2010

Design and evaluation of dynamic feature-based segmentation on music

Befus, Chad R.

Lethbridge, Alta. : University of Lethbridge, Dept. of Mathematics and Computer Science, c2010

http://hdl.handle.net/10133/2531

DESIGN AND EVALUATION OF DYNAMIC FEATURE-BASED SEGMENTATION ON MUSIC

CHAD R. BEFUS
Bachelor of Science, University of Lethbridge, 2007

A Thesis
Submitted to the School of Graduate Studies
of the University of Lethbridge
in Partial Fulfilment of the
Requirements for the Degree

MASTER OF SCIENCE

Department of Mathematics and Computer Science
University of Lethbridge
LETHBRIDGE, ALBERTA, CANADA

Abstract

Segmentation is an indispensable step in the field of Music Information Retrieval (MIR). Segmentation refers to the splitting of a music piece into significant sections. Classically there has been a great deal of attention focused on various issues of segmentation, such as: perceptual segmentation vs. computational segmentation, segmentation evaluations, segmentation algorithms, etc.

In this thesis, we conduct a series of perceptual experiments which challenge several of the traditional assumptions with respect to segmentation. Identifying some deficiencies in the current segmentation evaluation methods, we present a novel standardized evaluation approach which considers segmentation as a supportive step towards feature extraction in the MIR process. Furthermore, we propose a simple but effective segmentation algorithm and evaluate it utilizing our evaluation approach.

Acknowledgements

This thesis would not have been possible without the inimitable support and guidance of my supervisor Dr. John Zhang. In addition, I owe my deepest of gratitude towards my colleague Chris Sanden for his collaboration, industrious assistance, and committed friendship. I am further indebted to Dr. Matthew Tata for his indispensable aid and advice towards conducting my perceptual experiments. I would also like to thank my committee members, Dr. Yllias Chali and Dr. Wei Xu, for their continual support and constructive criticisms throughout my studies. Furthermore, I would like to show gratitude to my many other colleagues for their exceptional comradeship and encouragement, especially that of Tarikul Sabbir and Mahmudul Hasan. Finally, I am ever grateful to my friends and family who have contributed their enduring reassurances and inspirations towards my success.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Music Information Retrieval

With the recent advent of digital distribution, the size of music collections has grown at an exponential rate. This has led to a call for the ability to index, organize, search and navigate music libraries in novel ways. It is to address this call that the area of *Music Information Retrieval* (*MIR*, for short) has been introduced. MIR is a multidisciplinary area, bringing together primarily the fields of *computer science*, *musicology*, and *psychology*. The general purpose of MIR is to develop techniques and systems which help facilitate and promote the creation and experience of music. This general purpose engenders a wide variety of problems in MIR.

1.1.1 Problems in MIR

Due to the multidisciplinary nature of MIR, problems in the area can range from social to artistic to purely computational. However, the scope of our work pertains primarily to those problems which are faced by the computational aspect of MIR. In this section we discuss a selection of some prominent problems from a computational aspect in MIR to provide a generalized sense of the area and its challenges.

One of the traditional problems in MIR is *genre classification*. Genre classification is the problem of automatically separating musical pieces into different groups such that each group uniformly represents a genre. One of the greatest challenges facing genre classification is the subjectivity of the definitions of different genres [6]. Solutions, such as *multi-genre labelling*, are current attempts to tackle this issue. While the definitions surrounding genres may be highly controversial, their ubiquitous use in the music industry, as a method of categorization, provides impetus for research efforts towards automatic genre classification. For an extensive survey on the genre classification problem see Scaringella *et al.* [49].

One of the more interesting and mainstream problems in MIR is *music retrieval*. The concept of music retrieval is to recover a specific piece of music based on a query piece, which can either be a short sample recording of the original or someone's attempt to replicate the original through alternative means, e.g., humming. The main challenge in music retrieval is the development of accurate representations of both the original and the query piece such that either one is easily identified by the other. One such system is proposed by McNab *et al.* [41]. Shazam [50] is a popular music-retrieval application for mobile devices. By recording a short piece of music, Shazam searches through its database and returns the most likely title and author of that piece. With a claimed user base of over 75 million, Shazam is an excellent example of the viability and importance of the music retrieval problem.

Music recommendation services are becoming more and more prevalent on the Internet. Generally they are structured to find songs which are, in some way, similar to an input song. The greatest challenge in music recommendation is to develop similarity metrics between songs. The most common approach for developing similarities is through so-called *collaborative filtering* [27]. The general concept behind collaborative filtering is to develop relationships between items which have been associated a certain number of times by other users. The relationship between these items can then be used as a form of similarity metric. *Apple Genius Sidebar*, a recommendation service in *Apple iTunes* [4], is based on collaborative filtering. Genius Sidebar recommends music to a user based on relationships found between her/his library and the libraries of others'. Other methods, such as the one proposed in [14], have been developed which use content analysis of music itself instead of collaborative filtering.

Playlist generation is the problem of developing a list of music based on a user's preferences. Much like music recommendation, it faces challenges of developing similarity metrics and typically resorts to techniques like collaborative filtering. *Apple Genius Playlist* [4] works in a similar manner to Apple Genius Sidebar. However, instead of recommending the music which a user does not yet own through collaborative filtering with other music libraries, it discovers and develops the relationships among the music within the user's current library. Through these relationships it is able to create a playlist for the user which has a reasonable recommendation quality. For further discussions on playlist generation, see [5].

The development of new approaches for *visualizing and navigating music* [16] is also an interesting challenge in MIR. A popular approach is to create visual topographic self-organizing maps of music which allow users to explore their libraries in multiple dimensions. This perspective highlights new relationships which may not have been apparent to users through a conventional metadata view (see below for more discussions on metadata). One of the greatest challenges to this approach is the selection of dimensions on which to base the topographical map.

For other problems and results in MIR, the interested reader is referred to [16].

1.2 MIR Systems

MIR systems are generally of two types: meta-based and content-based. Meta-based MIR systems use the extraneously attached data, i.e., metadata, such as *Title*, *Genre*, or *Author* of a song, to search for connections and further information in a music collection. Due to the limited amounts of metadata incorporated into the standard forms of music storage, such as MP3 [28], and the uncertain quality associated with that metadata, meta-based systems have a lower potential and are less reliable than their alternative. Content-based MIR systems, on the other hand, attempt to extract some representative information directly from the content of the music itself. For this reason, content-based systems provide greater potential towards the continued development of novel approaches to MIR problems. It is within the constraints of these content-based systems on which we will focus throughout the remainder of this thesis.

Content-based MIR systems face the onerous challenges of dealing with audio signal processing as an initial step towards developing any kind of usable representative information. For this reason many of the techniques used are paralleled with the field of *speech recognition* [16]. The dominant methodology is to break up an audio signal into short meaningful sections and then extract some defining representational features for each of them. These representations can then be used for higher level processes, such as searching or classification.

Speech recognition systems break a signal up into individual words then develop a text version, transforming the problem into the text retrieval domain. Unfortunately, for MIR, the problem is more complex. Music contains critical non-verbal information which must be captured and represented in some manner. It is this additional requirement which presents unique challenges defining content-based MIR systems.

Most content-based MIR systems follow a similar work flow to that outlined in Figure 1.1. The sequential nature of this work flow makes each step heavily dependant on those that precede it. This dependence intrinsically contributes to the definition of each step's purpose. That is, each step is intended to augment the functionality of the next in some significant manner. Initially, various operations must occur, such as *decoding*, to transform the music data into a format which is efficient for processing. This step is called *pre-processing*. The next step is to separate the music data into smaller meaningful sections or segments; this step is called *segmentation*. The *feature extraction* step, as explained

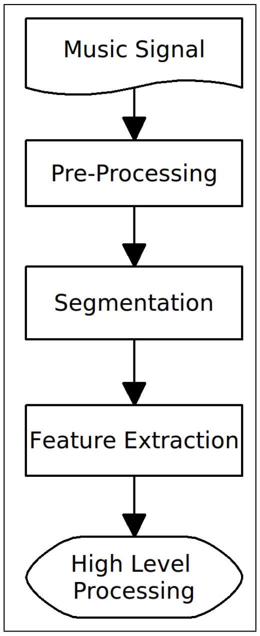


Figure 1.1: The flow of the standard content-based MIR process.

above, creates some representative characterization of each segment such that it can be effectively used for some higher level process. These higher level processes range from genre classification to playlist generation, as aforementioned.

1.3 Segmentation

This thesis focuses primarily on the step of segmentation in the MIR work flow. Generally speaking, segmentation is the problem of separating a music piece into smaller meaningful sections or segments such that they can be individually processed.

As Bruderer [12] puts it, segmentation is highly related to the *Gestalt Principles of Proximity and Similarity*: objects which are close temporally and appear similar tend to be grouped. Typically each individual segment is subject to the feature extraction process to create a representation for itself. The outputs from the feature extraction process over several segments usually becomes a usable representation of the original piece for high level tasks. The feature extraction process tends to be composed of some form of aggregate statistic. Therefore, extracting them from smaller self-similar segments will achieve a better representation quality than simply extracting them over the entire piece.

The challenge of segmentation is how to best select locations on which to separate a music piece to increase the representational accuracy of the feature extraction process. Since segmentation is the main focus in this thesis, we will discuss more on the algorithmic approaches to it in Chapter 4.

1.4 Some Issues Related to Segmentation

One incipient issue with the development of an segmentation algorithm is the confusion and ambiguity in the definitions of the term. This confusion has stemmed from three main areas of uncertainty: the definition of structure in music, the scale of segmentation, and the ultimate objective of segmentation.

The first cause of ambiguity around segmentation is the definition of musical structure. Segmentation is classically defined towards two different purposes, and quite often the two are used interchangeably. The first definition is for the purpose of discovering musical structure and the second is in order to group similar elements such that extracted statistics can be of maximum representational quality. These two definitions can be distinct, or the same, dependent on the subjectivity of the notion of musical structure. From a classical perspective, musical structure defines sections of music, like *intro*, *verse*, *chorus*, and *outro*. The ability to discover these musical sections automatically would provide several key advantages to MIR and much work has been done towards this goal [8, 36, 51]. However, from a data driven perspective, musical structure relates to the notion of the Gestalt Principles of Proximity and Similarity [12] previously mentioned. The disjunctive relationship between these two definitions causes a great deal of contention. It is clear, however, that this second definition is the one which will stipulate segments for maximizing the quality of the statistic extracted during the feature extraction process. Therefore, in our thesis we make use of this second definition.

The second issue contributing to the confusion around segmentation is the notion of its scale. Some [33, 42] make distinction between *short-term* and *long-term* segmentation. However, this distinction is rarely explicitly defined. Short-term segmentation involves segmenting at levels of high granularity and usually results in hundreds or thousands of segments for an entire song. Short-term segments are often called *frames* and are typically of a fixed length. It has been argued that using arbitrary fixed lengths can cause less accurate partitioning [42]. A variety of solutions have been proposed to compensate for this issue, ranging from overlapping frames to complex approaches such as onset-detection [32]. Long-term segmentation, on the other hand, attempts to create segments which cover a large portion of an entire song, separating it into usable sections for the feature extraction process. This thesis will differ between short-term segmentation and long-term segmentation by calling them framing and segmentation respectively.

The third, and in many ways the most critical, issue, on which confusion has arisen, is the conflicting ultimate objectives of segmentation. One of the greatest challenges in the development of segmentation algorithms is the generation of *ground truth*: a base set of data which is used to determine the accuracy of an algorithm. Ground truth is, by definition, equivalent to the desired output of an algorithm, and therefore, an inherent statement on the objective of that algorithm. In terms of a segmentation algorithm, its ground truth is commonly acquired through manually annotated segmentation based on human perception [22, 30, 31, 43, 44, 55]. Therefore, the objective of these segmentation algorithms is to simulate perceptual segmentation. However, this objective may be contentious to that defined by the MIR process: to provide self-similar segments, as explained above, from which the feature extraction process will return higher quality statistics. Whether or not an

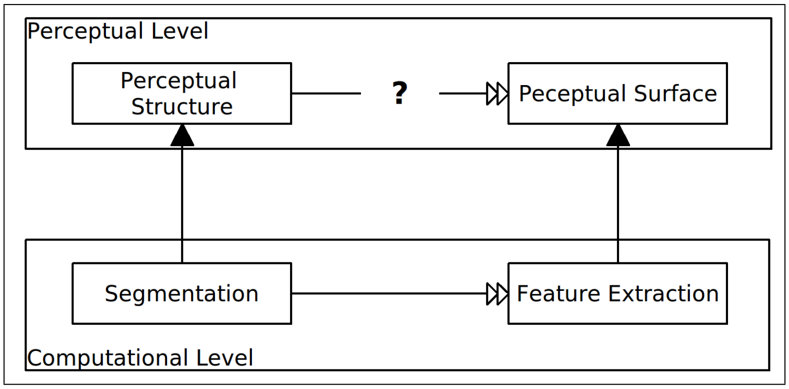


Figure 1.2: The relationship between perception and computation for segmentation and feature extraction.

exploitable relationship exists between perceptual segmentation and the quality of features which are to be extracted is, to the best of our knowledge, unknown. It is the purpose of this thesis to address this third issue of segmentation.

1.5 Our Contributions

As explained above, in terms of the objective of segmentation, there are two separate approaches in the current literature. We discuss the first approach below.

The first approach, simulating human perception of structure through segmentation, is shown in Figure 1.2. From the figure, the purpose of a segmentation is towards simulating perceptual structure and the feature extraction process is to simulate what is called the *perceptual surface features*.

The predominant justification for feature extractions in MIR is their ability to model the psychoacoustic features, i.e., *pitch*, *timbre*, *loudness* or *beat* [15][1] These psychoacoustic features are known as the perceptual surface features of music. The intuition behind this is to create a representation of a song which accurately models human perception.

[1] *Psychoacoustics* is the study of how humans perceive sound and the relationships between their perceptions and the actual sound itself [15].

Simulating human perception at this level (i.e., the perceptual level in Figure 1.2), to create an MIR system which is more acclimatized to its users, makes sense. Therefore, it also appears reasonable that simulating human perception at other stages in the MIR process will only further contribute to accommodating its users. For this reason, simulating human perception of structure through segmentation presents itself as an enlightened idea.

However, simulating perceptual structure through segmentation might actually detract from the quality of the feature extraction process. The dependent relationship between segmentation and feature extraction, due to the sequential nature of the MIR work flow, implies that the two should be considered as an inseparable process. While it is probably true that the perceptual structure of music is, in fact, based around some mixture of perceptual surface features, this relationship is yet undiscovered. We show this situation in Figure 1.2.

Our work in this thesis will look into whether this relationship, if any, exists to support the use of perceptual segmentation as a model for perceptual surface features. Furthermore, we will investigate whether or not using perceptual segmentation directly detracts from the quality of the feature extraction process.

Another drawback of using segmentation to simulate human perception is the challenge as how to evaluate its quality. This has led to the current and most prevalent method of segmentation evaluation in the literature, which is based on conducting human-involved experiments. The limitations of this method are vast. Primary among those limitations is the human-related cost in time with respect to the amount of data obtained. A typical experiment using this method could only involve 10 to 15 songs. Besides, in order to increase the number of songs, one often has to sacrifice the experiment quality by decreasing the number of human subjects assessing each song. A second limitation of human-involved experiments in this evaluation method is that the results are subjective and non-repeatable. That is, the ambiguity of what constitutes a good segmentation leads to highly variable conclusions. The inability to repeat experiments which are expected to produce identical results means that comparisons between experiments are challenging.

The second approach to segmentation is to consider it from a computational standpoint. From this perspective, the burden of simulating human perception is left to the feature extraction step. This approach is less popular but can provide fast, objective, repeatable methods of segmentation evaluation. Previous works making use of this approach have involved testing the accuracy of high-level tasks, such as classification, based on variations of segmentation algorithms.

However, involving high-level tasks in order to evaluate a segmentation algorithm can introduce more complexities and bias. Therefore, there is a need for a different and objective measure by which to design and evaluate segmentation algorithms. In our work, we propose and study a novel approach to the evaluation of segmentation from a computational standpoint, which does not rely on the use of any high-level task and, furthermore, stimulates a new but straightforward segmentation algorithm.

1.6 Outline

For the purposes of this thesis we divide our review of related works into each chapter separately in order to enhance the uniformity and clarity of our presentation. Several key related works are repetitively references over the chapters; however, their discussion in each chapter comes from the perspective directly supporting the theme of that chapter.

In Chapter 2 we conduct a series of descriptive perceptual segmentation experiments which promote our understanding of human perception through segmentation. Similar to previous studies, we ask human subjects to listen to a selection of music and mark locations they perceive as having significant changes. However, unlike previous works, they do this based on one surface feature at a time. We also ask them to segment using the method in the previous works. That is, based on their perception of structural change. From these results we attempt to analyze if there is an exploitable relationship between perceptual surface of music and the perceptual structure. The results are used to support our work in the following chapters.

In Chapter 3, we propose and study a novel evaluation approach for segmentation algorithms. It is intended to be objective and independent, and, as such, we hope to advance towards better quality segmentation algorithms. In our approach, we attempt to measure the information loss between an original song and the representation of it made by a segmentation algorithm. The segmentation algorithm is then held against the benchmark of simply partitioning the song into standard equal sized segments. This evaluation approach relies on the assumption that an intelligent segmentation algorithm should make better selections than an equidistant selection which is naive of the underlying data in the song. The amount of improvement of the segmentation algorithm over the equidistant segmentation is considered the measure of quality for that algorithm. Furthermore, we use the perceptual segments from our work in Chapter 2 to test if using perceptual segmentation detracts from the quality of feature extraction.

In Chapter 4, we propose a greedy merge-based segmentation algorithm and evaluate it using our approach described in Chapter 3. Our segmentation algorithm is feature-based, in that it makes its selections based on a supplied feature function and is intended then to select segments for maximizing the representation quality of that feature. Our segmentation algorithm not only selects segmentation locations but also returns the extracted feature as part of the same process. Though the merge selection for our algorithm is greedy, it is extendible in such a way as to allow for simple implementation of much more intelligent heuristics.

In Chapter 5, we present our conclusion and some discussions which include limitations and some future work.

In Appendix A, we discuss the software developed for our perceptual experiments in Chapter 2. Developed in Java, the software includes user tracking, a simple MP3 player, a segmentation selection recorder, and randomized instruction delivery, along with a simple graphical user interface.

In Appendix B, we introduce a framework, *Content-based Audio and Music Extraction Library* (*CAMEL* for short), to allow for fast and easy development of segmentation algorithms in C++. The framework is designed around concepts of simplicity and ease of use. Though there are several other frameworks available for use in MIR, most are developed towards providing platforms for high-level functionality, such as visualization, classification, etc. Because of this, other frameworks tend to be unnecessarily large and complex for our purposes. CAMEL implements a core group of the most popular feature extraction algorithms with a simple programming interface.

Chapter 2

Psychoacoustic Feature Based Perceptual Segmentation

2.1 Introduction

The most prominent MIR feature extraction functions are developed towards simulating a specific psychoacoustic feature of the human auditory system. These psychoacoustic features include *pitch*, *timbre*, *loudness*, and *beat* [15] and are called the surface features of music, as introduced in Chapter 1. In the previous works, segmentation algorithms are evaluated by their ability to model the perceptual structure of music: segments of music which are perceptually meaningful to the human listener. Perceptual segmentation is understood as the process to understand perceptual structure of music.

In this chapter we conduct a descriptive study in attempts to explore any relationship between the surface features and perceptual structure of music. Our experiment design models those of previous works as a means for generating ground truth (See Chapter 1.) towards segmentation evaluation. We extend the previous experiments by asking subjects to discern events not only at a structural level but also for the individual surface features. From our results, we attempt to analyze how the interplay of surface features affects the perceptions of structure. [1]

As identified by one early psychological work [17], perceptual segment boundaries [2] tend to occur at places other than rests, i.e., the starts and stops in a music piece. This indicates that a mixture of change in the surface features of music defines the boundaries in perceptual music structure. Many previous studies attempting to develop automatic segmentation algorithms have noted that a lack of knowledge into the importance of the

[1] This chapter is an extended version of our work in [9].

[2] Note that we often use the term segment in reference to a segment boundary since the former can be represented by the latter and vice versa.

individual surface features limited their ability to develop better algorithms [31, 55].

Furthermore, the evaluation of these segmentation algorithms is dependent on an assumption that there exists a relationship between these surface features and perceptual structure, as explained in Chapter 1. As aforementioned, it is the purpose of this chapter to explore this relationship.

This chapter is structured as follows. In Section 2.2, we review some of the previous relevant works in perceptual segmentation experiments on music. Section 2.3 describes the preparation of our experiment, including the description of the subjects, the music used, the experiment environment, etc. We present our results in Section 2.4 and analyze them in Section 2.5. Section 2.6 summarizes the chapter along with some discussions.

2.2 Related Works

Krumhansl [33] conducts a perceptual study investigating the relationship between musical ideas and perceptual segmentation. The term *musical idea* is introduced and utilized to identify any position which includes a change in features of *rhythm*, *pitch*, *register* and *dynamics*. Note that musical idea introduced is strongly related to the notion of musical surface features, as mentioned above.

As a result of this work, it is discovered that there exist significant correlations between perceptual segmentation and musical ideas. This supports the assumption to some extent that perceptual structure is defined by changes in perceptual surface. However, Krumhansl notes that, in music, there exist more musical ideas than segments. As such, it is apparent that changes in musical ideas do not automatically denote a perceptual segment. Therefore, either individual musical ideas have varied amounts of importance or some combination of change over them must occur to determine a segmentation event.

Most perceptual segmentation experiments are conducted with the assumption that a larger number of subjects selecting relatively the same position for a segment boundary increases the likelihood of that position being a good segment boundary. Bruderer [12] attempts to verify this assumption of correlation between the number of segment boundary indications by subjects and the *saliency* (or perceptual importance) of the boundary. To accomplish this, two perceptual experiments are conducted. The first, similar to previous perceptual segmentation experiments, asks subjects to split music into meaningful segments. The second experiment asks subjects to rate the saliency of the splits (the segment boundaries) produced in the first experiment. Bruderer finds that there is, in fact, a correlation between subject agreement on a segment boundary and its saliency.

However, because every position, when selected as a segment boundary, has some degree of saliency, it is concluded that, at least in terms of perception, "segments are not just Boolean truths but rather are represented by positions and their associated saliencies."

The majority of previous works on developing segmentation algorithms have, as a means of developing ground truth for segmentation evaluation, also run perceptual segmentation experiments [22, 30, 31, 43, 44, 55]. In order to present greater depth on their algorithms, discussions of the perceptual segmentation experiments are typically limited. Despite the limited amount of analysis on the perceptual segmentation experiments, several of these works have yielded key results.

In their early work in the area, Tzanetakis and Cook [55] discover that humans are consistent in their selection of segments. However, they extend this discovery by identifying a need to understand how each statistical feature, which represents some surface feature, should be weighted to better simulate human perception of music. Jian *et al.* [31] also distinguish the issue of selecting weights for statistic features as critical and, as a result, develop an individualistic feature weighting system to compensate for the lack of understanding in weighting. These works exemplify the need for further understanding into the relationship between the perceptual segmentation and the perceptual surface of music.

Conventionally, these studies ask subjects to segment music according to their own understanding of what a meaningful segment is. Our work, however, invites subjects to segment on individual surface features for comparisons against the more conventional perceptual segmentation. This will enable us to analyse perceptual segmentation in such a way as to reveal more insight into the relationship between perceptual surface and perceptual structure. To the best of our knowledge, there has been no previous study towards this end.

2.3 Experiment Setup

2.3.1 Human Subjects

For our experiments we select a total of 67 subjects, where 16 are male and 51 are female. The ages range from 17 to 54 years and have a mean of 23.1 years.

The musical skill of subjects is tracked as a self-reported judgement on a scale of 1 to 5. The exact definitions of the scale are reported in Table 2.1, where each number represents a distinct increase in exposure to music.

While musical skill has been repeatedly [12, 33] shown not to have any effect on subjects' abilities to select salient segments, we collect this information to show that our subject pool is unbiased. In our analysis, we find that the majority ($\sim$50%) of subjects claim to have a skill level of 3 while very few claim to have a level of 1 or 5. Therefore, the musical skill of our subjects mimics the behaviour of a normal distribution, which is desirable when conducting statistical experiments over a population.

We select the four aforementioned surface features, i.e., *pitch* (P), *timbre* (T), *loudness* (L), *beat* (B) and one additional feature, *global(G)*. The feature global defines the generalized structural segmentation of music wherein subjects are asked to find meaning-

Level	Skills	Description
1	Just listens	Never played or studied music to any degree. Only experienced music through listening.
2	Played some	Casually played an instrument or tried to learn on its own but have never taken lessons.
3	Took lessons	Took some lessons and learned the basics of an instrument (including singing) or played in a casual band regularly for some period of time.
4	Trained musician	Specialised and excelled in an instrument or played professionally.
5	Music major	Educated in music theory.

Table 2.1: Music skill levels and definitions.

ful positions of change.[3] For the full definition of each of the five (5) features used see Table 2.2.

We select a collection of eight (8) songs (to be discussed further in Section 2.3.2) for our experiments. However, to avoid over-training effects, subjects are only asked to listen to each song once, and, each time, for a different feature.

This follows the *within-subject* experiment design suggested by Levitin [15]. Within-subject experiments require that subjects take part in each of the different situations instead of splitting them into isolated groups. It is important to have at least 5-10 sets of unique data for each situation. For our experiment setup, since each subject only listens to each of the eight songs once, it takes five (5) subjects to create a single complete set of data, i.e., for each song for all the five features. On average there are 13.7 subjects participating in each song-feature test case. The numbers of subjects for each pair is reported in Table 2.3.

[3] We use the term global in our discussions to avoid preconception and confusion caused by the word “structure”, as discussed in Chapter 1.

Instruction Label	Instruction Text	Common Boundary Cue Descriptions
Pitch	Pitch is the property of a sound that allows the construction of melodies	Melody, Tone, Jump/Fall in Register, Key.
Timbre	Timbre is a measure of tone quality or colour	Instrument, Voice, Mood/Feeling, Texture.
Loudness	Loudness is the feature of a sound that is the primary psychological correlate of physical strength.	Volume, Level, Stops/Starts, Dynamic.
Beat	Beat is a measure of rhythmic periodicity	Tempo, Speed, Pace, Rhythm
Global	Typically associated as the beginning of a new idea of of the music or a significant change to the sound.	Phrase, Part, Section, Verse, and Theme.

Table 2.2: The description for each feature. Cue descriptions are from [12].

The subjects are instructed not to feel pressured to make selections if they feel that none exists. In such a case, those subjects are not to mark any segment selections and will not be counted as participating in that song-feature pair.

ID	P	T	L	B	G	Avg
Jaz	14	14	11	17	12	13.6
Pop	16	16	17	15	13	15.4
Eth	12	15	12	14	16	13.8
HaR	12	12	16	13	14	13.4
Ele	15	9	13	12	14	12.6
Cla	17	13	15	11	14	14
Pun	13	14	14	12	12	13
Roc	14	17	13	12	13	13.8
Avg	14.13	13.75	13.88	13.25	13.5	

Table 2.3: The number of subjects participating for each song and each feature.

2.3.2 Stimuli

The songs are a selection of eight (8) full polyphonic pieces (see Table 2.4) taken from the MIREX genre classification competition library which can be freely found on the website [19]. According to Bruderer [12], polyphony should have little effect on subjects' abilities to make quality segment selections.

To compensate for the facts, (1) that different styles of music might be more complex for subjects to identify structure in (as identified by Bruderer [12]), and (2) that musical styles might affect relationships between musical surface and structure, each song is selected as a subjectively representative of different genre of music.[4] As a result of this representativeness and for the sake of convenience, we hereby refer to songs by the genres they represent.

ID	Song Title	Artist	Album	Genre	Length
Jaz	Needs a Bridge	Scott Hill	Steps	Jazz	5:27
Pop	Nocturne	The West Exit	Nocturne	Pop	5:16
Eth	Didar	Kourosh Zolidar	Peacefull Planet	Ethnic	3:42
HaR	Release	Spinecar	Passive Aggressive	Hard Rock	4:11
Ele	Bass Vibrations	Domased	Return Back	Electronic	5:32
Cla	Albinoni	Le Serenissima	Per Monsieur Pisendel	Classical	2:23
Pun	Perfect Crime	Electric Frankenstein	Listen Up, Baby!	Punk	3:57
Roc	The Best In Me	Tom Paul	I Was King	Rock	3:14

Table 2.4: The songs selected and their associated genres and lengths.

[4] While genre has been identified as a weak representation for classification of music [6], we do not intend our results to divulge information regarding the individual genres themselves. Rather, we use genre labels to show the non-uniformity in the musical styles of the songs we have selected.

2.3.3 Apparatus

Over-the-ear headphones are used in order to counter for reverberation and noise pollution, as identified in [15]. Subjects are instructed to adjust the volume to a comfortable level before beginning the experiment. The experiments are conducted on a custom-made Java program which provides the definition for the current feature the subjects are to listen for (as per Table 2.2), the standard music player controls, and a segmentation marking button. At no time does the program return additional auditory feedback or any extent of extraneous visual feedback, to minimize distractions. For a full description of the program developed for our perceptual experiments, including screenshots, see Appendix A.

2.3.4 Procedure

The subjects are instructed to divide a song into smaller meaningful pieces (as per Bruderer [12]) based on a specific feature of music. They are explicitly told that there are no correct answers and everything is purely based on their own perception. They are also instructed not to feel pressured for time. At the beginning of each song a subject is presented a set of instructions that they are meant to read before listening. These instructions present a description of the feature for which they are currently listening. Furthermore, these instructions should help counter audio fatigue and the carry-over effect (i.e., the mood created by listening to some audio carries over to the next and can affect the subject's perception) as suggested by Levitin [15]. The subject is then instructed to hit a button each time that s/he believes a significant change has occurred according to the current feature. The instructions presented include the most common cue descriptions discovered for each psychoacoustic feature as discovered by Bruderer [12] and reported in Table 2.2.

The segmentation data for all subjects is then compiled into a master data file. Subjects' segments which coexist within a given amount of time difference, called a *window* (to be discussed shortly), are considered to be in agreement. Bruderer [12] shows that a high level of agreement within a window corresponds to a strong correlation to the saliency of a segment.

The size of an "optimal" window depends on the song and the feature currently selected. Krumhansl [33] uses a window size of 2 beats while Bruderer [12] cites that an "optimal" window size of 1.25 seconds was "in the same range as the one used in the majority of previous studies." A study by Bharucha and Stoeckig [10] finds that a reaction time of half (0.5) to one (1) second is needed to identify a single-note change. However, marking structural segments in polyphonic music is a far more complex task. To compensate for polyphony, other experiments [12] ask subjects to listen to the music ahead of time to create a measure of musical expectancy and lower the reaction time.

For the above reasons, when collecting and analyzing the results in our experiments, we evaluate all window sizes from one (1) to three (3) seconds, with each second divided into units of hundredths. Agreement among segment selections is calculated for each song-feature pair. The window size resulting in maximum agreement while minimizing the number of required windows is selected to report as "optimal". By this method, we have a variable "optimal" size and number of segments for each song-feature pair. Segments with less than 1/3 agreement among all subjects are ignored as noise. The window sizes for all song-feature pairs are reported in Table 2.5 and the respective numbers of windows are shown in Table 2.6.

ID	P	T	L	B	G	Avg
Eth	158	100	247	126	200	166
Jaz	296	124	253	259	257	238
HaR	100	100	166	129	212	141
Ele	100	178	223	119	189	162
Cla	192	174	276	196	252	218
Pop	182	283	223	113	100	180
Pun	184	179	122	100	100	137
Roc	121	228	100	187	100	147
Avg	167	171	201	154	176	

Table 2.5: The window size used for each song-feature pair in hundredths of a second.

ID	P	T	L	B	G	Avg
Eth	32	9	4	11	11	13.4
Jaz	24	11	9	18	10	14.4
HaR	14	21	18	15	11	15.8
Ele	23	23	20	21	19	21.2
Cla	13	10	8	6	7	8.8
Pop	14	15	19	11	16	15
Pun	16	18	13	11	13	14.2
Roc	12	10	8	8	6	8.8
Avg	18.5	14.63	12.38	12.63	11.63	

Table 2.6: The number of windows as used for each song-feature pair.

2.4 Experiment Results

2.4.1 Segment Saliency

Saliency of a segment for each song-feature pair is calculated as the average agreement over it. To calculate saliency, we slide a window of a given size across a list of the subjects' segments for a song-feature pair. The window which contains the maximum number of selections is declared a segment. The saliency of a segment is then calculated by the number of selections in that window divided by the total number of subjects participating in testing that song-feature pair. In the situation where a single subject has multiple selections falling within a single window, only one of them is counted.

Since saliency and agreement are highly correlated [12], we use saliency to describe both. The saliency for all the song-feature pairs from our experiments is reported in Table 2.7 and their analysis is described in Section 2.5.

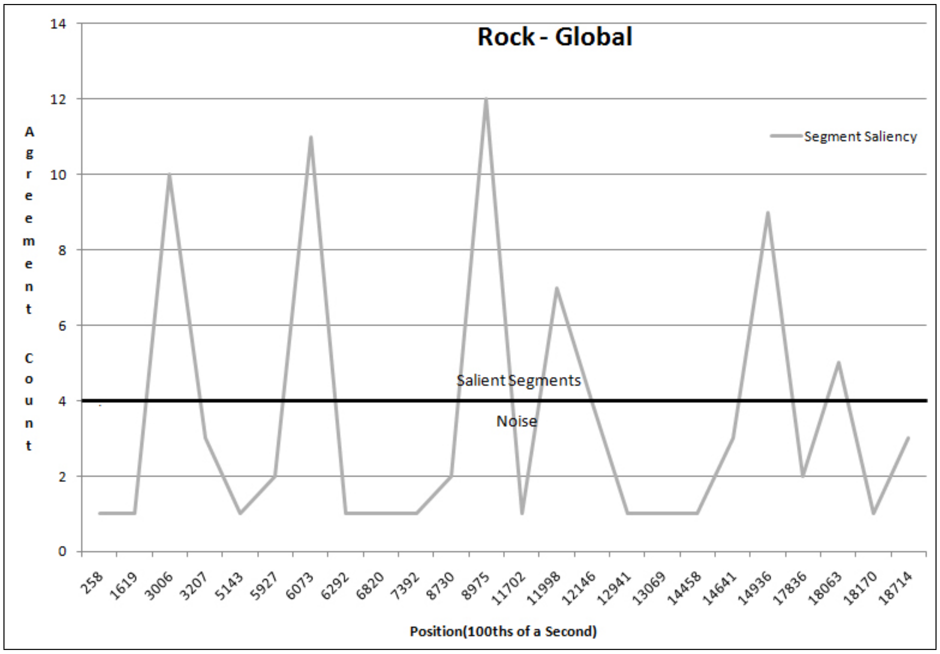


Figure 2.1: The agreement of selections by the 14 subjects over a Rock song using a window of 1 second.

As an example, the segmentation of a Rock song for the feature global is shown in Figure 2.1. It is obvious that there are six (6) salient segments selected above the noise threshold set at four (4).

ID	P	T	L	B	G	Avg
Eth	0.5	0.69	0.64	0.57	0.62	0.60
Jaz	0.5	0.51	0.58	0.48	0.52	0.52
HaR	0.45	0.55	0.6	0.6	0.61	0.56
Ele	0.57	0.67	0.58	0.58	0.72	0.62
Cla	0.45	0.43	0.58	0.42	0.5	0.48
Pop	0.59	0.72	0.58	0.53	0.63	0.61
Pun	0.5	0.56	0.55	0.52	0.65	0.56
Roc	0.46	0.61	0.59	0.65	0.7	0.60
Avg	0.50	0.59	0.59	0.54	0.62	

Table 2.7: The average saliency over all segments in each song-feature pair.

2.4.2 Feature Correlation

In order to calculate the correlation between our perceptual song-feature pairs, we create a correlation function. Our function matches the closest segments between two song-feature pairs, within a given window. We then weigh their difference in position against their combined saliency to create a correlation value.

Let $A[i]$ denote the i-th segment for a given song A under a feature. Function $S(\cdot)$ returns the saliency of a given segment, function $P(\cdot)$ returns the position (in hundredths of seconds) of a given segment in a song, and function $M(A[i],B)$ returns the index of the best matched segment in song B for i-segment in song A. With these functions, given two songs A and B, we calculate the weight between the i-th segment in A and B using:

$$Wt(A[i],B) = S(A[i]) + S(B[M(A[i],B)]) \tag{2.1}$$

Note it may be possible that there is no segment in *B* matching the *i*-th segment in *A*. In this situation the second part of the weight calculation returns 0.

For the *i*-th segment in song *A* we need to calculate the correlation based on position such that a larger distance between *A* and *B* on the *i*-th segment corresponds to a smaller *positional correlation* (*PC* for short). This is accomplished using function

$$PC(A[i],B) = (w - \mid P(A[i]) - P(B[M(A[i],B)]) \mid) \tag{2.2}$$

For our experiments we set window size w to be three (3) seconds since it is the maximum size tested. In the case where no matching segment is found in *B* for $A[i]$, $PC(\cdot)$ returns 0. From this, we calculate a *saliency-weighted one way correlation* (*OWC* for short) from *A* to *B* using function:

$$OWC(A,B) = \frac{1}{n}\sum_{i=0}^{n}(Wt(A[i],B) * PC(A[i],B)). \tag{2.3}$$

This calculates the distance between two songs in terms of their segments' saliency, where n is the number of salient segments in song *A*. It is obvious that $OWC(A,A)$ achieves the maximum.

We then calculate our total correlation between a particular song-feature pair for songs *A* and *B*.

$$Correl(A,B) = 1/2\left(\frac{OWC(A,B)}{OWC(A,A)} + \frac{OWC(B,A)}{OWC(B,B)}\right) \tag{2.4}$$

For each song we calculate the correlations among features. Through these correlations, we are looking for a consistent pattern over the various genres such that we can exploit it for development of segmentation evaluations. The results are collected and reported in Table 2.8 for Ethnic, Table 2.9 for Jazz, Table 2.10 for Hard Rock, Table 2.11 for Electronic, Table 2.12 for Classical, Table 2.13 for Pop, Table 2.14 for Punk, and Table 2.15 for Rock. We also report the average of these tables in Table 2.16. These results are analysed in Section 2.5.4.

2.5 Experiment Analysis

2.5.1 Window Sizes

As previously mentioned, we define the window to be the area in which subjects have selected segments that are considered to represent a single position within the music. The sizes of these windows are dynamically allocated based on a function of maximizing perceptual segment saliency while minimizing the number of segments. This means that each song has a variable "optimal" window size. From these window sizes, reported in Table 2.5, we observe certain patterns. One of the most interesting patterns is that the faster beat songs (such as Punk, Rock, Hard Rock) have lower average window sizes while the slower beat songs (such as Classical and Jazz) have larger windows. This confirms the assumption made by Krumhansl [33] that feature beat is used to determine window size.

Observing the data column-wise allows us to analyze it from a feature perspective. It is interesting to note that, on average, feature beat has the smallest window size while feature loudness has the largest. This may have something to do with the reaction times and expectancy in people over different kinds of features as how they pertain to music structure.

It is also observed that our "optimal" window sizes tend to be high with comparison to the previous related works. This difference can be accounted for based on the comparative complexity of our problem or the lack of musical expectancy in our subjects which existed in many of the previous works.

2.5.2 Number of Segments

Once an “optimal” window size is selected, it determines the number of perceptual segments that are to be counted for a given song-feature pair. Therefore, the number of segments for each song-feature pair is variable, and as such, produces some interesting results. From Table 2.6, we observe that feature pitch has the highest number of average segments. This confirms the assumption that, perceptually, feature pitch changes far more often than other features, such as feature beat. It can also be observed that the lowest number of segments occurs, on average, for feature global. As explained above, it is assumed that a mixture of the other features must change in order to define a perceived change in feature global. Alternatively, the ambiguity of the definition for feature global could also explain the low count. However, the saliency of feature global, to be discussed below, refutes this.

Observing the data from a song perspective, we find that the slower Classical song has the least number of segments on average while the faster Electronic song has the highest. This is a direct result of the window sizes being correlated with feature beat as explained in Section 2.5.1. It is surprising however that the Rock song has equally as few segments on average as the Classical song. Perhaps this is due to the clearly defined stereotypical structure of the Rock song contributing to musical expectancy.

With respect to individual interesting song-feature pairs, we observe that there is an especially low number of segments recorded for feature beat with respect to the Classical song. This result is especially interesting because we find in Table 2.12 that beat is one of the most important features for defining perceptual structure (feature global) for that song. Another interesting result from the number of segments in Table 2.6 is that the least number of segments occurs in loudness with respect to the Jazz song but the most number of segments occurs in feature pitch with respect to the same song. This may have something to do with the high amount of variations and complexities in the Jazz song.

2.5.3 Segment Saliency

The saliency of a segment is the number of subject selections made within that window divided by the number of subjects participating in the song-feature pair. Because this value is dynamic for each segment we average it for all salient segments (that is, segments which are above our noise threshold) and report it for each song-feature pair in Table 2.7. These results give us a rough concept of the quality of our segments, as explained by Bruderer [12].

From a song perspective, we notice the Classical and Jazz songs have the lowest average saliency. This observation could be explained because of the weakly identifiable structures or variability and complexity inherent in the styles of music. To support this, the highest saliency is found among those songs which are of highly predictable perceptual structure, such as the one in the Rock, Pop, and Electronic songs.

In terms of features, it is interesting to see that the highest saliency is associated with the ambiguously defined feature global. Recall the definition of feature global in Section 2.3.1. This lends support to the notion that a consistent and definable perceptual structure does exist. However, this does not mean that such a structure can be defined by any single function. It is also interesting to note that feature timbre has the next highest saliency. This is consistent with much of our following results where feature timbre is the most relevant feature, on average, for determining perceptual structure. Feature pitch has the lowest saliency and subjects are noted to have commonly claimed that it is the most challenging of the five features on which to identify structure.

2.5.4 Feature Correlation

We also look at the data in more detail for each song. For each of the songs, we can see the relationships among the various features. We expect that a certain amount of correlation should exist between any feature pairs. However, in our results there are certain cases which are more interesting, and in this section we attempt to highlight them. Ultimately, we hope to find a set of consistent relationships among our results such that we can define a correlation between perceptual surface and structure of music.

	P	T	L	B	G
P	1	0.81	0.86	0.66	0.71
T	0.81	1	0.72	0.74	0.73
L	0.86	0.72	1	0.73	0.68
B	0.66	0.74	0.73	1	0.61
G	0.71	0.73	0.68	0.61	1

Table 2.8: The correlation matrix of features for genre Ethnic.

From Table 2.8, we see a high correlation between features of loudness and pitch, followed closely by a high correlation between features timbre and pitch. However, there is a low correlation between feature pitch and feature beat. This probably explains the slightly lower correlation between features of pitch and global.

Despite this, it makes sense that feature pitch would be important as it is the main element in the melodic-based music used for the Ethnic song. We find that feature timbre, closely followed by feature pitch, contributes the most to the perceptual structure of Ethnic music.

	P	T	L	B	G
P	1	0.54	0.52	0.43	0.49
T	0.54	1	0.63	0.58	0.59
L	0.52	0.63	1	0.46	0.55
B	0.43	0.58	0.46	1	0.50
G	0.49	0.59	0.55	0.50	1

Table 2.9: The correlation matrix of features for genre Jazz.

In Table 2.9, we show our results for the Jazz song. Here, feature timbre correlates highest to defining perceptual structure of music. Timbre is a measurement of sound texture and, since the Jazz song bases its structure around changes in instruments, it makes sense that feature timbre would have a high level of precedence when it comes to describing perceptual structure. We also find that feature timbre correlates highest towards the other features, highlighting its importance in determining their roles as well.

	P	T	L	B	G
P	1	0.80	0.74	0.71	0.75
T	0.80	1	0.86	0.77	0.82
L	0.74	0.86	1	0.76	0.78
B	0.71	0.77	0.76	1	0.70
G	0.75	0.8	0.78	0.70	1

Table 2.10: The correlation matrix for genre Hard Rock.

For the song representing Hard Rock, our results in Table 2.10 show that feature timbre once again presents the highest correlations to all other features. This is especially true with its correlation to feature loudness. This finding seems practical since our Hard Rock song is defined by its loudness and texture. The overall perceptual structure of our Hard Rock song seems to be most heavily correlated with feature timbre, closely followed by feature loudness.

	P	T	L	B	G
P	1	0.70	0.64	0.69	0.73
T	0.70	1	0.70	0.72	0.75
L	0.64	0.70	1	0.75	0.72
B	0.69	0.72	0.75	1	0.78
G	0.73	0.75	0.72	0.78	1

Table 2.11: The correlation matrix for genre Electronic.

As would be expected from our Electronic song, we find that its perceptual structure is correlated with feature beat. In Table 2.11, we can see that feature beat also correlates highly to the song's features of loudness and timbre. Only feature pitch is slightly more correlated with other features than beat. Secondary to feature beat we find that feature timbre once again defines perceptual structure in our Electronic song.

	P	T	L	B	G
P	1	0.74	0.53	0.62	0.49
T	0.74	1	0.69	0.59	0.52
L	0.53	0.69	1	0.70	0.52
B	0.62	0.59	0.70	1	0.62
G	0.49	0.52	0.52	0.62	1

Table 2.12: The correlation matrix for genre Classical.

From Table 2.12, we find it interesting that the Classical song has its perceptual structure most heavily correlated with feature beat. This is especially interesting, if we consider that feature beat has so few perceptual segments for the Classical song. There is also a high correlation between feature pitch and feature timbre as well as between feature loudness and feature timbre. This reliance on feature timbre is understandable since our Classical song has its structure based in the addition of new instruments or changes in the texture and loudness.

	P	T	L	B	G
P	1	0.68	0.76	0.84	0.77
T	0.68	1	0.75	0.85	0.79
L	0.76	0.75	1	0.86	0.83
B	0.84	0.85	0.86	1	0.90
G	0.77	0.79	0.83	0.90	1

Table 2.13: The correlation matrix for genre Pop.

As we might expect, as shown in Table 2.13, our Pop song's perceptual structure is, by far, most heavily correlated to feature beat. In fact, according to the table, all features are heavily correlated to beat. This tells us that the perceptual structure based on any other features is largely controlled in some way by feature beat. Feature beat is followed by feature loudness in high correlation to perceptual structure.

	P	T	L	B	G
P	1	0.61	0.65	0.71	0.62
T	0.61	1	0.75	0.76	0.83
L	0.65	0.75	1	0.68	0.74
B	0.71	0.76	0.68	1	0.79
G	0.62	0.83	0.74	0.79	1

Table 2.14: The correlation matrix for genre Punk.

In Table 2.14 we show our results for our Punk song. Again, we find that feature timbre is the highest correlated to feature global. This result is followed closely by a high correlation of feature beat to feature global. We also note that there is a strong inter-correlation between features of timbre, loudness, and beat. However, of great interest is the unique relationship between feature pitch and feature beat as shown in the table.

	P	T	L	B	G
P	1	0.65	0.70	0.47	0.59
T	0.65	1	0.69	0.63	0.69
L	0.70	0.69	1	0.69	0.66
B	0.47	0.63	0.69	1	0.66
G	0.59	0.69	0.66	0.66	1

Table 2.15: The correlation matrix for genre Rock.

With respect to our Rock song in Table 2.15, we observe that all features tend to correlate to feature loudness. As is the common case, however, we find that feature timbre is the most predominant feature when it comes to correlations to feature global.

2.5.5 Averaged Correlations

In our quest to find a unifying function by which to relate the various perceptual surface features and perceptual structure, we aggregate the data in the tables from the previous section. This result is reported in Table 2.16, where we find feature timbre is the most highly correlated surface feature when it comes to perceptual structure. It also seems clear that features timbre, loudness, and beat are all highly correlated to one another. Feature pitch, on the other hand, seems to be the least relevant towards perceptual structure in the average case. However, we note that feature pitch is important in some specific styles of music, such as Ethnic and also is independent from the other features, making it equally interesting.

	P	T	L	B	G
P	1	0.69	0.67	0.64	0.64
T	0.69	1	0.72	0.70	0.72
L	0.67	0.72	1	0.70	0.68
B	0.64	0.70	0.70	1	0.70
G	0.64	0.72	0.68	0.70	1

Table 2.16: The correlation matrix for all features for all genres.

2.6 Summary and Discussions

Many of the results found in our experiments coincide with our intuition. For example, the driving perceptual feature of Electronic and Pop is beat. Such results in turn give support our experiments.

In our analysis, we have found that, when determining window sizes in order to group subjects' perceptual segments, the amount of time needed is variable, dependent on the feature and style of the music. In the general case, it appears that, the faster the beat the smaller the window size that is needed. As a result of this, we also observe that the number of segments tends to increase with the speed of the beat. From these results we conclude that feature beat plays a dominant role in the development of window size when perceptually segmenting musical data. This also implies that human reaction time, which is closely related to window size, is somehow linked to the beat of the music.

From our saliency of subjects segment selections, we observe that feature global maintains the greatest amount of agreement but also tends to have the lowest number of segments. This tells us that subjects have a much easier and more consistent time recognizing the general structure of music without considering other individual surface features. We further note that among the surface features, timbre has the highest saliency while feature pitch has the lowest. This means that the structure of music is easiest to recognize perceptually in terms of feature timbre and is the most challenging in terms of feature pitch. Perceptual structure is also better recognized in styles of music which have straightforward and predictable forms, such as Pop or Rock. For styles of music which are more variable and complex in form, such as our Jazz and Classical song, we find subjects have a greater challenge.

In support of using human experimentation for segmentation evaluation we have found that subjects are able to consistently discern perceptual structure in music, especially in its generalized global form. However, there is no apparent individual pattern which carries over between the feature correlation tables for all styles of music. Therefore, we can only take conclusions from the aggregated cases. That is, on average, feature timbre appears to have the greatest association with perceptual structure.

However, it would be naive to ignore some specific results, which show that certain styles of music are not perceptually represented in terms of feature timbre. For example, in terms of the Electronic song, its perceptual structure is better defined by surface features beat and loudness.

Therefore, it does not appear that there is any single exploitable relationship between the psychoacoustic surface features and the structures of music. Such a relationship seems to be variable, dependent upon musical style. Therefore, we find no apparent justification for deriving a correlation between the segmentation towards perceptual structure and the feature extraction towards perceptual surface, as shown in Figure 1.2.

While using human perception as a framework for ground truth might be the classical option, contrary to the study by Martin *et al.* [39], it is not necessarily the desirable solution when it comes to segmentation. Until an exploitable relationship between perceptual surface and structure can be found, we cannot claim that segmentation, which simulates perceptual structure, is advantageous in terms of supporting feature extraction, which simulates surface features.

For this reason, a new method is required to evaluate segmentation based on their effects on feature extraction. This is the purpose of the next chapter in this thesis.

Chapter 3

An Objective Evaluation of Segmentation Quality

3.1 Introduction

Segmentation is the process of separating a music piece into self-similar segments such that high quality representative statistics can be extracted from them, as introduced in Chapter 1. Traditionally, the evaluation of segmentation algorithms has relied on human perceptual experiments in which subjects are asked to segment music based on their own perception of the musical structure.

Jensen [30] states that segmentation has an “inherent perceptual and subjective nature”. However, as explained by Tichy [53], human experimentation should only be used when the implication of insights to be gained outweighs the costs of human-involved studies. In the case of perceptual segmentation, any insights of a music piece can only come at a perceptual level, which may not reflect an understanding of the music data itself. The costs associated with human experimentation as a means to generate adequate ground truth for segmentation are enormous. The primary cost is the time expense in terms of the labour intensive task. This cost is, furthermore, unreasonable relative to the size and quality of the result obtained. Due to the multitude of limitations in human experimentation, it is more desirable to seek alternative approaches for evaluating segmentation algorithms.

In this chapter, we will propose a novel and objective approach for evaluating segmentations. The effectiveness of our approach is demonstrated through experiments. The merit of our approach rests on allowing segmentations to be evaluated against an objective and independent ground truth instead of human perception.

The chapter is organized as follows. In Section 3.2, we will review related works in the area. Section 3.3 further discusses the current challenges we face when evaluating segmentations. Section 3.4 introduces our objective approach to the problem. In Section 3.5, we will set up our experiments to support the effectiveness of our approach. Section 3.6 reports our results and provides analysis. Finally, Section 3.7 summarizes our work with discussions and future work.

3.2 Related Works

3.2.1 Automatic Segmentation

Most proposed segmentation algorithms are modelled around the detection of “significant” local changes. The basic structure of this approach can be outlined in a series of steps. First the data is cut into frames, i.e., small units of equal length (or dynamic if short-term segmentation is used). A set of features are then extracted from each frame, which are then used to measure the difference between adjacent frames. After that, locations of maximal changes are considered “good” for segmentation. Variations of the approach usually consider different features to be extracted and the detection methods of changes.

Tzanetakis and Cook [55][1] develop an automatic segmentation algorithm following this general method. Features including *Spectral Centroid*, *Spectral Flux*, *Zero Crossing Rate*, and *Root Mean Squared Energy* are considered and represented as feature vectors. Using the derivative of the distance function between the feature vectors over adjacent segments, top-*k* segments are selected. Although it is claimed that the approach is not modelled based on the human auditory system, the segmentation quality is evaluated against a perceptual segmentation experiment. Jian *et al.* [31] follow a similar approach but use features of *roughness*, *periodicity pitch* and *loudness*. Unlike the work in [55], Jian *et al.* use a

[1] As mentioned in Section 1.6 we use many references repetitively. However, each use concentrates on a perspective which supports the theme of the chapter in which it resides.

ranking algorithm to seek locations of individual features where maximal changes occur. Segmentation evaluation is also done against a manual perceptual experiment.

Foote's work in [22] describes an algorithm which uses a similarity matrix between adjacent frames in the FFT log magnitude spectrum [38]. It uses a kernel to find a measure of self-similarity and cross similarity within the matrix. The difference between these similarities gives a novelty score. The locations corresponding to the extremes of the novelty scores are used for segmentation. The work does not report any measurement of segmentation accuracy. Both Peiszer *et al.* [43] and Ong [42] design experiments using similar approaches as above and, in addition, they both use perceptual segmentation experiments for evaluation. Other approaches can be found in [30, 57] and are discussed in greater detail in Chapter 4.

3.2.2 Perceptual Segmentation Experiments

Krumhansl [33] conducts an experiment showing how subjects relate perceptual segmentation to musical ideas. Musical ideas are defined to be changes in a combination of *rhythm*, *pitch*, *register*, and *dynamics*. It is found that there is a strong correlation between perceptual segmentation and musical ideas and, in general, the former is a subset of the latter.

Bruderer [12] shows the long-held assumption that a strong agreement among perceptual segments correlates to a strong perceptual saliency of those segments. It is also shown that there is little relation between a subject's musical skills and her/his ability to pinpoint salient segments.

Our perceptual segmentation experiments in Chapter 2 show, among other things, that the number of segments, the size of agreement window, and the saliency of segments vary greatly based on musical styles and features used.

3.2.3 MIR Benchmarks

Due to the concerns that research in MIR is not objective in its current state, Downie [18] organises an MIR/MDL evaluation project, calling for the need for standardized collections of music and retrieval metrics. Various criteria and approaches are proposed and discussed, such as developing a benchmark collection of music data for experimentation [24, 45, 46]. While such a benchmark would greatly help the task of evaluating segmentations, it is extremely challenging to compose a collection which is universally available and holistically representative of music. As noted by Tichy [53], the ultimate subjectivity of collection composition makes it inherently the weakest part of any benchmark. Furthermore, the development of a standardized collection does not inherently derive a standardized method of segmentation evaluation.

Abdallah *et al.* [1] use a measure of evaluation based on comparisons between segmentations annotated by human experts and those from their segmentation algorithm. The comparison is formed through a directional Hamming distance between segments by matching best candidates, or segments with maximum overlap. The sum of the differences between each of the matched segments creates the directional Hamming distance between them which is then normalized to the length of the overall track. By generating the matching from the two different directions, that is, matching the ground truth to their algorithm's output and vice versa, they are able to generate both measures of the missing boundaries and of extra boundaries.

In addition, the same work [1] proposes a second method which calculates an information-theoretic measure over the same ground truth. It accomplishes this measure by assigning labels to each segment in the ground truth and its algorithm's output. In one direction, by calculating the conditional entropy from the ground truth to their algorithm's output, the work obtains a measure of the amount of segments missing in the output. From the other direction, calculating the conditional entropy from its algorithm's output onto the ground truth, it generates a measure of the number of extra segments in its output. It also points out some limitations of basing the segmentation evaluation method on ground truth developed by a human expert. Specifically, "the expert's segmentation should not be taken as Platonic truth: equally valid segmentations, depending on the application, can be formed at greatly different time scales; in addition, in real music there is often a degree of ambiguity as to the exact point of transition between one segment and the next: an ambiguity which was not reflected in the expert's judgement."

Lukashevich [37] extends the information-theoretic method described by Abdallah *et al.* [1] for the evaluation of segmentation algorithms. The proposed method generates two scores called "over-segmentation" and "under-segmentation", which map the false positives and negatives that occur during the matching between automatic segmentations and human segmentations. The scores are normalized to show whether the matching is perfect or not. Lukashevich addresses and attempts to handle the limitation of using the pure conditional entropies, which returns a non-negative score with an unrestricted maximum, making comparison of scores between songs meaningless. To compensate, Lukashevich normalizes the conditional entropies by the maximal conditional entropy for a song under question.

Lukashevich's work [37] is unique in its sole purpose of standardizing a method of evaluation for segmentation. However, it is still dependent on the weakness of human subjective segmentations. Lukashevich states that, like the one in Abdallah *et al.* [1], the ground truth for his method is annotated by an "expert" instead of using a perceptual segmentation study. However, he does not mention what defines an "expert." Several works [11, 12, 33] have shown that human skill level with music has little or no influence on their ability to select consistent and accurate perceptual segmentations. As such, there is little difference between using human experts or experimentation (involving multiple subjects) to generate ground truth. As explained by Bigand and Poulin-Charronnat [11],the extensive exposure to music in everyday life has transformed non musicians to become “experienced listeners” who are not so different from musically trained listeners. In addition, Tichy [53] outlines the dangers of using experts over experimentation, specifically citing that it is unscientific to rely on ”so-called experts who fail to support their assertions with evidence.”

To the best of our knowledge, no standardized approach, which is not based on generated ground truth via human perception, either from experts or experimentation, has been proposed for evaluating segmentation algorithms.

3.3 Issues Related to Current Segmentation Evaluations

A key challenge to segmentation is an unbiased evaluation method by which segmentation quality can be measured and compared. Qualitative evaluation is difficult [22, 54], due to the plethora of data sets, features selected, and various parameters. As stated by Lukashevich [37], the evaluation of “song segmentation algorithms is not a trivial task” and “there is no commonly established standard way of performing a segmentation evaluation.”

3.3.1 Perception-Based Evaluations

In this section we discuss the potential deficiencies of the previous segmentation evaluation approaches, where evaluations are conducted through perceptual studies, i.e., subjects are asked to manually annotate a set of music for comparative purposes. It is for these reasons that the costs of human perceptual studies outweigh the potential insight and ground truth provided by them.

Reaction Time

The first issue stems from a subject's reaction time, which varies from one to the next. In perceptual segmentation experiments, subjects are asked to select positions in time which represent accurate segment boundaries. This presents a challenge in aggregating the selections of multiple subjects down to a single usable set of "ground truth." The prominent solution involves finding areas of dense annotations by subjects and considering those as the selected segments. This, in itself, creates further complexities, such as the definition of a reasonable size to allow for annotation area (also known as window size, as discussed in Chapter 2). Bharucha [10] tests the reaction times of subjects on individual pitch changes and finds that an average of 0.5 seconds is needed before reaction. On real-world polyphonic music, such a reaction time would be slower on average. As explained in Chapter 2, the size of a window for annotation area changes with each individual music piece and with different surface features. Differences in window size will dramatically change the segmentation accuracy.

Subjectivity

The second issue in terms of perceptual segmentation studies is the subjectivity of music structures. The correlation between the perceptual structure of music and the actual data itself is far from being explored. There have been several studies which relate specific statistical features to surface features, for example, *Mel Frequency Cepstral Coefficients* (*MFCCs*) to *Timber* [7] or *Spectral Energy Flux* to *Beat* [2]. However, no study has conclusively reported a single representation which completely expresses the human perception of music and we believe it is unlikely that any such representation exists.

Furthermore, subjects in perceptual experiments are often challenged with the concept of what determines a segment position. Typically a description is provided to guide subjects. However, these descriptions are either too generic, leading to ambiguity, or too specific, resulting in findings which have been predetermined and therefore are an unscientific selections.

Subjects' Background

The third issue facing the use of perceptual segmentation as ground truth is the inability to control many factors influencing each subject's selections. Subjects' exposure to music, musical tastes, age, gender, environment, etc. can all have possible effects on their selections. To our knowledge, no study has been conducted as how to control these factors or to measure their effects on perceptual segmentation quality.

Dataset

A fourth issue is the data size and selection involved. Perceptual studies are time intensive and, as such, are extremely difficult to perform over data sets with sizes that are statistically significant. One solution to this is to merge results from multiple perceptual studies, as the one in [43]. However, merging experiments conducted in different environments and under differing conditions can lead to complications. Data selection is further worsened by a variety of issues as outlined by Downie [18], such as legal issues, coverage, infrastructure, etc.

To summarize, as noted by Futrelle [24], an unbiased approach is necessary, such that different segmentations can be evaluated objectively.

3.3.2 Classification-Based Evaluation

A second and less common approach for evaluating the quality of a segmentation is through genre classification. The idea is that a better segmentation algorithm will lead to a higher genre classification accuracy. This assumes that genre is an objective measure through which musical classification can be used as a metric. The fallacies of this assumption, however, are discussed at length in [6] and are directly tested in our work [48]. In essence, genre classification is a purely perceptual and subjective activity and, as such, no true ground truth (that is, ground truth which has high agreement) can be achieved. Furthermore, there is no evidence that increasing accuracy in classification will have any correlation to an increase in accuracy in other unrelated high-level tasks, such as Music Retrieval.

3.4 An Objective Approach for Segmentation Evaluation

The basis of our approach is that the purpose of segmentation is to maximize a feature-based representation of an original music piece. Therefore, simulation of human perceptual structure is not the objective of segmentation, eliminating the need for human experimentation. Note that the importance of maintaining the human perceptual element when analyzing music is not lost here. But we are shifting the psychoacoustic burden to the feature extraction step. That is, we are representing human perception of music only through the feature extraction process.

A simple and naive way of segmenting a music piece is through the process of static segmentation, i.e., each segment is of a uniform size and the entire piece is divided into equal portions. Any other segmentation methods[2] should at least result in a better representation than the static counterpart because of the latter's naivety towards the data. How-

[2] For the purpose of this thesis we call them *dynamic segmentations*

ever, this does not necessarily mean that a dynamic segmentation is guaranteed to always perform better. It is theoretically possible that a static segmentation could be “optimal” for a particular music piece.

Our approach uses static segmentation as a basis for comparison, measuring improvement as a metric for evaluating the quality of a segmentation. This allows us to make comparisons which always use the same feature extraction functions and the same music pieces. Furthermore, this approach allows us to easily make comparisons between experiments of different segmentations, which is an acknowledged [37] limitation of other evaluation methods.

Our approach to evaluating segmentation quality is designed with the following criteria in mind.

1. Algorithm independence - In order to have a fair evaluation of various segmentation algorithms, it is desirable that one have a uniform basis for comparison. This basis needs to be well understood and easily implemented. In this way, for any comparison with others’ algorithms, one only needs to implement its own segmentation algorithm. Therefore, we do not need to compare two segmentation algorithms directly in order to evaluate them.

2. Evaluation objectivity - Objectively evaluate whether a segmentation increases the feature-based representativeness accuracy of a music piece.

3. Data independence - Evaluation is not dependent on any specific data set of music. Ideally, comparisons between experiments would take place on identical data sets. However, this can lead to a training bias effect. An evaluation method should allow for approximate comparison between experiments conducted using different data sets. Furthermore, the evaluation method itself should not be dependent on the style of music being used. That is, there should be no inherent requirement in the

evaluation method that the music have specific qualities, such as "semantically distinguishable and repeated parts" as expected in [37]. Requirements such as these limit the usability of an evaluation approach.

4. Feature flexibility - An unbiased evaluation approach should not depend on any specific feature extraction to be used and should work in any domain of representation, including statistic, spectral, peak, etc. However, it should be noted that comparisons made between segmentation algorithms can only work if they are performed using the same set of features.

In the following discussions, we refer to the diagram of our approach shown in Figure 3.1.

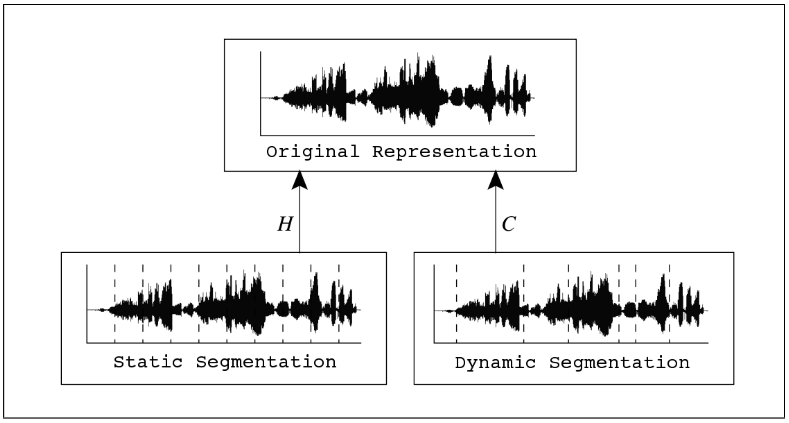


Figure 3.1: The structure of our evaluation approach.

3.4.1 Original Representation

Given a music piece, comparing the distance from any representation resulting from a segmentation to its original data implies that we must first describe the original by the same features. Many feature extractions, such as the one for MFCCs (which involves the data from the frequency domain of the given music piece), requires a certain number of data points from the music piece as the input. Therefore, to create an accurate representation of the original under which all features can be calculated, we must first partition the data points from the original music piece into frames of a particular small size. The frame size must be small enough such that it is insignificant to the sampling rate of the time domain. We can then extract a feature value for each frame to create a representation of the original. For the purposes of this thesis, we call these original frames collectively the *original representation* for the given music piece.

3.4.2 Comparing against Original Representation

To compare the segment selections from a segmentation with the original representation, we adopt a simple scoring method.

The objective of the scoring is to measure the distance between a segmentation and the original. To do so we must guarantee that every part of the original is measured against the appropriate part of the representation created from the segmentation.

Suppose that we want to evaluate a segmentation for song S on feature f, which is of k dimensions.

We represent S, through the segmentation algorithm, by $S = \{S_0, S_1, \cdots, S_i, \cdots, S_{m-1}, S_m\}$. For each segment selection S_i , we apply feature function f on it. For the original representation of S, we denote it by $F = \{F_0, F_1, \cdots, F_j, \cdots, F_{n-1}, F_n\}$. For each original frame F_j which overlaps with segment S_i, we apply feature function f on it.

Consider the Euclidean distance between $f(F_j)$ and $f(S_i)$. If S_i does not completely cover F_j, i.e., F_j starts before S_i or F_j ends after S_i, then we need to multiply the distance by the percentage that frame F_j is covered. We sum up the distance between selection segment S_i and each frame F_j covered by S_i and denote it as $D(i, j)$.

This summed score over all segment selections is the distance between the original representation and the given segmentation. The smaller the score, the better the segment selection.

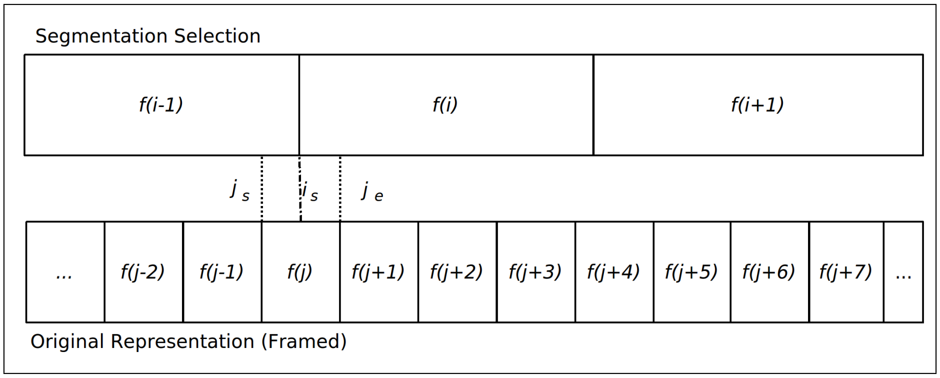


Figure 3.2: Comparison of a segmentation selection to the original representation (framed).

For example, in Figure 3.2 we denote the start position and end position of frame F_j as j_s and j_e, respectively. We also denote the start position of segment S_i with i_s. To evaluate the distance between the individual frame F_j and segment S_i we can use the Euclidean distance function:

$$Eu(S_i, F_j) = \sqrt{\sum_{p=0}^{k-1} (f(S_i)_p - f(F_j)_p)^2} \tag{3.1}$$

If F_j is not completely covered by S_i we must weigh its importance against the percentage of it covered by S_i. To do this we calculate a weighting function:

$$Wt(S_i, F_j) = \frac{j_e - i_s}{j_e - j_s} \tag{3.2}$$

Note that in Figure 3.2 that the first portion of F_j not covered by S_i is in fact covered by the previous S_{i-1}. However, because its evaluation over the end of the segment S_{i-1} instead of the start, its function $Wt = \frac{(i-1)_e - j_s}{j_e - j_s}$ instead, where $(i-1)_e = i_s$.

If F_j is completely covered by S_i then $Wt(i, j) = 1$. After we have calculated the weight we multiply it by the distance to get a weighted distance score which is then summed into the total distance score between the two representations:

$$D(S_i, F) = \sum_{j=0}^{l-1} Wt(S_i, F_j) * Eu(S_i, F_j) \tag{3.3}$$

where l is the number of frames which are covered by segment S_i.

For the entire song S, its distance to the original representation F can now be calculated using the following equation:

$$DS(S, F) = \sum_{i=0}^{m-1} D(S_i, F) \tag{3.4}$$

Normalisation of DS is not possible due to our inability to construct a maximum dissimilarity in a feature space f from the original representation F.

3.4.3 Static Segmentation vs. Dynamic Segmentation

Comparison between a dynamic segmentation and a static segmentation over feature f is now, given Section 3.4.2, simple, as shown in Figure 3.1.

For a music piece, we must first create a static segmentation which has the same number of segments as the one output from the dynamic segmentation. We then calculate the distance, denoted by H, between the static segmentation and the original representation by following the steps in Section 3.4.2. We do the same for the distance, denoted by C, for the dynamic segmentation and the original representation. If $C \geq H$, then the *difference indicator* $di = -1 * (1 - (H/C))$. Otherwise, $di = 1 - (C/H)$. It is easy to see that $di \in [-1, 1]$. The rational for this difference indicator is, if the dynamic segmentation results in a distance far away from the original representation, with respect to the static segmentation, di approaches -1, and conversely, di approaches 1.

Finally, since we should repeat the above process over multiple music pieces, we average the difference indicator, for feature f, over all of them. This capability, to summarize the segmentation score over multiple music pieces, is something other methods have had difficulty addressing as discussed in [37].

Because the static segmentation and dynamic segmentation are conducted on the same music piece, using the same number of segments and the same feature function, the comparison between them is objective.

3.5 Experiment Setup

In order to show the effectiveness of our objective segmentation evaluation approach, we have designed some experiments. Chapter 4 provides complementary experiments as well.

For our experiments, a frame size of 512 is used as a parameter to obtain the original representation. This size is selected due to its conventional use in calculating the frequency domain for a given music piece in the literature and since our original data is sampled at a rate of 44100Hz, this size is quite small in comparison.

We use the process described in Section 3.4 to compare the segment selections from our perceptual segmentation study in Chapter 2. Recall that the study asks subjects to segment music based on different surface features including *Pitch*, *Timbre*, *Loudness*, *Beat* as well as the perceptual recognition of music structure *global*. In the following, we collectively call them surface features, for the sake of convenience.

3.5.1 Dataset

For our experiments we use eight (8) polyphonic songs. They are selected as subjectively representatives of the different genres available in the MIREX genre classification competition library which is available online [19]. The songs are shown in Table 2.4. Our results have no bearing or reliance on the genres themselves. Rather, we use the term genre here to simply show that we are not picking homogeneous data. For the sake of convenience, we use the genre categorization of the songs as their identifiers for the remainder of the chapter.

3.5.2 Statistical Features

The features are extracted from the eight songs using our MIR framework CAMEL, which is further described in Appendix B. These features are *Spectral Centroid* (*SC*), *Spectral Irregularity J* (*SI*), *Spectral Flatness* (*SF*), *Spectral Tonality* (*ST*), *Spectral Slope* (*SSl*), *Spectral Spread* (*SSp*), *Spectral Rolloff* (*SR*), *Spectral Loudness* (*SL*), *Spectral Sharpness* (*SSh*), and *Mel-frequency Cepstral Coefficients* (*M*).

3.6 Results and Analysis

3.6.1 Comparisons

The experiment on the subjects perceptual segments follows the steps outlined in Section 3.4. However, since subjects are not segmenting on specific statistical features but rather on surface features, as explained in Chapter 2, we extract all of the statistical features for each surface feature and report three (3) different results. The first result is the case where we use the best statistical feature for the song-surface feature pair and is reported in Table 3.1. The second case is where we use the worst statistical feature for the song-surface feature pair and is reported in Table 3.3. The third case is the average over all surface features for each of the song-statistical feature pairs and is reported in Table 3.5.

For the best and worst cases we also report which feature is used in Table 3.2 and Table 3.4 respectively.

3.6.2 Analysis

The Best Case

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc	Avg
Pitch	50	9	2	-2	48	3	7	18	17
Timbre	23	13	9	-4	50	4	4	10	13
Loudness	29	12	-10	-5	50	16	9	13	14
Beat	35	5	-4	-4	47	1	3	1	10
Global	45	8	-2	-4	52	10	1	1	14
Avg	36	9	-1	-4	49	7	5	9	14

Table 3.1: The best case difference indicator, as a percentage, between the static and perceptual segmentation.

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc
Pitch	SSl	SF	SI	SSh	SI	SSh	M	SSh
Timbre	SSl	ST	SSh	SSh	SSl	SSh	M	SSh
Loudness	SSl	ST	SSh	SR	SSl	SSh	M	SL
Beat	SSl	SSp	SSh	SC	SSl	SSh	SR	SR
Global	SSl	SC	SI	SC	SSl	SSh	M	SSp

Table 3.2: The features used to represent each of the surface features for each song in the best case.

In Table 3.1, we note several different interesting observations. The lowest score for the perceptual segmentation is on the Ethnic-loudness pair. For this pair, even with the best case feature of Spectral Sharpness (SSh) being extracted, as shown in Table 3.2, the static segmentation had 10% better in difference indicator. On the other hand, we can see in Table 3.1, perceptual segmentation performs quite well for the Electronic song in general and specifically so for feature global. For this pair, using spectral slope (SSl), as shown in Table 3.2, the perceptual segmentation improves 52%, as shown in Table 3.1, against the static segmentation.

From the best case difference indicator in Table 3.1, we can see that the average over all the songs is only a 14% improvement over static segmentation. This amount of improvement is not significant enough to claim that it is a result of anything more than noise. In certain cases, that is, for specific songs and specific features, we can see that human subjects actually do quite well. However, these specific cases do not justify using human perception as an evaluation method. That is, designing an evaluation method around specific musical styles or features would violate the requirements of data independence and feature flexibility outlined in Section 3.4.

Overall, even in the best case, human perceptual segmentation does not reflect a significant improvement over just statically apportioned segments which have a much lower cost to be created and evaluated.

The Worst Case

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc	Avg
Pitch	-2	-70	-47	-100	-27	-96	-19	-21	-48
Timbre	-6	-66	-50	-100	-34	-89	-20	-32	-50
Loudness	0	-79	-67	-100	-33	-88	-18	-29	-52
Beat	-8	-67	-59	-100	-36	-99	-17	-44	-54
Global	-6	-63	-58	-100	-32	-93	-19	-38	-51
Avg	-4	-69	-56	-100	-32	-93	-19	-33	-51

Table 3.3: The worst case difference indicator, as a percentage, between the static and subjects segmentation selection.

It is interesting to note that, for the worst case, the perceptual segmentation scores

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc
Pitch	SR	SI	SF	SI	SF	SI	SSl	SI
Timbre	SR	SI	ST	SI	ST	SI	SSl	SI
Loudness	SR	SI	SF	SI	SF	SI	SSl	SI
Beat	SR	SI	SF	SI	ST	SI	SSl	SI
Global	SR	SI	SF	SI	ST	SI	SSl	SI

Table 3.4: The features used to represent each of the surface features for each song in the worst case.

severely low for the Hard Rock and Classical songs for all the surface features (close to -100%), as shown in Table 3.3. Furthermore, for both of these songs spectral irregularity (SI) is selected as the worst representative statistical feature across all surface features, as shown in Table 3.4.

Clearly, selecting the wrong statistical feature to represent a surface feature will result in poor representation. However, the extent of the improvement of the static segmentation over the perceptual segmentation in these cases is a prime case for the need in understanding the relationship between statistical features and surface features. This is further exemplified by the fact that spectral irregularity (SI) is selected as the best statistical feature to represent Electronic for pitch, as shown in Table 3.2, giving a 48% positive improvement over static segmentation in Table 3.1. Clearly the best feature to use is dynamic and is based on the styles of music and the surface features which we are trying to represent.

From Table 3.3, we can see that there is a rather highly negative difference indicator on average for the worst case from a perceptual stand point.

This result, along with the ones from our best case in Table 3.1, shows that there are indeed certain statistical features which better represent specific surface features. However, as described in the results from the experiments in Chapter 2, the surface features that are important to human perception of structure are different from one song to the next. This means that there is no way to know which surface feature would be significant to which music style and therefore which statistical feature to use for simulating human perception. As such, this means that we do not have any prior knowledge which would allow for the avoidance of these worst case scenarios. Again, this supports the notion that using human perception as a method of evaluation for segmentation is not desirable.

The Average Case

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc	Avg
SC	-2	6	-7	-5	-3	-8	1	1	-2
SI	32	-69	-8	-100	41	-93	-4	-33	-40
SF	6	9	-51	-40	-32	-1	1	-4	-14
ST	6	9	-51	-40	-32	-1	1	-4	-14
SSl	36	-61	-51	-94	49	-4	-19	-22	-25
SSp	-2	6	-7	-5	-3	-8	1	1	-2
SR	-4	5	-9	-6	-4	-7	4	2	-3
SL	8	-3	-6	-12	-2	3	-9	4	-2
SSh	7	2	-2	-9	-1	7	-8	7	0
M	5	3	-49	-38	-23	-1	5	0	-12
Avg	9	-9	-24	-35	-1	-15	-3	-5	-10

Table 3.5: The average over all surface features for each statistical feature, as a percentage, between the static and subject segmentation selection.

As can be seen from Table 3.5, the overall perceptual segmentation comes out to be 10% worse than the static segmentation. We believe that this falls within a threshold which is so close to the static segmentation that it can be simply explained by noise. Therefore, on this basis alone, we can see that there is no significant advantage to involving human perception for evaluating segmentation algorithms, at least towards the purpose of maximizing the representational quality of the feature extraction process.

Finally, it is important to note that significant and consistent improvement in the difference indicator can exist given that the segmentation is coming from a segmentation algorithm which is based on the inherent characteristics of music data. We will propose a new segmentation algorithm and test our evaluation approach using it in Chapter 4.

3.7 Summary and Discussions

In this chapter we describe a novel approach by which segmentations can be objectively evaluated without the need of either human music experts or perceptual experiments. We show that perceptual segmentation is questionable for measuring segmentation accuracy and that, with respect to our evaluations perceptual segmentation, is not a good indicator of representational quality.

We find that there is some variation in the results if we use different individual statistical features to represent the surface features of music. This lends support to the idea that some statistical features are designed towards simulating individual surface features.

However, from our results we have found the output to be consistent regardless of the statistical features used. We find that the quality of perceptual segmentation is similar to the quality of static segmentation when it comes to the purpose of creating a representation of the original music piece. It is highly probable that the number of segments used is of more importance than the segment selections themselves.

That is, if you are making segment selections which are not directly based on the characteristics of music data, such as in the case of static segmentation, then the number of segments you decide to use has greater impact than the segment positions you choose.

While there is some loose relationship between the music data and human perception on it, it appears as though that the relationship is dependent on specific features and songs and it is challenging to know it before hand. Therefore, human perceptual segmentation pseudo-approximates a static segmentation, supporting the need for a different approach to segmentation evaluation.

Note, however, that this may not be true for segmentations which are based on the characteristics of music data, as will be seen in Chapter 4.

To further this work, previous common segmentation approaches should be implemented and analyzed using our evaluation approach. In addition, experiments are under way to examine the effectiveness of our evaluation approach through high-level MIR tasks, such as classification.

Chapter 4

Segmentation Towards Facilitating Feature Extraction

4.1 Introduction

“Segmentation,”, as Ong [42] describes it, “which facilitates partitioning audio streams into short regions for further analysis, is an indispensable process”. This clearly highlights that segmentation is a key step in the MIR process, as shown in Figure 1.1.

West and Cox [57] find that using a dynamic segmentation (See Chapter 3.) increases the classification accuracy over using static segmentation. This confirms that the quality of a segmentation directly influences the quality of the feature extraction process, and thereby any further analysis. An ideal segmentation would separate the music data into highly self-similar segments such that any feature extraction process which follows would then hopefully return a maximum representation.

As Bruderer [12] puts it, segmentation is highly related to the Gestalt Rules of Proximity and Similarity (See Chapter 1.). An essential notion in its description of is how elements “appear” similar. This similarity is relative to the feature extraction process. That is, elements which appear similar according to one aggregate function may not do so according to another. Therefore, any segmentation algorithm should depend highly on the feature extraction process which it is intended to precede.

Many previous approaches to segmentation limit themselves to specific feature sets for their algorithm design. This has led to a series of results, which in essence, employ similar algorithmic techniques but just by using different feature sets [22, 31, 43, 55]. This search for the best mix of features in a segmentation process has developed around the lack of knowledge into the relationship between surface and statistical features, as outlined in Chapter 2. Without that understanding, the segmentation problem becomes combinatorial and therefore futile. Furthermore, it is often the purpose of these algorithms to construct a segmentation based on one particular feature set. It is claimed that such a segmentation can also be used for extraction purposes of other features [42]. Without an appropriate study into the relationships between the features used in the segmentation and the other features to be extracted, this claim needs further verification.

In addition, the evaluation of any segmentation algorithm is an integral part of the algorithm's goal. It is a common practice to use perceptual segmentation experiments to evaluate the quality of segmentation algorithms [30, 31, 34, 42, 55]. Previous segmentation algorithms have been designed and implemented towards simulating human perceptual segmentation. However, the relationship between perceptual segmentation and the computational representation is unexplored, and, as explained in Chapter 2, there is some evidence that the two are entirely disjunct. Note, however, that this does not imply that the psychological aspects of music should be ignored. Rather, this is exactly the purpose of the feature extraction step of the MIR process. A segmentation algorithm based on an individual feature encodes the psychoacoustic properties of that feature. By leaving the psychological aspects of music to the feature extraction process, a segmentation algorithm can focus on its singular goal of maximizing the quality of that feature's representation. This relieves a segmentation algorithm from having to deal with the complexities introduced due to simulating perceptual structure and exploring the relationship between perceptual structure and perceptual surface, as seen in Figure 1.2.

In this chapter, we attempt to design a segmentation algorithm towards this new description of desirability. That is, a segmentation algorithm which maximizes the representative quality of the feature extraction process. To this end, we have two criterion for our segmentation algorithm: it should be feature independent and feature-based. The former occurs when our algorithm can be used with any feature extraction function, while the latter is an algorithm which can use the supplied feature to perform its operations. To evaluate our algorithm we use the objective approach for evaluating segmentation algorithms outlined in Chapter 3.

This chapter is organized as follows. Section 4.2 discusses related works in the area. Section 4.3 looks at our algorithm in detail and its justifications. Section 4.4 describes any parameters and customizations which are required to reproduce our results. Section 4.5 evaluates our algorithm and provides some analysis. Finally, Section 4.6 contains a summary and some discussions.

4.2 Related Works

Most segmentation algorithms are based on the notion of finding locations of significant changes. This is achieved by first splitting the audio waveform into frames: small sections of equal distance. Frames are usually very small and as such little representative quality is lost in using them. From these frames a variety of features are extracted and a distance between adjacent frames over the features is calculated. Using these distances, it is typically determined that "good" segments are those locations which are maximal in distance between neighbouring frames.

Tzanetakis and Cook [55] develop such an algorithm. They use *Spectral Centroid*, *Spectral Flux*, *Zero Crossing Rate*, and *Root Mean Squared Energy* as features, to determine distance. They then use the derivative of the distance function to discover locations of maximal changes and use a simple heuristic to select which of these are to be used as segmentation locations. In order to evaluate their accuracy, they use a perceptual segmentation experiment and compare the distances between perceptual segmentation points against the algorithmic ones. As explained above, the limitations of this approach are characterized by its dependence on a set of specific features as well as its evaluation being based on the quality of perceptual segmentations.

Jian *et al.* [31] design a very similar algorithm to Tzanetakis and Cook's. However, Jian *et al.* attempt to develop their algorithm such that it models human perception. They use features of *Roughness* (a measure of spectral frequency distribution), *Periodicity Pitch*, and *Loudness* to characterize the psychoacoustic relationship to computational representation and develop distances. Distances are measured by two functions, the variance between neighbouring frames, and the difference in feature values between frames. A weighted aggregation of these two distances are then used to create a score for each feature independently. A ranking algorithm then considers the scores looking for the high scores among them. In essence, a maximal score among any individual quality corresponds to a segmentation point. The evaluation method used is identical to the one in [55]. However, the accuracies between the experiments cannot be compared since they are performed on different data sets. It is important to note that Jian *et al.* stumble onto the idea that each feature needed to be considered independently for segmentation. However, since their method is designed towards human perception, the selection of specific features and therefore dependence on them is critical to their design.

A slightly different approach is to develop a self-similarity matrix between frames and their neighbours. Foote [22] proposes such an approach. Foote creates a self-similarity matrix in the *Fast Fourier Transform Log Magnitude domain* [38] of each frame. He then finds a measure of self- and cross-similarity within and between neighbouring frames. The difference between the two is used to create a novelty score. The points representing the extremes of these novelty scores are selected as segmentation locations. No evaluation of accuracy is described in the work.

Peiszer *et al.* [43] develop an approach which uses the algorithm designed by Foote [22]. Peiszer *et al.* extract features *Spectrogram*, *Mel-Frequency Cepstral Coefficients (MFCCs)*, *Rhythm Patterns*, *Statistical Spectrum Descriptors* and *Constant Q Transform*. Evaluation is done against a large collection of perceptual studies from other works.

Jensen [30] also proposes a method using local self-similarity matrices. He uses features of *Spectral Flux*, *Perceptual Linear Predictor* and *Chroma* to represent psychoacoustic perceptions of *Rhythm*, *Timbre*, and *Harmony* respectively. To detect differences between neighbouring matrices a shortest path algorithm is used. Again, his results are evaluated against a selection of perceptual segmentations.

Ong [42] also uses self-similarity matrices over features of *MFCCs* and *Subband Energy* to determine candidate segmentation locations. Other features are then used to narrow down the exact locations for segmentation. Evaluation is reported by a comparison to manually labelled boundaries. The relationship between the features used to develop candidates and final locations is not explored.

West and Cox [57] propose a method of using an onset detection function which attempts to locate segments by finding locations at which the function surpasses a dynamic threshold. In essence, their method incrementally slides a window along the data and calculates the function value for that window. They test a variety of onset functions and report that the best results are returned from *Entropy*, *Spectral Centroid*, *Energy* and *Phase* based functions. They manually annotate a series of songs to create a training set. Using this as a ground truth, they then use exhaustive parameter optimization, to adjust the threshold and onset detection function window size. Evaluation is done slightly differently in the work, where they use a decision tree classification to evaluate the genre of the music samples and report the overall classification accuracy against other segmentation methods'. While this method of evaluation is more objective from a computational standpoint, as it evaluates directly the ultimate goal of the entire MIR process, it is still problematic. The most prominent of the problems is that the genre ground truth used is subjective and cannot stand alone as a model for objective evaluation. In [6], Aucouturier discusses the failures of using genre and our work [48] directly tests those failures. Similar to other works, West and Cox's work is dependent on certain features. Specifically, the *Mel-Frequency scale* and

Octave scale are used for integration of the spectral bands.

Levy and Sandler [34] describe a method by which a semi-supervised Hidden Markov model can be used to develop a segmentation algorithm. Once again the evaluation is conducted against perceptual segmentations. They point out, though, that the segmentation selection depends on the relationship between the features and the music used. Furthermore, they find that segment selections regularly occur at locations that do not accord with the perceptual ground truth.

The issues with all of these methods are in their lack of feature-independent and feature-based algorithms. That is, they are dependent on a specific set of features. Furthermore, these methods are developed towards evaluations which are not objective and, for the most part, have no bearing on the ultimate goal of segmentation. In this chapter, we attempt to develop an algorithm which is both feature-based and feature-independent. Furthermore, Our algorithm is designed towards the goal of maximising the representative quality of extracted features.

4.3 A Merge-based Greedy Segmentation Algorithm

Our segmentation algorithm is to merge the steps of feature extraction and segmentation into one. This means, the algorithm described here needs to be run separately for each feature. The output from the segmentation step includes both the segmentation positions and the extracted features.

Before presenting our algorithm, some notation is in order. For the inputs to the algorithm, we denote, by S, the music piece we are going to segment. Note that S is already in its Pulse Code Modulation (PCM) format. Let $|S|$ denote the number of data sample points in S. Since our segmentation is feature-based, we denote the feature to be used in the algorithm as F.

The initial segment length is denoted as L and has an initial value of 512 (For the reasons for this, refer to Section 4.4.). Thus, initially, the number of segments is $n = |S|/L$. The segments are maintained in a singly linked list. In each element of the list, field s points to the segment, i.e., the set of sample data points in the segment, field fv contains that vector or scalar value of feature F calculated from the data points in that segment, and field *next* points to the next element in the list. In order to traverse the list, the first element is pointed by pointer H. The output of our algorithm is a set of N segments in song S based on feature F. N is a parameter that will be further discussed in our experiments. The entire algorithm is outlined below.

A feature-based greedy segmentation algorithm

```
$P = H$;
while ($P \neq null$) { /* Calculate the feature for each segment */
    $P \rightarrow fv = F(P- > S)$
    $P = P \rightarrow next$;
}
$n = |S|/L$
while ($n \neq N$) { /* Until we only have the requested $N$ segments left */
    $P = H$;
    $d_{min}$ = a very large number;
    while ($P \rightarrow next \neq null$) { /* Find the minimum distance between neighbours */
        $d = Dist(P \rightarrow fv, P \rightarrow next \rightarrow fv)$;
        If $d < d_{min}$ {
            Let $M = P$; /* Remember $P$ as $M$ for merging later */
            $d_{min} = d$;
        }
        $P = P \rightarrow next$;
    }
    Append the data points pointed by /* Merge the min distance neighbouring segments */
        $M \rightarrow next \rightarrow s$ into the one by $M \rightarrow s$;
    Delete $M \rightarrow next$ by setting
```

```
            $M \to next = M \to next \to next$;
        Recalculate $M \to fv$;
        $n = n - 1$;
    }
```

Simply put, our algorithm begins by separating the data into a large number of small frames and extracting a feature over each. We then iteratively find the two neighbouring frames which are most similar in their feature representation and merge them, extracting a new feature for that frame. We do this process until the number of frames is reduced to the number of requested segments. The end results is N segments, each of which is represented by a feature vector.

It can be seen that our algorithm is straightforward and easy to implement, which lets us focus more on its evaluation. To do this, the naive way would be to implement the algorithms described in Section 4.2 for comparison. However, doing so would open ourselves to criticism, such as biased implementation, biased data set selection, etc.

To this end, we evaluate our segmentation algorithm against the evaluation approach described in Chapter 3. Recall, from that chapter, that we compare the perceptual segmentation from Chapter 2 with static segmentation and find that the former bears no significant improvement over the latter. For this reason, our algorithm needs only to show some significant improvement in order to surpass perceptual segmentation.

4.4 Experiment Setup

The only parameter which is needed for the setup of this algorithm is the number of resulting segments, described as N in the previous section. Because of this we run our experiment for all numbers of segments between six (6) and fifty (50), resulting in 45 trials. We have chosen this range of numbers because it represents the general range of the numbers of segments selected by subjects during the perceptual study. *C++ Audio and Music Extraction Library* (*CAMEL* for short.) is used as a framework for the development of our algorithm and is further explained in Appendix B.

The data set used was a selection of eight (8) songs from the MIREX genre classification data set. Each song was selected as a representative of a completely different genre and style. All the music is freely available online [19]. The songs used are reported in Table 2.4. Note that our work here is not intended to make any statement, or give any insight, into the specific genres. Rather we use genre as a method of showing that our data is representative of variable styles of music.

The features extracted are *Spectral Centroid*(*SC*), *Spectral Irregularity J* (*SI*), *Spectral Flatness* (*SF*), *Spectral Tonality* (*ST*), *Spectral Slope* (*SSl*), *Spectral Spread* (*SSp*), *Spectral Rolloff* (*SR*), *Spectral Loudness* (*SL*), *Spectral Sharpness* (*SSh*), and *Mel-frequency Cepstral Coefficients* (*M*). It is important to note that we also experimented with several basic statical features as well, including *Standard Deviation*, *Zero Crossing Rate*, and *Root Mean Squared Energy*. We will discuss more on them in Section 4.5. For convenience, the features used are referred to by their identifiers.

Our evaluation approach is based on the notion that any dynamic segmentation algorithm should at least return a better representation than a static segmentation algorithm, as discussed in Chapter 3. That is, if we are to simply use segments of equal length to divide a song, such that we have the same number of segments as the output of our dynamic segmentation, the dynamic segmentation should maintain, in general, a smaller distance (as shown by the distance indicator in Chapter 3) from the original representation.

We report two sets of results in the following order: (1) The results showing how often our segmentation is better than the static segmentation throughout the 45 trials in Table 4.1; and (2) The difference indicators showing how much better or worse our segmentation performs than the static one, averaged over the 45 trials for each feature and each song in Table 4.2. We discuss these results below.

4.5 Results and Analysis

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc	Avg
SC	2	29	96	100	100	87	91	100	76
SI	100	100	100	100	100	100	100	100	100
SF	0	7	100	100	100	58	100	100	71
ST	0	16	100	100	100	56	100	100	72
SSl	100	100	100	100	100	89	100	100	99
SSp	2	29	96	100	100	87	93	100	76
SR	22	2	100	100	100	7	0	100	54
SL	64	7	100	100	98	78	100	100	81
SSh	38	4	100	100	53	100	100	100	74
M	64	44	100	100	100	100	100	100	89
Avg	39	33	99	100	95	76	88	100	79

Table 4.1: The percentage of times, over 45 trials, that our greedy segmentation performs better than static segmentation.

The cross cell between Classical with SSp in Table 4.1 shows us that, over the 45 trials, our greedy segmentation algorithm performs better 87% of them. Table 4.2 reports accuracies over several trials, however, the averaged values making up these accuracies can be misleading. Therefore, Table 4.1 is useful as a measure of consistency in our algorithms performance.

From Table 4.1, our greedy merging algorithm is better than the static segmentation 79% of the time. Note that the results over a song are generally more consistent than over a feature. That is, if one feature scores low on a song, then most features score low on it. This shows that there exist certain situations in which our greedy approach makes bad decisions. A better heuristic could lead us to a solution around these decisions. However, even for those songs which return generally low results, some features score well, implying that there are certain features, for example, SSl, for which our algorithm is well suited.

	Jaz	Pop	Eth	HaR	Ele	Cla	Pun	Roc	Avg
SC	-1	-7	8	8	5	3	9	58	10
SI	98	82	92	97	61	89	100	92	89
SF	-5	-4	63	73	35	17	26	49	32
ST	-5	-4	63	73	35	5	26	49	30
SSl	73	70	81	86	46	34	94	84	71
SSp	-1	-7	8	8	4	3	14	6	4
SR	0	-1	8	6	3	-3	-2	3	2
SL	1	-4	15	21	5	64	10	10	15
SSh	-1	-7	12	20	1	10	9	7	6
M	-1	-1	68	64	35	12	24	41	30
Avg	2	1	42	45	23	17	29	40	25

Table 4.2: The percentage of error from static segmentation (difference indicator) that out greedy merge based segmentation selection achieved for each feature against each song.

In Table 4.2 we report the average difference indicator between our algorithm and static segmentation over the 45 trials. For example, looking at the intersection between the Ethnic column and the SSl row shows us an averaged difference measure of 81%. This result indicates that for this particular song-feature pair the algorithm performs quite well.

On average, according to Table 4.2, our segmentation performs 25% better than the static one. This performance is measured in terms of the difference indicator described in Chapter 3. It is interesting to note that our algorithm never returns highly negative results (close to -100 in terms of percentage). The results are either very positive, slightly positive or slightly negative. From a graphical standpoint, we find either our segmentation does significantly better or it tends to follow the same pattern as the static one. Examples of these cases can be seen in Figure 4.1 and Figure 4.2 respectively. This behaviour indicates that there is indeed a potential for selecting better quality segmentation points and that in the majority of cases a greedy selection is, in the worst case, in the same relative range as

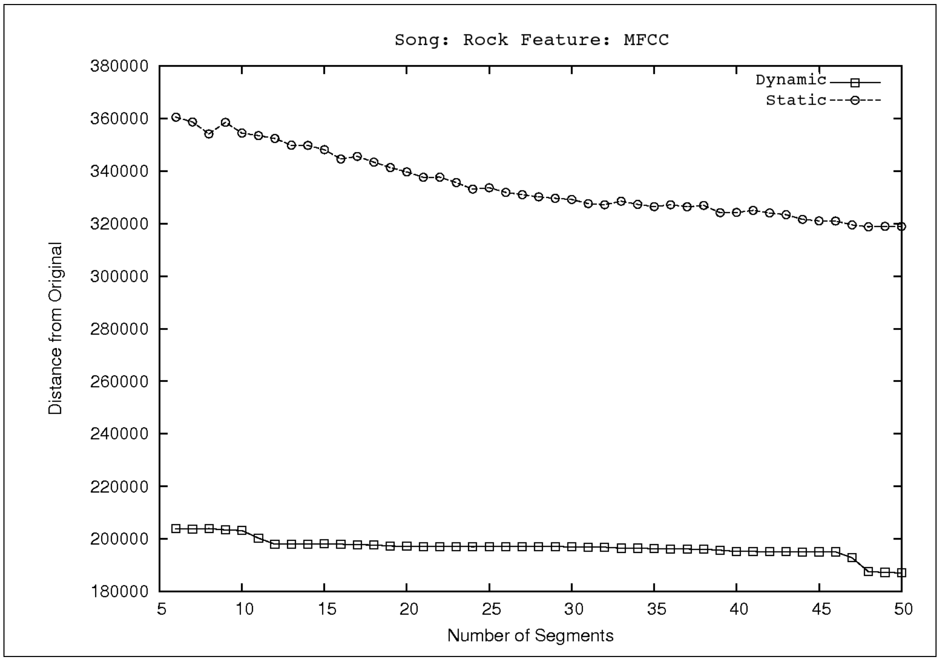


Figure 4.1: The results of the Rock song for MFCC's. Note the positive difference indicator in performance by the greedy segmentation.

using static segmentation.

We also have run the experiment on several time-domain statistical features, such as *standard deviation*. The tendency for those features replicates the same representative quality as the static segmentation as seen in Figure 4.3. For this reason we do not report those results here as they do not provide any interesting numeric information and have little effect on our overall evaluation.

4.6 Summary and Discussions

In this chapter, we have introduced a new approach to segmentation. This approach is based on a different goal than previous works in MIR: to maximize representation quality and not to mimic human perception.

Our segmentation algorithm described is a simple, greedy merge-based algorithm. The

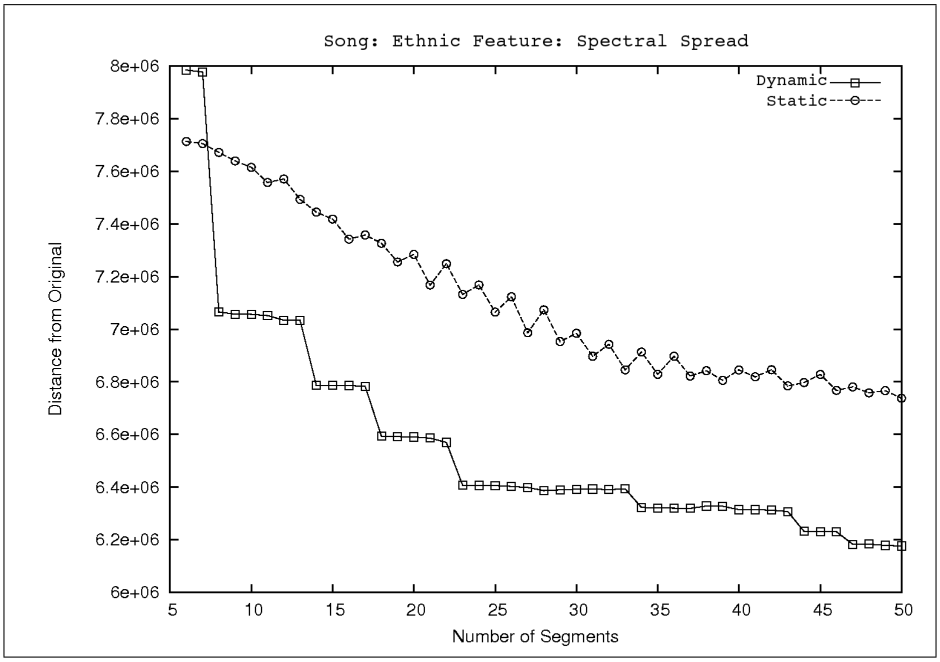


Figure 4.2: The results of the Ethnic song for Spectral Spread. Note the tendency of the dynamic selection to mimic closely the static segmentation.

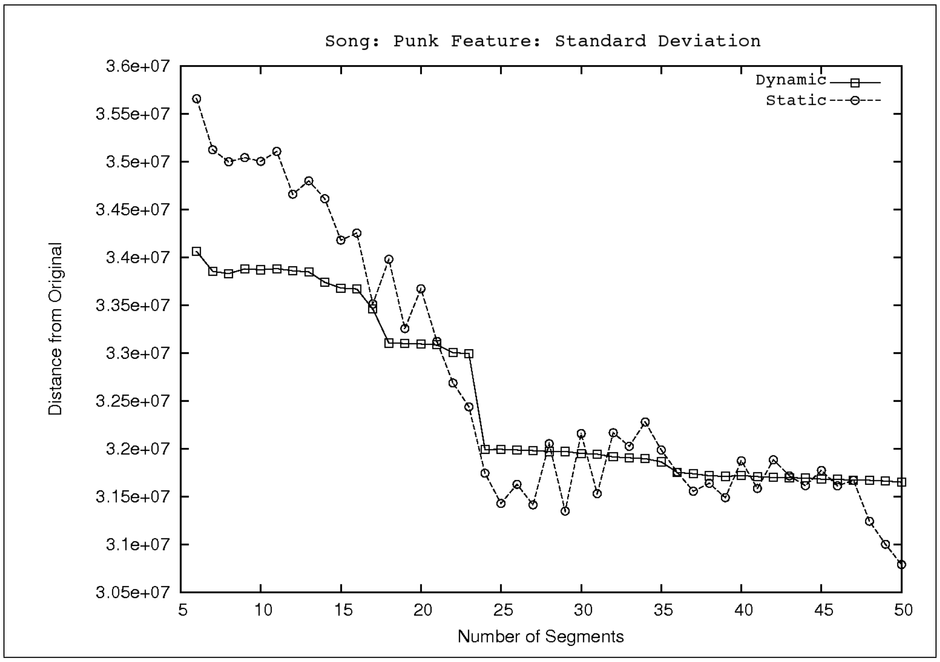


Figure 4.3: The results of Punk song for Standard Deviation. Note the tendency of the dynamic selection to follow the static segmentation.

experimental results show it performs, in general, significantly better than its static counterpart. This level of improvement is significant enough to allow us to conclude that it is a better result than the perceptual segmentations developed in Chapter 2 and evaluated in Chapter 3.

Having said that, this result does not actually imply that our algorithm is in fact better than those outlined in Section 4.2. Until those algorithms are specifically experimented against our evaluation approach from Chapter 3, we cannot make any conclusive comparisons.

It is important to note that our algorithm is easily modified to make more intelligent decisions by simply replacing the greedy selection with the one that uses a better heuristic (per Lines (12) - (15) in the algorithm). Furthermore the simplicity is one of the merits of our algorithm.

We also find that for time-domain statistical features the tendency is to closely model the behaviour of the static segmentation. We attribute this to one of the two following situations. (1) The greedy heuristic somehow consistently makes poor choices whenever it comes to time-domain statistical features; and, alternatively, (2) it seems more likely that time-domain statistical features are less reliant on the position of segments as much as the number of segments. If the latter situation is affirmative, then selection of "good" segmentation positions for these features may as well be based on the static segmentation. Further investigations into this result could be interesting and necessary.

A secondary result to the findings in this chapter is that using a data-based segmentation algorithm can in fact return a consistent and significant improvement against the evaluation approach described in Chapter 3. As such, this should be the motivation for other novel algorithms to be designed and tested.

Chapter 5

Conclusion

5.1 Summary

In Chapter 2, we describe a series of human perceptual experiments in which we attempt to discover the relationship between what is defined as the perceptual surface and perceptual structure of music. The discovery of this relationship is essential for the continued use of human perception for evaluation of segmentation algorithms. From this experiment we find many interesting results about human perceptual segmentation. However, we have found that the relationship between perceptual surface and structure is variable, dependent on particular musical styles. For this reason, we conclude from Chapter 2 that a new approach for segmentation evaluation would be desirable.

In Chapter 3 we propose a novel approach to the evaluation of segmentation algorithms. Most previous methods of segmentation evaluation have been based on either human experts or perceptual experiments. However our approach does not rely on either. Our approach is based around maximizing the representational quality of specific features which are meant to be extracted from the music itself. To test the validity of our approach, we apply it directly to the results of our perceptual experiment from Chapter 2. From this comparison we find some support for our claims that human perception has little or no relation towards improving the quality of features extracted. However, we acknowledge that by redefining the evaluation for segmentation algorithms, we have inherently redefined their goal as well. For this reason we conclude from Chapter 3 that new approaches to segmentation are necessary, as well as the need for a re-evaluation of the previous methods.

In Chapter 4, we propose a novel segmentation algorithm and evaluate it against our approach outlined in Chapter 3. Our approach is a greedy merge-based algorithm which shows significant improvements over the static segmentation. These improvements support the claim that intelligent segmentation algorithms do exist and provide higher quality feature representation. Our algorithm is easily extendible towards using more intelligent selection heuristics which could increase the improvements against the evaluation approach.

In summary, we attempt to take segmentation of music, a key step in the MIR process, in a new direction. We show through experiments that the base assumption of traditional segmentation algorithms, that is segmentation is a method of simulating human perception of structure, detracts from the quality of the sequential MIR process. For this reason, we propose that segmentation should be evaluated by its ability to maximize the representational quality of the feature extraction process. In doing so, we have redesigned the ultimate goals of segmentation algorithms. This goal has allowed us to develop a new, objective evaluation approach against which segmentation algorithms can be held. Furthermore we have designed a simple, yet effective, segmentation algorithm which is focused towards our new definition of "optimality".

5.2 Limitations and Future Work

There are several limitations to the work presented in this thesis. The first is the limited size of the data set used in the perceptual experiments. While the number of songs used in Chapter 2 is within the same range of several other similar works', it is by no means statistically representative of all music. We attempt to increase the representational quality of our work in Chapter 2 by selecting music from a variety of styles. However, it is possible that, with greater amounts of music, we may have been able to identify some patterns which would change our conclusions.

A second limitation is the use of the same minimal data set for evaluation of our segmentation algorithm in Chapter 4. It would be a more significant result to show how our algorithm is evaluated against a larger corpus of music. But due to time constraints, we are unable to do so. However, using only this limited set of music allowed us to compare our results more directly against the perceptual experiment, showing that a segmentation selection based on the data has significant improvements over selections which are naive or, as in the case of the perceptual segments, are relatively naive to that data.

A third identified limitation pertains to the testing of the evaluation approach outlined in Chapter 3. Though we are currently conducting more experiments, it would be beneficial to include results which show a direct correlation between increased accuracy in the evaluation approach and increased accuracy in various high-level MIR tasks, such as classification. At this time, preliminary results towards this end are supportive.

Perhaps the most obvious limitation of this thesis is the lack of perceptual representation of structure in the development of segmentation algorithms. The results in Chapter 2 show that the assumption, that a direct relationship exists between perceptual surface and structure, is unlikely. However it does not disprove anything. It is possible that simulating perceptual structure in conjunction with perceptual surface may create a representation which is superior for many high-level MIR tasks. For the area of MIR, the answer to this possibility is fundamentally important.

A final limitation to be addressed here is the notion of perception versus cognition. It is possible that our ideas of perceptual surface and perceptual structure may actually be linked to the separation between human perception of sound and human cognition of music. This distinction is of high importance to developing better models of representation for MIR systems and much work in this area is still needed.

Much of the necessary future work which follows this thesis is to address the limitations which have been identified above. Certainly the results from this work are incentive enough to promote further investigation into the methodology by which segmentation algorithms are developed and analyzed.

In terms of extending the results in Chapter 2, more work needs to be done in testing the relationship between perceptual surface and structure. If a generalized mapping can be found between them, it would be of great benefit to the MIR community. With respect to our novel evaluation approach outlined in Chapter 3, further work is required into the justification of its use. While our results seem to suggest that current evaluation approaches are subjective and limited, we do not claim that ours is in any way optimal and we fully recommend further attempts towards developing better ones. A great deal of future work needs to be done with respect to segmentation algorithms, such as the one we propose in Chapter 4. Specifically, a comprehensive survey and comparison of current methodologies would be highly benificial. In terms of our own algorithm, it would be of great interest to replace the greedy selection heuristic with one which is more intelligent.

Appendix A

Perceptual Segmentation Software

A.1 Introduction

In this appendix, we describe the software developed in order to assist in the perceptual segmentation experiment conducted in Chapter 2. In essence, we have a selection of music data on which we want subjects to select locations of significant change according to specific psychoacoustic features. For example, given a song, we might ask a subject to mark every location they believe the beat of the song to change in some significant manner. Therefore, for each song which is presented to the subject, they are also presented with instructions outlining which psychoacoustic feature on which they are to make their segmentation selections.

A.2 Design

The first principle of designing software for perceptual experiments it to make it as simple as possible for a subject to use. This limits the amount of distractions and extraneous stimuli that might affect subjects. For this reason our software is made with every attempt towards a minimal user interface. Because the experiment we conduct is based on audio perception, no other auditory stimuli is presented to the subject at any time, other than the music on which is being tested. Visual feed backs are also kept to a minimum and only provided for reasons of intuitive user control. No extraneous information is presented to the subject, such as visualization of the music waveform or other such stimuli in most media players, which may have unforeseen effects on subjects.

Our software is developed using Sun Java SE in JDK version 6 [52]. To assist our work with audio files, we use JLayer version 1.0 [29], which is a Java library that decodes, converts, and plays MP3's. It is released under the GNU Lesser General Public Licence (LGPL) and as such is free for use and revisions.

Our perceptual segmentation is developed in terms of a general Model-View-Controller design, where the front-end graphical user interface (GUI) is separated from its back-end logic and data. Two GUI's are developed, one for logging in and creation of subject user and the other for the perceptual software itself. Screenshots of each of the GUI's can be seen below in Figures A.1 and A.2, respectively. For each component in the GUI (See below.), such as a button or a text box, a listener object in the Controller is designed to take appropriate actions when activated. Activation takes place by the objects themselves at the View level. The Model layer handles the loading and playback of music, as well as the creation, and the edition of segments and user accounts.

The software is written to be dynamic to the content of a local folder structure. In this folder structure, there exist three separate subfolders. The first manages the music data and any music files which are added to this directory will be automatically added to the list of music being evaluated in our experiments. Similarly, the second folder manages the instruction files, which are simple text files outlining the psychological features and their associated instructions. Finally the third directory manages the subjects' output segments, storing the appropriate information for each song/instruction pair as created by subject. User account information is also stored in this third directory.

A.3 Functionality

The functionality of our software is outlined in this section in reference to the numerically highlighted screenshots.

For our user account management section of the software, we only offer the ability to

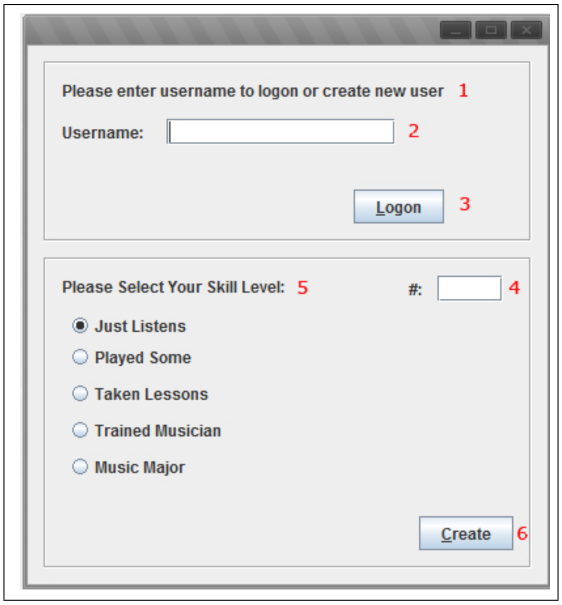


Figure A.1: A screenshot of the logon and user account GUI for our perceptual segmentation software with numerically highlighted functionality.

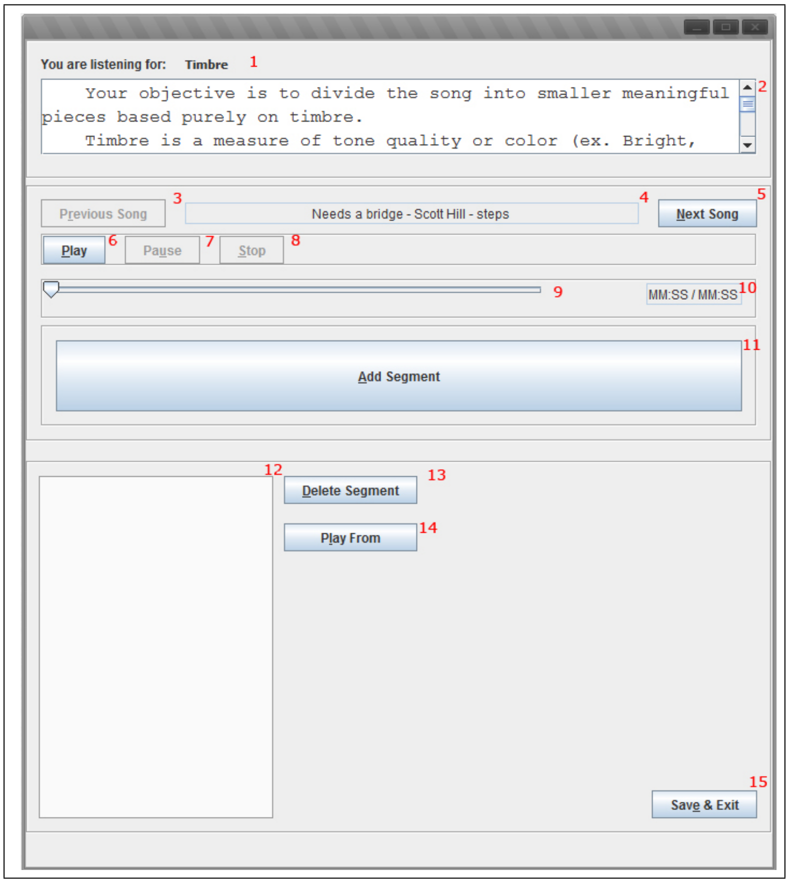
create new subjects, or for subjects to logon on to previously created accounts. Both of these functions are provided via the single GUI demonstrated by Figure A.1.

For the purposes of logging on, a subject is only expected to enter a user name, into the text field (9), which she/he has previously created, and hit the Logon Button (3). If any errors occur, such as an unknown user name is entered, then the instruction label (1) will change accordingly.

For a first-time subject, she/he is required to enter a new user name, in the text field (9), to select a skill level, from the radio button (5), enter a user number (4), and hit the Create button (6). Again if any error occurs, such as a pre-existing user name being entered, then the instruction label (1) will change accordingly.

Note that the user number is given to the subject when s/he signs up for the study and has two purposes: to maintain an ethical separation between the subject's activity and her/his identity and to select the order in which the music and instructions are presented to her/him. That is, since the experiment follows a within-subject design as described in Chapter 2, each subject is presented with a different, pseudo-random set of music and instructions. We use pseudo-random here because we hope to obtain a somewhat uniform distribution of results over the various combinations of music and perceptual features paired together. For a further definition of the various skill levels, as presented to subjects,see Table 2.1 in Chapter 2.

The main window of our software, as seen in Figure A.2, is designed to present the subject with all of the information which she/he requires for the testing. There are three main sections to the window, each with its own unified purpose and functionality.



You are listening for: Timbre

```
    Your objective is to divide the song into smaller meaningful
pieces based purely on timbre.
    Timbre is a measure of tone quality or color (ex. Bright,
```

Needs a bridge - Scott Hill - steps

MM:SS / MM:SS

Figure A.2: A screenshot of the main player and segmenter GUI for our perceptual segmentation software with numerically highlighted functionality.

The first section, as seen at the top of Figure A.2, is the instructions section. As explained above, we test subjects on one of five different surface features. Each time the subject selects a new song to listen to she/he is presented with a pseudo-random set of instructions to accompany that piece of music in this section. Reading these instructions gives the subject a pause between each song, which should help to counter-act the carry-over effects as suggested in [15]. The instruction text, seen in text field (2), informs the subject on details of the current action. The instructions include a definition and provide a list of common descriptors for the feature s/he is listening for, as per Bruderer's work [12].

The second section in the middle of Figure A.2 is the basic music player interface. The songs are presented to the subject in a pseudo-random order such that they only are presented with each song once and each subject listens to all the songs in the experiment. However the order in which each subject is presented with the songs is completely random. This prevents any bias in our results which might be caused by subjects listening to the music in a specific order. While this section of the interface is far more complex than the rest, because of the popularity of digital music players, the controls are familiar enough with the majority of subjects such that their existence should not affect their abilities. The basic controls provided are the Previous Song Button (3), Next Song Button (5), Play Button (6), Pause Button (7), Stop Button (8) and the Position Slider (9). In addition to these buttons, we also present the subject with the Current Song Information in text field (4) and the Current Position Time Readout in text field (10).

The third section at the bottom portion in Figure A.2 is the segmentation controls. Because these controls are most foreign to the subjects, we have made them sparsely positioned and much more clearly defined. The Add Segment Button (11) will add the current position of the slider to the History list (12) in milliseconds. This tells us the rough position of where the subject believes the segment to belong to. Due to subject's reaction time, a variance of around 1-3 seconds is expected/allowed for positional accuracy, as described in Chapter 2. Because of this variance we do not need to compensate for any delay in registering time in our software. The Segment History List (12) shows a list of segments (sorted) that the subject has selected for this song/instruction pair. A subject can select the items here for deletion (13) or to use them as a start position (14). Starting from a segment position means that we move the slider (9) to that position and continue playback from there.

Finally we provide a Save & Exit Button (15) to give the subject a chance to save the currently open song/instruction segment information when s/he exits the software.

Appendix B

CAMEL: Content-based Audio and Music Extraction Library

B.1 Introduction

In order to assist in our experiments in Chapters 3 and 4 we need a lightweight, easy to use, and flexible set of software tools for content-based audio and music analysis. Specifically, we need a software environment in which we could quickly develop and test various pieces of custom code. For this reason we develop a low-level MIR framework. In this appendix, we introduce a framework of *Content-based Audio and Music Extraction Library* (*CAMEL* for short)[1], which is our own implementation of a collection of feature extraction and segmentation algorithms with a focus towards rapid implementation and experimentation of audio analysis algorithms. It is designed to be heavily modular, making the addition of new features or segmentation algorithms extremely easy.

B.2 Related Works

There are several other pre-existing MIR frameworks available [3, 13, 20, 40, 56]. However, as they are typically developed to accommodate specific works and then later added upon, these frameworks tend to be highly complex with minimal documented or community support. For this reason, making customizations within them is highly challenging. Often these previous frameworks tend to sacrifice simplicity in exchange for an extensive breadth of capabilities. These capabilities create a greater amount of overhead and call for a need for a great depth of domain knowledge into their design before they can be used to

[1] The appendix is a brief summary of our work in [47]

fulfil even simple research needs. For these reasons, we find it to be more effective, for our purposes, to implement our own framework, on which we could assure ourselves of the productivity and quality of our research.

B.3 Design Considerations

In terms of developing a framework which would best facilitate the specific needs of our work, several key considerations are in order.

1. Easy to use - Our primary design concern is simplicity. We design CAMEL to be easily understood by any users, thus allowing for rapid development despite a users background. As described by Futrelle *et al.* [25], MIR is a heavily interdisciplinary area, and therefore needs tools which are developed for use without a specific set of domain knowledge.

2. Extendible - It is important to us that CAMEL be easily extendible. This allows for quick development and testing of various custom functionality. For example, the evaluation approach outlined in Chapter 3 is designed using CAMEL as a basis. To allow for extendability, CAMEL is written in C++ as a set of modular objects, making it easy to add or replace an existing functionality with new ones.

3. Lightweight and Portable - To maintain the simplicity of working with CAMEL and to allow it to be used on an individual component basis, the amount of extraneous code and external dependencies is minimized. The only external dependency CAMEL has is the FFTW [23] library for calculating the frequency domain of the audio signals in an efficient manner.

B.4 Implementation

In order to provide simple, accessible, content-based MIR functionality, CAMEL is designed around a series of modularized objects. The flow of work for these objects can be seen in Figure B.1. Each individual object is further described in this section.

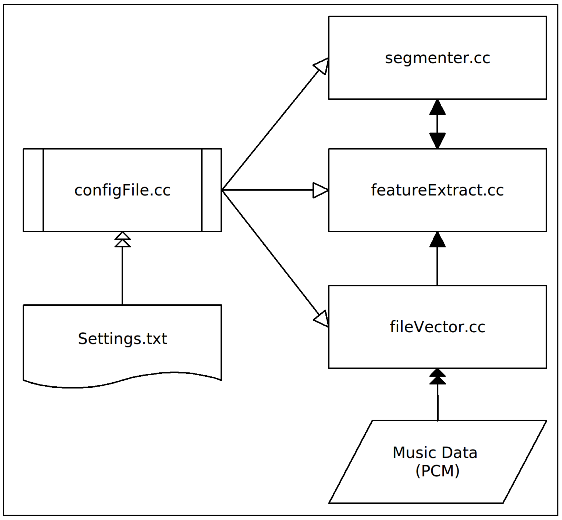


Figure B.1: The flow diagram of CAMEL.

B.4.1 *fileVector*

The fileVector object is a template class developed to associate itself with a pulse-code modulation (PCM) format audio file. The values of the PCM audio file are then added into a vector for efficient access by higher level objects. The resulting vector is offered as a public accessor back to the instantiating object.

B.4.2 featureExtract

The bulk of the work done in CAMEL happens at the level of the featureExtract object. When instantiating a featureExtract object, a user only needs to set the file name of a PCM formatted audio file, the requested feature for extraction, and a start and end position within the audio from which to return the value. Once these parameters have been set, the complexities of the feature extraction process are entirely handled by the featureExtract object, as shown in Figure B.2. The featureExtract object is dependent on the existence of the fileVector object.

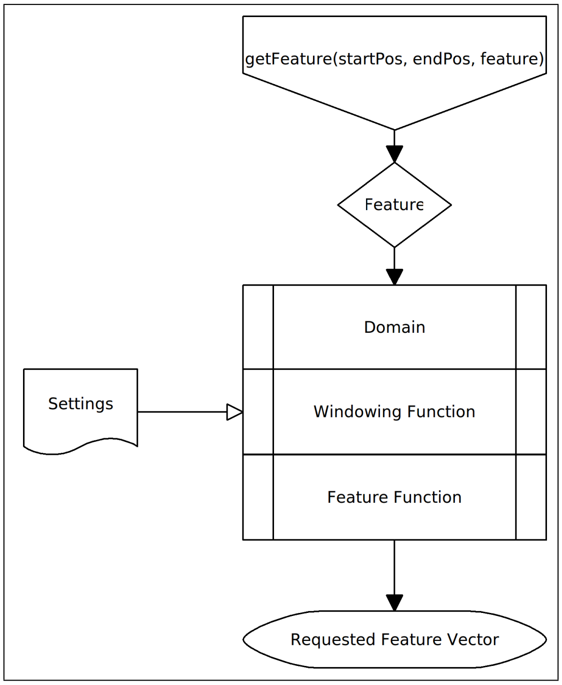


Figure B.2: A flow diagram the featureExtract object in CAMEL.

Dependent on the requested feature, the featureExtract object selects the appropriate domain function for that feature. Each domains provides a different set of data on which features can be extracted, essentially each is a transformation of the original data by some function.

There are several different domains in which a feature extraction function might work.

Many feature extraction functions are designed to work in a specific domain. However particular ones, such as statistical functions, can be applied to multiple different domains. In CAMEL we implement the *Time Domain*, *Frequency Domain*, *Peak Domain*, and *Harmonic Domain*. The Time Domain is simply the original PCM values which represent the amplitude value of the signal sampled at a specific rate (for example 44100.0 Hz) over time. The Frequency Domain is a Fourier Transformation over the Time Domain, giving us a representation of magnitude versus frequency values. In CAMEL the Fourier transform is provided through the use of the FFTW Library [23]. The Peak Domain is an evaluation of the Frequency Domain (actually of a spectrum of the Frequency Domain, as to be explained below), where only values of frequency peaks which surpass a threshold are kept, with some transformation according to neighbouring values. Finally, the Harmonic Domain keeps only values of the Peak Domain which are harmonics: whole number multiples (within some threshold of tolerance) of the fundamental frequency of the Time Domain.

Once the frequency domain has been calculated via a Fourier Transform, it can be converted into one of several spectrums via spectral functions. We implement four such functions: the *Magnitude Spectrum*, *Log Magnitude Spectrum*, *Power Spectrum* and *Log Power Spectrum*. Again this selection is handled within the featureExtract object, though it can be customised through a settings file, as described below.

When a start and end position are set within the featureExtract object, the values between them are separated into groups called *windows*. It is from these windows that the feature will ultimately be extracted. Once a window of values has been collected from the fileVector object and they have been transformed by the appropriate domain function, we then apply a *windowing function*. Windowing functions weigh the importance of values at different locations within the window by different amounts. We implement eight (8) such functions: *Rectangular Window*, *Hann Window*, *Hamming Window*, *Bartlett Window*, *Triangular Window*, *Bartlett-Hann Window*, *Blackman Window*, and *Blackman-Harris Win-*

dow.

Once we have a window of values in our specific domain, including any spectrum and window functions that are to be applied, we can then calculate the specific feature function on that window. In CAMEL we implement 32 such functions. Each of theses functions has been tested for correctness against the output of several other frameworks'. Note that for the more common functions we do not provide any explanation here and for the more complex functions we cite further reading.

Several simple statistical features are implemented in CAMEL, including: *Mean*, *Variance*, *Standard Deviation*, *Average Deviation*, *Skewness* and *Kurtosis*. Further to these statistical features, we implement *Zero Crossing Rate* (ZCR) which represents the frequency at which the signal crosses over from positive to negative or vice versa. ZCR has been associated with several different aspects of music including the dominant frequency [16]. Another statistical feature available in CAMEL is the *Root Mean Squared Energy*, which has been attributed as a good indication of loudness as well as a good statistic on which to conduct high-level MIR tasks such as segmentation [16]. *Non-Zero Count* is a statistical feature implemented in CAMEL which is a simple measure of audio content versus silence. Finally, for statistical features, CAMEL implements *Fundamental Frequency*, which has been attributed to pitch detection in music. For a detailed description of pitch detection and the implementation of *Fundamental Frequency* see [26].

In terms of spectral features, CAMEL implements several different features which are basic descriptions of the spectral shape or distribution of the audio. These spectral features include: *Spectral Centroid*, *Spectral Variance*, *Spectral Standard Deviation*, *Spectral Average Deviation*, *Spectral Skewness*, *Spectral Kurtosis*, *Spectral Irregularity K*, *Spectral Irregularity J*, *Spectral Flatness*, *Spectral Tonality*, *Spectral Min*, *Spectral Max*, *Spectral Crest*, *Spectral Slope*, *Spectral Spread* and *Spectral Rolloff*. In addition to these more basic spectral features we also implement several more challenging features. Spectral Loudness and *Spectral Sharpness* for instance are calculated over the *Bark Bands* and is a measure of brightness and noisiness respectively. *Mel Frequency Ceptral Coefficients* (MFCCs) are one of the most popular features used in MIR. The idea behind MFCC's is to bin frequencies into groups based on the Mel Scale. The Mel Scale attempts to simulates the changes in human perception of sound as frequency changes. A good discussion on MFCCs can found in [35] and our implementation is based on the description in [21]. *Bark Bands* is another highly popular feature function that separates the audio source into a series of bands which correspond to the psychological bands of hearing. The Bark scale is proposed by Zwicker [58].

For the Peak Domain the only feature which CAMEL includes at this time is *Peak Inharmonicity*, which is a measurement of the types of tones within the audio. As for the Harmonic Domain, we implement the *Harmonic Odd Even Ratio*, which is essentially a fraction between the two types of harmonics.

B.4.3 segmenter

The segmenter object provides a simple interface to gain a representation of an audio file. Simply by providing the file name of a PCM formatted audio file, the features which you wish to extract, and a segmentation method to use, the segmenter object will return a representation of the audio file in those features for each of the calculated segments. For segmentation functions in CAMEL, we implement two basic algorithms. The first is *static segmentation*, which divides the audio source up into segments of equal length. The second algorithm is described in detail in Chapter 4. Both algorithms take, as a parameter, the number of segments which the user wants in return. Any settings required in order for the operation of the segmentation object can be configured in the settings file as described below.

B.4.4 configFile

If a user wishes to select a custom domain or windowing function to be used or to set any of the multitude of variable parameters which are associated with many of the feature extraction or segmentation functions, she/he can do so in a settings file. However, if the user does not have the specific knowledge required to understand such settings, the settings are set to their popular defaulted values. The configFile class reads in these settings from the settings file and provides them to the other objects when they are necessary.

B.5 Summary

In this appendix, we have introduced our content-based MIR framework CAMEL for feature extraction and segmentation. Unlike many of the other frameworks in the area, CAMEL is designed around simplicity of use and extendability. It is lightweight and portable and includes a number of the current popular functions.

Future versions of CAMEL will include separating the various functionalities of the current featureExtract object into their own objects. Also we hope to add implementations of several more key features in the MIR area such that we can enhance the uniqueness of CAMEL. Improvements to the runtime of several of the features is also currently under way. Another aspect of CAMEL we hope to include in later versions is its ability to use other forms of aggregation, beyond average, over the windows. Finally, we plan to implement several other segmentation algorithms and add them to CAMEL in the near future.

Bibliography

[1] S. Abdallah, K. Noland, M. Sandler, M. Casey, and C. Rhodes. Theory and evaluation of a baysian music structure extractor. In *Proceedings of the International Symposium on Music Information Retrieval*, 2005.

[2] M. Alonso, B. David, and G. Richard. Tempo and beat estimation of musical signals. In *Proceedings of the International Symposium on Music Information Retrieval*, 2004.

[3] X. Amatriain, P. Arumi, and D. Garcia. Clam: A framework for efficient and rapid development of cross-platform audio applications. In *Proceedings of the ACM International Conference on Multimedia*, volume 14, pages 951–954. ACM, 2006.

[4] Apple. Apple itunes, May 2010. `http://www.apple.com/itunes/`.

[5] J. Aucouturier and F. Patchet. Scaling up music playlist generation. In *Proceedings of the International Conference on Multimedia Expo*, 2002.

[6] J. J. Aucouturier and F. Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.

[7] J. J. Aucouturier and F. Patchet. Tools and architecture for the evaluation of similarity measures: Case study of timbre similarity. In *Proceedings of the International Symposium on Music Information Retrieval*, 2004.

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

[9] C. R. Befus, C. Sanden, and J. Zhang. Psychoacoustic feature based perceptual segmentation. In *In Proceedings of the International Computer Music Conference*, 2010.

[10] J. Bharucha and K. Stoeckig. Reaction time and musical expectancy: Priming of chords. *Journal Exp. Psychology Human Perception Performance*, 1986.

[11] E. Bigand and B. Poulin-Charronnat. Are we experienced listeners? a review of the musical capacities that do not depend on formal musical training. *Cognition*, 2006.

[12] M. J. Bruderer. *Perception and Modelling of Segment Boundaries In Popular Music*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, Nederlands, 2008.

[13] J. Bullock. Libxtract: a lightweight library for audio feature extraction. In *Proceedings of the International Computer Music Conference*, 2007.

[14] H. Chen and L. Chen. A music recommendation system based on data grouping and user interests. In *Proceedings of the International Conference on Information and Knowledge Management*, 2001.

[15] P. Cook, editor. *Music, Cognition, and Computerized Sound.* MIT Press, 2001.

[16] M. Cord. *Machine Learning Techniques for Multimedia.* Case Studies on Organization and Retrieval. Springer, 2008.

[17] I. Deliège and A. El Ahmadi. Mechanisms of cue extraction in musical groupings: A study of perception on sequenza vi for viola solo by l. berio. *Psychology of Music*, 1990.

[18] J. S. Downie. Establishing music information retrieval (mir) and digital library (mdl) evaluation frameworks: Preliminary foundations and infrastructures. In *The MIR/MDL Evaluation Project White Paper Collection*. Workshop on the Evaluation of Music Information Retrieval Systems, 2003.

[19] J. S. Downie. The music information retrieval evaluation exchange (mirex). *D-Lib Magazine*, 12(12), 2006.

[20] A. Ehmann, X. Hu, and J. S. Downie. Music-to-knowledge (m2k): A prototyping and evaluation environment for music digital library research. *Proceedings of the Joint Conference on Digital Libraries (JCDL)*, page 376, 2005.

[21] ETSI. Speech processing, transmission and quality aspects (stq); distributed speech recognition; front-end feature extraction algorithm; compression algorithms, 2000. ETSI standard document ES 201 108.

[22] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of International Conference on Multimedia and Expo*, pages 452–455, 2000.

[23] M. Frigo and S. G. Johnson. The design and implementation of fftw3. In *In Proceedings of IEEE*, 2005.

[24] J. Futrelle. Three criteria for the evaluation of music information retrieval techniques against collections of musical material. In *The MIR/MDL Evaluation Project White Paper Collection*. Workshop on the Evaluation of Music Information Retrieval Systems, 2003.

[25] J. Futrelle and J. S. Downie. Interdisciplinary communities and research issues in music information retrieval. In *Proceedings of the International Symposium on Music Information Retrieval*, 2002.

[26] D. Gerhard. Pitch extraction and fundamental frequency: History and current techniques, 2003. technical report, Dept. of Computer Science, University of Regina.

[27] J. Han and M. Kamber. *Data Mining: Concepts and Techniques.* Morgan Kaufmann, first edition, 2000.

[28] ISO. Information technology - multimedia content description interface - part 4: Audio. *ISO/IEC 15938 - 4:2002. International Organization for Standardization*, 2002.

[29] JavaZoom. Jlayer, December 2009. `http://www.javazoom.net/javalayer/javalayer.html`.

[30] K. Jensen. Multiple scale music segmentation using rhythm, timbre and harmony. *EURASIP Journal on Advances in Signal Processing*, 2007.

[31] M. Jian, C. Lin, and A. Chen. Perceptual analysis for music segmentation. In *Proceedings of Storage and Retrieval Methods and Applications for Multimedia*, 2004.

[32] E. Kapanci and A. Pfeffer. A hierarchical approach to onset detection. In *Proceedings of the International Computer Music Conference*, 2004.

[33] C. L. Krumhansl. A perceptual analysis of mozart's piano sonata k. 28220 segmentation, tension, and musical ideas. *Music Perception*, 1996.

[34] M. Levy and M. Sandler. New methods in structural segmentation of musical audio. In *EUSIPCO*, 2006.

[35] B. Logan. Mel frequency cepstral coefficients for music modelling. In *Proceedings of the International Symposium on Music Information Retrieval*, 2000.

[36] B. Logan and S. Chu. Music summarization using key phrases. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 749–752, 2000.

[37] H. Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proceedings of the International Symposium on Music Information Retrieval*, 2008.

[38] R. Lyons. *Understanding Digital Signal Processing*. Pearson Education, second edition, 2004.

[39] K. D. Martin, E. D. Scheirer, and B. L. Vercoe. Musical content analysis through models of audition. In *Proceedings of the ACM Multimedia Workshop on Content-Based Processing of Music*, Bristol, UK, 1998.

[40] D. McEnnis, C. McKay, and I. Fujinaga. jaudio: A feature extraction library. In *Proceedings of the International Symposium on Music Information Retrieval*, 2005.

[41] R. McNab, L. Smith, I. Witten, C. Henderson, and J. Cunningham. Towards the digital music library: Tune retrieval from acoustic input, 1996.

[42] B. S. Ong. *Structural Analysis and Segmentation of Music Signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2007.

[43] E. Peiszer, T. Lidy, and A. Rauber. Automatic audio segmentation: Segment boundary and structure detection in popular music. In *Proceedings of the International Workshop on Learning the Semantics of Audio Signals*, 2008.

[44] A. Raskinis and G. Raskini. Application of symbolic machine learning to audio signal segmentation. In *Nonlinear Speech Modelling and Applications*, pages 397–403. Springer Berlin / Heidelberg, 2005.

[45] J. Reiss and M. Sandler. Beyond recall and precision: A full framework for mir system evaluation. In *The MIR/MDL Evaluation Project White Paper Collection*. Workshop on the Evaluation of Music Information Retrieval Systems, 2003.

[46] G. Richard. Towards large databases for music information retrieval systems development and evaluation. In *The MIR/MDL Evaluation Project White Paper Collection*. Workshop on the Evaluation of Music Information Retrieval Systems, 2003.

[47] C. Sanden, C. R. Befus, and J. Zhang. Camel: A lightweight framework for content-based audio and music analysis. In *Accepted to Audio Mostly*, 2010.

[48] C. Sanden, C. R. Befus, and J. Zhang. Perception based multi-label genre on music data. In *In Proceedings of the International Computer Music Conference*, 2010.

[49] N. Scaringella, G. Zoai, and D. Mlynek. Automatic genre classification of music content: A survey. *Signal Processing Magazine, IEEE*, 2006.

[50] Shazam. Welcome to shazam, May 2010. `http://www.shazam.com/`.

[51] D. V. Steelant, B. De Baets, H. De Meyer, M. Leman, J.P. Martens, L. Clarisse, and M. Lesaffre. Discovering structure and repetition in musical audio. In *In Proceedings of Eurofuse Workshop*, 2002.

[52] Sun. Java, December 2009. `http://java.sun.com/javase/index.jsp`.

[53] W. F. Tichy. Should computer scientists experiment more? *Computer*, 31(5), 1998.

[54] G. Tzanetakis and P. Cook. A framework for audio analysis based on classification and temporal segmentation. In *EUROMICRO99*, 1999.

[55] G. Tzanetakis and P. Cook. Multifeature audio segmentation for browsing and annotation. In *Proceeding of the Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 1999.

[56] G. Tzanetakis and P. Cook. Marsyas: A framework for audio analysis. *Organized Sound*, 4(3), 2000.

[57] K. West and S. Cox. Finding an optimal segmentation for audio genre classification. In *Proceedings of the International Symposium on Music Information Retrieval*, 2005.

[58] E. Zwicker. Subdivision of the audible frequency range into critical bands (frequenzgruppen). *The Journal of the Acoustical Society of America*, 33(2):248–248, 1961.