitable when the database is small and it is imperative that the most optimal solution is found.

A faster alternative to this is the Viterbi algorithm, which has already been used in several existing CSS systems (Schwarz, 2000; Maestre *et al.*, 2009). The Viterbi algorithm gives the best interpretation of the entire context and reduces computational complexity by using recursion (Forney, 1973). It is good for solving ambiguity when the confidence level is low, but because it looks at the whole sequence before deciding on a most likely final state in process known as backtracking, it also runs the risk of being too exhaustive (Figure 16). For

example, a search task involving four different states at six given times (24 nodes), returns 4096 possible paths (4^6).

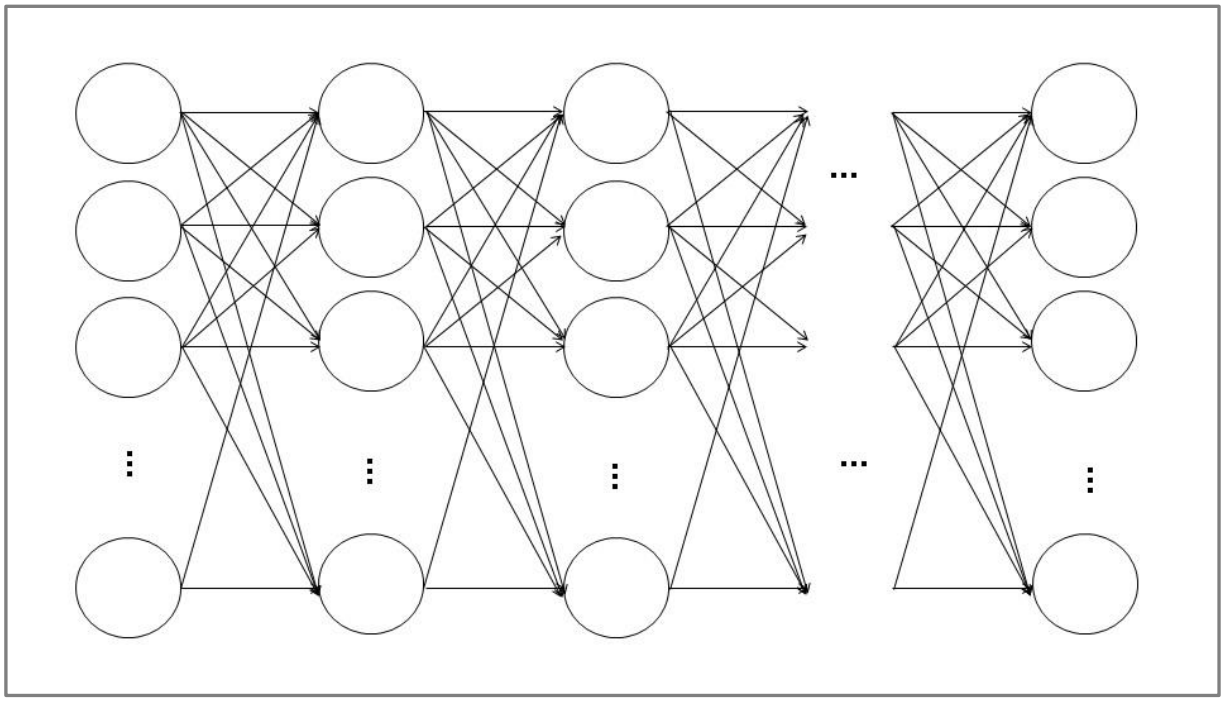


Figure 16: Trellis representation showing the parallel implementation adopted in the Viterbi algorithm. Every possible path needs to be first determined before backtracking to determine the most likely final state

The K-Nearest Neighbour (KNN) is another popular search method that is used in the unit selection process of a few CSS systems (Lazier and Cook, 2003; Schwarz *et al.*, 2006). When a new sample arrives in the database, KNN finds the K neighbours nearest to the new sample from the training space based on some suitable similarities or distance metric such as the Euclidean distance (Figure 17). It has been shown that KNN can perform well in many situations. The error of the nearest neighbour rule is bounded above by twice the Bayes error under certain reasonable assumption (Cover and Hart, 1967). However, its performance is normally reduced as its training set increases. The need for dataset training is also another disadvantage of this algorithm compared to other algorithms implemented for CSS systems. Training may take up additional time, but this can be remedied by indexing (K-D tree) or optionally having the process done offline in advance.

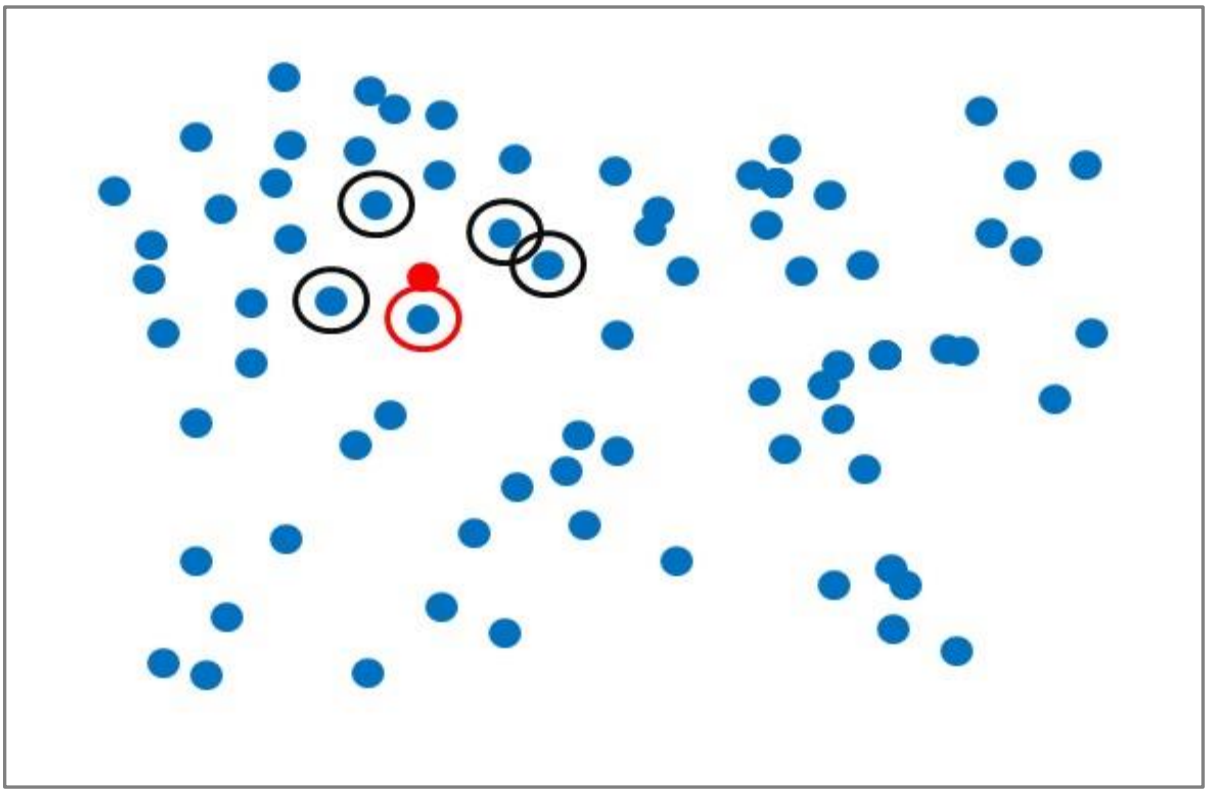


Figure 17: X-Y scatterplot of samples based on their pairwise distances. The red point is the query vector (target) and the blue points represent the data. Using the KNN algorithm, at k=5, all the circled points are the query's nearest neighbours. When only one match is expected (k=1), the closest point to the target (circled in red) is selected.

Local search algorithm is also known to have been implemented as the search selection algorithm of choice for several existing CSS systems (Zils and Pachet, 2001; Aucouturier and Pachet, 2005). Local search algorithms consist of several methods for combinatorial optimisation by performing a sequence of local changes in an initial solution, which improve each time the value of the objective function, until a local optimum is found (Mladenović and Hansen, 1997).Unlike certain exact search methods, it avoids systematic search throughout the database. Rather than iteratively trying to improve search results step-by-step until the closest solution is found, local search algorithms adopts a more randomised approach and returns approximated solutions. By doing so, it is able to complete the task fairly quickly, especially with problems of modest sizes where no known solution is found though other exact methods. This makes it an ideal search algorithm for applications where

approximated solutions can be accepted. Local search algorithm has been implemented to aid applications such as vehicle routing, job scheduling and network optimisation. There is a wide variety of local search algorithms, among them are random walk, hill climbing and simulated annealing. Direct descendants of the algorithm are also present, such as the adaptive search algorithm and incremental search algorithm.

Whichever algorithm is used in the path to finding the optimal matching segment, a measure of similarity must be used to compare the distance between the target unit and the units in the database. The most common way to solve this is through the use of Euclidean distance (Gower, 1985). Based on the Pythagoras theorem, the Euclidean distance measures the straight line distance between two points. When multidimensional features are used, the Euclidean distance calculates the distance between two vector points, *x* and *y*, and is given in the equation (1) below, where x_j (or y_j) is the coordinate of *x* (or *y*) in dimension *j*.

$$\mathrm{d_{x,y}} = \sqrt{\sum_{\mathrm{j=1}}^{\mathrm{J}} \left(x_j - y_j\right)^2} \tag{1}$$

There are many more search methods available that might be as useful for finding the match between the target unit and the source unit. However, it is clear that each of them is designed to carry out search in a slightly different manner. When options are made available, users can decide on which search method is most fitting, taking into account trade-offs such as accuracy, speed and computational load.

2.4 Summary

In order to understand the working mechanism of a CSS system, the underlying principles presented in this chapter need to be understood. This chapter discussed the roots of CSS, from the different approaches presented (rule-based and data-driven synthesis) to the examples of successful applications of concatenative synthesis in other more established domains such as speech, singing voice and other sounds.

The technical overview of a typical CSS system is also described; from audio segmentation, to feature extraction and finally unit selection, with in-depth details of each of the processes involved at every stage. This includes an introduction to feature vectors and trajectory space, discussions on different audio features representations, a number of search algorithms and the similarity measurement used to compute the distances between two sound segments.

The next chapter will present a review of several CSS systems and will compare the different approaches implemented by the systems in order to execute the processes described earlier in this chapter. Issues and limitations that arise from these systems will also be highlighted.

Chapter 3: Existing Concatenative Sound Synthesis Systems and Issues

Sound creation styles that inspired CSS, such as Musique Concrète; were originally used to compose music manually. It was not until a little over a decade ago that some of these processes became automated. As explained in Chapter 1, this change was, in large part, a direct result of technological advancements. This change meant that the interest in CSS was no longer contained within the artistic community, but has expanded into many other interdisciplinary areas including artificial intelligence and digital signal processing, in terms of research, applications and even commercial software.

The first half of this chapter compares the performance of several existing CSS systems. The second half of this chapter then presents the issues and challenges that exist between these systems, with the intention to provide possible solutions to overcome them throughout this thesis.

3.1 Review of Existing Concatenative Sound Synthesis Systems

The existing CSS systems included in this review are namely *Caterpillar* (Schwarz, 2000), *Musical Mosaic* (Zils and Pachet, 2001), *Mosievius* (Lazier and Cook, 2003), *MATConcat* (Sturm, 2004), *Soundspotter* (Casey, 2005), *Audio Analogies* (Simon *et al.*, 2005), *Ringomatic* (Aucouturier and Pachet, 2005), *CataRT* (Schwarz, 2006), *Expressive Jazz Synthesis System* (Maestre *et al.*, 2009), and the *Database System for Organizing Musique Concrète* (Bailey, 2010). The strengths and weaknesses of each system are discussed with respect to these five areas: (1) input mechanism, (2) feature analysis, (3) match specification, (4) synthesis and use of transformation, and (5) real-time capability. A summary of this can be found

presented at the end of this sub-chapter in Table 1, whilst brief descriptions of the CSS systems included in this review are as follows:

Caterpillar- The *Caterpillar* system (Schwarz, 2000) is one of the earliest automatic CSS systems developed. It was designed to synthesise sounds based on the most appropriate segment of sound units in the database. Through the application of various modifications, the desired melodic phrase can be built. *Caterpillar* has a sister system, *Talkapillar which is* an experimental text-to-speech (TTS) system that uses the same architecture to create hybrid synthetic speech with phrase units. This is particularly useful in tasks involving reconstruction of a speaker's voice.

Musical Mosaic- The name for *Musical Mosaic* came from Robert Silver's work on Photomosaic. Just as micro images are used to synthesise a different image at macro level, Zils and Pachet (2001) used small segments of audio to assemble a larger, different piece of sound. The new sound is composed by imitating the sequences in a target sound using constraint satisfaction programming (CSP).

Mosievius- Lazier and Cook (2003) base *Mosievius* on the same principles as previous CSS systems, but expand their mosaicing techniques to develop a system that has a more interactive control over the selection of sound units.

MATConcat- Sturm (2004) originally intended *MATConcat* to be a free and open application implemented on MATLAB that can be used by many so that the concept of CSS could be demonstrated and understood through navigation and first hand experimentation of the system. Although its algorithm is simpler than most other CSS systems, interesting pieces have been composed using MATConcat, namely *Dedication to George Crumb, American Composer* and the *Gates of Heaven and Hell: Concatenated Variations of A Passage by*

Mahler, both of which have been premiered at the International Computer Music Conference in 2004.

Soundspotter- *Soundspottter* (Casey, 2005) is an open source software that can be used to create new sounds. It is implemented in C++ and uses methods from music information retrieval. It follows the same idea as other CSS systems where it 'spots' the source sounds in the database that match the target sounds and concatenates them together, but in *Soundspotter*, this happens in real-time. Earlier works composed using *Soundspotter* include *Departure on the Chao-Phraya* (2005) and a piece composed with Roger B. Dannenberg, *SueMe No. 1* (2005), premiered at City University, London and at Goldsmiths College, University of London, in the years 2004 and 2005 respectively.

Audio Analogies- *Audio Analogies* creates new sounds by finding an audio recording in the database that matches target, which is in the form of MIDI score, through an example MIDI score and audio recording pair as a guide. The team of researchers from the University of Washington and Microsoft Research who came up with such concept described the mechanics of their system as "… using MIDI scores A and B and raw sound A' as input, to produce a new raw sound B', such that the relationship between A and A' is the same as the relationship between B and B'... " (Simon *et al*., 2005).

Ringomatic- *Ringomatic* is Aucouturier and Pachet's (Aucouturier and Pachet, 2005) real-time CSS system that is designed specifically for generating new audio drum track by concatenating drum segments together from pre-existing musical files in the database. Using CSP, the composition of drum tracks can be controlled, much like the sample-based virtual drummer system, Fxpansion's BFD (Fxpansion, 2003).

***CataRT*-** This is the second CSS system that Diemo Schwarz (Schwarz, 2006) developed, which was loosely based on his previous system, *Caterpillar*. Amongst the improvements that have taken place in *CataRT* include its capability of running in real-time, and also its ability to take in not only recorded audio, live audio, and MIDI scores as targets, but have now included descriptors and segmentation markers that have been pre-extracted from other programmes.

***Expressive Jazz Synthesis System*-** A team of researchers from the Music Technology Group at Universitat Pompeu-Fabra, Spain developed a CSS system that resynthesises audio recordings of jazz saxophone melodies, with special emphasis on the expressiveness of the performance (Maestre *et al.*, 2009). The scores and recorded audio of several performances are translated into a performance model, which are used to train the system to identify the characteristics and relationships that exist between score, description of performance and audio, at different temporal levels. Through the use of inductive logic programming techniques, notes that correspond to both score and expressiveness of the piece are concatenated.

***The Database System for Organizing Musique Concrète*-** Christopher Bailey intended the *Database System for Organizing Musique* Concrète to be a much simpler and more flexible system than most typical CSS system (Bailey, 2010). To achieve this, Bailey designed the sound data storage and entry module, where basic parameters are employed to describe the audio in place of complex audio descriptors. He also introduced the use of a graphical score to help visualise the sonic gestures of a sound, and provided the option to import the composition unto a mixing application such as Ardour for further transformation and adjustments.

3.1.1 Input Mechanism

The most common audio inputs recognised by existing CSS systems are in the form of audio recordings, the most popular formats being WAV, AIFF, or MP3, which are accepted by all ten of the systems included in this review. Some systems like *Mosievius* (Lazier and Cook, 2003) and *Audio Analogies* (Simon *et al.*, 2005) accept both audio and MIDI files, giving their users a wider option of source and target sounds to work with. *CataRT* (Schwarz, 2006) advanced a step further by not only accepting both audio and MIDI files, but also pre-processed segmentation markers, i.e. SDIF and ASCII files that can be piped directly from other programmes. It can also accept raw descriptors such as found in the MPEG-7 low-level descriptors or descriptors calculated in the original Max or MSP patches, exploiting the symbolic information that is already present within the input file. Such step reduces the need for the segmentation and feature extraction that typically follow, as unit selection can be made directly from the descriptor files. Additionally, in some systems the input sounds are not restricted to the use of audio recordings only, but can also include live input such as from a microphone, as seen in *Soundspotter* (Casey, 2005) and *CataRT* (Schwarz, 2006).

Once the audio has been entered into the system, it needs to be segmented into smaller sized audio, in order to make sense of any underlying pattern that might be found via the feature extraction stage that follows. There are various ways in which this can be performed. Within a CSS system itself there are at least four different segmentation approaches identified, such as fixed, blind, on-the-fly, and audio alignment, to name a few.

The simplest approach is quite possibly the fixed approach, as applied in *MATConcat* (Sturm, 2004). In this approach, a constant hop size is used to trim the audio, resulting into uniformed length segments. A more widely used approach, however, is known as the blind approach – *Musical Mosaic* (Zils and Pachet, 2001), *Mosievius* (Lazier and Cook, 2003),

CataRT (Schwarz, 2006) and *Expressive Jazz Synthesis* (Maestre *et al.*, 2009) all employed this approach. In comparison to the fixed approach, segmentation in this approach is not based on a constant hop size, but instead happens at a certain specified level, for instance, at every note level or intra-note level, e.g. *Mosievius* (Lazier and Cook, 2003), *Expressive Jazz Synthesis* (Maestre *et al.*, 2009); or at the detection of different segmentation stage such as the occurrence of silence or high frequency content e.g. *CataRT* (Schwarz, 2006).

A slight variation to this approach is known as the 'on-the-fly' approach, a name that is given to the approach that is similar in concept, but with its segmentation done in real-time. This is used in *Soundspotter* (Casey, 2005), with the option to segment the audio at three different levels; periodic windowing, inter-onset interval, and beat. In *Audio Analogies*, segmentation is also performed on the note level, either using the pitch and duration information, candidate frames or wave frames. However, this step is currently carried out manually.

In systems where musical scores and other symbolic information are accepted as input files, e.g. *Caterpillar* (Schwarz, 2000), the audio alignment approach is used. Audio alignment aligns the acoustical musical signal with symbolic information such as the score. In the CSS system for drum tracks such as in *Ringomatic* (Aucouturier and Pachet, 2005), the drum solo part in any large musical section is first identified and then segmented into a 4-beat drum bars.

Some systems such as the *Database System for Organizing Musique Concrète* (Bailey, 2010) may opt to skip the segmentation process altogether by pre-trimming the input audio into very small audio chunks, for example between 500 milliseconds to 1 second long.

The resulting segmented audio can be categorised as either homogeneous or heterogeneous in nature. Homogeneous segments are uniform in character, and are usually near-similar in

length. Segmentations that are performed using the manual and fixed approaches generally result in homogeneous segments.

3.1.2 Features

Survey of existing CSS systems show that the systems either utilise the low level features or both low and high-level features during extraction. The first group, those which use low-level features only typically involves the use of spectral (centroid, flux), cepstral (MFCC) and temporal information (ZCR, RMS), with possible inclusion of pitch, onset and beat information too. The use of low-level features generally means that often large and noisy raw data are included in the analysis of the sound. However, since such problem can normally be solved through the application of dimension-reduction method such as Principal Component Analysis (PCA), the features can still extract measurable properties from the audio signal to detect any relevant pattern that the audio may contain. Moreover, the low-level features contain information that are in simpler form and can therefore be stored more efficiently, making this type of feature representation still desirable and used in CSS systems such as *Musical Mosaic*, *Mosievius*, *MATConcat* and *Audio Analogies*.

The remaining six CSS systems extract high-level audio features in addition to the low-level audio features mentioned above. These high-level audio features can be in the form of MIDI note numbers or symbolic information such as style, artist, genre and duration. They can also be in the form of psychoacoustical descriptions such as roughness and sharpness. Although the extraction of high-level audio features is more time consuming, the integration of keywords and other symbolic information can bridge the semantic gap by relaying additional knowledge for a specific domain, as such feat cannot be accomplished through the use of low-level features alone. To minimise the issue that is present with regards to the time it takes to perform feature extraction, the process is sometimes performed offline.

3.1.3 Match Specification

There are several factors that determine how the match for a target unit is found in the database of a CSS system. One of the more important criteria is the algorithm employed to execute the search. Different search methods are designed to solve different problems but are all aimed at finding a match between the query (target sound) and the instances in the database. Selections of the match algorithm implemented in a CSS system can be a case of individual preferences or can be influenced by the nature of the task at hand, as some algorithms are designed to handle certain tasks better. Several of the more prominent search algorithms of choice for CSS systems are Viterbi algorithm, KNN and local search algorithm. The basic working mechanisms of these algorithms have been presented earlier in Chapter 2 (p. 20).

The Viterbi algorithm has been found to be the most used search algorithm in existing CSS systems. It is seen used in *Caterpillar*, *Audio Analogies* and the *Expressive Jazz Synthesis system*. Systems that used the Viterbi algorithm typically return the most probable sound segments, given the waveform of the target sound. However, as the Viterbi algorithm dictates that the entire search space containing all possible matches to be explored and compared before a final decision is made, it takes up more time than is ideal to be executed in real time. Thus, the three systems above are all unable to perform in real-time, such as in front of a live audience.

Perhaps this is why *CataRT*, an extension of the *Caterpillar* system, implemented the KNN algorithm instead. KNN is the second most common search algorithm implemented in existing CSS systems. It is most advantageous when little prior knowledge is known about the distribution of the data, but strong consistency in the result is required. KNN is known for its fairly simple and fast computation, thus allowing systems which undertook it such as

and *CataRT* and *Mosievius* to run in real-time. However, this is true only for the naïve version of KNN. This algorithm requires the data to undergo training first, which can increase the computational costs as the file size of the training set grows.

A few other existing CSS systems such as *Musical Mosaic* and *Ringomatic* used descendants of the local search algorithm, such as the adaptive search algorithm and the incremental search algorithm respectively. Local search algorithms can return a solution even if it is far from optimal. In comparison to exact matching, approximated solutions can return different and very interesting results which can be appealing in music synthesis. These 'accidental creations' can sometimes be pleasantly surprising, a situation described as the 'Aha' phenomenon. It is a situation where the unexpected generation of a sound is actually considered interesting (Aucouturier and Pachet, 2002). As such, systems that implemented this algorithm may be more suited for compositional pieces that loosely match the target sounds.

There are many more search methods available that might be just as useful for finding the match between a target unit and a source unit. Trade-offs such as accuracy, speed and computational load play an important part in deciding which search method is most fitting for a particular CSS system.

The most common way of determining the distance of matching units from the database with the target unit is by calculating their Euclidean distance. These distances are the metric measurements that amplify more of the difference of specific parameters of the feature vector than calculating the absolute difference on its own (Pantazis *et al.*, 2005). With the exception of *CataRT* that uses Squared Mahalanobis distance, and the *Database System for Organizing Musique Concrète* that compares graphical score that represents sonic gestures,

all of the other reviewed CSS systems use Euclidean distance as a measure to calculate the distance between the target unit and source units in the database.

3.1.4 Synthesis and Use of Transformation

Two types of synthesis control practised in CSS system are the fixed synthesis and unfixed synthesis. Fixed synthesis is referred to cases where the system has full or partial control over the segments to be synthesised, with little or no input from users other than the parameters which have been selected earlier. On the other hand, unfixed synthesis is where a system requires input from users to finalise the selection before synthesis takes place.

CSS systems typically fall into the first type of synthesis control due to the algorithms that were implemented during the selection process. This is true for *Musical Mosaic* and *Ringomatic*, where the adaptive search algorithm for the former and the constraint satisfaction programming for the latter force their sequences to be refined until only one match is found. However, it is possible with *Ringomatic* to choose different sound generating methods such as through via Genetic Algorithm (GA) and random generation. Similarly, whilst *MATConcat* does not offer its users the opportunity to hand-pick final sound segments, it does allow its users to specify their synthesis options. For example, users have the choice to choose between 'Force Match', 'Extend Match', or 'Leave Blank', when no match is found, and also the option to enable 'Random Match' when more than one match is found.

CSS systems that fall into the second type of synthesis control, or the unfixed synthesis, usually present users with the options in the form of a listed possible segments (e.g. in the *Database System for Organizing Musique Concrète*), or sometimes through the help of a visual map that consists of dots in space which represent all the possible segments in the

database and their relationship between one another (*Caterpillar*, *CataRT*, *Mosievius*). Figures 18 and 19 depict the different selection approaches in CSS systems that have unfixed synthesis control. Either approach will require the users to make a final selection on which segments that 'make the cut' for synthesis.

Despite all the conditions set prior to unit selection, the sounds generated from the system may still require some form transformation post-synthesis. Although not all CSS systems have the support for transformation, some systems such as the *Caterpillar*, *CataRT*, *Mosievius*, *Audio Analogy* and *Expressive Jazz Synthesis System* do. Transformation is most commonly offered in the form of loudness change, basic fade in or out, and also pitch and duration modifications, which can be achieved through different PSOLA techniques (Moulines and Charpentier, 1990; Lemmetry, 1999; Mousa, 2010). The *Database System for Organizing Musique Concrète* uses a different approach by allowing synthesised sounds to be realised into the mixing application Ardour Mix for transformation to commence.

Figure 18: Unfixed Synthesis through the use of list

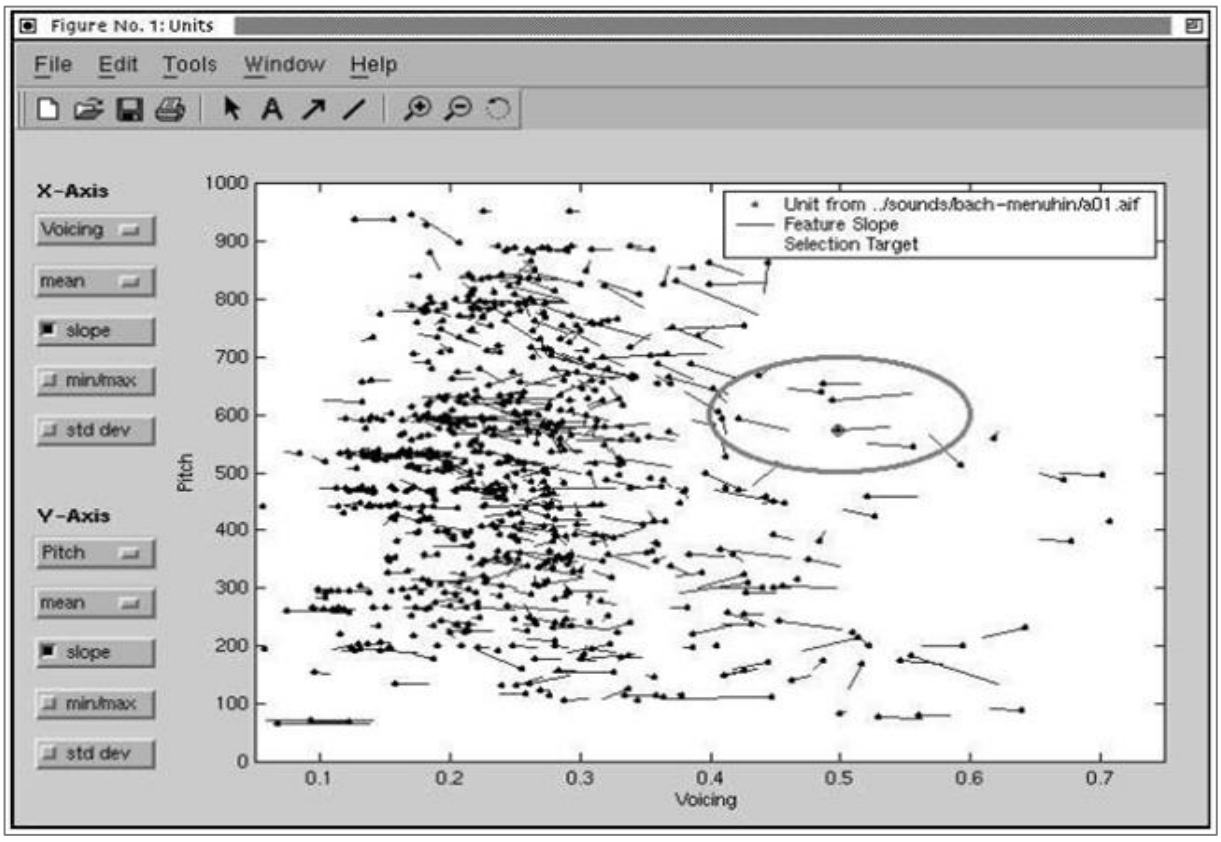


Figure 19: Unfixed Synthesis through the use of visual map

3.1.5 Real-time Capabilities

The term real-time in this sense refers to a system where the input data is processed within milliseconds so that it is available virtually immediately as feedback. A real-time CSS system is expected to be able to synthesise as soon as the new target or input sound is relayed unto the system. This suggests that the selection algorithm needs to be really efficient and able to perform all the necessary transformations quickly. Real-time CSS systems include *Mosievius*, *Soundspotter*, *CataRT* and *Ringomatic*. Although the sounds generated from real-time CSS systems may be more susceptible to loss of sound quality in comparison to non-real time CSS systems (Schwarz, 2006), real-time capability is still desired especially when live interaction is expected, for example in a concert in front of a live audience.

Table 1: Summary of the strengths and weaknesses of ten existing CSS systems

CSS system	Year	Corpus	Segmentation	Features		Unit Selection	Transform.	Real-time	Concat. Dist	Reference(s)
				Descriptors	Weight					
Caterpillar	2000	Audio Symbolic (MIDI score)	Hetero	Continuous (pitch, energy, spectral); Symbolic value (attack, sustain); Discrete (MIDI note number)	No	Viterbi	Fundamental frequency, energy, spectral, resampling filtering	No	Yes	Schwarz (2000)
Musical Mosaic	2001	Audio	Homo	Mean pitch, loudness, percussivity, global timbre	No	Adaptive search	-	No	Yes	Zils and Pachet (2001)
Mosievius	2003	Audio, Symbolic (MIDI score)	Homo	Voicing, energy, spectral flux,	Can set cardinal rules	KNN	OLA/PSOLA	Yes	No	Lazier and Cook (2003)
MATConcat	2004	Audio	Homo	ZCR, RMS, pitch, spectral centroid, spectral rolloff, harmonicity	Can set order, but not weight, i.e. 10% ZCR, 5% RMS	-	-	No	No	Sturm (2004)
Soundspotter	2005	Audio, Live Input	Homo (on the fly)	MFCCs, FFTs (cepstral coefficients); MPEG LLDs (ID3 tag)	No	Matched Filtering	-	Yes	No	Casey (2005)
Audio Analogies	2005	MIDI score, audio, recorded instrument	Manual (notes)	Pitch	No	Viterbi	OLA	No	Yes	Simon *et al.*, 2005
Ringomatic	2005	MIDI score	4-beat drum	LLDs; Symbolic (drum detection, energy, onset density, presence of cymbals or drums)	No	Increment Adaptive Search	-	Yes	Yes	Aucouturier and Pachet (2005)

CataRT	2006	Audio, segmentation markers, raw descriptors	Hetero	Spectral (loudness, spectral centroid, spectral tilt, HFC, harmonicity); Descriptors (unit ID, unit duration, file)	No	KNN	fade in/out, pitch resampling, loudness change	Yes	No	Schwarz (2006)
Expressive Jazz Synthesis	2009	MIDI score	Homo	MIDI (pitch, onset time, duration); LLDs (energy, mean spectral centroid, mean spectral tilt); descriptors (attack level, sustain slope)	No	Viterbi	global energy transform, pitch shifting, time stretching,	No	Yes	Maestre *et al.*, (2009)
Database System for Organisation of Musique Concrete	2010	Audio	Homo	LLDs; Symbolic (pitch class, duration, loudness, agitation)	Can set importance in the scale 1-7	-	Can transfer straight to Ardour Mix to transpose, filter, delay, etc.	No	No	Bailey (2010)

3.2 Issues in Existing Concatenative Sound Synthesis Systems

Previously, the state-of-the-art of ten CSS systems was reviewed. Comparisons have been made between those ten systems for five different criteria that covered the major steps involved in order for any basic CSS system to function – from the input of target sound to transformation of output sound. However, within those steps, there still lie issues that have not yet been fully resolved. This section of the study identifies and presents several of these issues at hand.

3.2.1 Order-Dependent Feature Selection

Audio features are the building blocks for many tasks involving audio, such as audio recognition, audio retrieval, audio classification, audio segmentation and audio synthesis. Usually, more than one feature needs to be exploited at any one time in order to draw any significant pattern of correlations that might exist. Having said that, although there are potentially endless combinations of audio features available, it is important to only include the more relevant audio features, as the algorithm used to perform these tasks, (segmentation, synthesis and retrieval) will always return some kind of result. On the other hand, including a poor feature representation will only yield results that do not reflect the real nature of the underlying data. Moreover, the overuse of audio features typically slows down processing time as it exhausts computational resources such as processing power and memory. Thus, it would be more computationally economic and time saving to have only the relevant features extracted.

Aptly, most CSS systems such as *Caterpillar*, *Musical Mosaic*, and *CataRT* have already enabled their features selection option, allowing users to take control of which features they would want to include. However, the majority of existing CSS systems assume that all

features carry the same weight and do not allow for these features to be sorted according to their order of importance. *MATConcat* offers a slightly flexible option by allowing its users to decide on the order of features and its tolerability level (the distance between target and match is allowed should there be no exact match in place), but it does not take into account the weight of each feature with respect to one another. *Musical Mosaic*, on the other hand, implements the use of weights, but its use is not targeted on differentiating the importance between features but as a mean to prioritise different cardinal rules, for example it is three times more important to obtain the correct pitches than to obtain all unique sound segments, as opposed to finding segments that match pitch is twice as important as those that match the intensity. Moreover, the weights are assigned manually by users, which can be arbitrary and may result in some form of inconsistency.

The inconvenience in lacking some form of weight-assignment mechanism becomes apparent when the importance of the features is not equal. For instance, although two or more features may both be important to be included in a particular search, there may come a time where one feature takes precedence over the other, for example it is twice as important to find a segment that matches the values of Feature A than it is Feature B. In such a case, weights must be assigned with respect to each feature, as this may affect the result of the unit selection, and ultimately, the final sound generated.

There are several situations where the importance of different features may not be equal. For example, if a composer wants synthesised sounds that are loud and dynamic, he may select the features such as low energy and ZCR to be included in the matching segment search. However, since fluctuations in ZCR can sometimes be the result of high level of noise in the sound signal, it is not always as reliable an indication for dynamicity, as low energy is.

Thus, due to its infidelity nature, the composer may set lower importance on the ZCR in comparison to low energy.

Favouring one feature over the other in this manner is termed features prioritisation and typically affects synthesis result. Consider the situation in Figure 20 as an example. A target segment is given with the criteria as shown in the box on the left, and a matching segment is needed to be found from the source segments that are available in the database (box on the right). For the sake of simplicity, only two audio features are included as the criteria to match in this search ($x_1$and x_2), and only two source segments are present in the database (*#1* and *#2*). For each of the cases presented below, a Euclidean distance is calculated using equation (1) as previously given (Chapter 2, p. 54) or its derivation there from is used to select the source segment which has closest feature values (y_1 and y_2) to the target. The cases that follow have been set up to demonstrate the need for an order-dependent feature selection in sound synthesis.

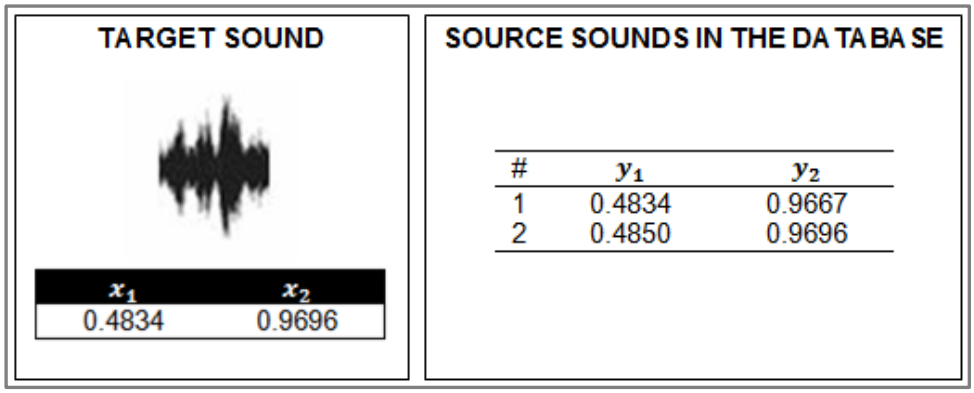


Figure 20: Target segment and source segments

Case 1: Feature 1 and Feature 2 are of equal importance

When both features are regarded as equally important, the Euclidean distance is calculated using equation (1) as previously described (Chapter 2, p.54) to determine the distances, $d_{(x,y)}$, of the two source segments in the database from the original target, as presented in Table 2(a). Thus, the source segment with the least distance (source #2) is selected.

Case 2: Feature 1 is five times as important as Feature 2

Since the importance of the features is dissimilar, a weighted Euclidean distance equation is now used to calculate the distances of the source segments from the target.

$$\mathrm{d_{x,y}} = \sqrt{\sum_{\mathrm{j=1}}^{\mathrm{J}} \mathrm{w_j}\left(x_j - y_j\right)^2} \qquad (2)$$

Being five times more important, Feature 1 has a weight value of *w*=5, whilst the weight value for Feature 2 is *w*=1. This recalculation changes the distance results to those shown in Table 2(b). Hence, in this second case, source #2 that was selected in the previous case is no longer the most optimum selection, but is replaced by source #1 which now has the least target distance.

Case 3: Feature 2 is five times as important as Feature 1

In this case where the situation is reversed, equation (2) above is once again used to calculate the distances. However, the situation is now reversed, where Feature 1 is assigned a weight of *w*=1, whilst Feature 2 is more important and carries the weight of *w*=5. From the result displayed in Table 2(c), under this newly set condition, the selected segment has once again changed to source #2, although the distance value is slightly different from Case 1.

Case 4: Feature 1 is three times as important as Feature 2

The selected segments and the target distance changes from one case to the next, depending on the condition and weights assigned for each features. For instance, if Feature 1 is kept as being more important than Feature 2 (as per Case 2), but is assigned with a lesser importance, i.e. w=3, the value of calculated target distances will change again, and source #2 is selected instead, as in Table 2(d). This suggests that not only the order of importance is critical, but also the intensity of how much more important a feature is over another is equally significant.

The four cases above demonstrate the effect which importance and intensity of audio features can have on concatenation. By assigning different importance values to different features, the distance between the source segment and target segment is altered. This influences the selection of segments that will be used for concatenation. As different segments may be representing different sounds, the entire production of sound synthesis is affected, especially when the database contains much larger segments than these examples.

SOURCE SOUNDS IN THE DATABASE

#	y_1	y_2	$d_{(x,y)}$
1	0.4834	0.9667	0.0029
2	0.4850	0.9696	0.0016

Table 2(a). Result of Euclidean distances between target and source when all features have equal importance

SOURCE SOUNDS IN THE DATABASE

#	y_1	y_2	$d_{(x,y)}$
1	0.4834	0.9667	0.0029
2	0.4850	0.9696	0.0035

Table 2(b). Result of Euclidean distances between target and source when Feature1 is five times as important as Feature2

SOURCE SOUNDS IN THE DATABASE

#	y_1	y_2	$d_{(x,y)}$
1	0.4834	0.9667	0.0065
2	0.4850	0.9696	0.0016

Table 2(c). Result of Euclidean distances between target and source when Feature2 is five times as important as Feature1

SOURCE SOUNDS IN THE DATABASE

#	y_1	y_2	$d_{(x,y)}$
1	0.4834	0.9667	0.0029
2	0.4850	0.9696	0.0027

Table 2(d). Result of Euclidean distances between target and source when Feature1 is three times as important as Feature2

3.2.2 Homosonic and Equidistant Unit Selection

The most crucial stage in a CSS system is unit selection, as this is the stage which determines which segments will be selected to make the final concatenated sound. As seen earlier, a slight change in the segment selection can alter the overall sequence of segments, resulting in different sound creations. Normally, the process is straightforward, where the system scans the database for a source segment that most closely matches the specified criteria (audio features) of the target segment, irrespective of the algorithm chosen to drive the search. However, if the database is large enough, several source segments that equally satisfy the criteria set by the target segment may become available. These segments are by no means redundant segments, but are in fact, different sounds that happen to be represented by the same sonic information with one another (Figure 21).

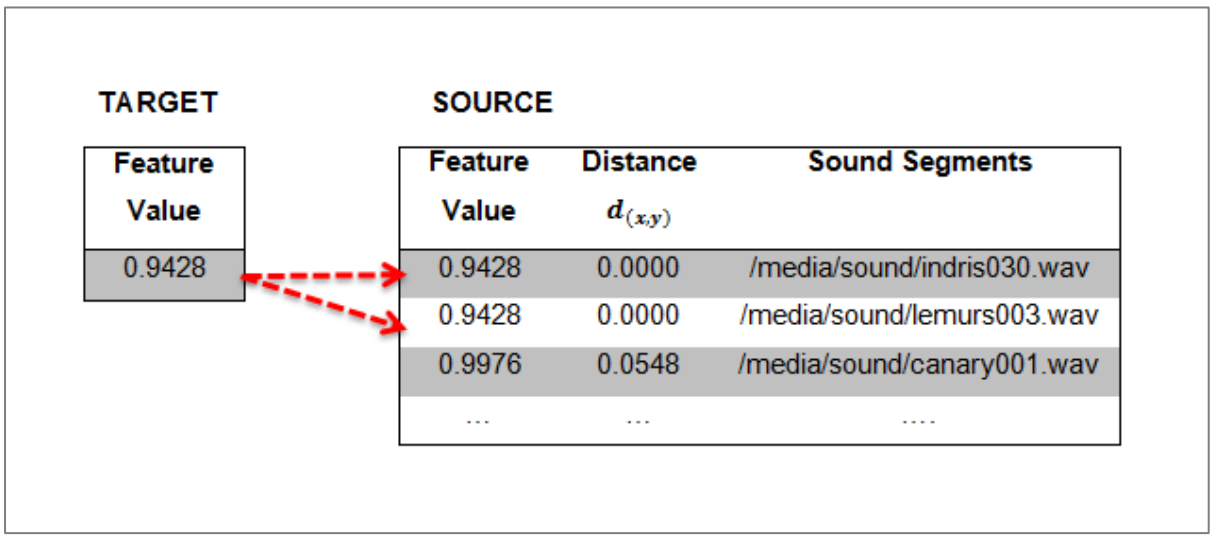


Figure 21: Unit selection involving homosonic segments

Audio segments that have the same sonic information, but are different acoustically and physically, are referred to as 'homosonic' in this thesis. Homosonic audio can be likened to the term 'Homograph' in the linguistic sense, where it is defined as a word that shares the same written form as another word but has a different meaning, and when spoken, the

meanings may be distinguished by different pronunciations. An example of a homograph is the word desert, where it can be meant for both the arid, dry region and the action of leaving. Only upon observing the whole context of the sentence that contains the word or upon pronouncing the word itself, will it become clear which meaning is relevant. Likewise, homosonic sounds are audio that may be represented to have the same sonic properties with each other, but do not sound the same when played. This can happen when the use of only one (or very few audio) features is compared, and the sound segments may appear to have identical values for these features. Only when additional features are revealed that it becomes apparent that the two sounds have different audio signal make up. For example, two homosonic sounds may carry the same values when the intensity level is compared, but when played, both sounds are very different timbrally. This happens because the timbral information has not been included in the initial comparison. Such is true in the case where two sounds that are played with the same note and intensity, but one is played on the guitar and the other on the piano. When the intensity values of the two sounds are compared, they would be the same, but when features that represent the timbral properties of a sound such as spectral centroid are also included in the comparison, their values would most likely be different.

In such situations, two most common solutions are practised in existing CSS systems: (1) to select the source segment that appears on the top of the list; and (2) to randomly select any of the segments that have the same sonic information. The former solution presents noticeable weaknesses, the most obvious being the tendency to select only the first matching source segment that appears in the list of possible solutions, disregarding other equally qualified segments. Since the list is typically arranged alphabetically, source

segments represented with the filename that begins with letters that are further down the alphabetical order are almost never selected, unless a 'taboo list' function or selection without replacement is enabled. The flaw is even more intensified when there are several segments in the target segment that occur more than once, which can give way to a very tediously repetitive sound. The latter solution reduces the chances of re-selecting the first line of segments in the list of matching units, but the randomness of this process suggests that there is very little intelligence or reasoning behind the selection.

Another challenge that stems from a similar situation is the occurrence of 'equidistant' segments in the returned list of matching segments. In contrast to homosonic segments, equidistant segments occur when there is no exact match found in the database, but several source segments with same distance from the target segment are present (Figure 22). Again, there is the issue of which segments should be selected from the list resurfaces. Selecting the first segment on the list or random selection will both result in the previously described flaws. Thus, a more intelligent solution to overcome unit selection issues involving homosonic and equidistant segments in existing CSS systems is needed.

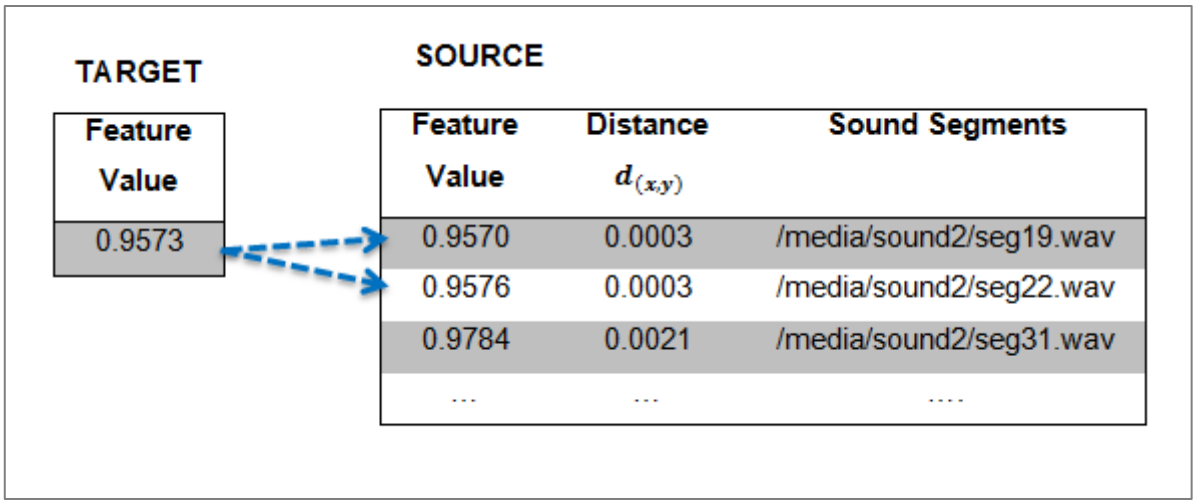


Figure 22: Unit selection involving equidistant segments

3.2.3 Basis of Sound Similarity

Previous issues that have been discussed in this chapter have all been concerning the technical aspects of existing CSS systems. However, that final issue that needs to be brought into attention is a rather subjective, but it is a crucially fundamental matter to the question of 'what makes humans perceive two sounds as similar?' The technical issues may have undergone many improvements, but unless the above question is answered, CSS systems may be generating sounds that are far from the expectation of its users.

A more visual example can be seen in determining image similarity. Figure 23 consists of a target image which is a picture of a centrally-located red circle. Of the three images: (a) a centrally located green circle, (b) a centrally located red square and (c) a red circle situated on the bottom left corner, which would be considered the image that has the closest match to the target image? Researchers in the field of image similarity have generally agreed that there are four major low-level attributes that influence this, which are colour, texture, shape and spatial constraint (Gudivada and Raghavan, 1995; Chen *et al.*, 2000; Laaksonen *et al.*, 2000).

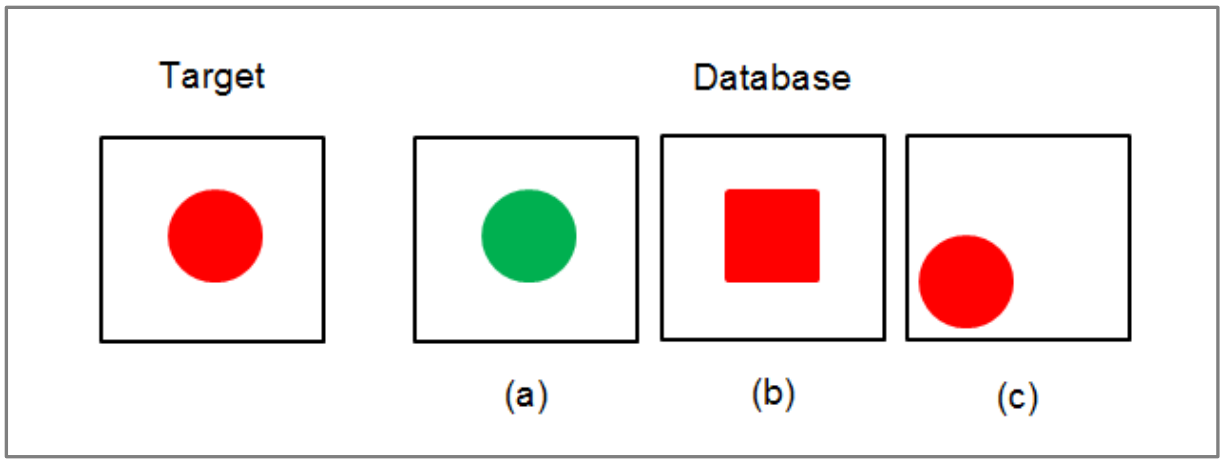


Figure 23: Presenting the issue with basis of similarity in image - which image in the database has the closest similarity to the target?

Similarly, if a target sound is of an A4 note played on a piano, which of the two segments that are available in the database (an A4 note played on a string instrument or a C4 note played on a piano) will be considered as most similar to the target sound? Which attribute does human find to be more dominant than others (if any)? As is the case with determining image similarity, there are different attributes that can become the basis of sound similarity, the basics being elements such as pitch, rhythm, tempo, timbre and loudness. Moreover, combinations of these elements then give rise to higher-order concepts such as meter, key, melody and harmony (Levitin, 2006; Mitrovic *et al*., 2010). Identifying the perceptual audio attributes that influence sound similarity in humans may reveal the audio feature sets that are more likely to extract relevant information from sounds, which can possibly return perceptually closer matching segments from the database. Determining which audio attributes are more dominant maybe the key to improving similarity in sounds generated by CSS systems.

3.3 Summary

Ten state-of-the-art CSS systems were reviewed in this chapter. In addition to brief introductions of each of the systems, their strengths and weaknesses were also compared, namely in the aspects involving their input corpus format, segmentation modes, features selection, search methods, use of concatenation distance and transformation, and also real-time capability.

The second half of the chapter then explored the issues that are still present in these systems, notably the need for an order-dependent feature selection process, a mechanism to handle homosonic and equidistant segments during unit selection and also the importance in determining the dominant perceptual audio features with regards to sound similarity. A simple case was demonstrated for each of these issues to highlight their problems and the significance in solving them was also emphasised.

A complete framework that provides solutions to overcome the aforementioned problems will be disclosed in the following chapter, Chapter 4 – Concatenative Sound Synthesis System: The Framework.

Chapter 4: Query-based Concatenative Sound Synthesis System: The Framework

This chapter aims to address the issues that are present in existing CSS systems, as discussed in the previous chapter. Solutions are proposed namely for: (1) an order-dependent feature selection, (2) the handling of homosonic and equidistant sound segments during unit selection and (3) identifying the audio feature sets that represent the dominant perceptual attributes applied where sound similarity is concerned. A new framework for CSS that incorporates solutions to the discussed issues system is then proposed.

4.1 Analytic Hierarchy Process as a Solution for Order-Dependent Feature Selection

Chapter 3 (Section 3.2.1, p.71) had previously presented the challenges when no reliable weight mechanism is implemented during feature selection. The challenges mentioned are the inability to specify order of importance between relevant features, ambiguous selection of sound segments, and arbitrary and inconsistent derivation of weight when manual assignment is attempted. A novel weight-applying mechanism to indicate the different level of importance between the features with a high level of consistency is thus proposed in this study, through the use of Analysis Hierarchy Process (AHP).

The AHP, a structured technique developed for dealing with complex decision has been proposed by Thomas Saaty (Saaty, 1977; 1983; 1994; 2008), and is one of the most well-known and widely approach used in multi-criteria analysis. AHP is intended to assist people to organise their thoughts and judgments so that more effective discussions via objective mathematical process can be made, whilst including the inescapably subjective and personal

preference of individual in making decisions (Saaty and Vargas, 2001). It takes elicit human judgments that reflect ideas, perceptions, feelings and memories (e.g. preferences of items that could be placed order such as high, medium, low); represents those judgments into meaningful numbers and then uses pairwise comparison to integrate the different measures that stemmed from the judgments into a single overall score, through which the results, given in a rank order, are synthesised.

An example of the use of AHP in everyday life is when purchasing a car. Typically, criteria that are taken into consideration by potential car buyers are: cost, safety rating, fuel consumption and appearance. For each criterion, buyers can compare any two car models and make an elicit judgment between them, for instance they may ask questions such as is the overall cost cheap or expensive? Is the safety rating poor, good or excellent? Is the fuel consumption low, medium or high? Is the appearance dull, sleek or sporty? Purchases that are made after undergoing these criteria comparisons are said to have gone through a process called multi-criteria decision analysis.

AHP is one of the most popular techniques used to solve many decision-making tasks that involve complex multi-criteria analysis. Its usefulness is evidently reflected in the vast number of applications developed using this technique to solve a broad spectrum of real-world problems, most commonly in problems including planning (Poh and Ang 1999; Chen, 2006), priority setting (Falconi, 1999; Salo and Liesio, 2006; Alwaer and Clements-Croome, 2010), forecasting (Finan and Macnamara, 2001; Chen and Chen, 2009) and business process re-engineering (Ashayeri *et al*., 1998; Rostamy *et al.*, 2012).

4.1.1 The Methodology of the Analysis Hierarchy Process

The basic steps that are involved in the AHP are as follows:

i) Decomposing the decision problem and selection of criteria

The first step is to decompose and identify the criteria that are important to a particular decision problem. This results in a hierarchy model, where the top most level is comprised of the goal or focus of the problem, followed by the criteria and sub-criteria (if applicable) at the intermediate level, and finally having the results or options presented at the lowest level. Hierarchy gives better understanding of the problem and the context of what is involved in the model. Once the criteria are established, pairwise set can be fixed. Figure 24 shows an example of this hierarchy, where it is first acknowledged that there are three levels to this problem: Level 0 (Goal), Level 1 (Criteria), and Level 2 (Options). In Level 1, the problem is constituted by two criteria which are Factor A and Factor B, whereas Level 2 includes three different options: Choice X, Choice Y and Choice Z.

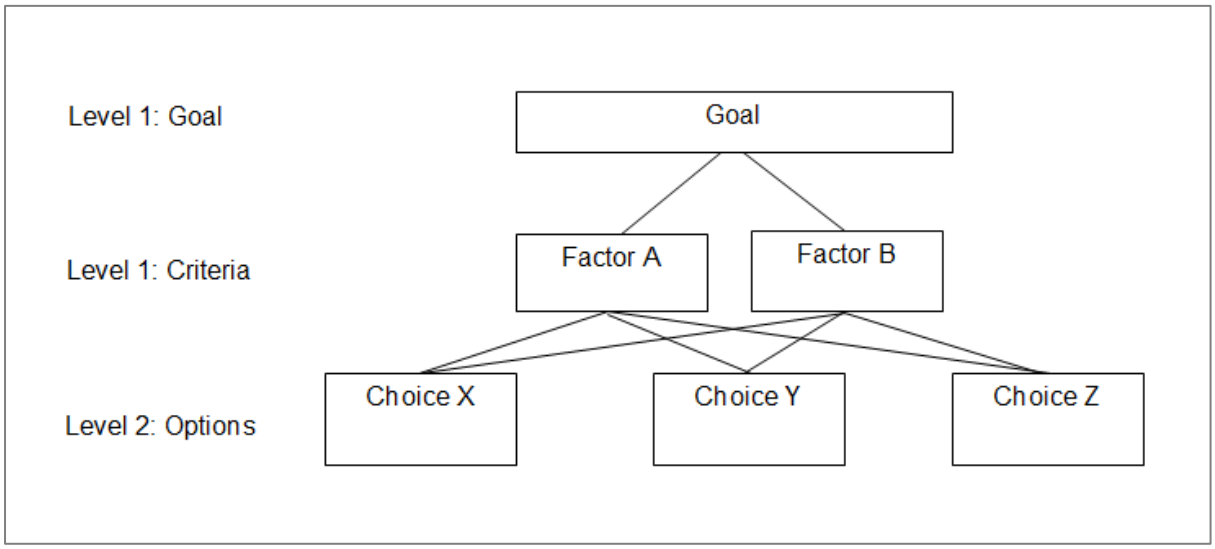


Figure 24: Example of AHP hierarchy

ii) Priority setting of criteria by pairwise comparison (weighing)

Once all the criteria that are important to a particular decision problem are identified, the relative priority of these criteria needs be determined. This is done through a method that is referred as pairwise comparison. Pairwise comparison, as its name suggests, presents each criteria in pairs and a judge[11] must state which one of the two criteria is preferred. Simplified, pairwise comparison aims to obtain an answer to the question, "How important is criterion *A* relative to criterion *B*?" This results in answers that can be articulated in expressions such as "*A*>*B*" (criterion *A* is preferred over *B*), or "*B*>*A*" (criterion *B* is preferred over *A*), or "*A* = *B*" (both criteria are indifferent).

Although the example presented here only involves 2 criteria, the number of comparisons that can be carried out by the pairwise comparison method is by no means restricted. However, the number of comparison does grow larger as the number of criteria involved increases, as given in equation (3) below.

$$Number\ of\ Comparison = \frac{N(N-1)}{2} \tag{3}$$

To derive qualitative values from the verbal comparison, a fundamental scale of importance is utilised (e.g. low, medium, high), typically presented in a 9-point scale (Table 3), where 1 represents equal importance and 9 represents extreme importance. This weight is assigned to the more important criterion, and the reciprocal of this value is assigned to the other criterion in the pair. The use of this scale gives answer to the next question that follows, which is "How much more important is one criteria over the other?"

[11] The word 'judge' here refers to an expert user, or in the case of a CSS system, the composer

It must be noted that comparisons of elements in pairs require that they are homogeneous or close with respect to the common attribute, otherwise significant errors may be introduced into the process of measurement (Saaty, 1990).

Table 3: The fundamental scale of absolute numbers (Saaty, 2008)

Intensity of Importance	*Definition*	*Explanation*
1	Equal Importance	Two criteria contribute equally to the objective
3	Moderate Importance	Experience and judgment slightly favours one criteria over the other
5	Strong Importance	Experience and judgment strongly favours one criteria over the other
7	Very Strong Importance	A criteria is favoured very strongly over another; its dominance demonstrated in practice
9	Extreme Importance	The evidence favouring one criteria is of the highest possible affirmation
2,4,6,8	Weak, Moderate plus, Strong plus, Very Strong Plus Importance respectively	For compromises between the above
Reciprocals of above	If activity i has one of the above non-zero numbers assigned to it when compared with activity *j*, then *j* has the reciprocal value when compared with *i*	A reasonable assumption
1.1 – 1.9	If the activity are really close	May be difficult to assign the best value but when compared with other contrasting criteria, the size of the small numbers would not be too noticeable, yet they can still indicate the relative importance of the activities

iii) Calculating the priority value and Eigenvalue from the pairwise comparison (scoring)

The values from the pairwise comparison are used to tabulate the pairwise comparison matrix, *A*. The numbers (a_{ij}) in the *i*th row and *j*th column represent the relative importance, or the weight, *W*, of the first criterion, O_i as compared with the second criterion, O_j. Another form which this expression can be visualised is in the more elaborated but familiar matrix form as seen in Figure 25.

To get the eigenvector of matrix *A*, the sum of each column in the matrix is calculated. Each element of the matrix is then divided with the sum of its own column, giving the normalised relative weight. Hence, the sum of each column is assumed to be '1'. The average of all cells in a row of matrix *A* is then summed up to get the normalised principle eigenvector. This gives the priority vector, *W*, or the weight of the criteria (Figure 26).

$$A = [a_{ij}] = \begin{bmatrix} W_1/W_1 & W_1/W_2 & \cdots & W_1/W_n \\ W_2/W_1 & W_2/W_2 & \cdots & W_2/W_n \\ \vdots & \vdots & \vdots & \vdots \\ W_n/W_1 & W_n/W_2 & \cdots & W_n/W_n \end{bmatrix}$$

$$\text{where } a_{ij} = W_i/W_j,\ a_{ji} = 1/a_{ij},\ i,j = 1,2,\ldots,n$$

Figure 25: Pairwise comparison matrix

$$W = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_n \end{bmatrix},\ i,j = 1,2,\ldots,n$$

Figure 26: Priority vector

Writing this matrix out as a system of equations gives equation (4).

$$\sum_{j=1}^{n} a_{ij} w_j = \lambda_{\max} w_i \; i = 1,\ldots,n \qquad (4)$$

subject to $a_{ji} = 1/a_{ij}$ (or simply $a_{ij}a_{ji} = 1$) known as the reciprocal condition resulting from the stronger consistency condition $a_{ij}a_{jk} = a_{ik}, i, j, k = 1,\ldots,n$, and the normalisation condition $\sum_{i=1}^{n} w_i = 1$.

This pairwise comparison and the calculations in Steps 2 and 3 are then carried out for every level in AHP. In this example, since size of pairwise comparison in Level 1 is 2 x 2, and in Level 2 is 3 x 3, there are six comparisons in total. Table 4(a), Table 4(b) and Table 4(c) respectively show the paired comparison matrices in Level 1, Level 2 and Level 3 from the earlier example.

Table 4(a): Comparison matrix Level 1 of the influence factors

	A	**B**	**Priority Vector (Weight)**
A	1	7	87.61%
B	1/7	1	12.39%

λ_{max} = 2.000, Consistency Index = 0.000, Consistency Ratio = undefined

Table 4(b): Comparison matrix Level 2 with respect to Factor A

	X	**Y**	**Z**	**Priority Vector (Weight)**
X	1	1	7	51.05%
Y	1	1	3	38.93%
Z	1/7	1/3	1	10.01%

λ_{max} = 3.104, Consistency Index = 0.050, Consistency Ratio = 8.97% < 10%

Table 4(c): Comparison matrix Level 2 with respect to Factor B

	X	**Y**	**Z**	**Priority Vector (Weight)**
X	1	3	5	63.33%
Y	1/3	1	3	26.05%
Z	1/5	1/3	1	10.62%

λ_{max} = 3.055, Consistency Index = 0.277, Consistency Ratio = 4.77% < 10%

iv) Obtaining overall relative score for each option

The weights for each criterion that were calculated in the previous step are now combined with the option scores to produce an overall score for each option. Judgments are made based on this overall score, where the scores represent the impact of all the elements and priorities that have been computed as a whole. In this example, Choice X appears to be the best solution. Table 5 presents the overall relative scores generated from this example. Figure 27 presents this example visually.

Table 5: Overall composite weights for the options

	Factor A	**Factor B**	**Composite Weight (*2 d.p.*)**
Choice X	0.5105	0.6333	**0.53**
Choice Y	0.3893	0.2605	**0.37**
Choice Z	0.1001	0.1062	**0.10**

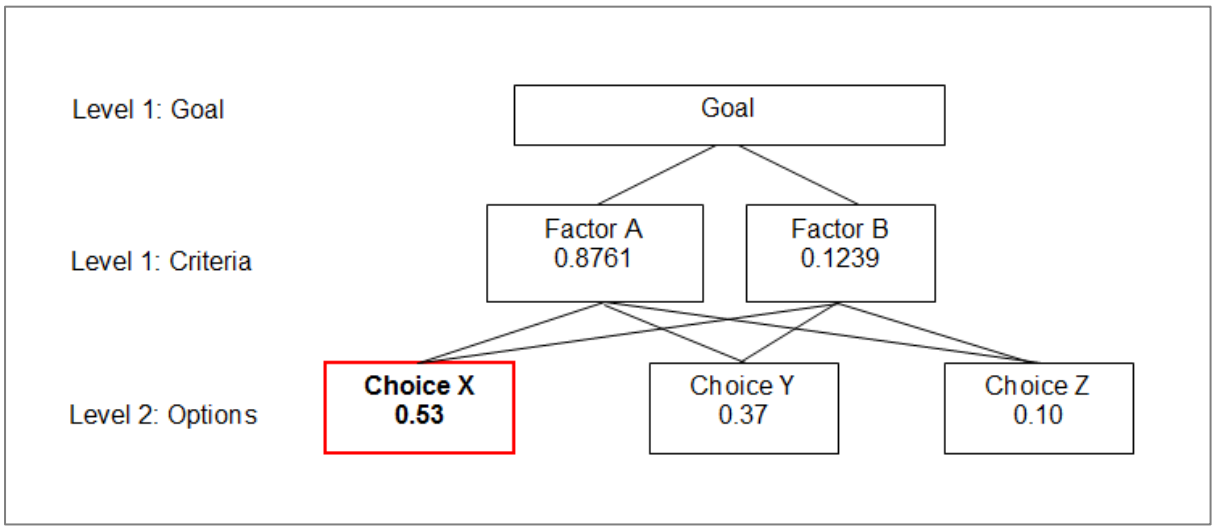


Figure 27: Visual representation of the overall relative score from the earlier worked example

v) Verifying the consistency of the result synthesised

There is always a chance that a cardinal inconsistency or an intransitivity inconsistency might occur in judging. As such, it is necessary to verify the consistency of the pairwise

comparison. First, the measure of consistency, or the Consistency Index (CI) is calculated using the formula in equation (5),

$$CI = \frac{(\lambda_{max} - n)}{n-1} \quad (5)$$

where λ_{max} is the largest Eigenvalue, and *n* is the size of the comparison matrix.

The index is then compared to the Random Consistency Index (RI) to determine whether it is approximately 10% or less. Table 6 below shows the average RI of sample 500 matrices.

Table 6: Random Consistency Index

N	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.59

Finally, the Consistency Ratio (CR) can be calculated to measure the coherence of the pairwise comparison using equation (6) below.

$$CR = \frac{CI}{RI} \quad (6)$$

In general, if the value for the CR is smaller than or equal to 10%, the evaluation is deemed acceptable as mathematically there is always a need to allow for a small inconsistency in measurement. Variation of no more than 10% within the elements is tolerated as it normally does not destroy the identity of the elements compared. An inconsistency value that is larger than this may suggest that the judgment made by the referee is biased or slightly inconsistent, at which a re-judgment is required to avoid the matrix to be rejected.

4.1.2 Analysis Hierarchy Process and Order dependent Feature Selection

Robustness and consistent scoring are two of the strongest reasons that make AHP a suitable approach to be implemented in the development of a CSS system that has order-dependent feature selection process. The same methodology as described in the previous section is followed through, and the problem at hand is first decomposed into its hierarchical components. However, since a CSS system with order-dependent feature selection process still relies on the target distance between the target segment and the source segments in the database in order to select the closest matching segments and not solely relying on the scores generated from human judgment (as typical AHP-based evaluations do), only a partial of the AHP component is included, i.e. the goal (Level 0) and the criteria (Level 1). These two levels are already sufficient to calculate the weights for criteria, which will then be inserted into equation (2) to obtain the weighted target distance (refer Chapter 3, p.74). Figure 28 shows the hierarchical model of an order-dependent feature selection CSS system, where the goal is to find matching segments in the database where the criteria are the audio features such as Centroid, ZCR and Pitch.

The order of importance between the three features is then set using the previously presented Fundamental Scale of Importance (Table 8). Assuming that the centroid is regarded as moderately more important than ZCR, and extremely more important than pitch, and the feature ZCR is strongly more important than pitch, an acceptable representation of the reciprocal matrix for this case is pictured in Table 7.

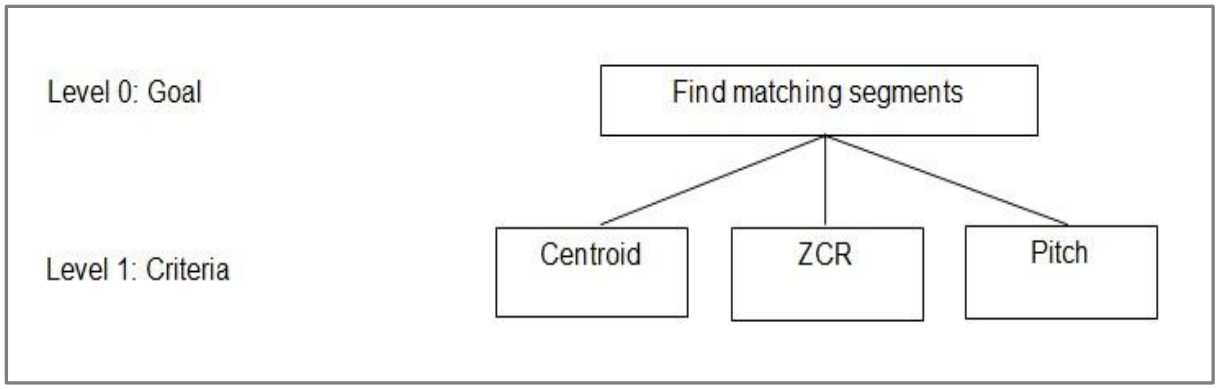


Figure 28: An AHP hierarchy for an order-dependent feature selection in CSS, between features Centroid, ZCR and Pitch

Table 7: The reciprocal matrix between features Centroid, ZCR and Pitch, where the Centroid is moderately more important than ZCR and extremely more important than Pitch, and ZCR is strongly more important than Pitch

Features	Centroid	ZCR	Pitch
Centroid	1	3	9
ZCR	1/3	1	5
Pitch	1/9	1/5	1

From the reciprocal matrix, the normalised eigenvector, also referred as the priority vector can then be computed. There are several steps involved in obtaining this vector, first of which is summing up each columns in this 3 x 3 reciprocal matrix, as shown in Table 8. Following this, each element of the matrix is divided with the sum of its own column. It should be noted the sum of each column on this step should return the value of '1' (Table 9).

Finally, the normalised principal eigenvector, or the weight (*W*) can now be obtained by averaging across the rows (Figure 29). The new weights of the three features are shown in Table 10. Since this calculation has been normalised, the sum of all the elements must again, be equal to 1.

Table 8: Sum of each column in the reciprocal matrix

Features	Centroid	ZCR	Pitch
Centroid	1	3	9
ZCR	1/3	1	5
Pitch	1/9	1/5	1
Sum	**13/9**	**21/5**	**15**

Table 9: Dividing each element in the matrix with the sum of each column

Features	Centroid	ZCR	Pitch
Centroid	9/13	15/21	9/15
ZCR	3/13	5/21	5/15
Pitch	1/13	1/21	1/15
Sum	**1**	**1**	**1**

$$W = 1/3 \begin{bmatrix} {}^{9}/_{13} + {}^{15}/_{21} + {}^{9}/_{15} \\ {}^{3}/_{13} + {}^{5}/_{21} + {}^{5}/_{15} \\ {}^{1}/_{13} + {}^{1}/_{21} + {}^{1}/_{15} \end{bmatrix} = \begin{bmatrix} 0.6689 \\ 0.2674 \\ 0.0637 \end{bmatrix}$$

Figure 29: Calculating the normalised principal eigenvector gives the weights, *W*, of each feature

Table 10: The new weights for features Centroid, ZCR and Pitch

Features	Centroid	ZCR	Pitch	Weights
Centroid	9/13	15/21	9/15	**0.6689**
ZCR	3/13	5/21	5/15	**0.2674**
Pitch	1/13	1/21	1/15	**0.0637**
Sum	**1**	**1**	**1**	**1.0000**

The priority vector shows the relative weights amongst the criteria, i.e. Centroid is 66.89%, ZCR is 26.74% and Pitch is 6.37%. These weights are then applied to the weighted Euclidean distance, i.e. equation (2) (Chapter 3, p.74) which is used to find the sound segment in the database which closely matches the target segment. In addition to establishing the rank or order of importance between the three features, AHP also reveals their ratio scale. For instance, this example, Centroid is found to be 2.5 times more important than ZCR, and 10.5 times more important than Pitch. Figure 30 below shows the generated weights for the three criteria using AHP.

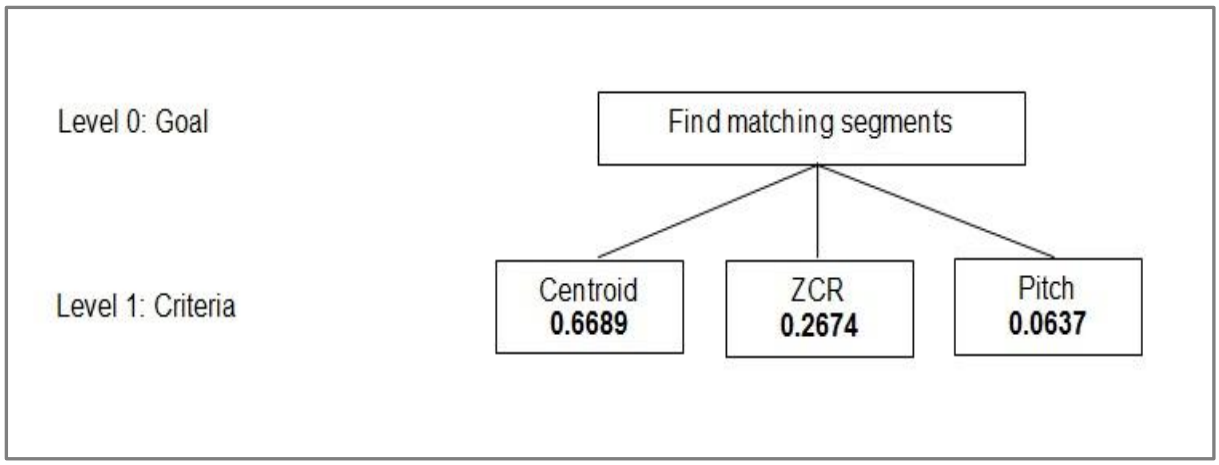


Figure 30: Weights generated through the use of AHP for features Centroid, ZCR and Pitch

The consistency of the weights calculated from the judgment made between the three features can also be determined through equations (5) and (6) as described in the past section (Section 4.1.1, p.92). The value of λ_{max} that is required in this calculation can be found by obtaining the summary of products between each elements of eigenvector and the sum of the columns of the reciprocal matrix.

$$\lambda_{max} = 13/9\ (0.6689) + 21/5\ (0.2674) + 15\ (0.0637)$$

$$= 3.045$$

This value can then be used to calculate the Consistency Index (equation (5)). In this particular example that is being followed, the size of comparison matrix, *n*, is equal to 3.

$$CI = (3.045 - 3) / (3 - 1) = 0.0225$$

To finish, the Consistency Ratio is calculated following the formula given in equation (6) and the Random Consistency Index in Table 6 (both mentioned in earlier in p.92).

$$CR = (0.0265 / 0.58) = 0.0388 = 3.9\%$$

Since 3.9% < 10%, it can be concluded that the subjective evaluation placed earlier to distinguish the order of the features is consistent.

Following the example above, Table 11 shows that order dependent feature selection via AHP can generate different results than when feature selection without priority setting is selected. The original values for features Centroid, ZCR and Pitch between the target segment and the five source segments in the database are also given. The highlighted rows show the vector distances (d_t), calculated with and without the use of AHP. The final results of the feature selection between the two approaches are underlined where Segment 5 for feature selection with no priority, and Segment 1 feature selection with priority. Evidently, these two segments will generate two different sounds, as can be observed in Appendix A2.

Table 11: Comparison of vector distances between basic feature selection (no priority) and order dependent feature selection (with priority)

Features	Target	Segment 1	Segment 2	Segment 3	Segment 4	Segment 5
Centroid	0.9835	0.9865	0.9970	0.9014	0.9909	0.8118
ZCR	0.0435	0.0870	0.0435	0.9130	0.1141	0.0635
Pitch	0.2444	0.5827	0.6320	0.2567	0.8644	0.2504
(d_t) No priority	**---**	**0.341**	**0.388**	**0.873**	**0.624**	**0.173**
(d_t) With Priority	**---**	**0.088**	**0.098**	**0.455**	**0.161**	**0.141**

The example in Table 11 above demonstrates a very small snapshot of the effect that order dependent features selection has in the segment matching stage of a CSS system. In real life applications, a composer may prioritise one audio feature over the other for different reasons. For instance, a composer may want to synthesise sounds that are bright and cheery. He therefore selected features centroid and pitch to be extracted and used for searching similar sounds, as they are good indicators of the said sound characteristics, i.e. higher pitch tends to give off happier, brighter feel to a sound, as does sound with higher centroid (Collier and Hubbard, 1998; Schubert *et al*., 2004). However, the composer may not want his synthesised sounds to consist of only high pitched segments and instead, he prefers some pitch fluctuations to give some sense of melodic contour to his composition. This means that although he wants to include both features in the search, he is less concerned that the pitch information is not matched as closely, as long as the centroid information is. With order dependent feature extraction, this is easily implemented as the composer can place less weight on the pitch in comparison to the centroid. Furthermore, by applying AHP, the composer can explicitly state the level of importance of the features in relation to one another, e.g. 80% more important, twice as important, and so on. In short, he can set the search conditions as 'find a segment that are close to both the centroid and

pitch values, but it is more important to find a segment that matches the centroid value first than it is the pitch value by *X* percent'. Sound outputs synthesised using the two approaches (order dependent features selection versus no priority) can be compared in Appendix A3.

4.1.3 Strengths and Weaknesses of the Analysis Hierarchy Process

AHP has been shown to be a very effective method to reliably assign weights to many criteria from human judgments that are known to be very subjective. There are several other reasons that further strengthen the decision to embed AHP into various tasks that require prioritising, such as the feature selection process in basic CSS systems; and these are listed as follows:

i) *AHP provides a solution for tasks that require prioritising option over multiple decision factors*

AHP enables different weights to be placed for each individual criterion, a feat that is very useful, but not always made available by other decision making approaches.

ii) AHP method is flexible and convenient

The AHP method is intuitive-based and is therefore more flexible compared to other multi-criteria methods (Ramanathan, 2001).

iii) AHP is clear and systematic

Through AHP, before judgments relating to these parts can be applied, the problem is described, its goal determined and the relations between all parts defined. The

decomposition of a decision problem into a hierarchy of its constituent parts results in a clearer view of all the elements involved (Macharis *et al.*, 2004).

iv) *AHP is capable of capturing both subjective judgment and objective evaluation measures*

AHP converts complex human judgments into a more accurate, reliable and mathematically proven score, which increases the understanding and confidence of a selection (Saaty, 2008).

v) AHP scores are consistent and reliable

AHP ensures the consistency of the evaluation measures, thus reducing the bias that may normally exist in most decision making process.

However, the AHP method has received a few criticisms too, which are listed as follows:

i) The issue of rank reversal

Rank reversal is a common problem in many multi-criteria decision making approaches. Although rank reversal can be the result of many different situations, the most is through the addition or deletion of new input (i.e. options), or influential factors (i.e. criteria), causing the result of the new score to be different or reverse of the original score. Another situation where rank reversal happens is when it is found that the overall result obtained through the aggregation of the score is dissimilar to (or the reverse of) the overall result obtained through the aggregation of rank. Fortunately, the proposed use of AHP method into the CSS system involves only the generation of weights and not the use of ranks, thus avoiding this issue altogether.

Nevertheless, even with the issue of rank reversal, AHP is still considered by many as the most reliable multi-criteria decision making method.

ii) Artificial limitation on judgment

Some form of information loss is expected when verbal comparison is converted into numerical gradation. By further forcing users to make comparisons using the 9-point scale from the Fundamental Scale of Importance puts a limit on the judgment that can be given. Also, it may be difficult to judge how many times a criterion is more important over another criterion. This confusion may also make the whole process a time consuming task to perform. A solution has been proposed to resolve this problem by replacing the 9-point scale with a 2-point scale that judges whether a criterion is either more or less important than, or equally important as another criterion to reduce comparison time and confusion (Hajkowicz *et al.*, 2000).

iii) Inconsistent judgment as a result of human error

Sometimes human input can introduce error by means of passing inconsistent judgments, particularly evaluations that are results of intransitivity or indifference. However, AHP has ensured that the validity of the scores can be tested through the calculation of the Consistency Ratio.

iv) Number of pairwise comparisons needed

For every level that exists in an AHP hierarchy, the number of pairwise comparison also increases. The disadvantage of this is that the number of comparison may get too large and the process too lengthy. Fortunately, in the case of the proposed order

dependent feature selection in CSS system, a maximum of two levels is all that is required, thus keeping the number of comparison to a manageable size.

Despite these shortcomings, AHP is still a widely accepted multi-criteria decision method both academically and commercially. In the academic community, the use of AHP has been merged with other methods including neural networks, fuzzy set and genetic algorithm (Ho *et al*., 2010). The practicality of AHP is made apparent in the development of decision support software tool such as Criterium, Decision Lab, Expert Choice, RightChoiceDSS and WebAHP. The success of AHP is possibly owing to the fact that the method has managed to reach a compromise between being the perfect model and a usable model.

4.2 Concatenation Distance as a Measure to Solve Homosonic and Equidistant Segments

Chapter 3 had previously described and demonstrated the shortcoming in existing CSS systems concerning homosonic and equidistant segments that may exist in the source sound database and the problem it inflicts during unit selection. This study proposes to solve this problem by manipulating a measure referred as the concatenation distance.

4.2.1 Outlining Concatenation Distance

Concatenation distance is another 'cost' that is sometimes measured in addition to the target distance during the unit selection stage in concatenative sound synthesis. Whilst the target distance measures the similarity or closeness between the target unit and the source unit in the database, the concatenation distance measures the quality of the join between two consecutive units. This is why the concatenation distance is interchangeably referred as the join cost. The relationship between the target distance and the concatenation distance is illustrated in Figure 31.

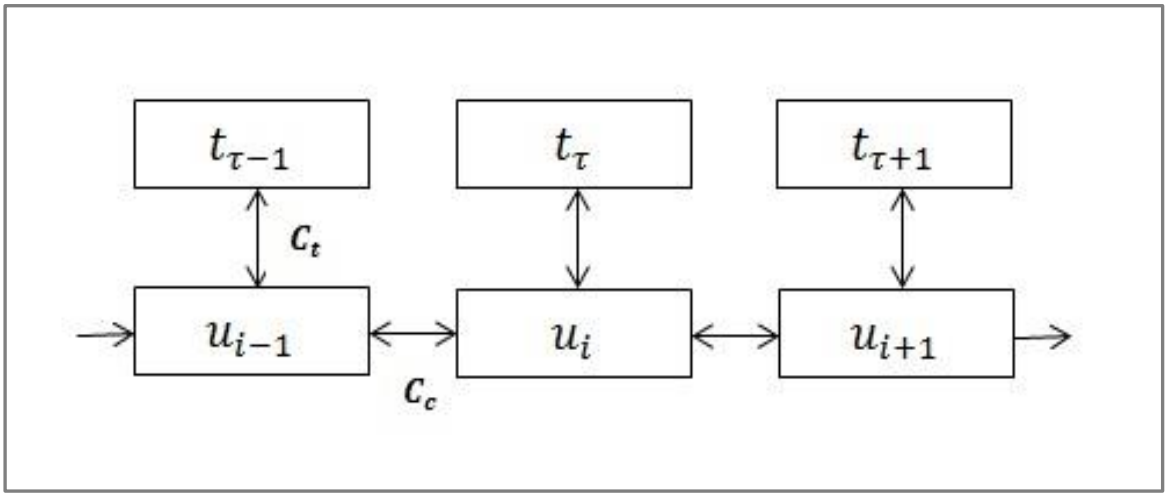


Figure 31: The relationship between Target Cost (C_t) and Concatenation Cost (C_c)

Where the target distance, C_t compares the feature value between the target segment ($t_{\tau-1}$) and the source segment (u_{i-1}), the concatenation distance, $C_C$compares the feature value at the beginning of a current segment (u_i) with the feature value at the end of a preceding segment (u_{i-1}), a working example of which is presented more clearly in Figure 32 below. If there are more than one audio features involved in the comparison, weights may be assigned to each feature. Rationally, if u_{i-1} and u_i are consecutive units in the source sound database, then their concatenation cost is equal to zero.

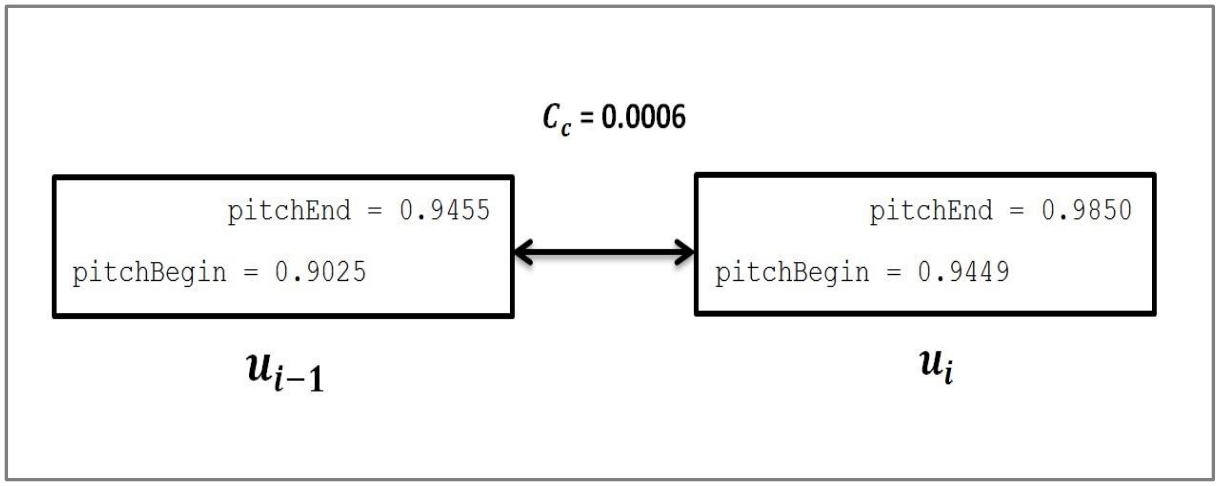


Figure 32: Comparing the feature value at the beginning of a current segment (u_i) with the feature value at the end of a preceding segment (u_{i-1}) to obtain the Concatenation Cost (C_c)

The concatenation cost, C_c can be calculated as follows:

$$C_c(u_{i-1},u_i) = \sum_{j=1}^{q} w_j^c \, C_j^c \, (u_{i-1},u_i) \qquad (7)$$

where i is the current unit, $w_j^c$is the weighted sum of concatenative sub-costs (if and when applicable), which is denoted by C_j^c (u_{i-1},u_i), (j=1,…,q), where q is the number sub-costs included, (i.e. if at the point of concatenation, the pitch and cepstral distance are used, then q = 2). The formula to calculate the target distance, i.e. equation (2), can be revisited in Chapter 2, p. 74.

Concatenation distance is first seen used as a measure to reduce segmental mismatch in concatenative speech synthesis that tends to occur at unit boundaries (Pantazis *et al*., 2005). By selecting adjoining segment with the least distance from the previous segment, the naturalness of the utterance is enhanced, as seen implemented in several well-known concatenative speech synthesis systems such as CHATR (Hunt and Black, 1996).

The same concept is adapted in several CSS systems, notably *Caterpillar* and *MusicalMosaic*. The use of concatenation distance in general is intended to reduce discontinuity between two adjoining segments, ensuring that the sounds are generated with a smoother flow, although there are some cases where this general rule is overridden, for instance, *Caterpillar* has an added function where it allows certain discontinuity during an attack and not during a sustain unit (Schwarz, 2004).

Regardless, the use of concatenation distance in existing CSS systems is primarily limited to increasing the continuity between two segments that are to be joined together. Excitingly, this study has found that the use of concatenation distance can also be extended to overcome the challenges faced when dealing with homosonic and equidistant segments during unit selection, which is currently dependent on random selection.

4.2.2 Concatenation Distance in Selection of Homosonic and Equidistant Segments

When there are two or more homosonic or equidistant segments present in the database that are equally suited to be returned as a match for the queried target segment, the information at the concatenation point can be used to further sift the segments in order of the concatenation importance. In this study, the pitch is used to make this comparison; hence the value of the pitch at the start of each homosonic or equidistant segments is

compared to the last value of pitch of the most recently concatenated segment in the chain. The homosonic or equidistant segment with the least concatenation distance is then selected.

Pitch is used over other audio features for this task in this study because the change (or rather, the lack of change) of pitch usually suggests continuity, or that two units are within the sustained phase. This is further supported by studies that track the melodic changes in music, where pitch is the major feature that indicates semantic continuity. Semantic continuity refers to the minimal change between successive time indices which is an indication of sustain. No large jump is expected in tracked melodies and this is a distinguishing characteristic of music (Pollastri, 1998; Cao *et al*., 2007; Smaragdis and Mysore, 2012). Furthermore, in the few CSS systems that take into account the concatenation cost, such as *Musical Mosaicing* (Zils and Pachet, 2001) and *CataRT* (Schwarz, 2004), pitch has been regarded as one of the more important continuity constraint parameters, making it an ideal choice of feature for this particular task.

Other audio features may also be used for this task, for instance, CHATR includes the cepstral distance, the absolute difference in log power and again, the pitch (Hunt and Black, 1996). However, the use of various features may mean that there is a need to assign weight for each feature, unnecessarily complicating the process. For this reason, only pitch is used in this study.

For this, the fundamental frequency (F_0) of a segment is extracted. Depending on the length of the audio segment, pitch extraction typically results in several pitch values due to fluctuations over time. The pitch values are normally averaged out and normalised against other feature values to obtain a single, global pitch value for the entire segment. However,

for this purpose, the pitch values are left in their raw form, with the first and last pitch values at each segment stored and used as basis of concatenative comparison.

To demonstrate the role of concatenation distance in solving unit selection problem involving homosonic and equidistant sound segments, consider the case presented in Figure 33. The task is to find a matching segment, u_i for the target segment, t_τ which currently has a feature value of `0.9835`. In the database, three segments have been identified to have the exact same feature value, which are `indis2.081.07828.wav`, `siamang.018.68045.wav` and `whales004.05188.wav` respectively. These segments may share the same feature value, but they sound somewhat different from one another, which is the root of the problem with homosonic segments (Appendix A4).

The typical solution adopted by existing CSS system when faced with this dilemma is to select the first segment found in the database (alphabetical arrangement is the norm), or by randomly selecting one between the three equally suitable segments. However, comparing the pitch value at the beginning of each homosonic segment in the database (`pitchBegin`) with the pitch value at the end of the most recently concatenated segment (`pitchEnd` at segment u_{i-1}) gives the concatenation distance C_c. Segment with the least distance suggests the highest possible continuation from the previous segment and is therefore selected.

Table 12 tabulates the concatenation distance between the three segments. It was found that the second segment (`siamang.018.68045.wav`) had the smallest concatenation distance, and in this instance, is selected. Previous approach would have selected the first segment (`indis2.081.07828.wav`) with no consideration over the other two segments present.

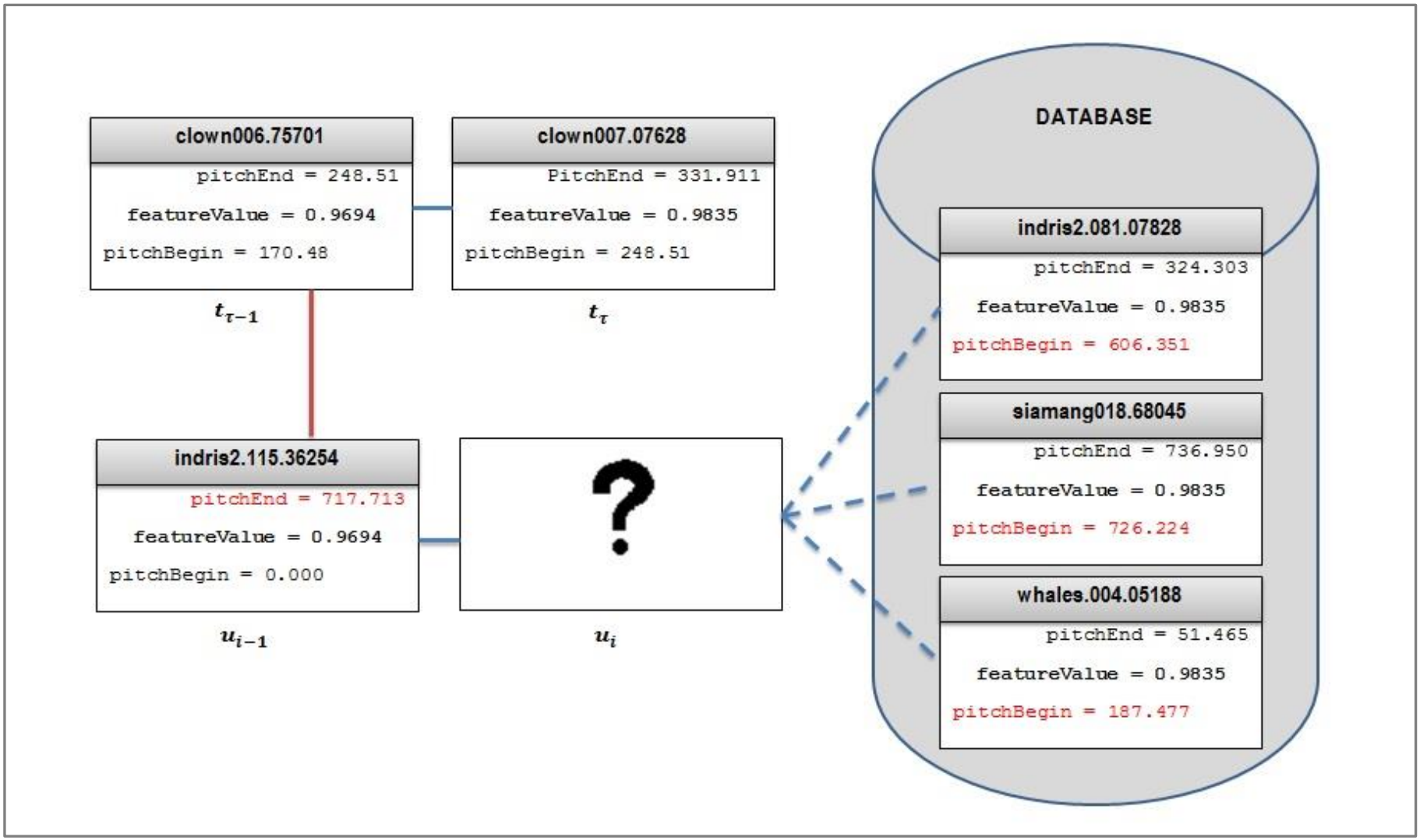


Figure 33: A demonstration of the role of the Concatenation Cost (C_c) in the selection over three homosonic segments in the database

Table 12: Comparison of concatenation distances between three different homosonic segments and how this affects unit selection

Feature Value	Target Distance (C_t)	PitchBegin Value	Concatenation distance (C_c)	Source Segment
0.9835	0.000	606.351	111.362	/media/indris2.081.07828
0.9835	0.000	726.224	**8.511**	/media/siamang.018.68045
0.9835	0.000	584.236	530.236	/media/whales.004.05188

Utilising the concatenation distance not only solves the problem with homosonic and equidistant segments in a more 'intelligent' approach, but it also increases the quality of concatenation by minimising the pitch gap between neighbouring segments. In addition, it can also avoid unintentional favouritism against several homosonic segments in the database, as would be the case if only the segment at the top of the list is selected every time.

It should be noted that whilst concatenation distance is no longer a novel concept and has been previously implemented on a few CSS systems before, the idea to use it to solve homosonic and equidistant segments during unit selection is a novelty. Furthermore, in other CSS systems, the concatenation cost is calculated together with the target cost to give the combined cost of selection (see equation (8)) before any selection is made. Equation (9) is the direct result of expanding equation (8) to include all sub-costs.

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C_t(t_i, u_i) + \sum_{i=2}^{n} C_c(u_{i-1}, u_i) \quad (8)$$

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} \sum_{j=1}^{p} w_j^t \, C_j^t \, (t_i, u_i) + \sum_{i=2}^{n} \sum_{j=1}^{q} w_j^c \, C_j^c \, (u_{i-1}, u_i) \quad (9)$$

In comparison, in this study, concatenation distance is implemented in a hierarchy, where it first identifies all the segments that have the least target distance, and the concatenation distance is only calculated when there are homosonic or equidistant segments present. The structural difference between the two models is shown in Figures 34 and 35.

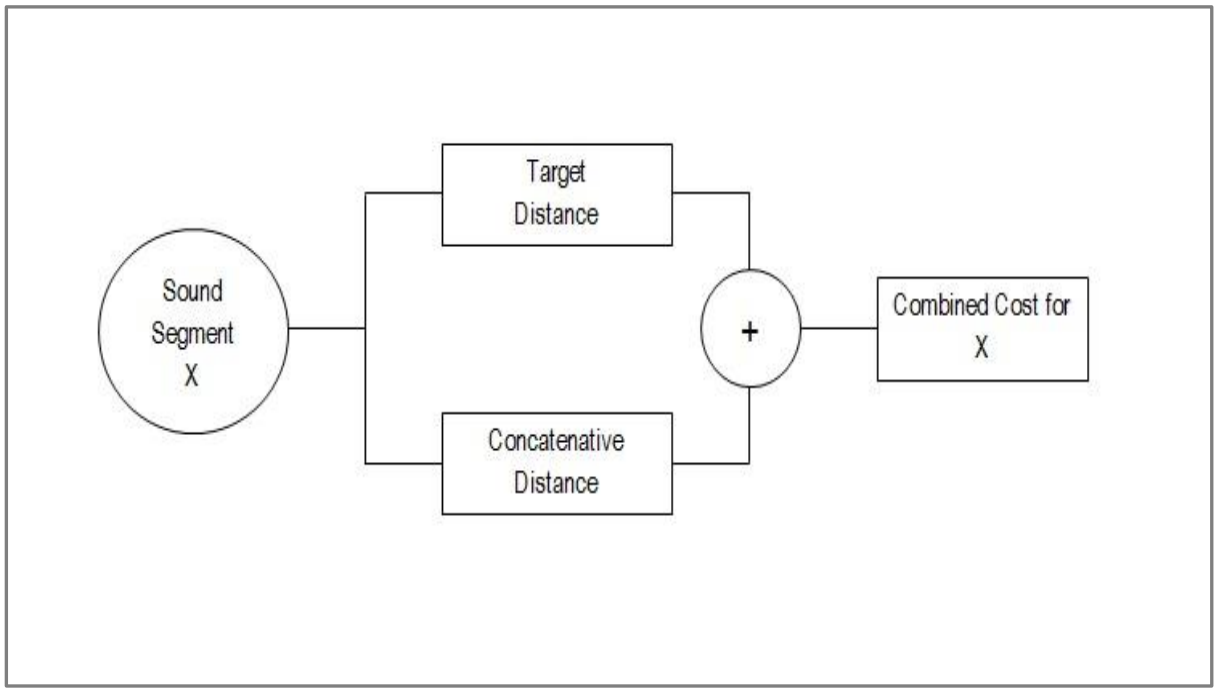


Figure 34: Non-hierarchical model implemented in existing CSS systems to determine the overall cost of segments

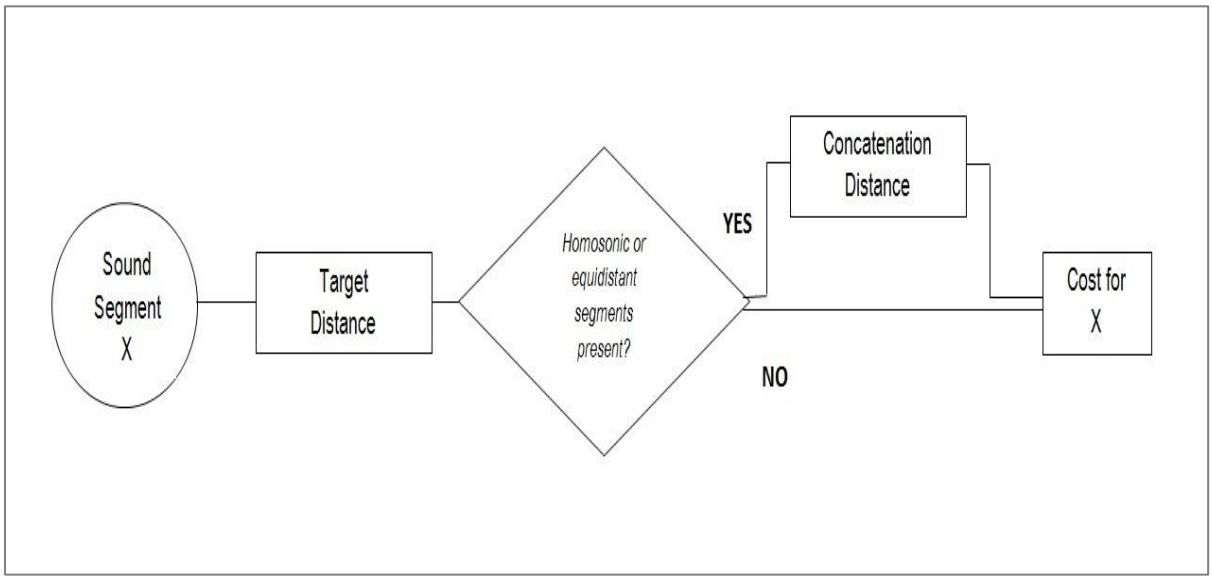


Figure 35: The newly implemented hierarchical model to determine the cost of segments

The hierarchical model is applied in this study for a number of reasons: first of all, it reduces the problem into smaller, manageable tasks. Secondly, instead of calculating the combined costs of both distances, the hierarchical approach only proceeds to calculate the concatenation distance when it is deemed necessary so as to cut down the occupation for an otherwise complex and time consuming process.

Although it is possible to argue that the 'best' unit is not necessarily be the one with the lowest target distance, but could lie in another candidate unit with a slightly larger target distance but smaller distortion, this situation appears more relevant in concatenative speech synthesis than it is in music, as the degradation in the naturalness of speech utterance can be very noticeable to human. Music, being a more subjective domain, is less affected by this. Nonetheless, whilst segment continuity is important in concatenative speech synthesis, it is still a challenge to ensure that finding the segment with the smallest concatenation distance does not happen at the expense of intelligibility (target distance). This suggests that target distance has a slight precedence over the concatenation distance, through which the hierarchical model is able to ensure that it remains so. Additionally, in situations where it is essential, continuity can be optionally remedied through linear smoothing or other transformation such as amplitude and pitch corrections, which is another reason why the hierarchical model is adopted.

4.3 Basis of Sound Similarity

In its simplest form, the physics of simple sound can be described as a function of frequency, amplitude and phase. Generally put, two sounds are similar if the values of these three criteria are the same. However, sounds very rarely exist in this simple form and usually the Fourier analysis is used to break down complex sounds into a series of simple sounds to achieve this. The psychology of sound, on the other hand, is based on the human perception of these criteria and also the time factor, giving rise to other sound elements such as pitch, intensity, timbre and rhythm.

Usually, human listeners have a well-developed feeling whether two songs sound similar or whether they do not (Allamanche *et al*., 2003). It is thus very important for any system that relies on finding similar sounds such as the CSS system to determine what these auditory characteristics are. Earlier works at the Muscle Fish research group have described the ways in which humans may describe similar sounds – simile, acoustical or perceptual features, subjective features and onomatopoeia; all of which have been used individually or in combination, as a query mechanism for many sound similarity-based multimedia applications such as audio classification, audio retrieval and audio search engine (Wold *et al.*, 1996).

An ideal CSS system would have the capability to tackle of all the variability above. Unfortunately, due to its extreme complexity (too many features to compute and extract, data too large to make analysis from, subjective nature of the topic), this level of perfection is yet to be accommodated. Nevertheless, two small-scaled studies have been designed and carried out to further understand how humans perceive sound similarity. The studies identify the dominant acoustic information on which judgements are based by humans

when performing a sound similarity task, and they also determine if humans are capable of displaying some form of agreement between them when the basis of sound similarity (perceptual attribute) is set or given, which is to find the sound which is most similar to the target in terms of their timbral quality. Results from these two small-scaled studies will ascertain the dominant attribute involved when humans perceive sounds to be similar and by applying this attribute into the CSS system, it is envisioned that the sounds generated will be able meet more of the users' expectation and satisfaction.

4.3.1 Determination of Dominant Perceptual Attribute

The objectives of this small-scaled study are threefold: (1) to identify the dominant perceptual attribute that humans base their judgment of sound similarity on, (2) to determine whether humans and computers differ in their judgment of sound similarity and (3) to observe whether there is a significant difference in the subjective judgments between musicians and non-musicians with regards to sound similarity.

The sound attributes that are included in this test, along with brief description for each of them are given as follow:

Melody – The melody is a sequence of notes of differing duration, or the linear succession of musical notes that gives the tune of a musical piece.

Timbre – The definition of timbre is very wide and ill-defined, but to simplify, it refers to the quality and texture of sound that distinguishes a voice or instrument from another. This includes information such as the relative brightness or brashness of a sound, which can also give clue to the mood of sounds (joyous or mellow). Timbre can also be synonymous to the tone colour of sound (nasal, rough or scratchy). In short, timbre allows a listener to judge

two sounds with the same loudness and pitch as dissimilar. For example, two sounds playing the same note with the same intensity are said to have different timbres when played on different instruments, e.g. piano and guitar.

Loudness – Loudness is the way in which humans perceive the amplitude of sound, where the auditory sensation can be put ascending order of quiet to loud.

Tempo – the tempo represents the speed or pace of music, indicating how slowly or fast a sound, usually music, is played.

These four attributes, when placed in a pairwise comparison against one another, resulted in a total of six comparison pairs (Table 13). The aim of this experiment was to observe which attribute from each pair is most often favoured.

Table 13: The six comparison pairs resulting from the four perceptual attributes of melody, timbre, tempo and loudness

Pairs	Melody	Timbre	Tempo	Loudness
Melody		--	--	--
Timbre	Timbre vs. Melody		--	--
Tempo	Tempo vs. Melody	Tempo vs. Timbre		--
Loudness	Loudness vs. Melody	Loudness vs. Timbre	Loudness vs. Tempo	

Details of the study are as follows:

i) Participants

Thirty-eight healthy participants with self-declared normal hearing, aged between 21–60 years old were asked to participate in this study on a voluntary basis. The subjects comprised of twenty-one females and seventeen males. Participants were divided into

two groups – musicians and non-musicians. In this test, the term 'musicians' were defined as those who have received formal musical training for four years and above, or have been and/or are currently employed in the music industry, e.g. performer, music researcher, music lecturer, tuner, etc. All participants were asked to detail any formal musical training they had had and the number of years that they had been trained for before the start of the test. The intended ratio between the two groups was at 1:1, so as not to create any bias in the results. However, the number of non-musician participants was larger (23 non-musicians to 15 musicians). A Chi-squared test was done to determine if the dataset was biased in terms of sex and musical training. At χ^2 (1) = 0.421, p<0.5164, it was found that there was no gender bias within these participants. Similarly, it was found that there was no musical background bias within these participants (χ^2 (1) = 1.684, p<0.1944). No other demographics effects (race, age or sex) were studied. The design of this listening test had been consulted with an expert[12] from the field of applied cognitive psychology of sound and music and followed the informed practices in the area. This study received clearance from the Faculty of Arts and Humanities Ethics Committee and followed strictly the ethical guidelines and protocols set by the committee. A copy of the clearance is attached and can be referred in Appendix B.

ii) Dataset

The audio dataset for this test is comprised of recordings from natural sounds (animals and environmental) and also music. The lengths of audio tracks varied from 1 to 10

[12]Judy Edworthy, Professor of Applied Psychology, School of Psychology, Faculty of Science and Technology, University of Plymouth

seconds, as in some cases, longer audio tracks were necessary in order to allow information to be amply presented and identified by subjects, i.e. melody or tempo. Sound similarity between the target and the source tracks were decided through the use of several sound analysis programmes such as MARSYAS[13] and Praat[14] for information on the timbre and loudness respectively. The tempo information was obtained at different websites[15][16] over the internet that provided ground truth on the beat per minute (BMP) of a particular track. Information on the melodic similarity was also obtained over several websites[17][18] that compared or surveyed melodic similarity manually. Since this information was submitted by humans and is open to preconception, the tracks' melodic contours were then compared visually in Praat to confirm similarities.

iii) Procedure

Tracks were delivered to the participants via headphones at a comfortable loudness level. Three sound tracks were presented; one of which was a target track, and two of source tracks. Participants were required to first listen to the target track, followed by the source tracks. They were then asked, in a forced choice manner, to make a selection between the two tracks, based on which tracks they felt were more similar to the target, e.g. '*Which of these two sounds do you feel match more closely to the target sound?*'. The test was designed so that each source tracks in the pair would correspond to a different attribute that was being compared. For example, in a

[13]http://sourceforge.net/projects/marsyas/
[14]http://www.fon.hum.uva.nl/praat/
[15]www.bmpdatabase.com
[16] www.djbmpstudio.com
[17]http://www.thatsongsoundslike.com
[18]http://ohnotheydidnt.livejournal.com/49811102.html

melody versus timbre pair, one source track would be melodically similar to the target, whilst the other would be closer in terms of timbral similarities, whilst other perceptual attributes that were not being compared were kept constant. This information was not revealed to the participants so as to allow selection to be made at will, since no basis of similarity or perceptual attribute was specified. Examples of the sound tracks from the test can be heard in Appendix A5. Each participant was presented with twelve of these sets, and re-playing of the tracks was allowed. The average time taken to complete this test was roughly ten to fifteen minutes.

iv) Results

Figure 36 shows the result of all six pairwise comparisons, for the combined average between all participants (musicians and non-musicians). The average between both groups is given in percentage values on top of each bar in bold. From the test, it was found out that Melody showed a striking pattern of domination, where out of the six comparison pairs, three which had involved Melody went unchallenged by other attributes, i.e. in pairs Timbre-Melody, Tempo-Melody and Loudness-Melody. It was also found that in general, Timbre appeared to be more dominant than Loudness, and Loudness more dominant than Tempo. However, in the Tempo-Timbre pair, no dominant attribute can be conclusively derived.

To ensure that the results of these pairwise comparisons were not biased, the significance of each result from the pairs was determined through the use of Chi-squared test. Results from four pairs (Tempo-Melody, Loudness-Melody, Loudness-Timbre and Loudness-Tempo) were all found to be statistically significant; indicating

the slight difference in the number of participants from the two groups (Musician and Non-Musician) had not introduced a bias into the result. Thus, the results are considered valid and it can be accepted that Melody was more dominant than Tempo and Loudness, whilst Timbre was more dominant than Loudness and Loudness dominated over Tempo in such relationship as Melody = Timbre > Loudness > Tempo. The chi-squared test workout for this part can be found in Appendix C.

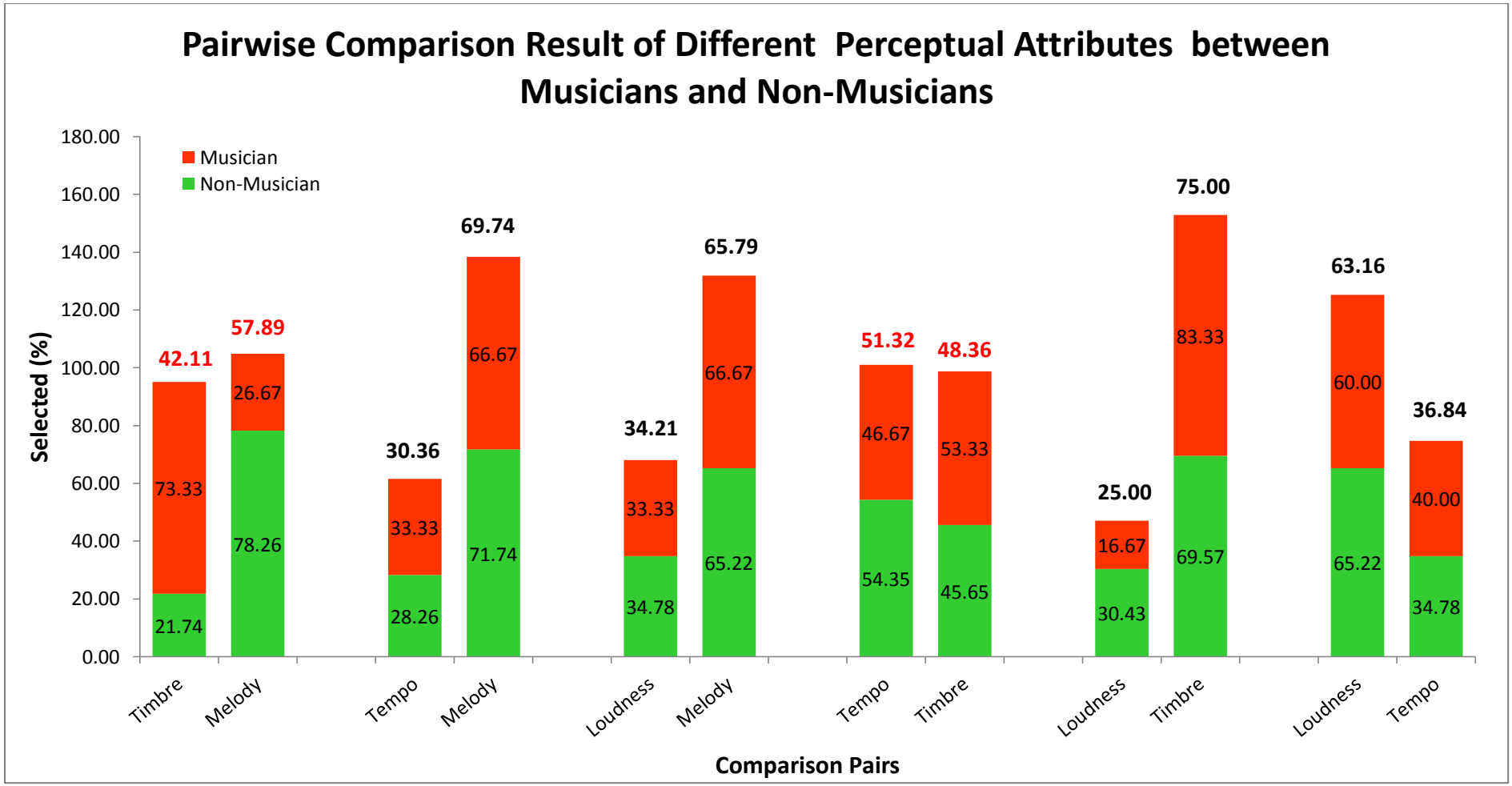


Figure 36: Pairwise Comparison Result of Different Perceptual Attributes to Determine the Dominant Perceptual Attribute in Each Pair

Also in the four pairs that were found to be significant, both showed that the two groups tend to agree on the same dominant attributes, e.g. when the majority of musicians thought the dominant attribute was Loudness in the Loudness-Tempo pair, non-musicians thought the same. However, there were two cases in which this agreement was not found to be true – the Timbre-Melody and the Tempo-Timbre pairs. The average selection percentages of these two cases are highlighted in red ink in the previous chart (Figure 26).

Interestingly, the Chi-squared test found that the result of these two pairs to be statistically insignificant too. At $\chi^2(1) = 1.895$, $p<0.1687$ for the former pair and $\chi^2(1) = 0.053$, $p<0.8185$ for the latter, the null hypothesis must be rejected, suggesting any pattern that might be present occurred only by chance. Hence, it cannot be accepted that Melody is more dominant than Timbre, nor can it be said that Tempo is more dominant than Timbre, as the values obtained from this test were not significant enough to deduce this.

Perhaps it was difficult to conclusively agree on the dominant perceptual attributes as the percentage of selection between the two attributes compared are split in the middle between the Musician and Non-Musician group. Looking closely at the isolated charts of these two pairs in Figures 37 and 38 that follow, this was indeed the case. A 2x2 Contingency Table of Chi-squared Test for Independence was done on both pairs to verify whether there was the case.

In the Timbre-Melody pair, the test of independence had found an extremely significant association between preferred perceptual attribute and participants' musical background ($\chi^2(1)=19.829$, $p<0.0001$). Referring again to the graphs in Figure

37, it can be clearly seen that in the Timbre-Melody pair, Melody was only found to be dominant amongst the vast majority of non-musicians, whereas more than 70% of the musicians selected Timbre.

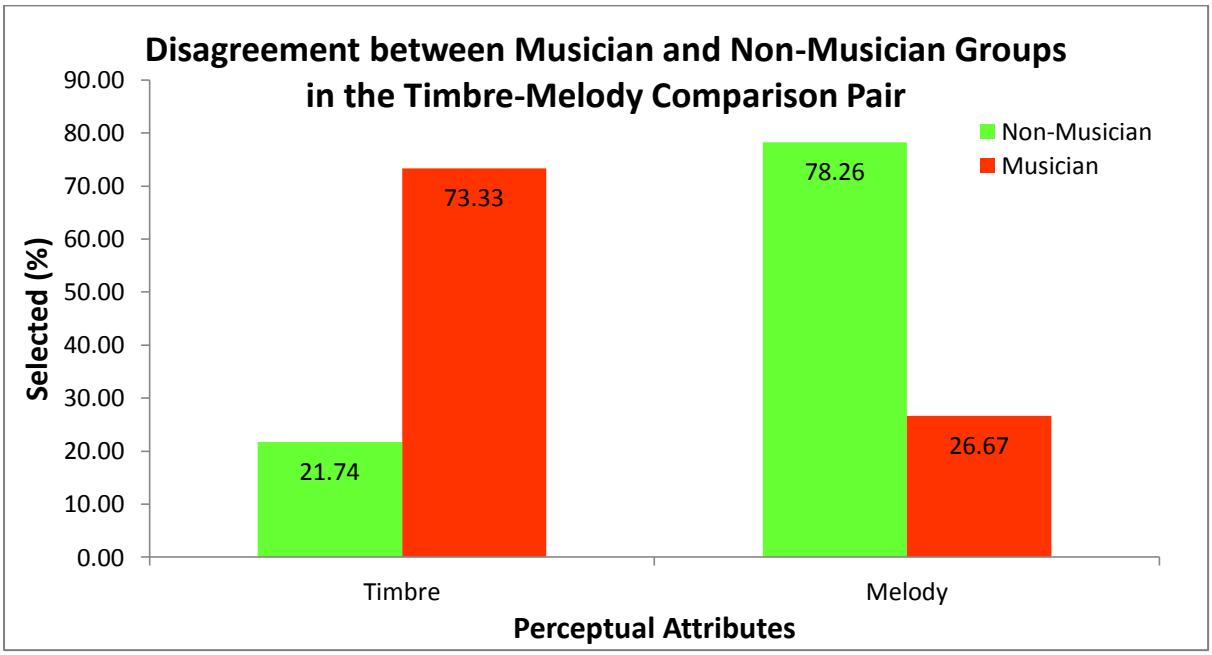


Figure 37: Disagreement between Musician and Non-Musician Groups in the Timbre-Melody Comparison Pair

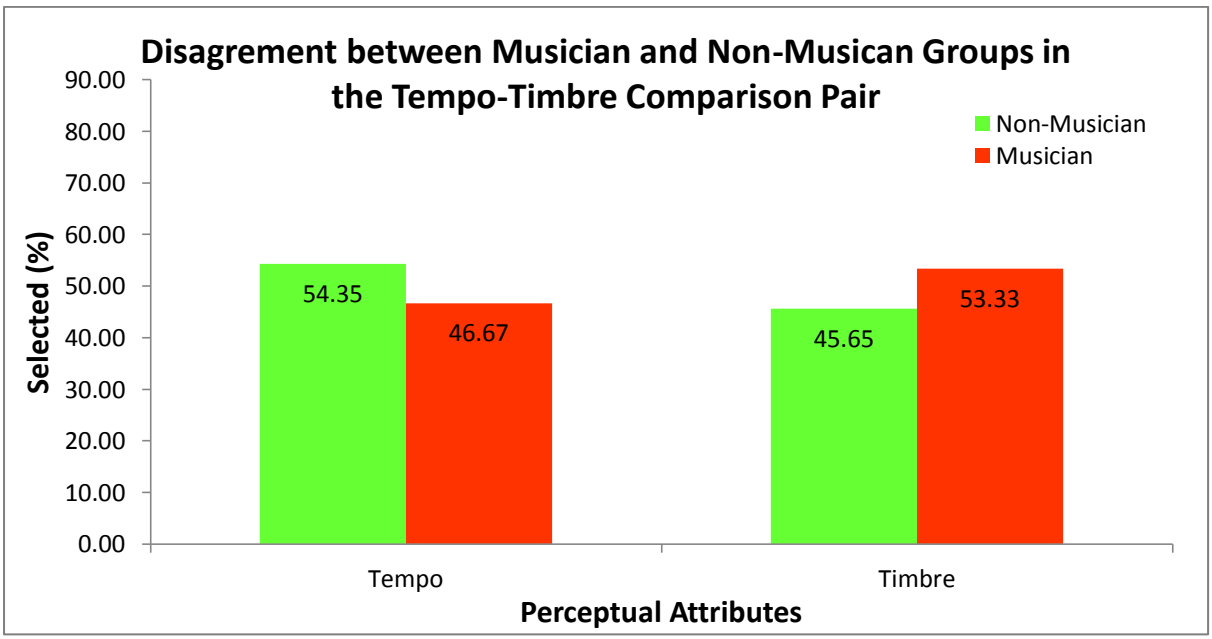


Figure 38: Disagreement between Musician and Non-Musician Groups in the Tempo-Timbre Comparison Pair

However, when a similar test of independence was performed on the Tempo-Timbre pair, it was not found to be statistically significant ($\chi^2(1) = 0.429$, $p<0.5126$). This means that unlike the previous pair, the different musical background of participants did not play a part in their decisions between the Tempo-Timbre pair. The 2x2 Chi-squared Test for Independence for both pairs can be referred in Appendix C.

Perhaps this was due to the flaw in the sound selections in the test design for this pair, or that the number of sound stimuli and size of participants was too small to effectively solve this. Unfortunately, for such a test, it must be remembered that it is difficult to obtain a large number of volunteers, especially for one which required participation of those with a specific expertise on the subject (Musician group). Moreover, in a listening test like this, there can only be a limited number of stimuli presented to the participants before it becomes too long for them to manage.

v) Discussion

From this study, it can be agreed that based on the average selection percentage, Melody seems to be the most dominant perceptual attribute for audio. This could be because Melody is perceptually grouped as part of the same event unfolding over time, based on the Gestalt's principles of perceptual organisation such as similarity, proximity and good continuation. As humans conform to these principles, Melody tends to be preferred over attributes such as Tempo or Loudness (Gates and Bradshaw, 1977).

This phenomenon could also be the direct result of how the human brain is designed. The human brain is divided into two hemispheres, the left lies the more logical and

calculative thinking and the right handles the more intuitive feelings. Musicians tend to use the left hemisphere of the brain to a larger extent when listening to music because they possess an analytical knowledge of it and thus approach music more intellectually. In comparison, those with no musical background mostly perceive music in the right hemisphere because they are not analysing, but are simply experiencing the music (Segalowitz, 1983).

The study also supports that human's musical background does affect the judgment in finding the dominant attribute as musical training alters the way music is perceived by humans. This test shows that musicians generally are more tuned to selecting sounds that are similar timbrally than they are melodically, whereas the reverse is true for non-musicians. Again, this is possibly owing to their analytical behaviour in listening to music, where experienced musicians can be very sensitive in assessing similarities based on the quality of musical expressions rather than the actual melody.

Therefore, sounds that are deemed similar melodically to the non-musicians may not be 'similar' enough for musicians. For example, two same melodies played at varying speed and intensity may still be perceived as two similar sounds by a layperson, but musicians may not agree so strongly, having scrutinised the discrepancies in the technical details such as the tempo and loudness. In comparison, timbre is fuzzy in nature to begin with. There is no clear cut classes or range for timbres which are normally found with other perceptual attributes (e.g. tempo and loudness can be described quantitatively such as slow, fast, low, medium, high or even in a given range such as 110-120 bpm). With timbre, two very different sound sources can be perceived to have very similar sounding timbre, e.g. sound of the rain hitting the roof

and sound of food frying in a pan of hot oil. Unable to approach timbral similarity in the same technical sense as it is for melody, musicians may deduce that two sounds are less dissimilar timbrally than melodically, hence explaining the result seen in this study.

As this is a small-scale study, it is difficult to conclusively conclude whether sound similarity perception in humans is influenced by their musical training alone. Age, experience and even sex might have also affected the result. However, the study highlights that sound similarity is still a very wide and complex area that is yet to be fully understood. To develop a working CSS system that can cater all these perceptual attributes that affect the way humans listen and judge sound similarity on would be a real challenge. Nevertheless, since the study found that musicians (the primary target user of CSS system) are more prone to base their sound similarity based on timbre, audio features that correspond to the timbral attributes will be incorporated in the framework of the new CSS system.

4.3.2 Sound Similarity Performance with Fixed Perceptual Attributes

The previous test has shown that certain perceptual attributes are more dominant than others, depending on the participants' musical training. The next step would be to investigate further if sound similarity agreement can be achieved in humans if the basis of similarity is made clear before the similarity task is conducted.

To conduct this, a slight modification from the previous test needs to be carried out as the objective is to no longer identify the most dominant perceptual attribute between the two studied groups, but it is now to observe if humans are able to agree with each other on

selecting the right source sound (answer) to a target if the perceptual attribute is made known. For instance, when a target sound is presented to humans with two other similar sounds, one which is similar in terms of melody, and the other sound which is similar in terms of timbre, would humans be able to reach an agreement in their answers if clear instruction was given on basis of perceptual attribute?

In order to conduct this second study, the number of comparison pairs of last test had to be doubled. This is because for each pair, the fixed attribute needs to be alternated. For example, in the Timbre-Melody pair, Timbre is first set as the fixed attribute and then it is changed to Melody. Table 14 below lists all twelve pairs, with the fixed attribute notated in brackets, e.g. Timbre-Melody (Timbre) is read Timbre-Melody pair, with Timbre as the fixed attribute.

Table 14: Twelve fixed attribute comparison pairs

	Pairs (Fixed Attribute)
1	Timbre-Melody (Timbre)
2	Timbre-Melody (Melody)
3	Tempo-Melody (Tempo)
4	Tempo-Melody (Melody)
5	Loudness-Melody (Loudness)
6	Loudness-Melody (Melody)
7	Tempo-Timbre (Tempo)
8	Tempo-Timbre (Timbre)
9	Loudness-Timbre (Loudness)
10	Loudness-Timbre (Timbre)
11	Loudness-Tempo (Loudness)
12	Loudness-Tempo (Tempo)

i) Participants

The same participants from the previous experiment took this test. There were several advantages to using the same participants, such as the convenience of keeping the same number of participants for both tests, and that any sound similarity judgments passed during this task were as close as it possibly could be to the previous test, as no new elements were introduced. Although the test was done in a single seating, participants were notified that there would be several different tests conducted with different objectives beforehand, and that the beginning and ending of each tests were clearly marked and announced. Participants were still kept under their original groups, the Musician and Non-Musician groups.

ii) Dataset

Different sound tracks were used this time to stop repetitions that could raise suspiciousness in the participants, but otherwise the audio format, audio lengths, method of procurement, pre-processing and sound similarity analysis were exactly the same.

iii) Procedure

The test set up remained identical to the first test, where participants were presented with a target sound and were asked to select one out of the two possible source sounds as the answer. However, instead of choosing the sound which the participants felt were closer to the target sound, they now needed to choose the sound that was similar to the target with regards to a specified attribute, e.g. '*Which of these two sounds match the target sound in terms of the TIMBRE?*'. Twelve sets of questions

were asked in total, and again, re-playing of the sounds was permitted. Participants took, on average, ten to fifteen minutes to complete this task.

iv) Results

The result of sound similarity performance with fixed attribute in Figure 39 shows that humans are indeed able to successfully select the sound that corresponds to the correct attributes specified. Out of the twelve pairs, ten had average scores of above 80% (the average scores are marked in bold on top of the columns). It can also be noted that participants from the Musician group fared better than the Non-Musician group, with at least seven occasions where complete perfection score were obtained (100% correct selection). Nevertheless, the Non-Musician group did not perform too badly either, where the score for most pairs soared above 70%, except for the two cases involving tempo that were slightly lower. With regards to individual attributes, there was not any significant difference in the scores between Melody, Timbre and Loudness, but a noticeable struggle was seen in the selection involving Tempo across both groups.

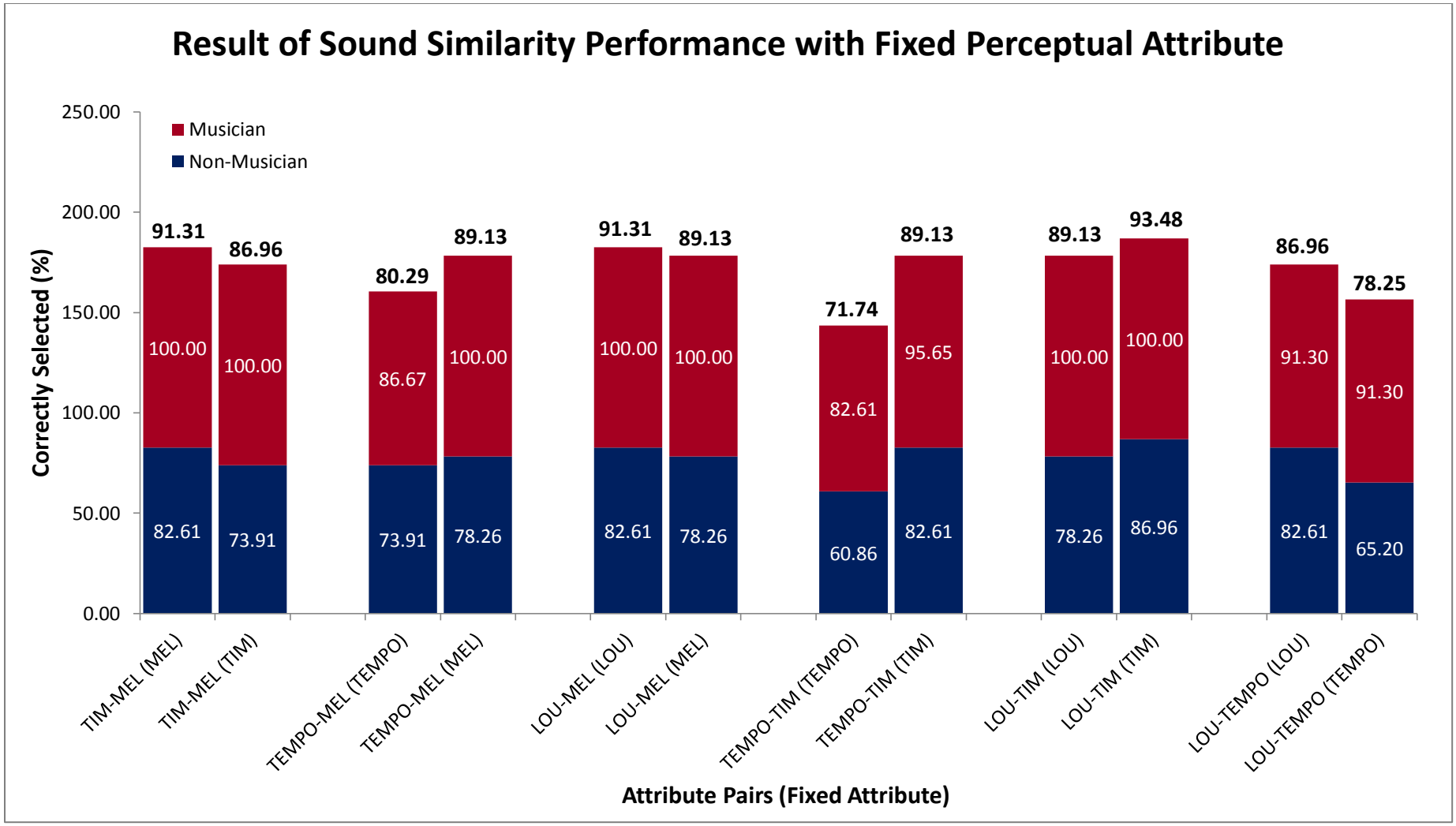


Figure 39: Result of Sound Similarity Performance with Fixed Perceptual Attribute

v) Discussion

The test confirms that it is very possible for humans to agree with each other in selecting the right source sound to a target when the basis of similarity is made clear. Confusion over the basis of similarity is in fact, the reason that caused split decisions in the earlier pairwise comparison test. Stating which perceptual attribute as the basis of sound similarity from the beginning will ultimately remove all confusions and possibly increase the chance of matching users' expectation. It is theorised that applying this knowledge to existing CSS system will improve the outcome, at least by generating sounds that better match of the expectation of its users.

As expected, participants with a musical background performed better than those without. Consistent pattern in their high scoring is also exhibited, suggesting that this ability is related to their training. In addition to the lack of formal training in the Non-Musician group, it is also probable that the slightly lower scores from this group were the result of confusion over the definition of the perceptual attributes. Although a brief description surrounding all four attributes prior to the start of the test was given and it was made clear that participants could seek clarifications or ask questions at any point during the test, only five participants from the Non-Musician group did so throughout the entire length of the study. It had not been possible to ascertain whether or not participants from this group had fully understood the definitions well enough.

It was also revealed from the test that between individual attributes, tempo fared the worst. Both groups were found to have struggled selecting the correct sound tracks that represented the tempo. From casual observation of the participants' behaviour during the test, only one participant from the Musician group had actually tapped his

hand on the desk during the test, an indication that he was comparing on the BPM of the sound. A plausible explanation could be that the human brain tends to regard songs with the same melody but played with different speeds as the same song. This phenomenon happens because listeners understand similarity as tempo invariant in context of isochronous fragments (Hofmann-Engl, 2001). Therefore, it does not come as a shock that tempo was the least dominant attribute in the previous test and the least correctly scored attribute in this test.

This second test further supports the notion that a CSS system should provide its users the option to select the basis of sound similarity, or at least, makes clear to users what is the basis of similarity of the sounds that are about to be synthesised. Understanding the basis of sound similarity will minimise any human-computer misperception.

4.4 Query-based Concatenative Sound Synthesis Model

Earlier sections in this chapter had proposed solutions to the problems that lie in existing CSS systems which have been brought up in Chapter 3 previously. Challenges involving order-dependent feature selection, handling homosonic and equidistant segments during unit selection and overcoming confusion over basis of similarity have been tackled through the use of AHP, concatenation distance and determination of dominant perceptual attribute respectively. Figure 40 presents the proposed CSS model that involves slight modification of the original CSS model to accommodate the incorporation of these solutions.

The new model retains all the components that had been originally present, but adds a 'Query' stage between the target input, database and unit selection process. Granted, this stage had always been implicitly present, however, it needs to be acknowledged that the query stage is essential and in fact, the core of the system, as all means of command from the user gets communicated through. By adding the query stage, it is made apparent that different parameters can be added, selected or enabled e.g. audio feature options, weight assignments for each feature, clarifying the perceptual attribute that defines the basis of sound similarity, etc. This had all been inexpressible previously, as existing CSS systems typically allowed limited exchange of information from user to the system, e.g. basic feature selection option, enablement of taboo list and threshold of match.

The proof-of-concept developed based on this novel framework is aptly named '*ConQuer*', short for CONcatentive sound synthesis system based on the QUERy-based model.

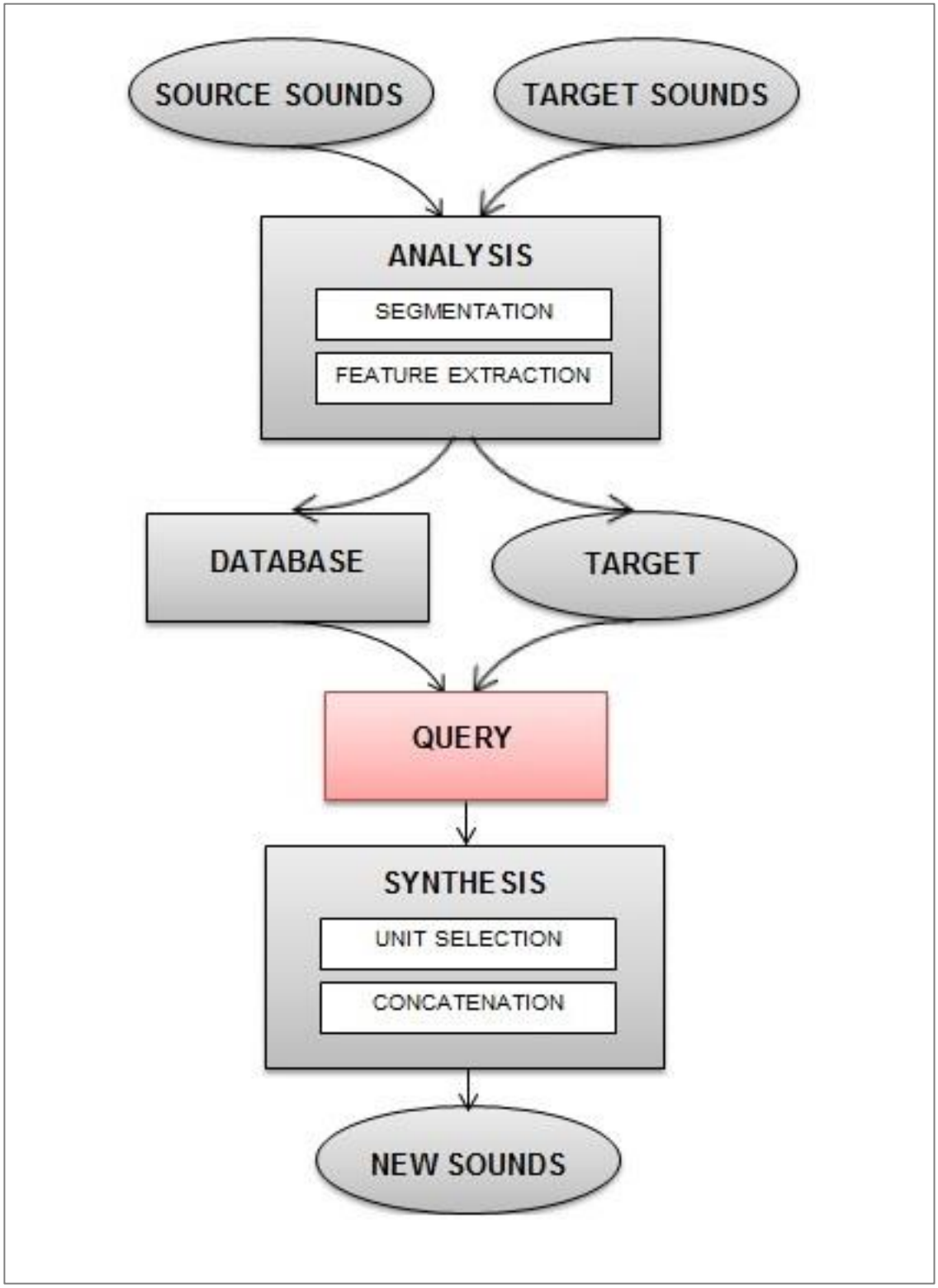


Figure 40: The new, 'Query-based Concatenative Sound Synthesis Model'

As the foundation of the new model is the query stage, it is most aptly termed as the 'Query-based Concatenative Sound Synthesis Model'. Among the positive and encouraging characteristics of this new model are:

i) *Enhanced user control flexibility*

This new model expands the flexibility of existing CSS systems as the focused query stage provides a proper channel for all commands and needs to be communicated by the user to the system, including directory of source files, selection of segmentation mode, selection of audio features and assignment of weight for each feature.

ii) *Robust to changes*

The generic query allows the exchange or addition of existing option with other parameters that may be relevant but are not mentioned in this study can also be included easily. For instance, in this study the timbral attribute was decided as the basis of sound similarity, but this can be easily changed to suit the target user.

iii) *Intelligent and methodical solutions*

Deriving solutions through the AHP method and the selective use of concatenative sound synthesis provide much more intelligent and soundly reliable solutions than through random selection, which was the classic approach before.

iv) *Reduction in post-synthesis adjustments*

Most CSS systems rely on the post-synthesis adjustments of the generated sounds to achieve the sound that they need. Failure to engage in a channel that allows them to express the criteria of what envisioned sound frustratingly result in blind synthesis, requiring numerous adjustments afterwards. Presenting the parameters and criteria clearly prior to unit selection will reduce the number of adjustments that is needed.

4.5 Summary

In this chapter, the problems cited previously in Chapter 3 were addressed. A summary of the solutions to the three main issues focused are described below:

i) Order-dependent features selection

The first part of this chapter explained the methodology of the Analysis Hierarchy Process (AHP) and how it could be utilised to generate weights for each audio feature according the order of importance as specified by the users. A complete simulated synthesis result was also provided to demonstrate the effectiveness of AHP in overcoming this challenge. It was found that AHP successfully processed qualitative judgment from users and transformed them into reliable quantitative format from which consistent results can be obtained.

ii) Homosonic and equidistant segments during unit selection

The concatenation distance was proposed as a solution to this challenge during unit selection. The concept of concatenation distance was presented, and the role it played in determining the synthesis result was also shown. In addition, a slight change was introduced to the original model of calculating the overall segment cost by following the newly proposed hierarchical model. Result from the original model and the hierarchical model was then compared.

iii) Basis of sound similarity

Two small-scaled listening tests involving human participants were conducted to identify the dominant perceptual attribute which humans most often use to pass their sound similarity judgment on. Whilst the area of sound similarity was indeed vast and

complex, the tests revealed that sound similarity in humans was affected by their musical background. Non-musicians generally regarded sound similarity in terms of melody, whilst musicians tended to base their similarity judgment on the timbral quality. It was deduced from the results of these tests that by customising the basis of similarity according to the respective target user, the human-computer sound similarity misinterpretation could potentially be minimised.

The final section in this chapter included a proposition for an additional stage to the original CSS model – the Query stage – thus fittingly re-naming the model as 'Query-based Concatenative Sound Synthesis System'. The query stage provided the option for users to specify their inputs into the system, particularly in the three elements above. The options are: option to different the importance intensity between audio features, option to enable the use of concatenation distance to select sound units in the event of homosonic and equidistant segments, and also the option to specify the perceptual attribute which the system should base sound similarity on.

In conclusion, this chapter intensively described the possible solutions to the problems that still occur in existing CSS systems. The feasibility and efficiency of the solutions proposed in this chapter will be verified in the next chapter, through series of simulated test and also a final listening test.

Chapter 5: Experiments, Results and Discussions

The three main aims of this study are to propose a novel framework that addresses the issues that still remain in existing CSS systems, to enhance user control flexibility of the system and also to achieve better sound similarity agreement between humans and system. These aims will be fulfilled through the extended use of Artificial Intelligence approaches which have been derived from the understanding of humans' sound cognitive domain. The methods proposed to achieve these aims have already been described in depth previously in Chapter 4 (the use of AHP in order dependent feature selection, the use of concatenation distance in homosonic and equidistant segments, and the use of timbral feature sets as the basis of sound similarity). This chapter intends to verify the feasibility and suitability of the proposed novel framework, whereby the significance of each method used is evaluated.

Several experiments have been designed for this purpose. The experiments were conducted in different phases, three of which involved computer simulations and one involving a listening test in the later phase using human subjects. The experimental sets are as listed below:

1) Phase 1: Parametric Input Evaluation
2) Phase 2: Audio Features Selection Evaluation
3) Phase 3: Search and Selection Evaluation
4) Phase 4: Listening Test

The first phase of the experiment investigated the initial factors namely the input parameters, and how they affected the concatenation results. The second phase evaluated the feasibility and performance of order-dependent feature selection in CSS system through the use of AHP. In a similar manner, the third phase evaluated the efficiency of using concatenation distance in solving homosonic and equidistant segments during search and selection. The final phase of the experiment included a listening test that investigated the correlation between the similarity and interestingness of sounds across two groups: musician and non-musician.

The components that were involved in the first three experimental sets will be described first as they are similar to one another in a sense that they shared the same dataset and were conducted following similar procedures. The components for the final experiment set will be explained separately towards the end of the chapter.

i) Methodology

The mechanism in which the three simulation-based experimental sets were conducted was very similar, whereby a target sound was first supplied to the system. Having undergone segmentation, matching source units would then be searched from the entire database according to the criteria specified at the beginning of the search and the closest matching source segments were then concatenated together and synthesised. All the tests carried out under these three experimental sets measured the target distance, along with the experiment process time in seconds. Tests in the third experimental set additionally measured the concatenation distance. Both distances were calculated using the equations formerly described in equations (2) and (7), which can be referred to in Chapter 3, p.74 and Chapter 3, p.104 respectively.

Where appropriate, sounds that were used in these experimental sets and the sound samples that were products of synthesis via *ConQuer* can be referred in the CD attached with this thesis. The full details of the sounds are as detailed in Appendix A.

ii) Dataset

Again, the same dataset was used for all the tests involved in the three simulation-based experimental sets. The target sounds comprised of several pieces of music from the classical and country genre. Such genres were selected to observe the sounds that were synthesised as a result of two very different sounds. The source sounds were made up of sounds of nature; ranging from the sounds of different species of primate screaming, to singing whales in the ocean, to the sounds of birds chirping in the rainforest. The exact and target and source sounds included in each individual test will be mentioned later at the beginning of each test. Overall, approximately thirty minutes worth of sounds made up the entire collection of the dataset that was used in these experimental sets. Although this number may appear relatively small, it must be recalled that these sounds were then further segmented into smaller sound units, resulting in over 1200 segments in total in the database.

iii) Tools

The simulation-based experiments relied on running several tests on the proof-of-concept prototype, *ConQuer*. *ConQuer* is written in using Bash script on Ubuntu Linux version 9.10 (Karmic Koala release). In order for it to perform all the necessary stages expected in a working CSS system, it combines the use of several other tools such as Aubio, MARSYAS, Praat, Audacity and SoX, to elicit the tasks of segmentation, feature

extraction and sound manipulation respectively. These tools were selected based on their functionality, portability and because they are free and made available to public. Below are brief descriptions of each tool:

a) ***Aubio***

Aubio is an open-sourced tool designed and developed by Paul Brossier (Brossier, 2006) for the extraction of annotations from audio signals. The tool is also capable of performing many tasks involving audio such as sound segmentation, pitch detection, beat and tempo tracking, among many others. Its `aubioonset` and `aubiocut` functions are particularly useful for taking in input sounds and automatically segmenting them at every detected onset or beat, creating, new small sound segments. Aubio is specifically used for this purpose in this study. Its `aubiopitch` function is also useful in extracting pitch information during sound analysis at a later stage in this study.

b) ***MARSYAS***

MARSYAS (Music Analysis, Retrieval and Synthesis for Audio Signals) is an open source framework for audio processing with specific emphasis on Music Information Retrieval applications. One of its exciting features is that it can extract audio information from segments of music based on one of these three audio contents: timbral texture, rhythm content and pitch content, which are exactly the feature sets involved in this study.

c) ***Praat***

Praat is a free scientific programme for the analysis of speech in phonetics. A variety of analyses is available for speech signals, including pitch, intensity, formants and spectrogrammes and spectral balance. It also has many additional features such as playback, labeling, contour editing and scripting. Although it has been primarily developed for use in speech analysis, it has been tested to work on musical dataset as well, with interesting results[14]. For instance, it was successfully used to find the note, duration of note and the amplitude in a flute clip of a Hindustani classical music (Makaran Ramesh and Sahasrabuddhe, 2008). Praat is used in this study to aid in pitch extraction, and also in the listening tests where its contour viewing and editing function are utilised.

d) ***SoX***

SoX, or the Sound eXchange is another free cross-platform audio editor. It is popularly nicknamed the Swiss Army knife of the sound processing programme as it can perform various tasks involving audio such as recording, playing, editing, concatenation, reverse playing and many other useful processes. It is used in this system mainly to aid basic manipulation of sounds.

[14] A basic tutorial on how Praat can be used in music analysis can be found on Praat's main page at: http://www.musicology.nl/wm/research/praat_musicologists.htm

5.1 Phase 1: Parametric Input Evaluation

Akin to the phrase 'rubbish in, rubbish out', it is assumed that one of the important factors to first affect the result of synthesis in CSS system is the input submitted and the criteria set at the start of the search query. Hence, the objective of the tests carried under this experimental set is to investigate the parametric factors at the initial stage that can affect the result of concatenation and ultimately the sounds synthesised by a CSS system. This study investigates the number of source files to be added into the database, the actual source files and target files to be incorporated, and the mode applied during segmentation.

5.1.1 The Effect of Number of Segments on the Synthesis Result

i) Experimental Set Up

In this experiment, a 10-second long country music was selected as the target sound, whilst the sound of the primate *Indris* was selected as the source sound. To keep it simple, the centroid was the only audio feature set to be compared against for the basis of similarity. Since the variable investigated in this experiment was the number of segments, different values were set for the source sound *Indris*. Starting at its highest point where all segments were included (382 segments), the number was progressively halved until five different values were obtained: giving 382, 191, 95, 47, and 23 segments respectively. The average target distance and time taken to complete the concatenation for each test set were noted.

ii) Results

Figure 41 shows the progression of the target distance and the run time between five different dataset sizes: 382, 191, 95, 47 and 23 segments from the *Indris* source sound.

It can be seen that the average target distance increases as the number of source segments in the database decreases. In other words it became increasingly difficult to find close matching segments as the dataset size grew smaller. Interestingly also in this particular case, that no noticeable difference in the target distance between the first three values is displayed, suggesting that the performance cannot infinitely grow better by solely increasing the number of segments alone. Instead, it seems that after the dataset grew to a certain size (the optimum number); increasing the source segments will bring little effect in improving synthesis result. The reverse effect that is happening between the number of source sounds and the time taken to complete the task (the larger the dataset size, the longer the time required to complete the task) was also noted.

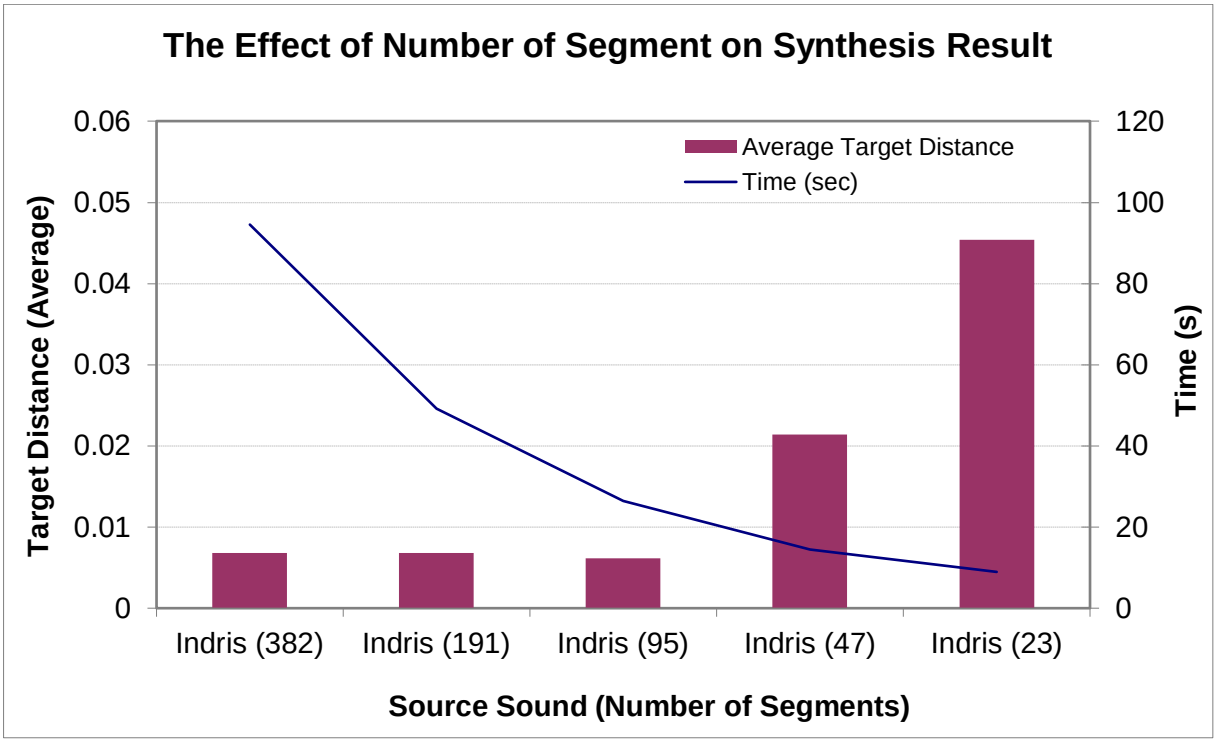


Figure 41: Result of Number of Segment on Synthesis

iii) Discussion

By increasing the size of the search pool, the possibility of finding an exact or closely matching segment is increased, thus closing the target distance gap between the target and source sounds. In sound synthesis, this means that there is a higher chance of synthesising sounds that are closer to the target. However, after a certain point, an increase in the dataset size no longer improves the synthesis result. This can happen when many of the sound segments in the database are actually redundant segments or segments that are represented with the same audio information. These segments bring no real improvement to the synthesis output. The phenomenon observed is known as the 'ceiling effect'.

With respect to the run time, a positive relationship between the dataset size and the time required to complete the task is expected and observed, as larger dataset means a wider search area needs to be covered during unit selection. Nevertheless, in this experiment, with the current size used in the database, coupled with the fact that it was not executed in real time, the maximum time taken to complete the task was reasonably acceptable at roughly $3/10^{th}$ of a second per segment.

5.1.2 The Effect of Different Source File on the Concatenation Result

i) Experimental Set Up

This experiment studied the effect that different source files have on the concatenation result. Seven different classes of source sounds were tested: *Canary*, *Indris*, *Lemur*, *Rainforest*, *Siamang*, *Tiger* and *Whales*. Each class of source sounds was alternately used as the source file from which the matching segments were selected

from for the target segment. Other variables were kept constant during each of the sub-tests, such as the target sound (10-second country music was re-used), audio feature (centroid) and the segmentation mode (onset mode). The result of this experiment is plotted below.

ii) Results

Figure 42 displays the concatenation result across seven source files. In general, different source files returned different sound outputs, both in terms of average target distance seen in the chart above and in the generated sounds which can be referred to in Appendix A6. In this particular setting, the performance of the source files seems to be divided into two clusters: top performers (*Siamang*, *Whales*, *Canary*, *Tiger*) and worst performers (*Rainforest*, *Lemur*, *Indris*). Consistent run time was also observed between all seven source files, in the range between nine and eleven seconds. This had been expected as the same number of segments was assigned for each source file.

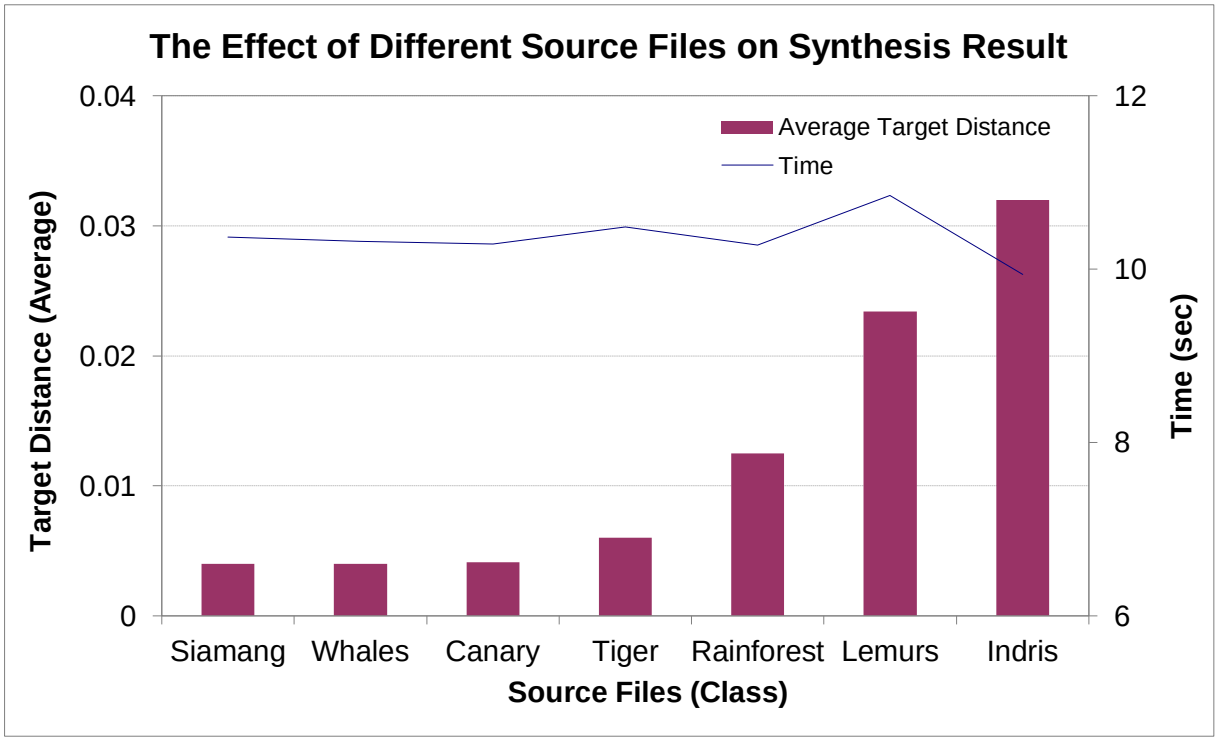


Figure 42: Result of Different Source Files on Synthesis

iii) Discussion

The experiment confirms the obvious, that the use of different source files causes different sounds to be synthesised. To achieve better synthesis result, it is useful to load sounds that closely resemble the final composition envisioned into the system's database.

On the other hand, adding what is initially thought as 'misfits' or 'odd sounding' files may actually bring in interesting surprises to the synthesis result. For instance, sounds of three primates were among those included in this test. Naturally, it was thought that results from these three groups would somewhat be closer to each other. However, the performance of the primates were divided into two, with *Siamang* at the very top alongside *Whales* and *Canary*, whilst *Indris* and *Lemurs* at the very bottom. This suggests that sounds can have roughly the same spectral information yet sound perceptually different, and vice versa. Perhaps there lies some underlying similarity in the musicality of between certain sounds that are not immediately noticed by humans. Only by experimenting with different source files will these interesting syntheses be discovered.

5.1.3 The Effect of Different Target File on the Concatenation Result

i) Experimental Set Up

The procedure taken for this experiment was fairly similar to the previous experiment, where all independent variables were kept constant (number of segments, audio features, segmentation modes). The only difference was that instead of looking at how different source files affect synthesis (as was the case previously), this experiment is

now focused on the effect of different target files on the synthesis result. Two target files were compared, *Classical* and *Country*. To ensure that any pattern that occurs was not a one-off occurrence, the experiment was repeated on four different source files – *Indris*, *Lemurs*, *Siamang* and *Whales*. The number of segments for all the target files and source files were set at forty segments each, to eliminate any pattern that emerge as a result of the dataset size differences.

ii) Results

The experiment found that just as different source files returned different synthesis results, different target files also affect the synthesis result both empirically and aurally. These can be evidently seen and heard in Figure 43 the sounds generated in Appendix A7. Between the two target files tested, *Classical* performed worst in all four different source files. *Classical* also took longer to run in all four cases.

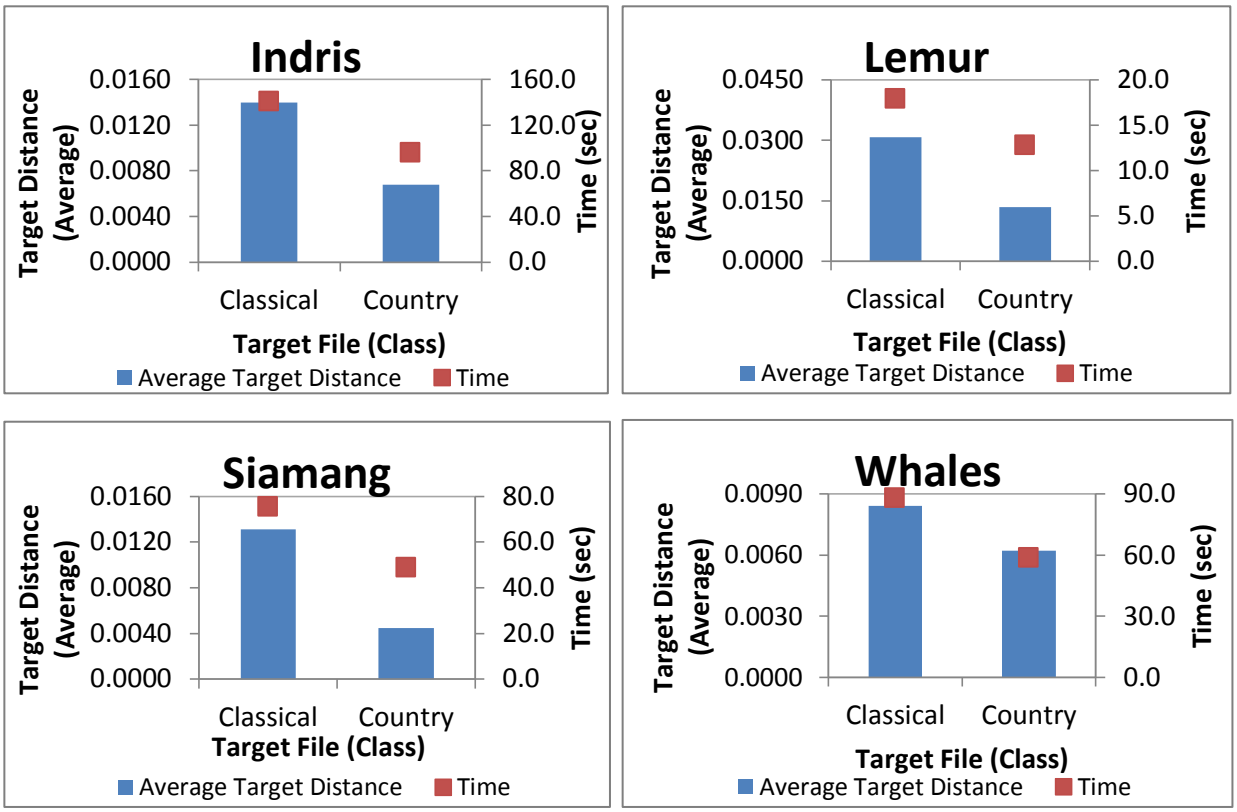


Figure 43: Results of Different Target Files on Synthesis

iii) Discussion

The results from this experiment re-iterate that different input impacts the output of synthesis, i.e. different results are seen and heard with different target files. In this particular comparison, it was much easier to find matching segments from the database when the target file was *Country* than it was for *Classical*. This was due to the louder and livelier nature of the former, which coincided with the loud and pitchy sounds of most of the source files in the database, the three primates in particular. It is also interesting to note that whilst the target file *Classical* did not manage to outperform its rival, the gap between the two target files was the smallest with the source file *Whales*. It is thought that the similar mellow the nature of both target and source files reduced the performance gap of the sound from the opposite target file.

Thus, the act of selecting the appropriate target file is equally as important as selecting the source files, because the foundations for the creation of new music through the use of a CSS system are laid by the content information retrieved from the target file.

5.1.4 The Effect of Different Segmentation Modes on the Concatenation Result

i) Experimental Set Up

This experiment continued looking at the effect of the final input parameter which is the segmentation mode and what effects does this parameter have on the overall synthesis result. Two segmentation modes were studied: homogenous segmentation (time-based) and onset segmentation (event-based). Homogenous segmentation was set to happen at every 500 milliseconds, whilst the onset segmentation was set to happen at every beginning of an attack in a sound signal. All other variables that were

not compared remained unchanged (centroid for the audio feature, *Country* for the target file, and *Indris*, *Lemur*, *Siamang*, and *Whales* for the source files). The dataset size could not be controlled as the ways in which the two segmentation modes were designed to function had led to varying number of segments. In addition to the usual average target distance and run time, the duration of the synthesised sounds was also measured and compared to the original 10-second long target sound.

ii) Results

Figure 44 shows the synthesis results between homogenous segmentation and onset segmentation at four different source files. No definite pattern is revealed from this experiment. As can be seen, when average target distances across four source files were compared, two out of the four cases favoured homogenous segmentation whilst the rest favoured onset segmentation. Nevertheless, as far as run time was concerned, homogenous segmentation was a clear winner having finished the task with the less amount of time in all four cases.

Table 15: Number of segments produced between Homogeneous Segmentation and Onset Segmentation

	Homogenous	**Onset**
Indris	434	382
Lemur	94	98
Siamang	244	197
Whales	55	235

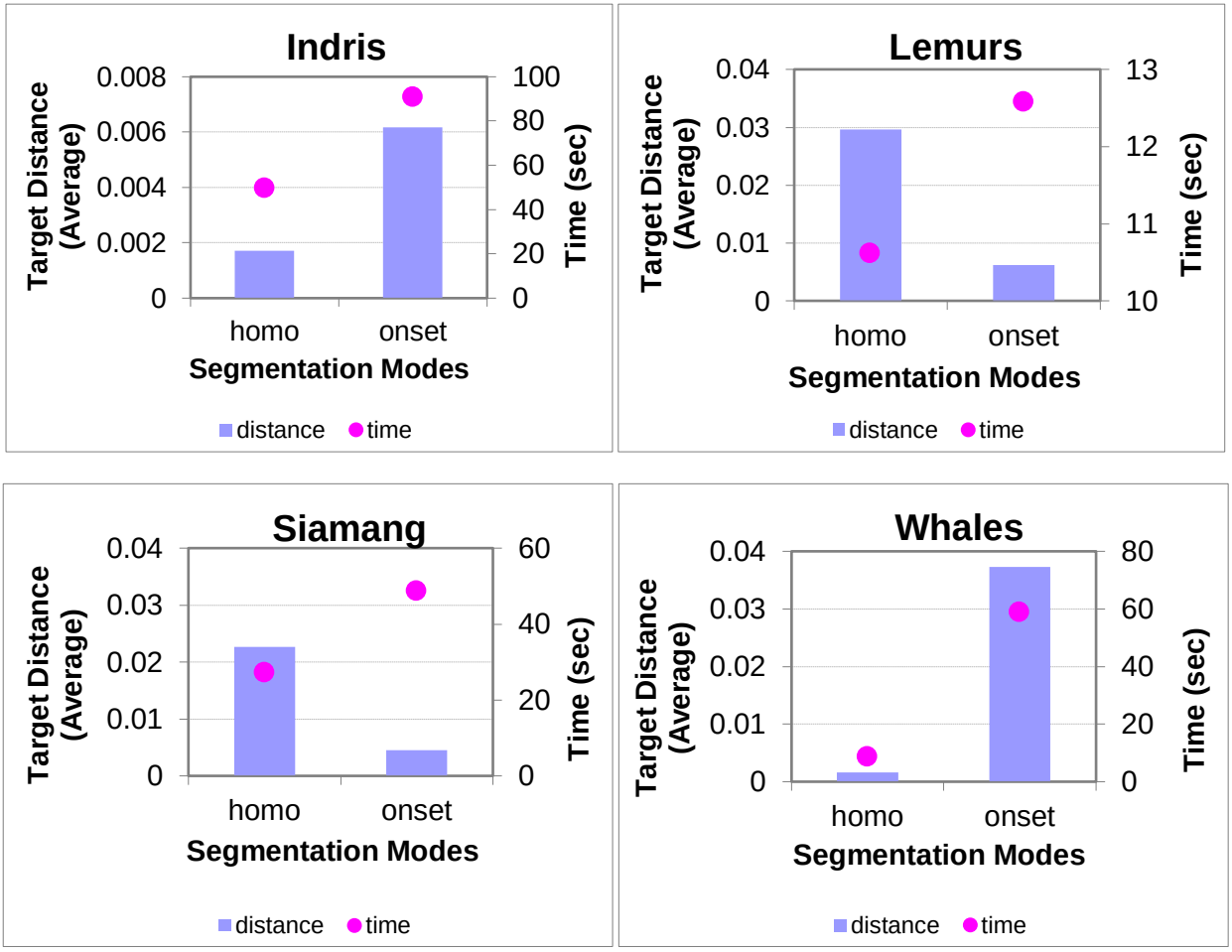


Figure 44: Results of Different Segmentation Modes on Synthesis

iii) Discussion

No concrete conclusion can be made from the result of this experiment. In some cases homogenous segmentation seemed to return closer matches than onset segmentation, whilst in others the reverse was true. Also, it cannot be agreed on which segmentation mode will produce more segments than others, as this highly depends on the actual make-up of the sound in question. Nevertheless, the general rule is to avoid using homogenous segmentation for sounds which are more rhythmic with plenty of attacks because segmentation at a pre-determined time tends to cause audio stream to be chopped at unfavourable positions. If, however, the condition states that

the duration of the synthesised sound must be equal to the length of the original target sound, then homogenous segmentation is most suited for this purpose.

5.1.5 Conclusion

This first experimental set investigated the effects that four main input parameters had on the sounds generated via CSS. Several conclusions that can be deduced from this experimental phase are:

i) *The larger the dataset size, the higher the likelihood of finding closer matching segments, at least with respect to the average target distance*.

The chances of finding source segments that are exact or near exact match to the target segments are greater when the selection is wider, although it must be recalled that after a certain point, the dataset size ceases to leave a positive impact anymore (the 'ceiling effect'). In addition, it depends closely on both the target and source sounds that are included in the query.

ii) *Synthesis result is dependent of the target and source sounds set by the user*

Various forms of improvements, optimisations and transformations may be able to enhance the sounds generated from the CSS system to a certain degree, but the target and source sounds are the key input parameters that ultimately determine the outcome of the synthesised sounds. Thus, it is important to ensure that the correct target sound is provided into the query and suitable source sounds are loaded into the database in order to increase the chance of generating sounds that correctly align the user's expectations.

iii) *Understanding the intended purposes and the working mechanisms of both homogenous and onset segmentations before selecting the segmentation mode for a particular concatenation task can help improve synthesis result*

Both segmentation modes have their own strengths and weaknesses, and serve to suit different functions from one another. Onset segmentation is more suitable when individual events or content information of the overall sound is needed to be extracted, e.g. rhythm, beat, attack; whilst homogeneous segmentation is more ideal when the sound units need to be in equal length. Knowing the output criteria of the segments before selecting between the two modes may help improve the synthesis result.

iv) *Larger dataset and complex segmentation algorithm contribute towards the increase in the run time of a CSS system*

Larger dataset size means that there are more comparisons that needed to be carried out between target and source segments in the database before the one with the least target distance is selected. Also, when the onset mode is enabled, the calculation involved during segmentation is more complex than the time-based homogenous segmentation of which the algorithm is significantly more straightforward, thus adding up the total run-time. For a system designed to be run in non-real time, this is normally not a major concern, but if the run-time is an issue, then the user must determine whether the use of larger dataset and onset segmentation are worthy trade-offs.

5.2 Phase 2: Audio Features Selection Evaluation

After the initial input parameters, the next factor that may affect the synthesis result is the audio features selection. An audio segment is characterised into compact numerical representation via a process known as feature extraction. This numerical representation becomes the basis of comparison between the target and source segments during search and selection. Numerous audio features can be extracted from a single audio segment. This experiment intends to demonstrate the effect of several audio features on the synthesis result. More than one audio feature may also be included as the basis of comparison, and since one feature may not carry the same weight as another, the AHP had been previously proposed to solve this problem. The feasibility and effectiveness of this approach are tested in the latter part of this experimental set.

5.2.1 The Effect of Different Audio Features on the Synthesis Result

i) Experimental Set Up

The effect of five audio features (spectral centroid, spectral rolloff, spectral flux, zero crossing rate, pitch) was studied –The feature combinations that were compared are listed in Table 16. The feature combinations tested did not include all of the possible combinations that were possible, but the sample was representative enough to show the effects of using different single feature (centroid against pitch) and multiple features (centroid and rolloff against centroid, ZCR and pitch). Other constant variables involved were the 10-second long country sound file as the target sound, the *Indris* sound file for the source sound and onset mode for segmentation. Both average target distance and run time were measured.

Table 16: List of the feature combinations tested in determining the effect of different audio features on synthesis result

	Feature Combination	Abbreviation
1	Centroid	CTD
2	Rolloff	RLF
3	Flux	FLX
4	Zero Crossing Rate	ZCR
5	Pitch	PCH
6	Centroid and Rolloff	CTD-RLF
7	Centroid and Flux	CTD-FLX
8	Centroid and Zero Crossing Rate	CTD-ZCR
9	Centroid and Pitch	CTD-PCH
10	Rolloff and Flux	RLF-FLX
11	Rolloff and Zero Crossing Rate	RLF-ZCR
12	Rolloff and Pitch	RLF-PCH
13	Flux and Zero Crossing Rate	FLX-ZCR
14	Flux and Pitch	FLX-PCH
15	Zero Crossing Rate and Pitch	ZCR-PCH
16	Centroid, Rolloff and Flux	CTD-RLF-FLX
17	Flux, Zero Crossing Rate and Pitch	FLX-ZCR-PCH
18	Centroid, Rolloff, Zero Crossing Rate and Pitch	CTD-RLF-ZCR-PCH
19	Centroid, Rolloff, Flux, Zero Crossing Rate and Pitch	ALL

ii) Result

Figures 45 and 46 show the average target distance and run time results of 19 feature combinations that were tested in this test. The use of different audio features definitely returned different results, evident both empirically and aurally (Appendix A8). Also, it can be seen that with the exception of combinations involving pitch, all combinations that included two or less features returned smaller target distance in comparison to combinations involving three of more features. When three or more features were used, the target distance became significantly larger, almost four times as large. Among the best performing features were flux, centroid and rolloff, and combinations derived from them (CTD-FLX, RLF-FLX and CTD-RLF). The worst performance was seen in the combinations which included all five features together (ALL). It can be heard that after the addition of three and more features, the essence of the target sound was lost and what was distinctively the sound of primates screaming as a result of synthesis using one or two features, had gradually morphed into the sounds of avian tweeting with the use of three or more features (Appendix A9).

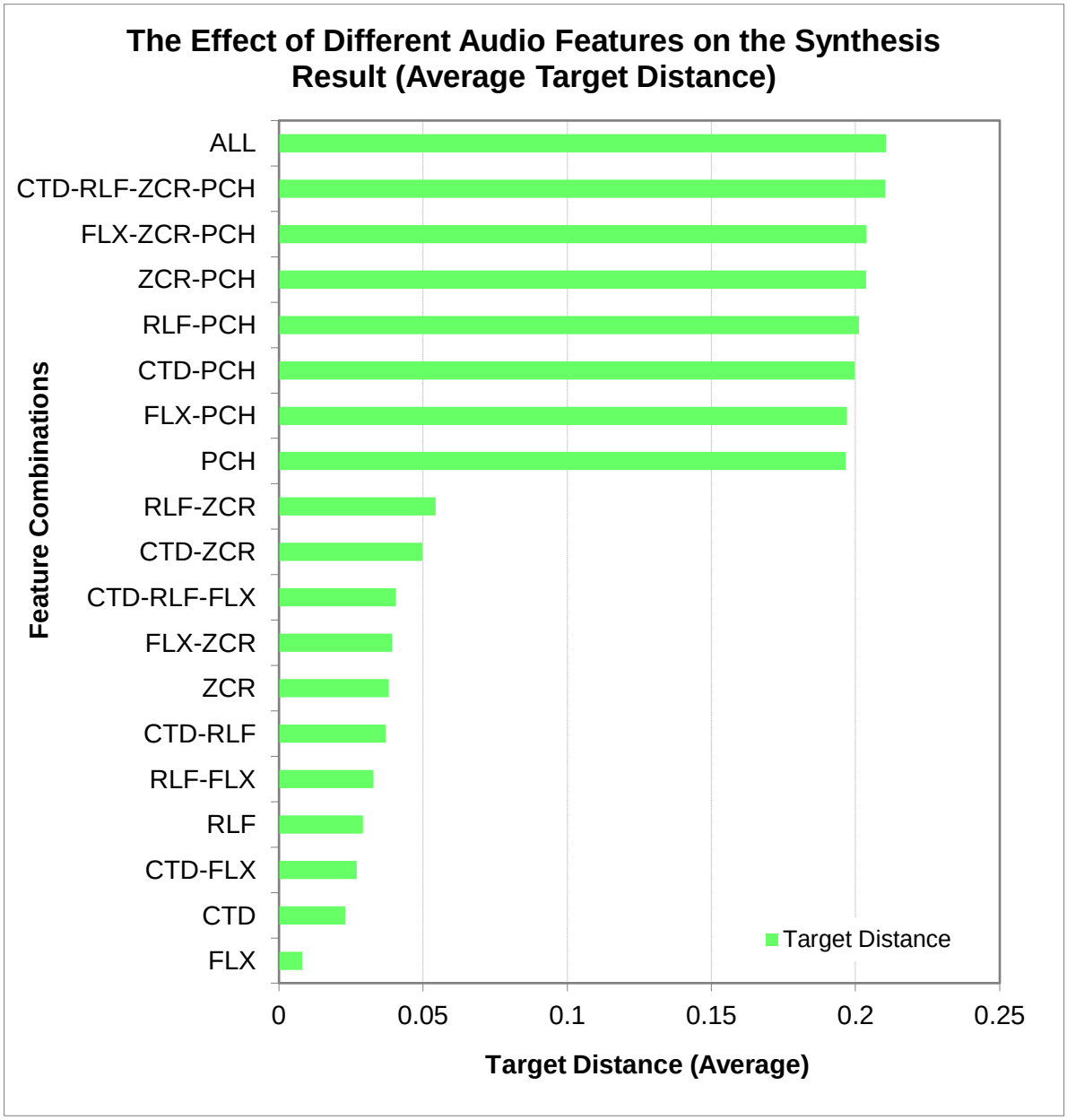


Figure 45: Result of Different Audio Features on Synthesis (Target Distance)

A similar pattern is seen with the run-time result. Single features ran faster compared to other combinations involving multiple features, the worst being when all five features were included. This time, however, no anomaly is exhibited with combinations including pitch. It is also interesting to note that the time increment from the use of single feature to dual features and from dual features to triple features occurred at roughly the same unit which is 70 seconds. This result emerged in a step ladder pattern of progression and can be visually identified in Figure 46.

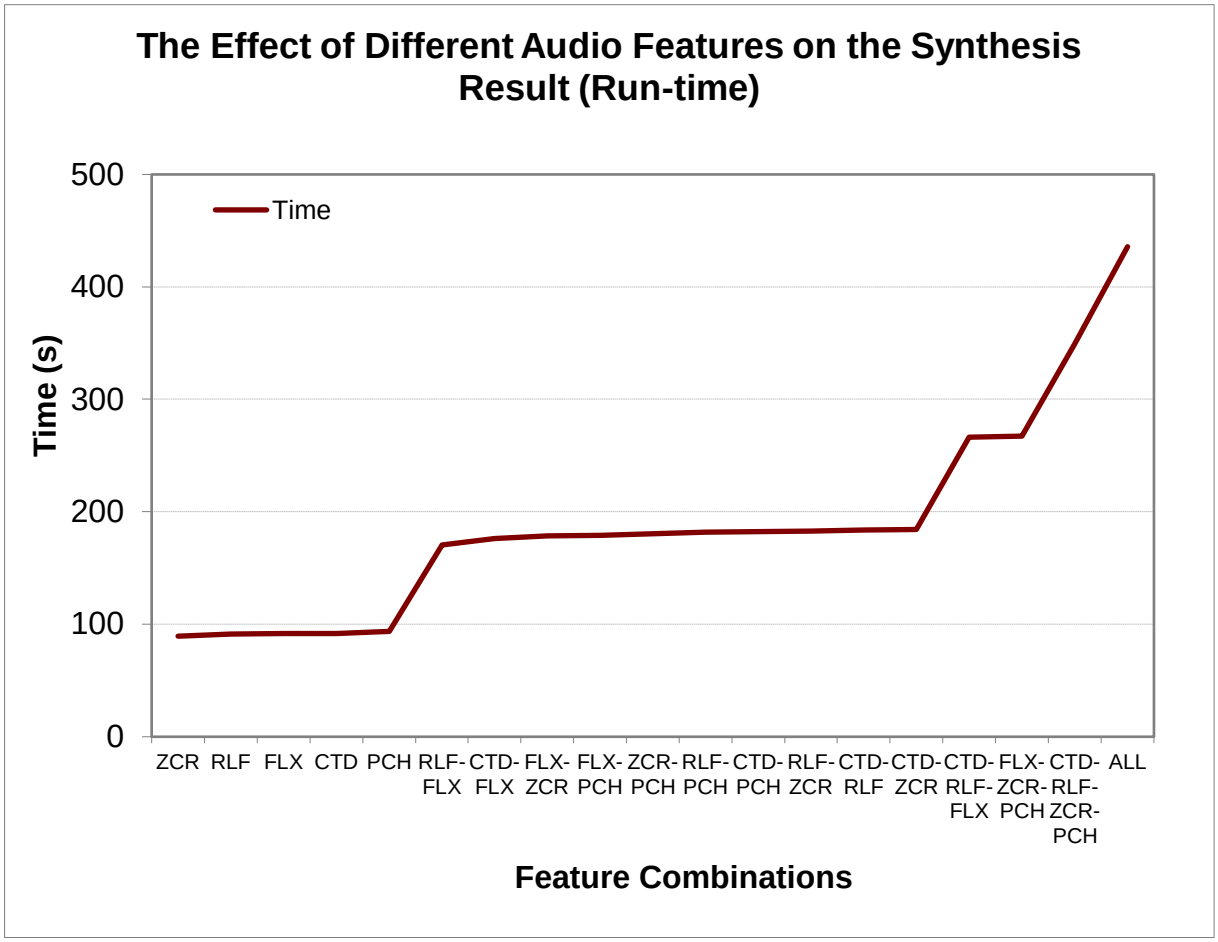


Figure 46: Result of Different Audio Features on Synthesis (Run-time)

iii) Discussion

This test displays a case where more is not always better. It shows that the inclusion of too many features result in larger average target distances compared to using only single or dual features. This happens because when many features are involved, it is harder to find source segments that perfectly match all the feature values possessed by the target segment. However, in some cases where the involvement of more than one feature is required, multiple features can still be used, but it is advisable to keep the number of features to a minimum (below three). Adding too many audio features into a query not only makes it difficult to find the matching segments in the database, but is also a computationally expensive process which involves multiple extractions and comparisons, thus consumes more time.

5.2.2 The Effect of Order-Dependent Audio Features Selection on the Synthesis Result

i) Experimental Set Up

There were two tests carried out in this experimental set. The first test aims to demonstrate the difference between the synthesis results obtained through the implementation of order-dependent feature selection process and the results obtained without it. A minimum of two features was required in demonstrating this, and list of the feature combinations used in this test is given in Table 17. For comparison's sake, the control sets (non-order dependent) were assumed to have no differentiating importance assigned to them, whilst the test sets always assumed that centroid was extremely more important than its partner feature, thus the value '9' was assigned to the Comparison Value[15], $CV_{CTD\_otherFeature}$ in all cases. $CV_{otherFeature\_CTD}$ represents the reciprocal value of the other features compared against Centroid, where the *otherFeature* was either Rolloff, Flux, ZCR or Pitch.

[13]The handle used to describe the comparison value between different features is read as 'The importance of Feature A compared to Feature B is by X intensity (referring to the Fundamental Scale of Importance, Table 3, p.75)'. Thus, CV_{CTD_RLF} = 9 is translated as Centroid is extremely more important than Rolloff, whilst CV_{RLF_CTD}= 9 suggests that the opposite is true. The reciprocals are given in fractions, i.e. CV_{RLF_CTD} = 1/9 is automatically assigned if CV_{CTD_RLF} is established, implying that Rolloff is extremely less important than Centroid. The handle is read in this manner regardless of the number of features included in the comparison, so CV_{CTD_RLF} = 3, CV_{CTD_PCH} = 7, CV_{RLF_PCH}= 5 is read Centroid is moderately more important than Rolloff, Centroid is very strongly more important than Pitch, and Rolloff is strongly more important than Pitch.

Table 17: List of the feature combinations tested with assigned comparison value (importance) to determine the effect which order-dependent feature selection has on synthesis result

	Feature Combinations	Comparison Value (Control)	Comparison Value (Order-dependent)
1	CTD-RLF	None	CV_{CTD_RLF}= 9, CV_{RLF_CTD} = 1/9
2	CTD-FLX	None	CV_{CTD_FLX}= 9, CV_{FLX_CTD} = 1/9
3	CTD-ZCR	None	CV_{CTD_ZCR}= 9, CV_{ZCR_CTD} = 1/9
4	CTD-PCH	None	CV_{CTD_PCH}= 9, CV_{PCH_CTD} = 1/9

After establishing the differences in the synthesis results generated between the control and order-dependent feature selection, the second test intends to express the effect of assigning different comparison value or importance to the audio features within the use of an order-dependent feature selection. The effect was demonstrated in cases involving dual and triple features. Scenarios with various intensity of importance were simulated in both cases. Due to the extremely large possibilities, the simulated cases were neither done on all the available combinations of features nor in all the important permutations possible, but only on selected conditions as samples. The list of the feature combinations and their comparison values are provided in Table 18 (dual features) and Table 19 (triple features).

For both of the tests ran in this experimental set, the independent variables that were involved, but not directly affecting the result of this test, were kept the same as they had been in the previous test (*Country* for target file, *Indris* for source file, and onset mode for segmentation). Again, both average target distance and run time were measured.

Table 18: List of the feature combinations tested with assigned comparison value (importance) to demonstrate the effect of dual features in order-dependent feature selection on synthesis results

	Feature Combination	Comparison Value (Importance)	Reciprocal of Importance
1	CTD-RLF	CV_{CTD_RLF} = 9	CV_{RLF_CTD} = 1/9
2		CV_{CTD_RLF} = 7	CV_{RLF_CTD} = 1/7
3		CV_{CTD_RLF} = 5	CV_{RLF_CTD} = 1/5
4		CV_{CTD_RLF} = 3	CV_{RLF_CTD} = 1/3
5		CV_{CTD_RLF} = 1	CV_{RLF_CTD} = 1
6		CV_{CTD_RLF} = 1/3	CV_{RLF_CTD} = 3
7		CV_{CTD_RLF} = 1/5	CV_{RLF_CTD} = 5
8		CV_{CTD_RLF} = 1/7	CV_{RLF_CTD} = 7
9		CV_{CTD_RLF} = 1/9	CV_{RLF_CTD} = 9

Table 19: List of the feature combinations tested with assigned comparison value (importance) to demonstrate the effect of triple features in order-dependent feature selection on synthesis results

	Feature Combination	Comparison Value (Importance)	Reciprocal of Importance
1	CTD-RLF-ZCR	CV_{CTD_RLF} = 1, CV_{CTD_ZCR} = 1, CV_{RLF_ZCR}= 1	CV_{RLF_CTD} = 1, CV_{ZCR_CTD} = 1, CV_{ZCR_RLF} = 1
2		CV_{CTD_RLF} = 3, CV_{CTD_ZCR} = 7, CV_{RLF_ZCR}= 5	CV_{RLF_CTD} = 1/3, CV_{ZCR_CTD} = 1/7, CV_{ZCR_RLF} = 1/5
3		CV_{CTD_RLF} = 1, CV_{CTD_ZCR} = 9, CV_{RLF_ZCR}= 7	CV_{RLF_CTD} = 1, CV_{ZCR_CTD} = 1/9, CV_{ZCR_RLF} = 1/7
4		CV_{CTD_RLF} = 5, CV_{CTD_ZCR} = 1, CV_{RLF_ZCR}= 7	CV_{RLF_CTD} = 1/5, CV_{ZCR_CTD} = 1, CV_{ZCR_RLF} = 1/7

ii) Results

The results of the first part of this test are presented in Figures 47 and 48. Figure 47 compares the average target distances between the control set and order-dependent feature selection. Three out of the four feature combinations tested showed that when the weight of centroid was set to extremely important (w_{CTD_RLF}= 9), the average distances between the newly synthesised sounds and the original target sound queried had been reduced. The CTD-PCH combination in particular showed significant reduction in the distance after the use of weights. In addition, the order of performance remained similar to the previous test (Section 5.2.1, p. 152), in which CTD-FLX returned the closest match, followed by CTD-RLF, CTD-ZCR AND CTD-PCH in descending order.

The run-time of the order-dependent set took slightly longer than the control set (Figure 48). However, in this dataset the difference was quite small, with the average increase of 33.8 seconds. In fact, for the CTD-PCH combination, the use of weight had actually resulted in a marginally faster run-time.

The difference in the outcome of the synthesised sounds that were generated from both the control and order-dependent sets can be listened to in Appendix A10 and the percentages of how much each sets differ from one another is displayed Table 20.

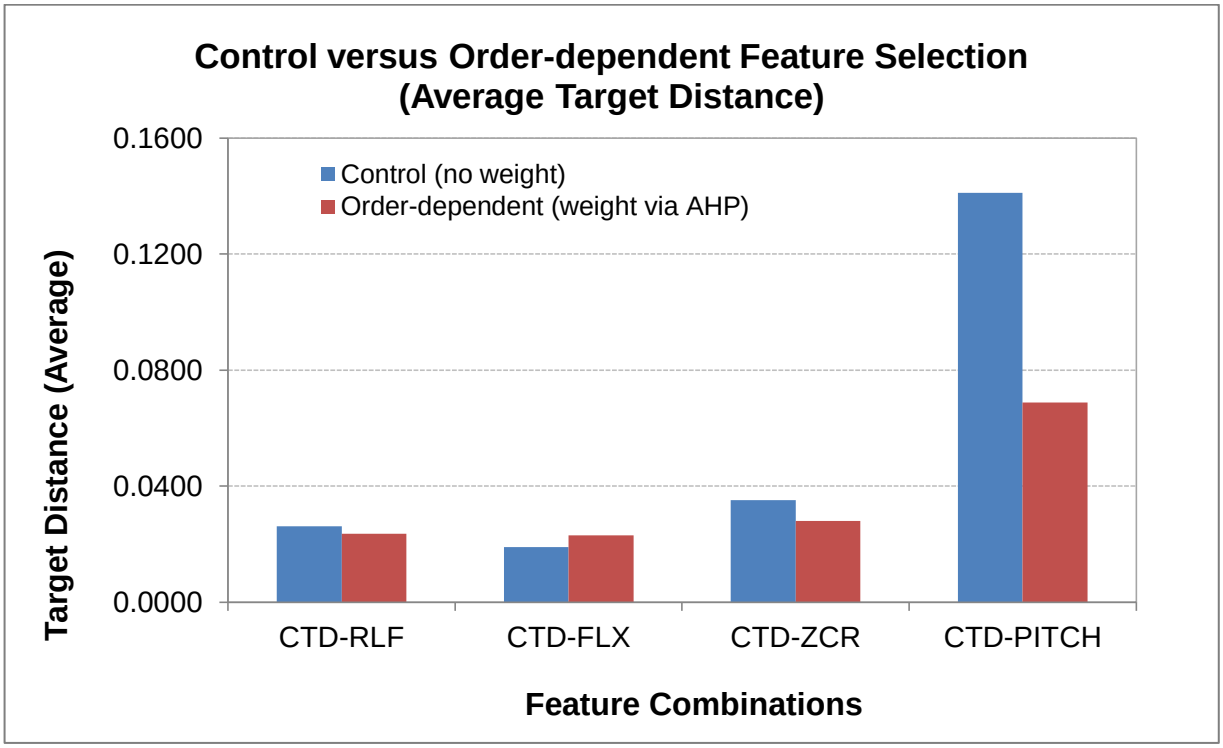


Figure 47: Result of Non-weighted Feature Selection against Order-dependent Feature Selection (Target Distance)

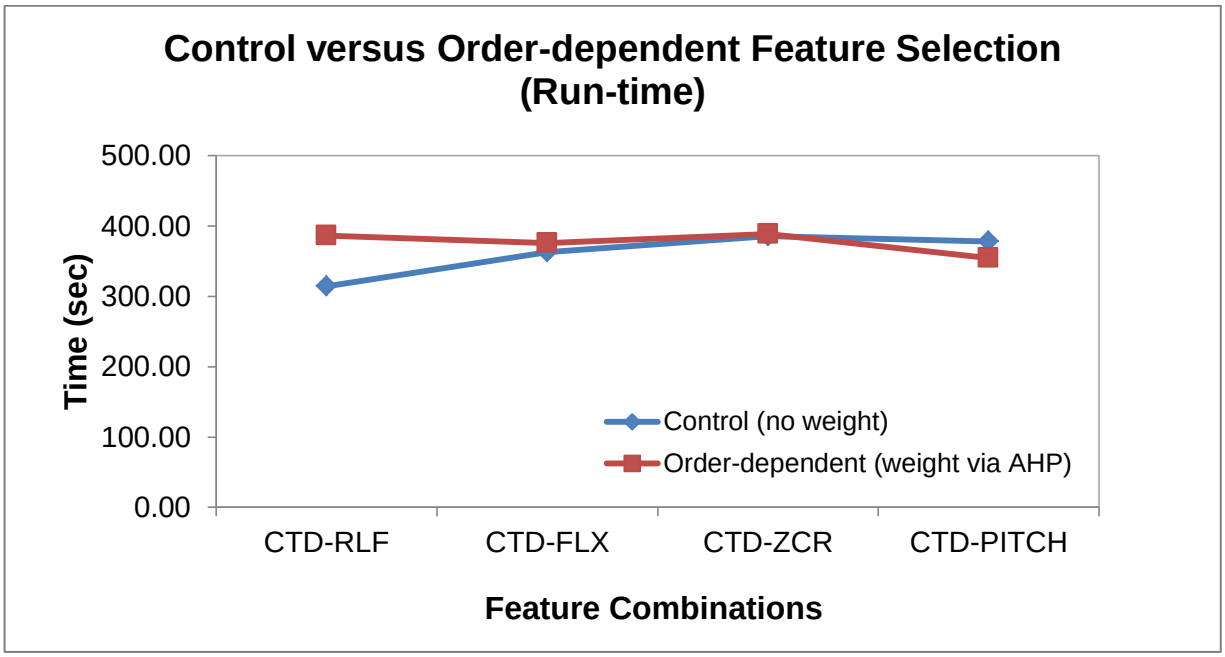


Figure 48: Result of Non-weighted Feature Selection against Order-dependent Feature Selection (Run-time)

Table 20: Result of the segment differences between Control and Order-dependent sets

Feature Combinations	Total different segments between the synthesised sounds (Control versus Order-dependent)	Percentages
CTD-RLF	22	55%
CTD-FLX	2	5%
CTD-ZCR	14	35%
CTD-PCH	7	18%

Results of the second part of this test demonstrated the effect of dual and triple features in order-dependent feature selection on the synthesis results and are presented in Figures 49 and 50 separately. In the case of dual features, the nature of progression can be observed as the importance of intensity was tested from all ends, such as from Centroid being extremely more important than Rolloff, to Centroid being extremely less important than Rolloff. In this case, the target distance was closer when higher importance was placed on Centroid, and became gradually larger as the importance shifted to Rolloff. However, the opposite was true for run-time, although the difference was only in the range of fifty seconds. The difference between the two extremes (i.e. CV_{CTD_RLF} = 9 andCV_{CTD_RLF} = 1/9) can be heard in Appendix A11 and in actual, the composition of segments that made up these two synthesised sounds differed by a massive amount of 72.5%.

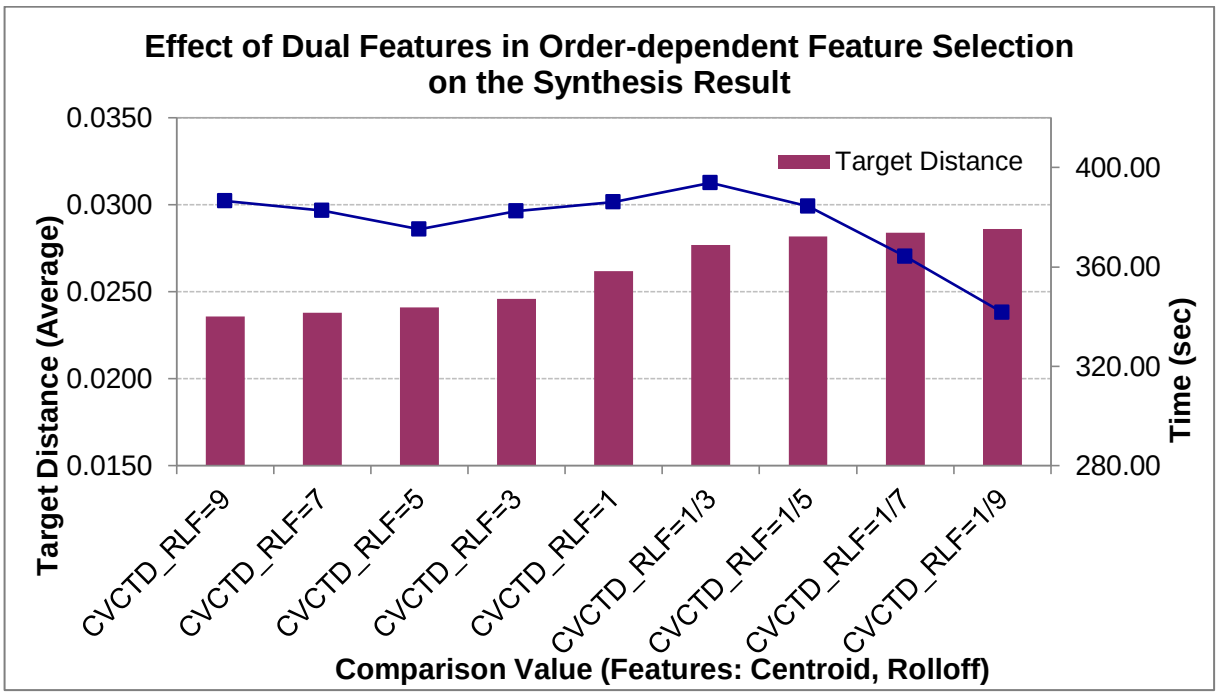


Figure 49: Result of Dual Features in Order-dependent Feature Selection (Target Distance and Run-time)

With triple features (and more), the algorithm implemented using AHP dictated that all comparison values were consistent (Consistency Ratio < 0.100). With the exception of the feature combination CV_{CTD_RLF} = 5, CV_{CTD_ZCR} = 1, CV_{RLF_ZCR}= 7 (CR = 1.6298) which was automatically eliminated by the system from being run in the simulation. All others values were found to be consistent.

It was found that the target distance was the closest when higher importance was assigned to Centroid (CV_{CTD_RLF} = 3, CV_{CTD_ZCR} = 7, CV_{RLF_ZCR}= 5) and progressively worsen as the importance of Centroid were lowered, the worst is seen in CV_{CTD_RLF} = 1/3, CV_{CTD_ZCR} = 1/7, CV_{RLF_ZCR}= 1/5. The run-time performance however, was in the opposite manner, where the lesser importance of Centroid resulted in a faster run-time, similar to the case pattern exhibited earlier with dual features. Also, it was noticed that the change from dual to triple features had resulted in a rise of run-time by approximately 200 seconds.

Another noteworthy discovery was that assigning the same order of importance to the features, regardless of the intensity difference would result in very similar outcomes both empirically and perceptually, (refer Appendix A12 for sound comparison). For instance, (CV_{CTD_RLF} = 3, CV_{CTD_ZCR} = 7, CV_{RLF_ZCR}= 5) and (CV_{CTD_RLF} = 1, CV_{CTD_ZCR} = 9, CV_{RLF_ZCR}= 7) both had different importance of intensity assigned to each features, but because they were in the same order, i.e. CTD>RLF, CTD>ZCR, RLF>ZCR, both had very close average target distance and almost identical sounds were synthesised.

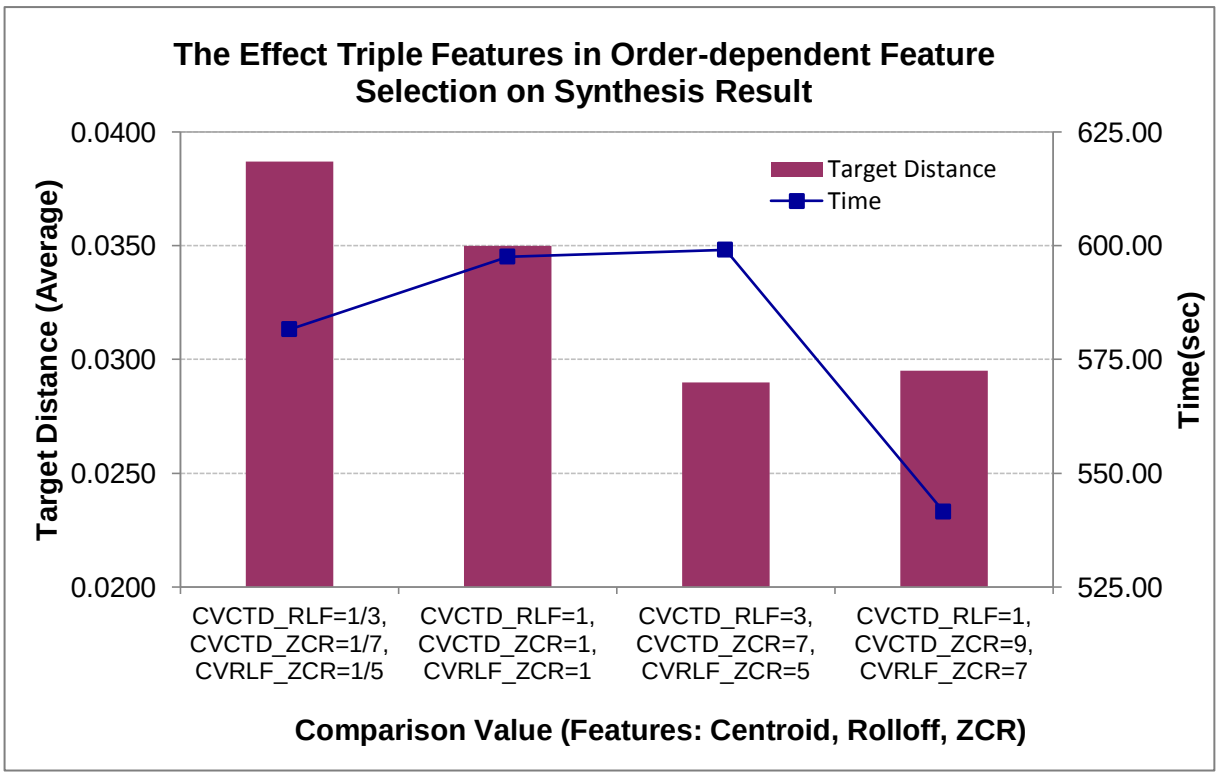


Figure 50: Result of Triple Features in Order-dependent Feature Selection (Target Distance and Run-time)

iii) Discussion

Although it is impossible to objectively deduce whether the sounds generated via order-dependent feature selection are better or worse compared to the sounds generated through basic feature selection (control set), it was encouraging to note the

reduction in the average target distances when order-dependent feature selection was used. This implied that the assignment of weight via AHP does, in fact improve synthesis results empirically, at least in the dataset tested. The pattern exhibited in this dataset suggests that perhaps the sound units in the database had more segments with values that match closer to Centroid than the other audio features. This was also true in the case involving dual and triple features, where the target distance was always smaller when heavier importance was assigned to Centroid than the rest of other features.

The output generated using order-dependent feature selection could differ as much as more than half of the total segments used in the generation of sound through the control set, which can be quite evident aurally too. Hence, when using concatenative sound synthesis, knowing which features have more precedent over the others and assigning suitable weights can help increase the possibility of creating sounds that match better to their targets. This is a great surplus to the existing CSS system, especially since no significant run-time drawback is seen when order-dependent feature selection is implemented.

With dual or triple features, it is evident that by changing the order of the features' importance, different results will be obtained. However, if they have the same general order of features, the outcomes of both cases will be fairly similar, regardless of the intensity of each individual features. This means that users need not be burdened with the task of guessing the exact weight to assign for each feature, but only suffice to know the order of importance between the features, as the changes in intensity affect the final outcome only minimally.

Another improvement seen through the use of order-dependent feature selection via AHP is that only sounds that are based on consistent and reliable features judgments are synthesised by the system, whilst all inconsistent judgments are flagged up and rejected earlier on. This way, users can be sure of the judgment given during the query stage is a sound one, and any mistakes can be rectified immediately.

5.2.3 Conclusion

This second experimental set investigated the effects of audio features and the order-dependent feature selection approach on the overall synthesis results via CSS. The main findings from this experimental phase are:

i) *Different features lead to different results*

Just as the inclusion of different input parameters was found to affect synthesis results, the use of selection of different audio features does too. Some features are found to perform better than others, for instance in this test, Flux and Centroid were the better features whilst Pitch was identified as the worst. However, for any CSS system, the inclusion of other features can be easily added or removed from the user option to suit the individual needs. The performance of the features is thus constrained to the features included in the task and also to the target and source sounds loaded into the database.

ii) *More (audio features) does not always mean better*

The likelihood of finding source segments with feature values that match exactly that of the target segments is already small, but by increasing the number of features that must be matched, the probability of this happening is further reduced. In addition,

since the use of multiple features is expensive in terms of computational power and time, users should therefore have a reasonable justification in their decision to include more than two features during feature selection, especially when order-dependent feature selection mode is enabled which further extends to the use of feature order and weights.

iii) *The use of order-dependent feature selection improves synthesis results*

Although it is established that the addition of more features may not always return better results, there may be situations where there is a need for multiple features to be used. In this case, order-dependent feature selection is shown to improve synthesis results. This is because it may be easier for the system to find matching segments by focusing on one or two features that have been indicated to be more important than to try and come up with segments that match all the features equally. Though the run-time is slightly longer as a result of this, the reduction in the target distance between the target and source segments may be considered by users as a worthy trade off.

iv) *The features' order of importance is more important than the features' intensity importance*

Features assigned with different order of importance from one another are likely to result in two more diverse sounds than features which have been assigned with the same order of importance but only at varying intensity. This means that users are allowed some flexibility or a wider margin with their intensity judgment, provided that the order of importance between the features is known.

v) *Order-dependent feature selection combines both qualitative and quantitative approaches in the process synthesising new sounds*

Creating new sounds that are fitting to every user's expectation based on a few parameters is an almost impossible task. However, the use of AHP with its newly implemented order-dependent feature selection process has shown encouraging results both quantitatively and qualitatively. The selection process is represented by the form of numerical improvements and also by taking into account the subjective judgment of humans as part of the input in the process. Furthermore, only sounds that are proven to be based from consistent and reliable judgments are synthesised in this approach.

Taking into account the above findings, it can be concluded that the AHP is a suitable method to be implemented for the proposed framework involving order-dependent feature selection. Issues such as order of the features and the weights for respective features are tackled systematically via this method, and results obtained through its implementation are also very promising.

5.3 Phase 3: Search and Selection Evaluation

This third phase of evaluation looks at the issues surrounding search and selection in a CSS system that may affect the synthesis result. Earlier, in Chapter 3 (Section 3.2.2, p.77), it has been shown that homosonic and equidistant segments are common occurrence in the database, especially when very few features are included in the feature comparison between target and source segments. Following this, the use of concatenation distance has been proposed as a solution to the problem (Chapter 4, Section 4.2.2, p.105).

It is in this third experimental set that the feasibility and the efficiency of concatenation distance as a solution to solve the challenges involving homosonic and equidistant segments are tested and measured. Two tests have been designed and carried out to determine this.

5.3.1 The Effect of Enabling Concatenation Distance to Overcome Homosonic Segments on the Synthesis Result

i) Experimental Set Up

To examine the feasibility and efficiency of concatenation distance in overcoming problems caused by homosonic segments and how this affected synthesis result, a bench mark test was conducted. The idea was to determine if the system was able to locate and select the exact same segments as queried through the target segments, if all of the segments that make up the target sounds were available in the source segment database. For this to happen, both target and source sounds used were the same one, which was the *Country* file, and another sound file, *Classical* was added into the source sound database to produce the homosonic segments effect. Centroid was

the audio features used to match the target and source sounds, and the onset mode was selected for segmentation.

The distribution of the homosonic segments contained in the dataset for this test is given in Figure 51. From the chart, it can be seen that out of the forty segments of the queried target sound, twenty-seven of them had at least two homosonic segments with equal potential being selected. For example, Target Segment #2 had three homosonic segments to choose from; whilst Target Segment #6 had five homosonic segments to choose from. This not only displays the distribution of the homosonic segments in this test set, but also reinforces the point that a solution is needed to handle unit selection involving homosonic distances, as it is a very common occurrence, as demonstrated here.

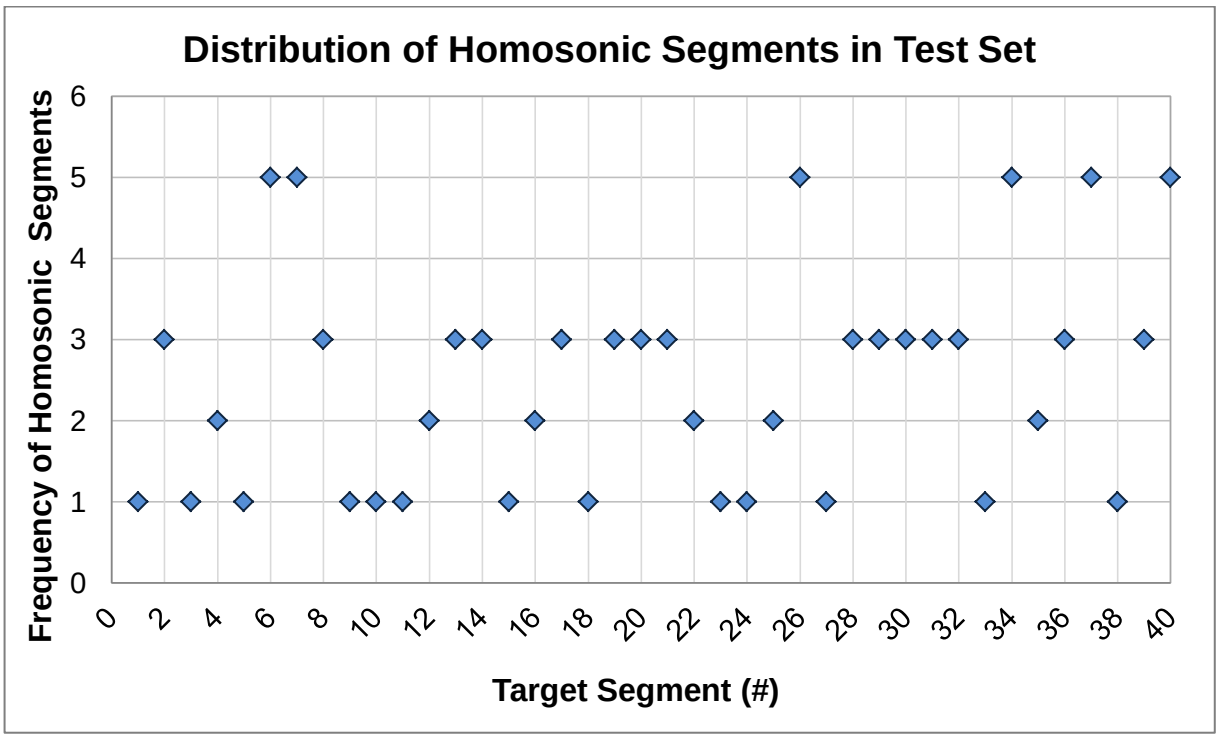


Figure 51: Distribution of Homosonic Segments in the Test Set

Since the same target distance was expected between all homosonic segments, the average target distance was not measured in this test, but was replaced with the concatenation distance (to observe the smoothness or flow of the sound at the joint between the segments), as well as the result accuracy (the ability to correctly select the right target segment) between the concatenation distance-enabled mode and the concatenation distance-disabled mode. The run-time between the two modes was also measured.

ii) Results

The results from this test set were measured in the form of target and concatenation distances, segment accuracy and waveform comparison between concatenation distance-enabled mode and concatenation distance-disabled mode. Firstly, the target and concatenation distances of the two modes were compared (Figure 52). No difference was seen between them regarding the target distance. This was expected when homosonic segments were present. On the other hand, the concatenation distance was significantly lowered when concatenation distance was enabled. This suggests that the performance of concatenation distance-enabled mode had managed to obtain better result where smoother sound flow was produced.

This was further supported by the result of segment accuracy displayed in Figure 53. As a benchmark test, all of the target segments were made present, along with segments from other sounds that made up the source sound in database. When the concatenation distance mode had been enabled, 80% of these targets were successfully located and selected. In comparison, only 32.5% of the segments were correctly selected when the mode was disabled.

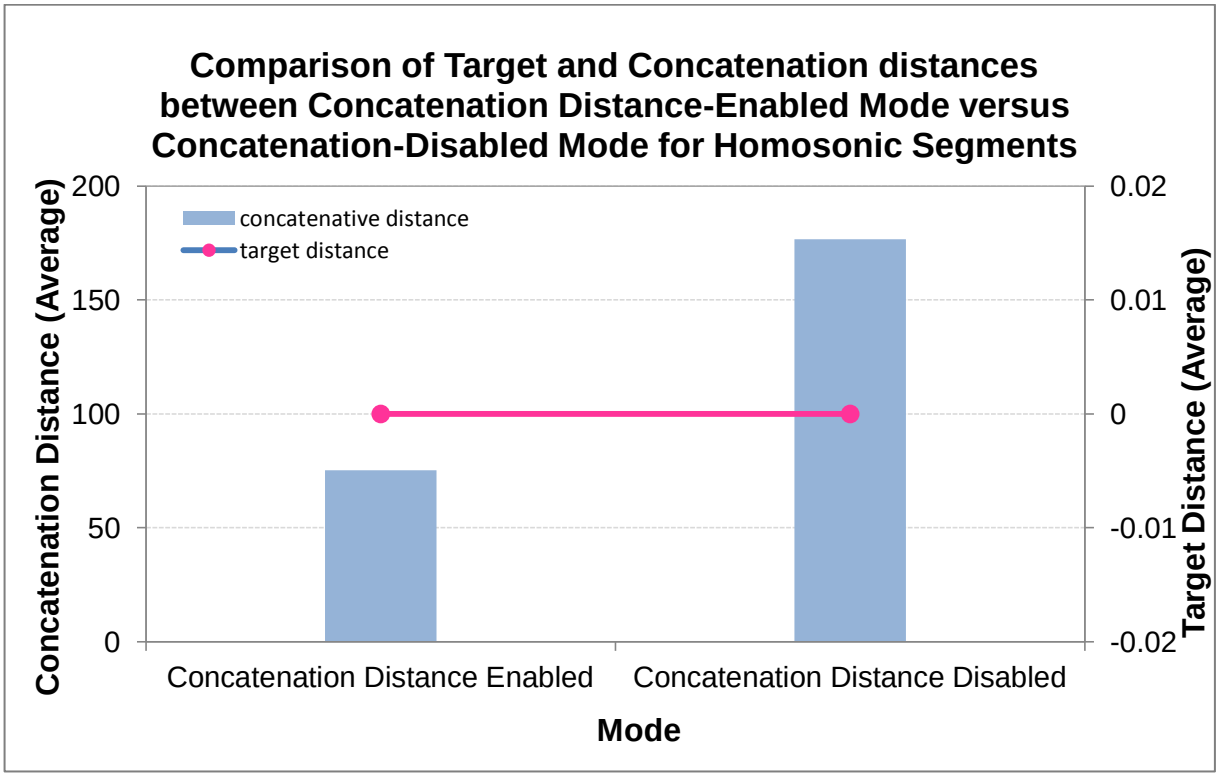


Figure 52: Result of Concatenation and Target Distances between the Two Concatenation Modes for Homosonic Segments

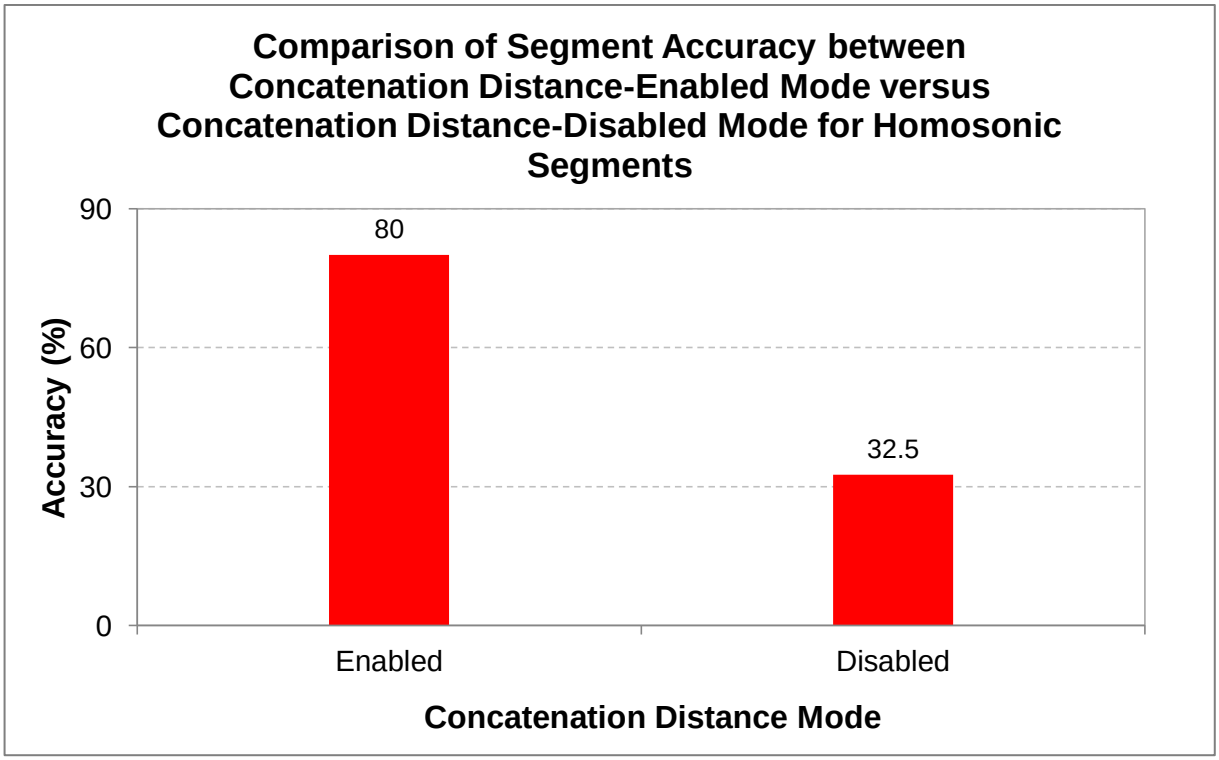


Figure 53: Result of Segment Accuracy between the Two Concatenation Modes for Homosonic Segments

The waveforms in Figure 54 further emphasise the result from this benchmark test. The top row is the waveform of the original target sound. The middle row is the waveform that resulted from the concatenation distance-enabled mode, whilst the waveform in the final row resulted from the concatenation distance-disabled mode. From the figure, it is evident that by enabling the concatenation distance mode, the system generated sound that was more similar to the original target than it had when the mode was disabled. Sounds for all three waveforms can be listened to and compared in Appendix A13.

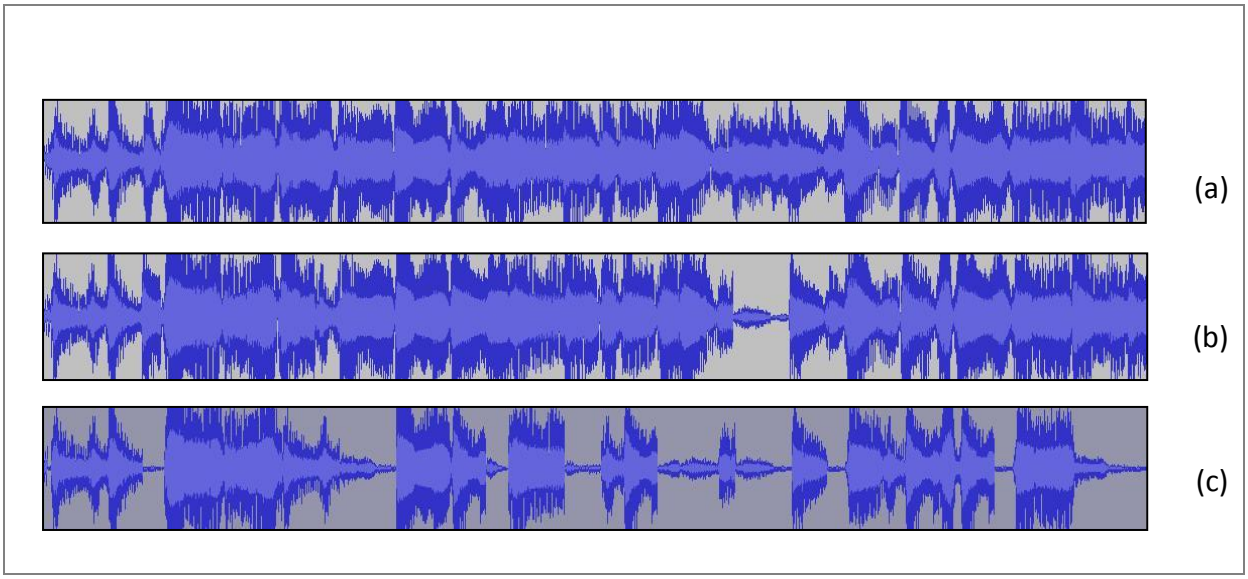


Figure 54: Waveform Comparison between (a) Target Sound; (b) Sound Synthesised by Concatenation Distance-Enabled Mode; and (c) Sound Synthesised by Concatenation Distance-Disabled Mode for Homosonic Segments

These improvements did, however, occur at the expense of run-time cost, where concatenation distance-enabled mode took almost six times as long to run (Figure 55).

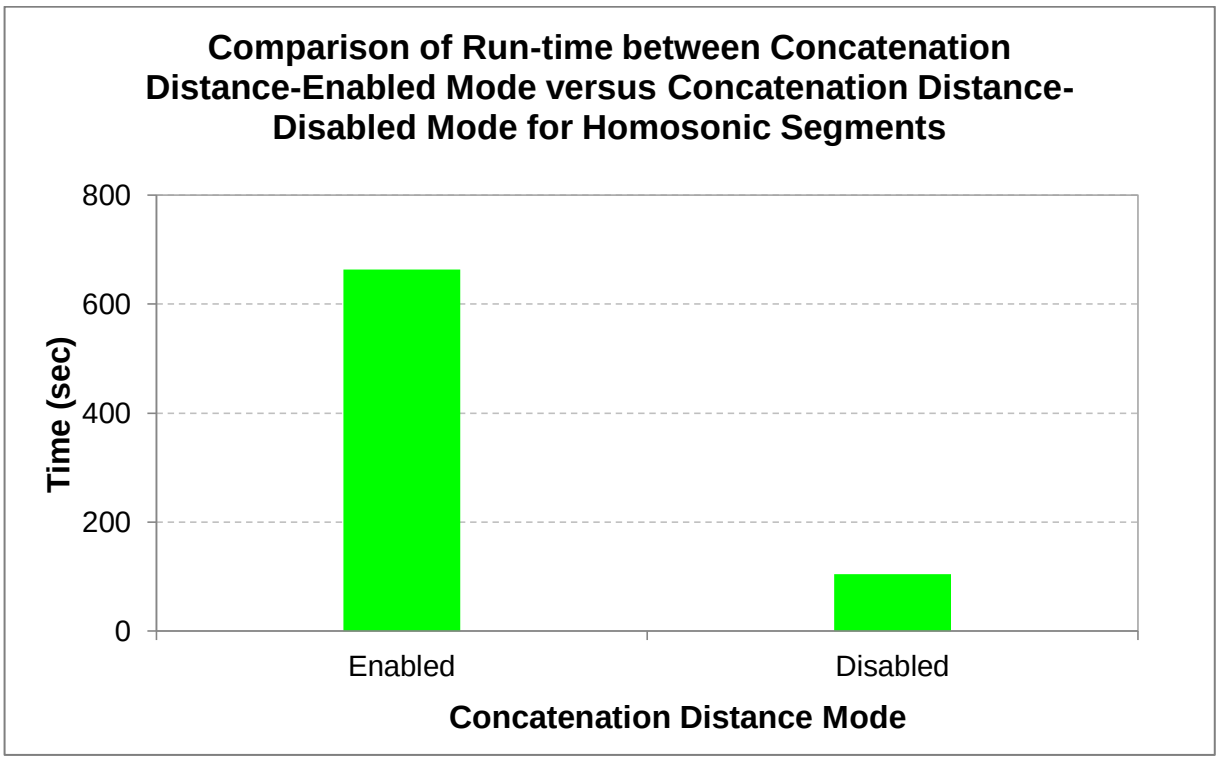


Figure 55: Result of Run-time between the Two Concatenation Modes for Homosonic Segments

iii) Discussion

In general, the use of concatenation distance has been shown to be both feasible and efficient in solving problems with homosonic segments. Enabling the concatenation distance mode has reduced the concatenation distance by more than half. This means that the sounds generated are more fluid and smooth, as implementation dictates that only segments with beginning pitch value that closely matches the pitch value at the end of the preceding segment is chosen. However, the concatenation distance was not zero (as would be expected for perfect concatenation distance). This was a direct result of the manner in which the pitch values were extracted. Instead of extracting the pitch at every millisecond, pitch extraction was done at a set interval along the signal, as to reduce the data load. This action causes a little discrepancy between the end value of a segment and the beginning value of a segment, even the two segments

occur consecutively as target segments. Nevertheless, this variance is very minor and does not affect the overall synthesis result.

The concatenation distance-enabled mode had also scored higher in the segment accuracy test, where 80% of the original target segments were located in the database, compared to a very low 32.5% when the function was disabled. The poor performance of the concatenation distance-disabled mode was attributed to the mechanism it took to handle homosonic segments which is random selection. As such, as long as the closest target distance is satisfied, segments are chosen without any regard for their concatenation distance. However, the 100% segment accuracy had not been achieved in this test, as would be the ideal case, because there were parts in the target segment that allowed discontinuity, for example during an attack. This 'attack' happened at several points along the target file and is accounted for the discontinuity. In any case, the concatenation distance-enabled mode without a doubt had outperformed the concatenation disabled-mode with respect to segment accuracy.

The only weakness of the concatenation distance-enabled mode was that it took longer for the sounds to be generated. This is understandable, given that in this particular dataset, almost three quarter of all the target segments had two or more homosonic segments. Occurrence of homosonic segments meant the concatenation distance needs to be calculated for each segment with the same sonic values, and after comparing these segments, the segment with the least concatenation distance was then selected.

Again, it is difficult to ascertain which of the sounds produced via the enablement or disablement sounded better, as it is a highly subjective and personal matter. However,

by enabling the concatenation distance mode, the results have been improved numerically, as the system was able to select the intended segments 80% of the time, which is an impressive feat.

5.3.2 The Effect of Enabling Concatenation distance to Overcome Equidistant Segments on the Synthesis Result

i) Experimental Set Up

The objective of this test is similar to that of the previous one conducted, it is to examine the feasibility and efficiency of concatenation distance in treating equidistant segments during the search and selection phase, and how its use affects the synthesis result. In this test, the *Country* file was kept as the target sound, whilst the source sound was changed to the *Rainforest* file, as this had the most frequently occurring equidistant segments in all of the sounds collected for the entire study (twenty-three out of sixty-nine segments in the *Rainforest* dataset were equidistant segments). The distribution of the equidistant segments can be referred in Figure 56 shows that ten out of forty segments from the target sound (*Country*) had at least two equidistant segments from the source sound (*Rainforest*) with equal potential being selected. It can also be seen that the highest frequency of equidistant segment occurring at Target Segment #1 (four equidistant segments). Centroid was the single feature used as the basis of comparison between the target and source sounds, and segmentation was performed in the onset mode.

Similar to the homosonic segments, equidistant segments are expected to have the same average target distance from one another, so it was not an indication of performance between the concatenation distance-enabled mode and the

concatenation distance-disabled mode. The concatenation distance was compared instead, as well as the run-time between the two modes.

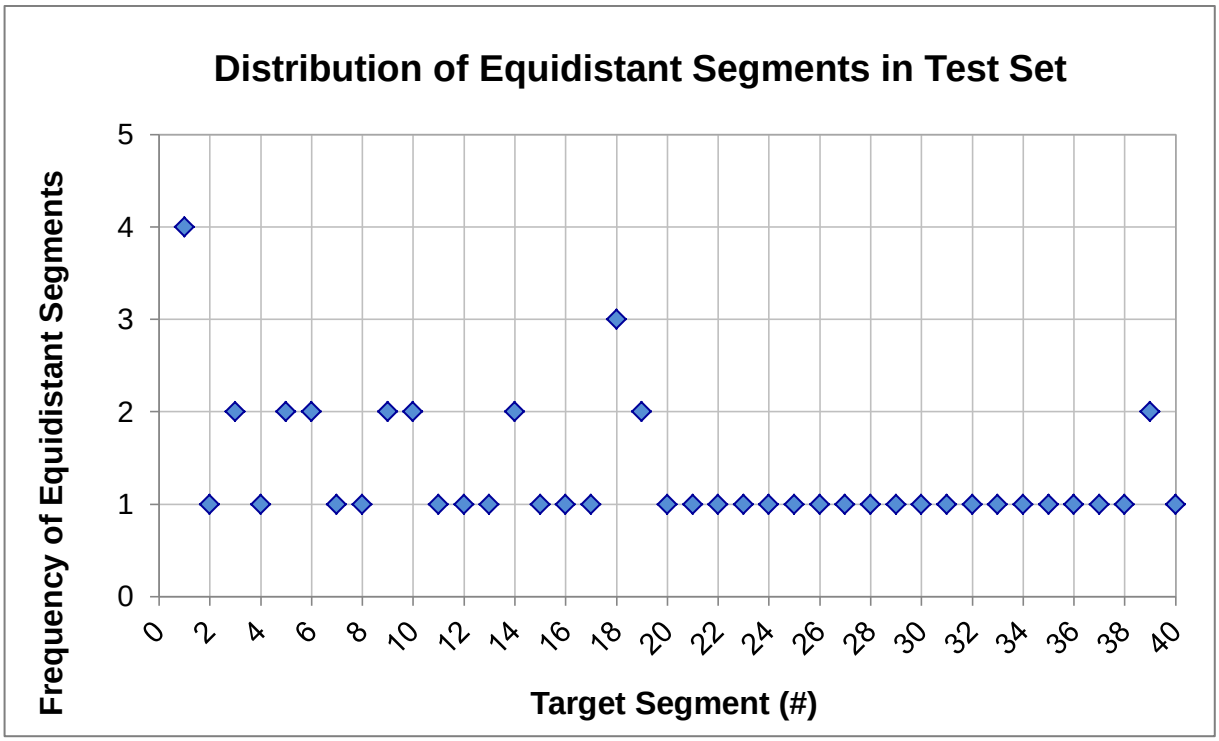


Figure 56: Distribution of Equidistant Segments in the Test Set

ii) Results

As far as target distance was concerned, there was no effect from the use of either modes and the average target distances in both cases were 0.0009. This had been expected with equidistant segments, as even when different source segments were selected by the system, their feature values which the target distance had been based on were still the same. On the other hand, a small drop in the concatenation distance was noticed during the concatenation distance-enabled mode (Figure 57).

Unlike the bench test carried out in the previous test, different target and source sounds were used in this test. Therefore, it was unreasonable to expect that the waveforms representing the target and synthesised sounds to be the same. However,

the waveforms in Figure 58 do show the slightest differences in the sounds synthesised by the two modes. The audio for these two results is attached in Appendix A14.

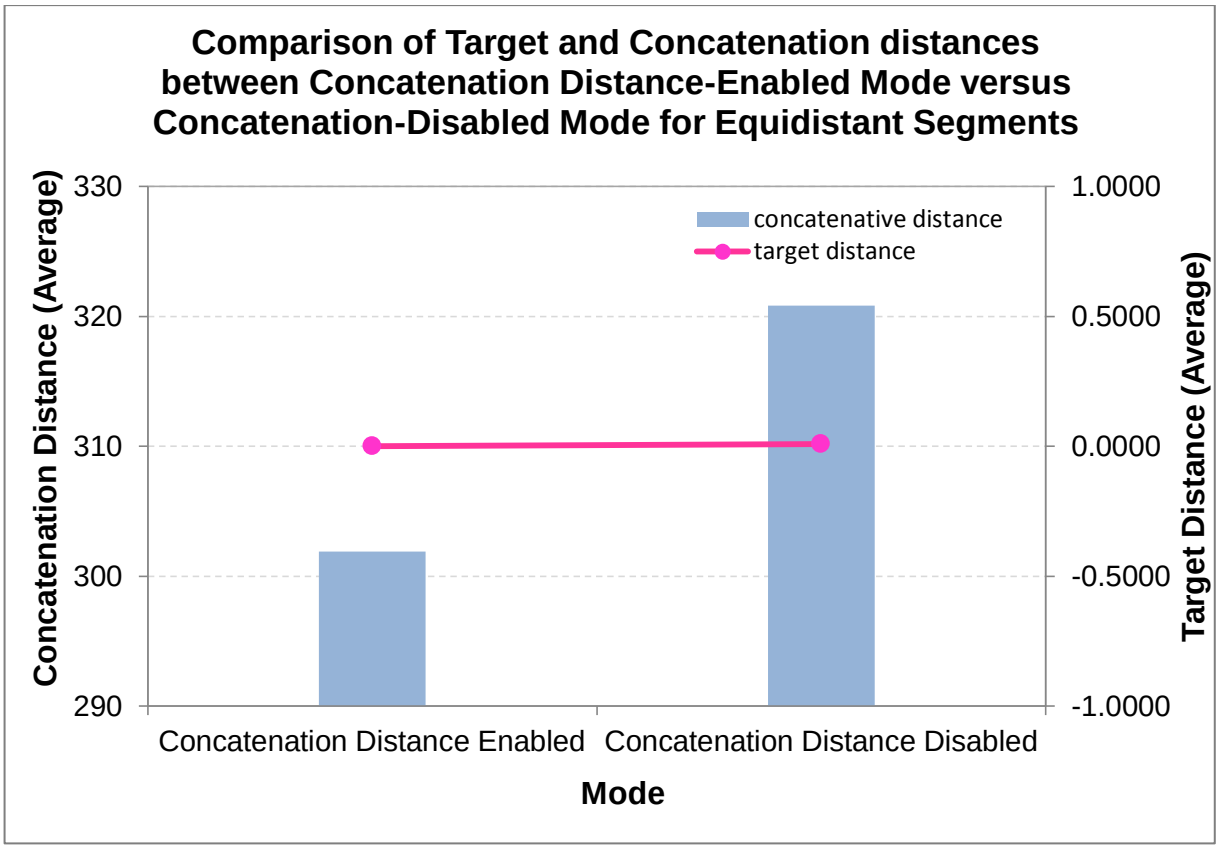


Figure 57: Result of Concatenation and Target Distances between the Two Concatenation Modes for Equidistant Segments

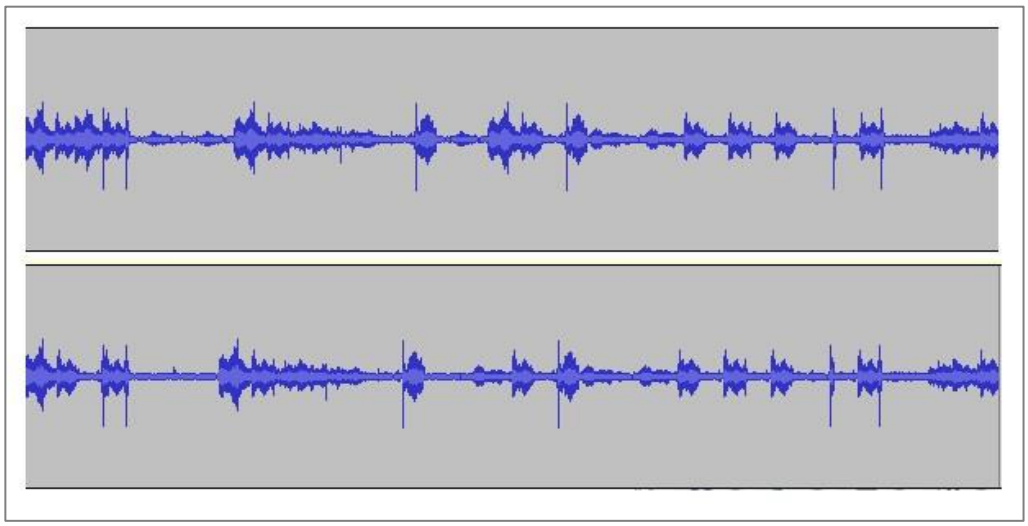


Figure 58: Waveform Comparison between Sounds Synthesised by Concatenation Distance-Enabled Mode (top); and Concatenation Distance-Disabled Mode (bottom) for Equidistant Segments

Figure 59 indicates that concatenation distance-enabled mode took longer to run than the opposing mode. However, in comparison to the previous result with homosonic segment (Section 5.3.1, Figure 44, p. 169), the run-time between concatenation distance-enabled and concatenation distance-disabled mode was significantly faster, despite being implemented on the same algorithm. It is thought that this is due to the smaller occurrence of equidistant segments present in this dataset (only ten), compared to twenty-seven occurrences of homosonic segments in the previous test. The higher the occurrence of these segments in the database, the more the system has to include concatenation distance into its calculation, which ultimately adds up the run-time. This suggests that the number of homosonic or equidistant segments that occur in a dataset also affects the run-time.

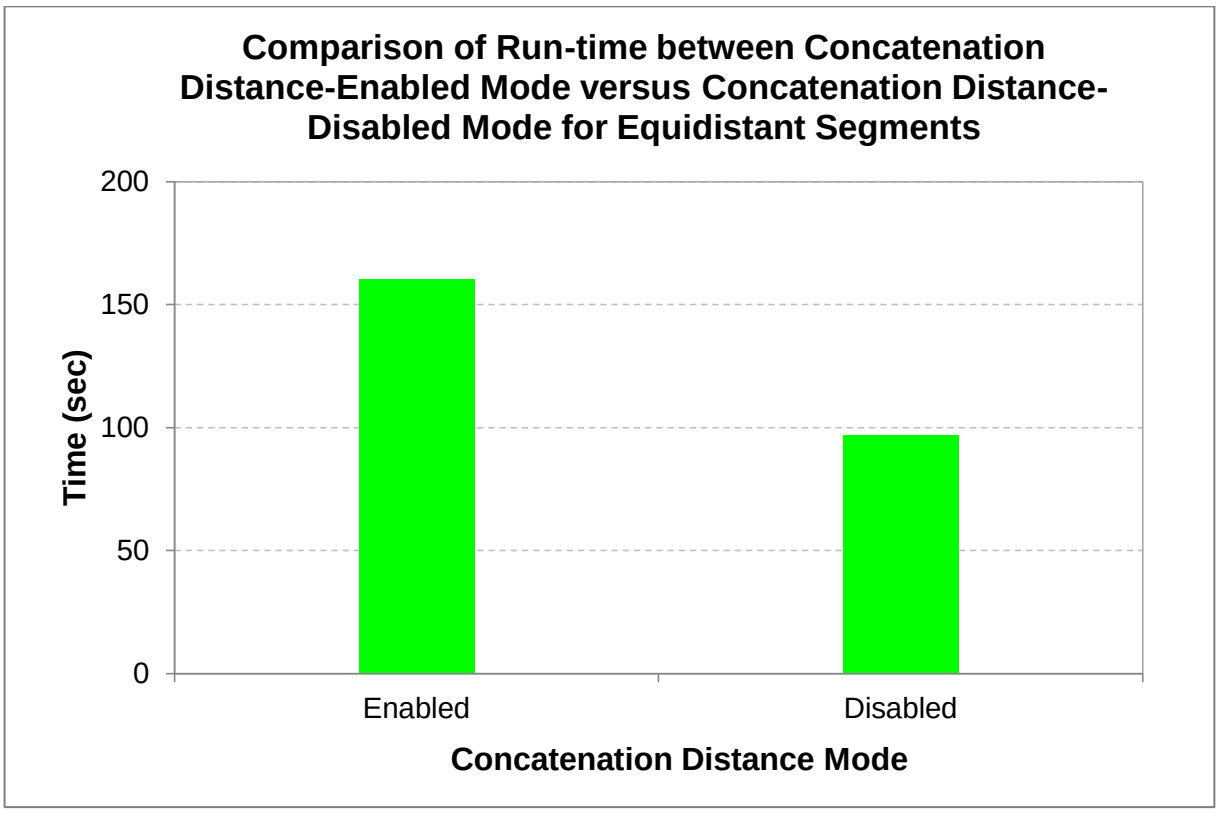


Figure 59: Result of Run-time between the Two Concatenation Modes for Equidistant Segments

iii) Discussion

Although the use of concatenation distance-enabled mode for equidistant segments has only managed to reduce the concatenation distance by a small percentage, this does not mean that its impact is not significant. Several reasons were enlisted to explain why the impact appeared less convincing. It is first thought that perhaps this particular dataset did not have sufficient equidistant segments to portray the merit of concatenation distance, despite it (*Rainforest* file) being the one with the most equidistant segments in the entire collection. For a dataset with higher occurrence of equidistant segments, the gap in the concatenation distance between the two modes may well become larger.

Secondly, it was later found that, by complete chance that in this dataset, the best segments to select even after concatenation distance was calculated, was the first source segment in the database. As a result, synthesis results of the modes were quite similar, as the default setting of the basic mode was designed to function by selecting the first source segment to appear in the database.

Concatenation distance-enabled mode did result in an increase to the run-time, but the difference was deemed acceptable given the improvement it had resulted in the concatenation distance and that the run-time had not even doubled. Moreover, when compared to the run-time result from the previous test for homosonic segments, the run-time result of this test with the equidistant segments was much faster. This was because the number of homosonic segments was higher in the previous dataset than the number of occurring equidistant segments in this set, and subsequently requiring more concatenation distance comparison to be made. This supports the earlier proposal of calculating the concatenation distance based on the novel hierarchical

model as opposed to the non-hierarchical model currently implemented in the existing CSS systems. Following the hierarchical model means that the unit selection process only calculates the concatenation distance when homosonic or equidistant source segments (potential matches) are found for a particular target segment, and proceeding with only target distance comparison when there are no homosonic or equidistant segments for the target segments. In comparison, the current non-hierarchical model calculates the concatenation distance for all target-source segments comparison, once the mode is enabled. This novel hierarchical model allows precious run-time and processing powers to be saved.

5.3.3 Conclusion

This third experimental set investigated the practicality and efficiency of using concatenation distance to solve challenges involving homosonic and equidistant segments during the unit search selection process, and how its use affects the synthesis results. Several conclusions that can be derived from the tests conducted in this phase are:

i) *Concatenation distance provides a feasible and effective solution for selection involving homosonic and equidistant segments*

The tests carried out in this phase have shown that concatenation distance can be used as a solution to overcome the challenges faced by the CSS system when challenged with homosonic or equidistant segments. It is able to make a more intelligent decision over which source segments to select in the case where several of them possess the same target distance from the target segment. By comparing the concatenation distance of these equally fit segments, the selection is drawn through

the second layer filtering. In addition to synthesising sounds with smoother transitions from one segment to another (lower concatenation distance), this method is also capable of doing it with high accuracy compared to when the concatenation distance mode is not in use. This is evident in the bench mark test results in Section 5.3.1, p.169.

ii) *Hierarchical model is the way forward*

One of the tests in this experimental set had shown that run-time is longer when the concatenation distance mode is enabled. The additional time to complete the task is expected, as enabling the mode means the concatenation distance needs to be calculated in addition to the target distance. However, with the hierarchical model in place, concatenation distance need not be calculated on all of the segments (as exercised by other CSS systems which include the concatenation distance option), but to only calculate the concatenation distance of the segments that are identified as either homosonic or equidistant. This cuts down the unnecessary processing power and time required. The hierarchical model is especially useful since in the same test it was also discovered that the run-time had in fact increased in the same proportion as the number of homosonic or equidistant segments contained in the dataset.

This solution is not without limitations. There are a few limitations of the approach proposed:

i) *Concatenation distance is never completely zero even when concatenation distance mode is enabled*

As explained earlier, this is primarily due to the pitch extraction algorithm, where it was extracted at certain intervals, resulting in a small difference in the pitch value

between one end of a segment and a beginning of another. This problem is easily resolved with a little change pitch extraction settings.

ii) *Concatenation distance-enabled mode sometimes overlooked the attack event attack in target segments*

The practice of always selecting the segment with the lowest concatenation distance when confronted with homosonic or equidistant segments does not always result in favourable matches, as even between the target segments, there can be large concatenation distance between two segments, due to event such as an 'attack' happening at that point. Concatenation distance-enabled mode does not register the occurrence of the attack, and continues to choose the segment with the lowest concatenation distance.

Despite its shortcomings, concatenation distance as a solution to homosonic and equidistant segments in the unit selection process provides a novel alternative to the practices of other CSS systems. On existing systems, this selection was either done through random selection or by simply picking the first sound segment in the database list, resulting in favouring certain segments over others. Through this approach, this problem is not only remedied, but also executed well too. Though in the case presented, the synthesised sounds may not sound expressively different from the basic mode, the tests have nevertheless revealed many important findings such as those mentioned earlier.

5.4 Phase 4: Listening Test

In this fourth and final phase of the experiment, a listening test was performed. As previously demonstrated in Chapters 3 and 4, humans judged sound similarity differently; some judged similarities based on perceptual attributes such as loudness, some based it on the timbral qualities, whilst others may use other information such as tempo or the melodic contour of a sound to perform this task. Through the preliminary listening test carried out in Chapter 4 (p.113), it was found that the two most prominent audio features used by humans in judging sound similarities are the timbral information (musicians) and the melodic information (non-musicians). This final test intends to probe the issue further by searching the answers to these three questions: (1) Is there a correlation between the perceived sound similarity and the perceived interestingness of sounds in humans? (2) Do musicians and non-musicians make different perceptual judgments surrounding the sounds that they hear? (3) Do they exhibit similar characteristics or behaviour when passing a judgment over the sounds? Unlike the previous three experimental sets which were all computer-simulated, this test set involved human participants. The general description of the listening test is described below, followed by the results, further discussion and conclusion.

5.4.1 General Description

i) Dataset

Sounds that were included in this listening test came from several sources. Two were synthesised using this study's own system, *ConQuer*, three others were the product of synthesis through another CSS system, *MATConcat* (Sturm, 2004), one was a remix of a well-known song, and the remaining were ten seconds length songs that were neither a product of a synthesis or remix.

ii) Participants

Forty-one students and staff from the University of Plymouth had voluntarily participated in this listening test. Sixteen of them (seven females, nine males) were musicians or were studying music, whilst the other twenty-five (sixteen females, nine males) were all non-musicians. A simple Chi-squared test confirmed that no bias existed in terms of sex and musical background with this particular make up of participants (at $\chi^2(1) = 0.610$, $p<0.4349$ and at $\chi^2(1) = 1.976$, $p<0.1599$ respectively), which can be referred in Appendix C9 and C10. The design of this listening test had been consulted with experts from the field of applied cognitive psychology of sound and music and followed the informed practices in the area. This study received clearance from the Faculty of Arts and Humanities Ethics Committee and followed strictly the ethical guidelines and protocols set by them.

iii) Procedure

There were eight sets of sound in this listening test. Each set contained a target sound and another sound which was supposedly synthesised from that target sound. Participants were asked to listen to both the sounds and then make a subjective judgment on their perceived similarity between both sounds. They were also asked to rate the 'interestingness' level of the synthesised sounds that is how pleasant or amusing they found the sound that was synthesised from the target sound to be. A Likert scale as shown in Figure 60 was used for this purpose. Participants were allowed to replay the sounds as many times as they needed to.

The test had been designed so that the target-synthesised sound pairs heard by the users came from a mixture of sounds synthesised using several CSS systems and also

non-synthesised sounds. The first five sounds were assortment of synthesis results based on loudness, spectral and timbral content, and sounded granular-like. The last three sounds in the test were not actually synthesised sounds, but mainstream songs which had been chosen because the analysis on their melodic contour showed that they were melodically similar to their target sounds. The breakup of the sounds used in this test is presented in Table 21 and the sounds can also be referred in Appendix A15. In addition to showing if a particular audio feature is preferred as the matching criteria in similarity judgment, it would also show whether there was any correlation between the perceived sound similarity and the perceived sound interestingness by the participants from either groups. As products of *ConQuer* were also included in this listening test, its composition capability could be indirectly compared against existing CSS systems.

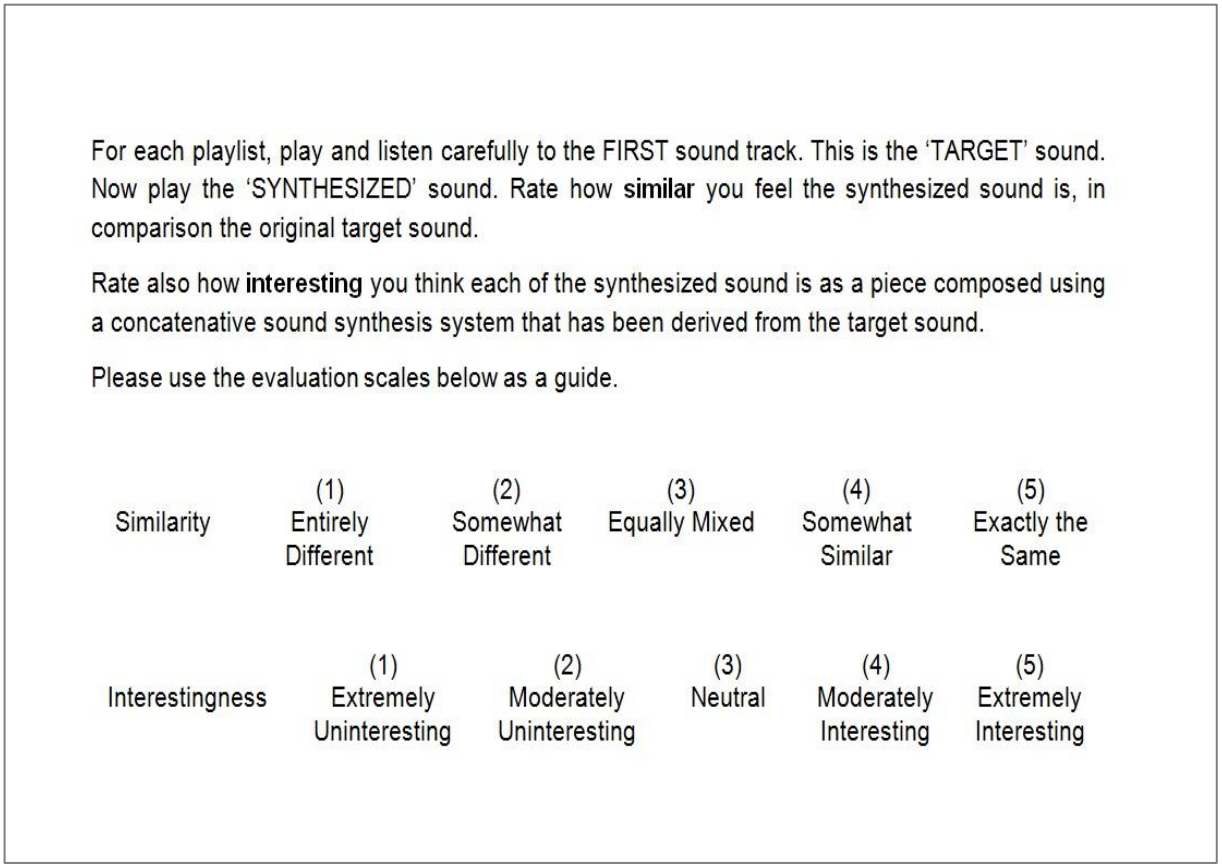


Figure 60: Likert Scale Used to Measure Perceived Sound Similarity and Perceived Interestingness

Table 21: Listening Test Sounds Breakup

	Target	Source	Matching Criteria	CSS System
1	Mahler, Ritenuto (2nd Symphony)	Monkeys	Loudness, Spectral Rolloff	*MATConcat*
2	Mozart, Sonata K 457 (3rd Mvmt)	Whales	Spectral Centroid	*ConQuer*
3	Meat Purveyors, Circus Clown	Indris	Spectral Centroid	*ConQuer*
4	George W. Bush, Military Speech	Monkeys	Unlisted	*MATConcat*
5	Schoenberg, String Qrt 4, (1st Mvmt)	Anthony Braxton	Spectral Centroid, Spectral Rolloff	*MATConcat*
6	Cornershop, Brimful of Asha	Cornershop, Brimful of Asha (remix)	Melody	N/A
7	Natasha Beddingfield, Pocketful of Sunshine	Lady Gaga, So Happy I Could Die	Melody	N/A
8	Green Day, Warning	The Kinks, Picture Book	Melody	N/A

5.4.2 Results

Two aspects were evaluated in this listening test: perceived sound similarity and perceived interestingness. The two aspects were non-clausal, which means that one can excel without the other. As an example, a sound can be thought to have low similarity to the target, but yet it can still be perceived as highly interesting, and vice versa. The results are presented separately in Figure 61 and Figure 62 below.

The general pattern that can be observed across the two groups of participants from this test with respect to similarity seems to suggest that both musician and non-musician groups were in agreement in their perception of sound similarity. Both groups indicated that as the listening test progressed, the sounds appeared to possess more similar qualities to their targets (Figure 61). For instance, the first track only received an average score of 2.520 from the non-musician group and a slightly higher average score from the musician group (3.188). The score then climbed up steadily until it reached its peak at the eighth track in the test, receiving average scores of 4.240 and 4.313 from respective groups.

The same cannot be said, however, for their perception of sound interestingness. The non-musician group had exhibited a general disinterest in the earlier sounds presented in the track, but grew fonder of the sounds towards the end of the test. This pattern was not present with the musician group, as participants in this group seemed have a neutral liking of all sounds initially, but an apparent drop in the interest was noticed for the last three sounds. In fact, the last two sounds in the track (Tracks 7 and 8) ranked last with the lowest average scores of at 3.000 among all eight tracks. A significant crossover is seen occurring between the two groups at Track 6, where the non-musician group continued to find the sounds with increasing interestingness, whilst the score spiralled down with the musician group from then on (Figure 62).

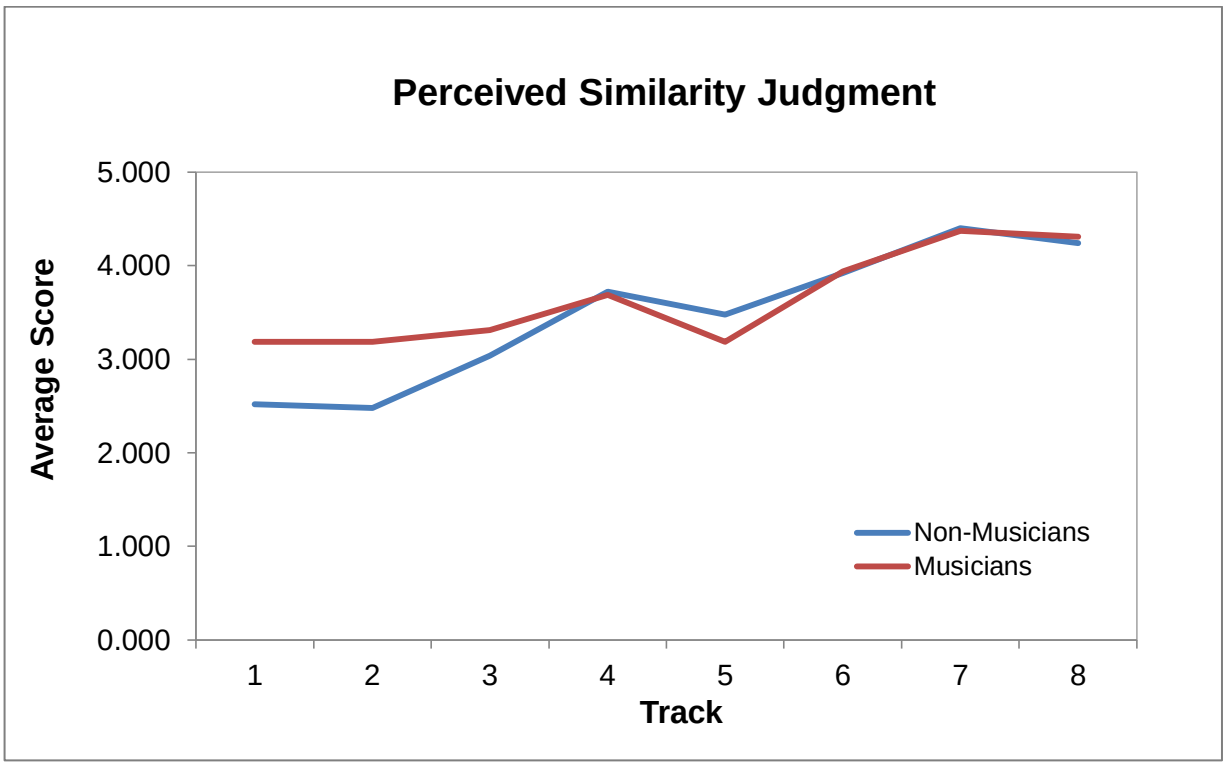


Figure 61: Result of Perceived Similarity Judgment between Musician and Non-Musician Group

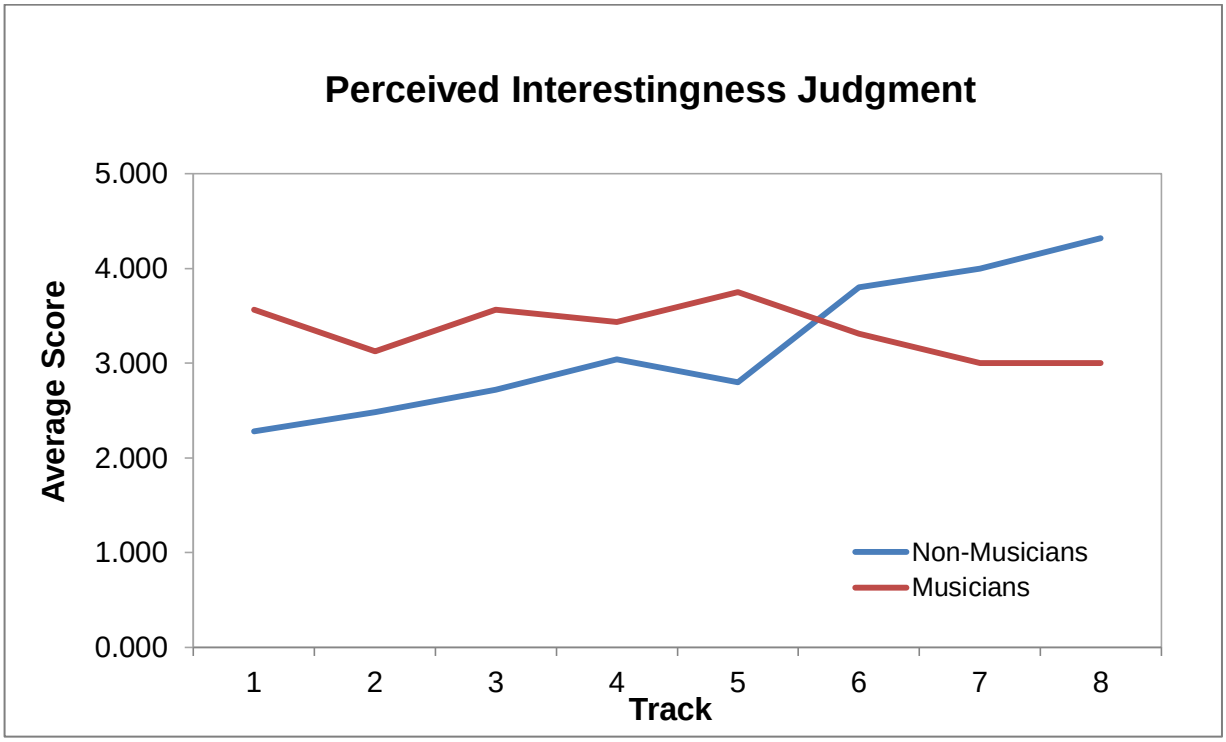


Figure 62: Result of Perceived Interestingness Judgment between Musician and Non-Musician Group

Using the scores obtained from the same listening test, the correlation between similarity and interestingness could be drawn across the two groups. To assess whether a relationship exists, a Pearson Product-Moment Correlation (PPMC) was computed. In non-musician, it was found that there was a strong positive correlation between the two variables [r=0.9326, n=8, p=0.765], suggesting that with non-musician, higher sound similarity equates to higher interestingness. A scatterplot that summarises the result is given in Figure 63.

The same observation was carried out for the musician group. It was discovered that the reverse of the above situation was true, where a moderate negative correlation between the two variables [r=0.654, n=8, p=0.765] was found. This suggests that the interest in the sound gradually decreases as their similarity to the target sound increases. This is represented visually in the scatterplot graph in Figure 64.

In addition, the coefficient of determination (R2) of the non-musician group shows that 86.99% of the interestingness is explained by the variation in the similarity, which implies that the regression is a really good fit. In comparison, only 42.8% of the variation in the interestingness is explained by the variation in the similarity for the musician group.

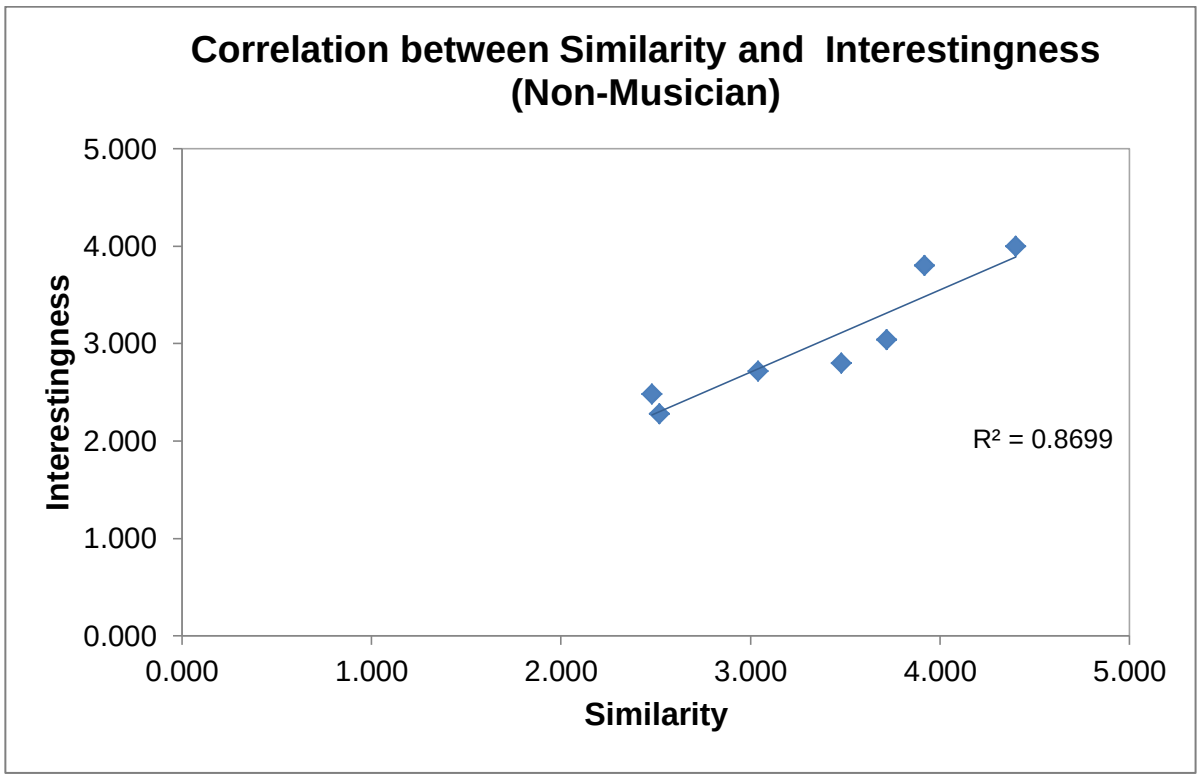


Figure 63: Result of Correlation between Judgment of Similarity and Interestingness in the Non-Musician Group

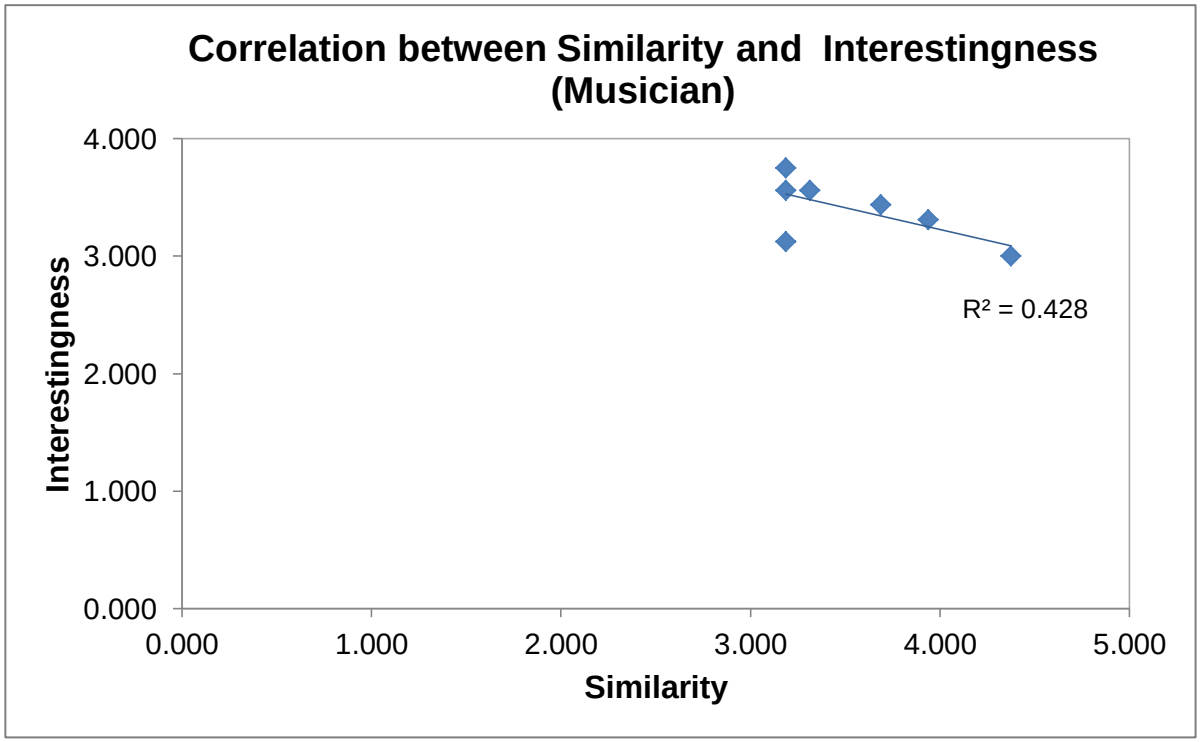


Figure 64: Result of Correlation between Judgment of Similarity and Interestingness in the Musician Group

Another useful finding from this listening test was the listeners' unanimity in scoring. For instance, the musicians were more unanimous with their judgment in similarity, where their range on the similarity score was smaller [min,max] = { $x \in \mathbb{R} \mid 3.188 < x < 4.375$ }, compared to the non-musician group which presented a more scattered judgment score [min,max] = { $x \in \mathbb{R} \mid 2.520 < x < 4.400$ }.

This trait was also present with the musician group where interestingness was concerned. The scores were fairly consistent [min,max] = { $x \in \mathbb{R} \mid 3.000 < x < 3.750$ }, whereas the non-musician group tended to give more extreme scores [min,max] = { $x \in \mathbb{R} \mid 2.280 < x < 4.320$ }, This means that for the musicians, when a sound was perceived to be interesting, a really high score was given, and if a sound was perceived to be uninteresting, a really low score was given. The full result for this listening test can be referred in Appendix D.

5.4.3 Discussion

It was found that humans, irrespective of their musical training background or knowledge of music, possessed the same ability and managed to achieve an agreement with regards to judging sound similarity. In the test, it could be clearly seen that for both groups, the first half of the tracks were marked as less similar to the target and higher marks were gradually scored as the test progressed. Since the test had been designed in such a way that the tracks at the beginning were synthesised through loudness or spectral similarities whilst the final three tracks were based on melodic similarity, the result from this listening test further supports the result in the previous test in Chapter 4 (Section 4.3.1, p.113) where under unrestricted conditions, humans tend to base their similarity judgment on the melodic element rather than other perceptual audio attributes.

Judgment on sound interestingness, however, was not as straightforward. Preferences were seen to be split into two according to the two listeners groups. The non-musician group tended to find the tracks to be more aesthetically pleasing as the test progressed, indicating a strong, positive correlation between sound similarity and sound interestingness. This was not found to be the case with the musician group, where a non-conformed agreement was found between them, thus no real relationship between the two variables can be substantially claimed.

This difference is perhaps due to the different ways in which the brain is programmed between the two groups upon hearing an audio event. Rigorous training and experience over the years has left musicians to perceive the musical experience primarily in the left hemisphere of their brains. This made them more analytical and approach music more intellectually. On the other hand, non-musicians dominantly occupy the right hemisphere of the brain during a listening task, and hence they do not analyse music, but are simply experiencing it (Segalowitz, 1983).

Also with respect to interestingness, many educated musicians may not appreciate music unless it is 'profound', whereas non-musicians, who are the majority, may prefer music that makes them feel good. So it is possible that a musician writes a piece of music that is extremely complex and is heralded by the academic music world as a masterpiece, but the same piece may only be perceived as boring or too cerebral by the general (David, 1994). Participants from the non-musician group might have also been affected by what is known as the 'exposure effect', where familiarity with, or exposure to, repeated songs bread partiality on the sounds that they favoured (Loui, Wessel and Hudson-Kam, 2010). This in some ways explains the rather low scores given by the participants from the non-musician

group at the synthesised sounds presented in test, including pieces generated from *ConQuer*.

Additionally, musicians may find that similar sounds are neither more interesting nor desirable as they understand more about the potential use of the sounds synthesised automatically by the CSS system than non-musicians. If the sounds are too similar, it is of little use for the composers as it lacks originality and may even tread into the serious issue of plagiarism. This opens up another thought-provoking question: how similar is acceptable? A definite answer to this question cannot be easily derived, and certainly beyond the scope of this study, but it is nonetheless interesting to note that the performance of a CSS system cannot simply be measured solely on the use of precision and recall as is the case in many sound similarity systems or speech synthesis systems.

Finally, it was found in the test that participants from the musician group were more inclined to give 'milder' and more consistent scores, compared to the non-musician group. Several reasons could be explained for this behaviour, including revisiting the earlier 'exposure effect' theory, where musicians who were already familiar with sounds generated or sounds to be expected from a product of sound synthesis system, were less likely to be surprised by how the earlier tracks sounded, compared to those from the non-musician group who might have expected the sounds to be somewhat different. Perhaps participants in the non-musician group were really focusing on the melodic similarity of the sound and thus overlooking similarities that might have existed in other perceptual attributes, resulting in some harsh scores when their expectations were not met. Last but not least, the involvement and knowledge in music making of the participants in the musician group meant that they have higher empathy and appreciation for the enormous amount of work that went behind such automated task, and therefore scored more perceptively.

5.4.4 Conclusion

This final experimental set highlights the following interesting findings:

i) *Sound similarity and sound interestingness do not always occur simultaneously*

Sounds that are perceived to be more similar to the target are not always found to be interesting. Likewise, sounds that are less similar to the target can sometimes be perceived as interesting. As a sound creation tool, the key lies in finding the balance between similarity and interestingness to generate sounds that are not too similar to the target to be perceived as boring or unoriginal, but at the same time not too dissimilar as to render the involvement of the target segment useless. Identifying the target user in which the CSS system is developed for will undoubtedly avoid synthesis results that mismatch user's expectations.

ii) *Musical training alters the way human listens and appreciates sounds*

Musical training does not only provide humans with additional musical knowledge that may affect their more intellectual approach to judging sounds, but the physiological way of how their brains function upon hearing a musical event is also altered. This is the reason why judgment in interestingness differs between the two groups and the explanation for the more consistent and mild scoring in the test.

iii) *ConQuer is proven to be a feasible and practical CSS system*

ConQuer's performance was at par with other CSS systems tested, where its synthesised sounds were found to be of high interestingness and generally well-received.

5.5 Summary

This chapter described a total of four experimental sets that were carried out during the length of this study to verify the validity of the solutions proposed to the problems in CSS systems as disclosed before. These experimental sets included evaluations of the effects which the parametric input, order and intensity importance in audio features selection, and use of concatenation distance on the search and selection process involving homosonic and equidistant segments had on the overall synthesis outcome. Results obtained from these experimental sets have shown positive evidence to support the ability of the solutions proposed to fulfil the objectives and to overcome the challenges that were undertaken in this study.

The listening test that was conducted in the final experimental set revealed several interesting findings, such as the relationship between perceived sound similarity and perceived sound interestingness in human listeners and also reported on the different behaviour observed between the two groups studied (musician and non-musician), as far as sound similarity judging was concerned. It was also pleasant to discover from the test that the performance of *ConQuer* was comparable to other CSS system and that the sound generated was regarded as fairly decent.

The strengths and limitations of this study, and specifically on the framework and performance of *ConQuer*, will be discussed in the next and final chapter of this thesis, Chapter 6: Conclusion.

Chapter 6: Conclusion

This chapter provides a summary of the findings from this study, along with the discussion on its contributions as well as its limitations. Several recommendations are also included for future works.

6.1 Research Findings

This study was set out to address the issues in existing CSS systems, or more specifically, to improve sound similarity between the sound synthesised by the system and the target sound. To overcome these challenges, the human cognitive domain must first be understood and the information obtained from the former must then be converted into some form of Artificial Intelligence solutions.

At the start of the study, it had been theorised that in order to improve sound similarity, the elements that are used by humans as a common ground for comparison (basis of sound similarity) in performing tasks that involve sound similarity perception must first be identified. This is because without a common ground declared, it is very likely that a CSS system will generate sounds that do not match the expectation of its users, despite being fed with a target sound at the start of the process. For example, a user may provide a target sound to the system, expecting that a new sound with similar beat will be generated. Without further clarification from the user regarding the basis of sound similarity, the system can synthesise sounds that are similar in terms of any other perceptual attributes such as loudness, melody or timbre. This mismatch can leave the user feeling puzzled by the output, and perhaps brandishing the system as a failure, even when it is fully functional. Based on this notion, it was apparent that a study to determine the most dominant perceptual attribute that humans use to form the basis of their sound similarity judgment

needed to be conducted, and findings from the study would be used to enhance the CSS system by including more features that correspond to the most dominant perceptual attributes.

Preliminary survey at the start of the study found that many challenges remain in existing CSS systems, one of which is the issue of user control. In many cases, sounds that are generated through CSS still rely heavily on random arrangements of sound units. In the more recent development of CSS systems, users are able to set certain parameters which the systems offered, but this mostly involves simple entrance and manipulation of numerical data. This process is seen as tedious, time-consuming and generally functioning on a trial-and-error basis. Oftentimes, this mindless tweaking of the parameters leaves users feeling overwhelmed and frustrated. In the long run, this mundane and uninspiring method of music making may hinder creative composition from happening. Moreover, the sole use of numerical data often means distancing any valuable qualitative input from users such as similarity judgment and feature priority judgment.

Another example that was discovered surrounding the issue of low user control flexibility in the current CSS system was the inability to assign weights on the different audio features. This may become a problem when two or more audio features are included in the similarity search, but each feature carries a different importance (weight). In the similarity search, it is imperative that some features to be matched closely, whilst some other features can afford to have a little more distance from that of the target sound, depending on the preferences set by the users regarding the compositional piece that he has in mind. One other flaw that was spotted in the existing CSS systems was the handling of homosonic and equidistant segments. Typically, when this situation occurs, without much intelligence, current systems return random segments to be concatenated and synthesised.

Evidently, very little intelligence is incorporated to tackle any of the issues described above. Thus, this study aimed to bridge these gaps through the extended use of AI. The inclusion of AI was thought to be able to automate certain tasks, as well as allowing some qualitative decisions from users to be included in the process of generating the sounds. Once the abovementioned problems have been identified, the following research questions which then shaped the thesis were synthesised:

1. What elements of sound play a major role when human performs sound similarity tasks?
2. Would extending some aspects of the AI implementation in a CSS system enhance user control and improve sound similarity result of the sounds composed?

To answer these questions, the study performed several steps, it had: (1) determined the most dominant perceptual sound attributes that humans use to judge sound similarity, (2) identified the key factors that affect synthesis results, (3) presented several problems within the existing CSS systems, (4) demonstrated possible solutions to overcome these problems, (5) proposed a novel framework (query-based CSS) that tied all these findings together, and (6) verified the validity of the framework and solutions provided through a series of experiments and listening tests, of which results are reported throughout Chapter 4 and 5 of this thesis.

Firstly, the dominant perceptual attribute that became the basis of humans' sound similarity judgment was determined. Through a preliminary listening test, it was revealed that musical training plays a major role when humans undertake a sound similarity task. Among non-musicians, the melody information was seen to be the most dominant perceptual attribute that became the basis of sound similarity judgment, whereas with musicians, timbre was

more common (Chapter 4, p.119). The split in the agreement between these two perceptual attributes was traced down to the different ways in which the brain is programmed between the two groups upon hearing an audio event. Musicians were inclined to use their left brains, resulting in a more analytical and intellectual hearing, whilst non-musicians primarily utilised the right hemisphere of the brain to simply experience the sound without much analysis complicating their judgment. This trait was not only exhibited during the similarity tests, but also during another listening test that evaluated the interestingness of sounds generated from several CSS systems too (Chapter 5, p.190). Additionally, musicians appeared to be more accepting of sounds that were more diverse in nature and melodically further away from the target, whereas non-musicians generally found that sounds which are melodically similar to be very interesting and dismissed those that are not. By understanding the listening behaviour of the target user group, a new CSS system that can cater certain groups can be developed. This will certainly help reduce the human-computer misperception of similar sounds during synthesis. As musicians are the prime target user for any CSS system, features that correspond to the timbral quality of a sound such as spectral centroid, spectral rolloff and ZCR, are given more emphasis in the final framework of the CSS system developed.

Secondly, the variables that affect the synthesis result were identified. It was hypothesised that by identifying the key factors which affect the synthesis result from a CSS system, and by providing them options which users can control, will improve the communication between the users and the system in the intended creations. Following this, an initial parametric input evaluation was carried out, and it was identified that the size of the dataset, the choice of source files and target files, and also the segmentation mode affected

the synthesis results (Chapter 5, p.136). Thus, it is important to recognise the purposes of selecting and enabling certain parameters in the search, as they directly affect output.

In addition to the flexibility of selecting the parametric input, this study also found that selecting different audio features resulted in the synthesis of different sounds. It was also established that the inclusion of more features did not necessarily result in closer matches. Furthermore, in the case where multiple audio features were used, the features might carry different importance intensities (or priority weights) from one another. To distinguish this, AHP was employed in this study as it could automatically convert human judgments on the relative order and importance of features into reliable weights, resulting in the generation of sounds that have closer target distance to the original target segment than those without. Not only did it encourage interaction with users by allowing varying importance intensities of the features to be set, but it also tackled it intelligently by converting qualitative human knowledge into quantitative unit of measurement.

This research also discovered that a database could contain several sound segments that were represented with the same sonic information, but were not duplicate copies and were aurally different. Likewise, in cases where no exact match was found, there could be two or more segments in the database with the same target distance from the original target segment. The terms ‘homosonic’ and ‘equidistant’ segments were invented in this study to describe the two respective conditions. The concatenation distance was found to be a feasible and effective solution to these problems (Chapter 5, p.169, Appendices A13 and A14 in the CD), and its implementation based on the hierarchical model was also an intelligent alternative to the random selection currently engaged when faced with such conditions.

Based on the findings from the above experiments, this study also proposed to replace the earlier CSS framework with a novel query-based framework (Chapter 4, p.132). The query-based framework suggests that there is an explicit 'Query' stage added to the original framework of CSS. In this 'Query' stage, all information that the users needed to convey to the system can be communicated which include what features to be included, weights for each feature and activating concatenation distance and several other information. The design of the framework took into account the findings that surfaced from this study and embedded the solutions to the issues addressed in the prototype system, *ConQuer*. The query-based framework was found to:

- increase user control by providing a centralised medium (the query stage) for users to communicate their specifications to the system,
- be flexible enough to allow changes in the variables offered to the users, i.e. to add or remove certain parameters,
- include intelligent, methodical solutions to overcome the challenges found in earlier CSS systems, e.g. using AHP and concatenation distance,
- reduce post-synthesis adjustments and transformation relaying all the specifications to the system before synthesis takes place, and
- be robust enough to suit users with different interests and musical backgrounds, i.e. similarity based on timbral quality for musicians, and melodic contour for non-musicians.

To portray the potential application of this study, a short composition using the sound generated via *ConQuer* is included in Appendix A16 in the CD.

6.2 Contributions

In order of importance, a summary of the contributions of this study is as follows:

i) Proposing the new query-based CSS framework that encourages flexible user control. A query stage is necessary to ensure that all the details including parametric input, audio features and their order of importance, as well as other options are communicated from the users to the system before synthesis takes place. Existing systems do not engage in this query stage, forcing the act of re-entering and adjusting of certain input to take place after synthesis has commenced. The query-based CSS model thus minimises the need for these post-synthesis adjustments and transformations.

ii) Establishing the need for an order-dependent feature selection process which prioritises match between target and source segments according to the weights assigned for individual features.

iii) Recognising the challenges with homosonic and equidistant segments during unit selection process and proposing a robust new hierarchical model approach to counter this.

iv) Comprehensive evaluations to validate the feasibility and effectiveness of the proposed framework.

v) Intensive technical and artificial intelligence survey carried out to comprehend the underlying problems in CSS.

vi) Implementation of the query-based concatenative sound synthesis on this study's 'proof-of-concept' – *ConQuer*.

6.3 Limitations

The contribution and achievement list from the previous page have demonstrated that this study has adequately achieved its aims and objectives. However, several challenges and limitations were also discovered, and are listed as follows:

i) *Limited factors tested*

There are several more factors that may potentially affect the synthesis result that were not investigated because they were beyond the scope of this study, for example, the effect of using different search algorithm and the inclusion of MIDI or other symbolic data. Also, due to the limitation in resources and time, it was impossible to cover all possible factors. However, some of these are listed in the following section as future works (p.207).

ii) *Tedious and exhaustive alternative*s

Concentration-demanding processes during both order-dependent feature selection and concatenation distance-enabled mode can become computationally-exhaustive and result in longer run-time. However, as discussed in the previous chapter, this issue could be alleviated by limiting the use or combinations of certain variables wisely.

iii) *Reliance on quantitative experiments*

It may be disputed that a study on such a qualitative subject had been conducted through a series of quantitative methods and measurements. However, in this case, the quantitative method is the most practical approach, and the empirical results obtained have provided a respectable indication of improvements. In addition, several qualitative tests involving human input were also carried out to validate the results from the quantitative experiments.

iv) *Timbral only restriction for the sound similarity basis*

During the query stage in the proposed query-based CSS system, only timbral was implemented as the basis of sound similarity. Ideally, a query-based CSS system should allow the different perceptual attributes that affect the basis of similarity in humans to be exchangeable to match the target users, but due to limited resources, the proof-of-concept, *ConQuer*, had only implemented sound similarity based on the timbral quality. The reason behind this was because *ConQuer* was originally developed to cater for the main target user of this system, for example the musicians. Nevertheless, its robust framework means that it is possible to integrate other attributes as well in the future.

v) *Restricted dataset size*

A moderate-size database was used in this study to ensure that the experiments were manageable. As a result, the significance of the approaches proposed might not have become immediately apparent in some of the sound examples. Using a larger corpus may result in more noticeable effect. However, the findings from this study should not be dismissed as the improvements are also supported by the empirical data.

vi) Offline synthesis

All syntheses from this study were generated offline. Extending the implementation online will benefit the users more; enabling live composition to take place as well as real-time human-computer interaction.

6.4 Recommendations for Future Work

A number of suggestions outside the scope of this study have been identified. Interesting studies can be led by following the list below:

i) *Conducting further parametric investigations*

Although the effects of the more important parameters have been investigated in this study, several other parameters such as the effect of different search methods or effects of including a wider range of audio features may also be studied and quite possibly bring forward different and intriguing results.

ii) *Allowing the basis of sound similarity to be changeable*

As previously mentioned, enabling the perceptual attributes that affect basis of similarity in humans such as melody, in addition to the already implemented timbral quality may drive the potential of the system to suit a wider target user (both musicians and non-musicians).

iii) *Developing a memory mechanism to handle concatenation without replacement or the 'taboo list'*

The taboo list is another factor that was not included in this investigation. During the unit selection process, the study had always assumed concatenation with replacements where a source segment in the database may be used more than once in the synthesis. This meant that the synthesised sounds were not made up of unique source sounds, and some repetitions were expected. If the condition states that only unique segments are allowed to be synthesised, then the taboo list may provide the answer. However, since a permanent barring of source segments can create many problems, including shortage of segments, some form of memory mechanism that

allows source segment repetition after a certain amount of time has lapsed resembling the short-term or long-term memory effect is worth researched into.

iv) Extending the study to include post-synthesis transformation options

This study did not include any post-synthesis information as it was developed on the notion of a query-based system where all variables are set before synthesis takes place. However, by offering a post-synthesis option such as spectral shifting, time-stretching and spectral freezing, users may able to adjust and tweak their generated sounds to create more diversely textured sounds.

6.5 Summary

This study intended to address the issues in existing CSS systems and to improve similarity of composed sounds by exploiting the AI approaches derived from the understanding of the human's sound cognitive domain. In some ways, this study has achieved its intentions, although at the current moment, all available CSS systems, including *ConQuer*, still rely on some form of human input in order to synthesise sounds. Nevertheless, CSS has come a long way since the days where magnetic tapes were cut and pasted manually by hand. Although complete automation is not yet achieved and the level of intelligence integrated within the CSS system is no match to that of humans, this study has managed to come up with methodical and reliable ways to tackle the challenges in existing CSS systems. It is hoped that through recommended works on CSS as previously listed, coupled with the exciting possibility of more intelligent solutions emerging in the near future, the day where a CSS system is able to 'read' and 'materialise' the minds of composers may become a reality soon. But for the time being, it can be said that overall, despite the limitations encountered, this study is successful in achieving all of its intended objectives.

References

Abeßer, J., Lukashevich, H., Dittmar, C., and Schuller, G. (2009). Genre classification using bass-related high-level features and playing styles. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2009 (pp. 453-458).

Abraham, A. (2005). 130: Rule-based Expert Systems. *Handbook of Measuring System Design, edited by Peter H. Sydenham and Richard Thorn, John Wiley & Sons, Ltd. ISBN: 0-470-02143*, *8*, 909-919.

Ahmad, A. (1997). *Lagu-lagu Gamelan*. Penerbit Universiti Malaya.

Allamanche, E., Herre, J., Hellmuth, O., Kastner, T., and Ertel, C. (2003, October). A multiple feature model for musical similarity retrieval. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2003.

Allen, J. B., and Rabiner, L. R. (1977). A unified approach to short-time Fourier analysis and synthesis. *Proceedings of the IEEE*, *65*(11), 1558-1564.

Alwaer, H., and Clements-Croome, D. J. (2010). Key performance indicators (KPIs) and priority setting in using the multi-attribute approach for assessing sustainable intelligent buildings. *Building and Environment*, *45*(4), 799-807.

Ariffin, Z. (1990). *Mengenal budaya bangsa*. Dewan BahasadanPustaka.

Ashayeri, J., Keij, R., and Bröker, A. (1998). Global business process re-engineering: a system dynamics-based approach. *International Journal of Operations & Production Management*, *18*(9/10), 817-831.

Aucouturier, J. J., and Pachet, F. (2002, October). Music similarity measures: What's the use. In *Proceedings of the 3rd International Symposium on Music Information Retrieval (ISMIR)*, 2002 (pp. 157-163).

Aucouturier, J. J., and Pachet, F. (2005, September). Ringomatic: A real-time interactive drummer using constraint-satisfaction and drum sound descriptors. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR), 2005* (pp. 412-419).

Bailey, C. (2010). A Database System For Organising Musique Concrete. In *International Computer Music Conference*, *2010* (pp. 428-431).

Bartsch, M. A., and Wakefield, G. H. (2001). To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2001 (pp. 15–18).

Battier, M. (2007). What the GRM brought to music: from musique concrete to acousmatic music. *Organised Sound*, *12*(03), 189-202.

Bernardes, G., Guedes, C., and Pennycook, B. (2012). EarGram: an Application for Interactive Exploration of Large Databases of Audio Snippets for Creative Purposes. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval (CMMR)* (pp. 265-277).

Blackwell, T. M. (2003). Swarm music: improvised music with multi-swarms. *Artificial Intelligence and the Simulation of Behaviour, University of Wales*.

Bowersock, G.W. (2006). *Mosaics as History: The Near East from Late Antiquity to Early Islam* (Vol. 16). Belknap Press.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Workshop on Speech and Natural Language,* 1992 (pp. 112-116).

Brossier, P. (2006). Automatic annotation of musical audio for interactive applications. *Centre for Digital Music, Queen Mary University of London*.

Cao, C., Li, M., Liu, J., and Yan, Y. (2007). Singing melody extraction in polyphonic music by harmonic tracking. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR),* 2007.

Cardle, M., Brooks, S., and Robinson, P. (2003). Audio and user directed sound synthesis. In *Proceedings of the International Computer Music Conference (ICMC),* 2003.

Casey, M. A. (2005). Acoustic lexemes for organizing internet audio. *Contemporary Music Review*, *24*(6), 489-508.

Chafe, C. (2001). A Short History of Digital Sound Synthesis by Composers in the USA. Unpublished*, http://www-ccrma.stanford.edu/~cc/lyon/historyFinal.pdf*.

Chapel, R. H. (2003). Realtime algorithmic music systems from fractals and chaotic functions: toward an active musical instrument. PhD thesis, Department of Technology, *University PompeuFabra*.

Chavarría, J. (1999). *The art of mosaics*. Watson-Guptill.

Chen, F. L., and Chen, Y. C. (2009). An investigation of forecasting critical spare parts requirement.In *Computer Science and Information Engineering, WRI World Congress,* 2009 (Vol. 4, pp. 225-230).

Chen, C., Gagaudakis, G., and Rosin, P. (2000).Similarity-based image browsing.In *Proceedings of the 16th IFIP World Computer Congress.International Conference on Intelligent Information Processing*.

Chen, C. F. (2006). Applying the analytical hierarchy process (AHP) approach to convention site selection.*Journal of Travel Research*, *45*(2), 167-174.

Collier, W. G., and Hubbard, T. L. (1998). Judgments of happiness, brightness, speed and tempo change of auditory stimuli varying in pitch and tempo. *Psychomusicology: Music, Mind & Brain*, *17*(1), 36-55.

Cook, P. R. (2002). *Real sound synthesis for interactive applications*. AK Peters.

Cover, T., and Hart, P. (1967). Nearest neighbour pattern classification. *Information Theory, IEEE Transactions on*, *13*(1), 21-27.

Davaatsagaan, M., and Paliwal, K. K. (2008). Diphone-Based Concatenative Speech Synthesis System for Mongolian.In *Proceedings of the International MultiConference of Engineers and Computer Scientists,* 2008 (Vol. 1).

David, J. H. (1994). In *The two sides of music*, http://jackhdavid.thehouseofdavid.com/papers/brain.html

Davies, H. (1996). A history of sampling.*Organised Sound*, *1*(1), 3-11.

Di Blasi, G., and Gallo, G. (2005).Artificial mosaics.*The Visual Computer*, *21*(*6*), 373-383.

Dierks, L. (2004). *Making mosaics: Designs, techniques and projects*. Lark Books.

Donovan, R., Ittycheriah, A., Franz, M., Ramabhadran, B., Eide, E., Viswanathan, M., and Kunzmann, J. (2001).Current status of the IBM trainable speech synthesis system.In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

Donovan, R. E., and Eide, E. M. (1998).The IBM trainable speech synthesis system. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)* (Vol. 98).

Ebcioglu, K. (1984). An expert system for schenkerian synthesis of chorales in the style of JS Bach. In *Proceedings of the International Computer Music Conference (ICMC)* (pp. 135-142).

El-Maleh, K., Klein, M., Petrucci, G., and Kabal, P. (2000).Speech/music discrimination for multimedia applications.In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'00),* 2000(Vol. 6, pp. 2445-2448).

Eronen, A., and Klapuri, A. (2000).Musical instrument recognition using cepstral coefficients and temporal features.In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'00),* 2000(Vol. 2, pp. II753-II756).

Falconi, C. A. (1999). Methods for priority setting in agricultural biotechnology research.*Biotechnology in Agriculture Series*, 40-52.

Farnell, A. (2007). An introduction to procedural audio and its application in computer games. In *Audio Mostly Conference* (pp. 1-31).

Finan, J. S., and Macnamara, W. D. (2001).An illustrative Canadian strategic risk assessment.*Canadian Military Journal*, *2*(3), 29-34.

Fitzgerald, D. (2010). Harmonic/percussive separation using median filtering.In Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10), Graz, Austria.

Fonseca, N., Ferreira, A., and Rocha, A. P. (2011).Concatenative singing voice resynthesis.In *Digital Signal Processing (DSP), 2011 17th International Conference on* (pp. 1-4).

Forney Jr, G. D. (1973). The viterbi algorithm.*Proceedings of the IEEE*, *61*(3), 268-278.

FXpansion.Bfd, premium acoustic drum library module, 2003. website: http://www.fxpansion.com.

Gates, A., and Bradshaw, J. L. (1977).The role of the cerebral hemispheres in music.*Brain and Language*, *4*(3), 403-431.

Gower, J. C. (1985). Properties of Euclidean and non-Euclidean distance matrices.*Linear Algebra and its Applications*, *67*, 81-97.

Grey, J. M. (1975). *An exploration of musical timbre* (Doctoral dissertation, Department of Music, Stanford University).

Gudivada, V. N., and Raghavan, V. V. (1995). Design and evaluation of algorithms for image retrieval by spatial similarity.*ACM Transactions on Information Systems (TOIS)*, *13*(2), 115-144.

Guo, G., and Li, S. Z. (2003). Content-based audio classification and retrieval by support vector machines. *Neural Networks, IEEE Transactions on*, *14*(1), 209-215.

Hajkowicz, S. A., McDonald, G. T., and Smith, P. N. (2000). An evaluation of multiple objective decision support weighting techniques in natural resource management.*Journal of Environmental Planning and Management*, *43*(4), 505-518.

Hess, W. (1983). *Pitch determination of speech signals: algorithms and devices* (Vol. 3). Springer.

Ho, W., Xu, X., and Dey, P. K. (2010). Multi-criteria decision making approaches for supplier evaluation and selection: A literature review. *European Journal of Operational Research*, *202*(1), 16-24.

Hofmann-Engl, L. (2001). Towards a cognitive model of melodic similarity. In *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR), 2001* (pp. 143-151).

Hunt, A. J., and Black, A. W. (1996, May). Unit selection in a concatenative speech synthesis system using a large speech database. In *Proceedings of the Acoustics, Speech, and Signal Processing, 1996.ICASSP-96.*(Vol. 1, pp. 373-376).

Jensenius, A. R., and Johnson, V. (2010, November). A video based analysis system for realtime control of concatenative sound synthesis and spatialisati. In *Norwegian Artificial Intelligens Symposium (NAIS)*.Tapir AkademiskForlag.

Johnson-Laird, P. N. (1991). Jazz improvisation: a theory at the computational level. *Representing musical structure, London,* 291-325.

Jones, K. (1981). Compositional applications of stochastic processes.*Computer Music Journal*, 45-61.

Keller, R., and Morrison, D. R. (2007, July). A grammatical approach to automatic improvisation. In *Proceedings, Fourth Sound and Music Conference*, Lefkada, Greece.

Kishore, S. P., and Black, A. W. (2003).Unit size in unit selection speech synthesis. In *Proceedings of EUROSPEECH* (Vol. 2003, pp. 1317-1320).

Klapuri, A. (2004). *Signal processing methods for the automatic transcription of music*. PhD thesis, Finland: Tampere University of Technology.

Korf, R. E. (1985). Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, *27*(1), 97-109.

Laaksonen, J., Oja, E., Koskela, M., and Brandt, S. (2000). Analyzing low-level visual features using content-based image retrieval.In *Proceedings of the 7th International Conference on Neural Information Processing (ICONIP'00),* 2000 (pp. 1333-1338).

Lazier, A., and Cook, P. (2003). MOSIEVIUS: Feature driven interactive audio mosaicing. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)* (pp. 312.317).

Lemmetty, S. (1999).*Review of speech synthesis technology*. M.S. thesis, Dept. Elect. Commun.Eng., *Helsinki University of Technology*.

Leslie, G., Zamborlin, B., Jodlowski, P., and Schnell, N. (2010). Grainstick: A collaborative, interactive sound installation. In *Proceedings of the International Computer Music Conference (ICMC)*, 2010.

Levitin, D.J. (2006). *This is your brain on music: The science of a human obsession*. Plume Books New York.

Li, T., and Tzanetakis, G. (2003, October). Factors in automatic musical genre classification of audio signals. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* (pp. 143-146). IEEE.

Ling, R. (1998). *Ancient mosaics*. Princeton University Press.

Loui, P., Wessel, D. L., and Kam, C. L. H. (2010). Humans rapidly learn grammatical structure in a new musical scale. *Music perception*, *27*(5), 377.

Macharis, C., Springael, J., De Brucker, K., and Verbeke, A. (2004). PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP. *European Journal of Operational Research*, *153*(2), 307-317.

Macon, M. W., Jensen-Link, L., Oliverio, J., Clements, M. A., and George, E. B. (1997, April). A singing voice synthesis system based on sinusoidal modeling. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on* (Vol. 1, pp. 435-438). IEEE.

Macrae, R. (2008). Linking music-related information and audio data.

Maestre, E., Ramírez, R., Kersten, S., and Serra, X. (2009). Expressive concatenative synthesis by reusing samples from real performance recordings.*Computer Music Journal*, *33*(4), 23-42.

Makaran Ramesh, V., and Sahasrabuddhe, H.V. (2008, July).Exploring Data Analysis in Music using tool praat. In *First International Conference on Emerging Trends in Engineering and Technology, 2008. ICETET'08.* (pp. 508-509). IEEE.

McCormack, J. (1996). Grammar based music composition. *Complex systems*, *96*, 321-336.

McKinney, M. F., and Breebaart, J. (2003, October). Features for audio and music classification. In *Proc. ISMIR* (Vol. 3, pp. 151-158).

Mierswa, I., and Morik, K. (2005). Automatic feature extraction for classifying audio data. *Machine learning*, *58*(2-3), 127-149.

Miranda, E. R. (1995). Granular synthesis of sounds by means of a cellular automaton. *Leonardo*, 297-300.

Miranda, E. R. (1998a). *Computer sound synthesis for the electronic musician*. Butterworth-Heinemann.

Miranda, E. R. (1998b). Machine Learning and Sound Design.

Mitrović, D., Zeppelzauer, M., and Breiteneder, C. (2010).Features for content-based audio retrieval.*Advances in computers*, *78*, 71-150.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, *24*(11), 1097-1100.

Moulines, E., and Charpentier, F. (1990). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones.*Speech communication*, *9*(5), 453-467.

Mousa, A. (2010). Voice Conversion Using Pitch Shifting Algorithm by Time Stretching with PSOLA and Re-Sampling. *Journal of Electrical Engineering*,*61*(1), 57-61.

Müller, M., Kurth, F., and Clausen, M. (2005).Audio matching via chroma-based statistical features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR). 2005*, (pp. 288-295).

Nasuruddin, M.G. (1992). *The Malay Traditional Music*.Dewan Bahasa dan Pustaka.

Negnevitsky, M. (2005). *Artificial intelligence: a guide to intelligent systems*. Addison-Wesley Longman.

Norowi, N. M., Doraisamy, S., and Wirza, R. (2005, September). Factors affecting automatic genre classification: an investigation incorporating non-western musical forms. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*. 2005, (pp. 13-20).

Pachet, F. (1999).Surprising harmonies. *International Journal of Computing Anticipatory Systems*, *4*, 139-161.

Panagiotakis, C., and Tziritas, G. (2005). A speech/music discriminator based on RMS and zero-crossings. *Multimedia, IEEE Transactions on*, *7*(1), 155-166.

Pantazis, Y., Stylianou, Y., and Klabbers, E. (2005). Discontinuity detection in concatenated speech synthesis based on nonlinear speech analysis. In *Proc. de Eurospeech* (pp. 2817-2820).

Papadopoulos, G., and Wiggins, G. (1999). AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity* (pp. 110-117).

Pellman, S. (1994). *An introduction to the creation of electroacoustic music*. Wadsworth Publishing Company.

Poh, K. L., and Ang, B. W. (1999). Transportation fuels and policy for Singapore: an AHP planning approach. *Computers & industrial engineering*, *37*(3), 507-525.

Pollastri, E. (1998). Melody-retrieval based on pitch-tracking and string-matching methods. In *Proc. Colloquium on Musical Informatics, Gorizia*.

Rabiner, L., Cheng, M., Rosenberg, A., and McGonegal, C. (1976). A comparative performance study of several pitch detection algorithms.*Acoustics, Speech and Signal Processing, IEEE Transactions on*, *24*(5), 399-418.

Ramanathan, R. (2001). A note on the use of the analytic hierarchy process for environmental impact assessment.*Journal of Environmental Management*, *63*(1), 27-35.

Roads, C. (1996). *The computer music tutorial*. The MIT Press.

Roads, C. and Wieneke, P. (1979). Grammars as representations for music. *Computer Music Journal*, *3*(1).48-55.

Rodet, X. (2002, November). Synthesis and processing of the singing voice. In *Proc. 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)* (pp. 15-31).

Rostamy, A. A. A., Shaverdi, M., Amiri, B., and Takanlou, F. B. (2012).Using fuzzy analytical hierarchy process to evaluate main dimensions of business process reengineering.*Journal of Applied Operational Research*, *4*(2), 69-77.

Russ, M. (2012). *Sound synthesis and sampling*. Focal Press.

Saaty, T. L. (1977). A scaling method for priorities in hierarchical structures. *Journal of mathematical psychology*, *15*(3), 234-281.

Saaty, T. L. (1983). Priority setting in complex problems. *IEEE Transactions on Engineering Management, 30*, 140-155.

Saaty, T. L. (1990). How to make a decision: the analytic hierarchy process. *European journal of operational research*, *48*(1), 9-26.

Saaty, T. L. (1994). How to make a decision: the analytic hierarchy process. *Interfaces*, *24*(6), 19-43.

Saaty, T. L., and Vargas, L. G. (2001).How to make a decision. *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*, 1-25.

Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, *1*(1), 83-98.

Sagisaka, Y. (1992). ATR v-talk speech synthesis system. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP),* 1992 (pp. 483-486).

Sak, H. (2004). *A corpus-based concatenative speech synthesis system for Turkish* (Doctoral dissertation, Bogaziçi University).

Salo, A., and Liesiö, J. (2006). A case study in participatory priority setting for a Scandinavian research programme.*International Journal of Information Technology & Decision Making*, *5*(1), 65-88.

Saunders, J. (1996). Real-time discrimination of broadcast speech/music.In *Acoustics, Speech, and Signal Processing (ICASSP-96)* (Vol. 2, pp. 993-996).

Scheirer, E., and Slaney, M. (1997, April).Construction and evaluation of a robust multifeature speech/music discriminator.In *Acoustics, Speech, and Signal Processing, (ICASSP-97)* (Vol. 2, pp. 1331-1334).

Schröder, M. (2001, September). Emotional speech synthesis: A review. In *Proceedings of EUROSPEECH* (Vol. 1, pp. 561-564).

Schubert, E., Wolfe, J., and Tarnopolsky, A. (2004, August). Spectral centroid and timbre in complex, multiple instrumental textures. In *Proceedings of the international conference on music perception and cognition, North Western University, Illinois* (pp. 112-116).

Schwarz, D. (2000, December). A system for data-driven concatenative sound synthesis. In *Digital Audio Effects (DAFx).* (pp. 97-102).

Schwarz, D. (2003, September). The caterpillar system for data-driven concatenative sound synthesis.In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)* (pp. 135-140).

Schwarz, D. (2004). *Data-driven concatenative sound synthesis*, PhD thesis.Academie de Paris, *Universite Paris 6 (Pierre et Marie Curie).*

Schwarz, D. (2005). Current research in concatenative sound synthesis.In *Proceedings of the International Computer Music Conference (ICMC)* (pp. 9-12).

Schwarz, D. (2006). Concatenative sound synthesis: The early years. *Journal of New Music Research*, *35*(1), 3-22.

Schwarz, D., Beller, G., Verbrugghe, B., and Britton, S. (2006). Real-time corpus-based concatenative synthesis with catart.In *Proceedings of the International Conference on Digital Audio Effects (DAFx-06).* (pp. 279-282).

Segalowitz, S. J. (1983). Two sides of the brain. *Englewoocl Cliffs: Prentice Hall*.

Self, H. (2001). Digital sampling: A Cultural Perspective. *UCLA Ent. L. Rev*., *9*, 347.

Simon, I., Basu, S., Salesin, D., and Agrawala, M. (2005). Audio analogies: Creating new music from an existing performance by concatenative synthesis. In *Proceedings of the 2005 International Computer Music Conference (ICMC)* (pp. 65-72).

Smaragdis, P., and Mysore, G. J. (2012).Following musical sources by example. In *Acoustics, Speech and Signal Processing (ICASSP).*2012(pp. 5373-5376).

Smith, J. O. (1991, October). Viewpoints on the history of digital synthesis.In *Proceedings of International Computer Music Conference* (pp. 1-1).INTERNATIONAL COMPUTER MUSIC ASSOCIATION.

Solomatine, D., See, L. M., and Abrahart, R. J. (2008). Data-driven modelling: concepts, approaches and experiences. In *Practical Hydroinformatics* (pp. 17-30).Springer Berlin Heidelberg.

Stowell, D., and Plumbley, M. D. (2010). Delayed decision-making in real-time beatbox percussion classification.*Journal of New Music Research*, *39*(3), 203-213.

Sturm, B. L. (2004). MATConcat: an application for exploring concatenative sound synthesis using MATLAB. In *Proceedings of Digital Audio Effects (DAFx)*.

Syrdal, A. K., Wightman, C. W., Conkie, A., Stylianou, Y., Beutnagel, M., Schroeter, J., and Makashay, M. J. (2000, October).Corpus-based techniques in the AT&T NextGen synthesis system.In *Proceedings of the International Conference on Spoken Language Processing. (ICSLP)*(Vol. 3, pp. 410-415).

Tan, L., and Karnjanadecha, M. (2003, September). Pitch detection algorithm: autocorrelation method and AMDF. In *Proceedings of the 3rd International Symposium on Communications and Information Technology* (Vol. 2, pp. 551-556).

Todd, P. M., and Loy, G. (Eds.) (1991). *Music and connectionism*.MIT Press.

Toiviainen, P., and Eerola, T. (2001). A method for comparative analysis of folk music based on musical feature extraction and neural networks.In *3rd International Conference on Cognitive Musicology* (pp. 41-45).

Tolonen, T., Välimäki, V., and Karjalainen, M. (1998). Evaluation of modern sound synthesis methods.

Tran, N. (1999, February). Generating photomosaics: an empirical study.In *Proceedings of the 1999 ACM symposium on Applied computing* (pp. 105-109).

Tzanetakis, G. (2002). *Manipulation*. *Analysis and Retrieval Systems For Audio Signals,* PhD. Thesis*, Princeton University*.

Tzanetakis, G., and Cook, P. (2002).Musical genre classification of audio signals.*Speech and Audio Processing, IEEE transactions on*, *10*(5), 293-302.

Ujlambkar, A. M., and Attar, V. Z. (2012). Automatic Mood Classification Model for Indian Popular Music.In *Modelling Symposium (AMS), 2012 Sixth Asia* (pp. 7-12).

Wessel, D. L. (1979).Timbre space as a musical control structure. *Computer music journal*, 45-52.

Wold, E., Blum, T., Keislar, D., and Wheaten, J. (1996). Content-based classification, search, and retrieval of audio.*MultiMedia, IEEE*, *3*(3), 27-36.

Zhang, T., and Kuo, C. C. J. (1998).Content-Based Classification and Retrieval of Audio. In *Advanced Signal-processing Algorithms, Architectures, and Implementations*, *3461*, 432.

Zils, A., and Pachet, F. (2001, December). Musical mosaicing.In *Digital Audio Effects (DAFx)*.

Appendices

The appendices are divided into the following four sections:

Appendix A – Sound examples

All the sounds examples referred in this thesis are included in the accompanying CD. The itemised list of the tracks and their descriptions are also provided in the following page.

Appendix B – Ethical Clearance

A copy of the ethical clearance received from the Faculty of Arts and Humanities Ethics Committee is attached in this section.

Appendix C – Full statistical test results

The workout for all the statistical tests done to verify the significance of the results obtained from the experiments in this thesis can be found under this section.

Appendix D – Full Result of Listening Test

The full result of the Perceived Similarity versus Perceived Interestingness Listening test from the charts presented in the Experimentations, Results and Discussion Chapter is included here should further clarification is seek.

Appendix E – Bash Scripts

This section contains excerpts of the original Bash scripts programmed in order to implement some of the algorithms in this thesis.

Appendix F – Record of Activities

This section provides a record of activities including the publications, public performances, as well as conferences, seminars, courses presented and attended during the study period.

Appendix A – Sound Examples

Below is an index of the sound examples in the CD, containing all the sounds which have referred to in this thesis.

Items A1 – A15 are audio files, presented in the .wav format, except for A4 and A15 which are .mp3 files. The sounds are best played on Windows Media Player or any equivalent audio player.

Item A16 is a .flv video file. It is best viewed using any standard video player such as the VLC media player.

	Description	Page	Tracks
A1	Examples of early sound experiments with CSS - Tracks 1_1, 1_1a, 1_2, 1_2a, 1_2b, 1_3, 1_3a are all results of self-experimentation	Section 1.1, p. 3	Track 1_1 – target (guitar) Track 1_1a – synthesised (tiger) Track 1_2 – target (country) Track 1_2a – synthesised (indris) Track 1_2a – synthesised (siamang) Track 1_3 – target (hip hop) Track 1_3a – classical
	- Tracks 1_4, 1_4a, 1_4b, 1_4c are created with other group members of the Augmented Sound tutorial, during the Sound and Music Computing Summer School 2010. Sounds were created using MEAPsoft, a basic CSS programme that automatically segments and rearranges audio recordings. • Head banger – finds the most common chunk length *L* and lengths related by a factor of 2, i.e. *L*/2, *L*/4, *L*/8, *L**2. The chunks are then shuffled to create a new piece with a clear beat. • HMM – uses a features file to train a simple statistical model of a song and uses it to randomly generate a new sequence of chunks.		Track 1_4 – target (motor) Track 1_4a – motor (head banger function) Track 1_4b – motor (HMM function) Track 1_4c – motor (rotation composer function)

	• Rotation – rotates the beats in each measure by a selectable number of positions. The number of beats/measure, the number of positions to rotate, and the direction of rotation can be changed. - Track 1_5 is a piece which had been composed using the CSS programme above (MEAPsoft), with the same group members at the said summer school, as part of the group's assignment. First, sounds that were 'uniquely Barcelona' were recorded during the Sound Walk activity around the city of Barcelona. From these sounds, a number of different audio features were extracted and then some trial-and-error manipulations followed (e.g. sort ascending pitch, segment mashup, intrachunks shuffle, etc.). Sounds that were thought best suited for the composition were selected and arranged (manually) using the Logic Pro 9 software. '*The Meeting Point*' tells a story of a walk by night in Barcelona. Everyone in the group came from different directions, and recorded sounds that were heard along the way to the meeting point. As some had walked, others had rode on the subway, took the tram and also cycled, various unique sounds managed to be recorded (i.e. the tram, the subway, whistling, chain of bicycle, night club, door slamming, etc.). Sounds are then manipulated by concatenating smaller sound units together, creating several new interesting sounds. The piece was premiered in a concert at closing night.		Track 1_5 – 'The Meeting Point' composition
A2	Comparison of sound examples between basic feature selection (no priority) and order dependent feature selection (with priority)	Section 4.1.2., p.97	Track 2 – target Track 2a – no priority Track 2b –with AHP

A3	Sound examples of outputs synthesised using the two approaches (order dependent features selection versus no priority)	Section 4.1.2., p.99	Track 3a – no priority Track 3b – with AHP
A4	Sound examples to demonstrate three homosonic segments with value `0.9835`	Section 4.2.2., p.108	Track 4a - `indis2.081.07828.wav` Track 4b – `siamang018.68045.wav` Track 4c - `whales004.05188.wav`
A5	Sound examples of the sound tracks from the 'Determination of Dominant Perceptual Attribute Test'	Section 4.3.1., p.117	Track 5_1 – target Track 5_1a – ans.1 (timbre) Track 5_1b – ans.2 (melody) Track 5_2 – target Track 5_2a – ans.1 (tempo) Track 5_2b – ans.2 (melody)
A6	Sound examples from Parametric Input Evaluation: Effect of Different Source File on the Concatenation Result Test Target file: *country* Source files: *siamang, whales, canary, tiger, rainforest, lemurs, indris* Feature: CTD Segmentation Mode: onset	Section 5.1.2., p.144	Track 6 – target Track 6a – siamang Track 6b – whales Track 6c – canary Track 6d – tiger Track 6e – rainforest Track 6f – lemurs Track 6g – indris
A7	Sound examples from Parametric Input Evaluation: Effect of Different Target File on the Concatenation Result Test Target files: *classical, country* Source file: *siamang* Feature: CTD Segmentation Mode: onset	Section 5.1.3., p.146	Track 7 – source (siamang) Track 7a – classical (original) Track 7b – country (original) Track 7c – classical (synth) Track 7d – country (synth)
A8	Sound examples from Audio Features Selection Evaluation: Effect of Different Audio Features on the Concatenation Result Test (Part 1) Target file: *country* Source file *indris* Features: CTD, RLF, FLX, ZCR, PCH Segmentation Mode: onset	Section 5.2.1., p.154	Track 8 – target Track 8a – CTD Track 8b – RLF Track 8c – FLX Track 8d – ZCR Track 8e – PCH

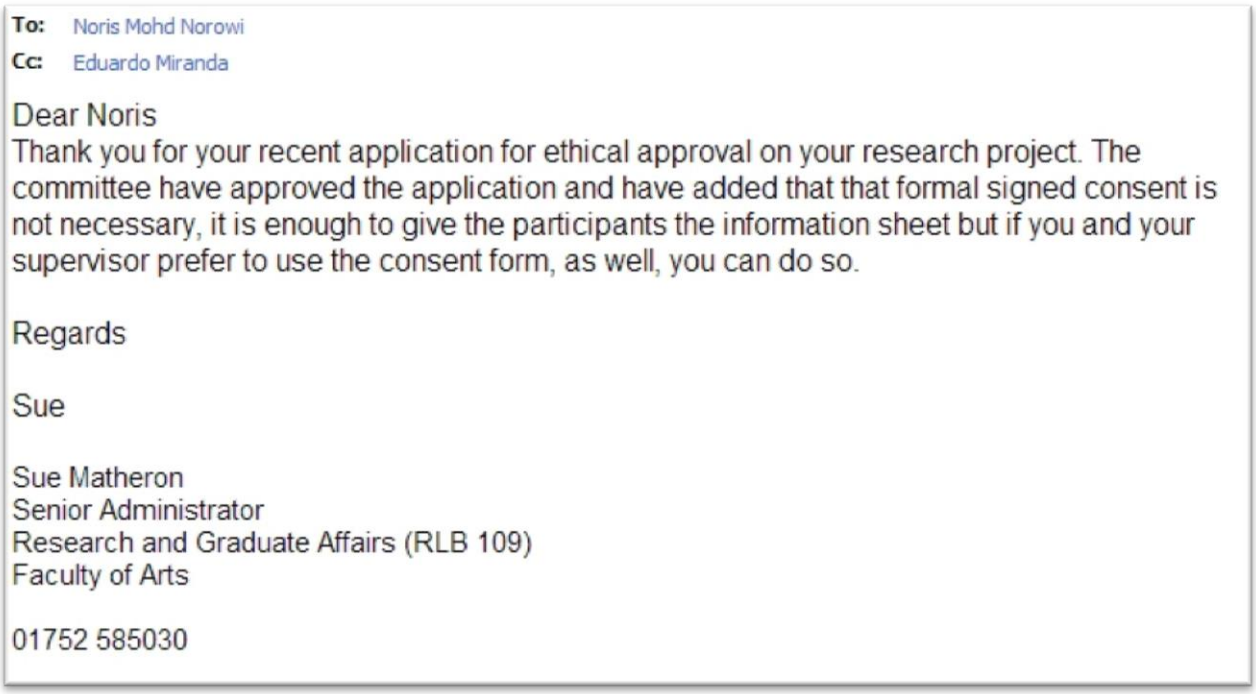
A9	Sound examples from Audio Features Selection Evaluation: Effect of Different Audio Features on the Concatenation Result Test (Part 2) Target file: *country* Source files: *indris* Features: CTD,CTD-RLF, CTD-RLF-FLX, CTD-RLF-FLX-ZCR, ALL Segmentation Mode: onset	Section 5.2.1., p.154	Track 9 – target Track 9a – CTD Track 9b – CTD-RLF Track 9c – CTD-RLF-FLX Track 9d – CTD-RLF-FLX-ZCR Track 9e – ALL
A10	Sound examples from Audio Features Selection Evaluation: Effect of Order dependent Feature Selection (Without versus With Order-dependent) Target file: *country* Source files: *indris* Features: CTD-RLF, CTD-FLX, CTD-ZCR, CTD-PCH Segmentation Mode: onset	Section 5.2.2., p.160	Track 10 – target Track 10_1a – CTD-RLF (w/o) Track 10_1b – CTD-RLF (w) Track 10_2a – CTD-FLX (w/o) Track 10_2b – CTD-FLX (w) Track 10_3a – CTD-ZCR (w/o) Track 10_3b – CTD-ZCR (w) Track 10_4a – CTD-PCH (w/o) Track 10_4b – CTD-PCH (w)
A11	Sound examples from Audio Features Selection Evaluation: Effect of Order dependent Feature Selection (Dual Features) Target file: *country* Source files: *indris* Features: CTD-RLF Segmentation Mode: onset Weights: CTD-RLF = 9, CTD_RLF = 1/9	Section 5.2.2., p.162	Track 11 – target Track 11a – CTD-RLF = 9 Track 11b – CTD-RLF = 1/9
A12	Sound examples from Audio Features Selection Evaluation: Effect of Order dependent Feature Selection (Triple Features) Target file: *country* Source files: *indris* Features: CTD-RLF Segmentation Mode: onset Weights: 3-7-5, 1-9-7, 1/3 - 1/7 - 1/5	Section 5.2.2., p.163	Track 12 – target Track 12a – CTD_RLF = 3, CTD_ZCR = 7, RLF_ZCR = 5 Track 12b – CTD_RLF = 1, CTD_ZCR = 9, RLF_ZCR = 7 Track12c – CTD_RLF = 1/3, CTD_ZCR = 1/5, RLF_ZCR = 1/7

A13	Sound examples from Search and Selection Evaluation: Effect of Enabling Concatenation Distance to Overcome Homosonic Segments (concatenation distance enabled vs. concatenation distance disabled) Target file: *country* Source files: *country, classical* Features: CTD Segmentation Mode: onset	Section 5.3.1., p.173	Track 13 – target Track 13a – enabled Track 13b – disabled
A14	Sound examples from Search and Selection Evaluation: Effect of Enabling Concatenation Distance to Overcome Equidistant Segments (concatenation distance enabled vs. concatenation distance disabled) Target file: *country* Source files: *rainforest* Features: CTD Segmentation Mode: onset	Section 5.3.2., p.178	Track 14 – target Track 14a – enabled Track 14b– disabled
A15	Sound examples from Listening Test (Similarity versus Interestingness)	Section 5.4.1., p.186	Track 15_1a – Mahler Track 15_1b – Mahler_Monkey Track 15_2a – Mozart Track 15_2b – Mozart_whales Track 15_3a – Meat Pvyr Track 15_3b – Meat Pvyr_indris Track 15_4a – Bush Track 15_4b – Bush_monkey Track 15_5a – Schoenberg Track 15_5b – Schoenberg_Braxton Track 15_6a – Cornershop Track 15_6b – Cornershop Remix Track 15_7a – Beddingfield Track15_7b– Lady Gaga Track 15_8a – Green Day Track 15_8b – The Kinks

A16	Example of potential application of sounds synthesised through *ConQuer.* *'Wooing Wails of Whales'* takes the synthesised sound generated from *ConQuer* and mixes it with a basic beat to portray the composer's vision of the sound of fights between several male whales for the right to mate. The piece had been inspired by the video of an epic humpback whale battle as can be viewed in: http://news.bbc.co.uk/earth/hi/earth_news/newsid_8318000/8318182.stm The piece starts off slow to represent the whales sizing their rivals up. The pace picks up with wail-like sounds from the whales to depict the fight, before slowing down and fading out to give impression that the mating right goes to one victorious whale. The piece was composed using the same target sound that was used throughout this study, the *Country* sound file, and the sound of whales singing as the source file. It was purposely kept really short, enough to convey the potential use of *ConQuer*.	Section 6.1, p.202	Track 16 – Wooing Wails of Whales

Appendix B – Ethical Clearance

B1 – Faculty of Arts and Humanities Ethics Committee Approval



To: Noris Mohd Norowi
Cc: Eduardo Miranda

Dear Noris
Thank you for your recent application for ethical approval on your research project. The committee have approved the application and have added that that formal signed consent is not necessary, it is enough to give the participants the information sheet but if you and your supervisor prefer to use the consent form, as well, you can do so.

Regards

Sue

Sue Matheron
Senior Administrator
Research and Graduate Affairs (RLB 109)
Faculty of Arts

01752 585030

B2 – Research Participant Information Sheet

Research Participant Information and Consent Form

You are being asked to participate in a research project. Researchers are required to provide a consent form to inform you about the study, to convey that participation is voluntary, to explain risks and benefits of participation, and to empower you to make an informed decision. You should feel free to ask the researchers any questions you may have.

Study Title: Human Perception of Audio similarity

Researcher and Title: Noris Mohd Norowi

Department and Institution: Interdisciplinary Centre for Computer Music Research, University of Plymouth

Address and Contact Information: Lab 206, Smeaton Building, University of Plymouth, PL4 8AA
+44 (0)1752 586219

PURPOSE OF RESEARCH:
You are being asked to participate in a research study of Human Perception and Audio Similarity, as part of a larger PhD research on Concatenative Sound Synthesis, conducted by the researcher above. From this study, the researcher hopes to learn the dominant factors that affect human judgment on sound similarity. Your participation in this study will take about 20-30 minutes.

POTENTIAL BENEFITS:
You will not directly benefit from your participation in this study. However, your participation in this study may contribute to the understanding of how humans perceive sound similarity, and in turn, as a model for a more intelligent automatic concatenative sound synthesis system.

POTENTIAL RISKS:
There are no foreseeable risks associated with participation in this study.

PRIVACY AND CONFIDENTIALITY:
The data for this project are being collected anonymously. Neither the researcher nor anyone else will be able to link the data to you. The data will be stored within the ICCMR research group, with hard copies stored in a locked cabinet, and only the researcher and her Directory of Study will have access to. The data will be kept confidential to the maximum extent allowable by law. The results of this study may be published or presented at academic conferences, but the identities of all research participants will remain anonymous.

YOUR RIGHTS TO PARTICIPATE, SAY NO, OR WITHDRAW
Participation in this research project is completely voluntary. You have the right to say no. You may change your mind at any time and withdraw.

COSTS AND COMPENSATION FOR BEING IN THE STUDY:
Unfortunately, you will not receive any money or other form of compensation for participating in this study.

CONTACT INFORMATION FOR QUESTIONS AND CONCERNS
If you have questions or concerns about your role and rights as a research participant, would like to obtain information or offer input, or would like to register a complaint about this study, you may contact, anonymously if you wish, the Faculty of Arts Research Ethics Committee, at 305 Roland Levinsky Building, University of Plymouth, or e-mail susan.matheron@plymouth.ac.uk.

Appendix C – Workout of Statistical Tests

C1 – Chi-squared test (Timbre-Melody) for Dominant Perceptual Attribute Test, p.118

Chi-square test results

P value and statistical significance:

Chi squared equals 1.895 with 1 degrees of freedom.
The two-tailed P value equals 0.1687
By conventional criteria, this difference is considered to be not statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Timbre	32	38	50.000%
2	Melody	44	38	50.000%

C2 – Chi-squared test (Tempo-Melody) for Dominant Perceptual Attribute Test, p.118

Chi-square test results

P value and statistical significance:

Chi squared equals 11.842 with 1 degrees of freedom.
The two-tailed P value equals 0.0006
By conventional criteria, this difference is considered to be extremely statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Tempo	23	38	50.000%
2	Melody	53	38	50.000%

C3 – Chi-squared test (Loudness-Melody) for Dominant Perceptual Attribute Test, p.118

Chi-square test results

P value and statistical significance:

Chi squared equals 7.579 with 1 degrees of freedom.
The two-tailed P value equals 0.0059
By conventional criteria, this difference is considered to be very statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Loudness	26	38	50.000%
2	Melody	50	38	50.000%

C4 – Chi-squared test (Tempo-Timbre) for Dominant Perceptual Attribute Test, p.118

Chi-square test results

P value and statistical significance:

Chi squared equals 0.053 with 1 degrees of freedom.
The two-tailed P value equals 0.8185
By conventional criteria, this difference is considered to be not statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Tempo	39	38	50.000%
2	Timbre	37	38	50.000%

C5 – Chi-squared test (Loudness-Timbre) for Dominant Perceptual Attribute Test, p.118

Chi-square test results

P value and statistical significance:

Chi squared equals 19.000 with 1 degrees of freedom.
The two-tailed P value is less than 0.0001
By conventional criteria, this difference is considered to be extremely statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Loudness	19	38	50.000%
2	Timbre	57	38	50.000%

C6 – Chi-squared test (Loudness-Tempo) for Dominant Perceptual Attribute Test, p.118

Chi-square test results

P value and statistical significance:

Chi squared equals 5.263 with 1 degrees of freedom.
The two-tailed P value equals 0.0218
By conventional criteria, this difference is considered to be statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Loudness	48	38	50.000%
2	Tempo	28	38	50.000%

C7 – 2x2 Contingency Table (Timbre-Melody) for Dominant Perceptual Attribute Test, p.122

Analyze a 2x2 contingency table

	Timbre	Melody	Total
Musician	10	36	**46**
Non-musician	22	8	**30**
Total	**32**	**44**	**76**

Chi-square without Yates correction

Chi squared equals 19.829 with 1 degrees of freedom.
The two-tailed P value is less than 0.0001
The association between rows (groups) and columns (outcomes) is considered to be extremely statistically significant.

Learn how to interpret the P value.

C8 – 2x2 Contingency Table (Tempo-Timbre) for Dominant Perceptual Attribute Test, p.122

Analyze a 2x2 contingency table

	Tempo	Timbre	Total
Musician	25	21	**46**
Non-musician	14	16	**30**
Total	**39**	**37**	**76**

Chi-square without Yates correction

Chi squared equals 0.429 with 1 degrees of freedom.
The two-tailed P value equals 0.5126
The association between rows (groups) and columns (outcomes) is considered to be not statistically significant.

Learn how to interpret the P value.

C9 – A simple Chi-squared test confirming no biased existed in terms of the participants' sex in the Phase 4 Listening Test, p.185

Chi-square test results

P value and statistical significance:
Chi squared equals 0.610 with 1 degrees of freedom.
The two-tailed P value equals 0.4349
By conventional criteria, this difference is considered to be not statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Female	23	20.5	50.000%
2	Male	18	20.5	50.000%

C10 – A simple Chi-squared test confirming no biased existed in terms of the participants' musical background in the Phase 4 Listening Test, p.185

Chi-square test results

P value and statistical significance:

Chi squared equals 1.976 with 1 degrees of freedom.
The two-tailed P value equals 0.1599
By conventional criteria, this difference is considered to be not statistically significant.
The P value answers this question: If the theory that generated the expected values were correct, what is the probability of observing such a large discrepancy (or larger) between observed and expected values? A small P value is evidence that the data are not sampled from the distribution you expected.

Row #	Category	Observed	Expected #	Expected
1	Musician	16	20.5	50.000%
2	Non-musician	25	20.5	50.000%

Appendix D – Result of Phase 4: Listening Test

The full result of the Perceived Similarity versus Perceived Interestingness Listening test, p.184

	SOUND		NON-MUSICIANS		MUSICIANS	
	Target	Corpus	Similarity	Interestingness	Similarity	Interestingness
1	Mahler_Ritenuto	Monkeys	2.520	2.280	3.188	3.563
2	Mozart_K457	Whales	2.480	2.480	3.188	3.125
3	MeatPurveyors_Clown	Indris	3.040	2.720	3.313	3.563
4	Bush_Speech	Monkeys	3.720	3.040	3.688	3.438
5	Schoenberg_String Qrt No4, Mv 1	Anthony Braxton_Saxophone	3.480	2.800	3.188	3.750
6	Cornershop_Brimful of Asha	Cornershop_Brimful of Asha (Remix)	3.920	3.800	3.938	3.313
7	Natasha Beddingfield_Pocketful of Sunshine	Lady Gaga_So Happy I Could Die	4.400	4.000	4.375	3.000
8	Greenday_Warning	The Kinks_Picturebook	4.240	4.320	4.313	3.000
		Average Score	**3.275**	**3.062**	**3.403**	**3.244**

Appendix E – Bash Scripts

This section provides excerpts of the original Bash scripts programmed in order to implement some of the algorithms in this thesis. To protect the intellectual property of this research, only parts of the scripts are listed here.

E1. Bash script for *userSelectFeatures* in non-prioritise feature selection

```
# asks users to select audio features for extraction from a check list
zenity --list --title "Selecting audio features for extraction" --text "Please
select the audio features to include below:" --checklist --column "" --column
"features" FALSE "ctd"  FALSE "rlf" FALSE "flx" FALSE "zcr" FALSE "pitch" >
/media/disk/script/prototype2/userSelectFeatures.txt;

bash chooseFeatures.sh


echo ""
echo "************************************************************************"
echo ""
```

E2. Bash script for *chooseFeatures* in non-prioritised feature selection (for 2 features, e.g. Centroid and Rolloff)

```
#once users have selected their features, this part finds the closest match between
the target sound and all possible source sounds in the database

#two features
 if [ "$line" == "ctd|rlf" ];

       then echo "Features chosen: centroid and rolloff" ;
       echo "Number of features chosen is 2 "
       targetSegments=`cat target.arff | wc -l`;
       echo "Number of target segment is " $targetSegments;
       sourceSegments=`cat source.arff | wc -l`;
       echo "Number of source segment is " $sourceSegments;
       echo $sourceSegments > numberOfSource.txt;

       cat source.arff | awk '{print $1 "\t" $2}' > chosenFeatures.txt ;
       cat target.arff | awk '{print $1 "\t" $2}' > chosenTarget.txt;
       sh matrix2.sh;

       paste feat1.txt feat2.txt | awk '{print sqrt(($1+$2))}' > distance.txt;


       for i in $(seq 1 $targetSegments)
       do
              head -$sourceSegments distance.txt > toSort.txt;
              paste toSort.txt songIndex.txt | awk '{print $1 "\t" $2}' >
toSortWithSongIndex.txt;
              sort -t, -n -k 1 toSortWithSongIndex.txt > sortedList.txt;
              head -1 sortedList.txt | awk '{print $2}' >> song.txt;
              head -1 distance.txt | awk '{print $1}' >> compDist.txt;
              bash removeDistance.sh;
              #sed -i '1,3359d' distance.txt;
       done

 fi;

echo ""
echo "************************************************************************"
echo ""
```

E3. Bash script for *matrix2* in non-prioritised feature selection (for 2 features)

```
for i in `cat chosenTarget.txt | awk '{print $1}'`
 do
   for j in `cat chosenFeatures.txt | awk '{print $1}'`
   do
       d=$(echo "scale=4; $i-$j" |bc)
       #echo $d
       e=`echo "scale=7; (($d*$d))" |bc)`
       echo $e >> /media/disk/script/prototype2/feat1.txt
   done
   echo "."
 done

echo "\n"

 for k in `cat chosenTarget.txt | awk '{print $2}'`
 do
   for l in `cat chosenFeatures.txt | awk '{print $2}'`
   do
       f=$(echo "scale=4; $k-$l" |bc)
       #echo $d
       g=`echo "scale=7; (($f*$f))" |bc)`
       echo $g >> /media/disk/script/prototype2/feat2.txt
   done
   echo "."
 done
```

E4. Bash script for *chooseFeaturesViaAHP* in order-dependent feature selection (for 2 features, e.g. Centroid and Rolloff). This script determines the order of the features, determines the weight of each features and then generates the priority vector, which is the value used to be substituted in Euclidean distance for finding the target distance between the target sound and source sound.

```
# Determine the order in which the features are considered more important

while read line
do

       #two features
       if [ "$line" == "ctd|rlf" ]; then

              echo "Features chosen: centroid and rolloff" ;
              echo "Number of features chosen is 2 "
              targetSegments=`cat target.arff | wc -l`;
              echo "Number of target segment is " $targetSegments;
              sourceSegments=`cat source.arff | wc -l`;
              echo "Number of source segment is " $sourceSegments;
              echo $sourceSegments > numberOfSource.txt;

              echo ""

              cat source.arff | awk '{print $1 "\t" $2}' > chosenFeatures.txt ;
              cat target.arff | awk '{print $1 "\t" $2}' > chosenTarget.txt;

              zenity --list --title "Arranging the features in order of importance"
--text "Please select the order of importance for the features from the list below"
--checklist --column "" --column "Orders" FALSE "ctd-rlf"  FALSE "rlf-ctd" >
userSelectOrder.txt;

       fi;
```

```
while read line
do
	#two features
	if [ "$line" == "ctd-rlf" ]; then
		firstFeature=ctd; secondFeature=rlf; numberOfFeatures=2;
		echo "You have chosen " $numberOfFeatures "features. Their order of
importance are as below: "
		echo "1st feature is: " $firstFeature;
		echo "2nd feature is: " $secondFeature;

		echo $firstFeature > firstFeature.txt;
		echo $secondFeature > secondFeature.txt;
		echo $numberOfFeatures > numberOfFeatures.txt;

		echo "";

	fi;

	if [ "$line" == "rlf-ctd" ]; then
		firstFeature=rlf; secondFeature=ctd; numberOfFeatures=2;
		echo "You have chosen " $numberOfFeatures "features. Their order of
importance are as below: "
		echo "1st feature is: " $firstFeature;
		echo "2nd feature is: " $secondFeature;

		echo $firstFeature > firstFeature.txt;
		echo $secondFeature > secondFeature.txt;
		echo $numberOfFeatures > numberOfFeatures.txt;

		echo "";

	fi;



#determine weight
while read line
do
	#two features --- SETEL
	if [ "$line" == "2" ]; then
		zenity --list --title "Determining weight" --text "How important is
the 1st feature over the 2nd feature? " --checklist --column "" --column "scale"
FALSE "9"  FALSE "7" FALSE "5" FALSE "3" FALSE "1" > weight1.txt;
		cat weight1.txt > combWeights.txt;
	fi;


#from combination of weight (i.e. 779), find the priority vector
while read line
do
	#two features --- SETEL
	if [ "$line" == "9" ]; then
		convertedFirstWeight=0.900;
		convertedSecondWeight=0.100;

		echo "The priority vector for " $firstFeature "is"
$convertedFirstWeight;
		echo "The priority vector for " $secondFeature "is"
$convertedSecondWeight;
		echo ""

		echo $convertedFirstWeight > convertedFirstWeight.txt;
		echo $convertedSecondWeight > convertedSecondWeight.txt;


	fi;
```

```
       if [ "$line" == "7" ]; then
              convertedFirstWeight=0.8750;
              convertedSecondWeight=0.1250;

              echo "The priority vector for " $firstFeature "is"
$convertedFirstWeight;
              echo "The priority vector for " $secondFeature "is"
$convertedSecondWeight;
              echo ""

              echo $convertedFirstWeight > convertedFirstWeight.txt;
              echo $convertedSecondWeight > convertedSecondWeight.txt;


       fi;

       if [ "$line" == "5" ]; then
              convertedFirstWeight=0.8333;
              convertedSecondWeight=0.1667;

              echo "The priority vector for " $firstFeature "is"
$convertedFirstWeight;
              echo "The priority vector for " $secondFeature "is"
$convertedSecondWeight;
              echo ""

              echo $convertedFirstWeight > convertedFirstWeight.txt;
              echo $convertedSecondWeight > convertedSecondWeight.txt;


       fi;

       if [ "$line" == "3" ]; then
              convertedFirstWeight=0.750;
              convertedSecondWeight=0.250;

              echo "The priority vector for " $firstFeature "is"
$convertedFirstWeight;
              echo "The priority vector for " $secondFeature "is"
$convertedSecondWeight;
              echo ""

              echo $convertedFirstWeight > convertedFirstWeight.txt;
              echo $convertedSecondWeight > convertedSecondWeight.txt;


       fi;

       if [ "$line" == "1" ]; then
              convertedFirstWeight=0.500;
              convertedSecondWeight=0.500;

              echo "The priority vector for " $firstFeature "is"
$convertedFirstWeight;
              echo "The priority vector for " $secondFeature "is"
$convertedSecondWeight;
              echo ""

              echo $convertedFirstWeight > convertedFirstWeight.txt;
              echo $convertedSecondWeight > convertedSecondWeight.txt;


       fi;


echo ""
echo "**************************************************************************"
echo ""
```

Appendix F – Record of Activities

E1 – Publications

1. Mohd Norowi, N and Miranda, E.R. (2011). "Order Dependent Feature Selection In Concatenative Sound Synthesis Using Analytical Hierarchy Process", In 8th Conference on Telecommunications, International Computer As A Tool Conference (EUROCON). Lisbon, Portugal
2. Mohd Norowi, N and Miranda, E.R. (2011). "Extending User Control In Concatenative Sound Synthesis", In 37th International Conference on Computer Music (ICMC 2011). Huddersfield, United Kingdom

E2 – Conferences Attended

1. Postgraduate Conference for Computing: Application and Theory, 6 June 2012, Plymouth
2. Making Sense of Sound, 20-21 February 2012, National Marine Aquarium, Plymouth
3. International Student Conferences, 27 May 2011, University of Plymouth
4. Postgraduate Society Short Conference, 17 March 2011, University of Plymouth
5. NeuroArts Conference, 10-11 February 2011, Royal William Yard, Plymouth
6. International Student Conferences, 29 May 2009, University of Plymouth

E3 – Summer School Attended

7th Sound and Music Computing Summer School, 17-20 July 2010, Universitat Pompeu Fabra, Barcelona, Spain. Lectures including: Soundscapes Compositions (Prof. Barry Truax), Music Content Processing (Prof. Fabien Gouyun), Recording Techniques (Enric Guaus), and Augmented Soundscapes: Real-time Machine Learning and Signal-Processing Techniques (Stefan Kersten)

E4 – Courses Attended

1. Overview to Searching and Accessing Information Resources – 23 May 2012
2. Word: Structuring your thesis – 5 March 2012
3. Preparing for the Viva - 28 February 2012
4. General Teaching Associates (GTA) Course (6 weeks from 26 January – 1 March 2012)
5. End Note Course – 29 November 2011
6. Impact Factor Course – 27 May 2011
7. Managing Stress – 19 November 2010
8. Academic Writing Workshop: Avoiding Plagiarism – 18 October 2010
9. The Transfer Process – 27 November 2009
10. Effective Reading Workshop - 11 November 2009
11. Effective Poster Workshop – 14 May 2009
12. Presentation Skills Part 1 & 2 – March & Feb 2009
13. Supercollider Intensive Course – December 2008 – February 2009
14. Latex Course – Jan 2009
15. A Dr. in 3 Years – Dec 2008

E5 – Workshops Attended

1. (Ab)Using MIR to Create Music: Corpus-based Synthesis and Audio Mosaicing, 21 July 2010, Universitat Pompeu Fabra, Barcelona, Spain. Instructed by: Dr. Diemo Schwarz
2. 4th Digital Music Research Network (DMRN+4), 22 December 2009, Queen Mary, University of London
3. Efficiency & Expression Symposium, 28 March 2009, University of West England

E6 – Seminars Presented

1. Departmental Seminar on 4 October 2012, titled "Issues in Concatenative Sound Synthesis". Presented at the Roland Levinsky Building, University of Plymouth, United Kingdom.
2. Departmental Seminar on 6 October 2011, titled "Advancement in the Concatenative Sound Synthesis Technology". Presented at the Plym Room, Babbage Building, University of Plymouth, United Kingdom.
3. Departmental Seminar on 18 November 2010, titled "Trends in Sound and Music Computing II: Music Content Processing". Presented at the Roland Levinsky Building, University of Plymouth, United Kingdom.
4. Departmental Seminar on 4 November 2010, titled "Trends in Sound and Music Computing I: Soundscape Composition". Presented at the Roland Levinsky Building, University of Plymouth, United Kingdom.
5. Departmental Seminar on 29 January 2010, titled "An Artificial Intelligence Approach to Concatenative Sound Synthesis". Presented at the Roland Levinsky Building, University of Plymouth, United Kingdom.
6. Departmental Seminar on 23 April 2009, titled "Improvement of the Automatic Genre Classification System of Traditional Malaysian Music Using Beat Features". Presented at the Roland Levinsky Building, University of Plymouth, United Kingdom.

E7 – Seminars Attended

1. Departmental Seminar, "What's Timbre Got To Do With It?", Dr. Duncan Williams, 29 November 2012
2. Departmental Seminar, "Idealogies of First and Last Draft", Sam Richards, 15 November 2012
3. Departmental Seminar, "Open Outcry: a Semi-Deterministic 'Reality Opera' where Traders exchange Stocks live by Call-and-Response Singing ", Dr. Alexis Kirke, 01 November 2012
4. Departmental Seminar, "The Techniques of Percussion Instrument", Christian Dimpker, 22 March 2012
5. Departmental Seminar, "Jamming With A Slime Mould", Prof. Eduardo Miranda, 8 March 2012
6. Departmental Seminar, "Sakuhachi As A Noise and Technology Interface", Dr. Mike McInerny, 23 February 2012
7. Departmental Seminar, "Subatomic Musical Instrument", Dr. Alexis Kirke, 12 January 2012
8. Departmental Seminar, "Cellular Automata Sound Synthesis", Jaime Serquera, 1 December 2011
9. Departmental Seminar, "The Warren: A Brain-Computer Music Interface", Joel Eaton, 17 November 2011
10. Departmental Seminar, "Writing Machine", Hanns Holger Rutz, 3 November 2011

11. Departmental Seminar, “Pulsed Melodic Processing - Using Music for natural Affective Computation and increased Processing Transparency”, Dr. Alexis Kirke, 20 October 2011
12. Public Lecture, “Creative Art, Creative Science: Their Connection and What They Tell Us About the Mind”, Arthur Miller, Jill Craigie Cinema, University of Plymouth, 10 February 2011
13. Departmental Seminar, “Electro-acoustic Music Notation”, Christian Dimpker, 10 March 2011
14. Departmental Seminar, “Rethinking the Supercollider Client”, Hanns Rutz, 27 January 2011
15. Departmental Seminar, “Neurogranular Sampler”, John Matthias, 13 January 2011
16. Departmental Seminar, “Application of Intermediate Multi-Agent Systems to Integrated Algorithmic Composition and Expressive Performance of Music”, Dr. Alexis Kirke, 7 October 2010
17. Departmental Seminar, “Cellular Automata Sound Synthesis with Multitype Voter Model”, Jaime Sequera, 16 March 2010
18. Departmental Seminar, “Articulating Noise and the Breakdown of the Interpretive Order”, Dr. Mike McInerney, 12 February 2010
19. Departmental Seminar, “Computer Wetware Project”, Dr. Anna Troisi and Mr. Antonino Chiaramonte, 4 December 2009
20. Integrated Engineering Services (IES) Seminar, “I-TALK: Integration and Transfer of Action and Language Knowledge”, Prof. Angelo Congelosi, 26 November 2009.
21. Departmental Seminar, “Music Neurotechnology for Sound Synthesis Using Artificial Spiking Neurons”, Dr Matthias, 20 November 2009
22. Departmental Seminar, “Artificial Social Composition”, Dr Kirke, 6 November 2009
23. Departmental Seminar, “A-Life for Music”, Prof. Miranda, 9 Oct 2009
24. Departmental Seminar, “Electroacoustic Music As A Devotion to Nature”, Dr Troisi, 18 June 2009
25. Departmental Seminar, “Diplomatic Guitar Player”, L Costalonga, 20 March 2009

E8 – Musical Performances

1. The Meeting Point (5' 5”), Nina Bjelajac, Noris Mohd Norowi, Adrien Sirdey, Thiago Duarte and Romain Pangaud. Premiered at the Sound and Music Computing Concert, 20 July 2010, Barcelona, Spain.
2. Performed 'Sekatian', 'Gambangan', 'Manuk Rawa' and 'Gopola' at the University of Plymouth Arts Degree Show, with the Chandra Gita Suara Gamelan Group, 17 May 2011, Roland Levinsky Crosspoint, University of Plymouth

E9 – Award Won

The Society of Artificial Intelligence and Simulation Behaviour (AISB) Prize for Best Poster, 6 June 2012, Postgraduate Conference for Computing: Application and Theory, Plymouth.